Part 1 - Planning Problems

Solutions

	Problem 1	Problem 2	Problem 3
1	Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C1, P1, SFO)
2	Load(C2, P2, JFK)	Load(C2, P2, JFK)	Load(C2, P2, JFK)
3	Fly(P1, SFO, JFK)	Load(C3, P3, ATL)	Fly(P1, SFO, ATL)
4	Unload(C1, P1, JFK)	Fly(P1, SFO, JFK)	Load(C3, P1, ATL)
5	Fly(P2, JFK, SFO)	Unload(C1, P1, JFK)	Fly(P1, ATL, JFK)
6	Unload(C2, P2, SFO)	Fly(P2, JFK, SFO)	Fly(P2, JFK, ORD)
7		Unload(C2, P2, SFO)	Load(C4, P2, ORD)
8		Fly(P3, ATL, SFO)	Fly(P2, ORD, SFO)
9		Unload(C3, P3, SFO)	Unload(C1, P1, JFK)
10			Unload(C2, P2, SFO)
11			Unload(C3, P1, JFK)
12			Unload(C4, P2, SFO)

Search methods comparison

PROBLEM 1

	BFS (1)	DFS (3)	Greedy (7)	A* (9)	A* (10)
Plan length	6	17	6	6	6
Expansions	42	19	7	41	39
Goal Tests	57	20	9	43	41
New Nodes	176	76	28	170	158
Time	0.0285	0.0116	0.0046	0.0322	0.9003

PROBLEM 2

	BFS (1)	DFS (3)	Greedy (7)	A* (9)	A* (10)
Plan length	9	456	15	9	9
Expansions	3345	457	990	1450	1129
Goal Tests	4613	458	992	1452	1131
New Nodes	30526	4128	8910	13303	10232

	BFS (1)	DFS (3)	Greedy (7)	A* (9)	A* (10)
Time	11.5570	1.8931	1.7647	3.3409	337.2031

PROBLEM 3

	BFS (1)	DFS (3)	Greedy (7)	A* (9)	A* (10)
Plan length	12	3291	22	12	Not Finished
Expansions	13667	3552	5614	5040	Not Finished
Goal Tests	17200	3553	5616	5042	Not Finished
New Nodes	120941	29564	49429	44944	Not Finished
Time	77.2006	51.4107	13.3790	13.9676	Not Finished

Conclusion

Depth First methods (3, 7) are very fast but do not guarantee an optimal path will be found. Breadth First Search guarantees optimality but the time it takes to find a solution increases exponentially with the number of possible states, this is probably made worst by the amount of memory used which is very high because the stack used to keep track of the frontier contains all the nodes at a given depth.

The heuristic methods guarantee optimality and do less search work than the other methods (comparisons, expansions, etc) but this does not necessarily reflect in faster execution time. A n expensive heuristic can result in A* taking longer time to complete or even using more memory than BFS. For example for problem 3 the A* search with a graph based heuristic didn't finish and even crashed my laptop (I suspect because of memory usage).

The good news seems to be that even a simple heuristic can result in faster execution times, in this case it is obvious that the best method overall is A* with a heuristic simply ignoring preconditions.