LOLLIPOP: aLternative Optimization with causaL Link Injection Partial Ordered Planning

Paper ID#42

Abstract. Abstract goes here

Introduction

Until of the end of the 90s, Partial-Order Planning (POP) was the dominant approach for tackling a planning problem. Its least commitment philosophy was the central interest of this popularity. However, since then state search planning took hold with more efficient forward search heuristics and fast backtracking.

This search of scalability and performance hides the greatest advantage of POP: flexibility. This advantage allows POP to efficiently build plans with the parallel use of actions. It also means that it can refine broken plans and optimize them further than a regular state-based planner.

In this paper, we explore new ideas to revive POP as an attractive alternative to other totally ordered approach. Some papers like UCPOP [1] and VHPOP [2] already tried to extend POP into such an interesting alternative. More recent efforts [3], [4] are trying to adapt the powerful heuristics from state-based planning to POP's approach.

Our approach is different: we prefer the use of the preprocessed domain causal graph and build an initial partial plan around it in order to already have an almost complete plan right away. This graph is also used during the refinement process for flaw and goal selection that will improve the branching factor of the algorithm. We also use it for consistency checks before applying flaws.

We also wanted to improve the plan quality as our input can be far from the optimal solution. This is the reason why we introduced two new negative flaws: the alternative and the orphan flaws.

In order to present our aLternative Optimization with causaL Link Injection Partial Ordered Planning (LOLLIPOP) system, we need to explain the classical POP framework and its limits.

1 The Partial Order Planning Framework

In this paper, we decided to build our own planning framework based on PDDL's concepts. This new framework is called WORLD as it is derived from more generalistic world description tools such as RDF and ontologies. It is about equivalent in expressiveness to PDDL 3.1 with object-fluents support.

We define our **planning domain** as a tuple $\Delta = \langle T, C, \mathcal{F}, F, O \rangle$ where

- T are the **types**,
- C is the set of **domain constants**,
- $\mathcal{F}$ is the set of **functions** with their arities and typing signatures,
- F represents the set of **fluents** defined as potential equations over the terms of the domain,
- O is the set of optionally parameterized **operators** with preconditions and effects.

The symbol system is completed with a notion of **term** (either a constant, a variable parameter or a property) and a few relations. We provide types with a relation of **subsumption** noted $t_1 \prec t_2$ with $t_1, t_2 \in T$ meaning that all instances of t_1 are also instances of t_2. On terms, we add two relations: the **assignation** (noted $\leftarrow$) and the **potential equality** (noted $\doteq$).

From there we add the definition of a planning problem as the tuple $\Pi = \langle \Delta, C_\Pi, I, G, P \rangle$ where

- Δ is a planning domain,
- C_Π is the set of **problem constant** disjoint from C,
- I is the **initial state**,
- G is the **goal**,
- P is a given **partial plan** formed as a tuple $\langle S, L, B \rangle$ with S the set of **steps** (instantiated operators also called actions), L the set of **causal links**, and B the set of **binding constraints**.

2 Motivating Examples

3 Improving POP's Performance

4 Improving POP's Quality

5 Properties of LOLLIPOP

6 Experiments

Conclusion

References

[1] J. S. Penberthy, D. S. Weld, and others, "UCPOP: A Sound, Complete, Partial Order Planner for ADL." *Kr*, vol. 92, pp. 103–114, 1992.

[2] H. akan L. Younes and R. G. Simmons, "VHPOP: Versatile heuristic partial order planner," *Journal of Artificial Intelligence Research*, pp. 405–430, 2003.

[3] A. J. Coles, A. Coles, M. Fox, and D. Long, "Forward-Chaining Partial-Order Planning." in *ICAPS*, 2010, pp. 42–49.

[4] O. Sapena, E. Onaindia, and A. Torreno, "Combining heuristics to accelerate forward partial-order planning," *Constraint Satisfaction Techniques for Planning and Scheduling*, p. 25, 2014.