

Artificial Intelligence Nanodegree
Air Cargo Planning Exercise
October 2017

The following are optimal plans for the given air cargo problems:

Problem 1:

- 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 2:

- Load(C1, P1, SFO)
- Load(C2, P2, JFK)
- Load(C3, P3, ATL)
- Fly(P1, SFO, JFK)
- Fly(P2, JFK, SFO)
- Fly(P3, ATL, SFO)
- Unload(C3, P3, SFO)
- Unload(C2, P2, SFO)
- Unload(C1, P1, JF)

Problem 3:

- 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)

Uninformed Search

The following tables capture the metrics of running uninformed searches (namely Breadth First Search (BFS), Depth First Search (DFS) and Uniform Cost Search (UCS)) on three air cargo problems of increasing complexity.

Uninformed Search: Problem 1:

	Breadth First Search	Depth First Search	Uniform Cost Search
Node expansions	43	12	55
Goal tests	56	13	57
Time elapsed (s)	0.0289	0.0095	0.0389
Optimality	Yes	No	Yes

Uninformed Search: Problem 2:

	Breadth First Search	Depth First Search	Uniform Cost Search
Node expansions	3041	187	4761
Goal tests	4672	188	4763
Time elapsed (s)	13.1991	0.5754	11.0627
Optimality	Yes	No	Yes

Uninformed Search: Problem 3:

	Breadth First Search	Depth First Search	Uniform Cost Search
Node expansions	14491	1948	17783
Goal tests	17947	1949	17785
Time elapsed (s)	94.8913	19.2450	47.2797
Optimality	Yes	No	Yes

BFS had the second highest memory requirement of the three searches. It was the slowest search to execute for problems 2 and 3, narrowly outperforming UCS for problem 1. BFS is an optimal search because it examines every node one level at a time.

DFS had the smallest memory requirement and was the quickest search to execute for all three problems. The memory and speed advantage versus UCS and BFS were significant. However, DFS is not an optimal search. This is because DFS does not consider if one node is better than the other. It simply expands nodes as deep as possible one path at a time and stops when a goal-state is found.

UCS had the highest memory requirement as reflected by the largest number of node expansions across all three problems. UCS had the second fastest execution speed for problems 2 and 3. It was the slowest for problem 1. UCS is an optimal search as it expands the nodes with the lowest costs first.

Informed Search

The following tables capture the metrics of running informed A* searches on the same three air cargo problems of increasing complexity.

Informed Search: Problem 1:

	A* (h ₁)	A* (ignore precon)	A* (level sum)
Node expansions	55	41	11
Goal tests	57	43	13
Time elapsed (s)	0.03518	0.03544	1.0423
Optimality	Yes	Yes	Yes

Informed Search: Problem 2:

	A* (h ₁)	A* (ignore precon)	A* (level sum)
Node expansions	4761	1450	86
Goal tests	4763	1452	88
Time elapsed (s)	10.6557	4.0253	150.5042
Optimality	Yes	Yes	Yes

Informed Search: Problem 3:

	A* (h ₁)	A* (ignore precon)	A* (level sum)
Node expansions	17783	5003	311
Goal tests	17785	5005	313
Time elapsed (s)	46.7169	15.1044	759.9272
Optimality	Yes	Yes	Yes

A* searches are optimal.

A* search with the h₁ heuristic had the highest memory requirement across all three problems. It was the second quickest search for problems 2 and 3, and the quickest for problem 1.

A* search with the ignore preconditions heuristic had the second highest memory requirement across all three problems. It was the quickest search for problems 2 and 3, and the second quickest for problem 1.

