How explainable plans can make planning faster*

Antoine Gréa and **Laetitia Matignon** and **Samir Aknine**

Abstract

In recent years the ubiquity of artificial intelligence raised concerns among the uninitiated. The misunderstanding is further increased since most advances do not have explainable results. For automated planning, the research often targets speed, quality, or expressivity. Most existing solutions focus on one criteria while not addressing the others. However, human-related applications require a complex combination of all those criteria at different levels. We present a new method to compromise on these aspects while staying explainable. We aim to leave the range of potential applications as wide as possible but our main targets are human intent recognition and assistive robotics. We propose the HEART planner, a real-time decompositional planner based on a hierarchical version of Partial Order Causal Link (POCL). It cyclically explores the plan space while making sure that intermediary high level plans are valid and will return them as approximate solutions when interrupted. These plans are proven to be a guarantee of solvability. This paper aims to evaluate that process and its results compared to classical approaches in terms of efficiency and quality.

Introduction

Since the early days of automated planning, a wide variety of approaches have been considered to solve diverse types of problems. They all range in expressivity, speed, and reliability but often aim to excel in one of these domains. This leads to a polarization of the solutions toward more specialized methods to tackle each problem. All of these approaches have been compared and discussed extensively in the books of Ghallab et al. [2004; 2016].

Partially ordered approaches are popular for their least commitment aspect, flexibility and ability to modify plans using refinement operations [Weld 1994]. These approaches are often used in applications in robotics and multi-agent planning [Lemai and Ingrand 2004; Dvorak *et al.* 2014]. One of the most flexible partially ordered approaches is called *Partial Order Causal Link planning (POCL)* [Young and Moore 1994]. It works by refining partial plans consisting of steps and causal links into a solution by solving all flaws compromising the validity of the plan.

Another approach is *Hierarchical Task Networks (HTN)* [Sacerdoti 1974] that is meant to tackle the problem using composite actions in order to define hierarchical tasks within the plan. Hierarchical domains are often considered easier to conceive and maintain by experts mainly because they seem closer to the way we think about these problems [Sacerdoti 1975].

In our work, we aim combining HTN planning and POCL planning in such a way as to generate intermediary high level plans during the planning process. Combining these two approaches is not new [Young and Moore 1994; Kambhampati *et al.* 1998; Biundo and Schattenberg 2001]. Our work is based on *Hierarchical Partial Order Planning (HiPOP)* by Bechon *et al.* [2014]. The idea is to expand the classical POCL algorithm with new flaws in order to make it compatible with HTN problems and allowing the production of abstract plans. To do so, we present an upgraded planning framework that aims to simplify and factorize all notions to their minimal forms. We also propose some domain compilation techniques to reduce the work of the expert conceiving the domain.

In all these works, only the final solution to the input problem is considered. That is a good approach to classical planning except when no solutions can be found (or when none exists). Our work focuses on the case when the solution could not be found in time or when high level explanations are preferable to the complete implementation detail of the plan. This is done by focusing the planning effort toward finding intermediary abstract plans along the path to the complete solution.

In the rest of the paper, we detail how the HiErarchical Abstraction for Real-Time (HEART) planner creates abstract intermediary plans that can be used for various applications. First, we discuss the motivations and related works to detail the choices behind our design process. Then we present the way we modeled our own planning framework fitting our needs and then we explain our method and prove its properties to finally discuss the experimental results.

1 Motivations and Potential Applications

Several reasons can cause a problem to be unsolvable. The most obvious case is that no solution exists that meets the requirements of the problem. This has already been addressed by Göbelbecker *et al.* [2010] where “excuses” are being investigated as potential explanations for when a problem has no solution.

*Univ Lyon, Université Lyon 1, CNRS, LIRIS, UMR5205, F-69621, LYON, France (first.lastname@liris.cnrs.fr)