Heuristic Analysis

Optimal plans

Air cargo problem 1

The optimal plan for the problem 1 has length 6, It can be achieved with uniform cost search in 55 expansions, in 0.033 sec

1. Load(C1, P1, SFO)
2. Load(C2, P2, JFK)
3. Fly(P1, SFO, JFK)
4. Fly(P2, JFK, SFO)
5. Unload(C1, P1, JFK)
6. Unload(C2, P2, SFO)

Air cargo problem 2

The optimal plan for problem 2 has length 9, it can be achieved with A* search with ignored preconditions heuristic in 1450 expansions, in 4.552 sec

1. Load(C3, P3, ATL)
2. Fly(P3, ATL, SFO)
3. Unload(C3, P3, SFO)
4. Load(C1, P1, SFO)
5. Fly(P1, SFO, JFK)
6. Unload(C1, P1, JFK)
7. Load(C2, P2, JFK)
8. Fly(P2, JFK, SFO)
9. Unload(C2, P2, SFO)

Air cargo problem 3

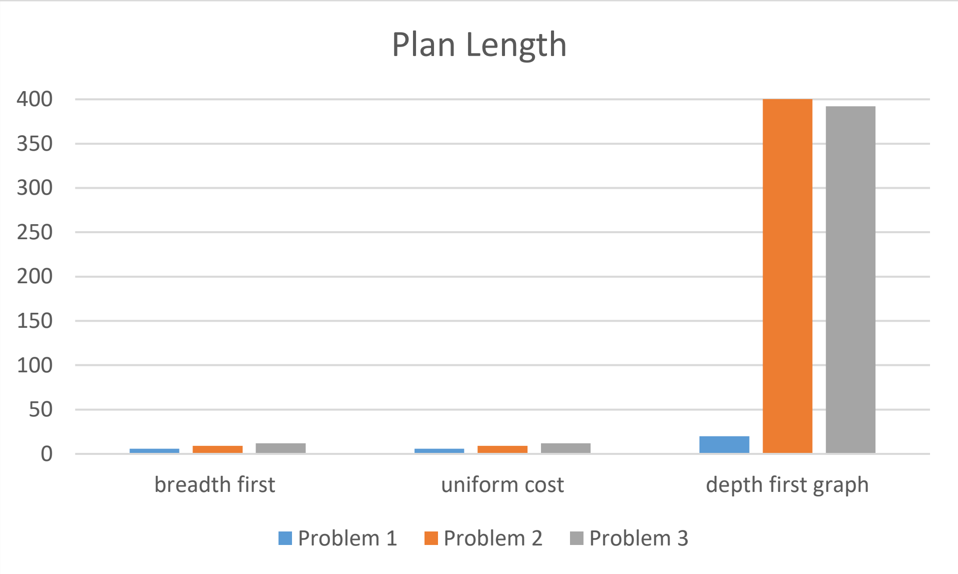
The optimal plan for problem 3 has length 12, it can be achieved with achieved with A* search with ignored preconditions heuristic in 5040 expansions, in 16.818 sec

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

Non-Heuristic Search: Breadth First, Uniform Cost and Depth First Graph Comparison

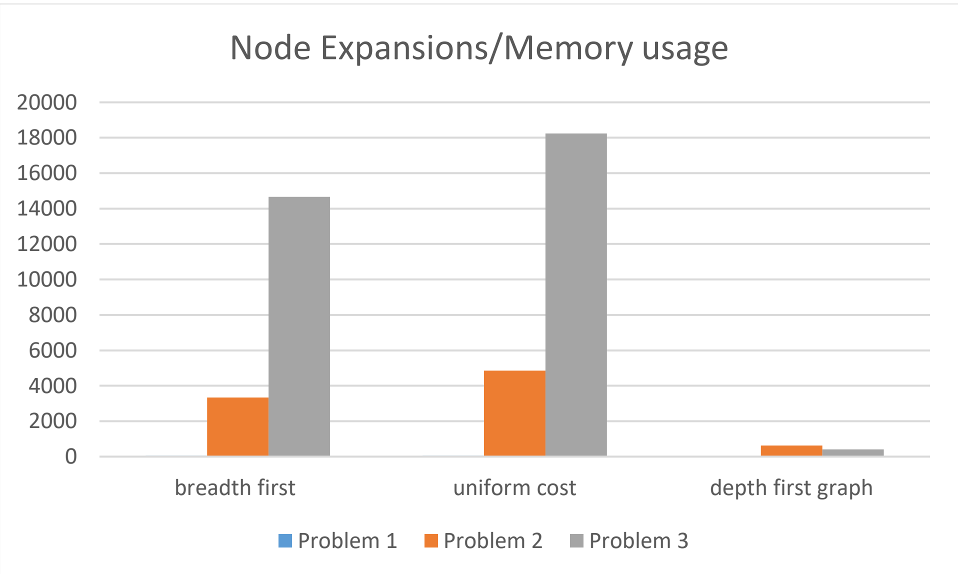
Breadth first search and **uniform cost** search yield an optimal action plan for all the problems. The **depth first graph** search is the fastest and uses the least memory. However, it does not generate an optimal action plan (problem 1: plan length of 20 against the optimal 6, problem 2: 619 against 9, problem 3: 392 against 12)

