

# Research review on planning and search

Rui Sá Pereira, Udacity AI nanodegree, term 1, project III

We start by giving an overview of the work developed by Waldinger ([1]), namely the introduction of goal-regression in order to achieve several goals simultaneously. In his paper, it is shown how it is possible to solve a given planning problem by considering a set of simplified problems, each corresponding to one of the sub-goals. This problem had already been considered in previous work from the author ([2]), but the resulting planning program could contain branches and recursive loops. In his later paper, the author makes use of backward regression to devise linear programming plans without branches or recursive loops, and whose construction depends crucially on the fundamental concept of protection. By this, it is meant the feature of a plan, in which we achieve a goal  $G = (G_1, \dots, G_n)$  by the recursive application of modifications where at a given step, the plan achieving a partial goal set  $G' = (G_1, \dots, G_j)$  is modified in order to achieve  $G_{j+1}$ , and the resultant modification preserves the attainment of  $G'$ . We remark the importance of this contribution to partial-order planning, seen for instance, on the use of goal-regression to the development of Warplan, the first planner entirely written in Prolog (see Russel and Norvig [5], p. 394 for more on this and other developments). Indeed, goal-regression was important for the development of partial-ordering systems, those last containing elements that integrated the development of more recent strategies, such as Graphplan that we now briefly present.

Graphplan, a planning algorithm that uses features of both partial-order and total-order planners, was introduced by Blum and Furst ([4]). Here, planning graphs are used both as heuristics and as output of the planning searches. The idea is to initialise the planning graph at  $S_0$  and then see if the plan is valid, i.e. the goal condition is present, in which case we try to extract a solution. Otherwise, we add  $S_1$  and  $A_1$ , and keep expanding until the graph either levels off or the goal is present at a certain level. Indeed this shows that "If a valid plan exists using  $t$  or fewer time steps, then that plan exists as a subgraph of the Planning Graph." Moreover, and an important advantage over most partial-order planners, the algorithm can be modified in a way that if no valid plan satisfies the problem goal, the planner will stop in a finite amount of time. Graphplan introduced for the first time the paradigm of Planning Graph Analysis, where a basic structure, the planning graph, is used to represent explicitly the inherent constraints to the problem, which reduces the amount of search needed. Indeed, the construction of a planning graph, where plans are not state-space paths but network flows, is polynomial both in size and in time. The use of Graphplan has become pervasive in planning, being used, for instance, as a benchmark method in a comparative study of some planning problems, whose main points we highlight in the last part of this review.

In their paper, Helmert [3] study the computational complexity of eight classes of planning problems in transportation domains. Some of these classes are variations of the plane-airport-cargo problem we have considered in the

first part of the project. The complexity of PlanSAT and Bounded PlanSAT of all of these classes is then studied. It is shown that Bounded PlanSat is *NP*-complete for all the classes considered, while PlanSat is also *NP*-complete, except for some simpler classes, whose complexity is *P*. Finally, it is shown that GraphPlan or other graph-planning system, IPP, provide the best approaches to the NP-hard problems, provided that the problems don't contain many actions. On the other hand, in problems where optimal plans are not required, local search planners such as Fast-Forward (FF) are preferable and GraphPlan and IPP do not perform as well.

In this review, we started by giving an overview of goal-regression, in the context of partial-ordering planning systems and their relevance to modern planners. Second, we collected the main features of Graphplan, an important planning system that remains relevant nowadays. Finally, we have partially illustrated why, rather than a consensual choice, the field of planning has benefitted from the use of several different type of planners such as GraphPlan or FF taken together as a toolbox, where to choose from when facing a certain class of problems.

## References

- [1] R. Waldinger. Achieving several goals simultaneously. *Machine Intelligence*, 8, 94–138, 1975.
- [2] R. Waldinger, K. Levit. Reasoning about programs. *Artificial Intelligence*, 5(3), 235–516, 1974.
- [3] M. Helmert. On the complexity of planning in transportation domains. *Proceedings of the European Conference on Planning-01*, 2001.
- [4] A. Blum, M. Furst . Fast planning through planning graph analysis. *Artificial Intelligence*, 90(1–2), 281–300, 1997.
- [5] S. Russel, P. Norvig. *Artificial Intelligence, A Modern Approach*, third edition, 2010.