Probabilistic Planning vs Replanning

Iain Little and **Sylvie Thiébaux**
National ICT Australia & Computer Sciences Laboratory
The Australian National University
Canberra, ACT 0200, Australia

Abstract

A theoretical comparison of probabilistic planning and replanning techniques, in the context of the planning competition. Our main contribution is a baseline test for *probabilistic interestingness*, along with some examples of its application. We also attempt an analysis of the latest probabilistic competition problems, and suggest some improvements that could be made for future instances of the competition.

Introduction

The name probabilistic planning usually refers, somewhat restrictively, to planning with probabilistic action effects, with a view to optimising the success probability of the plan. It is commonly accepted that probabilistic planning and classical planning are appropriate under rather different circumstances. For instance, classical planning, with replanning upon failure, is the option of choice when there is so little or so much uncertainty about the world that it is not worth modelling and reasoning about it. On the other hand, probabilistic planning is seen as the way to go when classical planning is not robust enough to avoid irreparable or costly failures, and when an accurate and well-confined probabilistic uncertainty model can be built.

Another popular belief about probabilistic planning is that it involves computing a plan that handles many or even all foreseeable contingencies. Or rather, such a contingent plan comes as a by-product of doing the required analysis. Therefore, probabilistic planning is particularly appropriate to generate plans for close-loop control situations, where the executive must react quickly. It also avoids solving the same subproblems over and over again. Plan reuse does not come for free with a replanning approach, although a limited contingency plan could in principle be built by a replanner over a period of time.

Replanning is an online process while probabilistic planning is typically performed offline. There are situations where there is no other choice but online planning and replanning. For instance, this occurs when goals dynamically change or when the probabilistic problem is too large to be solved at once. Symmetrically, there are critical situations where we have to resort to offline probabilistic planning; where a contingency plan *has* to be pre-generated and carefully analysed to ensure safety requirements are met. The replanning strategy can be adapted to the offline setting by recursively solving a deterministic problem for every possible contingency of the corresponding probabilistic plan. However, this would likely undermine the efficiency of planning and the quality of the plans produced.

The probabilistic track of the International Probabilistic Competition (IPC) (Younes *et al.* 2005; Bonet & Givan 2005) is a forum where planning systems compete to solve probabilistic planning problems described in a probabilistic extension of PDDL (Younes *et al.* 2005). The mode of evaluation, by execution of the plan over a number of trials, enables both probabilistic planners and replanners to compete. Following the report that the replanner FF-replan (Yoon, Fern, & Givan 2007) outperforms all the probabilistic planners featuring in the two editions of the IPC (2004,2006), the commonly accepted view of the territories covered by probabilistic planning and replanning has been shaken.

Here, we seek to more finely characterise the boundaries between the planning and replanning approaches to solving probabilistic planning problems. Our main contribution is a declarative test for when a problem is 'probabilistically interesting'. Problems that fail this test are classical planning problems in disguise, and for those, replanning is definitely the technique of choice. When combining the definition with an understanding of how a replanner makes its choices, it is trivially easy to design problems where this replanner will perform both incredibly well, and incredibly badly.

We illustrate this definition with some simple examples as well as some larger instances of probabilistically interesting problems. These exhibit structures that present an insurmountable challenge to a replanning strategy, showing that probabilistic planning isn't quite a lost cause. We then examine the domains and instances of the IPC-5 probabilistic track. We show that the problem instances for 5 out of the 9 domains satisfy the baseline test for interestingness. In these domains, the main reason why replanners can do comparatively well on the chosen problem instances is that the smaller instances tend to be adequately or even optimally solvable via replanning, the larger ones are often too large for current probabilistic planners to get any result at all, and the crucial intermediate region is hardly covered.

We should say upfront that the primary focus of our analysis is the case where the goal satisfaction probability is the only criterion for evaluating the quality of plans. This is essentially what ended up being the case in the IPC-5. But this competition still featured problems that are best seen as stochastic shortest path problems, and an extension of our analysis based on this perspective would also be of value.

Probabilistic Planning

For the purpose of this analysis, we define a probabilistic planning problem as:

- a finite set of states S;
- an initial state $s_0 \in S$;
- a set $G \subseteq S$ of (absorbing) goal states;
- a set O of outcomes; the probability of $o \in O$ is $\Pr(o)$;
- a (total) deterministic transition function $T(o, s) \in S$ for all outcomes $o \in O$ and states $s \in S$; and
- sets $A(s)$ of applicable actions for each $s \in S$, together with a function $out(a) \subseteq O$ mapping each action to a set of outcomes in such a way that (i) each outcome $o \in O$ belongs to exactly one action $act(o)$, and (ii) $\sum_{o \in out(a)} \Pr(o) = 1$ for all a.

A *trajectory* is a possibly infinite sequence $\tau = o_1, o_2, \ldots$ of outcomes such that the respective actions are applicable in sequence from the initial state, i. e. for all $i \in 1, 2, \ldots$ $act(o_i) \in A(T(o_{i-1}, \ldots T(o_1, s_0)))$. A finite trajectory $o_1, \ldots, o_n$ reaches the goal if $T(o_n, \ldots T(o_1, s_0)) \in G$. The probability $\Pr(\tau)$ of a trajectory is the product of the probabilities of its outcomes.

A *contingency plan* $\pi : S \mapsto A$ is a (partial) mapping from states to actions such that $\pi(s) \in A(s)$. Such a plan induces a possibly infinite set of trajectories $\mathcal{T}(\pi) = \{o_1, o_2, \ldots \mid o_i \in out(\pi(T(o_{i-1}, \ldots T(o_1, s_0))))\}$. We write $\mathcal{T}_G(\pi)$ for the subset of those that are goal trajectories.

The goal-satisfaction probability $G(\pi)$ of plan π is the sum of the probabilities of the goal trajectories induced by π: $G(\pi) = \sum_{\tau \in \mathcal{T}_G(\pi)} \Pr(\tau)$. Plan π' is better than plan π iff $G(\pi') > G(\pi)$. A contingency plan is optimal if no other plan is better.

A probabilistic contingent planner, or, perhaps abusively, probabilistic planner for short, takes as input the probabilistic planning problem description, and builds and outputs a (possibly optimal) contingency plan for the problem.

