Domain-Independent Relaxation Heuristics for Probabilistic Planning with Dead-ends

Florent Teichteil-Königsbuch and **Vincent Vidal** and **Guillaume Infantes**
name.surname@onera.fr
ONERA — The French Aerospace Lab
F-31055, Toulouse, France

Abstract

Recent domain-determinization techniques have been very successful in many probabilistic planning problems. We claim that traditional heuristic MDP algorithms have been unsuccessful due mostly to the lack of efficient heuristics in structured domains. Previous attempts like mGPT used classical planning heuristics to an all-outcome determinization of MDPs without discount factor ; yet, discounted optimization is required to solve problems with potential dead-ends. We propose a general extension of classical planning heuristics to goal-oriented discounted MDPs, in order to overcome this flaw. We apply our theoretical analysis to the well-known classical planning heuristics h_{max} and h_{add}, and prove that the extended h_{max} is admissible. We plugged our extended heuristics to popular graph-based (Improved-LAO*, LRTDP, LDFS) and ADD-based (sLAO*, sRTDP) MDP algorithms: experimental evaluations highlight competitive results compared with the winners of past competitions (FF-REPLAN, FPG, RFF), and show that our discounted heuristics solve more problems than non-discounted ones, with better criteria values. As for classical planning, the extended h_{add} outperforms the extended h_{max} on most problems.

Introduction

Significant progress in solving large goal-oriented probabilistic planning problems has been achieved recently, partly due to tough challenges around the probabilistic part of International Planning Competitions (Younes et al. 2005). Looking back at successful planners, most of them rely on a deterministic planner to solve the probabilistic planning problem: FPG-FF (Buffet and Aberdeen 2007), FF-REPLAN (Yoon, Fern, and Givan 2007), RFF (Teichteil-Königsbuch, Kuter, and Infantes 2010), FF-H (Yoon et al. 2010), GOTH (Kolobov, Mausam, and Weld 2010). These planners "determinize" the original domain, generally by replacing probabilistic effects by the most probable one for each action ("most probable outcome" determinization), or by considering as many deterministic actions as probabilistic effects ("all-outcomes" determinization). Other successful but non determinization-based approaches are FPG (Buffet and Aberdeen 2009) and FODD-PLANNER (Joshi, Kersting, and Khardon 2010). It is noteworthy to mention that FPG was later improved to FPG-FF using a deterministic planner to guide simulated trajectories to the goal.

Yet, to our knowledge, there does not really exist any domain-independent, optimal and fully probabilistic algorithm whose performances are competitive with determinization-based approaches. In particular, traditional Markov Decision Process (MDP) heuristic algorithms like (s)LAO* (Hansen and Zilberstein 2001; Feng and Hansen 2002), (s,L)RTDP (Bonet and Geffner 2003; Feng, Hansen, and Zilberstein 2003), or LDFS (Bonet and Geffner 2006), have not enjoyed competitive results on the competition domains. However, one can wonder whether the efficiency of determinization-based planners is due to solving many simpler deterministic problems, or rather to very efficient heuristics implemented in the underlying deterministic planner.

The mGPT planner by (Bonet and Geffner 2005) partially answered this question; by applying state-of-the-art classical planning heuristics to an all-outcome determinization, it achieved good performances on domains modeled as Stochastic Shortest Path Problems (SSPs). As formalized later, such problems assume that there exists an optimal policy that reaches a goal state with probability 1. If not, it means that any optimal policy may reach some states with a positive probability, from where no goal states are then reachable (called *dead-ends*). In the 2004 probabilistic competition, mGPT could not solve domains with dead-ends like exploding blocksworld.

In this paper, we propose to handle dead-ends by means of a *discounted* criterion, thus solving *discounted SSPs*. Our work is different from the inverse transformation done in (Bertsekas and Tsitsiklis 1996) where it is shown that any γ-discounted MDP *without termination state* can be stated as an equivalent SSP (without discount factor): in their work, the goal state of the SSP only models the anytime termination of the process with probability $1 - \gamma$, and is defined as an absorbing state that is reachable from any state. The main drawbacks of this approach are: (1) the goal states of the original problem cannot be related to any state of the equivalent SSP, so that an algorithm for the equivalent SSP cannot be guided towards the goal states of the problem; (2) dead-ends of the original problem lead themselves to the goal state of the equivalent SSP, thus trying to reach the goal state leads to a policy that is attracted by dead-ends. In order to avoid

these flaws, in our work, we rather reason about the true goal states of the original problem, which is seen as a SSP whose costs are discounted.

Therefore, we propose (1) a generic extension of any classical planning heuristic to classical SSPs *without dead-end* and (2) a further extension to discounted SSPs (able to deal with dead-ends) that directly target the original problem's goal states. We apply these approaches to the well-known h_{max} and h_{add} heuristics by (Bonet and Geffner 2001), but could have extended other classical planning heuristics like h_{FF} by (Hoffmann 2001) in a similar way. We use our heuristics in state-of-the-art MDP heuristic search algorithms (`Improved-LAO*`, `LRTDP`, `LDFS`, `sLAO*`, `sRTDP`) with discounted settings, which always converge in presence or not of reachable dead-ends. We experimentally show that these algorithms with our discounted heuristics provide better goal reaching probability and average length to the goal than the winners of previous competitions or the same algorithms without discount factor.

Goal-oriented Markov Decision Process

The following definition is slightly adapted from (Wu, Kalyanam, and Givan 2008).

Definition 1. A goal-oriented MDP is a tuple $\langle \mathcal{S}, \mathcal{I}, \mathcal{G}, \mathcal{A}, app, T, C \rangle$ with:

- $\mathcal{S}$ a set of states;
- $\mathcal{I} \subseteq \mathcal{S}$ a set of possible initial states;
- $\mathcal{G} \subseteq \mathcal{S}$ a set of goal states;
- $\mathcal{A}$ a set of actions;
- $app : \mathcal{S} \to 2^{\mathcal{A}}$ an applicability function: $app(s)$ is the set of actions applicable in s;
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0; 1]$ a transition function such that $T(s, a, s') = Pr(s' \mid a, s)$ and $T(g, a, s') = 0$ for all $g \in \mathcal{G}$ and $s' \notin \mathcal{G}$ (goal states are absorbing);
- $C : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ a cost function such that, for all $a \in \mathcal{A}$, $C(s, a) = 0$ if $s \in \mathcal{G}$, and $C(s, a) > 0$ if $s \notin \mathcal{G}$.

