PLANNING GRAPH HEURISTICS FOR SOLVING CONTINGENT PLANNING PROBLEMS

Incheol Kim and Hyunsik Kim
Department of Computer Science, Kyonggi University, San94-6, Yiui-Dong, Suwon, Korea

Keywords: Contingent Planning, Belief State Space, Search Heuristic, Planning Graph.

Abstract: In order to extract domain-independent heuristics from the specification of a planning problem, it is necessary to relax the given problem and then solve the relaxed one. In this paper, we present a new planning graph, Merged Planning Graph(MPG), and GD heuristics for solving contingent planning problems including both uncertainty about the initial state and non-deterministic action effects. MPG is a new version of the relaxed planning graph for solving the contingent planning problems. In addition to the traditional delete relaxations of deterministic actions, MPG makes the effect-merge relaxations of both sensing and non-deterministic actions. Parallel to the forward expansion of MPG, the computation of GD heuristics proceeds with analysis of interactions among goals and/or subgoals. GD heuristics estimate the minimal reachability cost to achieve the given goal set by excluding redundant action costs. Through experiments in several problem domains, we show that GD heuristics are more informative than the traditional max and additive heuristics. Moreover, in comparison to the overlap heuristics, GD heuristics require much less computational effort for extraction.

1 INTRODUCTION

Most of planning problems encountered in the real world environments have some uncertainty in both the initial state and action effects. We call it contingent planning to generate plans with conditional branching based on the outcomes of sensing actions for such environments with partial observability and non-determinism. A well-known technique for finding a contingent plan is to search over belief states (Bonet and Geffner, 2001). However, the size of the belief space for a contingent planning problem is exponentially larger than that of the corresponding state space. Therefore, in order to find a contingent plan in tractable time, we need powerful heuristics to guide efficiently the belief space search.

In order to extract domain-independent heuristics from the specification of a planning problem, it is necessary to relax the given problem and then solve the relaxed one (Hoffmann and Brafman, 2005). In this paper, we present a new planning graph, Merged Planning Graph (MPG), and GD heuristics for solving contingent planning problems. In addition to the traditional delete relaxations of deterministic actions, MPG makes the effect-merge relaxations of both sensing and non-deterministic actions. Parallel to the forward expansion of MPG, the computation of GD heuristics proceeds with analysis of interactions among goals and/or subgoals. GD heuristics estimate the minimal reachability cost to achieve the given goal set by excluding unnecessary action costs. Through experiments, the performance of our GD heuristics will be compared with those of other existing heuristics.

2 CONTINGENT PLANNING PROBLEMS

We assume to find effective heuristics for solving a contingent planning problem, like the one in Figure 1. The example problem given in Figure 1 is from the dinner domain, which includes one sensing action, *sense_garbage*, and one non-deterministic action, *cook*. We notice that both *sense_garbage* and *cook* actions have multiple possible outcomes as described in their action definitions. Figure 2 shows a contingent plan as solution for the example planning problem given in Figure 1. It contains multiple branches, every of which ends with a belief

Kim I. and Kim H..
PLANNING GRAPH HEURISTICS FOR SOLVING CONTINGENT PLANNING PROBLEMS.
DOI: 10.5220/0003830505150519
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 515-519
ISBN: 978-989-8425-95-9

state satisfying all goal conditions. While the occurrence of a sensing action during belief space search generates more than one AND branch, in general, the occurrence of a non-deterministic action generates more than one OR branch. Through this kind of AND-OR search on the belief state space, we can find a contingent plan whose every AND branch guarantees satisfaction of all goal conditions.

b_I = {unknown_garbage, ¬dinner, ¬present}
G = {dinner, present, clean}

Action	Preconditions	Effects
cook	{clean}	{dinner} {dinner, garbage, ¬clean}
carry	{garbage}	{clean, ¬garbage}
sense_garbage	{unknown_garbage}	{garbage, ¬clean ¬unknown_garbage} {clean, ¬garbage ¬unknown_garbage}
wrap	{clean}	{present}

Figure 1: An example of contingent planning problem.

