Joshua Krause
A.I. Nanodegree
February 4, 2018

Planning Performance Comparison

UNINFORMED PLANNING

Initial tests using uninformed planning to solve the airport problem reveal some issues with Depth-first search. Depth-first search only explores one path through the problem, which makes it faster than the other algorithms.[1] Unfortunately the plan it returns is not optimal. Cargos and airplanes bounce back and forth between airports until all goals are met. Whereas other planning algorithms solve the problems in six, nine, and twelve steps respectively, Depth-first search requires 20, 619, and 392 steps. The modest gain in computing speed is overshadowed by the incredible cost of manpower and jet fuel required to execute these plans.

PROBLEM 1	Expansions	Goal Tests	New Nodes	Length	Time
Breadth first search	43	56	180	6	0.015
Depth first graph search	21	22	84	20	0.007
Uniform cost search	55	57	224	6	0.018

PROBLEM 2	Expansions	Goal Tests	New Nodes	Length	Time
Breadth first search	3343	4609	30509	9	3.969
Depth first graph search	624	625	5602	619	1.652
Uniform cost search	4853	4855	44041	9	5.657

PROBLEM 3	Expansions	Goal Tests	New Nodes	Length	Time
Breadth first search	14663	18098	129631	12	20.342
Depth first graph search	408	409	3364	392	0.88
Uniform cost search	18223	18225	159618	12	25.32

In this problem, the step costs between levels is the same, so Uniform-cost search is functionally similar to Breadth-first search. However, Breadth-first maintains an advantage because it stops expanding as soon as a goal is found, whereas Uniform-cost continues searching for lower costs.[2] On the third problem, this results in an additional 30,000 unnecessary new nodes with a cost of an extra five seconds.

---

[1] Russell, Stuart and Norvig, Peter, *Artificial Intelligence: A Modern Approach*, 3rd Edition, (Pearson: 2009), 87

[2] Russell and Norvig, *Artificial Intelligence: A Modern Approach*, 85

HEURISTIC SEARCHING

Applying A* searching with the H1 heuristic to the problem did not result in any immediate benefits. Because the H1 heuristic is applying a constant to each level, it is functionally the same as a Uniform-cost search.[3] Therefore, it is not surprising that the results are nearly identical.

PROBLEM 1	Expansions	Goal Tests	New Nodes	Length	Time
H1	55	57	224	6	0.018
Ignore Precond	41	43	170	6	0.019
Levelsum	39	41	158	6	0.327

PROBLEM 2	Expansions	Goal Tests	New Nodes	Length	Time
H1	4853	4855	44041	9	5.739
Ignore Precond	1450	1452	13303	9	2.05
Levelsum	1129	1131	10232	9	113.672

PROBLEM 3	Expansions	Goal Tests	New Nodes	Length	Time
H1	18223	18225	159618	12	25.148
Ignore Precond	5040	5042	44944	12	8.205
Levelsum	206	2028	17933	12	402.439

Ignoring preconditions was clearly the fastest and best choice out of the tested search heuristics. It returned an efficient plan while expanding over a third less than basic A* searching. Creating a relaxed version of the airport problem makes it easier to solve.

The level-sum heuristic is an interesting study. It expands the fewest number of nodes, thus finding the most accurate solution out of all of the tested algorithms. Unfortunately looping through each level to determine its score is expensive. This heuristic took eight minutes to return a solution, whereas Uniform-cost search, the second slowest technique, took 25 seconds. Barring a more efficient use of code, this heuristic appears to be impractical for this problem.

[3] Russell and Norvig, *Artificial Intelligence: A Modern Approach,* 93

OPTIMAL PLANS

Problem 1
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)

Problem 2
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
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)