Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

Probabilistic Planning via Determinization in Hindsight

Sungwook Yoon
Department of CSE
Arizona State University
Tempe, AZ 85281
Sungwook.Yoon@asu.edu

Alan Fern
School of EECS
Oregon State University
Corvallis, OR 97331
afern@eecs.orst.edu

Robert Givan
Department of ECE
Purdue University
W. Lafayette, IN 47907
givan@purdue.edu

Subbarao Kambhampati
Department of CSE
Arizona State University
Tempe, AZ 85281
rao@asu.edu

Abstract

This paper investigates hindsight optimization as an approach for leveraging the significant advances in deterministic planning for action selection in probabilistic domains. Hindsight optimization is an online technique that evaluates the one-step-reachable states by sampling future outcomes to generate multiple non-stationary deterministic planning problems which can then be solved using search. Hindsight optimization has been successfully used in a number of online scheduling applications; however, it has not yet been considered in the substantially different context of goal-based probabilistic planning. We describe an implementation of hindsight optimization for probabilistic planning based on deterministic forward heuristic search and evaluate its performance on planning-competition benchmarks and other probabilistically interesting problems. The planner is able to outperform a number of probabilistic planners including FF-Replan on many problems. Finally, we investigate conditions under which hindsight optimization is guaranteed to be effective with respect to goal achievement, and also illustrate examples where the approach can go wrong.

Introduction

An interesting development in the planning community has been the seemingly paradoxical success of FF-Replan (Yoon, Fern, & Givan 2007) in the probabilistic planning track of the international planning competition. While the original intent of this track was to evaluate which of the "probabilistic planners" do well, the competition was dominated by FF-Replan, which determinizes the problem (e.g. assume that each action deterministically leads to only one of its possible outcomes) and then follows a simple replanning strategy. This unexpected result has led to significant scrutiny of the probabilistic planning track, including discussion of the domains used in the competition. Some (Little & Thiebaux 2007) have attempted to define a notion of "probabilistically interesting" domains, and claimed that FF-Replan's success can be attributed to the fact that the competition problems are not probabilistically interesting and/or are simple in some other sense. Indeed, the determinization strategies used by FF-Replan do fail in several domains that are deemed probabilistically interesting.

This paper takes a different perspective on this development. We re-interpret FF-Replan's basic strategy as a degenerate form of hindsight optimization, an "online anticipatory strategy" for control problems that has previously been applied successfully to problems such as online scheduling. Rather than solving a single determinized problem at each step, hindsight optimization randomly generates a set of non-stationary determinized problems (where the outcome selected for an action varies with time) and combines their solutions. Adapting this approach to FF-Replan essentially involves a form of "dynamic determinization".

The hindsight optimization approach thus provides a principled reduction from probabilistic planning to deterministic planning. We show empirically that this reduction is able to perform well even in the probabilistically interesting domains of (Little & Thiebaux 2007). Furthermore, we show conditions under which this reduction is guaranteed to be sound and efficient in goal-oriented domains. We also provide insight into the failure modes and implementation choices of this reduction.

Importantly reductions such as hindsight optimization provide a way for leveraging the large amount of effort and advancement in deterministic planning outside of the confines of deterministic domains. Similarly, the large amount of work on learning to plan, which has focused primarily on deterministic planning, becomes more relevant. In this respect, our work joins a number of other recent efforts on leveraging deterministic planners in non-deterministic and stochastic domains (Ng & Jordan 2000; Younes & Simmons 2004; Palacios & Geffner 2007; Foss, Onder, & Smith 2007; Bryce, Kambhampati, & Smith 2008).

The contributions of this paper are thus three fold: 1) to connect FF-Replan to the work on online anticipatory algorithms, 2) to provide a generalization of FF-Replan based on hindsight optimization that is effective in a broader class of probabilistic planning benchmarks, while retaining its ability to exploit deterministic planning techniques, 3) to provide conditions under which hindsight optimization constitutes a sound and efficient reduction in goal-oriented domains.

Problem Setup

A probabilistic planning problem is a Markov decision process (MDP) $M = \langle S, s_0, A, P, R \rangle$, with finite set of states

S, initial state $s_0 \in S$, finite set of actions A, Markovian transition function P giving the probability $P(s' \mid s, a)$ of reaching state s' after executing action a in state s, and reward function R mapping states to real-valued rewards. In this work, we assume that the transition function is represented using the probabilistic planning domain description language (PPDDL) (Younes 2003). This allows us to view each action a as being represented via a precondition, a finite set of outcomes $\mathcal{O}(a)$, each of which is a set of deterministic PDDL effects, and a discrete distribution D_a over those outcomes. An action a is available for execution when its precondition is satisfied and when executed leads to a next state generated by drawing an outcome from D_a and then applying the effects of that outcome to the current state.