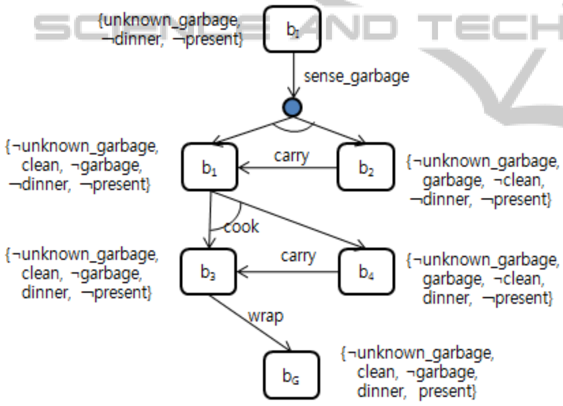

Figure 2: A contingent plan for the example problem described in Figure 1.

3 HEURISTICS FOR BELIEF SPACE SEARCH

Consider possible transitions from a belief state by executing an action. As illustrated in Figure 3 (a), execution of a deterministic action makes a deterministic transition to a single belief state. However, as shown in Figure 3 (b), execution of a sensing or non-deterministic action makes a transition to one of multiple different belief states. In our work, we assume that a sensing action has only two different effects.

In order to find a contingent plan from the belief space search, a good distance-based heuristic is needed. We should answer the questions of how to

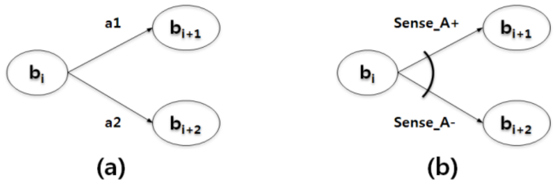

Figure 3: Possible transitions on a belief space.

compute belief state distances and which measures are most effective. Many approaches estimate belief state distances in terms of individual state to state distances between states in two belief states as shown in Figure 4 (a). The distance between two belief states in Figure 4 (b) can be estimated by aggregating the individual state distances in Figure 4 (a). Existing approaches to estimating the belief state distance are to select the maximal one from the corresponding individual state distances (*Max heuristics*), to sum all state distances (*Additive heuristics*), or to add some part of state distances by computing a relaxed plan (*Overlap heuristics*).

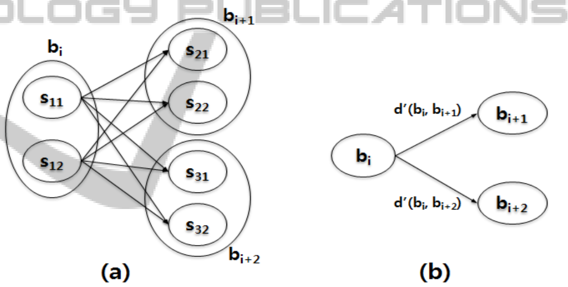

Figure 4: Estimating the distance between two belief states.

4 MERGED PLANNING GRAPH

The relaxed planning graph, which is an efficient data structure used to compute search heuristics for classical planning problems, is built from only *delete-relaxed* deterministic actions. However, in order to use the relaxed planning graphs for solving contingent planning problems, additional relaxations of sensing and non-deterministic actions are needed (Bryce, et al., 2006). Recent some works (Bonet and Geffner, 2005) tried to make *effect-determinization* of non-deterministic actions, which splits a non-deterministic action into multiple deterministic actions. In this paper, we propose a new kind of relaxations for both sensing and non-deterministic actions, *effect-merge relaxations*.

- *Effect-merge relaxation of a sensing action*: transformation of a sensing action $o_{s\text{-}f}$ having two different effect sets, $effect_1(o_{s\text{-}f})$={f} and

$effect_2(o_{s\text{-}f})=\{\neg f\}$, into the deterministic action $o_{s\text{-}f\text{-}merge}$ having a single effect set, $effect(o_{s\text{-}f\text{-}merge})=\{f, \neg f, \neg unknown\_f\}$.

- *Effect-merge relaxation of a non-deterministic action*: transformation of a non-deterministic action o_{nd} having k different effect sets, $effect_i(o_{nd})$ i=1, …, k, into the deterministic action $o_{nd\text{-}merge}$ having a single effect set, $effect(o_{nd\text{-}merge})=\bigcup_{i=1,\dots,k} effect_i(o_{nd})$.

With effect-merge relaxations, every sensing and non-deterministic action can be transformed into its corresponding deterministic action. We propose a new relaxed planning graph built from effect-merge relaxed actions instead of sensing and non-deterministic actions.

