Planning using Stochastic Enforced Hill-Climbing

Jia-Hong Wu and Rajesh Kalyanam and Robert Givan
Electrical and Computer Engineering, Purdue University, W. Lafayette, IN 47907
{*jw, rkalyana, givan*}*@purdue.edu*

Abstract

Enforced hill-climbing is an effective deterministic hill-climbing technique that deals with local optima using breadth-first search (a process called "basin flooding"). We propose a stochastic generalization of enforced hill-climbing for use in fully-observable probabilistic planning problems. We assume a provided heuristic function estimating expected cost to the goal with flaws such as local optima and plateaus that thwart straightforward greedy action choice. While breadth-first search is effective in exploring basins around local optima in deterministic problems, for stochastic problems we build and solve a local Markov-decision process model of the basin in order to find a good escape policy exiting the local optimum.

For the planning competition, we use a novel heuristic function derived from the ideas in the effective re-planner FF-Replan. This new "controlled-randomness FF heuristic" is the deterministic FF heuristic computed on the simple determinization of the probabilistic problem that makes available a deterministic transition wherever a probabilistic transition was possible.

Introduction

Heuristic estimates of distance-to-the-goal have long been used in deterministic search and deterministic planning[1]. Such estimates typically have flaws such as local extrema and plateaus that limit their utility. Various methods such as simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983; Cerny 1985) and A* (Nilsson 1980) search have been developed for handling flaws in heuristics. More recently, excellent practical results have been obtained simply by "flooding" local optima using breadth-first search; this method is called "enforced hill-climbing". (Hoffmann and Nebel 2001)

The essential goal of enforced hill-climbing is to find a descendant that is strictly better than the current state in heuristic value by performing a local search. Once such a descendant is found, the planner moves to that state and this process is then repeated. The effectiveness of enforced hill-climbing is demonstrated in the success of FF.

[1]This paper is based on a submission to ICAPS 08.

We propose a novel tool for stochastic planning by generalizing enforced hill-climbing to stochastic domains. The goal of enforced hill-climbing is to reach a state that is better than the current state. Here, we construct a breadth-first local Markov Decision Process (MDP) around any local optimum reached and seek a *policy* that expects to exit this MDP with a better valued state. Critical to this process is the direct incorporation of the probabilistic model in finding the desired policy. Here, we find this policy using value iteration on the local MDP, where the only rewards are the heuristic values assigned at the exits. As in enforced hill-climbing, breadth-first expansion around a state occurs only when there is no action available with Q-value (relative to the heuristic) achieving the current state's heuristic value. Note that although stochastic enforced hill-climbing is an explicit statespace technique, it can be suitable for use in astronomically large statespaces if the heuristic used has some flaws of bounded size.

We use a novel heuristic function based on the determinization from the successful re-planner FF-Replan. This heuristic function, the "controlled-randomness FF heuristic" (CR-FF), is the FF heuristic (Hoffmann and Nebel 2001) computed on the FF-Replan (Yoon, Fern, and Givan 2007) determinization of the probabilistic problem[2]. The determinization used in FF-Replan is constructed by creating a new deterministic action for each possible outcome of a stochastic action while ignoring the probability of such outcome happening. This effectively allows the planner to control the randomness in executing actions, making this determinization a kind of relaxation of the problem. We note that stochastic enforced hill-climbing applies to other heuristic functions as well.

Technical Background and Related Work

Goal-oriented Markov decision processes We give a brief review of Markov decision procceses (MDPs) specialized to goal-region objectives. For more detail on MDPs, please see (Bertsekas and Tsitsiklis 1996) and (Sutton and

[2]The deterministic FF heuristic, described in (Hoffmann and Nebel 2001), from FF planner version 2.3 available at http://members.deri.at/~joergh/ff.html, computes a sequential relaxed-plan length to the goal using a sequence of ordered subgoal steps.

Barto 1998).

