CONTINGENT PLANNING AS BELIEF SPACE SEARCH

Incheol Kim and Hyunsik Kim

Department of Computer Science, Kyonggi University, San 94-6, Yiui-Dong, Youngtong-Gu, 443-760, Kyonggi, Korea

Keywords: Contingent planning, Partial initial condition, Nondeterministic actions, Belief states, Heuristic search.

Abstract: In this paper, we present a new heuristic search algorithm for solving contingent planning problems with the partial initial condition and nondeterministic actions. The algorithm efficiently searches through a cyclic AND-OR graph with dynamic updates of heuristic values, and generates a contingent plan that is guaranteed to achieve the goal despite of the uncertainty in the initial state and the uncertain effects of actions. Through several experiments, we demonstrate the efficiency of this algorithm.

1 INTRODUCTION

Classical AI planning has the assumption that the entire initial state is known at planning time, and each action should deterministically produce an exact one outcome. A more realistic assumption is that the planner only knows part of this information at planning time, the rest must be acquired at plan-execution time through observations, and each action may nondeterministically produce any of several outcomes. Several planning algorithms have been formulated to address these partially observable, nondeterministic planning problems (Ghallab, et al., 2004). Contingent planning algorithms construct a plan that includes both sensing actions and conditional execution of the actions in the plan. In general, contingent planning can be considered as search in a belief space to find a solution plan of which all possible different execution paths beginning from the initial belief state should end at one of goal belief states. In this paper, we present a new heuristic search algorithm for solving contingent planning problems with the partial initial condition and nondeterministic actions. The algorithm efficiently searches through a possibly cyclic AND-OR graph with dynamic updates of heuristic values, and generates a contingent plan that is guaranteed to achieve the goal despite of the uncertainty in the initial state and the uncertain effects of actions. Through several experiments, we demonstrate the efficiency of this algorithm.

2 CONTINGENT PROBLEMS

Planning with incomplete information can be formulated as a problem of search in belief space, where belief states can be sets of states.

In this paper, we use the corresponding meta-predicate formula, \((\text{unknown predicate term}_1, ..., \text{term}_n)\) to represent that we do not know whether the
The contingent planning problem with both partial initial condition and nondeterministic actions is \( P_{\text{pnd}} = (D, b_0, b_f) \) and the domain is \( D = (L, B, A) \), where \( L \) is the set of all known and unknown literals, \( B \) is the set of all belief states, and \( A = A_d \cup A_n \cup A_s \) is the set of actions, that includes deterministic, nondeterministic, and sensing actions, \( b_0, b_f \) are the respective initial and goal belief states. Figure 2 shows an example of contingent planning problem with partial initial condition.

A solution of the given contingent planning problem \( P_{\text{pnd}} \) can be represented as a policy \( \pi \), the set of \((\text{belief state, actions})\) pairs, as shown in Table 1.

### Table 1: A solution policy.

<table>
<thead>
<tr>
<th>Belief State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>(robot_in robot l_corridor) (object_in cup office1) (unknown door open between conf1 r_corridor) (door_open_between l_corridor office1) (unknown door open between lounge r_corridor) ...</td>
<td>move robot l_corridor office1</td>
</tr>
<tr>
<td>(robot_in robot office1) (object_in cup office1) (unknown door open between lounge r_corridor) ...</td>
<td>carry robot office1 l_corridor cup</td>
</tr>
<tr>
<td>(robot_in robot lounge) (grap_object robot cup) (unknown door open between lounge r_corridor) ...</td>
<td>sense door open lounge r_corridor</td>
</tr>
</tbody>
</table>

Figure 2: A planning problem.
We propose a heuristic search algorithm for solving contingent planning problems, called HSCP (Heuristic Search for Contingent Planning). It repeats multiple AND-OR-search trials beginning from the initial belief state until it obtains a complete solution (i.e., until solved becomes true), as shown in Figure 4.

```plaintext
1 HSCP(bs₀ : initial belief state)
2 Begin
3 valueTable.CLEAR();
4 solved = false;
5 While solved = false do
6   AND-OR-Search-Trial(bs₀);
7   return (policyTable.GET_POLICY());
8 End
```

Figure 4: The HSCP algorithm.

