Set-Additive and TSP Heuristics for Planning with Action Costs and Soft Goals

Emil Keyder
Universitat Pompeu Fabra
Passeig de Circumvalació 8
08003 Barcelona Spain
emil.keyder@upf.edu

Abstract

We introduce a non-admissible heuristic for planning with action costs, called the *set-additive heuristic*, that combines the benefits of the *additive heuristic* used in the HSP planner and the *relaxed plan heuristic* used in FF. The set-additive heuristic is defined mathematically and handles non-uniform action costs like the additive heuristic but like FF's heuristic, it encodes the cost of a specific *relaxed plan* and is therefore compatible with FF's helpful action pruning and its enforced hill climbing search. This new formulation is used to introduce a further variation that takes certain deletes into account by forcing the values of certain multivalued variables in the relaxed plan to be spanned by a path rather than by a tree. We show last how soft goals can be compiled away and report empirical results using a modification of the FF planner that incorporates these ideas, leading to a planner that is as robust as FF but capable of producing better plans in a broader set of contexts.[1]

Planning Model and Heuristics

We consider planning problems $P = \langle F, I, O, G\rangle$ expressed in Strips, where F is the set of relevant atoms or fluents, $I \subseteq F$ and $G \subseteq F$ are the initial and goal situations, and O is a set of (grounded) actions a with preconditions, add, and delete lists $Pre(a)$, $Add(a)$, and $Del(a)$ respectively, all of which are subsets of F.

For each action $a \in O$, there is also a *non-negative cost* $cost(a)$. We take the cost $cost(\pi)$ of a plan $\pi = a_1, \ldots, a_n$ to be

$$cost(\pi) = \sum_{i=1,n} cost(a_i) \tag{1}$$

The search for plans is guided commonly by heuristics that provide an estimate of the cost-to-go that are extracted automatically from the problem encoding P. We discuss two of the most common heuristics below.

In order to simplify the definition of some of the heuristics, we introduce in some cases a new dummy End action with *zero cost*, whose preconditions G_1, ..., G_n are the goals of the problem, and whose effect is a dummy atom G.

[1] This is a shorter version of the paper by the same name by Emil Keyder & Héctor Geffner submitted to the ICAPS '07 workshop on heuristics. For extensions of the concepts discussed here and further information, please consult this paper.

The Additive Heuristic

Under the delete-relaxation P^+, the assumption that all subgoals are *independent* is normally false but results in a simple heuristic function (Bonet & Geffner 2001):

$$h_a(s) \stackrel{\text{def}}{=} h_a(G; s) \tag{2}$$

This function can be efficiently computed in every state s visited in the search:

$$h(p; s) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } p \in s \\ \min_{a \in O(p)}[h(a; s)] & \text{otherwise} \end{cases} \tag{3}$$

where $h(p; s)$ stands for an estimate of the cost of achieving the atom p from s, $O(p)$ is the set of actions in the problem that add p, and

$$h(a; s) \stackrel{\text{def}}{=} cost(a) + \sum_{q \in Pre(a)} h(q; s) \tag{4}$$

stands for the cost of achieving the preconditions of an action a and applying it.

The Relaxed Planning Graph Heuristic

Unlike h_a, the heuristic h_{FF} used in the FF planner makes no independence assumption for approximating h^+, instead computing one plan for P^+ which is not guaranteed to be optimal. This is done by a Graphplan-like procedure (Blum & Furst 1995), which due to the absence of deletes, constructs a planning graph with no mutexes, from which a plan $\pi_{\text{FF}}(s)$ is extracted backtrack-free (Hoffmann & Nebel 2001). The heuristic $h_{\text{FF}}(s)$ is set then to $|\pi_{\text{FF}}(s)|$.

The Set-Additive Heuristic

In the additive heuristic, the value of the best supporter a_p of p in s, $h(a_p; s)$, is propagated to obtain the heuristic value of p, $h_a(p; s)$. In contrast, in the *set-additive* heuristic, the best supporter a_p of p itself is propagated, and supports are combined by the *set-union* rather than the *sum* operation, resulting in a function $\pi(p; s)$ that represents a *set of actions* which can be defined similarly to $h_a(p; s)$:

$$\pi(p;s) = \begin{cases} \{\} & \text{if } p \in s \\ \pi(a_p;s) & \text{otherwise} \end{cases} \quad (5)$$

where

$$a_p = \text{argmin}_{a \in O(p)} Cost(\pi(a;s)) \quad (6)$$

$$\pi(a;s) = \{a\} \bigcup \{\cup_{q \in Pre(a)} \pi(q;s)\} \quad (7)$$

$$Cost(\pi(a;s)) = \sum_{a' \in \pi(a;s)} cost(a') \quad (8)$$

The *set-additive heuristic* $h^s_a(s)$ for a state s is then defined as

$$h^s_a(a) = Cost(\pi(G;s)) \quad (9)$$

Though $\pi(p;s)$ is a *set* and not a *sequence* of actions, its definition ensures that the actions it contains can be ordered into an action sequence that is a *relaxed plan* for p in the relaxed problem P^+ given start state s. The primary advantage of this relaxed plan over that computed by FF is that it attempts to minimize the *cost* of the plan rather than its size.