Plan Length Comparison



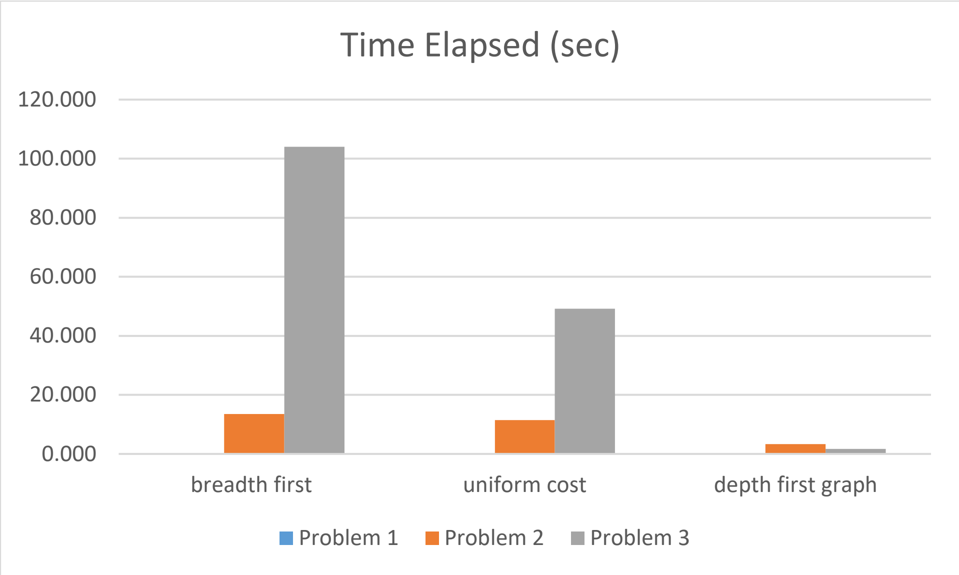
	Plan length		
Search	Problem 1	Problem 2	Problem 3
breadth first	6	9	12
uniform cost	6	9	12
depth first graph	20	619	392

Memory Usage Comparison



	Expansions		
Search	Problem 1	Problem 2	Problem 3
breadth first	43	3343	14663
uniform cost	55	4852	18235
depth first graph	21	624	408

