Udacity Artificial Intelligence Project3: Implement a Planning Search

Julianne Hong

Problem 1

	Plan length	Time elapsed	Expansions	Goal tests	New nodes
Breadth_first_search	6	0.163	55	57	224
Depth_first_search	20	0.083	21	22	84
Uniform_cost_search	6	0.181	55	57	224
Ignore_precondition	6	0.149	41	43	170
Pg_levelsum	6	0.655	11	13	50

Optimal Plan Length: 6

Optimal Sequence of Actions:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Problem 2

	Plan length	Time elapsed	Expansions	Goal tests	New nodes
Breadth_first_search	9	36.3	3343	4609	30509
Depth_first_search	619	7.484	624	625	5602
Uniform_cost_search	9	45.17	4853	4855	44041
Ignore_precondition	9	14.02	1450	1452	13303
Pg_levelsum	9	54.62	86	88	841

Optimal Plan length: 9

Optimal Sequence of Actions:

Load(C3, P3, ATL)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Load(C2, P2, JFK)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Problem 3

	Plan length	Time elapsed	Expansions	Goal tests	New nodes
Breadth_first_search	12	237.73	14663	18098	129631
Depth_first_search	392	5.04	408	409	3364
Uniform_cost_search	12	205	18223	18225	159618
Ignore_precondition	12	58.01	5040	5042	44944
Pg_levelsum	12	272.96	316	318	2912

Optimal Plan Length: 12

Optimal Sequence of Actions:

Load(C2, P2, JFK)

Fly(P2, JFK, ORD)

Load(C4, P2, ORD)

Fly(P2, ORD, SFO)

Unload(C4, P2, SFO)

Load(C1, P1, SFO)

Fly(P1, SFO, ATL)

Load(C3, P1, ATL)

Fly(P1, ATL, JFK)

Unload(C3, P1, JFK)

Unload(C2, P2, SFO)

Unload(C1, P1, JFK)

Non-heuristic Search

Breadth First Search (BFS) and Uniform Cost Search (UCS) can compute the optimal solution for all three problems. These two solutions are very similar in time spend, expansion and number of new nodes created. In Problem1, both solution results the same except the little bit of time difference of time elapsed. Depth First Search (DFS), on the other hand, consumes the shortest time and smallest memory among the three non-heuristic searches, but it fails to find the optimal solution, since the algorithm focuses on finding solution as soon as possible rather than generating an optimal one. This is explained at *Artificial Intelligence: A Modern Approach* by Norvig and Russell section 3.4.7. BFS is complete and optimal whereas DFS is not complete and not optimal. BFS's time and space complexity is O(b^d) where b is the branching factor and d is depth of the shallowest solution. DFS is much better at space complexity, O(bm) where m is maximum depth of the search tree. Time complexity of DFS is O(b^m). Since DFS expand tree significantly less than BFS, it should spend much less time and space compare to BFS. BFS Search is recommended to solve Air Cargo problem. Although BFS and UCS are both capable of producing optimal solution, and usage of resource are not much different, BFS is doing a little better on memory management as problem gets complicated. The difference is somehow significant in Problem3 compared to Problem1.

Heuristic Search

Ignored_precondition and Level_sum heuristics both can generate the optimal solution. The Ignored_precondition heuristic takes siginificantly less time to compute than level_sum simply because it's doing much simpler operation. As mentioned in *Artificial Intelligence: A Modern Approach* by Norvig and Russell explained in chapter 10 and lecture in udacity, making problem relaxed, the computational burden gets much lower, although it might expand nodes a little more than necessary due to loosen scrutiny when expanding node. For all the problems, level_sum approach uses much less node expansion.

When a problem is simple like Problem 1, there is almost no advantage of using heuristic function over non-heuristic, but as problem gets complicated the heuristic search becomes necessary. Problem 3 is more complex than Problem 1, but compared to real life situations where similar problems can be extremely different in terms of complexity, it's not much. Also, if we are going to use heuristics over non-heuristics, then the ignored_precondition is recommended, since level_sum simply takes longer than any other non-heuristic search. Moreover, ignore_preconditions memory consumption is still better than all of the non-heuristics.