# Implement a Planning Search

**Christopher Ohara**

AIND – Project 3 – Research Review

**1) Planning Graphs (GraphPlan) (1997)**

Carnegie Mellon developed a new approach to planning for domains similar to STRIPS. This became known as the "GraphPlan" algorithm and looks exactly like the structure used in Udacity's third AIND project. This algorithm ensures that the shortest path will be found and involves specifying mutually exclusive (mutex) relationships. This algorithm is useful because it reduces the search overhead time by explicitly encoding useful constraints. Interestingly enough, there are reports of the "Level Sum" heuristic outperforming uninformed heuristics and other A* search algorithms, possibly due to the complexity of the problem they were addressing. This algorithm, in practice, works differently than greedy searching algorithms. It should be noted that the state space search is a lot more work to build and expands exponentially. This might be problematic for complex problems, even though it simultaneously seems to perform best for complex systems (as a BFS or UCS will preform just as well and is much easier to code and implement).[1]

**2) OBDD-based planning algorithm (1994)**

This algorithm constructs a planner for image or pre-image processing. This has many applications in computer vision for robotics. This can be implemented in convolutional neural networks for data processes and object-oriented tracking. It is oriented around model-checking and the construction of forward/backwards based models. This algorithm can be useful in the detection of anomalous events (noise) and possibly paired with stochastic gradient descent or Kalman filtering. This would have applications in military situations or other security (CCTV) type detection applications. Similarly, this algorithm could be useful in the analysis of convolutions in linear systems for signals. The algorithm operates by traversing the state space in a similar fashion to BFS. Afterwards, it makes a decision to travel "backwards" or "forwards" depending on specifications in the parameter settings (pre-image versus image requirements). These planners are very useful for state space exploration and have a lot of potential to be expanded in the future for stochastic/dynamic/non-linear systems and processes, since the world is not truly static and relies on many approximations (and first-order linear representations) [2]

### 3) Heuristic Search Planner (HSP) (1998)

Based on heuristic search, this algorithm provides an estimate of the distance to the goal. The heuristics are derivative of the actions and goal nodes. It attempts to solve a relaxed version of the problem, and apply the results to the search. By generating states based on the actions from prior preconditions (this sounds familiar), it aims to estimate the total number of steps to arrive at a goal or subgoal. The subgoals are assumed to be independent and cause the heuristic to be not admissible. However, the algorithm seems to have a good performance in practice (like with the third project in Udacity's AIND program, in which it outperforms the GraphPlan heuristic model). However, this algorithm might not be ideal in situations that require high accuracy or precision like military applications, real-time systems or IoT web connectivity devices. [1]

1) https://towardsdatascience.com/ai-planning-historical-developments-edcd9f24c991

2) https://pdfs.semanticscholar.org/b089/5a0e741e2e3604a18ebb77df641742bf6077.pdf