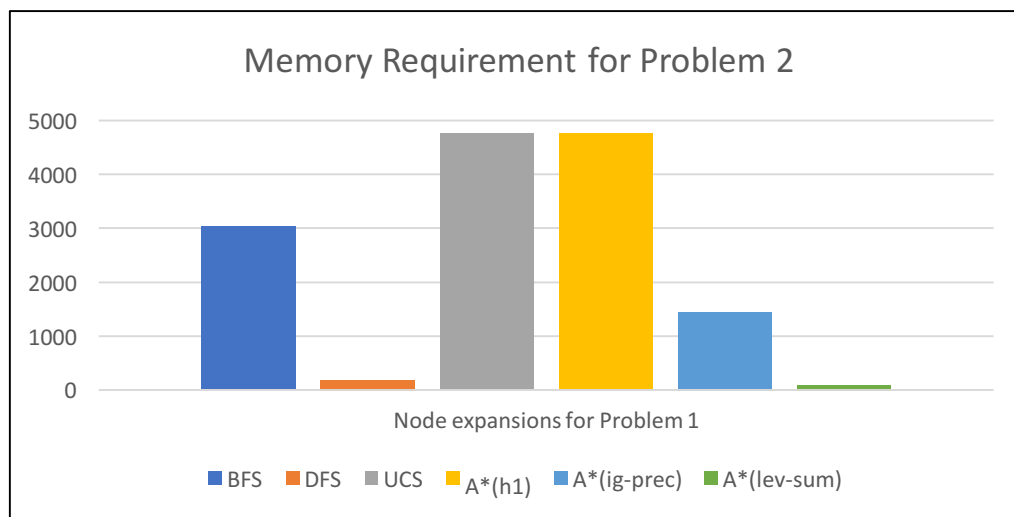
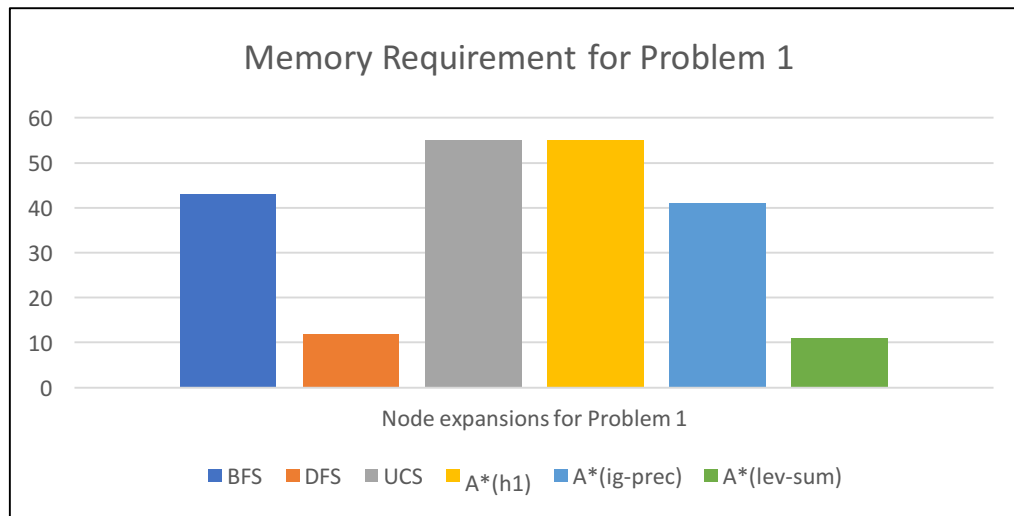
A* search with the level sum heuristic had the smallest memory requirement across all three problems, but was also the slowest search to execute. The difference in number of node expansions and execution time were significant. This suggests that the level sum heuristic is a very good estimator but that it is also very complex and hence computationally expensive.

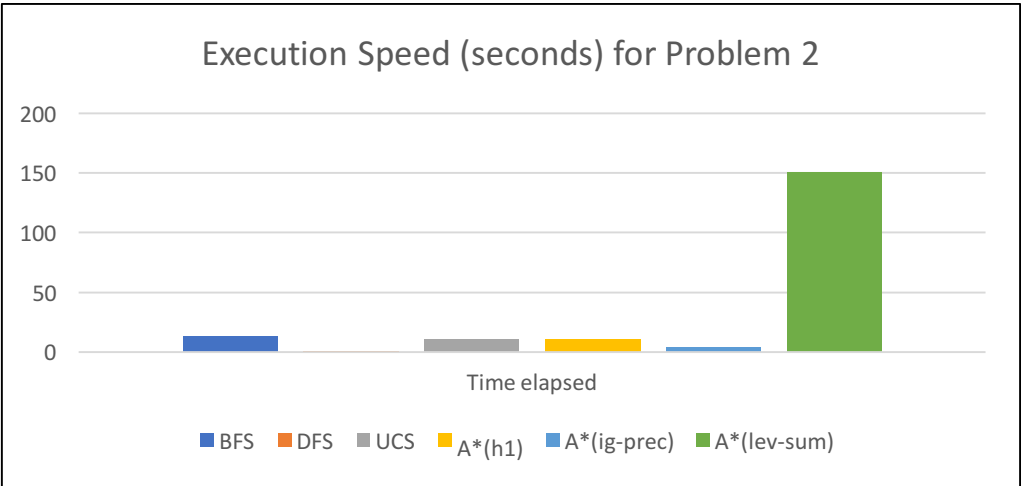
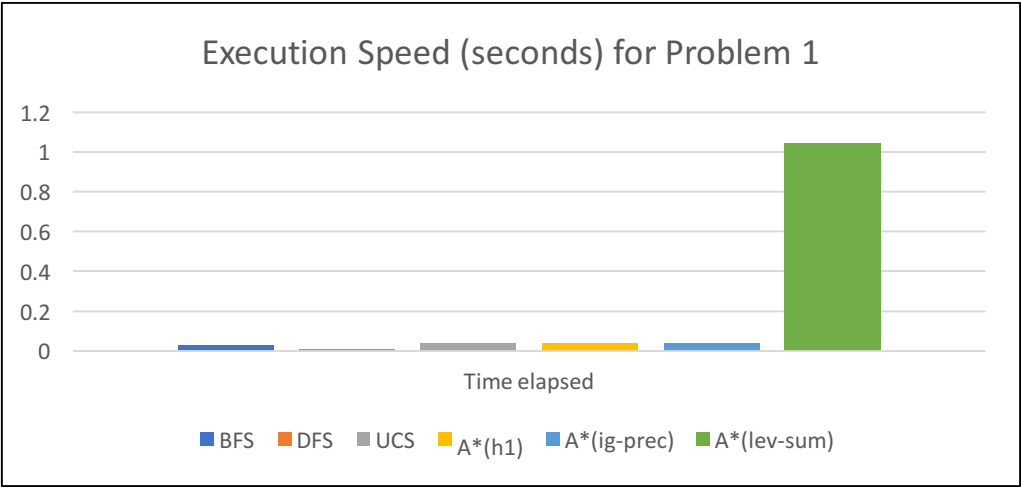
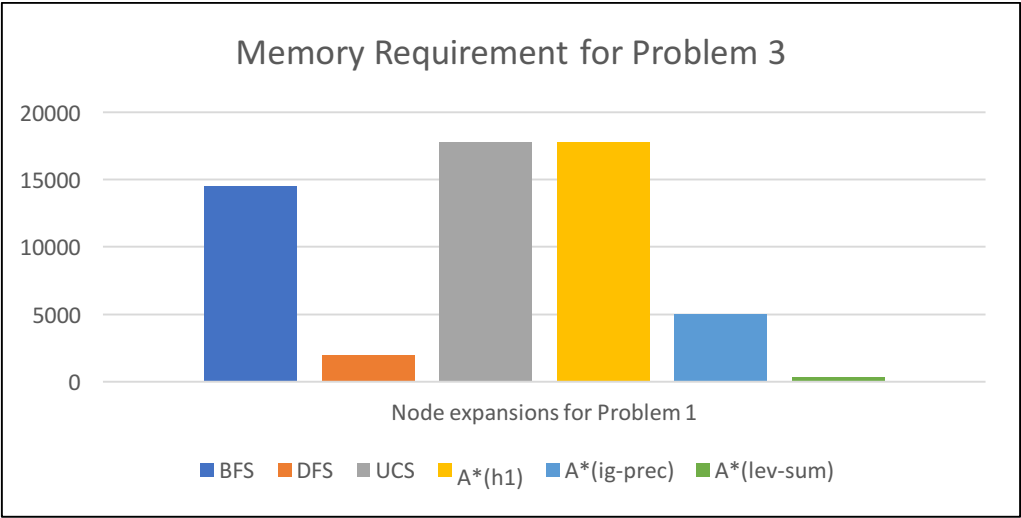
Comparing uninformed and informed searches

