Planning: Search Result Metrics and Heuristic Analysis

For this project, we built a planning search agent to solve problems in the Air Cargo domain. The optimal plans and result metrics are detailed on the following pages.

Non-Heuristic Search Results

Uninformed search (blind search) strategies like BFS, DFS, and UFS do not have any additional information about states beyond the problem definition. They just generate successor states for the current state and determine if there is a goal state among them. Our search metrics compare speed, memory usage, and optimality of solutions.

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]	No	No	Yes[a]	Yes[a,d]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes[c]	Yes	No	No	Yes[c]	Yes[c,d]

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

As supported by section 3.4.7[1] comparing uninformed search strategies, *Breadth First Search* and *Uniform Cost Search* are both complete and optimal, and they were the only uninformed searches that found the optimal path. While *Depth FIrst Search* can be faster and use less memory, the solution found is not optimal.

Heuristic Search Results

Per section 3.51[1], informed search strategies with problem-specific information beyond the problem definition can find solutions more efficiently. More specifically, A* Search is complete, optimal, and optimally efficient. While all of our heuristics find the optimal plan, the *A* Search with Ignore Preconditions Heuristic* does so the fastest. This is an admissible heuristic that drops all preconditions from actions, so every action becomes possible in every state and each goal fluent can be reached in a single step. Because this heuristic is easy to compute, it runs the fastest although it may use more memory while searching nodes. In comparison, *A* Search with Planning Graph Level Sum Heuristic* takes more execution time to set up the Planning Graph, but uses the least memory to reach the optimal solution.

[1] Russell, Stuart, and Peter Norvig. *Artificial Intelligence: A Modern Approach Third Edition*. Pearson, 2010.

Problem 1 Optimal Plan (6)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem 2 Optimal Plan (9)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem 3 Optimal Plan (12)
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)

Problem 1

Search Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Time elapsed in seconds
breadth first search	43	56	180	6	0.034269
breadth first tree search	1458	1459	5960	6	0.876022
depth first graph search	12	13	48	12	0.008531
depth limited search	101	271	414	50	0.089542
uniform cost search	55	57	224	6	0.038578
recursive best first search - constant heuristic	4229	4230	17029	6	2.942478
greedy best first graph search - constant heuristic	7	9	28	6	0.005746
A* search - constant heuristic	55	57	224	6	0.038899
A* search - ignore preconditions heuristic	41	43	170	6	0.03898
A* search - planning graph level sum heuristic	11	13	50	6	0.613392

Problem 2

Search Algorithm	Expansions	Goal Tests	New Nodes	Plan Length	Time elapsed in seconds
breadth first search	3346	4612	30534	9	13.779515
breadth first tree search					
depth first graph search	1124	1125	10017	1085	7.139091
depth limited search					
uniform cost search	4853	4855	44041	9	12.425988
recursive best first search - constant heuristic					

greedy best first graph search - constant heuristic	998	1000	8986	21	3.206396
A* search - constant heuristic	4853	4855	44041	9	13.402912
A* search - ignore preconditions heuristic	1450	1452	13303	9	4.308923
A* search - planning graph level sum heuristic	86	88	841	9	70.775124
Problem 3					
Search Algorithm	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time elapsed in seconds**
breadth first search	14120	17673	124926	12	104.684905
breadth first tree search					
depth first graph search	677	678	5608	660	3.513849
depth limited search					>600
uniform cost search	18223	18225	159618	12	54.138145
recursive best first search - constant heuristic					
greedy best first graph search - constant heuristic	5578	5580	49150	22	16.745688
A* search - constant heuristic	18223	18225	159618	12	55.512369
A* search - ignore preconditions heuristic	5040	5042	44944	12	17.654626
A* search - planning graph level sum heuristic	325	327	3002	12	289.618773