Planning Search: Analysis

Problem 1:

Optimal Plan (Greedy best first graph search algorithm)

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

Problem 1 Non-Heuristic Search Results					
Algorithm	Expansions	New Nodes	Plan Length	Time(s)	Optimal
Breadth first search	43	180	6	0.036	Yes
Breadth first tree search	1458	5960	6	1.096	Yes
Depth first graph search	21	84	20	0.021	No
Depth Limited Search	101	414	50	0.113	No
Uniform Cost Search	55	224	6	0.058	Yes
Recursive best first search	4229	17023	6	3.105	Yes
Greedy best first graph search	7	28	6	0.008	Yes

Table 1. Problem non-heuristic search result metrics.

Table 1, Provides insight into the performance of seven search algorithms. Results marked as “Yes” in Optimal columns are optimal solutions for the problem. The number of expansions and new nodes varied greatly amongst the optimal search algorithms. Greedy Best First Graph Search was the top performing algorithm for this problem. It had the lowest number of expansions and new nodes which lead to fast overall time. Depth First Graph Search also had the second lowest expansions and new nodes, and saw a fast overall time, but resulted in Plan Length of 20 which is more than double that of optimal Plan length of 6.

Problem 1 Heuristic Search Results					
Algorithm	Expansions	New Nodes	Plan Length	Time(s)	Optimal
A* search with h_1	55	224	6	0.051	Yes
A* search (ignore preconditions)	41	170	6	0.057	Yes
A* search (levelsum)	11	50	6	1.105	Yes

Table 2. Problem 1 heuristic search result metrics

Table 2, provides an insight into the performance of A* Search with 3 different heuristics. All 3 algorithms achieved the optimal plan length of 6. The level sum heuristic was the slowest of all 3, but it only expanded to 11 nodes versus the next closets of 41.

Problem 2:

Optimal Plan (A* search (ignore preconditions))
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 2 Non-Heuristic Search Results					
Algorithm	Expansions	New Nodes	Plan Length	Time(s)	Optimal
Breadth first search	3401	31049	9	9.927	Yes
Breadth first tree search	-	-	-	Timed Out	No
Depth first graph search	1192	10606	1138	10.139	No
Depth Limited Search	-	-	-	Timed Out	No
Uniform Cost Search	4761	43206	9	13.949	Yes
Recursive best first search	-	-	-	Times Out	No
Greedy best first graph search	550	4950	9	1.765	Yes

Table 3. Problem 2 non-heuristic search result metrics.

Table 3 contain the results of all seven non-heuristic search algorithms. Breadth First Tree Search, Depth Limited Search, and Recursive Best First Search timed out after 10 minutes. Greedy Best First Graph Search had the lowest time, number of expansions and new nodes. Its performance is 8 times better than Depth First Search result which did not yield the optimal plan length and 5 times better Breadth First Search

Problem 2 Heuristic Search Results					
Algorithm	Expansions	New Nodes	Plan Length	Time(s)	Optimal
A* search with h_1	4761	43206	9	12.675	Yes
A* search (ignore preconditions)	1450	13303	9	5.551	Yes
A* search (levelsum)	86	841	9	219.552	Yes

Table 4. Problem 2 heuristic search result metrics

Table 4 results for A* search with various heuristics show that levelsum only expanded to 86 and utilized only 841 new nodes. Ignore precondition heuristic expanded to 1450 with 13303 new nodes. Both heuristics achieved the optimal plan length but the ignore precondition took only 5.551 seconds vs. 219.552 seconds.

Problem 3:

Optimal Plan (Uniform Cost Search)

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P1, ATL, JFK)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 3 Non-Heuristic Search Results					
Algorithm	Expansions	New Nodes	Plan Length	Time(s)	Optimal
Breadth first search	14491	128184	12	50.350	Yes
Breadth first tree search	-	-	-	Timed Out	No

Depth first graph search	2099	17558	2014	30.150	No
Depth Limited Search	-	-	-	Timed Out	No
Uniform Cost Search	17783	155920	12	64.973	Yes
Recursive best first search	-	-	-	Timed Out	No
Greedy best first graph search	4031	35794	22	15.104	Yes

Table 5. Problem 3 non-heuristic search result metrics.

Table 5 contain the results of all seven non-heuristic search algorithms. Breadth first search(BFS) and Uniform Cost search both achieved the optimal plan length of 12. Uniform Cost's performance time was approx. 50% less than BFS. BFS had a lower number of expansions 14491 versus Uniform Cost's 17783.

Problem 3 Heuristic Search Results					
Algorithm	Expansions	New Nodes	Plan Length	Time(s)	Optimal
A* search with h_1	17783	155920	12	60.912	Yes
A* search (ignore preconditions)	5003	44586	12	21.490	Yes
A* search (levelsum)	323	2983	12	1108.312	Yes

Table 6. Problem 3 heuristic search result metrics

Conclusion

The Best Heuristic used in these problems really depends on performance and usage of memory space. If our preference is to find optimal plan quickly and don't bother about usage of memory space then we should prefer A* Search with ignore precondition. If memory space is limited and time to search optimal route is not a consideration then A* Search with level sum heuristic would be ideal.

A* search is best-first search where the cost function f(n) = g(n) + h'(n), the actual cost of the path so far (g(n)), plus the estimated cost of the path from the current node to the goal (h'(n)). A* will always find an optimal path to a goal if the heuristic function is admissible

Overall, A* search with ignore precondition is ideal planning search method because it performed equal or better than all 7 non-heuristic algorithms in all the problems. Also, ignore precondition in problem two and three had significantly less expansions and new nodes then any non-heuristic methods. Based on observation of search result, I conclude that ignore preconditions heuristic would be best to use.

Reference

http://www-cs-students.stanford.edu/~pdoyle/quail/notes/pdoyle/search.html
http://www.cs.cmu.edu/~arielpro/15780/lec/780-16.pdf