A solution of a goal-oriented MDP is a partial policy $\pi_{\mathcal{X}} : \mathcal{X} \subseteq \mathcal{S} \to \mathcal{A}$ mapping a subset of states $\mathcal{X}$ to actions minimizing some criterion based on the costs. $\mathcal{X}$ contains all initial states, at least one goal state. Moreover, $\pi_{\mathcal{X}}$ is closed: states reachable by applying $\pi_{\mathcal{X}}$ on any state in $\mathcal{X}$ are in $\mathcal{X}$.

STRIPS representation of MDPs

ADL representation of MDPs as formalized in the PPDDL language by (Younes et al. 2005), has facilitated benchmark sharing and planners comparison. Moreover, this representation can be transformed into a simpler STRIPS form (Gazen and Knoblock 1997), that enabled to derive efficient domain-independent heuristics from the model, which has been mainly exploited by determinization-based approaches through the underlying deterministic planner. Yet, as partially highlighted by (Bonet and Geffner 2005), we claim that efficient domain-independent heuristics can be directly derived from the original probabilistic domain without determinization, and plugged to MDP heuristic algorithms.

In probabilistic STRIPS, states can be seen as collections of atoms from the set Ω of all atoms. Actions are probabilistic operators of the form $o = \langle prec, cost, [p_1 : (add_1, del_1), \cdots, p_m : (add_m, del_m)] \rangle$ where $\sum_{i=1}^{m} p_i = 1$ (more details in (Younes et al. 2005; Bonet and Geffner 2005)) and:

- $prec$ is a collection of atoms such that the action is applicable in a state s iff $prec \subseteq s$;
- $cost$ is the cost of the action; for each $i \in [1; m]$, p_i is the probability of the i^{th} effect which is represented by the set add_i of atoms becoming true and the set del_i of atoms becoming false; atoms that are neither in add_i nor in del_i keep their values.

Note that the remaining of this paper only assumes a probabilistic STRIPS representation of goal-oriented MDPs, so our contribution is valid for any formalism that boils down to such representation.

We illustrate this concept with the well-known `blocksworld` problem, which consists in building an ordered stack of n blocks $(b_1, \cdots, b_n)$ by a robot with a single hand, where each block can initially belong to different stacks. There are $n(n + 3) + 1$ atoms: $(emptyhand)$; $(holding\ b)$, $(ontable\ b)$ and $(clear\ b)$ for each block b; $(on\ b_1\ b_2)$ for any two blocks b_1 and b_2. As an example of action, $(pickup\ b_1\ b_2)$ is defined by:

$$\begin{aligned}\langle \{&(emptyhand), (clear\ b_1), (on\ b_1\ b_2)\},\\ &[0.75 : (\{(holding\ b_1), (clear\ b_2)\},\\ &\quad \{(emptyhand), (on\ b_1\ b_2)\})\\ &\quad 0.25 : (\{(clear\ b_2), (ontable\ b_1)\},\\ &\qquad \{(on\ b_1\ b_2)\})]\rangle\end{aligned}$$

Dead-end states

The existence of a solution depends on structural properties of the goal-oriented MDP, precisely on the presence of reachable dead-ends, as defined below.

Definition 2. A *dead-end* is a state from which the probability to reach the goal with any policy is equal to zero.

A non-goal absorbing state is a dead-end state. Now, consider the following criterion, known as the *total criterion*:

$$\forall s \in \mathcal{S}, \pi^*(s) = \text{argmin}_{\pi \in \mathcal{A}^{\mathcal{S}}} E\left[\sum_{t=0}^{+\infty} C_t \mid s_0 = s, \pi\right] \quad (1)$$

where C_t is the cost received at time t starting from s and executing policy π. If π can reach some dead-end state s_d with a positive probability, as no goal state is reachable from s_d by definition, and because costs received in any non-goal state are strictly positive, the sum of future costs is $+\infty$. Thus, *eq. 1 has a solution iff there exists at least one policy that, with probability 1, does not reach any dead-end state*.

An example of goal-oriented MDP with dead-ends is the `exploding blocksworld` domain, which extends the `blocksworld` domain to blocks that can detonate and then destroy other blocks or the table itself. The domain contains $2n + 1$ additional atoms: $(nodetonated\ b)$ and $(nodestroyed\ b)$ for each block b; $(nodestroyed - table)$. To build a goal stack, two kinds of actions are required: putting a block on the table, which can detonate the block and destroy the table with probability 3/5; putting a block

on another block, which can detonate the handled block and destroy the target block with probability 1/10. Destroying the table or a block belonging to a goal stack prevents from reaching the goal, and thus all states reachable from these situations are dead-ends. As all policies need to apply such actions to reach the goal, these dead-ends are reachable with a positive probability by executing any policy from the initial state.

Whether the total criterion has a solution or not is unrelated to the general problem of finding a policy that reaches a goal state with a positive probability (possibly not 1), whatever the minimization criterion considered. It only means that a planner based on the total criterion, as `mGPT` is, will not find a working solution for any domain with reachable dead-ends like `exploding blocksworld`. But other planners that are not based on this criterion, or that do not optimize any criterion like `FF-REPLAN`, will possibly find a solution for this domain, that reaches a goal state with a probability lower than 1. And this is the expected result.

Unfortunately, deciding if a given domain contains dead-ends reachable by the optimal policy boils down to optimizing the total criterion itself. Thus, most algorithms based on this criterion, like `mGPT`, simply cross the fingers: they try to solve the problem and hope for the best. Yet, the total criterion is very popular in heuristic MDP planning because it allows to design efficient domain-independent admissible heuristics, as explained in the next section.

Solving *undiscounted* goal-oriented MDPs

This class of MDPs, known as Stochastic Shortest Path Problems (SSPs) extended to positive costs, has been extensively studied[1]. If there exists an optimal policy reaching a goal state with probability 1, the total criterion of eq. 1 is well-defined, and heuristic algorithms optimize only relevant states when starting from known initial states. A forward-chaining procedure iteratively expands the searched state space until the value of explored states has converged. The convergence condition actually depends on the way states are explored, i.e. on the heuristic and on the algorithm exploiting it. Yet, all these algorithms update the value of any explored state s by using the same Bellman backup equation (see (Bonet and Geffner 2003; 2006; Hansen and Zilberstein 2001)):

$$V(s) \leftarrow \min_{a \in app(s)} \left\{ C(s,a) + \sum_{s' \in \mathcal{S}} T(s,a,s') \widetilde{V}(s') \right\} \quad (2)$$

where $\widetilde{V}(s') = V(s')$ if s' has already been explored, and otherwise $\widetilde{V}(s') = H(s')$, with $H : \mathcal{S} \rightarrow \mathbb{R}_+$. H is a heuristic function that initializes the value of unexplored states. It is proved (e.g. (Hansen and Zilberstein 2001)), that heuristic algorithms converge to an optimal solution iff H is admissible, i.e.: $\forall s \in \mathcal{S}, H(s) \leqslant V^{\pi^*}(s)$. The closer H is to V^{π^*}, the less states are explored, and the faster the algorithm converges. To be efficient in a domain-independent context, heuristic functions must be much easier to compute than the value function itself, and as close as possible to the optimal value function. To achieve these antagonist objectives, a good compromise consists in computing heuristic values on a relaxed planning domain.