Given an MDP, the planning objective is typically to select actions so as to optimize some expected measure of the future reward sequence, for example, total reward or cumulative discounted reward. In this paper, as in the first two probabilistic planning competitions, we will focus on goal-based planning problems where the primary objective is to reach a state achieving a goal condition. We note, however, that our approach also applies to non-goal-based problems. One goal-based formulation is to restrict consideration to problems where all rewards are zero except for goals states with reward of one, and where all actions in goal states lead to zero-reward terminal states (where all actions are self-transitions). In this case, maximizing expected undiscounted cumulative reward is equivalent to maximizing the probability of goal achievement. If instead we choose to maximize expected cumulative discounted reward then this would result in a trade-off between goal-achievement probability and expected solution length. Other straightforward formulations can take into account arbitrary action costs.

Online Action Selection. The first two international probabilistic planning competitions used an online approach for evaluating planners. The competition host would first send out a description of the planning problem and then provide an interface to the probabilistic environment that allowed a planner to send actions to be executed and then receive state updates based on the simulated action outcomes. A number of planners in the competition (Little & Thiebaux 2006; Buffet & Aberdeen 2007; Sanner & Boutilier 2006) followed the approach of first solving for a partial action selection policy offline after receiving the planning problem, and then execute that policy during the online evaluation. Rather, in this paper, we focus on purely online action selection policies, where all planning is done during online evaluation. In particular, at each state encountered during execution, we run a fast heuristic planning procedure that computes an action to be executed. FF-Replan (Yoon, Fern, & Givan 2007), which won the first planning competition, followed an online selection strategy where a deterministic approximation of the MDP was solved by the planner FF in order to select an action. In this paper, we seek to improve the deterministic approximation used by FF-Replan via the technique of hindsight optimization.

Hindsight Optimization

Hindsight optimization (HOP) is an online action selection heuristic for stochastic control that has been successfully applied in a number of application domains such as packet scheduling and online reservation systems (Chong, Givan, & Chang 2000; Wu, Chong, & Givan 2002). The main idea underlying HOP is to approximate the value of a state by sampling a set of non-stationary deterministic problems originating from that state, and then solving those problems "in hindsight" and combining their values. In cases, where the resulting deterministic problems are much easier to solve than the original probabilistic problem, we get large computational gains. Below we describe HOP more formally and then relate it to the strategy employed by FF-Replan.

A *T-horizon future* F is a mapping from $A \times S \times \{1, ..., T\}$ to the real interval $[0, 1]$, where A is the action set and S the state set. Given a probabilistic planning problem M and future F, we let $M[F]$ denote the *T-horizon determinization* of M with respect to F, which simply assigns a deterministic outcome to each action a in state s at time t as determined by $F(a, s, t)$. This can be done, for example, by partitioning the interval $[0, 1]$ according the outcome distribution D_a for action a and then selecting the outcome for a at time t that $F(a, s, t)$ indexes. A non-stationary policy $\pi = (\pi_1, \ldots, \pi_T)$ is a sequence of policies, each π_i providing a mapping from states to actions for use at time-step i. We denote the total reward achieved by π when executed for T steps in $M[F]$ starting at s by $R(s, F, \pi)$.

The optimal T-horizon value function $V^*(s, T)$ for state s is given by,

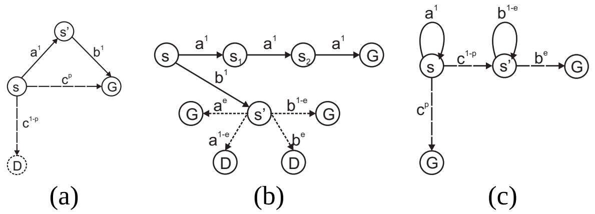
$$V^*(s, T) = \max_{\pi} E[R(s, F, \pi)],$$

where F is a random variable distributed uniformly over all T-horizon futures. The corresponding T-horizon Q-function is then given by $Q^*(s, a, T) = R(s) + E[V^*(s', T - 1)]$ where s' is distributed according to $P(\cdot \mid s, a)$. The policy of selecting actions that maximize $Q^*(s, a, T)$ is known as receding-horizon control, and can be made arbitrarily close to optimal for large enough T. Unfortunately, computing this Q-function is typically intractable and thus must be approximated in practice.

