Solving the Kitchen Benchmark using HCplan

Ahmed Nouman, Esra Erdem and Volkan Patoglu
Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey
Email: {ahmednouman,esraerdem,vpatoglu}@sabanciuniv.edu

I. Problem Description

We consider a dynamic robotic domain that consists of a bi-manual mobile robot in a kitchen environment. The robot must prepare a meal for a couple by cleaning two glasses, cooking two cabbages (located in a bowl), and setting up the table. Objects can be cleaned by placing them in the dishwasher and cooked by placing them in the microwave. A cabbage must be cleaned before it can be cooked. All objects on the table must be clean. Finally, the robot can encounter object obstructions during manipulation, in which case it should take necessary actions to manipulate by picking the objects that obstruct the motion plan and placing them back to their original locations after manipulation to keep the kitchen tidy. Figure 1 depicts a snapshot of the kitchen benchmark.

Fig. 1. Kitchen Table Setting Scenario

This benchmark is of interest as it involves several real-life challenges that appear in many target applications. In particular, the robot needs to perform high-level planning to decide for the order of actuation actions, while it also needs to perform geometric reasoning to check the feasibility of these actions. Kitchen domain benchmark is particularly challenging as due to object obstructions, the resulting plans may require rearrangement of objects that may be non-monotonic in nature.

In addition to the challenges listed above, we also consider uncertainty caused by incomplete knowledge about the domain during planning. Note that uncertainties are commonplace in social robot scenarios, especially in the ones that involve human interactions. Given that offline planning takes place before execution, for instance, the robot might not know location of an object or whether it is clean or not during planning, as it may not have access to it. We consider such uncertainties during planning, and compute conditional plans offline considering all possible contingencies so that the robot does not fail during execution.

II. Proposed Method

Conditional planning is concerned with planning the actuation actions for robots to achieve their goals in the presence of incomplete information and sensing actions [4, 5]. The complexity of conditional planning is Σ_2^P-complete, even for polynomially bounded plans with limited number of nondeterministic actions [1].

HCplan [3] is an offline compilation-based hybrid conditional planner, which extends hybrid sequential planning with nondeterministic sensing actions and utilizes this extension to compute branches of a conditional plan in parallel. It is a hybrid conditional planner as it not only computes high-level plans of actuation and sensing actions, but also performs geometric reasoning to check for the feasibility of both types of actions. We propose to use HCplan to solve the kitchen benchmark described above, by computing a hybrid conditional plan and verifying its feasibility in real-life scenarios through dynamic simulations implemented in OpenRAVE [2].

In order to solve kitchen table set-up scenario, three actuation actions are considered in our domain: $goto$, $pickup$ and $placeon$. Given that the environment is not completely observable during planning, two types of possible sources of uncertainties are considered. First, it is assumed that the locations of kitchenware are uncertain and might not be known by the robot during the planning phase. These locations can be reliably identified only if the robot actively searches for these objects when it needs to use them. Second, the cleanliness/dirtiness of the objects may not be known before picking them to check. Along these lines, two sensing actions are considered in our domain: $checkLoc$ and $checkisClean$. The first sensing action $checkLoc$ is utilized to resolve the uncertainty about locations of kitchenware and the $checkisClean$ sensing action is introduced to determine cleanliness of kitchenware.

TABLE I
Domain Description of the Kitchen Benchmark

Fluents	
rob_At	represents robot location
obj_At	represents object location
is_Clean	represents if an object is clean or not
is_Cooked	represents if a cabbage is cooked or not
$tableSet$	represents if table is set or not
Actuation Actions	
$goto$	robot navigation
$pickup$	robot picks up an object with manipulator
$placeon$	robot places an object at desired location
Sensing Actions	
$check\_loc$	checks location of an object
$check\_is\_clean$	checks if an object is clean or not

In our case studies, the robot implements these sensing actions by means of perception. Table I summarizes our domain description for the kitchen benchmark.

III. A Case Study

We consider an instance of the kitchen benchmark where initially both cabbages are raw and located on the extra table. The cleanliness of one cabbage is unknown, while the other one is known to be not clean. One of the glasses is located on the extra table and is dirty, whereas the location of the second glass is unknown, while it is known to be clean. Finally, the robot is initially located next to the table. The goal is to clean up, cook both cabbages, and setup the table. This problem is described to HCplan as follows:

```
% initial state
0: robAt=table,
isCooked(cabbage_1)=no,
isCooked(cabbage_2)=no,
objAt(cabbage_1)=extra_table,
objAt(cabbage_2)=extra_table,
isClean(cabbage_1)=no,
isClean(cabbage_2)=unknown,
... ;
% goal state
maxstep: tableSet.
```

A solution to this benchmark problem is a conditional tree computed by HCplan, and partially presented in Figure 3. The conditional plan has 8 leaf nodes, stating that the planner computes 8 possible ways to reach the goal under all possible contingencies due to uncertainty. The total time to compute the conditional plan is 5.47 seconds using 8 CPU cores of 2.4GHz Intel E5-2665. The plan has 107 nodes in total, 5 of which are sensing actions. The depth of longest branch is 24 (with 22 actuation actions and 2 sensing actions) and the length of the shortest branch is 18 (with 16 actuation actions and 2 sensing actions). Figure 2 presents snapshots from the dynamic simulation of one of the branches of this conditional plan given in Figure 3.

