Heuristic Analysis

Following table contains the results of the three problems

Problem 1

Search Strategy	Optimal	Path Length	Execution Time	Node Expansions	Goal Tests	New Node
breadth_first_search	Yes	6	0.028065395	43	56	180
breadth_first_tree_search	Yes	6	0.80468763	1458	1459	5960
depth_first_graph_search	No	12	0.007045876	12	13	48
depth_limited_search	No	50	0.079654677	101	271	414
uniform_cost_search	Yes	6	0.042744133	55	57	224
recursive_best_first_search h_1	Yes	6	2.336595606	4229	4230	17029
greedy_best_first_graph_search h_1	Yes	6	0.004809664	7	9	28
astar_search h_1	Yes	6	0.036459073	55	57	224
astar_search h_ignore_preconditions	Yes	6	0.03754027	41	43	170
astar_search h_pg_levelsum	Yes	6	0.910156524	11	13	50

Problem 2

Search Strategy	Optimal	Path Length	Execution Time	Node Expansions	Goal Tests	New Node
breadth_first_search	Yes	9	12.05155342	3401	4672	31049
breadth_first_tree_search						
depth_first_graph_search	No	346	1.267385031	350	351	3142
depth_limited_search	No	50	853.660577	254020	2344879	2345254
uniform_cost_search	Yes	9	9.462551091	4761	4763	43206
recursive_best_first_search h_1						
greedy_best_first_graph_search h_1	Yes	9	1.08202441	550	552	4950
astar_search h_1	Yes	9	9.652068039	4761	4763	43206
astar_search h_ignore_preconditions	Yes	9	3.559574226	1450	1452	13303
astar_search h_pg_levelsum	Yes	9	179.7641605	86	88	841

Problem 3

Search Strategy	Optimal	Path Length	Execution Time	Node Expansions	Goal Tests	New Node
breadth_first_search	Yes	12	84.78949499	14491	17947	128184
breadth_first_tree_search						
depth_first_graph_search	No	1878	16.94639373	1948	1949	16253
depth_limited_search						
uniform_cost_search	Yes	12	42.30773762	17783	17785	155920
recursive_best_first_search h_1						
greedy_best_first_graph_search h_1	No	22	9.349271817	4031	4033	35794
astar_search h_1	Yes	12	40.74106579	17783	17785	155920
astar_search h_ignore_preconditions	Yes	12	13.43286583	5003	5005	44586
astar_search h_pg_levelsum	Yes	12	948.7825005	311	313	2863

Question:

Provide an optimal plan for Problems 1, 2, and 3?

Solution: :

Optimal Plan for Problem 1:
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Optimal Plan for Problem 2:
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)

Optiomal Plan for Problem 3:
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(P1, ATL, JFK)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)

Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Question:

Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions)

Solution:

I’m going to compare and contrast breadth-first, depth-first and uniform-cost search

From the table, I can conclude that DEPTH FIRST search is not yielding optimal path but it is the Fastest algorithm among the three and uses least amount of memory for all 3 problems.

BREADTH FIRST and UNIFORM COST searches are providing the optiomal path. But if we compare on the basis of memory usage then, BREADTH FIRST search is better for all three problems.If we compare on execution time, then BREADTH FIRST is better for Problem 1 and UNIFORM COST is better for problem 2 and 3.

Question:

Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3?

Solution:

From above table, we can conclude that both A*_ignore_precondition and A*_level_sum produces the optimal solution for all 3 problem .But if we compare on the basis of execution time,A*_ignore_preconditions is better and if we compare on the basis of memory usage,A*_level_sum is better.

Question:

What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?

Solution:

A*_search_ignore_preconditions is the best choice of heuristic. This heuristic function drops all preconditions from actions. Every action becomes applicable in every state, and any single goal fluent can be achieved in one step [PAGE 376].

This is better than the Breadth First Search in time performance but number of expansions are much less. This is due to the fact that additional computation cost of the heuristics based search adds an increased amount of running per expansion .But it this increase in complexity will enhance the guidance requiring fewer exapnsions and producing better results. So as the complexity and search space increases,hueristic based A* will perform much better than non-heuristic based search.

Reference

Norvig, Peter. Russell, Stuart J. Artificial Intelligence A Model Approach: Third Edition. Pearson Education 2015.