HOP approximates the Q-function by interchanging the order of the expectation and maximization in the computation of value functions. The HOP value function approximation is thus given by

$$V_{hs}(s, T) = E[\max_{\pi} R(s, F, \pi)]$$

with the hindsight Q-function given by $Q_{hs}(s, a, T) = R(s) + E[V_{hs}(s', T - 1)]$. The T-horizon hindsight policy is then defined as $\pi_{hs}(s, T) = \arg\max_a Q_{hs}(s, a, T)$. $V_{hs}(s, T)$ is an upper bound on the optimal value function $V^*(s, T)$ and can sometimes grossly overestimate the true value. Intuitively, this is because V_{hs} is able to "peek" at the resolution of future randomness, and is allowed to select a different policy for each such resolution, whereas V^* must find a single policy that works in expectation across all futures. The gain we get for interchanging the expectation and maximization is computational. In particular, it is possible



(a) (b) (c)

Figure 1: Example Problems

to accurately estimate $V_{hs}(s,T)$ by sampling a set of futures and then for each one solving for the policy with maximum reward and averaging the results. In the next section, we provide a bound on the number of samples required for accurate estimation of the hindsight policy. When this bound is reasonable and it is possible to solve the deterministic problems efficiently, the policy $\pi_{hs}(s,T)$ can be computed efficiently.

From the above we see there are two factors that play into the performance of hindsight optimization. One is the impact of the inaccuracy of V_{hs} compared to the true value function, and the other is computational feasibility. In previous applications of HOP to stochastic scheduling, the deterministic problems are offline scheduling problems for which efficient solvers were developed. Furthermore, in that context the hindsight policy provided good performance since Q_{hs} was able to rank actions quite accurately despite its potential inaccuracy. Note that Q_{hs} can sometimes provide good action selection in spite of overestimating Q^* if the overestimation preserves the ranking.

These past successes provide little insight into the practicality and performance of HOP on PPDDL planning problems. First, the scheduling-style domains are qualitatively very different from planning benchmarks. For example, they are typically not goal-based and the uncertainty generally only arises from exogenous events that are independent of the actions; in particular, streams of incoming scheduling requests. Second, in PPDDL planning, the deterministic problems arising from each sampled future are potentially quite difficult and it is unclear whether fast satisficing deterministic planners will be fast enough and/or find good enough plans. Furthermore, the deterministic problems are qualitatively different than typical deterministic planning benchmarks in that the effects of actions depend on the time step, which effectively increases the number of actions that must be considered. One of the main contributions of this paper is to consider these issues and to show that HOP can be effectively applied to AI planning benchmarks, but also to highlight weaknesses to be addressed in future work.

Relation to FF-Replan. It is possible to view FF-Replan as a degenerate approximation to HOP. In particular, consider the all-outcome variant of FF-Replan, FFR$_a$, which forms a single deterministic problem with a distinct action for every possible outcome of every action in the probabilistic planning problem. FFR$_a$ solves this problem at every step (or just when an "unexpected" outcome occurs) and selects the first action in the plan. One view of this is as an optimistic approximation to HOP where V_{hs} is approximated by replacing the expectation over futures with a maximization, that is, $V_{FF}(s,T) = \max_F \max_\pi R(s,F,\pi)$. Clearly $V_{FF}(s,T) \geq V_{hs}(s,T)$ for any s and can often be a gross overestimate. This view shows that FF-Replan is an optimistic approximation of HOP as it is allowed to select the best possible future regardless of its actual likelihood.

To see the possible impact of this optimism, consider the example problem in Figure 1(a). The starting state is s and the goal state is G. Action c has probability p of leading to the goal from S but leads to a dead end with probability $1-p$. The optimal policy for this problem avoids the possible dead end and instead takes a deterministic path through s' to G. FFR$_a$ would treat a, b, c^p, c^{1-p} each as separate deterministic actions and would thus select action c since c^p leads directly to the goal. In contrast, assuming enough sampling, π_{hs} would choose the correct path—in this example, $Q_{hs}(s,a,2)$ is one but $Q_{hs}(s,c,2)$ would be p, less than one, since a fraction $1-p$ of the sampled futures result in a dead end after taking action c from state s.

Goal Achievement Analysis

We saw that HOP provides tighter upper-bound estimates on state value than the overly optimistic FFR$_a$ by accounting for the probability of various futures. However, HOP is still optimistic compared to the true value function and it is important to understand the effect that this optimism can have on performance. In this section, we consider some of the failure modes of HOP with respect to goal achievement and give a class of planning problems for which HOP is guaranteed to be optimal (when focusing solely on success ratio).

Here we focus our analysis on the objective of maximizing goal achievement probability. In this case, $R(s,F,\pi)$ is always either 1 or 0 depending on whether π reached the goal in the deterministic $M[F]$ or not. There are two HOP implementation choices that turn out to be important with respect to this objective. The first is the choice of distribution over futures. The literature on HOP and other anticipatory algorithms has often been described in terms of a correlated future distribution (sometimes called "common random numbers") where $F(a,s,t) = F(a',s',t)$ for all $a, a' \in A$, $s, s' \in S$, and t. That is, actions at the same time step share the same random number. This sharing is most reasonable where the source of uncertainty is external to the system, as in packet arrivals to a packet scheduler. If, on the other hand, it is the case that $F(a,s,t)$ is always independent of each $F(a',s',t')$ (unless $(a,s,t) = (a',s',t')$), then we say that we have an *independent future distribution*. Second, the hindsight policy π_{hs} selects an action that maximizes Q_{hs}. We say that there is *random tie-breaking* if this choice is random over all actions that achieve the maximum.

