HEURISTIC_ANALYSIS

PLANNING SEARCH COMPARISON ANALYSIS

Ruihan Zou

This report is the comparison among planning searches on the three planning problems. The following tables are showing the details of planning searches performance on each problem.

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time elapsed	Optimality
Breadth First Search	43	56	180	6	0.0319	Yes
Breadth First Tree Search	1458	1459	5960	6	0.961	Yes
Depth First Graph Search	21	22	84	20	0.015	No
Depth Limited Search	101	271	414	50	0.0935	No
Uniform Cost Search	55	57	224	6	0.0375	Yes
Recursive Best First Search H1	4229	4230	17023	6	2.824	Yes
Greedy Best First Search H1	7	9	28	6	0.0053	Yes
Astar Search with h_1	55	57	224	6	0.0403	Yes
Astar Search with h_ignore_preconditions	41	43	170	6	0.0389	Yes
Astar Search with h_pg_levelsum	55	57	224	6	1.724	Yes

Air Cargo Problem 1

The optimal plan for problem one is found by Breadth First Search, Breadth First Tree Search, Uniform Cost Search, Recursive Best First Search, Greedy Best First Graph Search, A* Search with a heuristic. The priority on choosing the best planning search is following this rule: plan length > Time elapsed > Expansions and the one has smaller value is the better one.

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

Based on the above rule, the Greedy Best First Search H1 is the best planning search solution for problem1. The greedy search tries to expand the node which distance is closet to the goal. Here we use the straight-line distance heuristic. In problem 1, the goal is straight forward, and there is no any obstacle between the destination. In this case, the greedy algorithm can guarantee the best solution would be found. The Breath First Graph Search is also doing well as the goal is at a shallow depth, which means it is a limited depth. However, it is not guaranteed that Breath First Search can give an optimal solution if it the actions cost is decreasing. The Depth-First Graph Search reaches the goal in short time, but the path length is not optimal. Here is because the property of Depth First Graph Search, which expanded the deepest node in the current frontier of the search tree. So once it reaches the goal, the search would stop though there is an optimal one from other nodes. The performance of Astar search depends on the heuristic function, and the optimal solution is guaranteed.

Search Type	Expansions	Goal Tests	New Nodes	Plan Length	Time elapsed	Optimality
Breadth First Search	3401	4672	31049	9	15.016	Yes
Breadth First Tree Search	-	-	-	-	-	-
Depth First Graph Search	1192	1193	10606	1138	9.287	No
Depth Limited Search	-	-	-	-	-	No

Search						
Uniform Cost Search	4761	4763	43206	9	12.377	Yes
Recursive Best First Search H1	-	-	-	-	-	No
Greedy Best First Search H1	550	552	4950	9	1.424	Yes
Astar Search with h_1	4761	4763	43206	9	11.984	Yes
Astar Search with h_ignore_preconditions	1450	1452	13303	9	4.415	Yes
Astar Search with h_pg_levelsum	-	-	-	-	-	No

Air Cargo Problem 2

Breadth First Tree Search, Depth-Limited Search, Recursive Best First Search H1 and Astar Search h_pg_levelsum exceeded the time limit.

The optimal solution is: - 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)

In problem 2, the Greedy Best First Search is still the best solution. The second choice is Astar Search with h_ignore_preconditions. The Breath First Search took much longer time(15s > 0.0319) comparing problem 1. Because the depth and branch of search graph has increased and the time complexity is BFS takes O(b^(d + 1)) time and memory, where b is the "branching factor" of the graph (the average out-degree)

Search Type	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time elapsed**	**Optimality**
Breadth First Search	14491	17947	128184	12	111.088	Yes

Breadth First Tree Search	-	-	-	-	-	-
Depth First Graph Search	2099	2100	17558	2014	24.345	No
Depth Limited Search	-	-	-	-	-	No
Uniform Cost Search	17783	17785	155920	12	52.357	Yes
Recursive Best First Search H1	-	-	-	-	-	No
Greedy Best First Search H1	4031	4033	35794	22	11.924	No
Astar Search with h_1	17783	17785	155920	12	53.047	Yes
Astar Search with h_ignore_preconditions	5003	5005	44586	12	17.453	Yes
Astar Search with h_pg_levelsum	-	-	-	-	-	No

Air Cargo Problem 3

From Problem 3, the Astar search with h_ignore_predictions is the best solution. The ignore preconditions relaxes all preconditions from actions. In this air cargo problem 3, no matter C1 in which airport, it can reach to JFK directly At any plane which will land at JFK. The below is the optimal solution:

- 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)

Greedy Best First Search H1 fails on giving the optimal solution. The reason is there are airports in between the route of an initial airport and destination airport. The Best first search is taking much longer time than in problem 1 and 2 when cargos(branch factor) and airports are increasing(depth factor).

In conclusion, we choose Breadth First Search, or Greedy Best First Search when the problem has a swallow goal. But one of the advantages of Breadth First Search is it can guarantee an optimal solution. If we have memory limit and we just want to know whether a solution of reaching to the goal exist for the problem, we choose Depth First Search. Otherwise, the Astar search with heuristic can be considered when the problem is complex which means the search space is huge.

REFERENCE

Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach (3rd ed.). Boston: Pearson.