Speed Comparison

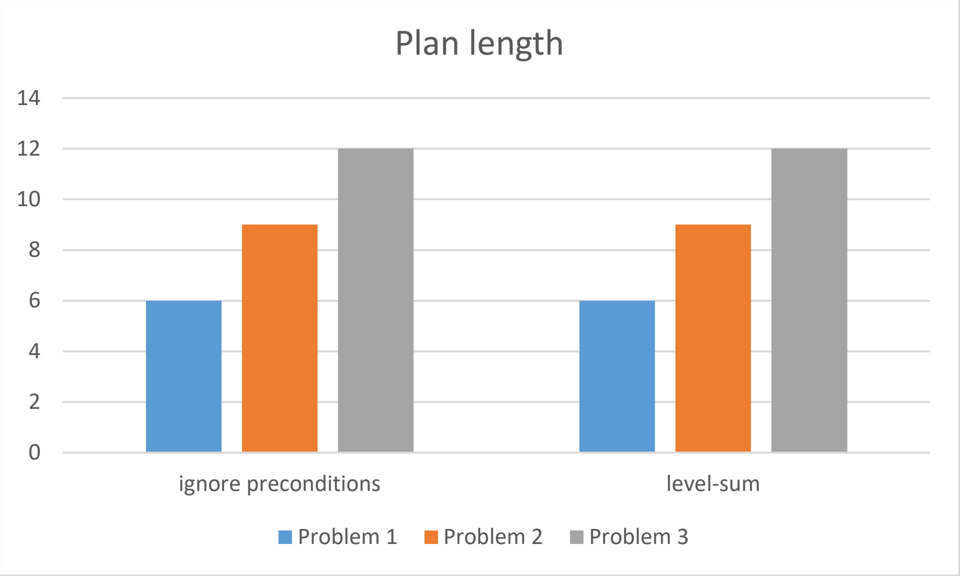


	Time elapsed (sec)		
Search	Problem 1	Problem 2	Problem 3
breadth first	0.030	13.479	104.036
uniform cost	0.033	11.461	49.182
depth first graph	0.013	3.312	1.679

Heuristic Search: A* with ignore preconditions vs level-sum heuristics Comparison

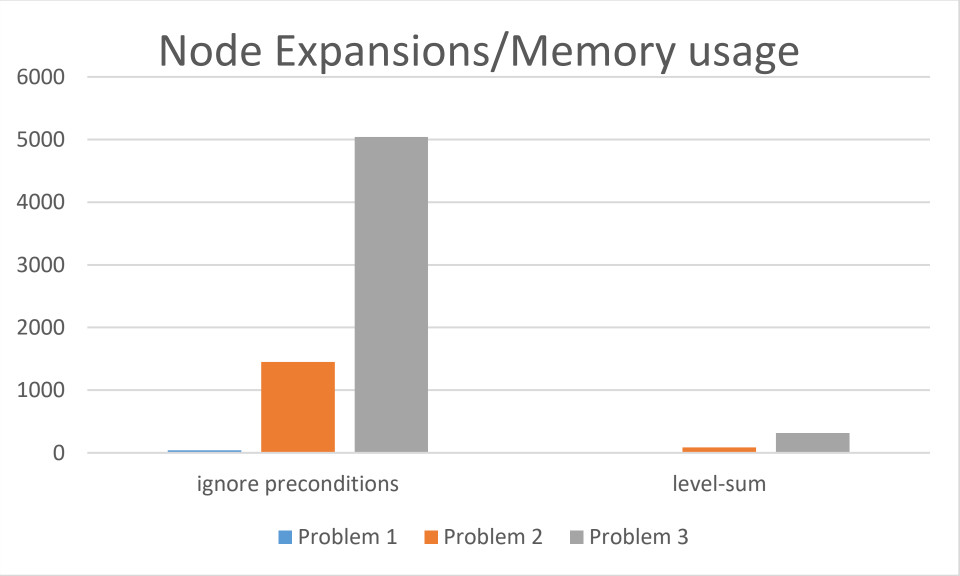
Both heuristics yield an optimal result. Ignore preconditions requires less memory usage and level-sum is faster.

Plan Length Comparison



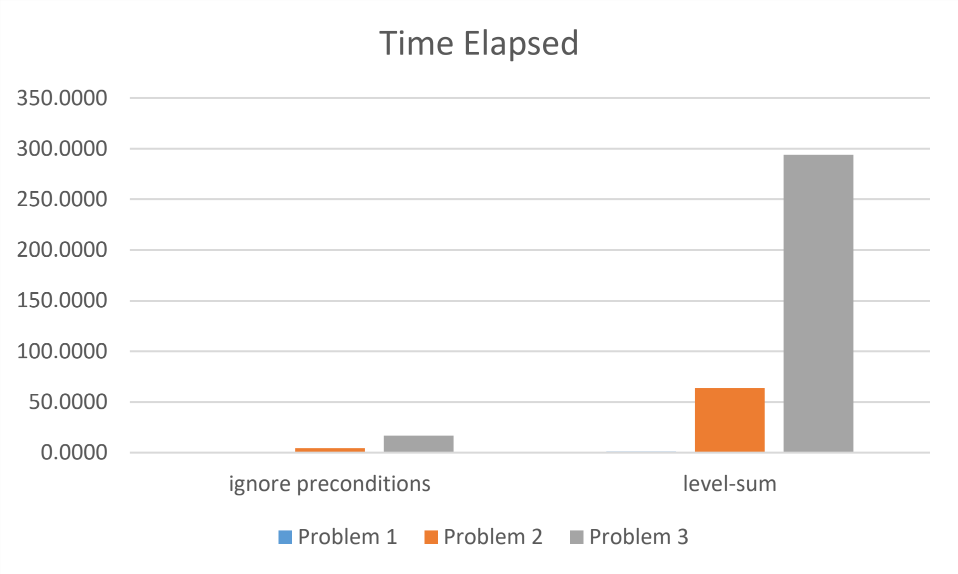
	Plan length		
Heuristic	Problem 1	Problem 2	Problem 3
ignore preconditions	6	9	12
level-sum	6	9	12

