Heuristic Analysis – Project 3

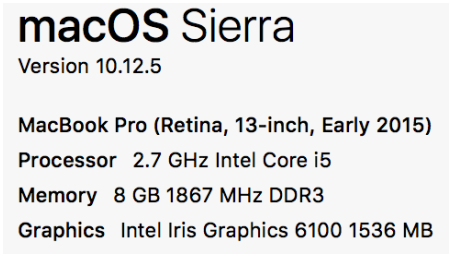
By Suresh Ooty

Summary

It was observed that the search algorithms identified different plans for the same problem, with consistent number of expansion, goal tests, new nodes and plan length in multiple trials. The Elapsed Time varied based on the computer's memory usage at the time of execution. It was generally observed that '*Breadth First Tree Search'*, '*Depth Limited search*' and '*Recursive* best *first search h_1*' did not converge after 10 mins (tested some of them for more than few hours) for the problems 2 and 3.

For each problem, the 'Correct'-ness value (in performance tables) is used to specify if the identified plan was with 'shortest' plan length possible. (Every plan identified is correct in terms achieving the goals, even if they are not shortest).

Hardware used:

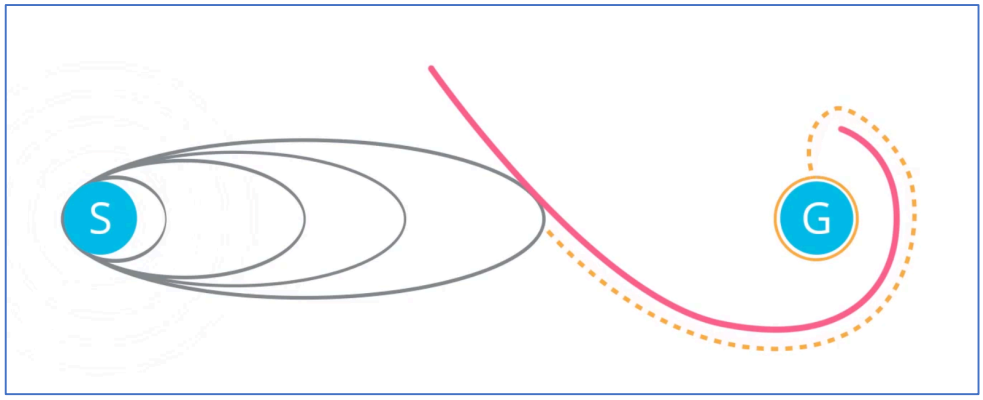


Air Cargo Problem 1

The tables below capture the performance of search algorithms and different plans identified by these search algorithms.

Greedy Best First Search:

For this problem, the greedy best first search finds the optimal path with least expansion of nodes, goal tests & new nodes in shortest time. This fits in to the schematic explanation of the same, given by Peter Norvig.



The A* star search (f = g + h) expanded more nodes ('g' – path.cost) than the greedy best first search and the 'h' distance gradually minimized from 'h_1' to 'h_ignore_precondition' and to 'h_pg_levelsum'. Evidentally, such behavior was noticed only in 'Problem 1' and could have been caused by 2 values of cargo, plane & airports utilizing all resources from source to destination, i.e., achieving the goal with some sort of an equilibrium.

It makes one to wonder how the results would be if the problem is defined when introduced with an 'equilibrium', would the A* star search performs best? (For e.g., if there is one more airport (say 'A3') in Air Cargo Problem 1 and 2 cargos reach 2 of the airports using 2 of the airplanes leaving the third Airport ('A3') with no cargo at the end.)

An observation on A* star search with level sum: The plan 'E' is in a fashion (same as plan 'D' for problem 2 & 3), such that the unload actions are done at the end of the plan, some of sort of behavior noticed in all the 3 problems. It could be that A* star search keeps a check on minimum distance to goal 'h' {keeps focused on goal as stated in course} and hence the end literals are arranged in this fashion.

Poor Performing searches:

'Breadth first tree search' & 'Recursive best first search h_1' were poor performers, but they did find a shortest plan.

Bad searches:

'Depth first graph search' & 'Depth limited search' did not find shortest plans.