A stochastic shortest path Markov decision process (MDP) M is a tuple (S, A, R, T) with finite state and action spaces S and A, reward function $R : S \times A \times S \to \mathbb{R}$, and a transition probability function $T : S \times A \to \mathcal{P}(S)$ that maps (state, action) pairs to probability distributions over S. To avoid discounting while still assigning finite value to each state, we require that S contain a zero-reward absorbing state $\perp$, i.e., such that $R(\perp, a, s) = 0$ and $T(\perp, a, \perp) = 1$ for all $s \in S$ and $a \in A$, and that all policies (as defined below) lead to this state with probability 1.

Goal-oriented MDPs are MDPs where there is a subset $G \subseteq S$ of the statespace, containing $\perp$, such that: (1) $R(s, a, s')$ is zero whenever $s \in G$ and minus one otherwise, and (2) $T(g, a, \perp)$ is one for all $a \in A$ and $g \in G$.

Given *policy* $\pi : S \to A$ for an MDP, the value function $V^\pi(s)$ gives the expected cumulative reward obtained from state s selecting action $\pi(s)$ at each state encountered[3]. There is at least one optimal policy π^* for which $V^{\pi^*}(s)$, abbreviated $V^*(s)$, is no less than $V^\pi(s)$ at every state s, for any other policy π. The following "Q function" evaluates an action a with respect to a future-value function V,

$$Q(s, a, V) = \sum_{s' \in S} T(s, a, s')[R(s, a, s') + V(s')].$$

Recursive Bellman equations use $Q()$ to describe V^* and V^π as follows. First, $V^\pi(s) = Q(s, \pi(s), V^\pi)$. Then, $V^*(s) = \max_{a \in A} Q(s, a, V^*)$. Also using $Q()$, we can select an action greedily relative to any value function. The policy Greedy(V) selects, at any state s, a randomly selected action from $\arg\max_{a \in A} Q(s, a, V)$.

Value iteration iterates the operation $V'(s) = \max_{a \in A} Q(s, a, V)$, computing the "Bellman update" V' from V, producing a sequence of value functions converging to V^*, regardless of the initial V used.

While goal-based MDP problems can be directly specified as above, they may also be specified exponentially more compactly using planning languages such as PPDDL (Younes et al. 2005), which we use for our experiments. Note that our technique avoids converting the PPDDL problem explicitly into the above form, but constructs a sequence of smaller problems of this form as it encounters heuristic flaws, as we discuss next.

Local Markov decision processes Given an MDP $M = (S, A, R, T)$, we define the sub-MDP induced by state set $S' \subseteq S$ and heuristic function $h : S \to \mathbb{R}$ to be the MDP that equals M inside S', but gives terminal reward according to h upon exiting S'. Formally, the sub-MDP induced by S' and h is given by (S, A, R', T'), where R' and T' are given as follows. Let s be any state in S and a any action in A. For any state $s' \in S'$, we take $R'(s', a, s) = 0$ and $T'(s', a, s) = T(s', a, s)$. For any state x in $S - S'$, $R'(x, a, s) = h(x)$ and $T'(x, a, \perp) = 1$.

The rough motivation for removing action costs for the analysis within S' is that such actions are being considered by our method to remediate a flawed heuristic; the action cost to reach a state of higher heuristic value is a measure of the magnitude of the flaw in the heuristic, but we remove this cost from the analysis in order to express the subgoal of "reaching a state with higher heuristic value". Instead of diluting this subgoal by adding in action costs, our methods attempt to minimize the cost to reaching a heuristic improvement by constructing choices for S' in a breadth first manner.

Note: throughout this paper we use the terminology of "climbing" in "value" rather than working with "cost", even though we think of heuristics as measuring distance to the goal. We simply use the negative value of the distance-to-go heuristic as our value function.

Stochastic Enforced Hill Climbing

Deterministic enforced hill-climbing (Hoffmann and Nebel 2001) searches for a strictly improving successor state and returns a path from the initial state to such a successor. This path is an action sequence that guarantees reaching the desired successor. In a stochastic environment, there may be no single improved descendant that can be reached with probability one, as the actions may have multiple stochastic outcomes. Thus, in stochastic enforced hill-climbing, we generalize the idea of searching for a single strictly improved successor state to finding a policy that expects to improve on the heuristic value of the initial state inside a dynamically constructed sub-MDP. As formalized in "Local Markov decision processes" above, the sub-MDP ignores the cost to reach such improved states; the secondary goal of minimizing this cost is embodied by constructing the smallest sub-MDP enabling success, as described formally below. Note that, in contrast to replanning techniques, this approach enables the system to adjust to the uncertainty in action outcomes without the need to replan.