Node Expansions/Memory Usage Comparison



	Expansions		
Heuristic	Problem 1	Problem 2	Problem 3
ignore preconditions	41	1450	5040
level-sum	11	86	318

Speed Comparison



	Time elapsed		
Heuristic	Problem 1	Problem 2	Problem 3
ignore preconditions	0.0356	4.5520	16.8181
level-sum	0.8183	63.8418	293.9784

Heuristics Comparison Conclusion

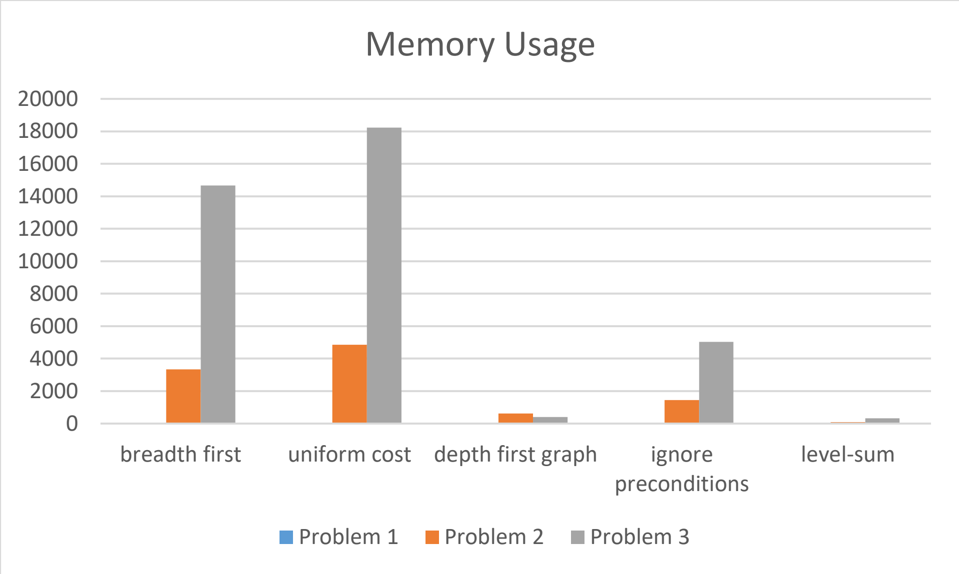
What is the best heuristic? The answer to this questions depends on our requirements and our resources. While both heuristics produce optimal plans we can choose the ignore preconditions heuristic when we need to optimize for speed and we have sufficient memory. If we want to optimize for memory usage, we can go with level-sum heuristics.

Non-Heuristic / Heuristic Comparison

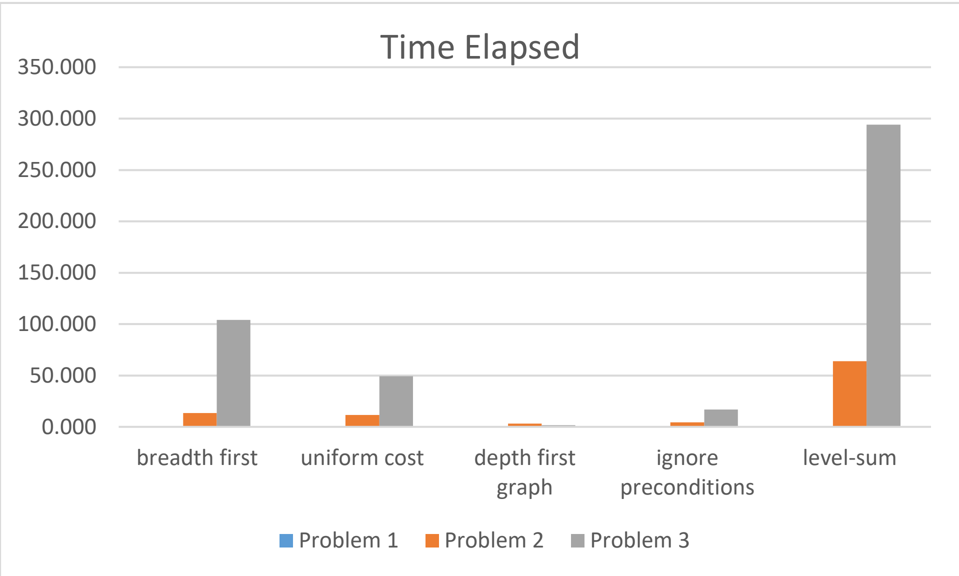
As discussed above both non-heuristic searches (**deep first** and **uniform cost)** and A* with both heuristic functions (**ignore preconditions** and **level-sum**) produce optimal plans. If we take a look at the memory usage and speed charts we notice that the **A* with ignore preconditions heuristic** function requires less memory and also less time to perform. Obviously, A* w ignore preconditions performs equal or better than the other searches in optimality, speed, and memory criteria. Thus is my recommendation.