Replanning

For the purpose of this paper, the object of a replanner is to solve a probabilistic planning problem through the exclusive use of a deterministic planner. This involves the compilation of a probabilistic problem into a deterministic one. We assume that the deterministic problem has the following characteristics:

- a finite set of states S^d;
- an initial state $s_0^d \in S^d$;
- a set $G^d \subseteq S^d$ of goal states;
- a set $A^d(s)$ of applicable actions for each $s \in S^d$;
- a deterministic transition function $T^d(a, s) \in S^d$ for all actions $a \in A^d(s)$ and states $s \in S^d$; and
- a cost $c^d(a)$ for each action a.

Here, a trajectory is a sequence of actions, applicable from the initial state. Goal trajectories are defined similarly as before; these are the solutions to the deterministic planning problem. The cost $C(\tau)$ of a trajectory is the sum of the costs of the respective actions: $\sum_i c(\tau_i)$.

Compilations

A *compilation* from probabilistic to deterministic planning problems is a function Δ satisfying the following properties: for each probabilistic planning problem $P = \langle S, s_0, G, O, T, A\rangle$ and its compilation $\Delta(P) = \langle S^d, s_0^d, G^d A^d, T^d, c^d\rangle$, there are two functions $\sigma : S^d \mapsto S$ and $\alpha : A^d \mapsto O$ such that:

1. $\sigma(s_0^d) = s_0$;
2. $s \in G^d$ iff $\sigma(s) \in G$; and
3. for each trajectory $a_1, a_2, \ldots$ in $\Delta(P)$, there is a corresponding trajectory $o_1, o_2, \ldots$ in P such that for all i
   (a) $\alpha(a_i) = o_i$, and
   (b) $\sigma(T^d(a_i, \ldots T^d(a_1, s_0^d))) = T(o_i, \ldots T(o_1, s_0))$.

These properties ensure the correctness of the plans produced by the replanner. Moreover, we say that a compilation *preserves trajectories* iff the reverse of the third condition above hold: for each trajectory $o_1, o_2, \ldots$ in P there is a corresponding trajectory $a_1, a_2, \ldots$ in $\Delta(P)$, such that for all i (a) and (b) above hold. Preservation of trajectories is related to the completeness of the replanning approach.

Two Specific Compilations

In the following, we will consider compilations that leave the set of states intact, that is $S^d = S$, and $\sigma(s) = s$. The interesting parts of these compilations are the derivations of the action set, the transition function, and of the cost function of the deterministic problem from the probabilistic one. In this paper, we focus on the two following compilations:

$\Delta 1$: only the most probable outcome (ties are broken arbitrarily) of each probabilistic action is kept to form the actions of the deterministic problem: $A^d(s) = \{o \in O \mid \exists a \in A(s) \text{ s.t.} act(o) = a \text{ and } o = \arg\max_{o' \in out(a)} \Pr(o')\}$. This compilation does not preserve trajectories. It makes sense when the highest probability outcome corresponds to the 'successful' execution of an action, and all other outcomes correspond to a 'failure'.

$\Delta 2$: there is exactly one deterministic action built per outcome of a probabilistic action: $A^d(s) = \{o \in O \mid \exists a \in A(s) \text{ s.t. } act(o) = a\}$. Obviously, this compilation preserves trajectories.

In both cases $T^d = T$. We additionally consider two possibilities for the cost function of the deterministic problem:

shortest: $c^d(a) = 1$. Here, trajectories are preferred according to their length.

most-likely: $c^d(a) = -\log \Pr(\alpha(a))$. Here, trajectories are preferred according to their probabilities. Since probabilities are multiplicative and in the $[0, 1]$ interval, the standard trick to turn them into positive additive costs is to use negated logarithms. This function leads an optimal deterministic planner to generate the most likely trajectory to the goal, among those allowed in the deterministic problem.

One can easily imagine more complex ways of discriminating between trajectories, for instance by using probabilities as the primary criterion and length to break ties.

Replanner

A replanner, such as those we consider, is a type of online planner which uses a deterministic planner to produce a solution trajectory for the compiled problem. It attempts to execute this solution. Whenever the current state deviates from the expected one, it replans, generating a new trajectory from the current state to the goal.

A replanner naturally induces a contingency plan, by mapping each state to the first action in the trajectory that the replanner would generate from that state. Only a few of the trajectories induced by this contingency plan will typically be generated by the replanner. In general this plan is suboptimal, even if generated with an optimal deterministic planner.

Below, we will consider the following replanners, which are both based on an optimal deterministic planner, i. e., one that generates the trajectory τ with minimal cost $C(\tau)$. This optimality assumption is mainly to simplify our analysis of the IPC problem instances:

REPLAN1: uses compilation $\Delta 1$ with cost function *shortest*. Except for the assumption of optimality of the underlying planner, the version of FF-replan that was entered in the IPC-4 is an implementation of REPLAN1. FF-replan is based on FF (Hoffmann & Nebel 2001), a suboptimal planner.

REPLAN2: REPLAN2 (shortest) uses compilation $\Delta 2$ with cost function *shortest*; REPLAN2 (most-likely) uses compilation $\Delta 2$ with cost function *most-likely*. Except for the optimality assumption, the 2006 version of FF-replan which was evaluated on the IPC-5 problems is an implementation of REPLAN2 (shortest).

Probabilistically Interesting Problems

We have identified a number of structural properties that probabilistic planning problems can have which are useful when comparing probabilistic planning with replanning. These include:

1. the presence—or lack thereof—of 'dead end' states; states from which the goal is unreachable through any combination of chance and choice,
2. the degree to which the probability of reaching a 'dead end' state can be reduced through the choice of actions,
3. the number of distinct trajectories from which the goal can be reached from the initial state, and
4. the presence of mutual exclusion; of choices that exclude other (useful) courses of action later.

(Avoidable) Dead Ends

Dead end states are a fundamental feature of probabilistic planning problems; it is what distinguishes probabilistic from conformant planning. When one considers how the replanning strategy interacts with planning problems that do not have any dead ends, it becomes immediately obvious that—assuming that there are no other constraints—a replanner will **always** reach the goal with 100% probability, as there is no possible deviation from a planned deterministic trajectory from which the goal becomes unreachable. Moreover, if the probability of achieving the goal is the only evaluation criteria, then both a probabilistic planner and a replanner will produce the same quality (i. e. 'perfect') solutions. The only difference will be in how long the respective planners take to achieve the goal. In any such setting where finding some goal trajectory is much faster than finding an optimal contingency plan, it is likely that the replanner will be faster. For planning problem where there are dead ends, it is possible that a probabilistic planner will be able to produce a higher quality solution than a replanner. The replanner is still likely to achieve the solution that it does find more quickly, however.

