Integrating Actions Preconditions Difficulty within the Relaxed Plan Heuristic Measure

Tomás de la Rosa and **Raquel Fuentetaja**
Departamento de Informática, Universidad Carlos III de Madrid
Avda. de la Universidad, 30. Leganés (Madrid). Spain
trosa@inf.uc3m.es, rfuentet@inf.uc3m.es

Abstract

Most of the heuristic search based planning systems perform guided search evaluating states to compute a heuristic measure. Although recent planners are quite efficient, the time spent in computing the heuristic measure is still an issue that the community has to address. In this work we present an extension to the heuristic of the relaxed plan introduced by the FF Planner. We propose to integrate in the heuristic measure the actions preconditions difficulty, which is computed during the relaxed plan extraction phase. Results show that the number of evaluated states and the time to compute plans are decreased using this integrated heuristic.

Introduction

Heuristic search is nowadays one of the main techniques for plan generation. It have been shown that using a domain independent heuristic measure significantly increases planning system performance. The principle of this type of heuristic search is to compute the heuristic solving a simplified (or relaxed) version of the original problem, which in general is easier to solve. Then, the solution to the relaxed problem is used to derive a heuristic measure, which represents an estimation of the distance from the current state to the problem goal. Ignoring the delete list of the STRIPS domains description, first introduced by (McDermott 1996), has become the most common relaxation used to derive heuristics. Several work has been done on this approach. For example, HSP (Bonet & Geffner 2001) used this kind of relaxation. Then, FF (Hoffmann & Nebel 2001a) improved this heuristic extracting it from a relaxed GRAPHPLAN (Blum & Furst 1995). This idea was based on the extraction of a solution from a planning graph computed for the relaxed problem. Other works use this approach to extend the planning tasks to temporal planning or planning with resources, such as SAPA (Do & Kambhampati 2003) or (Haslum & Geffner 2001).

Although these planning systems are quite efficient in many benchmark domains, they have to deal with the issue of spending a great part of the planning time evaluating states to obtain the heuristic measure. This fact is specially important on problems with many object instances. One way to address this issue is to obtain more accurate heuristic values in order to decrease the number of evaluated stated.

This paper presents a complementary heuristic measure to the relaxed plan distance computed by FF (Hoffmann & Nebel 2001b). The aim is to decrement the planning time using a heuristic which is expected to decrease the number of evaluated states. Our idea is based on the fact that additional information from the relaxed GRAPHPLAN could be used in the heuristic computation and the search process. YAHSP (Vidal 2004) is an example of using this additional information from the relaxed GRAPHPLAN when it introduced the Lookahead Strategy, a novel way of using the helpful actions, first introduced by FF.

In the following sections, we briefly describe how FF computes its standard heuristic, how we compute the new heuristic, and then we show the results of comparing both heuristics within three of the traditional IPC Planning Domains. Finally we present some conclusions to this work.

Heuristic of the Relaxed Plan

The heuristic of the relaxed GRAPHPLAN consists on building in any search state a planning graph from which a solution to the relaxed task could be extracted. Then the length of the relaxed solution is taken as an estimation of how far the goal is from the current state. The basic algorithm is shown in Figure 1.

Function $expand\_relaxed\_graphplan(s_0, G, A)$
  $t = 0$;
  $P_0 = s_0$;
  `while` $G \not\subseteq P_t$ `do`
    $A_t = \{a \in A \mid prec(a) \subseteq P_t\}$
    $P_{t+1} = P_t \cup add(a), \forall a \in A_t$
    `if` $P_{t+1} = P_t$ `then` fail `endif`
    $t = t + 1$
  `endwhile`
  final_layer$= t, succeed$

Figure 1: Traditional Graphplan expansion for relaxed planning tasks.

The planning graph in the relaxed case is simply represented as a sequence $P_0, A_0, \ldots, A_{t-1}, P_t$ of propositions sets and actions sets. These are built incrementally starting with $P_0 = s_0$ as the initial layer, and iteratively inserting the add effects of all applicable actions. Once the GRAPHPLAN is expanded, a sequential relaxed plan is extracted. The heuristic measure is the result of counting the number of actions in this relaxed plan. In the expanded planning graph there is no mutual exclusion between actions in the same layer as it was in the original GRAPHPLAN (Blum & Furst 1995). As a result, the process of the relaxed plan extraction can be done in polinomial time (Hoffmann & Nebel 2001b).

