Heuristic Analysis - Air Cargo Transport

The following includes a brief analysis of heuristics and other search algorithms used in solving the air cargo transport problems

Problem 1

This problem has the following setup: two cargos C1, C2; two planes P1, P2; and two destinations: SFO and JFK.

Start

- SFO has C1 and P1
- JFK has C2 and P2

Goal

- SFO has cargo C2
- JFK has cargo C1

Optimal Solution

The optimal solution has 6 steps:

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

Algorithm Analysis

The following table shows the performance of various algorithms.

No.	Algorithm	Expansions	Goal Tests	New Nodes	Steps	Time (s)
1	BFS	43	56	180	6	0.046
2	BFTS	1458	1459	5960	6	1.362
3	DFS	21	22	84	20	0.0198
4	DLS	101	271	414	50	0.124
5	UCS	55	57	224	6	0.054
6	R-BFS	4229	4230	17023	6	3.94
7	**G-BFGS**	7	9	28	6	**0.007**
8	A* h1	55	57	224	6	0.06
9	A* hIgnore	41	43	170	6	0.039
10	A* hLevelsum	11	13	50	6	1.09

The fastest performance is with the **Greedy Best First Graph Search with h1 heuristic** algorithm. This expands the fewest number of nodes (7) and fewest goal tests (9) and, as a result, has the lowest execution time (0.007 seconds). This is because the G-BFGS expands the node closest to the goal (assuming that it will lead to the quickest search). It evaluates nodes using the nodes using just the heuristic function, i.e. $f(n) = h(n)$

Problem 2

The second problem has: three cargos (C1, C2, C3); three airports (SFO, JFK, ATL); and three planes (P1, P2, P3).

Start

- SFO has C1 and P1
- JFK has C2 and P2
- ATL has C3 and P3

Goal

- JFK has cargo C1
- SFO has cargo C2 and C3

Solution

The optimal solution has 9 steps:

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

Algorithm Analysis

No.	Algorithm	Expansions	Goal Tests	New Nodes	Steps	Time (s)
1	BFS	3343	4609	30509	9	18.34
2	BFTS	-	-	-	-	Timedout
3	DFS	624	625	5602	619	4.475
4	DLS	-	-	-	-	Timedout

5	UCS	4853	4855	44041	9	18.52
6	R-BFS	-	-	-	-	Timedout
7	**G-BFGS**	998	1000	8982	21	**3.786**
8	A* h1	4853	4855	44041	9	18.463
9	A* hIgnore	1450	1452	13303	9	5.586
10	A* hLevelsum	86	88	841	9	266.587

In this problem as well, the **Greedy Best-First Graph Search with h1 heuristic** performs best, timing at 3.786 seconds; however, it does *not find the optimal solution!* The optimal solution of 9 steps is found by **Breadth First Search** or **A-star with ignoring pre-conditions heuristic**, though both of them take just a little bit longer than G-BFGS. BFS eventually finds its goal at a finite depth d, but takes longer; however it runs into space/memory constraints if d is large. A* with ignore heuristic performs better, finding the optimal solution in 9 step, with far fewer node expansions than BFS. One would prefer this A* with ignore heuristic over other algorithms as it finds optimal solution.

Problem 3

The second problem has: four cargos (C1, C2, C3, c4); four airports (SFO, JFK, ATL, ORD); and only two planes (P1, P2).

Start

- SFO has C1 and P1
- JFK has C2 and P2
- ATL has cargo C3
- ORD has cargo C4

Goal

- JFK has cargo C1 and C3
- SFO has cargo C2 and C4

Solution

The optimal solution has 12 steps:

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

Algorithm Analysis

No.	Algorithm	Expansions	Goal Tests	New Nodes	Steps	Time (s)
1	BFS	14663	18098	129631	12	137.74
2	BFTS	-	-	-	-	Timedout
3	DFS	408	409	3364	392	**2.787**
4	DLS	-	-	-	-	Timedout
5	UCS	18223	18225	159618	12	92.97
6	R-BFS	-	-	-	-	Timedout

7	G-BFGS	5578	5580	49150	22	29.007
8	A* h1	18223	18225	159618	12	95.79
9	*A hIgnore**	5040	5042	44944	**12**	**27.59**
10	A* hLevelsum	-	-	-	-	Timedout

In this problem, the **Depth First Search** performs the fastest with just 2.78 seconds; however the solution is highly **unoptimal** at a wieldy 392 steps. But the **A-star with ignore preconditions heuristic** really shines here, with an optimal solution at 12 steps, and a very reasonable 27.59 seconds, the second best time. For large search problems such as this one, A* with ignore precondition ignores any preconditions for all actions, and thus every action becomes applicable in every state, and a goal fluent can be achieved in a single step.

Reference:

AIMA - Uninformed Search Strategies (Chap 3.4), Informed Search Strategies (Chap 3.5)