Udacity – AIND – Planning Search and Heuristic Analysis

By: Cheryl Miller

INTRODUCTION

This project illustrates the use of planning search agents to solve deterministic logistics planning problems, specifically for an Air Cargo transport scheduling problem. It does this first using uninformed/non-heuristic searches such as Breadth First Search, Depth First Search, Uniform-cost, etc. Then we develop Planning Graphs and use domain independent heuristic searches utilizing heuristic functions such as ignore prerequisites and level sum.

UNINFORMED SEARCH

Uninformed search strategies have no additional information about states beyond that known at problem definition, they can only generate successors and distinguish between goal and non-goal states. These searches vary by the order in which nodes are expanded and explored. For this comparison I chose Breadth First, Depth First Graph, Uniform Cost and also included Greedy Best First Graph with h_1 heuristic (which really is none at all).

RESULTS

Figure 1- Uninformed Search Node Expansions

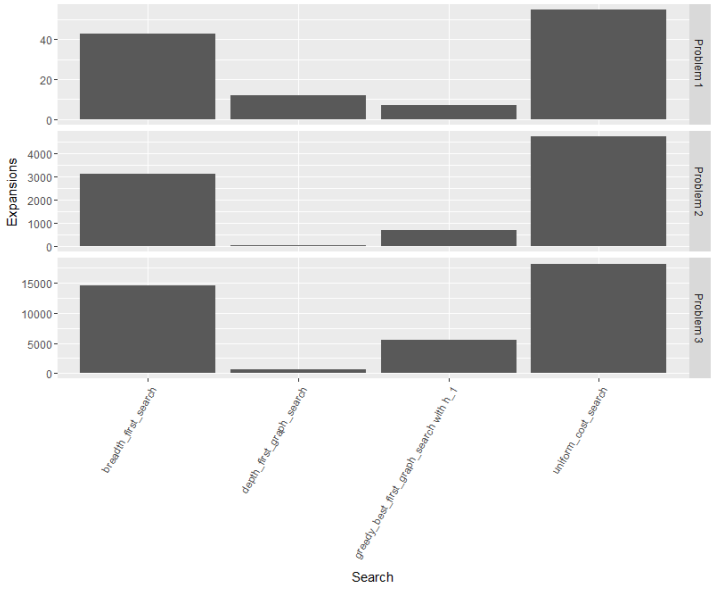

Table 1 - Uninformed Search Metrics

Problem	Search	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time In Seconds
Problem 1	breadth_first_search	43	56	180	6	0.076396247
Problem 1	depth_first_graph_search	12	13	48	12	0.020047441
Problem 1	greedy_best_first_graph_search with h_1	7	9	28	6	0.012739231
Problem 1	uniform_cost_search	55	57	224	6	0.099216847
Problem 2	breadth_first_search	3135	4330	26028	9	34.08708669
Problem 2	depth_first_graph_search	45	46	284	39	0.293353895
Problem 2	greedy_best_first_graph_search with h_1	701	703	5083	21	7.586322951
Problem 2	uniform_cost_search	4736	4738	38590	9	95.38413316
Problem 3	breadth_first_search	14663	18098	129631	12	302.3820175
Problem 3	depth_first_graph_search	627	628	5176	596	10.32648119
Problem 3	greedy_best_first_graph_search with h_1	5561	5563	49024	22	258.3068293
Problem 3	uniform_cost_search	18221	18223	159599	12	905.28807

ANALYSIS

The results highlighted in Figure 1- Uninformed Search Node Expansions and Table 1 show that while Depth First and Greedy Best First Graph searches have very fast elapsed times and low number of node expansions, neither presented an optimal plan. In this case Depth First offered a very bad plan with a plan length of 596 for problem 3. The plan for Greedy Best First Graph search was considerably closer at 22, but still not the optimal length of 12.

Both Breadth First and Uniform Cost searches found the optimal plan every time, but with considerably longer elapsed times and node expansions. Breadth First did better that Uniform Cost, both in elapsed time and nodes expanded. Breadth First stops as soon as it generates a goal state, while Uniform Cost needs to examine all the nodes at a given depth in the event there is a lower cost path.

For the non-informed searches, Breadth First search performed the best both in elapsed time and nodes expanded while still provided the optimal plan.