A Complementary Heuristic Measure

The process of the relaxed plan extraction consists on a backward chaining of actions through the graph such that an action is selected if it achieves at least a precondition of a selected action in the next level. Since finding an optimal linearization for actions in the relaxed plan extraction phase is a NP-Complete task (Hoffmann & Nebel 2001b), the process is heuristically guided to allow the heuristic estimator to return a solution as soon as possible. When more than one achiever could be selected in a graph layer, the decision consists on selecting the achiever whose preconditions are easier. This can be estimated using the difficulty measure D, which is computed using equation 1.

$$D(o) = \sum_{p \in pre(o)} min\{i|\ p\ in\ fact\ layer\ at\ time\ step\ i\} \tag{1}$$

In the search process we can see that some successors of a state could have the same heuristic measure as their father, because they have the same number of actions in the relaxed plan. These values do not reflect which of the successors of a state have less difficulty to achieve its relaxed plan actions. The relaxed plans extracted for two different successors of the same state could be different even if they have the same length. In this case, the difficulty to achieve each relaxed plan actions is probably different also. Therefore, having into account this measure together with the standard heuristic could help to distinguish states that otherwise will have the same heuristic evaluation. Based on this fact we propose to integrate the actions preconditions difficulty in the heuristic function. We think this could guide the search discriminating states with the same heuristic evaluation. Thus, a small number of states will be explored and the planning system performance will be improved.

To integrate the action preconditions difficulty within the standard heuristic we should have into account that the objective is to discriminate between states with the same heuristic evaluation. Therefore, we should maintain the relation with respect to the heuristic values between two states with different heuristics. As the standard heuristic estimates the number of operators from the current state to the goals, we guarantee that the relation is maintained if we add the complementary measure as a value between 0 and 1. Considering this, our new heuristic H-Plus is calculated as follows.

$$h_{plus}(S) = h(S) + (1 - \frac{1}{\sum_{k=1...|h|} D(O_k)}) \tag{2}$$

where S is the evaluated state, $h(S)$ the standard FF heuristic and O_k the k-action in the extracted relaxed plan. The values belong to the open interval $(h(S)\ , h(S) + 1)$. We can consider the decimal part of this measure as the difficulty of executing the relaxed plan extracted. If the sum of the preconditions difficulty is 0, the new heuristic measure is not considered and the standard heuristic is used.

Experiments

For the experiments we have implemented the H-Plus heuristic within the FF Planner. We have chosen three of the International Planning Competition (IPC) Domains to test both heuristics.

Depots Domain

The Depots domain is part of the IPC-2002 Collection. In this domain trucks transport crates between places, and crates must be stacked onto pallets at their destination. For the experiments we generated a test set of 50 random problems which have between 13 to 19 objects instances. Figure 2 shows the results of the accumulated time used by the planner to solve the problems. We consider that the fact that the new heuristic slows down the search for small instances, as can be observed in the figure, is not relevant as the planning time for these instances is very small (between 0.01 and 0.07 seconds).

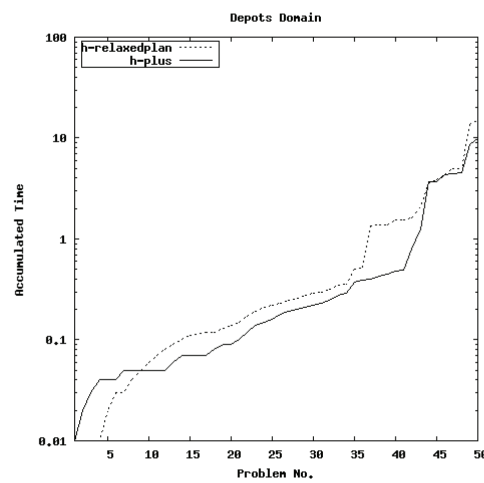

Figure 2: Accumulated time for the Depots random problems.

In this domain, 44% of the plans had the same length with both heuristics. The H-Plus heuristic leads in 8% of the problems to a better solution and in 48% to worse solutions. The reason is that we are trying to decrease the number of evaluated states with respect to FF, but not to improve the quality of the plans. The method to integrate the action preconditions difficulty in the standard heuristic measure suppose that a successor of a state with the same heuristic evaluation as the father can be selected by the Enforced Hill Climbing algorithm (this does not happen using just the standard heuristic). This selection may suppose the evaluation of less states though it may lead to the application of an unnecessary operator, making thus worse the quality of the obtained plan.