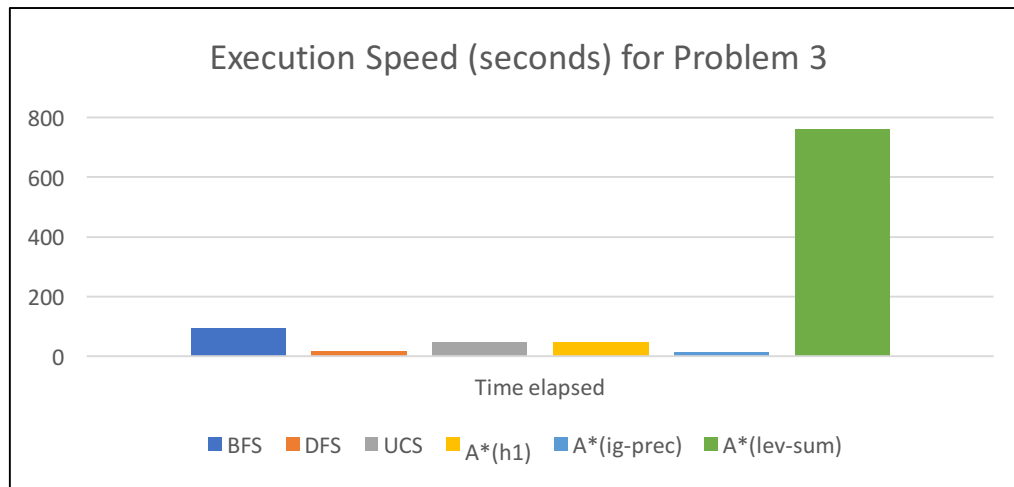
BFS found an optimal solution more quickly than the A* searches for problem 1, while A* search with the ignore preconditions heuristic found an optimal solution the quickest for problems 2 and 3. This suggests that an uninformed search can be more efficient for simple problems. As complexity increases, informed searches tend to outperform, and their higher computational costs start to pay off.

With regards to memory requirements, informed searches consistently expanded fewer nodes than uninformed searches, reflecting the value of the heuristics applied.

For simple problems, BFS is a good choice, while for more complex problems, A* search with the ignore preconditions heuristic is preferable. If minimizing memory requirement is the absolute priority and compute power is abundant, A* with the level sum heuristic would be preferable across all three problems.







References

1. Stuart Russell, Peter Norvig (2014), Artificial Intelligence: A Modern Approach (Third Edition)