*we need to mention that depth first search performs better on memory and speed, produces a far less optimal plan, thus is not recommended.

Combined Non-Heuristic / Heuristic Node Expansions-Memory Usage Comparison



Combined Non-Heuristic / Heuristic Time Elapsed-Speed Comparison



Overall Conclusion / Praxis Consistency

The empirical results of our 3 experiments were expected. For example, the observed not optimal solution, yield by the **depth first search** (DFS) was expected. As we can see in the figure 3.21 of our textbook (cited bellow), DFS does not guarantee optimal plan. DFS is not even complete so we were favoured for getting at least a solution without sinking into an endless loop. The observed memory performance is also explained by the linear space complexity of DFS.

In the charts, we can notice the exponential upward trend of space and time complexity of the **breath first** and **uniform cost search** in consistency with the figure 3.21

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.

Also A* with admissible heuristic function (like **ignore preconditions**) is complete and optimal[1]. It is also a simple algorithm which outcomes low execution time. On the other hand, the **level-sum** heuristic is much more accurate but much more complex and obviously more time consuming.

Data Tables

Air Cargo Problem 1 Results

Air Cargo Problem 1						
Search	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time**	**Optimal**
Breadth first search	43	56	180	6	0.030	YES
breadth first tree search	1458	1459	5960	6	0.956	YES
depth first graph search	21	22	84	20	0.013	
depth limited search	101	271	414	50	0.083	
uniform cost search	55	57	224	6	0.033	YES
recursive best first search with h 1	4229	4230	17023	6	2.554	YES
best first graph search with h 1	7	9	28	6	0.004	YES
A* search with h 1	55	57	224	6	0.036	YES
A* search ignore preconditions	41	43	170	6	0.036	YES
A* search with levelsum	11	13	50	6	0.818	YES

Air Cargo Problem 2 Results

Air Cargo Problem 2						
Search	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time**	**Optimal**
Breadth first search	3343	4609	30509	9	13.479	YES
breadth first tree search	-	-	-	-	-	
depth first graph search	624	625	5602	619	3.312	

[1] Stuart Russell, Peter Norvig-Artificial Intelligence A Modern Approach-Prentice Hall (2010), page 109

depth limited search	-	-	-	-	-	
uniform cost search	4852	4854	44030	9	11.461	YES
recursive best first search with h 1	-	-	-	-	-	
best first graph search with h 1	990	992	8910	17	2.391	
A* search with h 1	4852	4854	44030	9	12.612	YES
A* search ignore preconditions	1450	1452	13303	9	4.552	YES
A* search with levelsum	86	88	841	9	63.842	YES

Air Cargo Problem 3 Results

Air Cargo Problem 3						
Search	**Expansions**	**Goal Tests**	**New Nodes**	**Plan length**	**Time**	**Optimal**
Breadth first search	14663	18098	129631	12	104.036	YES
breadth first tree search	-	-	-	-	-	
depth first graph search	408	409	3364	392	1.679	
depth limited search	-	-	-	-	-	
uniform cost search	18235	18237	159716	12	49.182	YES
recursive best first search with h 1	-	-	-	-	-	
best first graph search with h 1	5614	5616	49429	22	15.009	
A* search with h 1	18235	18237	159716	12	53.619	YES
A* search ignore preconditions	5040	5042	44944	12	16.818	YES
A* search with levelsum	318	320	2934	12	293.978	YES