

Heuristic Analysis

- Author: Xin Chen
- Email: Bismarrck@me.com

1. Statistics

The statistics of `run_search` .

- Hardware: Macbook Pro mid 2015
- CPU: Intel i7-4850HQ

1.1 Problem 1

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Time (s)
<code>breadth_first_search</code>	43	56	180	6	0.086
<code>depth_first_graph_search</code>	22	22	84	20	0.012
<code>uniform_cost_search</code>	55	57	224	6	0.071
<code>A* h_1</code>	55	57	224	6	0.054
<code>A* h_ignore_preconditions</code>	41	43	170	6	0.049
<code>A* h_pg_levelsum</code>	11	13	50	6	0.776

1.2 Problem 2

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Time (s)
breadth_first_search	3399	4667	31027	9	4.619
depth_first_graph_search	363	364	3280	357	0.354
uniform_cost_search	4852	4854	44030	9	4.701
A* h_1	4852	4854	44030	9	4.683
A* h_ignore_preconditions	1450	1452	13303	9	2.858
A* h_pg_levelsum	86	88	841	9	18.941

1.3 Problem 3

Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Time (s)
breadth_first_search	414663	18098	129631	12	21.108
depth_first_graph_search	408	409	3364	392	0.508
uniform_cost_search	18223	18225	159618	12	19.931
A* h_1	18223	18225	159618	12	18.278
A* h_ignore_preconditions	5040	5042	44944	12	9.196
A* h_pg_levelsum	316	318	2912	12	82.753

2. Comparisons

- Depth-First Graph Search is the fastest but it is difficult to find the optimal plan.
- The costs of Breadth-First Search , Uniform Cost Search and A* h_1 are almost the same. All of these algorithms can find the optimal plan.
- Compared with BFS, DFS explores far less nodes. This may explain why DFS costs much less time but can not gurantee the optimality.
- A* h_ignore_preconditions is the fastest among all algorithms that can find the optimal plan.
- A* h_pg_levelsum is the slowest algorithm, though it requires the least nodes and goal tests.

So for these problems, I suggest A* h_ignore_preconditions should be the best.

3. Results

This section shows the optimal plans.

3.1 Problem 1

breadth_first_search	depth_first_graph_search	uniform_cost_search
Load(C1, P1, SFO)	Not Optimal	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Too many to display	Load(C2, P2, JFK)
Fly(P2, JFK, SFO)		Fly(P1, SFO, JFK)
Unload(C2, P2, SFO)		Fly(P2, JFK, SFO)
Fly(P1, SFO, JFK)		Unload(C1, P1, JFK)
Unload(C1, P1, JFK)		Unload(C2, P2, SFO)
A* h_1	A* h_ignore_preconditions	A* h_pg_levelsum
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Fly(P1, SFO, JFK)	Fly(P1, SFO, JFK)
Fly(P1, SFO, JFK)	Unload(C1, P1, JFK)	Load(C2, P2, JFK)
Fly(P2, JFK, SFO)	Load(C2, P2, JFK)	Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)
Unload(C2, P2, SFO)	Unload(C2, P2, SFO)	Unload(C2, P2, SFO)

3.2 Problem 2

breadth_first_search	depth_first_graph_search	uniform_cost_search
Load(C1, P1, SFO)	Not Optimal	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Too many to display	Load(C2, P2, JFK)
Load(C3, P3, ATL)		Load(C3, P3, ATL)
Fly(P2, JFK, SFO)		Fly(P1, SFO, JFK)
Unload(C2, P2, SFO)		Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)		Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)		Unload(C3, P3, SFO)
Fly(P1, SFO, JFK)		Unload(C2, P2, SFO)
Unload(C1, P1, JFK)		Unload(C1, P1, JFK)
A* h_1	A* h_ignore_preconditions	A* h_pg_levelsum
Load(C1, P1, SFO)	Load(C3, P3, ATL)	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Fly(P3, ATL, SFO)	Fly(P1, SFO, JFK)
Load(C3, P3, ATL)	Unload(C3, P3, SFO)	Load(C2, P2, JFK)
Fly(P1, SFO, JFK)	Load(C2, P2, JFK)	Fly(P2, JFK, SFO)
Fly(P2, JFK, SFO)	Fly(P2, JFK, SFO)	Load(C3, P3, ATL)
Fly(P3, ATL, SFO)	Unload(C2, P2, SFO)	Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)	Load(C1, P1, SFO)	Unload(C3, P3, SFO)
Unload(C2, P2, SFO)	Fly(P1, SFO, JFK)	Unload(C2, P2, SFO)
Unload(C1, P1, JFK)	Unload(C1, P1, JFK)	Unload(C1, P1, JFK)

3.3 Problem 3

breadth_first_search	depth_first_graph_search	uniform_cost_search
Load(C1, P1, SFO)	Not Optimal	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Too many to display	Load(C2, P2, JFK)
Fly(P2, JFK, ORD)		Fly(P1, SFO, ATL)
Load(C4, P2, ORD)		Load(C3, P1, ATL)
Fly(P1, SFO, ATL)		Fly(P2, JFK, ORD)
Load(C3, P1, ATL)		Load(C4, P2, ORD)
Fly(P1, ATL, JFK)		Fly(P2, ORD, SFO)
Unload(C1, P1, JFK)		Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)		Unload(C4, P2, SFO)
Fly(P2, ORD, SFO)		Unload(C3, P1, JFK)
Unload(C2, P2, SFO)		Unload(C2, P2, SFO)
Unload(C4, P2, SFO)		Unload(C1, P1, JFK)
A* h_1	A* h_ignore_preconditions	A* h_pg_levelsum
Load(C1, P1, SFO)	Load(C2, P2, JFK)	Load(C2, P2, JFK)
Load(C2, P2, JFK)	Fly(P2, JFK, ORD)	Fly(P2, JFK, ORD)
Fly(P1, SFO, ATL)	Load(C4, P2, ORD)	Load(C4, P2, ORD)
Load(C3, P1, ATL)	Fly(P2, ORD, SFO)	Fly(P2, ORD, SFO)
Fly(P2, JFK, ORD)	Unload(C4, P2, SFO)	Load(C1, P1, SFO)
Load(C4, P2, ORD)	Load(C1, P1, SFO)	Fly(P1, SFO, ATL)
Fly(P2, ORD, SFO)	Fly(P1, SFO, ATL)	Load(C3, P1, ATL)
Fly(P1, ATL, JFK)	Load(C3, P1, ATL)	Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)	Fly(P1, ATL, JFK)	Unload(C4, P2, SFO)
Unload(C3, P1, JFK)	Unload(C3, P1, JFK)	Unload(C3, P1, JFK)
Unload(C2, P2, SFO)	Unload(C2, P2, SFO)	Unload(C2, P2, SFO)
Unload(C1, P1, JFK)	Unload(C1, P1, JFK)	Unload(C1, P1, JFK)

