Heuristic Analysis of Planning Search Algorithms for Air Cargo Transport System

The heuristic analysis for various search algorithms in the planning domain is being done for Air Cargo Transport System. Three planning problems are being explored for the analysis.

Problem 1:

Initial and Goal state	**Optimal Plan (Length: 6)**
Init (At(C1, SFO) ∧ At(C2, JFK) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Plane(P1) ∧ Plane(P2) ∧ Airport(JFK) ∧ Airport(SFO)) Goal (At(C1, JFK) ∧ At(C2, SFO))	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)

Air Cargo Problem 1					
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time Elapsed**
	Non-heuristic Search				
Breadth First Search	43	56	180	6	0.0517
Breadth First Tree Search	1458	1459	5960	6	1.5912
Depth First Graph Search	21	22	84	20	0.0236
Depth Limited Search	101	271	414	50	0.1592
Uniform Cost Search	55	57	224	6	0.0648
Recursive Best First Search h_1	4229	4230	17023	6	4.7021
Greedy Best First Graph Search h_1	7	9	28	6	0.0087
	Heuristic Search				
A* Search h_1	55	57	224	6	0.0651
A* Search h_ignore_preconditions	41	43	170	6	0.0788
A* Search h_pg_levelsum	11	13	50	6	3.3382

Analysis:

1. Since there are total 12 clauses, the search space consists of 2^{12} states.
2. Optimal plan has length of 6. *"Greedy Best First Graph Search"* generates this plan in least amount of time and is the clear winner. BFS, DFS, and Uniform Cost Search are other non-heuristic searches that give comparable performances.
3. A* search also reaches goal state quite fast, but is not needed for such limited state space.
4. Greedy Best First Search also generates least number of new nodes and performs minimum expansions/goal tests. Thus, the amount of memory required for this problem is also least for this search method.

Problem 2:

Initial and Goal States	**Optimal Plan (Length: 9)**
Init (At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3) ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL)) Goal (At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))	1. Load(C1, P1, SFO) 2. Load(C2, P2, JFK) 3. Load(C3, P3, ATL) 4. Fly(P2, JFK, SFO) 5. Unload(C2, P2, SFO) 6. Fly(P1, SFO, JFK) 7. Unload(C1, P1, JFK) 8. Fly(P3, ATL, SFO) 9. Unload(C3, P3, SFO)

Air Cargo Problem 2					
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time Elapsed**
	Non-heuristic Search				
Breadth First Search	3343	4609	30509	9	20.9285
Breadth First Tree Search	Aborted				
Depth First Graph Search	624	625	5602	619	5.2687
Depth Limited Search	Aborted				
Uniform Cost Search	4853	4855	44041	9	67.3762
Recursive Best First Search h_1	Aborted				
Greedy Best First Graph Search h_1	998	1000	8982	21	10.8479
	Heuristic Search				
A* Search h_1	4853	4855	44041	9	65.3930
A* Search h_ignore_preconditions	1506	1508	13820	9	22.2351
A* Search h_pg_levelsum	86	88	841	9	356.7744

Analysis:

1. Since there are total 27 clauses, the search space consists of 2^{27} states.
2. Optimal plan has length of 9. In non-heuristic search algorithms, BFS performs best as it gives optimal plan in least amount of time. In heuristic searches, A* with “ignore preconditions” heuristic performs best results and even generates lesser number of nodes than BFS.
3. Recursive best first search does not return an optimal plan as it recursively expands the same node repeatedly. Breadth First Tree Search and Depth Limited Search also fail to report a plan in expected time, and hence aborted.
4. Though A* with level_sum heuristic generates much fewer nodes than A* with ignore preconditions heuristic, it takes more time as calculation of this heuristic value is computationally more expensive.

Problem 3:

Initial and Goal States	**Optimal Plan (Length: 12)**
Init (At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD) ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4) ∧ Plane(P1) ∧ Plane(P2) ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD)) Goal (At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))	1. Load(C1, P1, SFO) 2. Fly(P1, SFO, ATL) 3. Load(C3, P1, ATL) 4. Fly(P1, ATL, JFK) 5. Unload(C1, P1, JFK) 6. Unload(C3, P1, JFK) 7. Load(C2, P1, JFK) 8. Fly(P1, JFK, ORD) 9. Load(C4, P1, ORD) 10. Fly(P1, ORD, SFO) 11. Unload(C2, P1, SFO) 12. Unload(C4, P1, SFO)

Air Cargo Problem 3					
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time Elapsed**
	Non-heuristic Search				
Breadth First Search	8602	11196	64308	12	68.0131
Breadth First Tree Search	Aborted				
Depth First Graph Search	1292	1293	5744	875	5.4694
Depth Limited Search	Aborted				
Uniform Cost Search	11482	11484	85785	12	241.5744
Recursive Best First Search h_1	Aborted				
Greedy Best First Graph Search h_1	907	907	5581	19	4.7595
	Heuristic Search				
A* Search h_1	11482	11484	85785	12	237.6082
A* Search h_ignore_preconditions	2494	2496	19532	12	38.7255
A* Search h_pg_levelsum	306	308	2148	12	958.6801

Analysis:

1. Since there are total 32 clauses, the search space consists of 2^{32} states.
2. Optimal plan has length of 12. Again, in non-heuristic search category, BFS performs best as it gives optimal plan in least amount of time. In heuristic search category, A* with ignore preconditions heuristic performs best; and for this problem it outperforms BFS as it takes lesser time and generates much fewer nodes.
3. So, we see that as the problem size increases, heuristic search algorithms like A* start outperforming non-heuristic methods. Since A* search is admissible it always gives optimal plans.

4. We also note that A* Search with level sum heuristic takes much more time than ignore preconditions heuristic. Though level_sum heuristic generates fewer new nodes and performs lesser goal tests. This is because level sum heuristics is computationally more expensive.
5. So, we finally conclude that for large size of Air Cargo Transportation System, A* Search with ignore preconditions heuristic outperforms all other search methods.