Planning review
Eduardo Carrasco Jr.

A.I. is the study of making rational actions within an achievable and a reasonable amount of time. Constraints and planning were for a long a challenge to tackle. Over the last few decades planning has been a topic which has made huge impact in computer science. One of the most influential automated planner is STRIPS (Stanford Research Institute Problem Solver) developed by Richard Fikes and Nils Nilsson.[1] The STRIPS system proposed for formally structuring and solving a give planning problem. This is important because people were just formulating problems without structuring in order for the problem to be solved. This is what gave STRIPS a huge advantage.

Examples of the possibility that STRIPS can do varied. For example, of some task this system could do was:

- Moving objects
- Traveling to another location
- Climbing up and down from rigid objects
- Turning light switches on and off
- Opening and closing doors

STRIPS gave birth to a sub disciple of programming called *linear programming*. This was a formal way of defining instances consisting of initial states, and specification of the goal states and actions. Then each actions have pre-conditions and post-conditions.

Another huge innovation that came out of STRIPS was the STRIPS programming language that again, using formal planning language that took in inputs for the planner. This allowed for problem that were consider NP-incomplete problems into NP complete problems. [2] The complete planner must allow for interleaving of actions from different sub-plans within a single sequence.

One solution to this problem was proposed by Waldinger, by goal regression constructs by a totally ordered plan and then constructively modifies to satisfy all sub-goals. [3] This approach was implemented in planning WARPLAN. This planner was first written in a logic programming language Prolog.

SatPlan [4] tries to cast the problem of planning as one of satisfiability. It converts the planning problem into a Boolean satisfying problem. A satisfying problem is a Boolean formula in a conjunctive normal form, find an assignment of truth values to literals that make it true. This is intuitively what SatPlan does, it constructs a proposal sentence that includes: initial state, set of actions and a goal. Then a SAT solver is called to create a model based on the sentence. It extracts variables that represent actions and create a plan from them if a model can be constructed.

In recent years with the development of deep learning, a new network called a QMDP-network. The QMDP-net combines the strengths of model-free learning and model-based planning. It is a recurrent policy network, but it represents a policy by connecting a model with a planning algorithm that solves the model, thus embedding the solution structure of planning in a network learning architecture. The QMDP-net is fully differentiable and allows end-to-end training. We train a QMDP-net in a set of different environments so that it can generalize over new ones and "transfer" to larger environments as well. [5]

Planning review
Eduardo Carrasco Jr.

REFERECNES

[1] Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving" (PDF). *Artificial Intelligence*. **2** (3–4): 189–208. doi:10.1016/0004-3702(71)90010-5.

[2] Bylander, Tom. 1994. “The Computational Complexity of Propositional Strips Planning.” Artificial Intelliegence. 69 (1-2)

[3] Waldringer, Richard. 1981. “Achieving Several Goals Simultaneously.” Readings in Artificial Intelligence. Los Altos, CA: Morgan Kaufmann, 250-71.

[4] Kautz, Henry A, Bart Selman, and others. 1992 “Planning as Satisfiability.” In ECAI, 92:359-63. Citseer.

[5] Karkus, Peter, David Hsu, and Wee Sun Lee. "QMDP-Net: Deep Learning for Planning under Partial Observability."