We present pseudo-code for stochastic enforced hill-climbing in Figure 1, and explain the terminology used in the pseudo-code next. The algorithm assumes a non-positive heuristic function $h : S \to \mathbb{R}$ as input that assigns zero to all goal states. Stochastic enforced hill-climbing iteratively builds and solves sub-MDPs and seeks improved policies inside such sub-MDPs. Each sub-MDP is induced as defined previously by a state set Σ together with the heuristic function h. We use two parameters k and α to control the aggressiveness with which the sub-MDPs are constructed in an attempt to escape the local optimum or plateau. The horizon parameter k limits the number of steps from the entry state s_0; the heuristic radius parameter α limits the heuristic distance from $h(s_0)$ that states in the sub-MDP may have. We assume some schedule for selecting more and more aggressive values of k and α to construct increasingly large sub-MDP problems seeking an exit. Our algorithm applies to any such schedule.

For any pair (k, α) in the schedule, we define a state set $\Sigma_{k\alpha}$. For any α, we define $\Sigma_{0\alpha}(s_0) = \{s_0\}$. We de-

[3]Formally, we restrict consideration to domains where all policies reach a goal state with probability one from all states, to avoid discounting. In practice we can and do handle deadend states by assigning them large negative values if they are recognized by simple reachability tests.

Stochastic enforced hill-climbing

1. Repeat
2. // for some schedule of (k, α) pairs (k_i, α_i)
3. $s_0 \leftarrow$ current state
4. $i = 0$, $\Sigma = \Sigma_{k_0 \alpha_0}$
5. Repeat
6. $V \leftarrow$ Value iteration in sub-MDP(Σ,h)
7. $i = i + 1$, $\Sigma = \Sigma_{k_i \alpha_i}$
8. Until $V(s_0) > h(s_0)$ or $|\Sigma| > \tau$
9. Follow Greedy(V) until exit Σ or a state visited σ times
10. Until goal achieved

Figure 1: *Pseudo-code for stochastic enforced hill-climbing.*

fine an operation Extend on a state set to be Extend(Σ) $= \{s' \mid \exists s \in \Sigma, \exists a \in A, P(s, a, s') > 0\}$. We can then iteratively extend $\Sigma_{k\alpha}$ to be $\Sigma_{(k+1)\alpha}(s_0) = \Sigma_{k\alpha} \cup \{s' \in Extend(\Sigma_{k\alpha}) \mid |h(s') - h(s_0)| \leq \alpha\}$.

Thus, in line 6 of Figure 1 , value iteration is conducted for increasingly large sub-MDPs around s_0, seeking a policy improving $h(s_0)$. Depending on the schedule chosen for k and α, implementation choices may allow reuse of states-pace and value information as larger MDPs are constructed, expanding on smaller ones. The implementation may also choose to proceed down the schedule without interleaving value iteration if no new states of high value are added to the sub-MDP.

Early termination The primary termination condition for repeated sub-MDP construction is the discovery of a policy improving on the heuristic estimate of the initial state. However, for flawed heuristic functions that overestimate state value, especially in domains with unavoidable dead-ends, this may not be possible. For this reason, in practice, we impose an overall threshold τ on the size of the sub-MDP constructed and if τ is exceeded the algorithm stops and uses the best policy found to that point.

Once a sub-MDP is constructed assigning $V(s_0) > h(s_0)$ prior to having at least τ states, the system executes the greedy policy within the last sub-MDP constructed until the policy exits the sub-MDP. Again, in practice, we impose a bound σ on the number of times a state may be repeated during the execution.

References

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

Cerny, V. 1985. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. Optim. Theory Appl.* 45:41–51.

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

Kirkpatrick, S.; Gelatt, Jr, C.; and Vecchi, M. 1983. Optimization by simulated annealing. *Science* 220:671–680.

Nilsson, N. 1980. *Principles of Artificial Intelligence*. Tioga Publishing, Palo Alto, CA.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning*. MIT Press.

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

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