We start by showing an example of where π_{hs} can perform poorly if correlated futures are used. Consider the problem in Figure 1(b) with single goal state G. Here s has the choice of taking a deterministic path to the goal or instead going to state s' and then selecting either action a or b, each of which chooses stochastically between the goal state and a dead end. Consider the case of correlated futures where the outcomes of a and b are ordered such that for any number in $[0,1]$ either the goal outcome of a or goal outcome of b is selected. In this case, $V_{hs}(s',T) = 1$ since in all futures the outcomes for a and b are selected together by a single random number that always results in a path to the goal. This allows π_{hs} to choose to go to s' even though this is suboptimal, because, the true value of s' is less than 1 since there is always a chance of reaching a dead end. If one uses independent futures then the optimal path will be selected, since some of the futures will show that both a and b lead to a dead end.

This suggests that there can be less over-estimation of value and correspondingly stronger guarantees about HOP using independent futures. A similar example can easily be constructed to show the necessity for state-dependent random numbers, making the future distribution F depend on the state s as well as the action a and time t.

It is also straightforward to find problems where random tie-breaking improves the performance of HOP. A simple example has a single initial state s that has a self-loop action and an action that leads directly to the goal. In this case, the hindsight Q-values for each action are equal to 1 and thus an unlucky choice of deterministic tie-breaking could select the self-loop in s repeatedly and never reach the goal. One might think that this problem could be overcome with a smart deterministic tie-breaking mechanism such as selecting the action that has the shortest expected distance to the goal in hindsight, which would correct the behavior in our simple example. However, it is possible to construct examples, where even this tie-breaking mechanism results in such looping behavior. Figure 1(c) gives one such example.

Given independent futures and random tie-breaking, we can show at least one problem class where π_{hs} is guaranteed to reach the goal.

Theorem 1. *Given a planning problem M, if there exists a policy π with probability 1 of reaching the goal in T or fewer steps, then π_{hs} with horizon T, independent futures, and random tie breaking has success probability 1 at the infinite horizon.*

Proof. (Sketch) We say that a state s is *T-solvable* if $V^*(s,T) = 1$, and T-unsolvable otherwise. The key property to proving the theorem is that for any state s and horizon T, $V^*(s,T) = 1$ iff $V_{hs}(s,T) = 1$. It follows that $Q^*(s,a,T) = 1$ iff $Q_{hs}(s,a,T) = 1$ for any s, a, and T. From this it follows that $\pi_{hs}(s)$ will never select an action that can lead to an T-unsolvable state. Furthermore, there is always a trajectory of at most T state-action-outcome triples (s,a,o) leading to the goal, where $Q_{hs}(s,a,T) = 1$ and o is a non-zero probability outcome of a in s leading to the state of the next triple. We can easily bound the probability that π_{hs} will follow this trajectory away from is zero, and since this repeats at every state, we will eventually execute such a trajectory and reach a goal.

It remains to prove the key property. The forward direction is trivial since we know that $1 \geq V_{hs}(s,T) \geq V^*(s,T)$. For the backward direction we prove that if $V^*(s,T) < 1$ then $V_{hs}(s,T) < 1$. To do this we prove below that for any s and T, if $V^*(s,T) < 1$ then there is a set of T-horizon futures $\mathcal{F}$ of measure greater than zero such that for each $F \in \mathcal{F}$, $\max_\pi R(s,F,\pi) = 0$. That is, there is a measurable set of futures such that no policy can reach the goal. Given this fact, the backward direction follows since with independent futures there is a non-zero probability of generating a future from the set $\mathcal{F}$, which implies $V_{hs}(s,T) < 1$.

It is easy to show that for every action a' and t-unsolvable state s', there must be some outcome for taking a' in s' that leads to a $(t-1)$-unsolvable state (otherwise a' would "solve" s' within t steps). Call such outcomes "t-failure outcomes." One can show that there is a measurable set of futures $\mathcal{F}$ such that each $F \in \mathcal{F}$ selects, for each time t in $\{1,\ldots,T\}$, a t-failure outcome at every t-unsolvable state. For each such future F, in the deterministic problem $M[F]$, paths preserve unsolvability with decreasing horizon: every state reached by a k-step or shorter path from a T-unsolvable state is $(T-k)$-unsolvable and thus is not a goal. It follows that $R(s,F,\pi) = 0$ for all π, as desired. □

An example problem that satisfies the assumptions of this theorem is the triangular tireworld (Little & Thiebaux 2007), which is a so-called probabilistically interesting domain designed to defeat replanners. There are many other problems not covered by this proposition, including those where there exist policies with success probability 1, but for which the length of solution trajectories is unbounded. Figure 1(c) depicts such a problem. Here π_{hs} will choose no-op action a from the initial state s and never reach the goal, although there is a policy with success probability 1. Here, no-op is selected because $V_{hs}(s,T)$ is greater than $V_{hs}(s',T)$ since there are more opportunities to reach the goal from s.