We distinguish between *avoidable* and *unavoidable* dead ends. An unavoidable dead end is a dead end state where for *any* given plan:

1. there is a positive probability of reaching it when executing the plan, and
2. the probability of reaching the state cannot be reduced without also reducing the probability of achieving the goal.

Conversely, an avoidable dead end is a dead end state with a positive probability of being reached when executing *some* plan that can be reduced without also reducing the probability of reaching the goal.

We view unavoidable dead ends as uninteresting, in a probabilistic sense. The reason for this is that they are a feature that makes a problem look like a probabilistic problem, but its structure is such that—as with problems that have no dead ends at all—it is impossible to improve on the solution that one finds when using a replanning strategy (again, assuming that the probability of achieving the goal is the only evaluation criteria).

Multiple Goal Trajectories

The number of different ways in which the goal can possibly be achieved is another important property when considering the suitability of planning vs replanning for a given problem. In the extreme case when there is only a single way of achieving the goal, it is clear that a replanning strategy will (by definition) find it. Moreover, in problems with only a single goal trajectory, a probabilistic planner just doesn't have anything to work with. There is no point in trying to optimise the contingent probability of success when there aren't even any contingencies!

It might seem that the more goal trajectories there are, the better a probabilistic planner will compare to a replanner. This is not necessarily the case. In fact, if there are too many different possibilities, then a probabilistic planner might get bogged down trying to find the best way of integrating all of them, while a replanner might quickly and directly find a decent (and possibly even optimal) solution.

And more to the point: while multiple goal trajectories are a necessary condition for a problem to be considered probabilistically interesting, it is not sufficient. Consider, for example, the case when there are no common actions between any pair of goal trajectories. Again, a probabilistic planner isn't needed to construct a contingency plan if there is no possibility of needing to plan for contingencies.

We expand the previous statement to formulate the first two conditions of our definition of probabilistic interestingness. *A probabilistically interesting planning problem includes: (1) multiple goal trajectories; and (2) at least one* *pair of distinct goal trajectories, τ and τ', that share a common sequence of outcomes for the first $n-1$ outcomes, and where τ_n and τ'_n are distinct outcomes of the same action.*

The second condition is really a refinement of the first, since it cannot be satisfied unless there are multiple goal trajectories; it specifies the minimum structural requirement for it to be possible to find a solution that actually contains contingencies. We shall nevertheless keep the first condition to make it clear that multiple goal trajectories are necessary.

Mutual Exclusion

To further develop our definition of a probabilistically interesting planning problem, we now return to the idea of 'avoidable dead ends'. For it to be possible that there be avoidable dead ends, there has to be a degree of mutual exclusion in the problem structure. Specifically, there have to be *choices* as to which course of action to take that can—either *potentially* or *necessarily*—exclude alternative choices. This can be seen by going back to the conceptual definitions of avoidable and unavoidable dead ends; for if there is no possibility of excluding potential courses of action, then eluding dead ends is entirely due to *chance*, and not influenced by choice.

We now state these ideas more formally. *A probabilistically interesting planning problem additionally satisfies the property that: (3) there exist two distinct goal trajectories τ and τ' and outcomes $o \in \tau$ and $o' \in \tau'$ of two distinct actions $a = act(o)$ and $a' = act(o')$ such that executing a strictly decreases the maximum probability*[1] *of reaching a state where a' can be executed.*

As a useful aside, a property that is sufficient to show the presence of mutual exclusion is the presence of non-reversible exclusive choices. This is a stronger condition than what we require for a problem to be considered probabilistically interesting that can arise from either propositional or metric resource constraints. In particular, if one can show that there are exclusive sets of actions (that have goal trajectory outcomes) such that executing any action on one set permanently excludes the possibility of executing any action in another set, then this is sufficient to show the presence of mutual exclusion.

Putting It All Together

Definition 1 (Probabilistically Interesting Problem)**.** *A probabilistic planning problem is considered to be 'probabilistically interesting' if and only if it has all of the following structural properties:*

1. *there are multiple goal trajectories;*
2. *there is at least one pair of distinct goal trajectories, τ and τ', that share a common sequence of outcomes for the first $n-1$ outcomes, and where τ_n and τ'_n are distinct outcomes of the same action; and*
3. *there exist two distinct goal trajectories τ and τ' and outcomes $o \in \tau$ and $o' \in \tau'$ of two distinct actions $a = act(o)$ and $a' = act(o')$ such that executing a strictly decreases the maximum probability of reaching a state where a' can be executed.*

[1]To clarify: executing a rules out any plan with a maximal probability of executing a'.

We assert that unless a probabilistic planning problem satisfies all of the structural conditions in this definition, then it is inevitable that a well-written replanner will outperform a well-written probabilistic planner. Unless a probabilistic problem contains these structural properties, then it is effectively a deterministic planning problem in disguise.

It is important to remember that it is possible that a replanner will perform optimally even for probabilistically interesting planning problems. In fact, this will occur whenever attempting the 'most promising' goal trajectory is the correct thing to do, which can occur quite often for constructed or generated problems. When combining the definition with an understanding of how a replanner makes its choices, it is trivially easy to design problems where this replanner will perform both incredibly well, and incredibly badly.

There looks to be a true challenge when deciding what are the appropriate structural properties to include in the problems for future probabilistic planning competitions. We consider our definition of a probabilistically interesting planning problem to be a useful tool to aid in this process.

Demonstration Problems

To demonstrate the application of the preceding theory, we have created a number of very simple problems that explore the issue of probabilistic planning vs replanning. We start with a probabilistically uninteresting problem that is not solvable by FF-replan and continue with two problems that are probabilistically interesting in different ways. We have also identified a series of `tireworld` problems as an example application of the theory to scalable problems. Experimental results obtained by various IPC planners for these problems are given in Table 1. The PDDL source for these problems can be found at `http://rsise.anu.edu.au/~thiebaux/benchmarks/pddl/`.

