

# Research Review

## Introduction

In this review, I will look at the original Graphplan paper and two paper that make improvements on Graphplan. The original Graphplan is similar to the version used in the planning project and the paper explore interesting extensions to the original algorithm.

## Fast Planning Through Planning Graph Analysis

In *Fast Planning Through Planning Graph Analysis* [1], the authors introduce the Graphplan algorithm to solve STRIPS problems. Before searching, a planning graph encoding useful constraints is generated. The plan is similar to a parallel plan where the actions in each time-step can be executed in any order. The planning graph consists of propositional and action levels connected by precondition, add or delete edges. Mutually exclusive actions are recorded if they have competing needs or interfere with each other. These actions can be omitted from an action level the planning graph

The planning graph process can be used to determine if a legal plan is possible; if a graph cannot be constructed, a valid solution is not possible. The graph also provides a structure and useful information during search. Graphplan uses a level by level back chaining approach to search. Forward checking using the planning graph checks for cut off goals.

The Graphplan algorithm was able to get very good results compared to contemporary planners (Prodigy and UCPOP) for established problems. A major limitation is that it only works in STRIPS-like problem areas. Domain that cannot be efficiently expressed in STRIPS can affect Graphplan's performance.

## Extending Planning Graphs to an ADL Subset

Koehler et al[2] developed IP2 to use the Graphplan algorithm on subset of the Action Description Language (ADL), allowing it to express more problems. STRIPS has no concept of explicit negative operators; any proposition that was not stated as true was considered false. This means that STRIPS also

does not express unknown literals. IP2 was able to use this expressivity to outperform Graphplan on most examples.

## Extending Graphplan to Handle Uncertainty & Sensing Actions

Weld et al.[3] also made improvements to Graphplan in SGP. This version of Graphplan adds two main concepts: multiple possible world states and the sensing action. By being able to satisfy parallel world states (PW), SGP can come up with solutions where a condition is not known. If a sensing action is possible, PWs can be reduced when a plan is executed. This improvement allows Graphplan to expand to domains where conditions aren't known either at all or at the time of execution.

## Conclusion

Graphplan introduced a novel approach to search problems that allowed for very good performance for certain domains. IP2 brought the approach to ADL and introduce additional expressivity. SGP allowed Graphplan to deal with uncertain situations in parallel or through responding to sensing actions.

## References

- [1] Blum, A., and Furst, M. 1997. Fast Planning Through Planning Graph Analysis. *J. Artificial Intelligence* 90(1-2):281-300
- [2] Koehler J., Nebel B., Hoffmann J., Dimopoulos Y. (1997) Extending planning graphs to an ADL subset. In: Steel S., Alami R. (eds) Recent Advances in AI Planning. ECP 1997. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1348. Springer, Berlin, Heidelberg
- [3] Weld, D., Anderson, C., and Smith, D. 1998. Extending Graphplan to handle uncertainty and sensing actions. In Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence (AAAI '98/IAAI '98). American Association for Artificial Intelligence, Menlo Park, CA, USA, 897-904.