Chad Iverson

Artificial Intelligence Nanodegree Program

February 1, 2018

Analysis of Search and Results and Heuristics

Planning search experiments were conducted with three separate air cargo problems. Results are listed below followed by a brief discussion. Various search algorithms and heuristics are compared and contrasted, the best heuristic is selected, and an explanation is provided. An optimal solution was found for each of the three problems with plan lengths of 6, 9, and 12.

Optimal Solutions

Problem 1	**Problem 2**	**Problem 3**
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C1, P1, SFO)
Fly(P1, SFO, JFK)	Fly(P1, SFO, JFK)	Fly(P1, SFO, ATL)
Unload(C1, P1, JFK)	Unload(C1, P1, JFK)	Load(C3, P1, ATL)
Load(C2, P2, JFK)	Load(C2, P2, JFK)	Fly(P1, ATL, JFK)
Fly(P2, JFK, SFO)	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)
Unload(C2, P2, SFO)	Unload(C2, P2, SFO)	Load(C2, P2, JFK)
	Load(C3, P3, ATL)	Fly(P2, JFK, ORD)
	Fly(P3, ATL, SFO)	Load(C4, P2, ORD)
	Unload(C3, P3, SFO)	Fly(P2, ORD, SFO)
		Unload(C2, P2, SFO)
		Unload(C3, P1, JFK)
		Unload(C4, P2, SFO)

Problem 1

Search Algorithm	New Nodes	Expansions	Goal Tests	Time Elapsed	Plan Length
Breadth First	180	43	56	0.037145706825	6
Breadth First Tree	5960	1458	1459	1.015700467862	6
Depth First Graph	48	12	13	0.007323919795	12
Depth Limited Search	414	101	271	0.077881430276	50
Uniform Cost	224	55	57	0.032873608172	6
Recursive Best First	17029	4229	4230	2.613217785023	6
Greedy Best First Graph	28	7	9	0.005002802238	6
A*	224	55	57	0.035432716832	6
A* Ignore Preconditions	170	41	43	0.038264886476	6
A* Planning Graph Level Sum	50	11	13	4.130283502862	6

Problem 2

Search Algorithm	New Nodes	Expansions	Goal Tests	Time Elapsed	Plan Length
Breadth First	30509	3343	4609	8.077382829972	9
Breadth First Tree	N/A	N/A	N/A	N/A	N/A
Depth First Graph	5211	582	583	3.302069066092	575
Depth Limited Search	2054119	222719	2053741	989.437869959511	50
Uniform Cost	43381	4780	4782	12.466046800837	9
Recursive Best First	N/A	N/A	N/A	N/A	N/A
Greedy Best First Graph	5382	598	600	1.534111144952	21
A*	43381	4780	4782	12.231280655600	9
A* Ignore Preconditions	13303	1450	1452	4.305903484114	9
A* Planning Graph Level Sum	841	86	88	487.203487392515	9

Problem 3

Search Algorithm	New Nodes	Expansions	Goal Tests	Time Elapsed	Plan Length
Breadth First	129631	14663	18098	45.471069600433	12
Breadth First Tree	N/A	N/A	N/A	N/A	N/A
Depth First Graph	5176	627	628	3.238938299008	596
Depth Limited Search	N/A	N/A	N/A	N/A	N/A
Uniform Cost	156769	17882	17884	54.557302473113	12
Recursive Best First	N/A	N/A	N/A	N/A	N/A
Greedy Best First Graph	39970	4498	4500	12.576589633711	26
A*	156769	17882	17884	53.434710237198	12
A* Ignore Preconditions	44886	5034	5036	16.510180269368	12
A* Planning Graph Level Sum	2894	314	316	2601.619759829710	12

The results for uninformed non-heuristic search include two types of breadth-first search, two types of depth-first search, and a uniform cost search. Both types of depth-first search find an optimal solution, as does uniform cost. On the other hand, both versions of depth-first search fail to find an optimal solution. As search complexity increases, strengths and weaknesses of each algorithm are revealed. For example, breadth-first and uniform cost have the lowest and similar elapsed times for the first problem, while diverging significantly with problems two and three.

In the experiment, breadth-first tree search showed early indication of scaling issues, as did depth limited search. With the first problem, breadth-first tree search has an elapsed time surpassing all of the other searches by more than ten times. Breadth-first tree search also creates more nodes than any other uninformed search, more than 14 and 24 times the number of nodes when compared to depth limited search and uniform cost search respectively. While breadth-first tree search finds an optimal solution for the first problem, it runs somewhere past 45 minutes with the second problem. The long running time is likely because breadth-first tree search is not keeping track of explored nodes. Instead, breadth-first tree search produces a new node, expands, and goal tests states that have already been explored. Depth-limited search exhibits this same explosion of new nodes for different reasons, with the second problem, and ultimately fails to solve problem three after running for more than 45 minutes.

Overall, the uninformed non-heuristic search with the best time is depth-first search. Unfortunately, the depth-first search produces plans that are not optimal and that approximate the depth of the search tree in length. Breadth-first search and uniform cost search both produce optimal solutions for all three problems. However, breadth-first has better timings making it the best of the uninformed searches for the given problems. According to Russell, Norvig, Canny, Malik, and Edwards (2009), uniform cost search can find the optimal solution when the step costs are considered. With these cargo problems, all costs are equal, making that optimization unnecessary. The uniform cost search exploring for lower cost paths leads to more nodes being created and expanded when compared to breadth-first search, as expected. The downside of breadth-first, like all uninformed search algorithms, is that it has no concept of path cost relative to the problem goal.

Informed search algorithms take into account the problem and the path cost relative to the goal. Recursive and best-first search rely exclusively on the cost of a path from the present node to the problem goal. A* search calculates the total cost from start to goal with help from the heuristic function. The heuristic function is responsible for estimating path cost for the segment between the present node and problem goal. Results for informed heuristic search include recursive and greedy best-first search, A* search, A* ignoring preconditions, and A* planning graph level sum.

Recursive and greedy-best first search both have issues. Greedy-best first search has competitive timings and appears to be optimal for the first problem, but reveals its true self with problem two, where it fails to find an optimal path. Recursive-best first search showed early indications of trouble with an explosion of new node creation, almost 3 times breadth first search. Not surprisingly, recursive-best first search runs somewhere past 45 minutes with the second problem. A* with no heuristic function performed identical to the unweighted uniform cost, as expected. The experiment did not uncover any advantage to informed search until planning heuristic functions were introduced.

While all of the A* algorithms, with admissible heuristic functions are optimal, A* search with ignore preconditions produced the best results in this experiment. A* search with ignore preconditions consistently returns an optimal solution in a fraction of the time it takes for A* with level sum to do the same. The ignore preconditions heuristic has an edge on the level sum heuristic because it uses a factored representation of the planning problem. A factored representation allows heuristic functions to work with a relaxed version of the problem at hand.

Factored representations support derivation of flexible and domain-independent heuristics. Planning heuristics that leverage a factored representation and an action schema, like ignore preconditions, can consider more actions per level when compared to heuristics using an atomic problem representation (Russell, et al., 2009). For the second and third problems, A* search with ignore preconditions has the best timings among all informed and uninformed searches returning an optimal solution, making it the most promising algorithms of this experiment.

References

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (2003). Artificial intelligence: a modern approach (Vol. 2, No. 9). Upper Saddle River: Prentice hall.