Ai Nanodegree - Project 03 Implement a Planning Search - Heuristic Analysis

"This project objectives the study of classical PDDL (Planning Domain Definition Language) for the air cargo domain discussed in the lectures. You will then set up the problems for search, experiment with various automatically generated heuristics, including planning graph heuristics, to solve the problems, and then provide an analysis of the results". - from Udacity AI Nanodegree, implementing a Planning Search

Part 01. Non-heuristic Planning Solution Searches

The non-heuristics Breadth First Search, Breadth First Tree Search, Depth First Graph Search, Depth Limited Search and Uniform Cost Search, have been tested for all three air cargo problems. The metrics of the searches are located on Table 2, Table 3 and Table 4.

The best of the non-heuristic planning strategies was Breadth First Search: This search strategy always tries the shortest path first, it optimal and complete, but it has a time Complexity of O(V+E) where V is the number of vertices and E is the number of edges. The space complexity for he algorithm is 2^V, where V is the number of nodes. Because of the space complexity, the algorithm is not adequate for problems with a large number of nodes. [1]

The strategies Breadth First Tree Search and Depth Limited Search were not been able to find the solutions for all problems in a reasonably amount of time. The exponential complexity of the Depth Limited Search is apparent on its metrics for problem 2, where the number of nodes expansions and new nodes were 222719 and 2054119, respectively.

The best non-heuristic strategy, considering the number of node expansions, optimality of the solution, new nodes created and time elapsed, was Breadth First

Search followed by Uniform Cost Search. Depth First Graph Search strategy was been able to find a solution in all three air cargo problems, but the result was not been optimal.

Part 2 - Domain-independent heuristics

The heuristic strategies Recursive Best First Search, Greedy Best first Graph Search, and A* Best first search, have been tested for all three air cargo problems. The metrics of the searches are located on Table 2, Table 3 and Table 4.

The best of the heuristic planning strategie was A* with Ignore Preconditions heuristic: This search strategy always expands the path with the lowest f: g(i) + h(i), where g(i) is total cost of the path i and h(i) is the estimated distance of i to the goal. This strategy is the best possible in the sense that finds the shortest length path while expanding a minimum number of nodes possible[2]. If the heuristic function is optimistic and admissible the solution found is guaranteed to be the optimal one[1]. The best of the three heuristics analysed for this problem was “Ignore Pre-Conditions”, followed by “h(i) = 1” and “Level Sum”. It is clear by the results that the heuristic chosen greatly affect the computational resources that the A* search uses, with the “Level Sum” being the worst one, considering the time elapsed but it was the the best considering the number of node expansions and new nodes created. The time was greater using the ‘’Level Sum” strategy because the function that calculates the h(i) value is more complex than the others, requiring more time for the calculation. Probably this time could be improved using some Dynamic Programming technique to not recalculate the same things in different iterations of the algorithm.

The strategie “Recursive Best First Search” with “h(i) = 1” as heuristics, was not been able the find solutions for th Air Cargo 2 and 3 problems, in a reasonably amount of time.

Part 3: Written Analysis

The optimal plans for the air cargo problems can be found in Table 1. The algorithms that were been able to achieve the optimal plan can also be found on Table 1. The algorithms didn't provide the exactly same optimal plan for each problem, usually there was some difference in the order of the actions, but not in the number of actions. This happens because the order of expansions on the space state is possibly different for each algorithm.

The best search strategy was A* with “Ignore Preconditions” considering the combination between time elapsed of node expansions, goal tests, new nodes created, optimality and time elapsed for all three air cargo problems, with exception of time elapsed in the air cargo problem 1, where Breadth First Search won in this category, but lost considering the others categories cited. The search strategy was A* with “Level Sum” heuristics had the lowest overall number of node expansions, goal tests and new nodes, but the time required was significantly greater than the other “Ignore preconditions “ heuristics