[1] SSPs are often defined as goal-oriented MDPs with unit costs when not in a goal state, rather than any positive value like in our more general definition.

New STRIPS relaxation heuristics for SSPs

We here propose a generic extension of classical planning heuristics to SSPs, by reasoning about the "all-outcome determinization" of the MDP, generalizing the work by (Bonet and Geffner 2005). We show how to design admissible heuristics for SSPs from the deterministic case, and apply our theoretical extension to the h_{max} and h_{add} heuristics by (Bonet and Geffner 2001). Note that we could also have applied our extension to the h_{FF} heuristic (Hoffmann 2001).

As suggested by (Bonet and Geffner 2005), the `min-min` admissible heuristic $h_{m\text{-}m}$ is recursively defined for every reachable state $s \in \mathcal{S} \setminus \mathcal{G}$ by:

$$h_{m\text{-}m}(s) \leftarrow \min_{a \in app(s)} \left\{ C(s,a) + \min_{s' : T(s,a,s') > 0} h_{m\text{-}m}(s') \right\} \quad (3)$$

with the initial conditions: $h_{m\text{-}m}(s') = 0$ if $s' \in \mathcal{G}$ and $h_{m\text{-}m}(s') = +\infty$ otherwise. This heuristic counts the minimum number of steps required to reach a goal state in a non-deterministic relaxation of the domain. The `min-min` heuristic is well-informed but it naively searches in the original state space, so that it might explore as many states as non-heuristic algorithms. But clever heuristics that return a value lower or equal than $h_{m\text{-}m}$ still are admissible.

Let us give the intuition of the STRIPS relaxation heuristics by considering deterministic effects. As states are collections of atoms, only atoms added by successive actions need to be tracked down. As in (Bonet and Geffner 2001), we note $g_s(\omega)$ the cost of achieving an atom ω from a state s, i.e. the minimum number of steps required from s to have ω true. This value is computed by a forward chaining procedure where $g_s(\omega)$ is initially 0 if $\omega \in s$ and $+\infty$ otherwise:

$$g_s(\omega) \leftarrow \min_{\substack{a \in \mathcal{A} \text{ such that:} \\ \omega \in add(a)}} \{ g_s(\omega), cost(a) + g_s(prec(a)) \} \quad (4)$$

where $g_s(prec(a))$ denotes the cost of achieving the set of atoms in the preconditions of a. This requires to define the cost of achieving any set $\Delta \subseteq \Omega$ of atoms, what can be computed by aggregating the cost of each atom:

$$g_s(\Delta) = \bigoplus_{\omega \in \Delta} g_s(\omega) \quad (5)$$

When the fixed-point of eq. 4 is reached, the cost of achieving the set of goal states can be computed with eq. 5 and a heuristic value of s is $h_{\oplus}(s) = g_s(\mathcal{G})$. Two aggregation operators have been investigated by (Bonet and Geffner 2001): $\oplus = \max$ that gives rise to the h_{max} heuristic, such that $h_{max} \leqslant V^*$; $\oplus = \sum$ that provides the h_{add} heuristic, which is not admissible but often more informative.

Based on this helpful background, we can now extend h_{max} and h_{add} to the total criterion of the probabilistic case. For proof of admissibility, we are searching for a STRIPS relaxation heuristic whose value is lower than the `min-min` relaxation heuristic. Looking at eq. 3, this heuristic works

on a relaxed non-deterministic version of the original problem, known as the "all-outcome" determinization. This allows us to translate the search of a heuristic for the probabilistic problem into the search of a heuristic for a deterministic problem, as highlighted by the following proposition.

Proposition 1. *Let $\mathcal{M}$ be a goal-oriented MDP without reachable dead-ends. Let $\mathcal{D}$ be a deterministic planning problem (the "deterministic relaxation of $\mathcal{M}$"), obtained by replacing each probabilistic effect of each action of $\mathcal{M}$ by a deterministic action having the same precondition and the add and delete effects of the probabilistic effect. Then, an admissible heuristic for $\mathcal{D}$ is an admissible heuristic for $\mathcal{M}$.*

Proof. Let $app^{\mathcal{D}}$ be the applicability function for the deterministic problem. Eq. 3 reduces to $\mathrm{h}_{\text{m-m}}(s) \leftarrow \min_{a \in app^{\mathcal{D}}(s)} \{C(s,a) + \mathrm{h}_{\text{m-m}}(s')\}$, which is the update equation of optimal costs for $\mathcal{D}$. So the optimal cost of plans for $\mathcal{D}$ is equal to the value of the `min-min` relaxation for $\mathcal{M}$. Let h_{X} be an admissible heuristic for $\mathcal{D}$. h_{X} is lower than the optimal cost of plans for $\mathcal{D}$, i.e. than the value of $\mathrm{h}_{\text{m-m}}$ for $\mathcal{M}$, so it is admissible for $\mathcal{M}$. □

This proposition means that the $\mathrm{h}_{\max}$ heuristic computed in $\mathcal{D}$ is admissible for $\mathcal{M}$. It also demonstrates the admissibility of the heuristics used by `mGPT` (Bonet and Geffner 2005), but the `FF`-based one that is not admissible in $\mathcal{D}$.

However, such heuristics require to construct the deterministic relaxation of a goal-oriented MDP before solving it. To avoid this preliminary construction, we make this deterministic relaxation implicit in a new procedure that directly computes costs of atoms in the probabilistic domain:

$$g_s(\omega) \leftarrow \min_{\substack{a \in \mathcal{A} \text{ such that:} \\ \exists i \in [1; m_a], \omega \in add_i(a)}} \{g_s(\omega), cost(a) + g_s(prec(a))\} \quad (6)$$

Eq. 5 does not need to be changed for the probabilistic case. As in the deterministic case, we define the new $\mathrm{h}^+_{\max}$ for SSPs that is equal to $g_s(\mathcal{G})$ with the $\max$ aggregation operator.