Acknowledgments

We thank anonymous reviewers for their useful feedback.

References

[1] C. Baral, V. Kreinovich, and R. Trejo. Computational complexity of planning and approximate planning in presence of incompleteness. In *Proc. IJCAI*, 1999.

[2] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, CMU, 2010.

[3] A. Nouman, I. F. Yalciner, E. Erdem, and V. Patoglu. Experimental evaluation of hybrid conditional planning for service robotics. In *Proc. ISER*, 2016.

[4] M. A. Peot and D. E. Smith. Conditional nonlinear planning. In *Proc. AIPS*, 1992.

[5] Louise Pryor and Gregg Collins. Planning for contingencies: A decision-based approach. *JAIR*, 4:87–339, 1996.

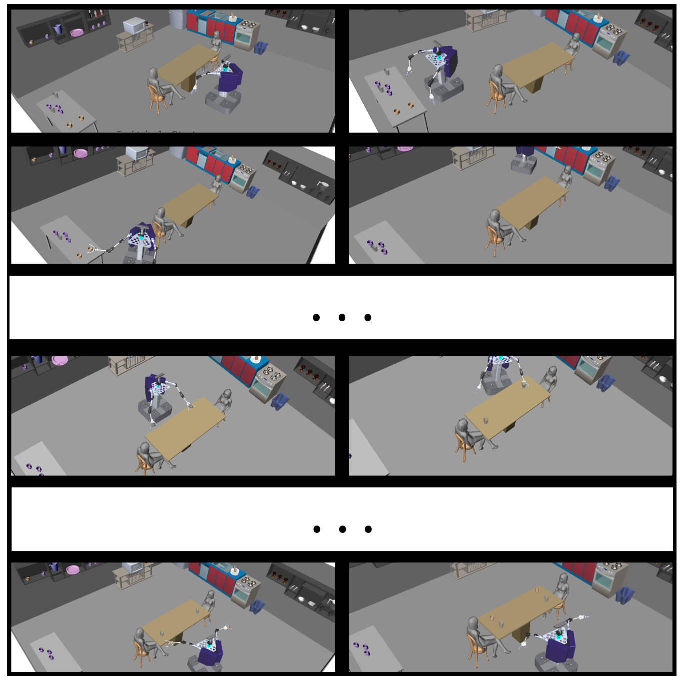

Fig. 2. Snapshots showing the dynamic simulation of one branch of the conditional plan (shown in Figure 3) of the kitchen benchmark: (a) Initial state (b) 1: goto(extraTable) (c) 2: pickup(glass_1,l_hand) pickup(cabbage_1,r_hand) (d) 3: goto(dishwasher) (e) 32: goto(table) (f) 33: placeon(table,l_hand) placeon(table,r_hand) (g) 41: placeon(table,l_hand) placeon(table,r_hand) (h) Goal State

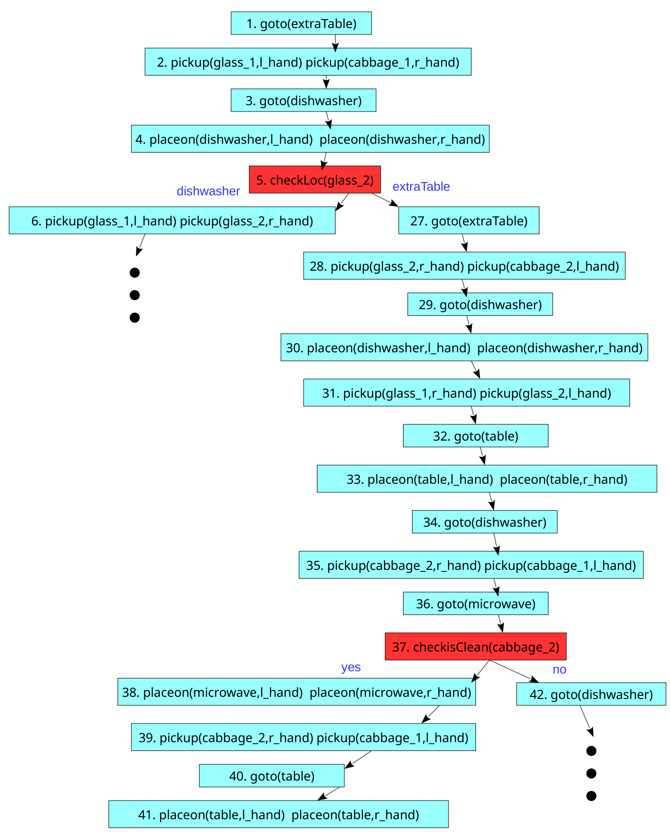

Fig. 3. (Partial) Conditional Plan for Kitchen Benchmark