Sampling Bounds. We approximate $Q_{hs}(s,a,T)$ by sampling w futures $\{F_1,\ldots,F_w\}$ and then computing the approximation $\hat{Q}(s,a,T) = 1/w \sum_i R(s) + \max_\pi R(s', F_i, \pi)$, where $s' \sim p(s'|s,a)$. The empirical hindsight policy $\hat{\pi}(s,T)$ then selects actions randomly among those that maximize $\hat{Q}(s,a,T)$. We are interested in bounding the sampling width w required to guarantee that with high probability $\hat{\pi}(s,T)$ only returns actions that maximize Q_{hs}. If this is the case then $\hat{\pi}(s,T)$ provides an accurate simulation of $\pi_{hs}(s,T)$. It does not appear possible to provide a problem independent bound on w. Rather, we provide a bound in terms of the hindsight Q-advantage Δ_T of a planning problem, which is equal to the minimum over all states of the difference between Q_{hs} for the best action and Q_{hs} of the second best action. If we let $Q_1(s,T)$ be the maximum hindsight Q-value at s and $Q_2(s,T)$ be the second best Q-value then $\Delta_T = \min_s Q_1(s,T) - Q_2(s,T)$. The Q-advantage measures the minimum gap between any hindsight optimal action and any other non-optimal action. With this definition we get the following bound.

Theorem 2. *For any planning problem with hindsight Q-advantage at least Δ_T, if $w > 4\Delta_T^{-2} \ln \frac{|A|H}{\delta}$ then with probability at least $1-\delta$, $\hat{\pi}(\cdot, T)$ will select actions that maximize $Q_{hs}(s,a,T)$ for at least H consecutive decisions.*

Proof. (Sketch) Using a Chernoff bound we get that at each step with probability at least $1-\delta'$, $|Q_{hs}(s,a,T) - \hat{Q}(s,a,T)| \leq \sqrt{\frac{-\ln \delta'}{w}}$. By setting $\delta' = \frac{\delta}{H}$ we guarantee that the bound will hold for H independent steps with probability $1-\delta$. Using this value and the bound on w from the proposition we get that $|Q_{hs}(s,a,T) - \hat{Q}(s,a,T)| < \frac{\Delta}{2}$ with probability at least $1 - \delta$ for H steps, which is equivalent to saying that no hindsight optimal action will be ranked worse than a non-optimal action throughout the H steps and thus $\hat{\pi}$ will not return a non-optimal action. □

This result shows that the HOP reduction from probabilistic to deterministic planning is efficient for constant Δ_T. We note, however, that it is possible to construct problems where Δ_T becomes exponentially small in T, which means that the required sampling width can grow exponentially in T. The dependence on Q-advantage appears unavoidable, since with a fixed sampling width it is always possible to create a problem with small enough Q-advantage that the probability of returning hindsight suboptimal actions is arbitrarily high.

Implementation

While our analysis was able to establish guarantees for only one class of problems, we were sanguine that in practice the relative performance of the technique may be significantly better. To verify this, we implemented HOP for PPDDL problems in a system called FF-Hindsight. FF-Hindsight is based on three approximations to the true hindsight policy π_{hs}. First, we utilize sampling to estimate the $Q_{hs}(s,a,T)$ values rather than computing the true expectation. Second, rather than sampling futures from a uniform independent future distribution, we utilize the idea of common random numbers to sample futures from a different distribution where certain components of the futures are tied together. In particular, our current implementation samples a future by drawing a sequence of T random numbers $(r_1, \ldots, r_T)$ and then letting $F(s,a,t) = r_t$ for all s, a, and t. The use of common random numbers is a common way to reduce variance at the potential cost of estimation bias. Third, rather than compute the optimum policy for each sampled future we use a heuristic deterministic planner.

More precisely, we compute an approximation $\hat{\pi}(s,T) = \arg\max_a \hat{Q}(s,a,T)$ to $\pi_{hs}(s,T)$ where $\hat{Q}(s,a,T)$ is an approximation to the hindsight Q-function. $\hat{Q}(s,a,T)$ is computed by first sampling w number of T-horizon futures $\{F_1, \ldots, F_w\}$ from the non-uniform distribution described above and then for each action a computing $\hat{Q}(s,a,T) = 1/w \sum_i \hat{R}(s', F_i)$, where s' is the result of taking action a in state s in future F_i and $\hat{R}(s,F)$ is an approximation to $\max_\pi R(s,F,\pi)$. In our implementation $\hat{R}(s,F)$ is implemented by running a variant of FF (Hoffmann & Nebel 2001) on the deterministic problem and then returning the negative plan length as the reward or $-T$ if no plan is found.