Theorem 1. *The h^+_{max} heuristic is admissible for goal-oriented MDPs without reachable dead-ends.*

Proof. Let $\mathcal{M}$ be a goal-oriented MDP without reachable dead-ends and $\mathcal{D}$ be its deterministic relaxation. Eq. 6 for $\mathcal{M}$ boils down to eq. 4 for $\mathcal{D}$. Thus, $\mathrm{h}^+_{\max}$ in $\mathcal{M}$ has the same value as $\mathrm{h}_{\max}$ in $\mathcal{D}$. Since $\mathrm{h}_{\max}$ is admissible for $\mathcal{D}$, and using proposition 1, $\mathrm{h}_{\max}$ and thus $\mathrm{h}^+_{\max}$ are admissible in $\mathcal{M}$. □

We also define the new $\mathrm{h}^+_{\text{add}}$ for SSPs as $g_s(\mathcal{G})$ with the $\sum$ aggregation operator, but as in the deterministic case, $\mathrm{h}^+_{\text{add}}$ is not admissible. It is however more informative than $\mathrm{h}^+_{\max}$ and, as will be shown in the experiment section for the general discounted case, it is more efficient in practice.

Discounted Stochastic Shortest Path Problem

The previous $\mathrm{h}^+_{\max}$ and $\mathrm{h}^+_{\text{add}}$ heuristics, as well as heuristics by (Bonet and Geffner 2005), unfortunately are useless for goal-oriented MDPs where a policy execution may reach some dead-end state with a positive probability. As no goal state is reachable from a dead-end, $\mathrm{h}^+_{\max}$ and $\mathrm{h}^+_{\text{add}}$ may both return an infinite value for such state. Thus, because of eq. 2, the value of any preceding state will be infinite as well; after some iterations, this infinite value will propagate to at least one initial state. In fact, this fatal issue arises whatever the heuristic used: eq. 1 shows that, independently from heuristic values, the value of any dead-end state is equal to $+\infty$.

To cope with the divergence issue of the total criterion, we extend the previous SSP generalization of classical planning heuristics to *discounted* SSPs, which, like general MDP approaches, maximize the following *discounted* criterion:

$$\forall s \in \mathcal{S}, \pi^*(s) = \mathrm{argmin}_{\pi \in \mathcal{A}^S} E\left[\sum_{t=0}^{+\infty} \gamma^t C_t \mid s_0 = s, \pi\right] \quad (7)$$

where $0 < \gamma < 1$ is a discount factor that ensures the convergence of the series of discounted costs for all classes of MDPs. In particular, values of all dead-ends are now finite and properly propagate to the initial state in the Bellman equation. For goal-oriented MDPs, this criterion allows us to define the following class of problems, that includes SSPs for $\gamma = 1$, and that has always a solution.

Definition 3. A *Discounted* Stochastic Shortest Path Problem (DSSP) is a goal-oriented MDP whose optimization criterion is given by eq. 7.

As highlighted in the introduction of this paper, note that DSSPs are different from the transformation of discounted MDP without termination states to an equivalent SSP done in (Bertsekas and Tsitsiklis 1996): in our case, we reason about goal states of the problem, whereas in the aforementioned book, either the MDP does not have any goal state or the SSP has at least one proper policy (i.e. does not have any dead-end). In this section, we present an extension of classical planning heuristics to goal-oriented MDPs with potential dead-ends, using the discounted criterion and targeting the true goal states, contrary to (Bertsekas and Tsitsiklis 1996). We prove the admissibility of the extended heuristics under some assumptions for the original heuristic, and apply this extension to the $\mathrm{h}_{\max}$ and h_{add} heuristics.

When reasoning on individual states, a well-informed heuristic for DSSPs can be obtained by inserting γ in eq. 3. Unfortunately, contrary to the non-discounted case, this "discounted" $\mathrm{h}_{\text{m-m}}$ heuristic does not generalize well to atom-space reasoning: eq. 6 that gave rise to our $\mathrm{h}^+_{\max}$ and $\mathrm{h}^+_{\text{add}}$ heuristics for SSPs, cannot be modified by simply inserting γ in the equation. The reason is that γ discounts *future* values, whereas eq. 6 is a forward procedure that updates *past* values. Naively inserting γ in this equation would lead to totally incoherent heuristic values.

For the sake of generality, we keep general positive costs in the definition of DSSPs, as did (Bonet and Geffner 2005) for instance in the case of SSPs. However, we caution the reader against using DSSPs with non-unit costs. Indeed, because of the discount factor and different transition costs, near dead-ends could become more interesting than far goal-states; thus, optimal policies could lead to dead-ends with a higher probability than to goal states. Yet, traditional approaches using the total criterion do not provide a better model, because they cannot solve the problem if there are reachable dead-ends. A more sophisticated approach, relying on bi-optimization of goal reachability probability and

costs of only paths reaching the goal, has been very recently proposed (Teichteil-Königsbuch 2012). This approach provides new theoretical foundations as well as some simple algorithmic means, to solve goal-oriented MDPs with reachable dead-ends and general costs (unit or non-unit, positive or negative). However, it has not been yet applied in a heuristic search context.

The generalization of our $\mathrm{h}^+_{\max}$ and $\mathrm{h}^+_{\mathrm{add}}$ heuristics to the discounted case relies on the computation of a lower bound on the minimum non-zero transition cost received along all paths starting from a state s, by means of a procedure inspired by eq. 6, that also reasons on individual atoms. This lower bound is required in our approach to handle general non-unit costs, but it is simply 1 in the unit-costs case. We compute an admissible heuristic for DSSPs by discounting successive steps and lowering all transition costs with this bound. To this purpose, we state and prove the following theorem, that is valid for general DSSPs, based on lower bounds on the minimum cost received along paths and the minimum number of steps required to reach a goal state.

