Problem 1

Optimal Plan

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

Uninformed Planning Algorithm Comparison

Out of the 5 uninformed search algorithms tested, Breadth First Search performed the best. Breadth First Search and Breadth First Tree Search found optimal solutions, as is expected when the path cost is a nondecreasing function of the depth. Uniform Cost Search, as a complete and optimal search algorithm given nonzero step costs, also found an optimal solution, though more nodes were expanded than Breadth First Search. Depth First Graph Search expanded very few nodes and found a solution about a factor of 2 faster than breadth first search, but the solution found was suboptimal. Depth Limited Search performed relatively poorly on all fronts.

Algorithm	Expansions	Goal Tests	Time (s)	Plan Length	Optimality
Breadth First Search	43	56	0.04	6	Optimal
Breadth First Tree Search	1458	1459	1.31	6	Optimal
Depth First Graph Search	21	22	0.02	20	Suboptimal
Depth Limited Search	101	271	0.13	50	Suboptimal
Uniform Cost Search	55	57	0.06	6	Optimal

Table 1: Uninformed Search Comparison, Problem 1

Heuristic Comparison

A* search was tested with the heuristics shown in Table 2. While A* with the ignore preconditions heuristic made less expansions and goal tests than A* with h1, it took very nearly the same amount of time, indicating that its heuristic is slightly more expensive to compute. Very few expansions were made within the planning graph, but each expansion and goal test in a planning graph was more expensive to compute than the expansions made in state-space, leading to a significantly longer computing time. A* search found the optimal plan with all 3 heuristics.

Heuristic	Expansions	Goal Tests	Time (s)	Plan Length	Optimality
h1	55	57	0.08	6	Optimal
Ignore preconditions	41	43	0.08	6	Optimal
Planning Graph Level-sum	11	13	1.93	6	Optimal

Table 2: A Search with Heuristics Comparison, Problem 1*

Problem 2

Optimal Plan

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

Uninformed Planning Algorithm Comparison

Once again, Breadth First Search and Uniform Cost Search found optimal solutions, with Breadth First Search expanding fewer nodes and taking less time than Uniform Cost Search. Depth First Graph Search again made fewer expansions and goal tests, and took less time than any other uninformed search algorithm tested, but found a suboptimal solution. Breadth First Tree Search and Depth Limited Search did not find solutions within 10 minutes.

Algorithm	Expansions	Goal Tests	Time (s)	Plan Length	Optimality
Breadth First Search	3343	4609	12.11	9	Optimal
Breadth First Tree Search			> 600		
Depth First Graph Search	624	625	7.89	619	Suboptimal
Depth Limited Search			> 600		
Uniform Cost Search	4853	4855	18.04	9	Optimal

Table 3: Uninformed Search Comparison, Problem 2

Heuristic Planning Algorithm Comparison

For this problem, A* with Ignore preconditions expanded significantly fewer nodes than A* with h1, and took less than half as much time. As before, the level-sum heuristic provided by the planning graph led to A* performing two orders of magnitude less expansions and goal tests than it did with the other heuristics, but at the cost of a lot more time taken. As all 3 heuristics are admissible, A* found optimal solutions as expected.

Heuristic	Expansions	Goal Tests	Time (s)	Plan Length	Optimality
h1	4853	4855	20.29	9	Optimal
Ignore preconditions	1450	1452	7.56	9	Optimal
Planning Graph Level-sum	86	88	325.56	9	Optimal

Table 4: A Search with Heuristics Comparison, Problem 2*

Problem 3

Optimal Plan

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

Uninformed Planning Algorithm Comparison

The uninformed algorithms performed similarly relative to each other in Problem 3 as in Problem 2. The performance gap between Breadth First Search and Uniform Cost Search was not as relatively large this time. Once again, Breadth First Tree Search and Depth Limited Search failed to find solutions within 10 minutes. Depth First Graph Search behaved as in the other problems, finding a suboptimal solution more than 20 times faster than the other algorithms.

Algorithm	Expansions	Goal Tests	Time (s)	Plan Length	Optimality
Breadth First Search	14663	18098	86.49	12	Optimal
Breadth First Tree Search			> 600		
Depth First Graph Search	408	409	3.27	392	Suboptimal
Depth Limited Search			> 600		
Uniform Cost Search	18222	18224	92.63	12	Optimal

Table 5: Uninformed Search Comparison, Problem 3

Heuristic Planning Algorithm Comparison

In problem 3, A* with the ignore preconditions heuristic expanded about 3 times fewer nodes, and ran about 3 times faster than A* with the h1 heuristic. Once again, h1 with the level-sum heuristic expanded very few nodes, but took far longer than the other two heuristic algorithms. All three heuristics allowed A* to find an optimal solution.

Heuristic	Expansions	Goal Tests	Time (s)	Plan Length	Optimality
h1	18222	18224	106.53	12	Optimal
Ignore preconditions	5040	5042	32.89	12	Optimal
Planning Graph Level-sum	316	318	1567.95	12	Optimal

Table 6: A Search with Heuristics Comparison, Problem 3*

Conclusion

The ignore preconditions heuristic performed the best on all problems. In problems 2 and 3, it found optimal solutions with significantly less expansions than the uninformed search algorithms. In problem 1, Breadth First Search and Uniform Cost Search were faster than A* with ignore preconditions, suggesting that in that smaller state space, the cost of computing the heuristic no longer was worth the savings in nodes with unpromising heuristic values left unexpanded. The results confirm the optimality of A* given admissible heuristics, and the optimality of breadth first search and uniform cost search given path costs as nondecreasing functions of depth and nonzero step costs respectively. Depth first search provided large time savings and expanded very few nodes, but failed to find an optimal solution in all three problems. That depth first graph search is complete in a finite state space was also confirmed, and that depth first tree search is incomplete, liable to find itself stuck in infinite loops was also suggested by its failure to find solutions within 10 minutes for problems 2 and 3.