

Planning Agent Final Report

In this project we focused on the problem of solving a deterministic logistics planning problem for Air Cargo Transport System. Our search and planning agent efficiently plans a strategy for this process.

I have experimented several search algorithms and heuristics on four different types of Air Cargo problem 1, 2, 3, 4 respectively.

Questions:

Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

Since execution time to find the goal critical here, we need to settle on the algorithm that runs the fastest. Breath First Search would be the ideal choice given that the problem domain is very restricted, also it helps to find the optimal plan. Greedy Best First would be good only if we can come up with a heuristic that works well for the problem domain.

Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

Uniform Cost Search or Breath First Search are the two choices for finding optimal plans as seen from results above. However, of the two Breath First Search would be a better choice as it is more efficient (speed and node expansion).

Table Report:

Problem 1 (Actions 20)

Search Strategy	Optimal	Path Length	Execution Time(s)	Node Expansions
breadth_first_search	Yes	6	0.027	43
depth_first_graph_search	No	20	0.01	21
uniform_cost_search	Yes	6	0.024	60
greedy_best_first_graph_search_h_unmet_goals	Yes	6	0.021	7
greedy_best_first_graph_search_h_pg_levelsum	Yes	6	0.52	6
greedy_best_first_graph_search_h_pg_maxlevel	Yes	6	0.13	6
greedy_best_first_graph_search_h_pg_setlevel	Yes	6	0.95	6
astar_search_h_unmet_goals	Yes	6	0.015	50
astar_search_h_pg_levelsum	Yes	6	0.32	28
astar_search_h_pg_maxlevel	Yes	6	0.19	43
astar_search_h_pg_setlevel	Yes	6	0.53	33

Example Optimal plan length (greedy_best_first_graph_search with h_pg_maxlevel)

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Fly(P1, SFO, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

Problem 2 (Actions 72)

Search Strategy	Optimal	Path Length	Execution Time(s)	Node Expansions
breadth_first_search	Yes	9	0.35	3343
depth_first_graph_search	No	619	0.56	624
uniform_cost_search	Yes	9	0.62	5154
greedy_best_first_graph_search_h_unmet_goals	Yes	9	0.027	17
greedy_best_first_graph_search_h_pg_levelsum	Yes	9	0.46	9
greedy_best_first_graph_search_h_pg_maxlevel	Yes	9	0.75	27
greedy_best_first_graph_search_h_pg_setlevel	Yes	9	2.15	9

astar_search_h_unmet_goals	Yes	9	0.67	2467
astar_search_h_pg_levelsum	Yes	9	11.24	357
astar_search_h_pg_maxlevel	Yes	9	64.54	2887
astar_search_h_pg_setlevel	Yes	9	172.92	1037

Example Optimal plan length (greedy best first graph search with h_pg_levelsum)

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Problem 3 (Actions 88)

Search Strategy	Optimal	Path Length	Execution Time(s)	Node Expansions
breadth_first_search	Yes	12	1.02	14663
greedy_best_first_graph_search_h_unmet_goals	No	15	0.026	25
greedy_best_first_graph_search_h_pg_levelsum	No	14	1.87	14
astar_search_h_unmet_goals	Yes	12	1.36	7388
astar_search_h_pg_levelsum	Yes	12	20.84	369

Example Optimal plan length (astar_search with h_unmet_goals)

Load(C2, P2, JFK)
 Fly(P2, JFK, ATL)
 Load(C3, P2, ATL)
 Fly(P2, ATL, ORD)
 Load(C4, P2, ORD)
 Fly(P2, ORD, SFO)
 Unload(C4, P2, SFO)
 Unload(C2, P2, SFO)
 Load(C1, P2, SFO)
 Fly(P2, SFO, JFK)
 Unload(C3, P2, JFK)
 Unload(C1, P2, JFK)

Problem 4 (Actions 104)

Search Strategy	Optimal	Path Length	Execution Time(s)	Node Expansions
breadth_first_search	Yes	14	5.67	99736
greedy_best_first_graph_search_h_unmet_goals	No	18	0.03	29
greedy_best_first_graph_search_h_pg_levelsum	No	17	2.52	17
astar_search_h_unmet_goals	Yes	14	4.96	34330
astar_search_h_pg_levelsum	No	15	109.85	1208

Example Optimal plan length (astar_search with h_unmet_goals)

Load(C2, P2, JFK)
 Fly(P2, JFK, ATL)
 Load(C3, P2, ATL)
 Fly(P2, ATL, ORD)
 Load(C4, P2, ORD)
 Load(C5, P2, ORD)
 Fly(P2, ORD, SFO)
 Unload(C4, P2, SFO)
 Unload(C2, P2, SFO)
 Load(C1, P2, SFO)
 Fly(P2, SFO, JFK)
 Unload(C5, P2, JFK)
 Unload(C3, P2, JFK)
 Unload(C1, P2, JFK)

Discussion of results:

As far as the search complexity i.e. nodes expanded for the search is considered, greedy best first search consistently performs better than other algorithms even as the problem size increases. The uniform cost search expands the most nodes followed by breadth first search. For smaller problem size where we have limited actions, depth first search work better than the a* star algorithms but the a*star algorithms having better heuristic out-perform the non-heuristic searches as the problem domain increases.

For search time i.e. execution speed is considered, for a small problem domain the execution time is very close for different algorithms with uniformed searches doing slightly better. The greedy best first searches are once again better even with increasing actions. The uniformed search execution times tends to worsen as the actions increases and the a*star algorithms are the slowest except for `astar_search_h_unmet_goals`.