HEURISTIC ANALYSIS

Hien, Hoang Chung

March 10, 2018

1. Problem Solutions

Step	**Problem 1**	**Problem 2**	**Problem 3**
01	Load(C1, P1, SFO)	Load(C3, P3, ATL)	Load(C2, P2, JFK)
02	Fly(P1, SFO, JFK)	Fly(P3, ATL, SFO)	Fly(P2, JFK, ORD)
03	Unload(C1, P1, JFK)	Unload(C3, P3, SFO)	Load(C4, P2, ORD)
04	Load(C2, P2, JFK)	Load(C2, P2, JFK)	Fly(P2, ORD, SFO)
05	Fly(P2, JFK, SFO)	Fly(P2, JFK, SFO)	Load(C1, P1, SFO)
06	Unload(C2, P2, SFO)	Unload(C2, P2, SFO)	Fly(P1, SFO, ATL)
07		Load(C1, P1, SFO)	Load(C3, P1, ATL)
08		Fly(P1, SFO, JFK)	Fly(P1, ATL, JFK)
09		Unload(C1, P1, JFK)	Unload(C1, P1, JFK)
10			Unload(C4, P2, SFO)
11			Unload(C2, P2, SFO)
12			Unload(C3, P1, JFK)

2. Experiments

Problem 1:

Search Method	Plan Length	Optimal?	Time Elapsed	Expansions	Goal Tests	New Nodes
breadth_first_search	6	Yes	0.034	43	56	180
depth_first_graph_search	20	No	0.015	21	22	84
uniform_cost_search	6	Yes	0.038	55	57	224
greedy_best_first_graph_search	6	Yes	0.005	7	9	28
astar_search with h_1	6	Yes	0.038	55	57	224
astar_search with h_ignore_preconditions	6	Yes	0.040	41	43	170
astar_search with h_pg_levelsum	6	Yes	0.638	11	13	50

Problem 2:

Search Method	Plan Length	Optimal?	Time Elapsed	Expansions	Goal Tests	New Nodes
breadth_first_search	9	Yes	9.211	3343	4609	30509
depth_first_graph_search	619	No	3.782	624	625	5602
uniform_cost_search	9	Yes	13.387	4853	4855	44041
greedy_best_first_graph_search	21	No	2.630	998	1000	8982
astar_search with h_1	9	Yes	12.842	4853	4855	44041
astar_search with h_ignore_preconditions	9	Yes	4.636	1450	1452	13303
astar_search with h_pg_levelsum	9	Yes	61.718	86	88	841

Problem 3:

Search Method	Plan Length	Optimal?	Time Elapsed	Expansions	Goal Tests	New Nodes
breadth_first_search	12	Yes	50.572	14663	18098	129631
depth_first_graph_search	392	No	2.059	408	409	3364
uniform_cost_search	12	Yes	62.897	18204	18206	159449
greedy_best_first_graph_search	27	No	12.617	3929	3931	34950
astar_search with h_1	12	Yes	55.120	18204	18206	159449
astar_search with h_ignore_preconditions	12	Yes	19.659	4804	4806	42655
astar_search with h_pg_levelsum	12	Yes	332.244	299	301	2760

From the performance tables above, Breadth First Search seem to be a great choice if we looking for a non-heuristic searching algorithm. BFS always finds an optimal solution as explained in AIMA book, Peter Norvig, 3rd edition, section 3.4.1. But the search tree expands very fast as the state space become bigger in problem 3. This is clearly not a efficient planning

way if the planning problem contains too much states. Compared to the heuristic searching algorithm, A* search with Ignore Preconditions heuristic seem to be a best candidate with a simple heuristic algorithm. As proven in AIMA book, 3rd edition, section 3.5.2, A* search always find the optimal solution with a consistent heuristic algorithm. In this experiment, Ignore Preconditions heuristic was simple and consistent to achieve a very good performance.

3. Conclusion

From the experiment above, we have compared the performance of different algorithms to search for the solution in the small state space to the big state space. A* search with the Ignore Preconditions heuristic algorithm was the best choice to solve this planning problem. The algorithm give the optimal solutions in all the experiments with a reasonable compute time.