AIND-Planning Heuristic Analysis

Air Cargo Problem 1 Chart 1.1

SEARCH FUNCTION		EXPANSIONS	GOAL TESTS	NEW NODES	TIME seconds	PLAN LENGTH
Non-Heuristic	BFS: Breadth First	43	56	180	0.02	6
	BFT: Breadth First Tree	1458	1459	5960	0.82	6
	DFG: Depth First Graph	21	22	84	**0.01**	20
	DL: Depth Limited	101	271	414	0.08	6
	UC: Uniform cost	55	57	224	0.03	6
	RBFG: Recursive Best First Graph	4229	4230	17023	2.43	6
	GBFG: Greedy Best First Graph	**7**	**9**	**28**	0.006	**6**
	A *	55	57	224	0.53	6
Heuristic	A*IP: A * Ignore Preconditions	41	43	170	0.03	6
	A*LS: A * Level Sum	11	13	50	0.48	6

Optimal Plan Length: 6

OptimaL Plan: GBFG

Search Space Size: 2^{12}

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

AIND-Planning Heuristic Analysis

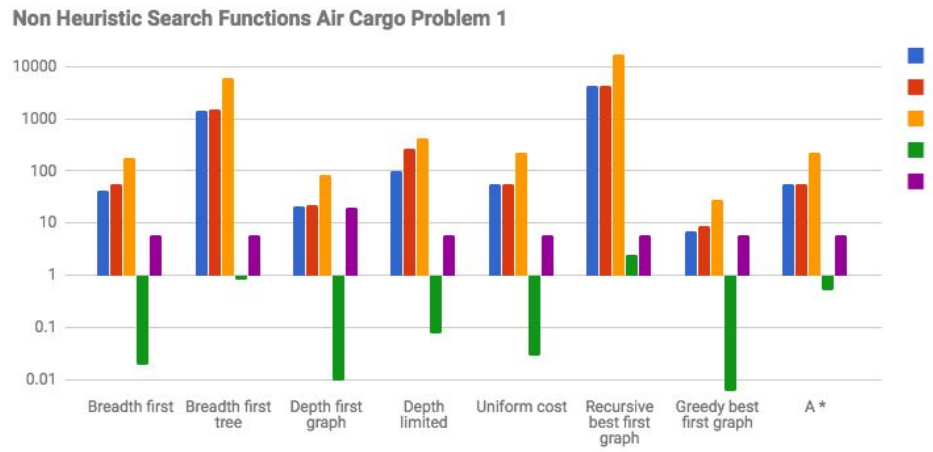

Chart 1.2

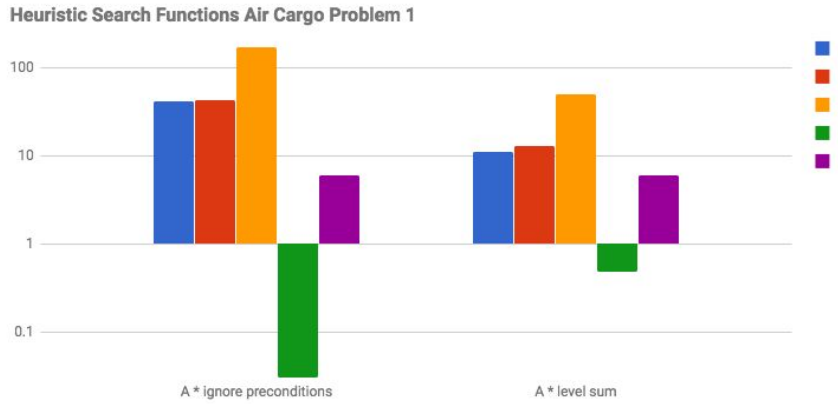

Chart 1.3

Problem 1 Analysis

This is the smallest problem of all three to be analyzed with a state space of 2^{12}.
Non-Heuristic DFG performed the fastest but had the highest plan length. GBFG had the least expansions, goal tests, new nodes and shortest plan length. DFS is not optimal with the highest plan length because it goes the deepest before considering frontier nodes. UC found the optimal plan but is slower and continues searching even after finding the optimal plan to expand the next node with the lowest cost. The heuristic search functions did not perform the best for this small problem.