Climber

The first problem is called `climber`, and is described by the following story: *"You are stuck on a roof because the ladder you climbed up on fell down. There are plenty of people around; if you call out for help someone will certainly lift the ladder up again. Or you can try the climb down without it. You aren't a very good climber though, so there is a 40% chance that you will fall and break your neck if you do it alone. What do you do?"*

The `climber` problem consists of the actions `climb-with-ladder`, `climb-without-ladder` and `call-for-help`. There are two solution trajectories: the short path, where one climbs without the ladder but risks a 40% chance of dying; or the slightly longer path where one calls for help then climbs with the ladder, which has no risk at all. This problem is not probabilistically interesting because it violates the second condition of Definition 1. As ridiculously simple as this problem is, it is still not (optimally) solvable by either REPLAN1 or REPLAN2 (shortest). In general, the alternative that a replanner will choose depends on how it measures the cost of goal trajectories; whether by length or probability. Our optimal planners REPLAN1 and REPLAN2 (shortest) would both choose the option with a 40% chance of dying. So would both versions of FF-replan.

	climber	river	bus-fare	tire1	tire2	tire3	tire4	tire5	tire6
OPTIMAL	100%	65%	100%	100%	100%	100%	100%	100%	100%
REPLAN1	60%	50%	1%	50%	13%	3%	1%	0%	0%
REPLAN2 (shortest)	60%	50%/65%	1%	50%	13%	3%	1%	0%	0%
REPLAN2 (most-likely)	100%	50%	1%	50%	13%	3%	1%	0%	0%
FF-replan	60%	65%	1%	50%	0%	0%	0%	0%	0%
FPG	100%	65%	22%	100%	92%	60%	35%	19%	13%
Paragraph	100%	65%	100%	100%	100%	100%	3%	1%	0%

Table 1: Results for the demonstration problems. **OPTIMAL**, **REPLAN1** , **REPLAN2 (shortest)**, and **REPLAN2 (most-likely)** show the theoretically expected success percentages for an optimal probabilistic planner and for the various replanners we consider in this paper. The bottom part of the table shows experimental results obtained with IPC-5 planners: FF-replan (Yoon, Fern, & Givan 2007) is a replanner, FPG (Aberdeen & Buffet 2007) is a suboptimal probabilistic planner, and Paragraph (Little & Thiébaux 2006) is an optimal probabilistic planner. Each planner was given 10 minutes to solve each problem. The `river` result for FF-replan changes to 50% if the action order is reversed in the domain description. Paragraph's results for `tire4`, `tire5` and `tire6` reflect the solutions that were found in 10 minutes; they improve as the time limit is increased.

River

The second problem is called `river`, and its story is as follows: *"You are on one side of a river, and want to get to the other side. There are some rocks that look like they could be traversed. They are slippery though, so there is a 75% chance you would slip and fall. If that happened, there would be a 1 in 3 chance that you would drown in the current, but you would probably be able to make it to a small island in the middle of the river. An as alternative, there is a place further down the river where you might be able to swim across. The current is strong through, so you give yourself an even chance of making it. If you could get to the island you would have a better chance, around 80%. There is no way of swimming there directly, though; the current is just too strong. What do you to to maximise your chance of getting to the other side without drowning?"*

This problem describes a slightly more complicated situation than the first. There are three goal trajectories: swimming across the river with a 50% chance of success, traversing the rocks with a 25% chance of success, and slipping on the rocks then swimming from the island with a 40% chance of success (50% chance of making it to the island, then 80% chance of swimming from the island). This problem is fundamentally based on an exclusive choice between traversing the rocks and swimming across the river. Because the highest probability trajectory is through initially swimming, that is the option that a replanner that optimises trajectory probabilities will choose. However, the optimal solution is to attempt to traverse the rocks, as the contingent probability of success is 65%, the sum of both goal trajectory probabilities that are associated with this option.

For this problem, a replanner that optimises trajectory length might find the optimal solution, but then again it might not. If the replanner happens to choose to swim initially, then the solution won't be optimal. However, if it was lucky enough to choose the (equal length) trajectory where one traverses the rocks, then the replanner will correctly swim from the island when that contingency arises.

This problem is probabilistically interesting, having multiple goal trajectories, contingencies and mutual exclusion. The `river` problem embodies the essence of situations where goal trajectory probabilities (or lengths) do not correlate to contingent probabilities of success. We believe that this is the most important distinction between problems that can be solved satisfactorily with a replanner, and problems that require full probabilistic reasoning.

Bus Fare

The final simple problem is called `bus-fare`: *"You are at a bus stop, and need to buy a bus fare to get home. Unfortunately, you only have 1 dollar, and you need 3 dollars to buy the bus fare. There is a man sitting to the side of the bus stop playing with some dice, and he agrees to give you the money for the bus fare if you correctly guess the roll of a 10-sided die twice in a row. However, he doesn't do things for free, and charges a dollar each time you want to try this. There is another man who is raising some beer money by washing car windscreens at a nearby intersection. He is willing to take a break and let you take over for a time, but requires compensation for lost earnings. Since the amount that the drivers pay is unpredictable—typically 1 or 2 dollars—the car washer wants you to empty your pockets before washing each car, but will let you keep the earnings. How can you get home with the highest probability?"*

This problem is structured as follows: (1) if at any time you run out of money, then it is game over, (2) if you have 1 dollar, then you have the option of a single shot at winning, but with only a 1% success probability, (3) it is always possible to (eventually) get 2 dollars as long as you have some money, and (4) if you have 2 dollars, you have the same 1% shot at winning, but it isn't game over if it doesn't work.

This problem is a more extreme example of trajectory probabilities not corresponding to full contingent probabilities, because the low probability trajectory can be tried again and again until it works, and the (marginally) higher probability trajectory can be tried only once. While this particular problem might seem unlikely, it serves as a microcosm of resource management scenarios where it is important to keep something in reserve. It is also intended to demonstrate that it doesn't take a large number of actions for the penalty of *not* using full probabilistic reasoning to become obscenely large. All that is needed is for there to be a greater degree of repeatability for the contingencies on the 'bad trajectory' branch of an exclusive choice.

