Heuristic analysis

By Daniel Yovchev

Problem 1

Initial state and goal	*Optimal Plan*

Init(At(C1, SFO) ∧ At(C2, JFK)
∧ At(P1, SFO) ∧ At(P2, JFK)
∧ Cargo(C1) ∧ Cargo(C2)
∧ Plane(P1) ∧ Plane(P2)
∧ Airport(JFK) ∧ Airport(SFO))
Goal(At(C1, JFK) ∧ At(C2, SFO))
	Load(C1, P1, SFO)
	Fly(P1, SFO, JFK)
	Unload(C1, P1, JFK)
	Load(C2, P2, JFK)
	Fly(P2, JFK, SFO)
	Unload(C2, P2, SFO)

#	*Algorithm*	*Exp*	*Goal Tests*	*New Nodes*	*Plan Length*	*Time*
1	***Breadth first search***	43	56	180	6	0.0312
2	***Breadth first tree search***	1458	1459	5960	6	0.9199
3	***Depth first graph search***	21	22	84	20	0.0143
4	***Depth limited search***	101	271	414	50	0.0803
5	***Uniform cost search***	55	57	224	6	0.0343
6	***Recursive best first search with h_1***	4229	4230	17023	6	2.5464
7	***Greedy best first graph search with h_1***	7	9	28	6	0.0052
8	***A star search with h_1***	55	57	224	6	0.0347
9	***A star search with h_ignore_preconditions***	41	43	170	6	0.0343
10	***A star search with h_pg_levelsum***	11	13	50	6	2.1201

All the algorithms performed very well on this problem the best algorithm for this problem is the ***"Greedy best first graph search with h_1"*** it found the optimal plan with the fastest time and the fewest expansions and goal tests. Only 2 algorithms failed to produce the optimal solution but let bare in mind that this problem have a very small state space.

Problem 2

Initial state and goal	*Optimal Plan*

Init(At(C1, SFO) ∧ At(C2, JFK)
∧ At(C3, ATL) ∧ At(P1, SFO)
∧ At(P2, JFK) ∧ At(P3, ATL)
∧ Cargo(C1) ∧ Cargo(C2)
∧ Cargo(C3) ∧ Plane(P1)
∧ Plane(P2) ∧ Plane(P3)
∧ Airport(JFK) ∧ Airport(SFO)
∧ Airport(ATL))
Goal(At(C1, JFK) ∧ At(C2, SFO)
∧ At(C3, SFO))
	Load(C3, P3, ATL)
	Fly(P3, ATL, SFO)
	Unload(C3, P3, SFO)
	Load(C1, P1, SFO)
	Fly(P1, SFO, JFK)
	Unload(C1, P1, JFK)
	Load(C2, P2, JFK)
	Fly(P2, JFK, SFO)
	Unload(C2, P2, SFO)

#	*Algorithm*	*Exp*	*Goal Tests*	*New Nodes*	*Plan Length*	*Time*
1	*Breadth first search*	3343	4609	30509	9	11.700
2	*Breadth first tree search*	N/A	N/A	N/A	N/A	N/A
3	*Depth first graph search*	624	625	5602	619	2.8568
4	*Depth limited search*	222719	2053741	2054119	50	774.30
5	*Uniform cost search*	4852	4854	44030	9	10.301
6	*Recursive best first search with h_1*	N/A	N/A	N/A	N/A	N/A
7	*Greedy best first graph search with h_1*	990	992	8910	17	2.0696
8	*A star search with h_1*	4852	4854	44030	9	10.574
9	*A star search with h_ignore_preconditions*	1450	1452	13303	9	3.7384
10	*A star search with h_pg_levelsum*	86	88	841	9	363.98

In problem 2 we can see some of the algorithms started to fail by not producing optimal solution or not giving any result in a reasonable time. The best algorithm based on the speed and the optimal result is ***"A star search with h_ignore_preconditions".*** The version of the algorithm using the ***"h_pg_levelsum"*** produced optimal solution with the least expansions, goal tested and new nodes but because this heuristic is too expensive it took a bit more than 6 minutes to terminate.

Problem 3

Initial state and goal	*Optimal Plan*

Init(At(C1, SFO) ∧ At(C2, JFK)
∧ At(C3, ATL) ∧ At(C4, ORD)
∧ At(P1, SFO) ∧ At(P2, JFK)
∧ Cargo(C1) ∧ Cargo(C2)
∧ Cargo(C3) ∧ Cargo(C4)
∧ Plane(P1) ∧ Plane(P2)
∧ Airport(JFK) ∧ Airport(SFO)
∧ Airport(ATL) ∧ Airport(ORD))
Goal(At(C1, JFK) ∧ At(C3, JFK)
∧ At(C2, SFO) ∧ At(C4, SFO))
	Load(C2, P2, JFK)
	Fly(P2, JFK, ORD)
	Load(C4, P2, ORD)
	Fly(P2, ORD, SFO)
	Load(C1, P1, SFO)
	Fly(P1, SFO, ATL)
	Load(C3, P1, ATL)
	Fly(P1, ATL, JFK)
	Unload(C4, P2, SFO)
	Unload(C3, P1, JFK)
	Unload(C1, P1, JFK)
	Unload(C2, P2, SFO)

#	*Algorithm*	*Exp*	*Goal Tests*	*New Nodes*	*Plan Length*	*Time*
1	*Breadth first search*	14663	18098	129631	12	83.765
2	*Breadth first tree search*	N/A	N/A	N/A	N/A	N/A
3	*Depth first graph search*	408	409	3364	392	1.5048
4	*Depth limited search*	N/A	N/A	N/A	N/A	N/A
5	*Uniform cost search*	18235	18237	159716	12	46.300
6	*Recursive best first search with h_1*	N/A	N/A	N/A	N/A	N/A
7	*Greedy best first graph search with h_1*	5614	5616	49429	22	14.490
8	*A star search with h_1*	18235	18237	159716	12	46.603
9	*A star search with h_ignore_preconditions*	5040	5042	44944	12	15.041
10	*A star search with h_pg_levelsum*	318	320	2934	12	1895.2

In this problem, we really put the algorithms to the test. Some of the algorithms were not able to terminate in a reasonable time. We also observed that algorithm 3 and 7 was not able to produce an optimal plan. Again the ***"A star search with h_ignore_preconditions"*** produced the optimal result in the fastest time but the winner for the smallest expansions, goal tests and new nodes is the ***"A star search"*** that uses ***"h_pg_levelsum"*** heuristic as I said in the previous problem this heuristic is too expensive and it took a bit more than 30 minutes to terminate.

Examine the results

The algorithm ***“Breadth first search”*** performed very well considering the simplicity of the algorithm but as the complexity of the problem and the state space becomes bigger we can see the hit on the performance.

The algorithm ***“A star search”*** performed the best using the ***“h_ignore_preconditions”*** heuristic that why this is my winner for the 3 problems we tested. The use of ***“h_pg_levelsum”*** heuristic displayed great potential by using the smallest amount of expansions and nodes but the hit on the performance of the heuristic makes it unusable.