AIND Planning Search Research Review

This review summarizes three key developments in the field of AI planning and search. By taking these key technic, we are able to convert the problem into propositional or relational representations of states and actions and search with domain independent heuristic to the goal that we want to achieved.

PDDL

The first key development is that we learned how to break down the problem into PDDL. The PDDL is a language representation problem which a state of the world is represented by a collection of variable. It primarily describes a system using a set of preconditions and post-conditions[1]. In this project, we break down the air cargo planning problem into PDDL. Taking the first problem as example:
Proposition: There are two cargos{C1,C2}, two plane{P1,P2} and two airport{JFK,SFO}. C1,P1 at the SFO and C2, P2 at the JFK in the very beginning. The goal is to bring the C1 to the JFK and C2 to SFO. There are three action we can take, load/unload the cargo to/from the plane and fly the plane to some airport. Each of the action have its own precondition and effect. Such as Fly action must the plane at the “from" airport and effect the result that plane is not in the “from” airport, but it is actually in the “to” airport, etc. Now the we can convert the proposition into following PDDL:

Problem 1: action schema:

```
Action(Load(c, p, a),
        PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
        EFFECT: ¬ At(c, a) ∧ In(c, p))
Action(Unload(c, p, a),
        PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
        EFFECT: At(c, a) ∧ ¬ In(c, p))
Action(Fly(p, from, to),
        PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
        EFFECT: ¬ At(p, from) ∧ At(p, to))
```

Problem 1: init state and goal

```
Init(At(C1, SFO) ∧ At(C2, JFK)
        ∧ At(P1, SFO) ∧ At(P2, JFK)
        ∧ Cargo(C1) ∧ Cargo(C2)
        ∧ Plane(P1) ∧ Plane(P2)
        ∧ Airport(JFK) ∧ Airport(SFO))
Goal(At(C1, JFK) ∧ At(C2, SFO))
```

Planning as State-Space search

The second key development is the search algorithm. Once the problem break down the proposition into PDDL. we start to init the problem by taking the above action schema , init state and goal.
By taking this problem, we are still able to use the searching that we already implemented in the previous project to archive the goal. There are several common used search algorithm, such as forward state-space search, backward relevant-states search, etc. But here we

summarize the most effective search algorithm, A* search.

- It is an informed search algorithm, or a best-first search algorithm[3].
- It combine two cost as total cost *f(n) =g(n) + h(n)*
    - g(n) is the cost of the path from the start node to n,
    - h(n) is the heuristic estimates the cost of the cheapest path from n to the goal
- We use the PriorityQueue to get the lowest cost of the path.
- The complexity is much less than BFS O(b^d) if the A* heuristic performance is better. It can be O(A*^d).

Planning Graph

The third key development is the Planning Graph. This graph provide a better heuristic for searching. The benefit of the planning graph is able to apply to any of the search techniques we have seen so far[2]. The biggest benefit is polynomial size approximation instead of exponential size like bread first tree search.

- The planning graph beak down the problem into two iterate levels: state and action levels.
- Adding the mutex according 3 rule for action and 2 for state
- Sum all the level costs of the individual goals

Impact on the field of AI

Planning research has been central to AI since its inception, and papers on planning are a staple of mainstream AI journals and conferences[3].
The representation language like PDDL,STRIPS, and ADL make it possible to encode more realistic problems into computer-parsable, computable, searchable representation. Such standardized syntax has been for representing planning problems and has been used as the standard language for the International Planning Competition since 1998.
Search algorithm is able to provide general problem solving method and apply different heuristic for informed search or uninformed search.
Planning Graph are the best for NP-hard domains on the constrain-based approaches, while search-based approaches do better in domains where feasible solutions can be found without backtracking.

References

[1].Martin Kalisch, Stefan Konig: Comparison of STRIPS, ADL and PDDL, 2005
[2].Stuart Russel, Peter Norvig: Aritificial Intelligence: A modern Approach(3rd Edition)
*[3].https://en.wikipedia.org/wiki/A*_search_algorithm*