Figure 3 summarizes the AND-OR search trial of the HSCP algorithm. During AND-OR search trial, just one candidate of solution subgraphs keeps to be expanded until its every tip node successfully arrives at one of goal belief states. Therefore, the AND-OR search trial is like the heuristic depth-first search that traverses along every AND branch of the solution subgraph. The initial value of a belief state b is assumed to be its heuristic value, \( h(b) \). Whenever a node is selected to be expanded, the value of the belief state in the node is updated with the values of its successors. However, HSCP does not backpropagate value updates over the entire subgraph. So the AND-OR search trial proceeds fast. When it meets a dead end state that has no successors, it sets the value of the state to the infinite number \( \infty \). If the candidate subgraph is no longer expanded to be a complete solution, the search trial fails. And then a new search trial begins again with the updated value table. With the help of updated values of the states, the new search trial is able to expand another part of the AND-OR graph. As the number of search trials increases, the possibility to obtain a complete solution also increases. In this way, HSCP can find a suboptimal solution of the given contingent planning problem very efficiently.

Our HSCP algorithm can distinguish open cycles from closed ones. The former has the possibility to arrive one of goal belief states, but the latter does not. (A) and (B) in Figure 5 show an open cycle and a closed cycle, respectively.

Figure 5: Open and closed cycles.

4 RELATED WORK

There are several different algorithms for contingent planning problems: LAO∗, RTDP, CondFCP, and ND-FCP. LAO∗ (Hansen and Zilberstein, 2001) is an extended version of AO∗, the heuristic AND-OR search algorithm, so that it can find a solution subgraph containing open cycles. This algorithm obtains an optimal solution with just one search trial. However, during the search trial, it evaluates and expands multiple candidates of solution subgraphs simultaneously, and backpropagates value updates over the entire solution subgraph whenever it gets a new heuristic value of a tip node. Moreover, whenever it meets an open cycle, it continues tracing and value updating along the cycle until every state nodes in the cycle gets a converged value.

RTDP (Bonet and Geffner, 2003) is a well-known dynamic programming algorithm for solving nondeterministic contingent planning problems. It has two key advantages comparing with other DP algorithms: first, it obtains an optimal policy without computing the whole space, second, it has a good anytime behavior. However, RTDP usually requires a lot of search trials and value updates, so its convergence is slow. CondFCP (Kuter, et al, 2007) is an extended version of the classical forward-chaining algorithm for solving planning problems.

Figure 6 shows the part of search space expanded by HSCP for solving the example problem of Figure 2.
with partial initial condition. On the other hand, ND-FCP (Kuter and Nau, 2004) is an extended one for solving nondeterministic planning problems. Both algorithms do just one search trial and expand only one candidate of solution subgraph during the search trial. They do not try to update heuristic values with new information. Therefore, they do not guarantee to find a solution and cannot be used for solving contingent planning problems with both partial observations and nondeterministic actions.

5 EXPERIMENTS

We implemented the HSCP algorithm, and compared it with LAO* and RTDP on two partially-observable, nondeterministic planning domains that are well-known from previous experimental studies: Robot Navigation and Blocks World. Three random problems were generated from each domain for experiments. We compared three different search algorithms in terms of the number of search trials, the number of value updates, and the number of generated states. Figure 7 ~ Figure 9 show the experimental results. While RTDP has performed lots of search trials to get the optimal values of states, our HSCP and LAO* have done just one or two trials for each problem. Out of three search algorithms, RTDP has tried value updates the most, but HSCP has done the least. Considering the number of generated states, we find out HSCP has explored much smaller search space than the other two algorithms.

6 CONCLUSIONS

We have presented a new heuristic search algorithm for solving contingent planning problems with the partial initial condition and nondeterministic actions. Through several experiments, we have evaluated the efficiency of this algorithm.

ACKNOWLEDGEMENTS

Industrial Strategic Technology Development Program (10032108), funded by the Ministry of Knowledge Economy(MKE, Korea).

REFERENCES