Triangle Tireworld

To give a demonstration of how the theory can be applied to scalable problems, we have identified a series of problems for the IPC `tireworld` domain. The basic idea is that a car can move between different locations via (directional) roads, with the goal being to get from a 'start' to an 'end' location. However, for each move between locations there is a chance of getting a flat tire; so the idea is to find the shortest path to the goal to maximise the probability of success. To complicate matters, it is possible to replace a flat tire with a spare. Some locations contain a spare, and the car itself is equipped to carry a single spare. This means that the best course of action can depend on which route has the highest proportion of 'spare' locations, and also whether or not the car is already carrying a spare.

Significantly, versions of this domain have featured in both probabilistic planning competitions. In the most recent competition, it was perhaps the only domain where probabilistic planners got systematically better results than replanners. The problem instances, however, were randomly generated, and did not contain much structure. They did not reflect the typical performance gap we expect to see between probabilistic planners and replanners on this domain. Here, we demonstrate one possible way of systematically constructing a series of problems with exponentially larger state spaces that are guaranteed to be probabilistically interesting.

There are two ways in which this domain was modified between IPC-4 and IPC-5. The first is that the action for replacing a flat tire was given a probability of failing, and the second was that the probability of getting a flat tire when moving was increased. Neither of these changes affect how probabilistically interesting this domain is, although the first affects the size of the search space. The probability of getting a flat tire can also affect whether or not it is worth going on a detour to pick up a spare tire, and how close a replanner's solution will be to the optimal.

For this problem series, we have made the probability of getting a flat tire 50% for each move, and have re-eliminated the probability of needing to retry the replace tire action. We also decided to prevent the car from storing a spare tire. This change was primarily to make a completely optimal solution easier to distinguish from suboptimal ones, but also to simplify the analysis of individual problems. As it is, there is a 50% chance of getting stuck for every 'mistake' that is made. If the car was able to carry a spare then the penalty for making a mistake is potentially much less than this. It does not make this domain less probabilistically interesting, and does not affect a replanner's performance in any significant way.

We refer to this series of tireworld problems as `triangle-tireworld`, in reference to the fundamental shape that it is patterned after. A visual representation of the first three problems is given in Figure 1. The series has the following significant properties:

1. there is a single optimal solution with a 100% success rate to every problem, and the next best solution(s) will have a 50% success probability;

2. the difficulty of finding the optimal solution increases exponentially with each problem; and

3. a solution based on the shortest (or most likely) trajectory will have a success probability of $0.5^{(2n-1)}$, where n is the problem number.

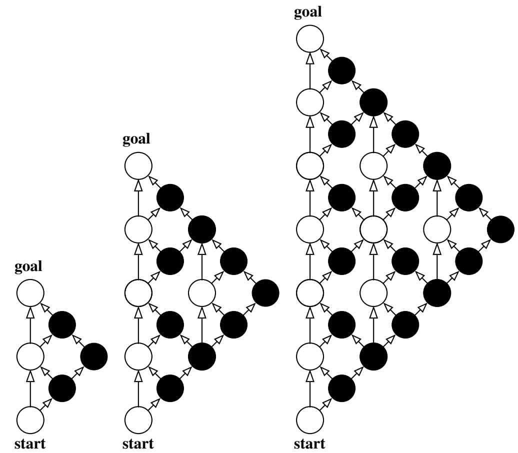

Figure 1: The first three triangle tireworld problems. White locations do not have a spare, while black locations do. The roads are all 'one way'.

For example, in the 3rd problem there are five trajectories that imply 50% solutions, ten 25% solution trajectories, and many more lower probability solutions. Each problem's optimal solution is based on the longer route around the outer edge of the largest triangle.

This series of problems is expected to provide a challenge even for many probabilistic planners. Certainly a replanning strategy is not expected to find a satisfactory solution. Even a replanner that was extended to do basic probabilistic reasoning—for example by using a look-ahead—would be expected to find a progressively worse solution for each successive problem. This is because a completely optimal solution to any problem in this series must be based on one of the longest possible (and lowest probability) 'best case' trajectories, when there are numerous goal trajectories that are both shorter and have a higher probability.

Competition Problems

In 2004 and 2006, the International Planning Competition featured a probabilistic track. The goal of this track is "to assess the state of the art in probabilistic planning, to evaluate and motivate research in the field, and to identify lines for future research" (Bonet & Givan 2005).

Planners competing in the probabilistic track are given a series of problem instances from various domains described in a probabilistic extension of PDDL (Younes *et al.* 2005). They are not required to produce an explicit solution, but are instead evaluated by completing, for each instance, a certain number of random trials within a certain amount of time. Each trial corresponds to a possible trajectory of the underlying contingency plan: starting from the initial state, the planner supplies an action, the evaluator randomly chooses one of the possible resulting states, the planner supplies an action for that state, and so on. The trial ends when a goal state is reached, when the planner explicitly abandons the trial (typically because it has identified the current state as a dead end), or when the number of steps taken exceeds a large bound. The number of success trials, as well as the run time and cost for each trial are recorded. In the IPC5, cost was measured as the length (number of steps) of the trial.

The above mode of evaluation opens the possibility for replanners to compete in the probabilistic track. In 2004, a version of FF-replan using compilation $\Delta 1$ (i. e. it ignores any but the most likely effect of each operator), was entered in the competition. This was a useful initiative to remind the organisers that problems which have no dead ends, and for which action costs are irrelevant, do not require reasoning about uncertainty at all. As expected, FF-replan won the sub-track that consisted of deadend-free goal-achievement problems, but was ineffective on probabilistically interesting benchmarks. Since the first type constituted the vast majority of problems, it was declared the overall winner.

In 2006, FF-replan did not enter the competition, as one of its authors was one of the organisers of the track. Nevertheless, it was reported that a version of FF-replan using compilation $\Delta 2$ (i. e. it splits each probabilistic action into one action per probabilistic outcome), was still able to significantly outperform all competition participants. At first glance, this seemed much more surprising, as there had been some effort to include many more domains with dead ends and which, a priori, required reasoning under uncertainty.

As those results tend to positively reflect on replanning, and rather negatively on probabilistic planning, it is important that they be carefully analysed. Otherwise, they might cause an inaccurate perception of the worth of the respective approaches, and maybe even precipitate the abandonment of promising avenues of research. In this section, we present a first attempt at such an analysis.

Probabilistically Interesting Domains

We start with the 5 competition domains which have probabilistically interesting instances. Each of these domains has potential, but we believe that the instance selection should be improved.