DOMAIN INDEPENDENT HEURISTIC SEARCH

Informed or Independent Heuristic searches use problem specific knowledge to find solutions more efficiently. These searches are similar to the Uniform Cost search, except they use a heuristic function to evaluate the cost rather than a set cost value. In this analysis, we look at the AStar searches using the h_1 heuristic as a baseline for AStar in general, then also look at the h_ignore_preconditions and the h_levelsum heuristics in combination with using a Planning Graph.

The h_ignore_precondition heuristic is a 'relaxed problem' in which all preconditions on an action are ignored so that all actions become applicable in every state and any single goal state can be achieved in a single step. This turns out to be an instance of the set-cover problem that can be solved using a simple greedy algorithm that returns a set covering size within a factor of log n of the true minimum covering.

The i_levelsum heuristic returns a sum of level costs of all the goals based on the **subgoal independence assumption** in which the cost of solving a conjunction of subgoals is approximated by the sum of the costs of solving each subgoal independently.

RESULTS

Figure 2- Heuristic Search Node Expansion

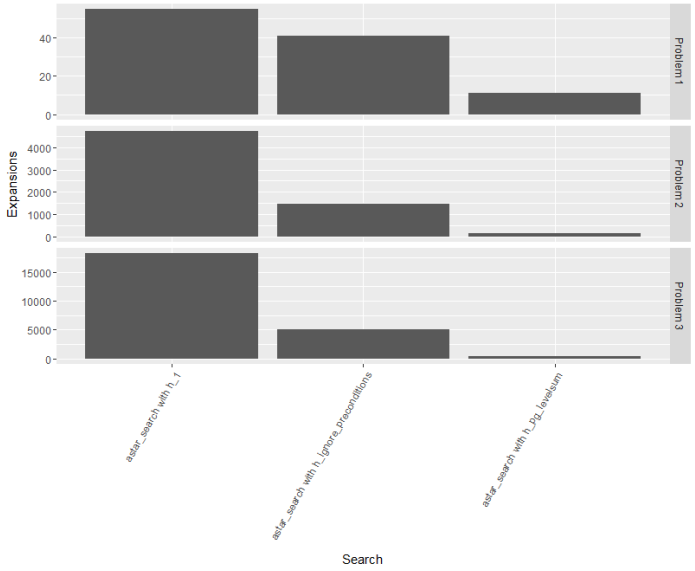

Table 2- Heuristic Search Metrics

Problem	Search	Expansions	Goal Tests	New Nodes	Plan Length	Elapsed Time In Seconds
Problem 1	astar_search with h_1	55	57	224	6	0.095631406
Problem 1	astar_search with h_ignore_preconditions	41	43	170	6	0.094212486
Problem 1	astar_search with h_pg_levelsum	11	13	50	6	3.025389207
Problem 2	astar_search with h_1	4736	4738	38590	9	91.80437931
Problem 2	astar_search with h_ignore_preconditions	1473	1475	12483	9	34.35804519
Problem 2	astar_search with h_pg_levelsum	133	135	1056	9	262.9190258
Problem 3	astar_search with h_1	18221	18223	159599	12	909.99928
Problem 3	astar_search with h_ignore_preconditions	5118	5120	45650	12	239.5802998
Problem 3	astar_search with h_pg_levelsum	414	416	3818	12	1968.385032

ANALYSIS

As the results in Table 2 and Figure 2 show, in general for the larger problem 2 and 3, both the AStar/h_ignore_preconditions and AStar/h_levelsum perform much better than the non-informed searches. The exception to this was for problem 1 in which the Greedy Best First Graph search did exceptionally well due to the simple and small graph generated. The Uniform Cost search was identical to the AStar/h_1 search, since they are ultimately equivalent due to the constant cost of 1 returned by the heuristic function.

RECOMMENDATION

For very small and simple planning graphs, the Breadth First search and Greedy Best First Graph searches prove to work well without a lot of complex overhead from the generation of a graph plan. For the larger and more complex problems, generating the graph plan and using the heuristics proved to be fast and complete. The h_levelsum heuristic proved to be the fastest and with the least number of node expansions and can be very useful for searching large problem graph plans.