- *Merged Planning Graph* (MPG): the merged planning graph expanded from a belief state b_m, during belief space search to solve a contingent planning problem $P_{pond} = (b_I, G, O_d \cup O_{nd} \cup O_s)$, is built from multiple literal layers and action layers in the following way:
  - The initial literal layer L_0 includes all literals representing the belief state b_m.
  - The k-th action layer A_k is built from any actions $o \in O_d \cup O_{nd\text{-}merge} \cup O_{s\text{-}merge}$ whose every precondition is satisfied with literals on the k-th literal layer. O_d denotes the set of deterministic actions. $O_{nd\text{-}merge}$ and $O_{s\text{-}merge}$ represent the set of effect-merged non-deterministic actions and the set of effect-merged sensing actions respectively.
  - The (k+1)-th literal layer L_{k+1} is built by adding the delete-relaxed effects of all actions on the k-th action layer A_k into the set of literals on the k-th literal layer L_k.
  - When the literal layer L_{k+1} includes all goal literals in G, or is equal to the literal layer L_k, the graph expansion ends. L_{k+1} becomes the last layer of the merged planning graph for the belief state b_m.

5 GD HEURISTICS

Computation of our GD(Goal Dependency) heuristic for a belief state b_m proceeds parallel to the forward expansion of a merged planning graph (MPG) from the belief state b_m, layer to layer. Whenever the graph expands a new literal layer L_k, the set of goal literals $G_k \subset G$ put on the layer L_k is found, and then the minimal cost to reach G_k from the belief state b_m is computed based on the equation (1) and (2).

$$cost_{bm}(G_k) = \sum_{g \in G_k} cost_{bm}(g) \qquad (1)$$

$$cost_{bm}(g) = \min\{cost(o) + cost_{bm}(pre(o)) \mid g \in effect(o) \text{ and } o \in A_{k-1}\} \qquad (2)$$

In order to estimate the minimal cost to reach the goal set G_k, possible positive interactions among each goals $g \in G_k$ are analyzed using a data structure called *closeGoals*, as illustrated in Figure 5.

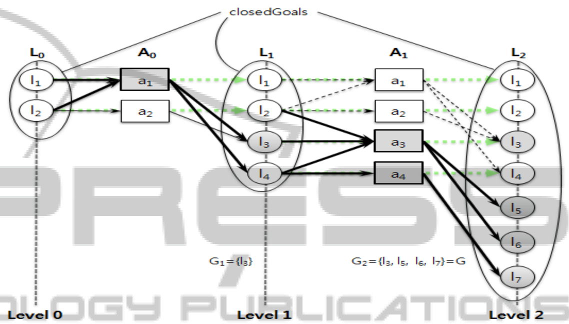

Figure 5: An example illustrating the process to compute GD heuristics.

By summing up the minimal costs to reach the goal set G_k, for k=0, ..., n, the GD heuristic for the belief state b_m is obtained, as formulated in (3).

$$h_{GD}(b_m) = \sum_{k=0}^{n} cost_{bm}(G_k) \qquad (3)$$

Figure 6 and 7 summarize the algorithm for computing the GD heuristic for a belief state b_m.

```
GD_Heuristics(b_m, G, O)
/* b_m: a belief state, G: the set of goals
   O: the set of effect-merged actions, i.e.
   O = O_d U O_nd_merge U O_s_merge        */
Begin
   total_cost = 0;
   L_0 = closedGoals = b_m;
   k = 1;
   while (G ⊄ L_k-1) {
        (A_k-1, L_k) = MPG_Expand_Level(L_k-1, O);
        G_k = Find_Goals(L_k, G);
        (goal_cost, closedGoals) =
                 Estimate_Cost(G_k, A_k-1, closedGoals);
        total_cost += goal_cost;
        k = k + 1;
   }
   return total_cost;
end
```

Figure 6: Algorithm for computing GD heuristics.

6 EXPERIMENTS

In order to evaluate the accuracy and computational efficiency of our GD heuristics based on the merged planning graphs (MPG), we conducted some

```
Estimate_Cost(G_k, A_{k-1}, closedGoals)
Begin
    goal_cost = 0;
    For each g∈G_k s.t. g∉ closedGoals do
        Select an action o from A_{k-1} s.t. g ∈ effect(o);
        If pre(o) ⊂ closedGoals then
            goal_cost += cost(o);
        else
            goal_cost += cost(o) + max_level(pre(o));
        closedGoals = closedGoals ∪ effect(o);
    End
    Return (goal_cost, closedGoals);
End
```

Figure 7: Estimate_Cost function.

experiments solving the contingent planning problems randomly generated from four different domains. Table 1 shows the reachability cost estimates of each different heuristic for the same initial belief state. We notice that the cost estimates of our GD heuristics are much closer to the actual minimal costs than those of the max and additive heuristics, and are not much worse than the overlap heuristics.