Performance details on Air Cargo Problem 1:

Search	*Expansion*	*Goal Tests*	*New Nodes*	*Time elapsed (sec)*	*Plan Length*	*Correct? (Plan #)*
Breadth first search	43	56	180	0.0359	6	Y (A)
Breadth first tree search	1458	1459	5960	0.9767	6	Y (A)
Depth first graph search	21	21	84	0.0167	20	N
Depth limited search	101	271	414	0.1038	50	N
Uniform cost search	55	57	224	0.0439	6	Y (B)
Recursive best first search h_1	4229	4230	17023	**3.0488**	6	Y (C)
Greedy best first graph search h_1	7	9	28	0.0056	6	Y (B)
A star search h_1	55	57	224	0.0489	6	Y (B)
A star search h_ignore precondition	41	43	170	0.0461	6	Y (D)
A star search h_pg levelsum	11	13	50	1.0734	6	Y (E)

Different Plans identified by Search Algorithms:

Plan A	Plan B	Plan C	Plan D	Plan E
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C2, P2, JFK)	Load(C1, P1, SFO)	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Load(C2, P2, JFK)	Load(C1, P1, SFO)	Fly(P1, SFO, JFK)	Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)	Fly(P1, SFO, JFK)	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)	Load(C2, P2, JFK)
Unload(C2, P2, SFO)	Fly(P2, JFK, SFO)	Unload(C2, P2, SFO)	Load(C2, P2, JFK)	Fly(P2, JFK, SFO)
Fly(P1, SFO, JFK)	Unload(C1, P1, JFK)	Fly(P1, SFO, JFK)	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)
Unload(C1, P1, JFK)	Unload(C2, P2, SFO)	Unload(C1, P1, JFK)	Unload(C2, P2, SFO)	Unload(C2, P2, SFO)

Air Cargo Problem 2

The second problem is of one such 'equilibrium' (may be!!?) as one of the airports do not have an 'unload' of any cargo. The results are different partly caused by the increased dimensions (3 values) on airports, cargo & planes as the number of nodes need to searched will be higher than previous problem.

The expected number of literals in the plan is 9; the shortest plan achieved with fewer expansions is by 'A* search h_pg_levelsum' and the same was achieved quickly by 'A* star

h_ignore preconditions'. When the preconditions are ignored, there will be minimum number of actions constructed through the graph and hence the better performance.

An observation on A* search h_ignore preconditions: The identified plans 'D' In problem 1 & 'C' in problem 2 seem to follow a trend of sequentially calling the actions 'load', 'fly' & 'unload' for one cargo at a time. Or in other words when number of cargos = number of planes operated., or in more abstract way when n-1 dimensions have equal number of values in a n dimension problem. {x, x, x, not x}, where x is the number of values. Note that, this is behavior was observed in problem 3, as the problem does not meet the above described pattern.

Poor Performing searches:
'Breadth first tree search' & 'Recursive best first search h_1' did not converge after running for more than 10 minutes (some of them ran more than an hour) and hence captured as poorly performing searching methods.

Bad searches:
'Depth first graph search', 'Depth limited search' & 'Greedy best first search h_1' did not find a shortest plan. The depth searches expanded went deep and did not find the shortest plans. The Greedy search might have circled around the goal for a long distance before reaching the goal.

Performance details on Air Cargo Problem 2:

Search	*Expansion*	*Goal Tests*	*New Nodes*	*Time elapsed (sec)*	*Plan Length*	*Correct (Plan #)*
Breadth first search	3343	4609	30509	15.2010	9	Y (A)
Breadth first tree search				>10 min		
Depth first graph search	624	625	5602	4.3481	619	N
Depth limited search	222719	2053741	2054119	~18 mins	50	N
Uniform cost search	4830	4832	43831	14.8408	9	Y (B)
Recursive best first search h_1				>10min		
Greedy best first graph search h_1	835	837	7508	2.3271	29	N
A star search h_1	4830	4832	43831	13.3032	9	Y (B)

A star search h_ignore precondition	1428	1430	13090	4.7118	9	Y (C)
A star search h_pg levelsum	84	86	821	117.6930	9	Y (D)

Different Plans identified by Search Algorithms:

