Hybrid Conditional Planning for Robotic Applications (Extended Abstract)

Ahmed Nouman\textsuperscript{1}, Volkan Patoglu\textsuperscript{2}, Esra Erdem\textsuperscript{2}

\textsuperscript{1}Department of Mechanical Engineering, University of Engineering and Technology, Taxila, Pakistan
\textsuperscript{2}Faculty of Engineering and Natural Sciences, Sabancı University, İstanbul, Turkey
ahmed.nouman@uettaxila.edu.pk, \{vpatoglu, esra.erdem\}@sabanciuniv.edu

Hybrid classical planning, where classical task planning is integrated with low-level feasibility checks (e.g., motion planning that utilizes collision checks), has received attention in AI and Robotics communities (Erdem et al. 2011; Hertle et al. 2012; Plaku 2012; Kaelbling and Lozano-Pérez 2013; Srivastava et al. 2014; Lagriffoul et al. 2014; Hadfield-Menell et al. 2015; Erdem, Patoglu, and Schüller 2016), due to the cognitive skills required by autonomous robots in the real-world. Since these studies rely on classical planning methods, they assume complete knowledge about the initial states, and full observability of the environment, while computing a hybrid classical plan (i.e., a sequence of actuation actions) offline. During the execution of these plans, discrepancies might occur between the expected states and the observed states, due to contingencies that were not or could not be considered during offline planning. For instance, a robot may consider that initially all utensils in a cabinet are clean, and compute a hybrid classical plan to set up a table accordingly. While executing this plan, the robot may detect (e.g., by its sensors) that the plate that it picks from the cabinet is not clean. To cope with such contingencies, the robots are usually equipped with a plan execution monitoring algorithm. According to such an algorithm, when a discrepancy is detected, the robot tries to recover from the discrepancy, generally by replanning from that point on.

As an alternative approach to computing offline hybrid classical plans and then dealing with discrepancies/surprises online during plan execution by monitoring and replanning, we propose a parallel offline hybrid method, called HCPLAN. This method suggests extending hybrid planning beyond classical planning, by inheriting advantages of conditional planning to deal with contingencies due to incomplete knowledge and partial observability. In conditional planning, in addition to actuation actions with deterministic outcomes, sensing actions with nondeterministic outcomes are considered as part of planning in order to gather the relevant missing knowledge. Every possible outcome of sensing actions leads to a possibly different conditional plan. Therefore, a conditional plan looks like a tree of actions, where the branching occurs at vertices that characterize sensing actions, and the other vertices denote actuation actions. Each branch of such a tree, from the root to a leaf, essentially represents a possible execution of actuation actions and sensing actions to reach a goal state from the initial state. HCPLAN integrates this advantageous aspect of conditional planning while computing offline plans: planning for nondeterministic sensing actions to gather missing knowledge when needed, while planning for deterministic actuation actions. Moreover, HCPLAN integrates feasibility checks not only for executability of actuation actions, but also for executability of sensing actions.

HCPLAN is a parallel algorithm: it computes a hybrid conditional plan by intelligently orchestrating the computation of its branches in parallel. According to HCPLAN, computation of every branch starting from a vertex that characterizes a sensing action is viewed as a computation task with a priority (e.g., defined relative to the depth of that vertex), and the batches of computation tasks are solved in parallel with respect to their priorities so as to consume the computational resources more effectively. Furthermore, HCPLAN avoids re-computation of the same task that may appear at different parts of the tree.

Each computation task handled by HCPLAN takes as input an incomplete initial state, and returns a hybrid sequential plan (i.e., a sequence of deterministic actuation actions and nondeterministic sensing actions) that describes a branch of the tree. A hybrid sequential plan is not a hybrid classical plan, since it may involve nondeterministic sensing actions as well. Therefore, solving a computation task (i.e., the computation of each branch of a hybrid conditional plan) also requires cognitive abilities beyond hybrid classical planning. For that, HCPLAN adapts some advantages of causality-based non-monotonic logics (Turner 2008; Gelfond and Lifschitz 1998) for a novel solution to deal with these challenges. In particular, HCPLAN utilizes defaults to express assumptions (e.g., the location of an object remains to be the same unless changed by an actuation action), exogeneity of actions to express that sensing actions may occur at any time when possible, and nondeterministic causal laws to choose an outcome of a sensing action included in the computation of a branch. The defaults are useful for formalizing assumptions in that, when an exception occurs contrary to the assumptions, it does not lead to an inconsistency. This is an important aspect of defaults, providing a solution to the famous frame problem. Exogeneity of actions is useful in that it is not required in advance to specify the order of nondeterministic sensing actions while solving
a task. HCPPLAN uses the nonmonotonic reasoning system CCA\textsubscript{L}C (McCain and Turner 1997) to compute each branch of a hybrid conditional plan.

We show a real-world application of HCPPLAN where a mobile bi-manual service robot sets up a table for lunch. While computing a hybrid conditional plan, the robot has to consider various contingencies that are not known in advance. Furthermore, while planning for its actions, the robot has to consider different types of feasibility checks (e.g., reachability, graspability, collisions), as well as commonsense knowledge. Once a hybrid conditional plan is computed by HCPPLAN, we also illustrate possible executions of this plan.

To evaluate the strengths and weaknesses of HCPPLAN by means of experimental evaluations, we construct a benchmark suite over the kitchen table setup domain. We evaluate HCPPLAN from the perspectives of computation time and plan quality, compare it with the closely related planners, and with an execution monitoring algorithm that utilizes hybrid planning and guided re-planning.

We refer the reader to our journal paper (Nouman, Patoglu, and Erdem 2021) for further information.

References