AIND-Planning Heuristic Analysis

Air Cargo Problem 2 Chart 2.1

SEARCH FUNCTION		EXPANSIONS	GOAL TESTS	NEW NODES	TIME seconds	PLAN LENGTH
Non-Heuristic	**BFS: Breadth First**	3343	4609	30509	**12.44**	**9**
	BFT: Breadth First Tree	too long to compute on my mac osx				
	DFG: Depth First Graph	624	**625**	5602	**3.15**	619
	DL: Depth Limited	too long to compute on my mac osx				
	UC: Uniform cost	4849	4851	44001	10.83	9
	RBFG: Recursive Best First Graph	too long to compute on my mac osx				
	GBFG: Greedy Best First Graph	990	992	8910	28.73	17
	A *	4849	4851	44001	10.59	9
Heuristic	A*IP: A * Ignore Preconditions	1443	1445	13234	**3.71**	9
	A*LS: A * Level Sum	**85**	**87**	**831**	44.75	**9**

Optimal Plan Length: 9

Optimal Plan: BFS

Search Space Size: 2^{27}

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

AIND-Planning Heuristic Analysis

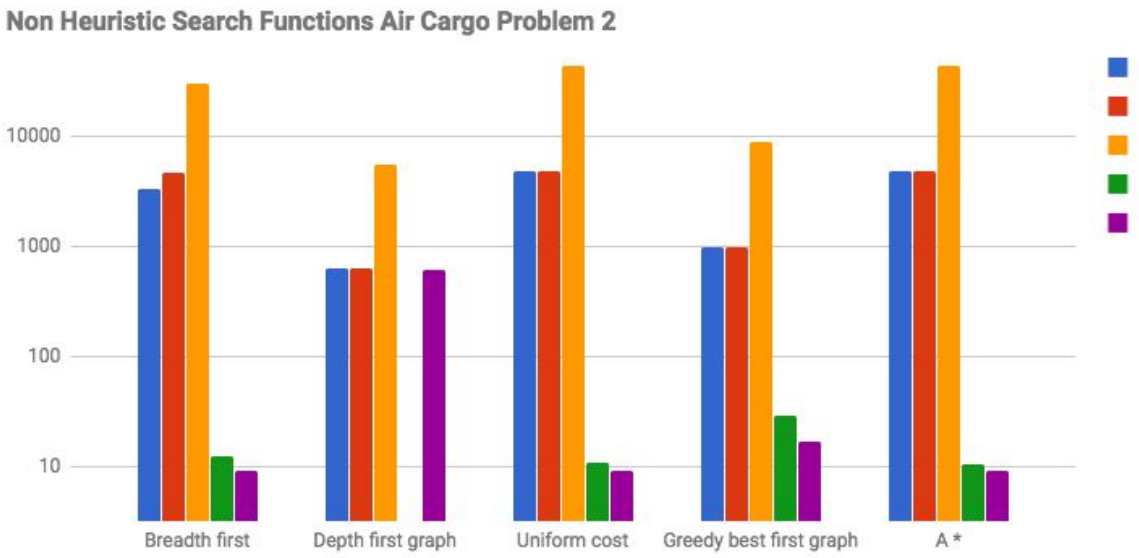

Chart 2.2

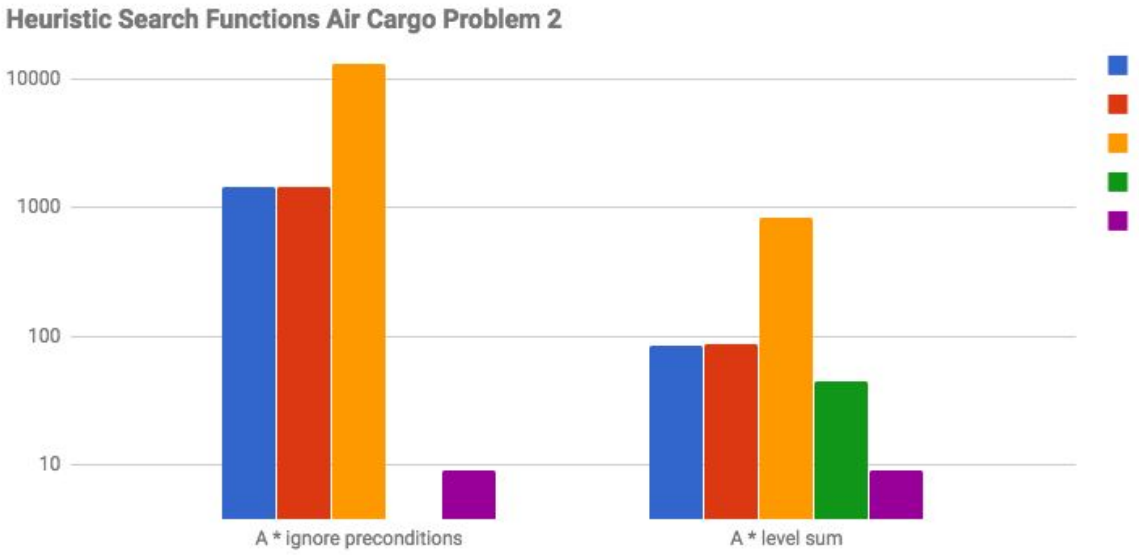

Chart 2.3

Problem 2 Analysis