Plan A	Plan B	Plan C	Plan D
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C3, P3, ATL)	Load(C1, P1, SFO)
Load(C2, P2, JFK)	Load(C2, P2, JFK)	Fly(P3, ATL, SFO)	Fly(P1, SFO, JFK)
Load(C3, P3, ATL)	Load(C3, P3, ATL)	Unload(C3, P3, SFO)	Load(C2, P2, JFK)
Fly(P2, JFK, SFO)	Fly(P1, SFO, JFK)	Load(C1, P1, SFO)	Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)	Fly(P2, JFK, SFO)	Fly(P1, SFO, JFK)	Load(C3, P3, ATL)
Fly(P1, SFO, JFK)	Fly(P3, ATL, SFO)	Unload(C1, P1, JFK)	Fly(P3, ATL, SFO)
Unload(C1, P1, JFK)	Unload(C3, P3, SFO)	Load(C2, P2, JFK)	Unload(C3, P3, SFO)
Fly(P3, ATL, SFO)	Unload(C1, P1, JFK)	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)
Unload(C3, P3, SFO)	Unload(C2, P2, SFO)	Unload(C2, P2, SFO)	Unload(C2, P2, SFO)

Air Cargo Problem 3

The 'A star search h_pg_levelsum' expanded few nodes and found the shortest plan. The identified plan follows the behavior described for this search in previous sections. (all 'unloads' at the end of the plan)

And the better performing search in terms of time consumption is A* star with ignored preconditions.

Poor Performance searches: Breadth first tree search', Depth limited search' & 'Recursive best first search h_1' did not converge after running for more than 10 minutes

Bad Searches: 'Depth first graph search' & 'Greedy best first graph search h_1' did not find the shortest the plans. In the increased dimensions the greedy search gets in to a longer search than when the dimensions are reasonable.

Performance details on Air Cargo Problem 3:

Search	*Expansion*	*Goal Tests*	*New Nodes*	*Time elapsed (sec)*	*Plan Length*	*Correct? (Plan #)*
Breadth first search	14663	18098	129631	126.8924	12	Y (A)
Breadth first tree search				>10 min		
Depth first graph search	408	409	3364	2.2041	392	N
Depth limited search				>10 min		
Uniform cost search	18233	18235	159697	63.7006	12	Y (B)
Recursive best first search h_1				>10 min		
Greedy best first graph search h_1	5165	5167	45511	17.17730	22	N
A star search h_1	18233	18235	159697	57.7713	12	Y(B)
A star search h_ignore precondition	4951	4953	44051	18.1674	12	Y(C)
A star search h_pg levelsum	316	318	2926	603.0395	12	Y(D)

Different Plans identified by Search Algorithms:

Plan A	Plan B	Plan C	Plan D
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C2, P2, JFK)	Load(C2, P2, JFK)
Load(C2, P2, JFK)	Load(C2, P2, JFK)	Fly(P2, JFK, ORD)	Fly(P2, JFK, ORD)
Fly(P2, JFK, ORD)	Fly(P1, SFO, ATL)	Load(C4, P2, ORD)	Load(C4, P2, ORD)
Load(C4, P2, ORD)	Load(C3, P1, ATL)	Fly(P2, ORD, SFO)	Fly(P2, ORD, SFO)
Fly(P1, SFO, ATL)	Fly(P2, JFK, ORD)	Unload(C4, P2, SFO)	Load(C1, P1, SFO)
Load(C3, P1, ATL)	Load(C4, P2, ORD)	Load(C1, P1, SFO)	Fly(P1, SFO, ATL)
Fly(P1, ATL, JFK)	Fly(P2, ORD, SFO)	Fly(P1, SFO, ATL)	Load(C3, P1, ATL)
Unload(C1, P1, JFK)	Unload(C4, P2, SFO)	Load(C3, P1, ATL)	Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)	Fly(P1, ATL, JFK)	Fly(P1, ATL, JFK)	Unload(C4, P2, SFO)
Fly(P2, ORD, 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(C2, P2, SFO)
Unload(C4, P2, SFO)	Unload(C1, P1, JFK)	Unload(C1, P1, JFK)	Unload(C1, P1, JFK)