It remains to specify the details of our deterministic planner. One approach would be to create a non-stationary PDDL planning problem, by introducing a distinct deterministic version of each action for each time point with outcomes as specified in the future. This problem could then be passed to any PDDL planner. This approach would be particularly well suited to a SAT-style planner that can naturally encode non-stationary actions with little additional overhead. However, in this work, we bypass the actual construction of the non-stationary PDDL problems and instead extend the best-first search component of FF to directly read in the probabilistic action definitions along with a future. In particular, it is straightforward to alter the action expansion function so that actions at search depth t follow the deterministic effects specified in the provided future at time t.

	Results from IPPC-04		
Domains	FFR$_s$	FFR$_a$	FF-Hindsight
bw-c-pc-8	30 (1)	30 (0)	30 (5)
bw-c-pc-nr-8	30 (1)	30 (0)	30 (5)
bw-nc-pc-11	30 (1)	30 (0)	8 (30)
bw-nc-pc-15	0 (-)	30 (1)	0 (-)
bw-nc-pc-18	0 (-)	30 (28)	0 (-)
bw-nc-pc-21	30 (19)	30 (3)	0 (-)
bw-nc-pc-5	30 (0)	30 (0)	30 (2)
bw-nc-pc-8	30 (0)	30 (0)	30 (3)
bw-nc-pc-nr-8	30 (0)	30 (0)	30 (3)
bx-c10-b10-pc-n	30 (3)	30 (0)	10 (30)
bx-c10-b10-pc	30 (2)	30 (0)	10 (30)
bx-c15-b10-pc	30 (3)	30 (0)	20 (30)
bx-c5-b10-pc	30 (1)	30 (0)	30 (5)
bx-c5-b10-pc-nr	30 (1)	30 (0)	30 (5)
exploding-block	3 (0)	5 (0)	28 (7)
file-prob-pre	14 (30)	29 (29)	14 (30)
g-tire-problem-	7 (0)	7 (0)	18 (2)
r-tire-problem-	30 (0)	30 (0)	30 (2)
toh-prob-pre	0 (-)	11 (0)	17 (11)
ztravel-1-2	0 (-)	30 (0)	0 (-)

Figure 2: Results on IPPC-04 for FF-Replan and Hindsight Approaches. FFR$_s$ is replanning with single outcome determinization and FFR$_a$ is with all outcome determinization. The notation $n(m)$ means n trials were solved successfully in m minutes.

A key question with this implementation is the choice of heuristic function. One approach would be to upgrade the relaxed plan computation to account for the known future, which raises a number of complications due the mismatch between sequential plans, which the futures pertain to, and the layered plangraph used to construct relaxed plans. In this work, we rather take a simpler approach and use a relaxed-plan heuristic based on the all-outcomes encoding of FF-Replan. While this all-outcomes encoding has the potential to produce highly optimistic heuristic values, we found that it performed on par with more sophisticated attempts we made on heuristics that take into account the provided future.

An important implementation issue that arose was that we also need to extend the native hashing mechanism of FF to take both the time and state into account. That is, rather than just hashing state, we hashed state and time altogether, to prevent returning failure just because a particular outcome of an action has not been realized at certain time point. This compromised the scalability of the planner to some degree as we will see in some of our results.

Experimental Results

We tested FF-Hindsight on the suite of IPPC-04 competition problems with sample width $w = 30$ and horizon $T = 100$.

The choice of sample width was based on our empirical testing over the benchmarks. In particular, we experimented with values of w between 10 to 50. Although we obtained good performance for relatively small values of w in most domains, for IPPC-04 problems, width less than 30 tended to produce variable results. In contrast, increasing the width beyond 30 did not result in much improvements. We thus opted to set $w = 30$ for all our experiments. The setting of $T = 100$ was done in light of the knowledge that most of the problems can be solved within 100 steps (the only exception was the Zeno-travel problem, the bottom row problem in Figure 2).

Figure 2 summarizes the results with each table entry showing how many of the 30 trials were solved and the number in parentheses giving the total time in minutes used during the evaluation period. We also listed the performance of two versions of the replanning techniques, FFR$_s$ and FFR$_a$ (replanners based on single outcome and all outcome determinization respectively) (Yoon, Fern, & Givan 2007).

