# Udacity – AIND – AI Planning and Search Historic Topic Review

By: Cheryl Miller

## AI Planning and Search – GraphPlan and Distributed Computing

### Introduction

Artificial Intelligence (AI) has a long and rich history in Computer Science as does Distributed Computing. Distributed AI (DAI) has also been around for some time as well, at least since 1975. [1] DAI areas include Parallel Problem Solving, Distributed Problem Solving and Multi-Agent Based Simulation. [1].

With the current prevalence of distributed cluster environments, such as Apache Spark, I was curious to see where AI has gone in relation to Planning Problems in a distributed computing environment. I was particularly curious how the GraphPlan [2] algorithm may have evolved in the distributed world.

### Issues on Distributed Computing [1]

As [3] mentions, the GraphPlan algorithm had a major, positive impact on automated planning, however, the algorithm is geared toward a one-agent planning paradigm. Enhancing it to be usable in a distributed computing environment serves up a number of challenges, namely:

- How to break up the problem domain such that independent agents can solve sub-problems
- How agents communicate results and sub-problem domains with each other
- How to ensure the plan returned is correct and the most efficient possible, if not optimal

### Potential Distributed Solutions

The following 3 methodologies show different approaches to try to solve some of the issues above and still take advantage of the distributed computing advantages. While the last method does not deal with the GraphPlan algorithm directly, it does provide an alternate method of 'dividing and conquering' the problem domain, as do both other methods reviewed.

#### Distributed GraphPlan [3]

The Distributed GraphPlan algorithm uses 'Interaction Graphs' to define the problem domain. It first performs a centralized pre-planning technique using FFS to find an initial sequence of actions that allows the 'interaction Graph' to be split into 2 or more disconnected sub-graphs, each which can be solved in a distributed manner.

#### Goal directed GraphPlan [4]

The Goal-Directed GraphPlan implementation method took an approach of creating an 'operator-graph' by using a backwards-chaining process starting from the goal state. It only considers propositions in the goal state so it produces a graph with only relevant actions. Once this operator-graph is constructed a forward expansion of the planning graph can be created that only considers operators in the operators-graph, significantly reducing the size of the graph space. The operators-graph can be built distributed, however, the final backward graph search used to find a valid solution must be done in a centralized manner.

For the multi-agent communication, this method uses a Semantic Overlay Graph to discover and self-organize agents and their semantic connections with one another. This aids in building a more intelligent agent to agent communication process for the distributed interconnections.

### CSP+Planning [5]

The CSP+Planning method, while not strictly using a GraphPlan algorithm does, nevertheless, use similar 'divide and conquer' methods to solve a planning problem. This method distinguishes between public and private aspects of the problem solution. Public actions serve as coordination points between distributed agents in which Constraint Propagation is used, while private actions are solved in distinct agents using a local planning methodology.

## Summary

The GraphPlan algorithm proved to be an extremely valuable method for solving AI Planning problems, however, in today's highly distributed computing environments used by Big Data applications, it's important that this algorithm be implementable in a distributed environment. Breaking the problem domain up into smaller sub-problems is critical in this endeavor. Doing so manually may be possible for simpler problem domains, but certainly not viable in large complex planning problems. The 3 methods presented here attempt to solve this 'divide and conquer' issue and did so in different ways. They all have the same basic issues to consider, namely, how to break up the problem into distribute-ably solvable sub-problems, how to manage agent-to-agent communication in an efficient manner, and finally, to ensure the plan returned is complete, and as efficient as possible, hopefully optimal. Providing an optimal plan is not always feasible in a distributed GraphPlan algorithm, but it certainly needs to be as efficient as possible to warrant the overhead in running it in a distributed environment.

# REFERENCES


[1] Wikipedia, "Distributed Artificial Intelligence," [Online].

[2] A. Blum and M. Furst, "Fast planning through planning graph," *Artificial Intelligence,* pp. 90:281-300, 1997.

[3] M. Iwen and A. D. Mali, "Distributed Graphplan," in *Proceedings of IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Washington DC, Nov 2002.

[4] A. Lopes and L. Botelho, "Distributed Coordination of Heterogeneous Agents Using a Semantic Overlay Network and a Goal-Directed Graphplan Planner," *PLos ONE,* p. e62931. doi:10.1371/journal.pone.0062931, 2013.

[5] R. Nissim, R. I. Brafman and C. Domshlak, "A General, Fully Distributed Multi-Agent Planning," in *AAMAS*, Toronto, 2010.