Drive This domain describes a path planning problem in a grid of locations. To move between the locations, it is necessary to risk dying. The actual risk of death is a function of several variables (the direction in which the car is heading, light settings, length of segments), which determine how many times the dangerous actions (e. g. waiting for a traffic light to turn green or proceeding through a segment) need to be repeated to successfully move. Some risk is unavoidable, but some paths through the grid have a slightly higher risk than others.

The problems for this domain satisfy the definition of a probabilistically interesting problem, but this needs to be interpreted carefully. Most importantly, it needs to be understood that there is a relatively small difference between the best possible route and the worst possible route. This is to the point that the extra computation needed to work out which route is optimal might only make a small difference in the actual results. And with the competition evaluation methodology, this can easily be cancelled out by getting unlucky with statistical variance.

In short, replanners using the $\Delta 2$ compilation can be expected to do reasonably well on problems for this domain, even though one would expect a probabilistic planner that could handle large enough problems to have a noticeably higher success rate at least some of the time. The $\Delta 1$ compilation is ineffective here, as the most likely outcomes implement the less desirable effects of actions.

Exploding Blocksworld This is a domain that appeared in the 2004 competition, with slightly different probabilities. This is a stochastic variant of the blocks world with 4 operators (pick-up, pick-up-from-table,put-down, put-down-on-block) where blocks are set to detonate. When putting down an undetonated block onto another block or the table, the block detonates with probability 1/10 (resp. 2/5), resulting in the object it is being put onto being destroyed and not available any longer to achieve the goal. Once detonated, blocks behave as normal. The encoding has flaws as it enables a block to be put onto itself, which makes it easy to get rid of irrelevant blocks without destroying anything else.

The domain is probabilistically interesting and was in fact especially designed for a replanning strategy to perform poorly. The reason why FF-replan performs better than its competitors here is twofold. The small 5-blocks problem instances selected, to which probabilistic planners can easily scale, are trivial. They are solved optimally by all the replanners we consider, including FF-replan, and by all IPC-5 probabilistic planners.[2] Moreover, probabilistic planners, unlike (suboptimal) deterministic ones, have trouble scaling to the next size (10 blocks), for which it is easy to randomly generate difficult instances. So from 10 blocks onwards, we have a situation were replanners are killed by their inability to do probabilistic reasoning, and probabilistic planners by their inability to scale.

Pitchcatch This domain looks to be based on baseball ideas, with a cycle between pitching and catching phases. The goal is to 'deposit' a ball for each of a given set of ball types. We note that this is another domain for which the $\Delta 1$ compilation is inappropriate.

The problem instances for this domain are probabilistically interesting. Multiple goal trajectories are a given, several of the critical actions have multiple outcomes from which the goal is reachable, and there is an interaction between at least two of the actions (setting an individual 'bit' and 'pitching') that allows choice to minimise the probability of 'dying' to a degree.

Although it is clear that the probability of dying can be reduced through action selection, it is not clear how big the difference between planning and replanning is for this problem. Most of these problems are too large to be solved effectively by any of the probabilistic planners entered into the competition. The problems for this domain have a search space that blows up fairly quickly, due to the number of possible probabilistic outcomes. There aren't that many things that one can do, however, irrespective of which outcome occurs. It might be that the potential for improvement over a replanning solution for these problems is minimal.

[2] The difference across planners on P04 is due to the variance over a small number of trials. The optimal success probability for this problem is 60%.

Schedule This domain is based on the concept of packet scheduling. The basic idea is that packets are 'arriving' and need to be 'served' before they are 'dropped'. If a packet is dropped, then we 'reclaim' it, but there is an chance of dying. The arrival of packets in determined probabilistically, one minimises the chance of dying by optimising the order in which packets that have arrived are dealt with. As in Drive and Pitchcatch, the $\Delta 1$ compilation is not appropriate for this domain.

It is the probabilistic arrival of multiple packets, along with the potential to minimise the chance of dying that makes the problems of this domain probabilistically interesting. The smaller problems do not exhibit much of this potential, and the need for probabilistic reasoning is minimised by having fewer classes of packets than time steps before a packet is dropped. The 6th problem is where the probabilistic potential starts to be realised, and the 11th problem onwards are particularly good. Unfortunately FF-replan cannot handle them, and the probabilistic planners cannot scale well enough to handle them either.

We believe that this selection of problems would have been improved if all problems had more packet classes than time steps before dropping a packet. This would have made probabilistic reasoning a requirement right from the start, and not just for the later problems.

There was also an issue with some successive instances being highly similar, the only difference being slightly different probabilities for some of the outcomes. In fact, some of the problems were identical to one another: problem P06 is exactly the same as problem P07, problem P09 is exactly the same as problem P10, and problem P11 is exactly the same as problem P12. This shows that more care needs to be taken when randomly generating problems.

Tireworld We gave an overview of that domain earlier. To recap, there are three actions: moving the car between connected locations, loading a spare tire, and replacing a flat tire with a spare. In the version used for the latest planning competition, the probability of getting a flat tire when moving is 40%, and there is also a 50% probability of failing to replace a flat tire (without penalty, so the action can always be repeated).

In the set of generated problems that was used, a number of locations were randomly connected by (bi-directional) roads, and spares assigned to random locations. All problems generated for this domain are probabilistically interesting. Specifically, they all contain multiple goal trajectories, they all have the potential of getting to the goal irrespective of whether certain individual movement actions have a flat tire or not, and there is definitely a possibility of getting stuck in a dead end that can be reduced by choosing a different route.

Here FF-replan doing well is due to a combination of factors. The first is that despite their definitional interestingness, the problems instances are rather trivial. 3 of them have single step solutions (P02, P10, P12) and together with another (P04) are solved optimally by all replanners considered in this paper. Moreover, 14 out of 15 problems (all except for problem 1) have shortest goal trajectories of three steps or less. This leads to a reasonably high chance that FF-replan could get lucky, which happens for instance on P05, P11 and P13. Finally, there are some anomalies between the theoretically expected performance of FF-replan and the official competition results on some problems that is large enough not to be explainable by statistical variance. This showed up most clearly for problems P07 and P08. An investigation of this in collaboration with Sungwook Yoon revealed that there must be a bug in the competition server. We are unsure as to whether or not this affected any other domains or competitors, but do not believe that it affected the competition results in any significant way.

