Planning Search Heuristic Analysis

After running different test cases and analyzing the results, it clearly illustrates the benefits of using informed search strategies with custom heuristics over uninformed search techniques when searching for an optimal plan. It benefits both in terms of speed and memory usage.

Uninformed Search Strategies Analysis – one that don't have no additional information about states beyond that provided in the problem definition. All they can do is generate successors and distinguish a goal state from a non-goal state

The below measures were collected using the following commands:
python run_search.py -p 1 -s 1 2 3 4 5 6 7
python run_search.py -p 2 -s 1 3 5 7
python run_search.py -p 3 -s 1 3 5 7

For Problem 2, as the execution time exceeded 10 minutes and still searching, cancelled data collection for Breadth First Tree Search, Depth Limited Search, and Recursive Best First Search. For the same reason, with Problem 3 data for Breadth First Tree Search, Depth Limited Search, Uniform Cost Search, and Recursive Best First Search was not collected.

Problem 1			
Search	Length	Time	Nodes
BFS	6	0.0655	180
BF_tree_Search	6	2.734	5960
Depth first graph search	12	0.0183	48
Depth Limited Search	50	0.2385	414
Uniform Cost Search	6	0.1042	224
h_1	6	7.7601	17029

Problem 2			
Search	Length	Time	Nodes
BFS	9	31.3177	31049
BF_tree_Search		>10000.00	
Depth first graph search	346	2.967	3142
Uniform Cost Search	9	24.945	43206
h_1	9	2.5937	4950

Problem 3			
Search	Length	Time	Nodes
BFS	12	201.81	128184
Depth first graph search	1878	41.0625	16253
Uniform Cost Search	12	110.433	155920
h_1	22	25.2586	4033

informed search strategy — one that uses problem-specific knowledge beyond the definition of the problem itself — can find solutions more efficiently than can an uninformed strategy. In this section, we compare the performance of A* Search using three different heuristics.

The below measures were collected using the following commands:
python run_search.py -p 1 -s 8 9 10
python run_search.py -p 2 -s 8 9
python run_search.py -p 3 -s 8 9

For Problems 2 and 3, because its execution time exceeded 10 minutes, we did not collect data for A* Search with Level Sum heuristic.

Problem 1			
Search	Length	Time	Nodes
A* with H1	6	0.1552	224
A* with h_ignore_preconditions	6	0.1593	224
A* with h_pg_levelsum	6	5.3269	50

Problem 2			
Search	Length	Time	Nodes
A* with H1	9	48.8433	43206
A* with h_ignore_preconditions	9	48.6791	43206
A* with h_pg_levelsum	-	-	-

Problem 3			
Search	Length	Time	Nodes
A* with H1	12	155.367	155920
A* with h_ignore_preconditions	12	117.343	155920
A* with h_pg_levelsum	-	-	-

Based on the above results we can observe that all heuristics yield an optimal action plan,but only the h1 and Ignore Preconditions heuristics return results within the 10mn max execution time. Of the two strategies mentioned above, A* Search with Ignore Preconditions heuristic is the fastest. If we let search run to completion on our machine, A* Search with Level Sum heuristic uses the least memory, but its execution time is much slower.

Informed vs Uninformed Search Strategies
As we saw earlier, when it comes to execution speed and memory usage of uninformed search strategies, Depth First Graph Search is faster and uses less memory than Uniform Cost Search. As for informed search strategies, A* Search with Ignore Preconditions heuristic is the fastest and uses the least memory. So, really, the choice is between Depth First Graph Search and A* Search with Ignore Preconditions heuristic.

A* Search with Ignore Preconditions heuristic would be the best choice overall for our Air Cargo problem because of it is faster and uses less memory

Optimal Solutions:

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, JFK)

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)

Conclusion:

The results above clearly illustrate the benefits of using informed search strategies with custom heuristics over uninformed search techniques when searching for an optimal plan. The benefits are significant both in terms of speed and memory usage.

References, further readings:

- Russell, S, and P Norvig. **Artificial Intelligence: A Modern Approach,**Chapter 10: Classical Planning
- https://en.wikibooks.org/wiki/Artificial_Intelligence/Definition
- https://stackoverflow.com/questions/39760905/what-is-the-main-difference-between-informed-search-and-uninformed-search-algori
- https://artificialintelligentsystems.wordpress.com/category/ai-searching-techniques/page/2/