Testing the Heuristics

We tested the three heuristics h_a, h^s_a, and h_{FF} in combination with the Enforced Hill Climbing (EHC) and WA* search algorithms. All combinations were implemented within the framework of Metric-FF. When testing h^s_a in combination with EHC search, the relaxed plan computed from the planning graph was replaced with the plan computed by the set-additive heuristic. Modified versions of the domains Satellite, Driverlog, Depots, Rovers, and Zenotravel were used for the experiments.

- **FF vs. FF(h^s_a):** EHC with the set-additive heuristic yields higher quality plans in domains in which there are significant differences in the costs of the relaxed plans computed for the initial state, indicating that it pays off to take cost information into account. The differences in quality change from domain to domain.
- **Time Overhead:** FF(h^s_a) takes longer than normal FF (typically by a factor between 4 – 10) but scales up similarly. The reasons are two: the overhead due to propagating sets rather than numbers in the heuristic computation, and the fact that the plans that FF(h^s_a) finds are sometimes longer(though with less overall cost). This overhead does not affect coverage in the domains studied.
- **Heuristics in WA*:** when the heuristics h_a, h^s_a, and h_{FF} are used in the context of the WA* search described above, the first two heuristics do better than the latter, both in terms of quality and coverage.
- **Heuristics for Uniform Costs:** The heuristic values resulting from h^s_a and h_{FF} are very much alike in Strips domains, and both are consistently lower than the normal additive heuristic h_a that tends to overcount.

Richer Labels and the TSP Heuristic

We now consider labels $L(p;s)$ that result from treating one designated multivalued variable X in the problem in a special way. A multivalued variable X is a set of atoms $x_1, \ldots, x_n$ such that exactly one x_i holds in every reachable state. We refer to the atoms x_i as encoding the different values of X. We show that part of the problem space can be solved by a TSP procedure that is integrated into the h^s_a heuristic described above.

In order to define the graph upon which the TSP algorithm will be used in the computation of the heuristic, we replace the set of actions $\pi(p;s)$ that constitute a relaxed plan for p in Equation 5), with a *pair of sets*: roughly, the actions in $\pi(p;s)$ that do not affect the TSP variable X, denoted by $\pi_x(p;s)$, and the X-atoms required by these actions, denoted as $\alpha_x(p;s)$.

The TSP heuristic $h_X(s)$ is then defined as:

$$h_X(s) \stackrel{\text{def}}{=} Cost(\langle \pi_x(G;s), \alpha_x(G;s) \rangle) \quad (10)$$

with the set of actions $\pi_x(p;s)$, the set of X-atoms $\alpha_x(p;s)$, and the function $Cost$ mapping such pairs into numbers, defined below.

$$\langle \pi_x(p;s), \alpha_x(p;s) \rangle \stackrel{\text{def}}{=} \begin{cases} \langle \{\}, \{\} \rangle & \text{if } p \in s, \text{ else} \\ \langle \pi_x(a_p;s), \alpha_x(a_p;s) \rangle \end{cases} \quad (11)$$

where

$$a_p = \text{argmin}_{a \in O(p)} Cost(\langle \pi_x(a;s), \alpha_x(a;s) \rangle)$$

$$\pi_x(a;s) = \{a\} \bigcup \{\cup_{q \in Pre(a), q \notin X} \pi_x(q;s)\}$$

$$\alpha_x(a;s) = (Pre(a) \cap X) \bigcup \{\cup_{q \in Pre(a), q \notin X} \alpha_x(q;s)\}$$

$$Cost(\pi_x(a;s), \alpha_x(a;s)) = \sum_{a' \in \pi_x(a;s)} cost(a') + TSP_x(\alpha_x(a;s);s) \quad (12)$$

and $TSP_X(V;s)$ stands for an estimate of the cost of the best plan that achieves all atoms $x \in V$, starting from s. This plan can be computed by a fast but suboptimal TSP algorithm over a graph G_V with vertex set $V \cup \{x_d\}$, where x_d is a dummy vertex, and edges (x_i, x_j) with costs c_{ij}:

$$c_{i,j} = \begin{cases} D_s(x_i, x_j) & \text{if } x_i \neq x_d \text{ and } x_j \neq x_d \\ 0 & \text{if } x_i = x_d \text{ and } x_j = x_s, \text{ or } x_j = x_d \\ \infty & \text{if } x_i = x_d \text{ and } x_j \neq x_s, x_j \neq x_d \end{cases}$$

where x_s is the X-atom true in s, the dummy vertex x_d is linked to x_s with zero cost, and all $x_i \in X$ are linked to x_d with zero cost as well, so that the TSP tours for computing $TSP_X(V;s)$ will effectively start at x_s and visit all atoms x in V before reaching the dummy vertex x_d.