Probabilistically Uninteresting Domains

Finally, we end with the four domains that do not have dead ends and are therefore probabilistically uninteresting. Viewed from a goal-satisfaction probability angle, these problems are best solved by a replanner. However, it would be useful to know whether they would be more interesting, if viewed as stochastic shortest paths problems (SSPP) instead. In an SSPP, there exists at least one contingency plan with goal-satisfaction probability 1, and the problem is to find one such plan with minimum expected cost (which we take to be length here). The analysis in this paper does not cover SSPPs yet. Nevertheless, we shall informally report observations related to the potential suitability of the competition domains and instances for that purpose.

Blocks World This is an extension of the traditional blocks world with 4 operators. The first extension is that 3 of the basic operators have a 1/4 probability of failing, changing nothing or dropping the block held on the table. The second extension is the presence of 3 actions that manipulate towers of blocks at once; 2 of those have a very high probability 9/10 of failing, again either changing nothing or putting the tower on the table. Unfortunately, the domain description used in the IPC is flawed, and leads to inconsistent states. It is not possible to fix the problem without resorting to quantified preconditions or to using 'above' as a predicate. Below we are assuming a correct version.

REPLAN2 (shortest) often does *not* find optimal plans here, as minimal-length trajectories may use highly unlikely outcomes which will lead to a large number of repeats of failed actions. This was often the case in the competition where FF-replan generated longer plans than its competitors.

On the other hand, REPLAN1 and REPLAN2 (most-likely) will generate the plan with minimal expected length for most problems of reasonable size. This happens because (1) the 'failure' outcomes of the basic actions are not needed to solve the SSPP optimally, and (2) the unlikely 'success' outcomes of the tower actions can only help when one can move towers of 8 blocks at once. In an n-block random instance of the IPC, the expected tower height is $\sqrt{n}$ (Slaney & Thiébaux 2001), so it takes large instances for this to happen in non-pathological cases; much larger than even domain-independent deterministic planners can handle. In particular, all 5-blocks IPC instances can be solved optimally by REPLAN1 and REPLAN2 (most-likely), and the optimal planner underlying our implementation of REPLAN2 could not scale to the next size (10 blocks) instances.

Altogether, we do not expect to find challenging Blocks World SSPPs that are of manageable size for current planning systems. This means that the domain is probably better suited to deterministic replanning than probabilistic planning at this stage.

Elevators Someone needs to collect a number of coins at various places in a two dimensional building. Vertically, a number of elevators enable getting from one floor to the next. Elevator shafts can be located at several locations of the horizontal dimension. There are actions for an elevator going up and down, and for getting in or or out of an elevator. On each floor, one can walk to the left or right. The only uncertainty is at a so called 'gate' location, where one falls to the leftmost location on the bottom floor with probability 1/2. Clearly, the goal is always reachable, and so the problem is not probabilistically interesting.

When viewed as a SSPP, none of the replanners we considered can solve the problem optimally in general. In brief, this is because optimal solutions can sometimes use the gate uncertainty to their advantage. Other observations concern the instances. The first is that the size of the competition instances chosen seems reasonable in this domain. The second is that just looking at the 10 first SSPP instances, the first 7 are solved optimally by REPLAN1 as there is no need to use gates, all 7 but the first are solved optimally by REPLAN2 (shortest), and the first 5 are solved optimally by REPLAN2 (most-likely).

Random We have not analysed this domains in detail. A precise analysis is made difficult because there is no generic PDDL domain description for Random, but just a random generator of domain/problem instances. A Random domain is obtained by generating predicates and action templates with random arities, fleshing out the latter by randomly generating preconditions and probabilistic effects. Then, initial and goal states are selected in such a way as to guarantee that the problem has at least one goal trajectory with a maximum length of 100 and whose probability is above a certain threshold (20%). Finally, FF is used to check that some goal trajectory can be found easily (within less than 10 ms) – those problems for which it can't are discarded. To rule out the existence of dead ends, there is a pair of deterministic reset actions which, when applied in sequence, return one to the initial state from any other. The first action resets to a state where all propositions are false, and the second from that state to the initial one.

It is worth noting that the size of the problem instances generated is huge. P01 for example has about 300 atomic propositions and 4000 outcomes. This did not prevent FF-replan and to a large extent FPG to do well in terms of achieving the goal. However, we conjecture that the IPC random instances viewed as SSPPs will prove quite difficult to solve optimally: our implementation of REPLAN2, which is based on an optimal planner couldn't scale up to P01; an implementation of forward search using the admissible h^1 heuristic did not manage to find a goal trajectory for that same problem within 2 hours.

So altogether, it would seem that Random shows promise as an SSPP, but the size of the instances should be decreased.

Zeno This domain was (in a very similar form) also present at the IPC-4, whose organisers pointed out that it "presented no real challenge because [they] neglected to include action costs" (Younes *et al.* 2005). Zeno is about getting a number of people from an initial to a goal location using a number of airplanes. There is uncertainty about the duration of boarding and disembarking from an airplane. This uncertainty is modelled via two actions for each activity (e. g. `start-boarding` and `complete-boarding`), the latter succeeding with probability 1/k and failing with probability 1 - 1/k, thereby modelling a geometric duration distribution. Flying from a location to another consumes a certain amount of fuel which is independent of the actual duration of the trip: either 1 unit of fuel for normal flying, or 2 units for zooming. Again, the duration of flying and zooming have geometric distributions, with flying taking longer on average than zooming. Finally, there is an action for getting 1 unit of fuel, also with a geometric duration.

There are no dead ends, as it is always possible to fly provided that the maximum possible fuel level is at least 1 (and if it isn't the problem is trivially unsolvable). So a trajectory-preserving replanner like REPLAN2 will always reach the goal. REPLAN1 cannot solve the problem in most cases because the most likely outcomes lead to failure.

Zeno seems more interesting when viewed as an SSPP. Both versions of REPLAN2 do not necessarily find the optimal SSPP solution. The shortest trajectory to the goal never needs to make use of zooming, as it might require additional refuelling actions. Unless the planes have so much fuel that there is never any need to refuel even when zooming, the probabilities are such that the most likely trajectory will be identical to the shortest one, and so does not need to make use of zooming either. Yet the optimal solution to the SSPP is not always a concatenation of most likely or shortest trajectories. The difference occurs when we can upgrade from flying to zooming by refuelling one additional level.

