Heuristic Analysis

Part 1. Metrics for non-heuristic planning solution searches

The tables below contain the metrics for running three different search algorithms on three different air cargo problems. The examined search algorithms are: breadth_first_search, depth_first_graph_search, and uniform_cost_search.

Table 1. Metrics for Air Cargo Problem 1 with non-heuristic planning

Search Algorithm	Expansions	Goal Test	Time (s)	Optimality
breadth_first	43	56	0.04	Yes
depth_first_graph	21	22	0.02	No
uniform_cost	55	57	0.05	Yes

Table 2. Metrics for Air Cargo Problem 2 with non-heuristic planning

Search Algorithm	Expansions	Goal Test	Time (s)	Optimality
breadth_first	3343	4609	10.3	Yes
depth_first_graph	624	625	3.9	No
uniform_cost	4853	4855	14.44	Yes

Table 3. Metrics for Air Cargo Problem 3 with non-heuristic planning

Search Algorithm	Expansions	Goal Test	Time (s)	Optimality
breadth_first	14663	18098	52.1	Yes
depth_first_graph	408	409	2.1	No
uniform_cost	18223	18225	67.8	Yes

Part 2. Metrics for heuristic-based planning solution searches

The next set of tables contain the metrics for running three different heuristics on the same three air cargo problems from above. The examined heuristics are: greedy_best_first_graph_serach with h_1, astar_search with h_1, and astar_search with h_ignor_preconditions.

Table 4. Metrics for Air Cargo Problem 1 with heuristic planning

Heuristic	Expansions	Goal Test	Time (s)	Optimality
greedy_best_first	7	9	0.01	Yes
astar_search_h_1	55	57	0.04	Yes
astar_search_ignr	41	43	0.05	Yes

Table 5. Metrics for Air Cargo Problem 2 with heuristic planning

Heuristic	Expansions	Goal Test	Time (s)	Optimality
greedy_best_first	998	1000	2.9	No
astar_search_h_1	4853	4855	14.4	Yes
astar_search_ignr	1450	1452	5.35	Yes

Table 6. Metrics for Air Cargo Problem 3 with heuristic planning

Heuristic	Expansions	Goal Test	Time (s)	Optimality
greedy_best_first	5578	5580	19.4	No
astar_search_h_1	18223	18225	63.0	Yes
astar_search_ignr	5040	5042	20.3	Yes

Part 3. Summary of optimal plans and strategy comparisons

The following table contains the optimal plans for each problem using the fasted search or heuristic algorithm.

Problem	Search/Heuristic	Time(s)	Plan

Air Cargo Problem 1	greedy_best_first	0.001	`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	Search/Heuristic	Time(s)	Plan
Air Cargo Problem 2	astar_search_ignr	5.35	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)
Air Cargo Problem 3	astar_search_ignr	20.3	Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P2, ORD, SFO) Unload(C4, P2, SFO) Load(C1, P1, SFO) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C3, P1, JFK) Unload(C1, P1, JFK) Unload(C2, P2, SFO)

As the results in the table above indicate, the introduction of heuristics can result in optimal searches in reduced amounts of times. This effect becomes more pronounced as the problem becomes more complex.

In these set of problems, the non-heuristic search algorithms examined all found a solution to the problem. The key distinction can be seen between the breadth_first and depth_first_graph algorithms, where the former is optimal and latter is not. When working with simple problems, like the first two air cargo problems, it is useful to use algorithms like breadth_first that look for the shortest path first. Such approach finds an optimal solution in a short amount of time.

However, when working with more difficult problems, like air cargo problem 3, then it becomes important to start using heuristic-based searches. Heuristics open up multiple avenues to find optimal solutions in a reduced amount of time relative to a breadth_first search approach. A key challenge for heuristics is that they are problem dependent and require a throughout understudying of the problem for their design and implementation. Thus, it is useful to test non-heuristics searches to gain understanding of the problem complexity and use the information to design the heuristic algorithms.