The state space for this problem is 2^{27}. A*LS provided optimal plan, least number of expansions, goal tests and new nodes; but required the most time to compute. BFS was the best performer with the optimal plan of 9. DFG was fast but did not find an optimal plan.

AIND-Planning Heuristic Analysis

Air Cargo Problem 3 Chart 3.1

SEARCH FUNCTION		EXPANSIONS	GOAL TESTS	NEW NODES	TIME seconds	PLAN LENGTH
Non-Heuristic	BFS: Breadth First	14663	18098	128631	92.98	12
	BFT: Breadth First Tree	too long to compute on my mac osx				
	DFG: Depth First Graph	408	409	3364	1.59	392
	DL: Depth Limited	too long to compute on my mac osx				
	UC: Uniform cost	18235	18237	159716	45.43	12
	RBFG: Recursive Best First Graph	too long to compute on my mac osx				
	GBFG: Greedy Best First Graph	5462	5464	48176	13.84	21
	A *	18235	18237	159716	46.16	12
Heuristic	**A*IP: A * Ignore Preconditions**	4945	4947	43991	14.25	12
	A*LS: A * Level Sum	318	320	2934	231.18	12
Optimal Plan Length: 12 **Optimal Plan: A*IP** **Search Space Size: 2^{32}**						
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C1, P1, JFK) Unload(C3, P1, JFK) Fly(P2, ORD, SFO) Unload(C2, P2, SFO) Unload(C4, P2, SFO)						