While Zeno could turn out to be an interesting SSPP, the IPC-5 results clearly show that unless the size of the Zeno instances is reduced to be commensurable with the state of the art in SSPP solving, probabilistic planners will inevitably be outperformed by replanners.

Conclusion

Following our analysis, we can safely say that the pre-IPC-5 understanding of where replanning and probabilistic planning are respectively appropriate has not suddenly become obsolete. The competition results should not be surprising. A large proportion of the instances were either probabilistically uninteresting, too trivial to present any challenge for replanners, or insurmountably large for probabilistic planning to be suitable. It is rather inevitable that a replanner would 'win' in such a situation.[3]

Consequently, we believe that the current negative perception of probabilistic planning by the ICAPS community is inaccurate. If anything, there has been significant progress and

[3] Domains, instances, and set up, are not the only factors influencing the results of competitions. As it is well-known from the determinstic track, the language also has an influence on participation and results. For instance, despite efforts from the IPC-4 organisor in providing translations to ADDs, the choice of PPDDL seems to have discouraged many structured-MDP approaches that were florishing a few years ago to participate in the probabilistic track.

innovation since IPC-4.[4] Noticeably, some of the IPC-5 competitors are designed to be much more expressive than required by the competition problems and address difficult areas of research. For instance, FOLDP produces *generalised* policies for first-order Markov decision processes (Sanner & Boutilier 2007). FPG is originally designed for probabilistic *temporal* planning problems which may contain continuous variables and distributions, uncertain durations, and where makespan, resources, and failure probability need to be optimised (Aberdeen & Buffet 2007). So, the fact they could perform reasonably well in a very restricted setting is a rather remarkable accomplishment.

We believe that the most important lesson from the competition probabilistic track is that a synthesis of planning and replanning techniques could make a much larger number of probabilistic planning problems practicably solvable than is currently possible. In a panel talk at the ICAPS-06 Workshop on Planning under Uncertainty, David Smith pointed out that while the field has developed reasonably powerful approaches to replanning, conformant planning, and contingent planning, the most important thing it is missing is a theory of how to integrate them all. In such an integration, replanning should be the baseline. When uncertainty is not uniformly neglectable in a problem, we might want to build parts of plans that are robust (or conformant) to certain sources of uncertainty. Unfortunately, robustness also has a cost, because it may prevent taking advantage of certain opportunities. If such opportunities are important, then, and only then, contingent planning should be used to account for them. This integration of replanning with limited conformant and contingency planning should become one of the main priorities in field's research agenda. See (Buffet & Aberdeen 2007; Foss, Onder, & Smith 2007) for works in this direction.

Finally, we have demonstrated that while it is easy to design benchmarks that are optimally solvable with a replanner, it is just as easy to design benchmarks that require a significant amount of probabilistic reasoning to obtain a satisfactory solution. Therefore, (unless perhaps it featured real-world problems) a single probabilistic planning track is unlikely to be measuring anything but which camp the majority of problems selected belong to. It would be much more interesting to have two tracks. The first should encourage the field to develop planners that integrate replanning and probabilistic planning. It would use a mixed set of problems and would retain the current mode of evaluation. The second track would measure progress in solving problems that *require* a significant amount of probabilistic reasoning to obtain an even remotely satisfactory solution. It would require contestants to compute an explicit policy or contingency plan, and then compute its 'score' exactly. The current set of probabilistically interesting benchmark domains, coupled with a careful selection of problem instances might even suffice as a basis for both tracks.

[4]For what this observation is worth, mGPT (Bonet & Geffner 2005), which we consider as the best performing probabilistic planner in the IPC-4, is outperformed by FPG, the official IPC-5 winner, on all IPC-5 domains except Tireworld. The IPC-4 version of FF-replan, the official IPC-4 winner, is also outperformed by FPG globally, and on four of the five probabilistically interesting domains (Yoon, Fern, & Givan 2007).

Acknowledgements

A big thank you to the organisers of the probabilistic tracks of the IPC-4 and IPC-5, Blai Bonet, Bob Givan, Michael Littmann, and Håkan Younes. Special thanks to Sungwook Yoon for helping us with FF-replan and the competition domains, and Olivier Buffet and Doug Aberdeen for their help with FPG. This paper benefited from discussions with Doug Aberdeen, Blai Bonet, Ronen Brafman, Olivier Buffet, Maria Fox, Hector Geffner, Bob Givan, Patrik Haslum, Sarah Hickmott, Sven Koenig, Mausam, Scott Sanner, David Smith, Sungwook Yoon, and probably others whom we have forgotten. Thanks also to the anonymous reviewers for their comments and suggestions.

We thank National ICT Australia (NICTA) for its support, in particular via the DPOLP (Dynamic Planning, Optimisation & Learning) project. NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian National Research Council.

References

Aberdeen, D., and Buffet, O. 2007. Temporal probabilistic planning with policy gradient. In *17th International Conference on Automated Planning and Scheduling (ICAPS-07)*.

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. `http://www.ldc.usb.ve/~bonet/ipc5/docs/cf-ipc-prob.pdf`.

Buffet, O., and Aberdeen, D. 2007. FF+FPG: Guiding a policy gradient planner. In *17th International Conference on Automated Planning and Scheduling (ICAPS-07)*.

Foss, J.; Onder, N.; and Smith, D. 2007. Preventing unrecoverable failures through precautionary planning. In *ICAPS-07 Workshop on Moving Planning and Scheduling Systems into the Real World*.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Little, I., and Thiébaux, S. 2006. Concurrent probabilistic planning in the Graphplan framework. In *16th International Conference on Automated Planning and Scheduling (ICAPS-06)*, 263–272.

Sanner, S., and Boutilier, C. 2007. Approximate solution techniques for factored, first-order MDPs. In *17th International Conference on Automated Planning and Scheduling (ICAPS-07)*.

Slaney, J., and Thiébaux, S. 2001. Blocks world revisited. *Artificial Intelligence* 125(1-2):119–153.

Yoon, S.; Fern, A.; and Givan, B. 2007. FF-replan: a baseline for probabilistic planning. In *17th International Conference on Automated Planning and Scheduling (ICAPS-07)*.

Younes, H. L. S.; Littman, M.; Weissmann, D.; and Asmuth, J. 2005. The first probabilistic track of the International Planning Competition. *Journal of Artificial Intelligence Research* 24:851–887.