Table 1: Comparison of cost estimates.

Problems	Robot Domain					Block Domain				
Heuristics	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
Minimal Cost	5	6	8	9	11	3	5	7	10	14
Max	3	4	7	5	8	3	3	4	3	4
Additive	7	9	16	13	20	8	10	17	20	34
Overlap	5	6	9	8	10	5	5	8	10	16
GD	5	6	10	8	13	8	9	11	18	22
Problems	Dinner Domain					Truck Domain				
Heuristics	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
Minimal Cost	4	5	6	7	8	4	6	8	9	10
Max	2	2	3	4	4	3	4	5	6	7
Additive	7	6	9	11	14	5	9	12	14	17
Overlap	3	5	6	7	8	4	6	8	8	9
GD	4	5	6	8	9	4	6	8	9	11

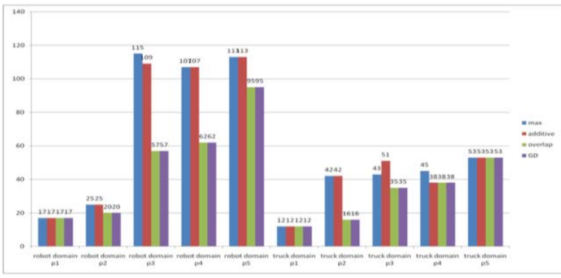

Figure 8: Comparison of search space sizes.

Figure 8 compares the search space sizes in terms of generated states. Our GD heuristics and the overlap heuristics expanded much smaller search space than both the max and additive heuristics. This result implies that our GD and overlap heuristics are much more informative than the max and additive heuristics.

Table 2 and 3 compare our GD heuristics with the overlap heuristics (Hoffmann and Nebel, 2001) in terms of subgoals generated and actions investigated during extraction process, respectively. We notice

Table 2: Comparison of generated subgoals.

Problems	Robot Domain					Block Domain				
Heuristics	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
Overlap	11	13	17	19	23	13	13	17	26	34
GD	3	3	3	3	3	6	6	8	12	16
Problems	Dinner Domain					Truck Domain				
Heuristics	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
Overlap	8	9	14	14	18	8	12	16	16	18
GD	5	4	5	4	5	3	3	3	3	3

that the overlap heuristics building a complete relaxed plan for each belief state consumed much more computational effort than our GD heuristics.

Table 3: Comparison of investigated actions.

Problems	Robot Domain					Block Domain				
Heuristics	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
Overlap	5	6	8	9	11	8	12	14	24	28
GD	3	3	3	3	3	4	6	8	12	16
Problems	Dinner Domain					Truck Domain				
Heuristics	p1	p2	p3	p4	p5	p1	p2	p3	p4	p5
Overlap	6	6	10	10	13	4	6	8	8	9
GD	5	4	5	4	5	3	3	3	3	3

7 CONCLUSIONS

In this paper, we proposed Merged Planning Graphs (MPGs), and GD heuristics for solving contingent planning problems. Through experiments in some problem domains, we showed that GD heuristics are more informative than the traditional max and additive heuristics. Moreover, in comparison to the overlap heuristics, GD heuristics require much less computational effort for extraction.

ACKNOWLEDGEMENTS

This work was supported by the GRRC program of Gyeonggi province.

REFERENCES

Bonet, B., Geffner, H., 2001. GPT: A Tool for Planning with Uncertainty and Partial Information, In *IJCAI'01, International Joint Conference on Artificial Intelligence*. MITPress.

Hoffmann, J., Brafman, R., 2005. Contingent Planning via Heuristic Forward Search with Implicit Belief States, In *ICAPS'05, International Conference on Automated Planning and Scheduling*. MITPress.

Bryce, D., Kambhampati, S., Smith, D., 2006. Planning Graph Heuristics for Belief Space Search, *Journal of Artificial Intelligence Research*, Vol. 26, pp.35-99.

Bonet, B., Geffner, H., 2005. mGPT: A Probabilistic Planner Based on Heuristic Search, *Journal of Artificial Intelligence Research*, Vol.24, pp.933-944.

Hoffmann, J., Nebel, B., 2001. The FF Planning System: Fast Plan Generation through Heuristic Search, *Journal of Artificial Intelligence Research*, Vol.14, pp.253-302.