Table 1: Average of plans length and states evaluated in Depots domain.

Heuristic	Applied Ops.	States Eval.
H-Relaxedplan	24.68	2331.34
H-Plus	27.34	1544.22

Table 1 shows the average of the plans length and the number of states evaluated during the search. We can see that the H-Plus heuristic considerably decreases the number of states evaluated, but with the trade-off of the increase of plans length.

Rovers Domain

The Rovers domain is also part of the IPC 2002 Collection, which was inspired by planetary rovers problems. This domain requires that a collection of rovers navigate a planet surface, finding samples, taking pictures and communicating them back to a lander. For the experiments we generated a test set of 100 random problems which have between 10 to 33 objects instances. Figure 3 shows the results of the accumulated time used by the planner to solve the problems, which in this case is similar.

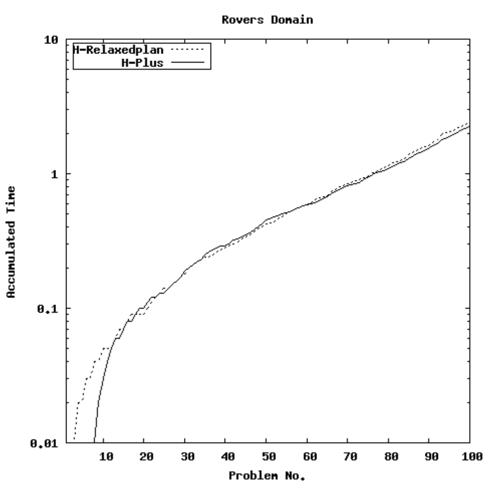

Figure 3: Accumulated time for the Rovers random problems.

For this domain 46% of the problems had the same solution length with both heuristics. The H-Plus heuristic leads in 54% of the problems to a worse solution and there wasn't any better plan than with the standard heuristic. Table 2 shows the average of plans length and the number of evaluated states during the search. We can see the same trade-off as in the Depots domain, but in this case the number of states evaluations skipped is less significant, since the Rovers domain is quite easier.

Table 2: Average of plans length and number of states evaluated in the Rovers domain.

Heuristic	Applied Ops.	States Eval.
H-Relaxedplan	24.92	108.10
H-Plus	26.37	103.83

We also tested this domain with the competition set. This set has 20 problems which have between 13 to 60 object instances. In this set we have a more wide range of problem difficulty to observe the heuristics performance.

Table 3: Solving Rovers Competition Set

	H-Relaxedplan		H-Plus	
Problem	Actions	Time	Actions	Time
1	10	0.0	12	0.0
2	8	0.01	8	0.0
3	13	0.01	13	0.01
4	8	0.0	8	0.01
5	22	0.01	22	0.01
6	38	0.02	38	0.01
7	18	0.01	18	0.01
8	28	0.02	27	0.02
9	36	0.03	36	0.03
10	36	0.03	39	0.03
11	36	0.03	38	0.03
12	19	0.01	20	0.02
13	46	0.07	46	0.05
14	28	0.03	29	0.03
15	42	0.05	44	0.11
16	45	0.07	47	0.09
17	57	0.16	55	0.14
18	44	0.27	45	0.3
19	70	1.10	72	0.9
20	96	5.98	104	5.73
Accumulated	700	7.91	721	7.53

Table 3 shows the time spent solving the competition problems and the number of actions in each plan. In this case 40% of the problems have the same solution length with both heuristics. The H-Plus heuristic leads in 2 problems to a better solution. In this set 4.56% less states were evaluated with the H-Plus heuristic. This confirm our initial assumption that the decrease in the number of evaluated states is proportional to the decrease in the time spent in solving a problem. Therefore we suggest a deeper analysis of the search algorithms performing with this heuristic, in our case the Enforced Hill-Climbing, to determine if it is possible to make adjustments to consistently maintain the plan quality measured in terms of solution length.

