Ahmed Yamout
a.yamout@nu.edu.eg

Heuristic analysis

Heuristic analysis:

Problem one	h_1	h_ignore_precond	h_pg_levelsum
Expansions	55	41	55
Goal Tests	57	43	57
New Nodes	224	170	224
Plan length	6	6	6
Runtime	0.035	0.035	1.58
Optimal in comparison?	Yes	Yes	Yes

Optimal plan:

Load(C1, P1, SFO)

Load(C2, P2, JFK)

Fly(P1, SFO, JFK)

Fly(P2, JFK, SFO)

Unload(C1, P1, JFK)

Unload(C2, P2, SFO)

Comparison:

Ignore precondition heuristic expands less,does less goal tests and has less new nodes than levelsum;however, it reached the optimal plan length in less runtime than levelsum heuristic

Problem two	h_1	h_ignore_precond	h_pg_levelsum
Expansions	4853	1450	86
Goal Tests	4855	1452	88
New Nodes	44041	13303	841
Plan length	9	9	9
Runtime	11.6	4.2	25
Optimal in comparison?	Yes	Yes	Yes

Optimal plan:

Load(C2, P2, JFK)

Ahmed Yamout
a.yamout@nu.edu.eg

Load(C1, P1, SFO)

Load(C3, P3, ATL)

Fly(P2, JFK, SFO)

Unload(C2, P2, SFO)

Fly(P1, SFO, JFK)

Unload(C1, P1, JFK)

Fly(P3, ATL, SFO)

Unload(C3, P3, SFO)

Comparison:

Ignore preconditions heuristic expands more than levelsum and takes less runtime and reaches optimal length while levelsum expands less, takes more runtime. Both however do reach optimal length.

Problem Three	**h_1**	**h_ignore_precond**	**h_pg_levelsum**
Expansions	17783	5003	325
Goal Tests	17785	5005	327
New Nodes	155920	44586	3002
Plan length	12	12	12
Runtime	51.2	16.4	134
Optimal in comparison?	Yes	Yes	Yes

Optimal plan:

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)

Ahmed Yamout
a.yamout@nu.edu.eg

Unload(C1, P1, JFK)

Comparison

Ignore preconditions heuristic expajnds more than levelsum and reaches optimal solution in less runtime. Levelsum expands less,takes more time yet still reaches the optimal length.

Non-heuristic analysis:

Problem one	Breadth First Search	Breadth First Tree Search	Depth first graph search
Expansions	43	1458	12
Goal Tests	56	1459	13
New Nodes	180	5960	48
Plan Length	6	6	12
Time	0.03	0.9	0.008
Optimality	yes	yes	no

As shown in the table above, depth first search does what it is supposed to do, that is find the first solution thus we can see optimality in runtime but not in solution as it got us a path length of 12 , while breadth first tree search took the longest time to expand but got us the optimal plan length. As for breadth first search it got us the best plan length in low amount of time and expansions compared to BFTS.

Problem two	Breadth First Search	Breadth First Tree Search	Depth first graph search
Expansions	3343	-	1669
Goal Tests	4609	-	1670
New Nodes	30509	-	14863
Plan Length	9	-	1444
Time	8	600> ; cancelled	12.7
Optimality	yes	-	no

Depth first search got us a non-optimal length in a big runtime and has expanded lower amount of nodes than breadth first search thus breadth first search is better in all aspects here.

Problem three	Breadth First Search	Breadth First Tree Search	Depth first graph search
Expansions	14491	-	-
Goal Tests	17947	-	-
New Nodes	128184	-	-
Plan Length	12	-	-
Time	41	600> ; cancelled	600> ; cancelled
Optimality	yes	-	-

As shown in the table above only breadth first search was able to reach to a plan length after expanding 14491 nodes.

Ahmed Yamout
a.yamout@nu.edu.eg

Conclusion:

To sum up, the ignore condition heuristic had the edge over the other heuristics because it had a balance between good node expansion and runtime while always reaching the optimal plan length. **According to the AI degree lesson 11 , this is due to the fact that by constraining our propagation with preconditions that need to be propagated we increase runtime, while the ignore condition relaxes this constraint thus allowing the algorithm to obtain better results in less time**. The heuristic was indeed better than the non-heuristic search methods we used, it always reached optimal plan length and has less runtime than the non-heuristic ones, not to mention the fact that it expanded more nodes on problem three than the breadth first search; thus, making it the best method between heuristics and non-heuristics.

References

AI Nanodegree lesson 11