Conformant Planning Heuristics Based on Plan Reuse in Belief States

Dunbo Cai, Jigui Sun and **Minghao Yin**
College of Computer Science and Technology, Jilin University, Changchun, China, 130012
dunbocai@gmail.com

Introduction

Conformant planning involves generating plans under the condition that the initial situation and action effects are undeterministic and sensing is unavailable during plan execution. (Bonet & Geffner 2000) has shown that conformant planning can be transformed into a search problem in the space of belief states. Since then, heuristic based conformant planning has become a promising approach. Following the direction, some effective heuristics, such as those introduced in MBP (Bertoli & Cimatti 2002), Conformant-FF (Brafman & Hoffmann 2004) and POND (Bryce, Kambhampati, & Smith 2006), were proposed. An additive heuristic that obtains the heuristic value of a given belief state by summing the heuristic value of each world state is sensitive to the progress of the search procedure and often guides a search algorithm efficiently. However, it suffers from the problem of overestimating the distances-to-goal of belief states and therefore leads to overlong plans generally. One of the reasons is that the additive heuristic does not take into account relationships between different world states in a belief state. For example, an action sequence for one world state may have progressive effects on others. In this work, we use plan reuse information to make the additive heuristic more reasonable. The information is discovered with the aid of a classical planning heuristic function and in a real time learning style (Barto, Bradtke, & Singh 1995). The resulted heuristic is tested on a conformant planning system that is implemented as an extension of the classical planner Fast Downward (Helmert 2006) to the conformant setting. Preliminary results show that our heuristic is promising in getting better plans on tasks where plan reuse possibilities exist.

An Example

A conformant planning problem is a quadruple $P = \langle F, O, bs_0, G\rangle$ where F is a set of relevant fluents, O is a set of grounded actions, bs_0 is the initial situation and G is a set of goal conditions. We focus on deterministic actions and represents a belief state as $bs = \{s_0, s_1, \ldots, s_n\}$ where s_i is a world state that the agent deemed in. In the conformant planning as search in the belief space framework, heuristic functions are key elements.

On belief state, the additive heuristic is defined as: $h_{add}(bs) = \Sigma_{s\in bs}h(s)$, where $h(s)$ is the estimation of s computed by any classical planning heuristic function h. It often overestimates the distance of a belief state due to that it assumes a plan for a world state doesn't affect other ones in the same belief state. To show the aggressive effects among world states, we take a planning instance in the *Blocks world* domain from the conformant track of IPC5 (International Planning Competition). In the problem, two blocks A and B are involved and the goal is to place B on A. The initial situation is completely uncertain (see Figure 1). It is easy to get that the optimal costs to goal of s_0, s_1, s_2, s_3 and s_4 are 4, 3, 2, 1 and 0 respectively. If we use the additive heuristic, the overall estimate of the belief sate is 10, which is excessive.

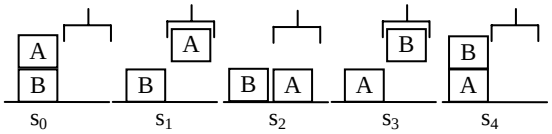

Figure 1: A conformant problem in Blocks world.

However, we may consider these world states according to their costs in an ascendant order. It is easy to see that the classical plan for s_3 is π_3 = <stack(B, A)>, and the one for s_2 is π_2 = <pickup(B), stack(B, A)>. We may find out that if pickup(B) is executed in s_2 then the agent will be in s_3 and may follow the plans for s_3. Similarly, agent may execute unstack(A, B) in s_0 to get into s_1, and then execute putdown(A) to get into s_2. As an observation, we may reuse plans of easier world states (with lower goal distances) in harder states (with higher goal distances). Following this strategy, we get the optimal cost of the belief state: 1+1+1+1 = 4, which is optimal. However, the strategy is not always applicable as there are belief states where one world state is unreachable from any other one. As a result, we pursue plan reuse heuristically and incorporate the information into the additive heuristic.

Heuristic Function

To predicate the chance and cost of reusing plans between two possible states in a belief state and form the overall estimation, we exploit a classical heuristic function h_{CS}. h_{CS} takes two arguments: the first one represents the initial situation and the second one represents the goal conditions. h_{CS} is used for estimating both the cost between two world states and the goal-distance of a world state.

A model for computing heuristics

With h_{CS} and a belief state $bs = \{s_0, s_1, \ldots, s_n\}$, a directed graph $\Pi = (V, E)$ can be constructed, where $V = \{n_i \mid s_i \in bs\} \cup \{n_g\}$, $E = \{<n_i, n_j>\} \cup \{<n_i, n_g>\}$. In such a graph, n_i represents a state s_i in bs, and n_g denotes the set of states that satisfies the goal condition. Each edge is associated with a weight that is calculated by h_{CS}: $w(n_i, n_j) = h_{CS}(s_i, s_j)$ is the estimated distance between s_i and s_j and $w(n_i, n_g) = h_{CS}(s_i, G)$ is the heuristic distance of state s_i. $w(n_i, n_g) = \infty$ indicates unreachability. Given a graph Π, the additive heuristic can be redefined as $h_{add}(bs) = \Sigma_{n \in V'} w(n, n_g)$, where $V' = V - \{n_g\}$.

