heuristic_analysis

Table of Contents

1 Heuristic Analysis

1.1 Metrics for non-heuristic planning solution searches

1.1.1 Problem 1

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed	Optimal
breadth_first_search	43	56	180	6	0.04683144300361164	Yes
depth_first_graph_search	21	22	84	20	0.01650939101818949	No
uniform_cost_search	55	57	224	6	0.05334271999890916	Yes

1.1.2 Problem 2

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed	Optimal
breadth_first_search	3343	4609	30509	9	17.596341395023046	Yes
depth_first_graph_search	624	625	5602	619	4.402504776982823	No
uniform_cost_search	4853	4855	44041	9	14.847201219003182	Yes

1.1.3 Problem 3

Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed	Optimal
breadth_first_search	8602	11196	64308	12	63.57823093398474	Yes
depth_first_graph_search	1292	1293	5744	875	5.141792179987533	No
uniform_cost_search	11482	11484	85785	12	47.939871934009716	Yes

BFS and UCS gives us the optimal and complete solution. DFS while being fastest for this small problem and more efficient in term of resources gives us a non optimal solution. BFS is optimal when path cost is same or non decreasing function of depth. It is because when a goal is reached, at that depth all other nodes with lesser depth have already been rejected. DFS isn't optimal because it searches till the leaf one by one which causes it to ignore depth. [1] One solution to this is iterative deepening search.

1.2 Metrics for non-heuristic planning solution searches

1.2.1 Problem 1

Heuristic	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
h_1	55	57	224	6	0.04679326497716829
h_ignore_preconditions	41	43	170	6	0.05496499201399274
h_pg_levelsum	39	41	158	6	1.2185538750200067

1.2.2 Problem 2

Heuristic	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
h_1	4853	4855	44041	9	17.488993248000043
h_ignore_preconditions	1450	1452	13303	9	6.052451260999078
h_pg_levelsum	1129	1131	10232	9	532.737853553990

1.2.3 Problem 3

Heuristic	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
h_1	11482	11484	85785	12	52.21306998498039
h_ignore_preconditions	4119	4121	31485	12	22.876345383003354

Heuristic	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
h_pg_levelsum	1958	1960	15626	12	1196.0613465080096

h_ignore_preconditions heuristic provides results in least time. It expands less nodes than h_1(the worst heuristic possible/not a true heuristic). h_pg_levelsum provides results in slowest time. A* search gives optimal results, so these results were optimal. Level sum while having less node expansions, tests and new nodess is quite slower because it needs to have planning graph constructed to work.

1.3 Conclusion

In this problem if we ignore optimality, dfs might gives us faster results but they are too exorbitant plans. Heuristic with A* search gives us near dfs speed. Among heuristics, if we have low memory, we can go with levelsum as it expans least number of nodes, does least goal tests and expansions. If we need higher speed, we could go with ignore preconditions heuristic.

1.4 Optimal Plans for Problems

1.4.1 Problem 1

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Fly(P2, JFK, SFO)
4. Unload(C2, P2, SFO)
5. Fly(P1, SFO, JFK)
6. Unload(C1, P1, JFK)

1.4.2 Problem 2

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Load(C3, P3, ATL)
4. Fly(P2, JFK, SFO)
5. Unload(C2, P2, SFO)
6. Fly(P1, SFO, JFK)
7. Unload(C1, P1, JFK)
8. Fly(P3, ATL, SFO)
9. Unload(C3, P3, SFO)

1.4.3 Problem 3

1. Load(C1, P1, SFO)
2. Fly(P1, SFO, ATL)
3. Load(C3, P1, ATL)
4. Fly(P1, ATL, JFK)
5. Load(C2, P1, JFK)
6. Unload(C1, P1, JFK)

7. Unload(C3, P1, JFK)
8. Fly(P1, JFK, ORD)
9. Load(C4, P1, ORD)
10. Fly(P1, ORD, SFO)
11. Unload(C2, P1, SFO)
12. Unload(C4, P1, SFO)

Footnotes:

[1] Artificial Intelligence: A Modern Approach 3ed, p82

Author: Khurram Baig
Created: 2017-04-26 Wed 21:54
Emacs 25.1.1 (Org mode 8.2.10)