Plan	Possible Optimal Sequence of Actions	Smallest Optimal Processing Time(s)	Optimal Algorithms
Air Cargo P1	Load(C1, P1, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO)	0.0031	breadth_first_search breadth_first_tree_search uniform_cost_search recursive_best_first_search(h_1) greedy_best_first_graph_search(h_1) astar_search(h1_1) astar_search(h_ignore_preconditions) astar_search(h_pg_levelsum)
Air Cargo P2	Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Fly(P3, ATL, SFO) Unload(C3, P3, SFO) Unload(C2, P2, SFO) Unload(C1, P1, JFK)	2.3453	breadth_first_searchbreadth_first_search uniform_cost_search astar_search(h1_1) astar_search(h_ignore_preconditions) astar_search(h_pg_levelsum)
Air Cargo P3	Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Fly(P1, ATL, JFK) Unload(C4, P2, SFO) Unload(C3, P1, JFK) Unload(C2, P2, SFO) Unload(C1, P1, JFK)	9.0777	breadth_first_search uniform_cost_search astar_search(h1_1) astar_search(h_ignore_preconditions) astar_search(h_pg_levelsum)

Table 1. Optimal Plan for Air Cargo Problems

Air Cargo P1						
Search Stategy	**node expansions**	**goal tests**	**plan length**	**new nodes**	**time elapsed(s)**	**optimal**
breadth_first_search	43	56	6	180	0.0170	yes
breadth_first_tree_search	1458	1459	6	5960	0.5329	yes
depth_first_graph_search	21	22	20	84	0.0077	no
depth_limited_search	101	271	50	414	0.0512	no
uniform_cost_search	55	57	6	224	0.0205	yes
recursive_best_first_search(h_1)	4229	4230	6	17023	1.6662	yes
greedy_best_first_graph_search (h_1)	7	9	6	28	0.0031	yes
astar_search(h1_1)	55	57	6	224	0.0231	yes
astar_search(h_ignore_preconditions)	41	43	6	170	0.0235	yes
astar_search(h_pg_levelsum)	39	41	6	158	0.4002	yes

Table 2. Metrics of the non-heuristic and heuristics searches for Air Cargo Problem 1

Air Cargo P2						
Search Strategy	**node expansions**	**goal tests**	**plan length**	**new nodes**	**time elapsed(s)**	**optimal**
breadth_first_search	3343	4609	9	30509	4.5699	yes
breadth_first_tree_search	-	-	-	-	-	-
depth_first_graph_search	624	625	619	5602	1.9550	no
depth_limited_search	222719	2053741	50	2054119	510.0288	no
uniform_cost_search	4852	4854	9	44030	6.5220	yes
recursive_best_first_search(h_1)	-	-	-	-	-	-
greedy_best_first_graph_search (h_1)	990	992	15	8910	1.3216	no
astar_search(h1_1)	4852	4854	9	44030	6.4771	yes
astar_search(h_ignore_preconditions)	1450	1452	9	13303	2.3453	yes
astar_search(h_pg_levelsum)	1129	1131	9	10232	139.5200	yes

Table 3. Metrics of the non-heuristic and heuristics searches for Air Cargo Problem 2

Air Cargo P3						
Search Strategy	**node expansions**	**goal tests**	**plan length**	**new nodes**	**time elapsed(s)**	**optimal**
breadth_first_search	14491	17947	12	128184	23.3548	yes
breadth_first_tree_search	-	-	-	-	-	-
depth_first_graph_search	2099	2100	2014	17558	12.6050	no
depth_limited_search	-	-	-	-	-	-
uniform_cost_search	18235	18237	12	159716	29.4296	yes
recursive_best_first_search(h_1)	-	-	-	-	-	-
greedy_best_first_graph_search (h_1)	5614	5616	14	49429	8.9372	no
astar_search(h1_1)	18235	18237	12	159716	28.7245	yes
astar_search(h_ignore_preconditions)	5040	5042	12	44944	9.0777	yes
astar_search(h_pg_levelsum)	2028	2030	12	17954	467.6350	yes

Table 4. Metrics of the non-heuristic and heuristics searches for Air Cargo Problem 3

References

[1]. Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).

[2]. AIND Lesson 10 - Search