Logistics Domain

Our last experiment is with the Logistics Domain, in which packages need to be transported to given destinations. Destinations can be either post offices or airports. Airplanes fly between airports, and, trucks transport packages withing the same city. For comparing the heuristics, we have generated 40 random test problems which have between 38 to 95 objects instances. These problems are harder than problems in the IPC-2000 Logistic domain set. We decided to use a harder set because this domain is too easy to the planner and it doesn't spend significant planning time solving the competition set.

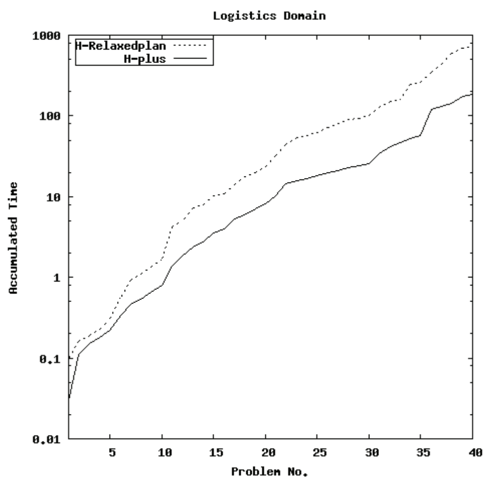

Figure 4: Accumulated time for the Logistics problems

Figure 4 shows the results of the accumulated time solving the problems. In this domain it can be observed again a better performance of the H-Plus heuristic with respect to the standard heuristic. In this case 17.5% of the problems had better solution length with the H-Plus Heuristic. The rest of problems had worse solution length with the new heuristic and none of them have the same solution length. One explanation to this difference is that the plans for the problems used in the logistics domain are very long (comparing with the plans in the other domains used in the experiments). If the solution plans are long is less frequent to obtain plans with the same length. However, it can be observed that the decrease on evaluated states was significantly bigger than in the other domains. Table 4 shows the average of solution length and the number of evaluated states.

Table 4: Average of plan length and number of evaluated states in the Logistics domain.

Heuristic	Applied Ops.	States Eval.
H-Relaxedplan	185.7	4654.85
H-Plus	193.7	1301.20

Conclusions

We have presented an extension to the relaxed-plan based heuristic, combining the standard heuristic (the number of actions in the extracted relaxed plan), with the actions preconditions difficulty. Since these difficulties are computed during the process of the relaxed plan extraction, it doesn't mean any considerable effort to obtain the complementary heuristic measure. We can assume also that though we have tested our new heuristic within a Enforced Hill-Climbing algorithm, the new heuristic could be applicable with any type of forward state-space heuristic algorithm, such as a standard Hill-Climbing or such as lookahead optimistic best-first used in YAHSP (Vidal 2004).

We have presented also empirical results within three of the IPC planning domains, which show the performance time improvement based on the less number of states evaluations required. This improvement on the performance time globally compromises the quality of plans in terms of solution length. Nevertheless, longer plans do not imply always a small number of states evaluations or viceversa. In a more detailed observation of the experiments, we have seen that sometimes the same solution lengths could be achieved by H-Plus heuristic with less, the same or even more evaluated states than the standard heuristic. This fact could be explained because the nature of the search algorithm. The Enforced Hill-Climbing is a greedy local search algorithm using a non-admissible heuristic.

Acknowledgments

This work has been partially supported by the Spanish MEC project TIN2005-08945-C06-05 and regional CAM-UC3M project UC3M-INF-05-016.

References

Blum, A. L., and Furst, M. L. 1995. Fast planning through planning graph analysis. In Mellish, C. S., ed., *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI-95*, volume 2, 1636–1642. Montréal, Canada: Morgan Kaufmann.

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

Do, M., and Kambhampati, S. 2003. Sapa: A scalable multi-objective metric temporal planner. *Journal of Artificial Intelligence Research (JAIR)* 20:155–194.

Haslum, P., and Geffner, H. 2001. Heuristic planning with time and resources. In *Proceedings IJCAI-01 Workshop on Planning with Resources*.

Hoffmann, J., and Nebel, B. 2001a. What makes the difference between HSP and FF? In *Proceedings IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*.

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

McDermott, D. 1996. A heuristic estimator for means-ends analysis in planning. In *Proceedings of the Third International Conference on AI Planning Systems*.

Vidal, V. 2004. A lookahead strategy for heuristic search planning. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, 150–160.