Development in Artificial Intelligent Planning

Planning is a key ability for intelligent systems, increasing their autonomy and flexibility through the construction of sequences of actions to achieve their goals. It has been an area of research in artificial intelligence for over three decades. Planning techniques have been applied in a variety of tasks including robotics, process planning, web-based information gathering, autonomous agents and spacecraft mission control (Jim Blythe, 2017). Following three are major development in AI planning field.

STRIPS

The Stanford Research Institute Problem Solver (STRIPS) is an automated planning technique that works by executing a domain and problem to find a goal, developed by Richard Fikes and Nils Nilsson in 1971. (Richard E. Fikes, 1971). STRIPS the first major planning system. With STRIPS, you first describe the world than problem set and goals. Then we use various search techniques to solve for solution. A common language for writing STRIPS domain and problem sets is the Planning Domain Definition Language (PDDL). STRIPS can used to solve many problem like Rubik's cube, playing games like StarCraft, shortest distance calculation, robotics movement just to name few. Almost all planning systems since then have used one variant or another of the STRIPS language. (Russell, 2009)

Temporal planning

When multiple things can be happening at a time, it is necessary to model the duration and concurrency of actions and events. Due to the possibility of multiple concurrent actions, transition systems cannot be modelled as easily as in classical planning as a (finite) set of states together. (Rintanen, 2017). Various modelling languages are developed for transaction planning like PDDL 2.1, timed petris and time Automata. Search techniques for classical planning like explicit state-space can’t be used for temporal planning as number of states is high and limited by the necessity to do search one state at a time. Hence we used constraint- based search like SAT, SMT, Mixed Integer Linear Programming, Constraint Programming.

Markov decision process

Markov Decision Processes (MDPs) are widely popular in Artificial Intelligence for modeling sequential decision-making scenarios with probabilistic dynamics (Mausam, 2012). Often an agent must reason about an ongoing process or it does not know how many actions it will be required to do. At each stage, the agent decides which action to perform; the resulting state depends on both the previous state and the action performed. For ongoing processes, you do not want to consider only the utility at the end, because the agent may never get to the end. Instead, an agent can receive a sequence of rewards. These rewards incorporate the action costs in addition to any prizes or penalties that may be awarded. Negative rewards are called punishments (David Poole, 2010). Applications include robot navigation problems, machine maintenance, and planning under uncertainty in general (Leslie Pack Kaelbling; Michael L Littman & Anthony R Cassandra, 1998).

Works Cited

David Poole, A. M. (2010). *Artificial intelligence foundations of computational agents.* Cambridge University Press.

Jim Blythe, J.-L. A. (2017, 3 19). *CS 541: Artificial Intelligence Planning*. Retrieved from http://www.isi.edu/~blythe/cs541/

Leslie Pack Kaelbling; Michael L Littman & Anthony R Cassandra. (1998). Planning and acting in partially observable. *Artificial Intelligence*, 99-124.

Mausam. (2012). Planning with Markov Decision Processes: An AI Perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1-210.

Richard E. Fikes, N. J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *2nd IJCAI, Imperial College* (pp. 1-3). London: Stanford University.

Rintanen, J. (2017, 3 20). Retrieved from A brief overview of temporal planning in AI: https://users.ics.aalto.fi/rintanen/temporalplanning.html

Russell, P. N. (2009). *Artificial Intelligence: A Modern Approach.* Prentice Hall.