Theorem 2. *Let s be a non-goal state of a DSSP, $c_s > 0$ be a lower bound on the minimum over all non-zero transition costs received from s by applying any policy, and $d_s > 1$ a lower bound on the number of steps required to reach a goal state from s by applying any policy ($d_s = +\infty$ if s is a dead-end). Then, the h^γ function defined as follows is an admissible heuristic:*

$$h^\gamma(s) = \left\{ \begin{array}{ll} c_s \sum_{t=0}^{d_s-1} \gamma^t & \textit{if } d_s < +\infty \\ c_s/(1-\gamma) & \textit{otherwise} \end{array} \right\} = c_s \frac{1-\gamma^{d_s}}{1-\gamma}$$

Proof. Let $\Phi^{\pi^*}(s)$ be the infinite but countable set of execution paths of π^* starting in s. Let $P(\phi)$ and $c(\phi)$ be resp. the probability and the (accumulated) cost of a path $\phi \in \Phi^{\pi^*}(s)$. Let $d(\phi)$ be the length of a path ϕ until it reaches a goal state ($d(\phi) = +\infty$ if ϕ does not reach a goal state). By definition of goal-oriented MDPs, all costs received after a goal is reached are equal to zero. By noting C_t^ϕ the cost received at time t along a path ϕ, we have: $c(\phi) = \sum_{t=0}^{+\infty} \gamma^t C_t^\phi = \sum_{t=0}^{d(\phi)-1} \gamma^t C_t^\phi \geqslant c_s \sum_{t=0}^{d_s-1} \gamma^t$ because $d(\phi) \geqslant d_s$ and $C_t^\phi \geqslant c_s$. Thus: $V^{\pi^*}(s) = \sum_{\phi \in \Phi^{\pi^*}(s)} P(\phi) c(\phi) \geqslant \sum_{\phi \in \Phi^{\pi^*}(s)} P(\phi) \left(c_s \sum_{t=0}^{d_s-1} \gamma^t \right) = c_s \sum_{t=0}^{d_s-1} \gamma^t$ because $\sum_{\phi \in \Phi^{\pi^*}(s)} P(\phi) = 1$. In the special case $d_s = +\infty$ (i.e. s is a dead-end), $V^{\pi^*}(s) \geqslant c_s \sum_{t=0}^{+\infty} \gamma^t = c_s/(1-\gamma)$. □

The previous theorem provides a new admissible heuristic for all discounted goal-oriented MDPs. In the next section, we will propose atom-space procedures to efficiently compute the lower bounds c_s and d_s.

New STRIPS relaxation heuristics for DSSPs

Atom-space computation of c_s. We can reuse the idea of the $\mathrm{h}_{\max}$ and $\mathrm{h}_{\mathrm{add}}$ heuristics, consisting in forgetting delete effects of the STRIPS domain. We collect all non-zero costs received by successively applying all applicable actions in the relaxed domain and keep the minimum one. Let $s \in \mathcal{S}$ be a state, $\omega \in \Omega$ be an atom and $c_s(\omega)$ be the minimum non-zero transition cost received until ω is added, starting from s. This value is computed by a forward chaining procedure where $c_s(\omega)$ is initially 0 if $\omega \in s$ and $+\infty$ otherwise:

$$c_s(\omega) \leftarrow \min_{\substack{a \in \mathcal{A} \text{ such that:} \\ \exists i \in [1; m_a], \omega \in add_i(a) \\ cost(a) > 0, c_s(prec(a)) < +\infty}} \left\{ c_s(\omega), cost(a), \overline{c_s(prec(a))} \right\} \tag{8}$$

where, for all set of atoms $\Delta \subseteq \Omega$, $c_s(\Delta)$ is the minimum cost received to add all atoms in Δ:

$$c_s(\Delta) = \min_{\omega \in \Delta} c_s(\omega) \tag{9}$$

When the fixed point of update equation 8 is reached, a lower bound on the minimum non-zero transition cost received along all paths starting from a state s is: $\mathrm{c_m}(s) = c_s(\Omega)$. For initialization purposes, we define a filter $\overline{c_s(prec(a))}$ as $cost(a)$ if $prec(a) \subseteq s$ and as $c_s(prec(a))$ otherwise.

Atom-space computation of d_s. d_s can be obtained by computing the $\mathrm{h}^+_{\max}$ heuristic on a slight variation of the input domain, where all non-zero costs are set to 1. Indeed, in such a case, $g_s(\omega)$ represents the minimum number of steps required to add the atom ω. Thus, $\mathrm{h}^+_{\max}(s) = \max_{\omega \in \mathcal{G}} g_s(\omega)$ is lower than the minimum number of steps required so that all atoms in $\mathcal{G}$ have been added, what is the intended lower bound. This simple observation combined with Theorem 2 allows us to derive the admissible heuristic $\mathrm{h}^\gamma_{\max}$ for DSSPs from $\mathrm{h}^+_{\max}$, the latter being computed by assuming that all non-zero action costs are equal to 1. We also derive a non-admissible but well-informed heuristic $\mathrm{h}^\gamma_{\mathrm{add}}$ from $\mathrm{h}^+_{\mathrm{add}}$. We note $\mathrm{h}^{1,+}_{\max}$ and $\mathrm{h}^{1,+}_{\mathrm{add}}$ the resp. $\mathrm{h}^+_{\max}$ and $\mathrm{h}^+_{\mathrm{add}}$ heuristics obtained when all non-zero action costs in eq. 6 are replaced by 1.

Definition 4. Let $\mathrm{c_m}(s)$ be a lower bound on the cost received from a state s by applying any policy as defined above, and $\mathrm{h}^{1,+}_{\mathrm{X}}$ a heuristic with values in $\mathbb{N}_+$ such that $\mathrm{h}^{1,+}_{\mathrm{X}}(s) = 0$ if s is a goal and $\mathrm{h}^{1,+}_{\mathrm{X}}(s) = +\infty$ if s is a dead-end. The $\mathrm{h}^\gamma_{\mathrm{X}}$ heuristic is defined as:

$$\mathrm{h}^\gamma_{\mathrm{X}}(s) = \mathrm{c_m}(s) \frac{1-\gamma^{\mathrm{h}^{1,+}_{\mathrm{X}}(s)}}{1-\gamma}$$

The $\mathrm{h}^{1,+}_{\max}$ and $\mathrm{h}^{1,+}_{\mathrm{add}}$ both satisfy above conditions, so that we can define $\mathrm{h}^\gamma_{\max}$ and $\mathrm{h}^\gamma_{\mathrm{add}}$. Note that the heuristics of (Bonet and Geffner 2005), that work only for goal-oriented MDPs without reachable dead-ends, can be generalized to the discounted case in the same way, so become helpful in presence of dead-ends. Moreover, with unit costs ($\mathrm{c_m}(s) = 1$ if s is not a goal state) and no dead-ends (we can safely set γ to 1), $\mathrm{h}^\gamma_{\mathrm{X}}$ reduces to $\mathrm{h}^+_{\mathrm{X}}$.

Finally, we prove that the $\mathrm{h}^\gamma_{\max}$ heuristic is admissible for all goal-oriented MDPs, with or without reachable dead-ends.

Theorem 3. *The h^γ_{max} heuristic is admissible for all goal-oriented MDPs (with or without dead-ends).*

Proof. Directly follows from Theorems 1 and 2. □

Another nice result is that $h^1_{max} = h^+_{max}$ and $h^1_{add} = h^+_{add}$: it means that h^γ_{max} and h^γ_{add} generalize h^+_{max} and h^+_{add} for DSSPs and for SSPs as well, rather than simply deriving them to a different use case.