AIND-Planning Heuristic Analysis

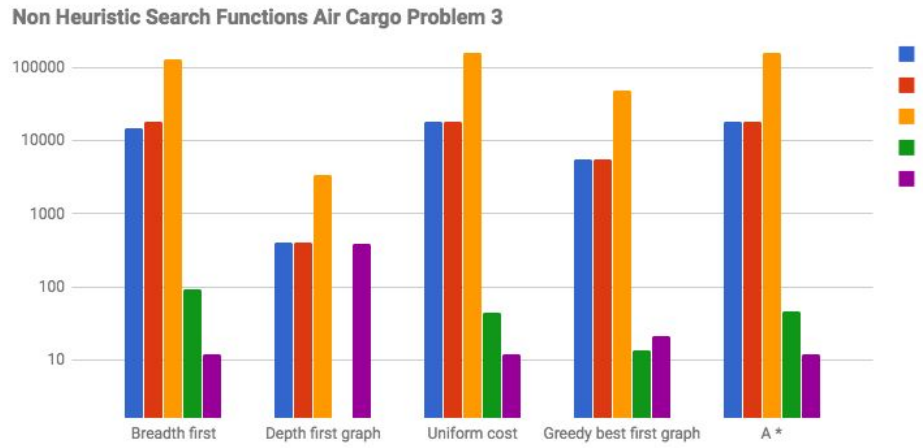

Chart 3.2

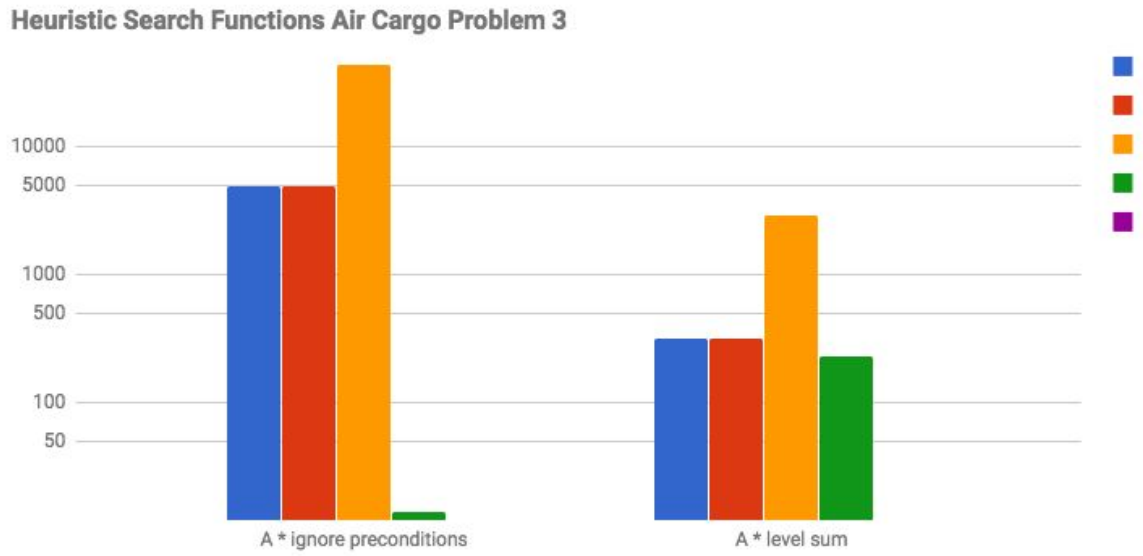

Chart 3.3

AIND-Planning Heuristic Analysis

Problem 3 Analysis

State space is 2^{32}, the optimal plan is up to 12 and the time in seconds to run the search is longer than the previous two problems. This is the first of our three problems to have a heuristic search as the best option; A * ignore precondition (A*IP).

A*IP provided the optimal plan of 12 and performed better than the rest of the heuristic and non-heuristic functions. It was much faster than A * level sum (A*LS). The calculation time of A*LS was much higher, even though it had less node expansions.

Conclusion

Breadth first search (BFS) was the most optimal non-heuristic search function with less expansions and faster computing time for our smallest problem with search state space of 2^{12}, while the heuristic search functions (A*IP and A*LS) were not.

The heuristic functions had less expansions with better results for our larger problems with a search state space of 2^{27} and 2^{32}.

It appears that between the three algorithm choices of BFS, A*IP and A*LS we can get the best results and find the optimal plan with the following assumptions:

State Search Space	Best Algorithm Choice	Worse Algorithm Choice
2^{12}	BFS Breadth first search	RBFG, BFT, DFG
2^{27}	A*IP A * Ignore Preconditions	RBFG, BFT, DL, DFG
2^{32}	A*LS A * Level Sum	RBFG, DL, BFT, DFG

REFERENCES

P.Norvig, S. Russell. Artificial Intelligence A Modern Approach Third Edition. Chapter 10, pages 372 - 406

AI Engineer NanoDegree Course. https://www.udacity.com/ai

When is it practical to use DFS vs BFS? Retrieved June 27th 2017 from http://www.stackoverflow.com/questions/3332947/when-is-it-practical-to-use-dfs-vs-bfs