Among the IPPC problems, toh-prob-pre, exploding-block and g-tire-problem were known to be difficult for the replanning approaches as these contain dead-end states. The planner needs some form of probabilistic reasoning to maximize the number of success trajectories to the goal. In all of these problems, FF-Hindsight outperformed the replanners. toh-prob-pre is similar to the Tower-of-Hanoi problem but there are added actions which can move two disks at the same time but they can lead to deadend states with higher probability than single-disk actions. Replanners naively select two-disk actions and fail, while FF-Hindsight avoids such pitfalls. The g-tire-problem has two routes, one without spare tires and the other with spare tires. Replanners do not distinguish the routes and select the former, resulting in worse performance than FF-Hindsight.

Exploding Blocksworld was the hardest problem in IPPC-04 with only FF-Replan having a positive success rate. These problems have a policy with success probability 1, which chooses to "detonate" blocks before using them. Unlike the replanners, FF-Hindsight was able to determine the importance of detonation and achieved a high success rate.

In contrast, problems from Blocksworld (problems starting with bw), Boxworld (problems starting with bx), file-prob-pre, r-tire-problem, and ztravel-1-2 correspond to domains without any dead end states. For these problems, the replanners perform quite well, and solved most of the problems. FF-Hindsight does not do as well in these domains given the fixed time limit as it reasons with significantly more futures (30 instead of 1). Note that each planner is allowed only 30 minutes for 30 trials. In our testing, we observed that for the larger sized Blocksworld and Boxworld problems, FF-Hindsight made positive progress towards solving the first trial before the 30 minutes timeout expires. Thus, for FF-Hindsight to solve these problems we need at least 15 hours for one problem. As expected, we also found that our modified FF for the non-stationary actions does not scale as well as the FF for normal stationary actions (the latter is used by FF-Replan).

For Zenotravel the failure of FF-Hindsight is primarily due to small probability outcomes that are required for success, which would require significantly larger values of w and T. We conducted an experiment where we solved Zenotravel using FF-Hindsight with importance sampling where futures were drawn from a proposal distribution with equal probability for each outcome. This variant solved 26 out of 30 problems, which points to an interesting future research avenue of automatically selecting good proposal distributions.

Probabilistically Interesting Domains We have also tested FF-Hindsight on the "probabilistically interesting" domains from (Little & Thiebaux 2007). These are "climber", "river", "bus-fare", and "triangle-tireworld" domains. We compare FF-Hindsight with FPG (winner of IPPC-06) (Buffet & Aberdeen 2007), Paragraph (Little & Thiebaux 2006) and FFR$_a$ (Yoon, Fern, & Givan 2007).

Figure 3 summarizes the experiments with all entries other than FF-Hindsight taken from (Little & Thiebaux 2007) giving the percentage over 30 trials that a planner solved for a particular problem. The problems climber, river, and bus-fare are very small problems but require some form of probabilistic reasoning and all planners except for FFR$_a$ perform well. For the triangular-tireworld, Paragraph and FPG are unable to scale to the larger problems, while FF-Hindsight achieves 100% success rate throughout.

In summary, the experimental results demonstrate that FF-Hindsight is able to perform well over a broader class of stochastic planning benchmarks. It is able to outperform replanners in problems where probabilistic reasoning is critical. At the same time it can be computationally more tractable compared to full offline probabilistic planning methods. However, on problems where replanners work well, FF-Hindsight will not always perform as well given a fixed evaluation time.

Related Work

Mercier & Van HenTenRyck (2007) have also studied theoretical properties of HOP. While certain details of their setting and ours differ, their results carry over in a straightforward way. Their main result shows that the performance of the hindsight policy is related to a quantity called the anticipatory gap which measures the difference between $V_{hs}(s, T)$ for a state and $\max_a Q_{hs}(s, a, T)$, which can be thought of as the value of one time step of foresight at state s. The performance bound is stated in terms of the global anticipatory gap, which is a pessimistic estimate of the worst case sum of anticipatory gaps that could possibly be encountered. There is no clear relationship between their result and our Theorem 1. For example, their result is not strong enough to prove the optimality of HOP in the triangular tire world. It is also easy to construct examples where their result applies but ours does not. We consider understanding the relationships between these results and possible extensions as an interesting direction for future work.

It is interesting to note the connection between HOP and the PEGASUS approach (Ng & Jordan 2000) for solving large stochastic planning problems. PEGASUS approximates V^* by replacing the expectation with an average over sampled futures, but does not interchange the order of expectation and maximization as does HOP. The result is a single

Planners	climber	river	bus-fare	tire1	tire2	tire3	tire4	tire5	tire6
FFR$_a$	60%	65%	1%	50%	0%	0%	0%	0%	0%
Paragraph	100%	65%	100%	100%	100%	100%	3%	1%	0%
FPG	100%	65%	22%	100%	92%	60%	35%	19%	13%
FF-Hindsight	100%	65 %	100%	100%	100%	100%	100%	100%	100%

Figure 3: Percentage success (over 30 trials) on problems from Little & Thiebaux's probabilistically interesting domains

deterministic problem that requires finding a single policy that is optimal with respect to the set of sampled futures. This deterministic planning problem is considerably harder than the individual problems that are required to be solved by HOP, where it is only necessary to optimize a policy for a single future. Furthermore deterministic PEGASUS problems are not of the form that can be accepted by standard deterministic PDDL planners. However, unlike HOP it is possible to provide uniform convergence bounds on the number of sampled futures that are sufficient to ensure that the solution to the deterministic problem is approximately optimal for the original stochastic problem.