To our knowledge, the new h^γ_{max} heuristic, generalized from classical planning heuristics, is an original atom-space *admissible* heuristic for all goal-oriented MDPs with or without dead-ends. Its complexity, like the original h_{max}, is *polynomial in the number of atoms*.

Experimental evaluations

Testbed description

We tested our approach on 150 problems from the fully observable probabilistic track of the 2008 International Planning Competition (IPC). Each problem was solved by 5 MDP heuristic algorithms (detailed below) with our h^γ_{max} and h^γ_{add} heuristics for each of them, and by previous winners of the past IPCs using their last available versions from their websites and optimized settings (such as 'all-outcome' effects for determinization-based planners and the ATLAS library): `FF-REPLAN` (Yoon, Fern, and Givan 2007), `FPG` (Buffet and Aberdeen 2009) and `RFF` (Teichteil-Königsbuch, Kuter, and Infantes 2010). Each problem was solved by $6 \times 2 + 3 = 15$ planners; in order to evaluate all planners on all problems in a reasonable time, each planner was given at most 10 minutes and 1.8 GB of RAM per problem. The machine used was an Intel Xeon running at 2.93 GHz in 64 bits mode on Ubuntu Linux.

We used the `mdpsim` simulator from IPCs to evaluate different criteria. As soon as a planner converges or exceeds its time or memory limits, the simulator executes the planner's policy 100 times starting from the problem's initial state. At each step of an execution, the simulator samples an outcome among the effects of the action specified by the planner's policy in the current state. Each run stops when a goal state is reached or after 1000 steps. Each IPC problem was solved as SSP with unit costs, and three relevant criteria were evaluated: **% Goal**: percentage of runs where the policy reaches a goal state; **Length (avg)**: average length of execution runs *that reach a goal state*; **Time**: total time (in seconds) to optimize the policy and execute the runs *that reach a goal state*. A problem is considered *solved* by a planner if its policy reaches a goal state with a non zero probability.

Candidate MDP heuristic algorithms.

We plugged h^γ_{max} and h^γ_{add} in several heuristic algorithms: `Improved-LAO*` (Hansen and Zilberstein 2001), `LRTDP` (Bonet and Geffner 2003), `LDFS` (Bonet and Geffner 2006), `sLAO*` (Feng and Hansen 2002), and `sRTDP` (Feng, Hansen, and Zilberstein 2003). For all planners and problems, the discount factor was $\gamma = 0.9$ and the precision $\epsilon = 0.001$. Of course, some problems without reachable dead-ends could have been solved with $\gamma = 1$ and achieve better '% goal' end 'length' criteria; but as we want to be domain-independent and solve all problems without analyzing them beforehand to detect possible reachable dead-ends (as in `exploding blocksworld`), we set the same $\gamma = 0.9 < 1$ for all problems.

Experimental results

We aggregated the results into three distinct categories for better readability: IPC winners, h^γ_{max} and h^γ_{add}. For each problem, the value of a given criterion for a given category is the best value of the criterion over all algorithms of the category. We first compare the overall performance of categories on a domain-by-domain basis, then we analyze performances of individual algorithms inside each category.

We first present "point clouds" plots that compare, for each criterion, all algorithms in a given category (e.g. IPC winners) against all algorithms of another category. Such plots exhibit a global view of categories performances without any aggregation (such as averages). Each point corresponds to one problem and two algorithms: it gives the criterion value of one algorithm in the x-axis category against one algorithm in the y-axis category for the same problem. (each problem appears as many times as the number of combinations of algorithms from both categories).

Comparison with winners of the past IPCs. The first row of Figure 1 compares all algorithms in the h^γ_{max} and h^γ_{add} categories, against all algorithms in the 'IPC winners' category. The '% goal' criterion shows that MDP heuristic algorithms with h^γ_{max} or h^γ_{add} globally reach the goal with a higher frequency than overall IPC winners for the problems which they could solve. Another interesting result is that `exploding blocksworld` problems do contain reachable dead-ends, and they are now solved by goal-oriented MDP heuristic approaches — what was not the case of `mGPT` (Bonet and Geffner 2005) nor `sLAO*` in the 2004 competition (Younes et al. 2005) — and better than IPC winners. The same holds for the `triangle-tireworld` problems. Moreover, for the problems that could be solved, the average lengths of h^γ_{max} and h^γ_{add} algorithms are often comparable to those of IPC winners, except for some domains where visible differences appear: for instance in `rectangle-tireworld`, our heuristics outperform IPC winners; this is not the case in `zenotravel`. Finally, as we expected, IPC winners slightly outperform h^γ_{max} or h^γ_{add} algorithms in terms of total time.

h^γ_{max} vs. h^γ_{add} comparisons. The second row of Figure 1 compares all MDP heuristic algorithms using the h^γ_{max} heuristic, against the same algorithms using the h^γ_{add} heuristic. Remind that h^γ_{max} is admissible but not h^γ_{add}: yet, strangely enough, the '% goal' criterion plot shows that h^γ_{add} achieves slightly better performances for this criterion for the `exploding blocksworld` and `triangle-tireworld` domains. The reason is that h^γ_{max} is less informative than h^γ_{add}: heuristic algorithms using h^γ_{max} could not converge within the 10 minutes limit for this domain (contrary to the same algorithms using h^γ_{add}), so that they produced worse suboptimal policies. The 'time' criterion plot also confirms that h^γ_{max} often consumes more time resources (including optimization time) to reach the goal than h^γ_{add}. Finally, the 'length' plot shows that h^γ_{add} often achieves better performance than h^γ_{max} for this criterion.

