Heuristic Analysis of Project 3 - Implement a Planning Search

Non-Heuristic Search

Optimal Sequence of Actions

Optimal sequence of actions is obtained for all problems.
Here the optimal paths are presented for all the problems with Breadth-First-Search, the statistics of "all" search algorithms are presented in the next section.

BFS:
P1:
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

P2:
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Load(C3, P3, ATL)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

P3:
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)

Performance Results

For Problem 3, depth_limited_search takes longer than 10 minutes.

P1	# Node Expansions	# Goal Tests	Time Elapsed (sec)	New Nodes	Plan Length
BFS	43	56	0.028238148	180	6
DFS	12	13	0.007673719	48	12
UCS	55	57	0.032504742	224	6

P2	# Node Expansions	# Goal Tests	Time Elapsed (sec)	New Nodes	Plan Length
BFS	3343	4609	11.93877162	30509	9
DFS	582	583	2.672747353	5211	574
UCS	4853	4855	10.63798922	44041	9

P3	# Node Expansions	# Goal Tests	Time Elapsed (sec)	New Nodes	Plan Length
BFS	14663	18098	93.78778289	128554	12
DFS	627	628	2.848914197	5176	596
UCS	18223	18225	47.9988163	158186	12

- It can be observed that DFS takes much shorter time to reach a solution but the solution is not optional. Hence DFS is ideal for fast execution.
- DFS also results in much lower nodes expanded. Hence takes also less time to reach a solution. DFS traverses the branches in depth first and then backs up, hence enters many nodes
- BFS and UCS take longer but reach **optimal solutions** (as explained in the course videos).

Heuristic Search

With A* search we get the following results:

P1	# Node Expansions	# Goal Tests	Time Elapsed (sec)	New Nodes	Plan Length
h_1	55	57	0.03184958	224	6
h_ignore_preconditions	41	43	0.024180792	170	6
h_levelsum	11	13	0.611260611	50	6

P2	# Node Expansions	# Goal Tests	Time Elapsed (sec)	New Nodes	Plan Length
h_1	4853	4855	10.60355647	44041	9
h_ignore_preconditions	1450	1452	3.220270479	13303	9
h_levelsum	86	88	53.19310277	841	9

P3	# Node Expansions	# Goal Tests	Time Elapsed (sec)	New Nodes	Plan Length
h_1	18223	18225	48.61868646	158186	12
h_ignore_preconditions	5040	5042	13.9217372	44720	12
h_levelsum	445	447	394.6557884	4028	12

- h_ignore_preconditions is fastest and uses decent amount of memory, based on # node expansions.
- h_level_sum is the slowest but uses very little memory compared to others, again based on # node expansions.
- The reason that h_levelsum takes longer to reach a solution is because the calculation of the level costs.
- All three seem to provide the optimal solution.
- h_ignore_preconditions is the best heuristic and performs even better than BFS in terms of execution time and # of node expansions.