The *distance matrix* $D_s(x_i, x_j)$ encodes the cost of achieving x_j from the state s_i obtained from s by deleting x_s and adding x_i, and can be computed by any delete-based heuristic.

Soft Goals

In many problems, there is a preference over atoms p in a problem expressed by positive rewards $rwd(p)$, so that plans are sought that achieve all the 'hard' goals (if any) while satisfying such preferences as much as possible. We denote that a plan π makes an atom p true as $\pi \models p$. The value $v(\pi)$ of a plan that achieves all hard goals can then be formalized as the sum of the gathered rewards minus the cost of the plan:

$$v(\pi) = \sum_{p:\pi \models p} rwd(p) - \sum_{a_i \in \pi} cost(a_i) \qquad (13)$$

so that plans with max value are sought.

Variations of this model of planning with soft goals have been recently considered in (Smith 2004; Sanchez & Kambhampati 2005; Bonet & Geffner 2006). Here we want to show that it is possible to compile away soft goals.

Let P be a Strips planning problem extended with cost information $c(a) \geq 0$ over its actions, and reward information $rwd(p) \geq 0$ over a subset of its atoms. We will assume that atoms p with strictly positive reward $rwd(p) > 0$ cannot be deleted.

We define a Strips problem P' with action costs $c(a) \geq 0$ and *no rewards*, such that there is a direct correspondence between the optimal plans for P and P'.

Let $S(P)$ stand for the set of 'soft goals' in P:

$$S(P) = \{p \mid p \in F \ \wedge \ rwd(p) > 0\}$$

and let

$$S'(P) = \{p' \mid p \in S(P)\}$$

be a set of atoms not in P.

Then for $P = \langle F, I, G, O \rangle$ where F is the set of fluents, I and G are the initial and goal situations, and O is the set of actions, the problem $P' = \langle F', I', G', O' \rangle$ can be defined as:

- $F' = F \cup S'(P)$
- $I' = I$
- $G' = G \cup S'(P)$
- $O' = O \cup \{Collect(p)\ , Forgo(p) \mid p \in S(P)\}$

where $Collect(p)$ has precondition p, effect p', and cost 0, while $Forgo(p)$ has an empty precondition, effect p' and cost equal to $rwd(p)$.

Proposition 1 (Elimination of Soft Goals) *An action sequence π is an optimal plan for the problem P with* persistent soft goals *if and only if π is the result of stripping away the $Collect$ and $Forgo$ actions from an optimal plan for the problem P' with* no soft goals.

More Experiments

We tested the $h_X(s)$ heuristic with both the WA* and EHC search algorithms, in addition to all combinations discussed above, on 3 domains.

- **FF vs. FF(h^s_a)vs $h_X(s)$:** The TSP heuristic finds markedly better plans in a simple grid domain in which the agent must collect packages. This is also true of a rover-style domain with compiled soft goals.
- **Time overhead** The TSP heuristic does impose a large time overhead compared to FF in these types of domains, but this is partly due to the fact that longer plans of lower cost are found.

Summary

We have introduced a new non-admissible heuristic for planning, the *set-additive heuristic*, that combines the benefits of the *additive* and *relaxed plan* heuristics. The motivation is similar to the work in (Sapena & Onaindia 2004; Fuentetaja, Borrajo, & Linares 2006), but rather than modifying the planning graph construction or extraction phases to take action costs into account, we have modified the cost-sensitive additive heuristic to yield relaxed plans. The resulting formulation suggests further refinements that can result from the propagation of symbol labels (supports) rather than numbers in the basic formulation.

The TSP heuristic $h_X(s)$ aims to approximate the cost of the relaxation P^+_X of the problem P where the only deletes that are preserved are the ones involving the atoms x_i associated with X.

We show also that *soft goals* can be compiled to produce a problem with action costs but no rewards, through a simple and fast procedure.

References

Blum, A., and Furst, M. 1995. Fast planning through planning graph analysis. In *Proceedings of IJCAI-95*, 1636–1642. Morgan Kaufmann.

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

Bonet, B., and Geffner, H. 2006. Heuristics for planning with penalties and rewards using compiled knowledge. In *KR*, 452–462.

Fuentetaja, R.; Borrajo, D.; and Linares, C. 2006. Improving relaxed planning graph heuristics for metric optimization. In *Proc. 2006 AAAI Workshop on Heuristic Search, Memory Based Heuristics and its Applications*, 79–86.

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

Sanchez, R., and Kambhampati, S. 2005. Planning graph heuristics for selecting objectives in over-subscription planning problems. In *Proc. ICAPS-05*.

Sapena, O., and Onaindia, E. 2004. Handling numeric criteria in relaxed planning graphs. In *Advances in Artificial Intelligence: Proc. IBERAMIA 2004, LNAI 3315*, 114–123. Springer.

Smith, D. E. 2004. Choosing objectives in over-subscription planning. In *Proc. ICAPS-04*, 393–401.