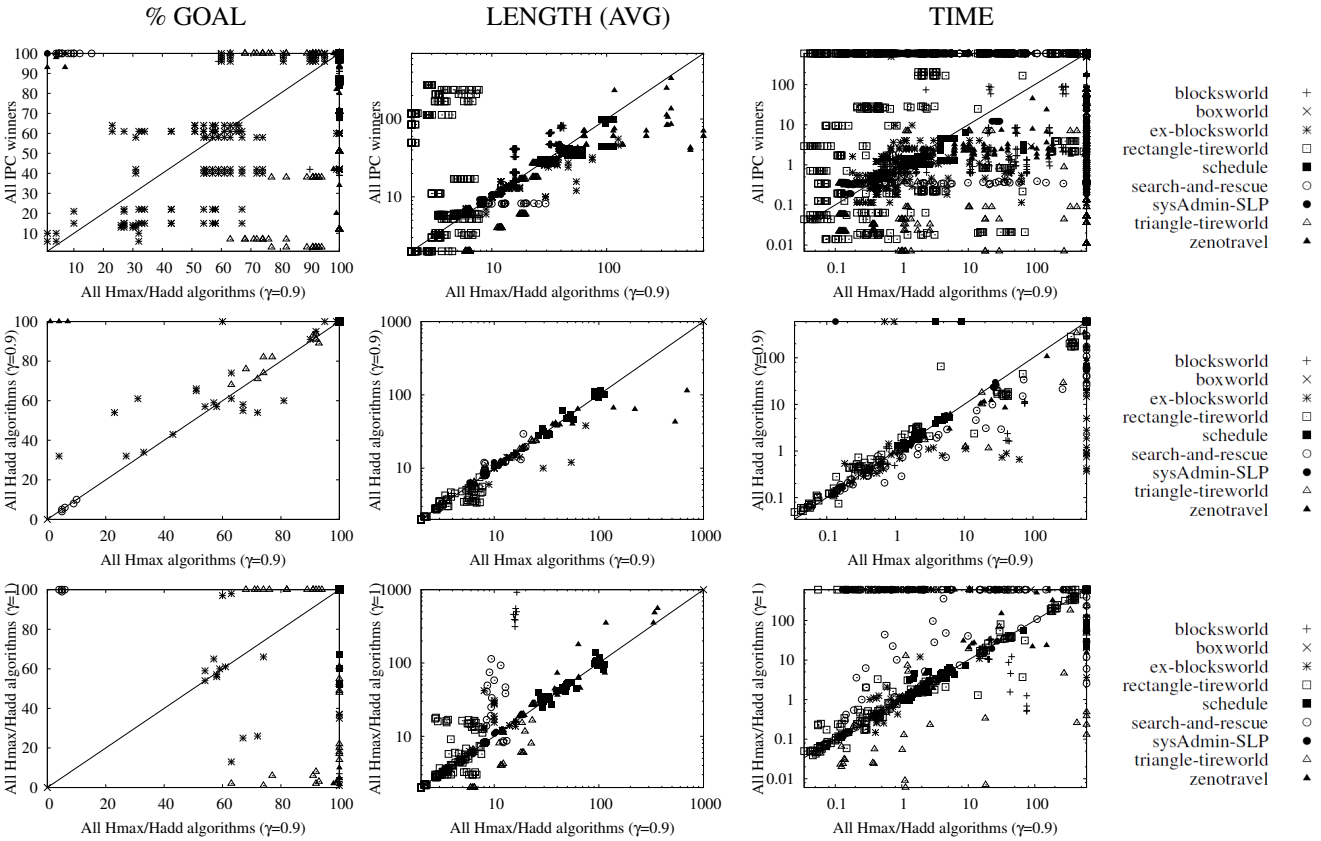

Figure 1: Face-to-face comparisons on problems of IPC 2008. 1st row: all $h^{\gamma=0.9}_{max}$ and $h^{\gamma=0.9}_{add}$ algorithms vs. all IPC winners. 2nd row: all $h^{\gamma=0.9}_{max}$ algorithms vs. all $h^{\gamma=0.9}_{add}$ algorithms. 3rd row: all $h^{\gamma=0.9}_{max}$ and $h^{\gamma=0.9}_{add}$ algorithms vs. all $h^{\gamma=1}_{max}$ and $h^{\gamma=1}_{add}$ algorithms.

$h^{0.9}_{max}$/$h^{0.9}_{add}$ vs. h^{1}_{max}/h^{1}_{add} comparisons. The third row of Figure 1 compares all MDP heuristic algorithms using the h^{γ}_{max} and h^{γ}_{add} heuristics with $\gamma = 0.9$, against themselves with $\gamma = 1$ (h^{+}_{max} and h^{+}_{add}). The '% Goal' criterion shows that discounted heuristics reach the goal more frequently than undiscounted ones in most domains except `search-and-rescue`, where the 0.9 discount factor is too small and prevents the planner from seeing the goal, and `triangle-tireworld`, where the best among $\gamma = 0.9$ and $\gamma = 1$ depends on the heuristic used. In all other domains, undiscounted heuristics return an infinite value so that the final policy is almost random and reaches the goal with a low probability (a proper policy with $\gamma = 1$ outperforms discounted policies). The 'length' criterion shows that discounted heuristics reach the goal with significantly less steps than undiscounted ones, suggesting that the latter give rise to almost random policies for most domains. Finally, the 'time' criterion clearly points out that discounted heuristics solve far more problems than undiscounted ones, what highlights the relevance of solving DSSPs as we propose, instead of SSPs as usual in the literature. Again, the SSP transformation by (Bertsekas and Tsitsiklis 1996) does not apply, because the goal of the transformed problem does not correspond to actual goals of the original problem.

Detailed comparisons. Table 1 presents more detailed algorithm performances in each category: IPC winners, h^{γ}_{max} and h^{γ}_{add}. The second column represents the number of problems solved by a given algorithm (i.e. it returns a policy reaching the goal with a positive probability). Other columns give for each criterion the number of problems where a given algorithm is the best among all algorithms that solved the same problems, over all categories or over its category (in addition to the average of the criterion over all problems for this algorithm): the highest the better. This table shows four main results: (1) all MDP heuristic algorithms achieve comparable performances on all criteria, meaning that h^{γ}_{max} and h^{γ}_{add} are not really impacted by the encoding of the policy graph nor by the search ordering of the algorithm; (2) h^{γ}_{add} performs slightly better than h^{γ}_{max} on all problems (it is not admissible but well-informed); (3) heuristic MDP algorithms using h^{γ}_{max} or h^{γ}_{add} globally achieve better performances than all IPC winners for the '% goal' and 'length' criteria; (4) heuristic MDP algorithms using h^{γ}_{max} or h^{γ}_{add} globally solve as many problems as `FF-REPLAN` and `FPG`, and outperform them on all other criteria.

Conclusion

We provide new atom-space heuristics for probabilistic planning with dead-ends generalized from efficient classical

Table 1: Comparisons over all problems of IPC 2008 between algorithms in the three categories: IPC winners, h^{γ}_{max}, and h^{γ}_{add} ($\gamma = 0.9$); subcategories are: IPC winners without RFF, and best of h^{γ}_{max}/h^{γ}_{add}. For each criterion: a **bold** number indicates the best element in overall or in its category, a framed number indicates that the overall category is the best for this criterion, and a $\star$ symbol indicates the best overall category between h^{γ}_{max}/h^{γ}_{add} and FPG/FF−REPLAN. For columns 3 to 5: the format of cells is 'a *(b)* [c]', where 'a' (resp. 'b') is the number of problems where the corresponding algorithm is the best among the algorithms that solved this problem in *all* categories (resp. *its* category), and 'c' is the average of the criterion among all problems solved. (Sub)categories are compared among problems that were solved by at least one algorithm in each category.