To discover plan reuse information, for each node n_i we would like to find the shortest path from n_i to n_g. If n_i, n_j, n_g is such a path for n_i, then it is estimated that in state s_i reusing the plan of s_j will lead to a shorter plan for s_i. In this sense, after computing the shortest path for each node in V', we may define a heuristic function

$$h_{CF}(bs) = \Sigma_{n \in V'} w(n, suc(n)) \qquad (1)$$

where $suc(n)$ is the successor on the shortest path of n. However, the computation of h_{CF} is costly, as $|V| = |bs| - 1$ is always exponential in the initial situation. To reduce the computation effort, we computed (1) approximately.

```
algorithm LRT(belief_state bs)
  SNU(bs) = {s∈bs | h_CS(s, G) = max_{s∈bs} h_CS(s, G)} //initialize
  foreach s∈bs
    lrh(s) ← h_CS(s, G)
  foreach s ∈ SNU(bs)  //look one-step ahead
    best_suc(s) ← NULL
    foreach s′∈bs – SNU(bs)
      new_cost ← h_CS(s, s′) + h_CS(s′, G)
      if (new_cost ≤ h_CS(s, G)) then
          if ((new_cost == h_CS(s, G))
              && (best_suc(s) !=NULL)
              && (h_CS(s, s′) ≥ h_CS(s, best_suc(s)))) then
            continue
        h_CS(s,G) ← new_cost
        best_suc(s) ← s′
        //update helpful actions for s
        HA(s) ← helpful_actions(s, s′)
    lrh(s) ← h_CS(s, best_suc(s))
 return Σ_{s∈bs} lrh(s)
```

Figure 2: The pseudo code to compute h_{lrt}.

Look ahead one-step

We follow a greedy policy to discover plan reuse possibilities. Specifically, only states in SNU(bs) = $\{s \in bs \mid h_{CS}(s) = \max_{s \in bs} h_{CS}(s)\}$ are considered. For each $s \in$SNU(bs), we search for a better successor s' with $h_{CS}(s, s') + h_{CS}(s', G) \leq h_{CS}(s, G)$ if one exists. The resulted heuristic is called h_{lrt}, which is computed by the algorithm shown in Figure 2.

Preliminary results and Conclusions

We extended the Fast Downward planning system for conformant planning, and used the casual graph heuristic h_{CG} as an instance of h_{CS}. Both h_{add} and h_{lrt} were implemented and compared with the conformant planner Conformant-FF (CFF). The test was done on a wide range of benchmark problems on a PC with a Pentium 4 2.6 GHz processor and 512M memory. Here, we only show some results on the *Blocks world* and *Coins* domain from the conformant track of IPC5 (Table 1). The cut-off time is 300 seconds.

Table 1: Experimental Results: time (s)/plan length, dashes indicate time-outs or out of memory.

problems	b01	b02	b03	c07	c08	c09	c10
h_{add}	0.01 /8	0.08 /30	3.93 /179	0.15 /33	0.16 /42	0.18 /32	0.11 /31
h_{lrt}	0.01 /7	0.01 /17	2.11 /157	0.62 /33	0.65 /35	0.72 /38	0.22 /35
CFF	0.01 /6	-	-	0.19 /34	0.05 /28	0.02 /26	0.16 /38

In this work, we designed a conformant planning heuristic function based on a classical one in a way that pursues plan reuse between world states in the same belief state. h_{lrt} shows a marked improvement to h_{add}, yet our system does not scale well due to the explicit representation of belief states. We are currently working on ways to handle such problems. This work is mostly related to *FragPlan* (Kurien, Nayak, & Smith 2002), but is more flexible than that method.

Acknowledgements

We thank helpful comments from anonymous reviewers. This work was supported by the National Natural Science Foundation of China (No. 60773097 and 60573067) and the Doctoral Program of Higher Education of China under Grant No. 20050183065.

References

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proc. AIPS-00*.

Bertoli, P., and Cimatti, A. 2002. Improving heuristics for planning as search in belief space. In *Proc. AIPS-02*.

Brafman, R., and Hoffmann, J. 2004. Conformant planning via heuristic forward search: A new approach. In *Proc. ICAPS-04*.

Bryce, D.; Kambhampati, S.; and Smith, D. 2006. Planning graph heuristics for belief space search. *Journal of Articial Intelligence Research*, *26*.

Barto, A.; Bradtke, S.; and Singh, S. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 72.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Articial Intelligence Research*, 26.

Kurien, J.; Nayak, P.; and Smith, D. 2002. Fragment-based conformant planning. In *Proc. AIPS-02*.