HEURISTIC ANALYSIS: AI PLANNING AND SEARCH

CHRISTOPHER BRIAN CURRIN

Problem	Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Time Elapsed
1	breadth_first_search...	43	56	180	**6**	0.08
	breadth_first_tree_search...	1458	1459	5960	**6**	2.10
	depth_first_graph_search...	21	22	84	**12**	0.03
	depth_limited_search...	101	271	414	**50**	0.19
	uniform_cost_search...	55	57	224	**6**	0.08
	recursive_best_first_search with h_1...	4229	4230	17023	**6**	5.86
	greedy_best_first_graph_search with h_1...	7	9	28	**6**	0.01
	astar_search with h_1...	55	57	224	**6**	0.08
	astar_search with h_ignore_preconditions...	41	43	170	**6**	0.06
	astar_search with h_pg_levelsum...	11	13	50	**6**	2.75
2	breadth_first_search...	3343	4609	30509	**9**	78.00
	breadth_first_tree_search...	-	-	-	**-**	-
	depth_first_graph_search...	624	625	5602	**619**	20.81
	depth_limited_search...	222719	2053741	2054119	**50**	4830.82
	uniform_cost_search...	4853	4855	44041	**9**	152.89
	recursive_best_first_search with h_1...	-	-	-	**-**	-
	greedy_best_first_graph_search with h_1...	998	1000	8982	**21**	36.28
	astar_search with h_1...	4853	4855	44041	**9**	200.88
	astar_search with h_ignore_preconditions...	1506	1508	13820	**9**	25.64
	astar_search with h_pg_levelsum...					
3	breadth_first_search...	14663	18098	129631	**12**	312.87
	breadth_first_tree_search...	-	-	-	**-**	-
	depth_first_graph_search...	408	409	3364	**392**	8.32
	depth_limited_search...	-	-	-	**-**	-
	uniform_cost_search...	-	-	-	**-**	-
	recursive_best_first_search with h_1...	-	-	-	**-**	-
	greedy_best_first_graph_search with h_1...	4255	4257	37660	**14**	324.52
	astar_search with h_1...	-	-	-	**-**	-
	astar_search with h_ignore_preconditions...	5087	5089	45349	**12**	205.80
	astar_search with h_pg_levelsum...	-	-	-	**-**	-

Optimal Plans

A sample of the optimal plans are in the table below. Deviations from the below were produced depending on the algorithm.

Problem 1 (plan length 6)	Problem 2 (plan length 9)	Problem 3 (plan length 12)
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	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)	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)

Discussion

For each problem, A* with ignore preconditions heuristic seems to be the most optimal heuristic and depth first (graph) search the worst. This was judged on accuracy (did it give the correct result) vs time elapsed vs new nodes. While other heuristics gave the correct length, they were generally slower and/or visited more new nodes. Better heuristics may result in fewer node expansions, but take longer to compute. The trade-off of the A* search with ignore preconditions heuristic is that it is fast, but has more node expansions than the more accurate (in terms of distance to goal) A* level sum heuristic. In certain problems, here air cargo, this speed over accuracy trade-off is a clear winner, but other problems may benefit from more accurate heuristic to reach the optimal solution.

The depth first search, while consistently the fastest to complete, also gave a sub-optimal plan length. This is due to its completion once it has reach a valid solution (passing the goal test) and not the *optimal* solution. A* in contrast chooses the best node of possible actions (according to its heuristic), and therefore can reach the optimal solution.

Breadth first search performed the best of the uninformed heuristics. Both breadth-first and A* searches are *complete* and *optimal* and given no actual heuristic, A* acts equivalent to breadth-first search. Due to the 10-min timeout (sometimes relaxed) many algorithms did not complete to give a result, but for the given results, problem 1 is indicative of the performance of an algorithm on the other problems. On the air cargo problem, we can expect that A* with level sum heuristic will provide the optimal solution with the minimal number of new nodes, but in a substantial amount of time compared to another optimal algorithm. Of course, timing is dependent on the simulation hardware, and a dual-core 2011 laptop probably does not provide the best results.