Algorithm	# Problems solved		% Goal		Length (avg)		Time	
RFF	**89**		**7** ***(16)*** [83.22]		**4** ***(13)*** [76.82]		**4** *(9)* [13.20]	
FPG	44		6 *(11)* [83.56]		1 *(4)* [86.41]		0 *(0)* [44.10]	
FF−REPLAN	48		4 *(4)* [51.60]		**4** *(12)* [42.50]		0 ***(10)*** [20.34]	
FPG/FF−REPLAN	**73** $\star$		26		15		10	
IPC winners	**100**		43		26		**30**	
	h^{γ}_{max}	h^{γ}_{add}	h^{γ}_{max}	h^{γ}_{add}	h^{γ}_{max}	h^{γ}_{add}	h^{γ}_{max}	h^{γ}_{add}
Improved−LAO*	**55**	**68**	**7** *(24)* [91.74]	**7** *(26)* [94.66]	1 ***(12)*** [29.11]	2 ***(14)*** [26.21]	0 ***(15)*** [20.04]	2 ***(9)*** [36.50]
LRTDP	41	50	**7** ***(25)*** [95.14]	**7** *(26)* [94.69]	1 *(10)* [13.49]	1 *(13)* [16.95]	0 *(1)* [24.58]	0 *(4)* [29.69]
LDFS	38	52	**7** *(24)* [92.60]	**7** ***(27)*** [94.78]	1 *(11)* [20.97]	1 *(11)* [19.18]	1 *(4)* [26.83]	1 *(7)* [14.45]
sLAO*	32	33	**7** *(24)* [93.62]	**7** *(26)* [94.45]	2 *(10)* [15.28]	1 *(13)* [14.48]	0 *(1)* [72.46]	0 *(0)* [40.28]
sRTDP	33	36	6 *(23)* [81.18]	**7** ***(27)*** [78.19]	1 *(3)* [13.41]	2 *(7)* [14.70]	0 *(1)* [102.59]	0 *(0)* [88.71]
Sub. h^{γ}_{max}/h^{γ}_{add}	55	**70**	42 $\star$	**45** $\star$	23 $\star$	**24** $\star$	9	**11** $\star$
h^{γ}_{max}/h^{γ}_{add}	72		**46** $\star$		**35** $\star$		19 $\star$	

planning heuristics: h^{γ}_{max}, for which we prove its admissibility, and h^{γ}_{add}, which is not admissible but often more informative than the former. These heuristics allow MDP heuristic algorithms to work in presence (or not) of dead-ends, i.e. when all possible policies reach with a positive probability a state from where no goal state can be achieved. Experimental results show that implementing efficient atom-space heuristics at the core of MDP heuristic algorithms rather than in an underlying deterministic planner, achieves nearly competitive performances with winners of past international planning competitions in terms of number of problems solved and computation time, while providing better goal reaching probability and average length to the goal.

One could argue that our heuristics do not take probabilities into account. Yet, many state-space heuristics for goal-oriented MDPs like the min−min relaxation also forget probabilities in their computation: the reason is that probabilities of paths in the relaxed problem are often very different from the ones in the original probabilistic problem, leading to non-admissible or poorly-informed heuristics. However, In the future, we intend to design well-informed atom-space heuristics that incorporate probabilities in their computations. Other interesting future research involves the direct insertion of γ inside the update equations of h^{+}_{max} or h^{+}_{add}, as well as the extension of our heuristics to hybrid goal-oriented MDPs with discrete and continuous state variables.

References

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *AIJ* 129(1-2):5–33.

Bonet, B., and Geffner, H. 2003. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *Proc. ICAPS*.

Bonet, B., and Geffner, H. 2005. mGPT: A probabilistic planner based on heuristic search. *JAIR* 24:933–944.

Bonet, B., and Geffner, H. 2006. Learning depth-first search: A unified approach to heuristic search in deterministic and non-deterministic settings, and its application to mdps. In *Proc. ICAPS*.

Buffet, O., and Aberdeen, D. 2007. FF+FPG: Guiding a policy-gradient planner. In *Proc. ICAPS*, 42–48.

Buffet, O., and Aberdeen, D. 2009. The factored policy-gradient planner. *Artificial Intelligence* 173(5-6):722–747.

Feng, Z., and Hansen, E. A. 2002. Symbolic heuristic search for factored markov decision processes. In *Proc. AAAI*, 455–460.

Feng, Z.; Hansen, E. A.; and Zilberstein, S. 2003. Symbolic generalization for on-line planning. In *Proc. UAI*, 209–216.

Gazen, B. C., and Knoblock, C. A. 1997. Combining the expressivity of UCPOP with the efficiency of graphplan. In *Proc. ECP*, 221–233.

Hansen, E. A., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *AIJ* 129(1-2):35–62.

Hoffmann, J. 2001. FF: The fast-forward planning system. *AI Magazine* 22(3):57–62.

Joshi, S.; Kersting, K.; and Khardon, R. 2010. Self-taught decision theoretic planning with first order decision diagrams. In *Proc. ICAPS*, 89–96.

Kolobov, A.; Mausam; and Weld, D. S. 2010. Classical

planning in MDP heuristics: With a little help from generalization. In *Proc. ICAPS*, 97–104.

Teichteil-Königsbuch, F.; Kuter, U.; and Infantes, G. 2010. Incremental plan aggregation for generating policies in MDPs. In *Proc. AAMAS*, 1231–1238.

Teichteil-Königsbuch, F. 2012. Stochastic safest and shortest path problems. In *AAAI'12: Proceedings of the 26th AAAI conference on Artificial intelligence*. AAAI Press.

Wu, J.-H.; Kalyanam, R.; and Givan, R. 2008. Stochastic enforced hill-climbing. In *Proc. ICAPS*, 396–403.

Yoon, S. W.; Ruml, W.; Benton, J.; and Do, M. B. 2010. Improving determinization in hindsight for on-line probabilistic planning. In *Proc. ICAPS*, 209–217.

Yoon, S.; Fern, A.; and Givan, R. 2007. FF-Replan: A baseline for probabilistic planning. In *Proc. ICAPS*, 352–359.

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