One computational bottleneck in scaling FF-Hindsight is efficiently handling plans for multiple sampled futures. Because FF-Hindsight reasons with each future separately, it fails to exploit the significant common structure among the futures. In this context, the work on McLug by Bryce *et. al.* (Bryce, Kambhampati, & Smith 2008) may be relevant as it describes a way of deriving informed heuristics by simultaneously reasoning over a set of sampled futures. While FF-Hindsight computes an executable deterministic plan for each sampled future, McLug considers all the futures simultaneously and can thus lead to significant efficiency. On the other hand, although McLug's analysis considers positive interactions between the plans for different futures, it ignores negative interactions among them. It would be interesting to evaluate the tradeoffs between these approaches.

Conclusion

The success of FF-Replan in the probabilistic planning track of the IPC has lead to some controversy about the competition, as well as creation of alternative benchmarks intended to defeat FF-Replan and showcase the superiority of offline probabilistic methods for stochastic planning. In this paper, we suggested a view of FF-Replan as a degenerate form of hindsight optimization, and have shown that adaptation of standard hindsight optimization can significantly improve performance, while still retaining the key benefit of FF-Replan–*the ability to exploit the advances in deterministic planning*. Indeed, empirical studies with our current implementation, FF-Hindsight, show that it provides state of the art performance in a broader class of probabilistic planning benchmarks, including those designed to defeat FF-Replan.

In future we hope to focus on (i) efficiently generating plans for multiple sampled futures (including exploiting the common structure among the futures, as well as making the relaxed plan heuristics sensitive to the non-stationary determinization of actions) (ii) investigating the effect of using independent futures (as against futures correlated by common random numbers) on the accuracy of action selection and (iii) adapting hindsight approach to problems with more general cost/reward structures (as against simple goals of achievement), where we expect it to be even more competitive.

Acknowledgements

Yoon and Kambhampati's work is supported in part by the DARPA Integrated Learning Program (through a sub-contract from Lockheed Martin), and by ONR grants N000140610058 and N00014-07-1-1049 (MURI sub-contract from Indiana University). Fern's work is supported in part by NSF grant IIS-0546867 and DARPA contract FA8750-05-2-0249.

References

Bryce, D.; Kambhampati, S.; and Smith, D. E. 2008. Sequential monte carlo in reachability heuristics for probabilistic planning. *Artif. Intell.* 172(6-7):685–715.

Buffet, O., and Aberdeen, D. 2007. FF+FPG: Guiding a policy-gradient planner. In *Proceedings of International Conference on Automated Planning and Scheduling*.

Chong, E.; Givan, R.; and Chang, H. 2000. A framework for simulation-based network control via hindsight optimization. In *IEEE Conference on Decision and Control*.

Foss, J.; Onder, N.; and Smith, D. 2007. Preventing unrecoverable failures through precautionary.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:263–302.

Little, I., and Thiebaux, S. 2006. Concurrent probabilistic planning in the graphplan framework. In *Proceedings of International Conference on Automated Planning and Scheduling*.

Little, I., and Thiebaux, S. 2007. Probabilistic planning vs replanning. In *ICAPS Workshop on Planning Competitions: Past, Present, and Future*.

Mercier, L., and Van HenTenRyck, P. 2007. Performance analysis of online anticipatory algorithms for large multistage stochastic programs. In *International Joint Conference on Artificial Intelligence*.

Ng, A. Y., and Jordan, M. 2000. PEGASUS:A policy search method for large MDPs and POMDPs. In *Proceedings of International Conference on Uncertainty in Artificial Intelligence*, 406–415.

Palacios, H., and Geffner, H. 2007. From conformant into classical planning: Efficient translations that may be complete too. In *Proceedings of ICAPS-07*.

Sanner, S., and Boutilier, C. 2006. First order approximate linear programming. In *International Probabilistic Planning Competition Booklet of ICAPS*.

Wu, G.; Chong, E.; and Givan, R. 2002. Burst-level congestion control using hindsight optimization. *IEEE Transactions on Automatic Control*.

Yoon, S.; Fern, A.; and Givan, R. 2007. FF-Replan: A baseline for probabilistic planning. In *ICAPS*.

Younes, H., and Simmons, R. 2004. Policy generation for continuous-time stochastic domains with concurrency.

Younes, H. 2003. Extending pddl to model stochastic decision processes. In *Proceedings of the ICAPS-03 Workshop on PDDL*.