Udacity Artificial Intelligence Nanodegree

Research Paper: Key Developments in AI Planning and Search

> Author: Ke Zhang
>
> Submission Date: 2017-07-15 (Revision 1)

Introduction

This research paper briefly describes the major milestones in the automated planning and scheduling (or simply AI Planning).

Linear Planner

A simple linear planner generates plans as a totally ordered sequence of steps [11]. We'll take a look at two well-known examples of linear planners: STRIPS (a non-hierarchical linear planner) and GPS (a hierarchical linear planner).

Stanford Research Institute Problem Solver (STRIPS)

The earliest automated planning systems were linear. STRIPS was one of them. It was developed by Richard Fikes and Nils Nilsson in 1971 [3] and was primarily used in robot research.

The first step in STRIPS is always to define the initial world state by providing objects, actions, preconditions, and effects. The STRIPS planner works by identifying a series of operators in a space of models and executing the planning domain and problem to find the goal [4]. The STRIPS solver is non-hierarchical and linear. It assumes that there is a sequence of actions to attain the goal in strict ordering and makes no distinction between important and less important parts of the actions.

In the AI Planning literature, STRIPS is associated with the STRIPS First-Order Logic language which was popular to describe and defining complex planning problems. The classical Planning Domain Definition Language (PDDL) was inspired by the STRIPS language and had a very similar representation [5].

General Problem Solver (GPS)

GPS was the first hierarchical linear planner and one of the earliest AI programs, created by Herbert A. Simon, J.C. Shaw, and Allen Newell in 1959. It was intended to work as a universal problem solver machine, using the same reasoning mechanism to solve all kinds of problems [6].

In order to solve a problem, the GPS solver generates heuristics by means-ends analysis in a space of user-defined objects and operations that could be done on the objects. It tries both the upward and the downward solution property and creates subgoals to get closer to the goal.

In the AI history, the ideas of GPS evolved in the 80s into the Soar cognitive architecture for artificial intelligence which helps to develop the fixed computational building blocks necessary for general intelligent agents to realize a range of tasks such as decision making, planning, and problem solving [7].

Non-Linear Planner

A nonlinear planner (a.k.a. partial-order planner) builds up a plan as a set of steps with some temporal constraints. If the resulting plan is not linear, either linearization for a single agent, or execution by multiple agents can be used [11]. Constraints and ordering decisions are only made when necessary to maximize the parallel processing. Some details about of a nonlinear planner is demonstrated by the next distributed multi-agent planner.

Multi-Agent Planner

The notion Multi-Agent can be described as a network of distinct entities. A Multi-Agent planning is an approach to solve complex planning problems that are difficult or impossible for an individual agent to solve by exploiting large scale computation and spatial distribution of computing resources[1].

The key characteristic of multi-agent planning is the combination of planning and coordination. A Multi-Agent coordinator ensures that the individual agents execute the actions in the correct order and work efficiently without any conflicts. Furthermore it provides the agents with distributed information and guarantees that all dependencies of actions and global constraints are held as expected.

In 2012, Nissim & Brafman used the distributed planning through a Multi-Agent A* algorithm maintaining lists of visited and unvisited states for all agents. Each of the agents could practice their individual plan and local heuristics to decide which unvisited state should expand in the extraction phase. The coordination among the agents were tackled by broadcasting shared status messages between the agents in order to align the execution order and to search at interaction points where the other agents can follow. The process ends with each agent yielding their individual solution. Although it's a distributed planner solution where the agents are loosely coupled, due to the flexibility of the algorithm and its domain-independent model, it could also used to tackle planning tasks of any type.

Further Thoughts

Today, the AI planning research still encounters issues when dealing with real-world problems with complex conditions, probabilistic uncertainty, partial observation or continuous changes [2]. A distributed multi-agent approach seems to be predestined to avoid the time and resource limitations on a single workstation. Actually many recent papers relating to multi-agent planning have tried to distill findings from other latest AI techniques like the distributed neural network [8], reinforcement learning [9] or sampling-based algorithms [10]. We believe that in the near future there will be a truly breakthrough in multi-agent AI planning and that will well be the success.

References

- [1] Wikipedia: Multi-Agent Planning
- [2] [AI: Complexity and Limits of Planning] (https://www.cs.duke.edu/brd/Teaching/Previous/AI/Lectures/Planning/planning.html#Complexity and Limits of Planning)
- [3] Fikes and Nilsson: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, 1971
- [4] Becker: Artificial Intelligence Planning with STRIPS, A Gentle Introduction
- [5] Wikipedia: Planning domain Definition Language
- [6] Wikipedia: General Problem Solver
- [7] Wikipedia: Soar cognitive Architecture
- [8] Zhang and Li: A type of biased consensus-based distributed neural network for path planning, 2017
- [9] Cruz and Yu: Path planning of multi-agent systems in unknown environment with neural kernel smoothing and reinforcement learning, 2016
- [10] Quindlen and How: Machine Learning for Efficient Sampling-Based Algorithms in Robust Multi-Agent Planning Under Uncertainty, 2017
- [11] Doyle: AI Qual Summary - Planning
- [12] Nissim and Brafman: Multi-Agent A* for Parallel and Distributed Systems