Carlos Conde
Domain-Independent Planner
Artificial Intelligence Nanodegree
September 21, 2017

History of Planning in Artificial Intelligence

AI Planning (a.k.a Automated Planning and Scheduling on a more generalized way) is a branch of Artificial Intelligence which main focus lays on finding Action sequences to reach a Goal (often form by a set of subgoals) starting from an known Initial State.

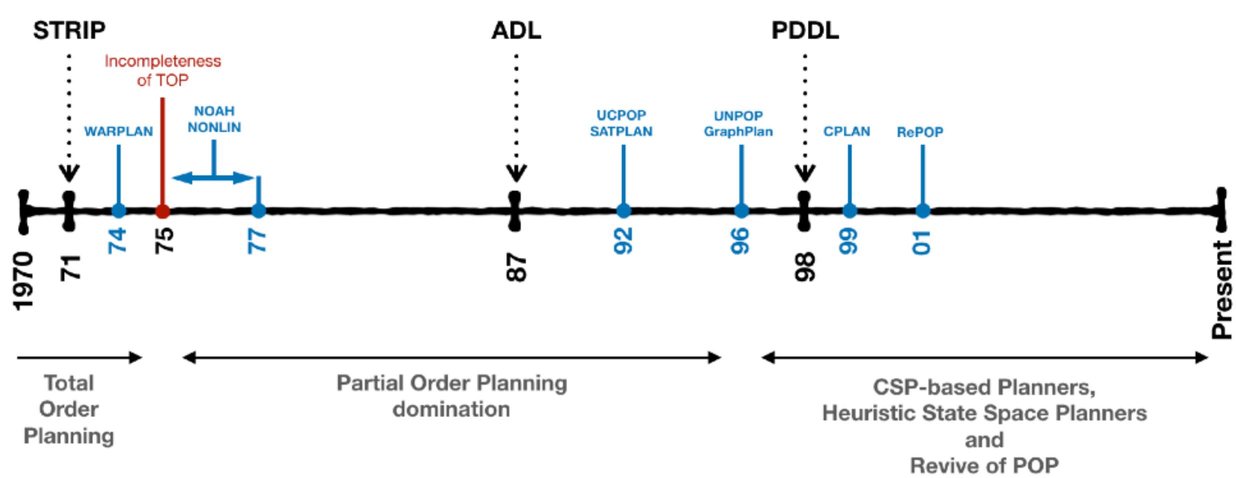
On a very high level abstraction, the main challenges reside on dealing with the representation of a real infinite-state world into a bound finite-state world and on being able to efficiently traverse this reduced world with finite states in a reasonable time making use of a reasonable amount of resources (computational).

This paper presents some of the main achievements/developments in the history of AI Planning that have shaped the field of Planning to the state it is at now a days.

Introduction

With the aim of presenting the aforementioned developments, the research will be divided in 2 main sections: the language used to reduce and represent the world into a usable structure a machine/agent can traverse and the algorithms and techniques used to traverse the problem space.

Also, to support the developments that will be presented below, a timeline is provided with the main dates in the history of Planning from the 70s to present date.



Representation of Planning Problems

In Artificial Intelligence the language that is commonly used to represent the world states is known as **action language**, a language specifying state transition systems that is used to create models of the effects of actions in the states of the world.

The first important development in the field of Planning was the creation of the **Stanford Research Institute Problem Solver (STRIPS)** the “first” intelligent robot the *SRI International* created on 1971. The same name was later used to refer to the formal language of the inputs of the planner and has been the baseline for other action languages used for automated planning since.

Consisting of an initial state, a set of goals and a set of actions with preconditions (conditions to be met to be able to apply the action) and postconditions (what is established once the action has been applied), the language is a simplification of First-Order Logic formal-system collection and is very restricted, what allows inference to be done efficiently.

On 1987, *Edwin Pednault* proposed the **Action Description Language** (**ADL**), considered to be an advancement/relaxation of STRIPS, with the main differences been the consideration of an Open World (everything not specified in the conditions is considered unknown instead of False) and the usage of negative literals and disjunctions.

The next important development in this respect was made by *Drew McDermott* on 1998, by developing the **Planning Domain Definition Language (PDDL)** based on STRIPS, ADL and others with the intention of providing a standardized representational language to enable the first International Planning Competition (IPC).

Techniques and Algorithms

In the early 70s, most of the planners were using a totally ordered action sequence to find a plan, looking for separated plans for each subgoals and then stringing together this plans in some order. This way of approaching planning problems was known as **Total Order Planning (TOP)** or **Linear Planning**. This approach was soon discovered to be incomplete by *Sacerdoti* on 1975, as it appeared to not be able to solve some simple problems (like the *Sussman Anomaly*).

As a solution to the limitations of TOP, goal-regression planning was used, a technique in which the steps of a TOP plan are reordered to avoid conflicts between goals. This approach that includes the detection of conflicts and the protection of achieved conditions from interference is known as **Partial Order Planning (POP)** and was one of the most important developments in the field, an approach that was intensively used for the following 20 years, pioneered but the **NOAH** (by *Sacerdoti*) and **NONLIN** (by *Tate*) planners.

On the 90s, and after a general loss of interest in the planning field that run through the first half of this decade, the field was revived thanks to new and faster planners that made POP fall out of favor. These new planners were dominated by 2 main themes: heuristic-based state-search planners (like **UNPOP** or **Heuristic Search Planner, HSP**) and Constraint Satisfaction or CSP-based planners (like **GraphPlan** or **SATPLAN**). Both of the approaches run away from the efforts of the last years that only focused on scaling up POP and benefit from the increased computational power of more recent computers.

Nevertheless, and taking advantage of the advances in plan synthesis made on both state-based and CSP-based planners, **Reviving Partial Order Planning** (**RePOP**) proposes improvements to POP approach using distance based heuristics, reachability analysis and disjunctive constraint processing techniques.

Summary

There has been lots of discussion during the last half decade regarding both, the representation of planning problems and the approaches and techniques to traverse the state-space/plan-space efficiently looking for complete and optimal algorithms that can provide solutions to complex planning problems in a independent way (Domain-Independent planners).

What is clear from all the information available and also the conclusion of many articles and blog posts is that there is no perfect generic planner that can solve different planning problems, and that different approaches work best on different types of environments.

This is why many of the real world planning problems that companies and organizations need to solve now a days use a mix of the techniques presented in this papers and many others with domain-dependent information that improves the efficiency of those planners.