

Heuristic Analysis

by Babatunde Adeyemi

In this project, deterministic logistics planning problems for an Air Cargo transport system was implemented using a planning search agent. A *Planning Graph* with *Automatic Domain-Independent Heuristics with A* search* was employed, and a comparison of their results/performance using different search algorithms is provided in this write-up.

Three planning problems in the Air Cargo domain were given, with all of them having the same action schema defined below, but with different initial states and goals.

Air Cargo Action Schema

```
Action(Load(c, p, a),  
    PRECOND: At(c, a)  $\wedge$  At(p, a)  $\wedge$  Cargo(c)  $\wedge$  Plane(p)  $\wedge$  Airport(a)  
    EFFECT:  $\neg$  At(c, a)  $\wedge$  In(c, p))  
Action(Unload(c, p, a),  
    PRECOND: In(c, p)  $\wedge$  At(p, a)  $\wedge$  Cargo(c)  $\wedge$  Plane(p)  $\wedge$  Airport(a)  
    EFFECT: At(c, a)  $\wedge$   $\neg$  In(c, p))  
Action(Fly(p, from, to),  
    PRECOND: At(p, from)  $\wedge$  Plane(p)  $\wedge$  Airport(from)  $\wedge$  Airport(to)  
    EFFECT:  $\neg$  At(p, from)  $\wedge$  At(p, to))
```

The initial state and goals of the three problems are provided below:

Problem	Initial State and Goals
1	$\text{Init}(\text{At}(\text{C1}, \text{SFO}) \wedge \text{At}(\text{C2}, \text{JFK})$ $\wedge \text{At}(\text{P1}, \text{SFO}) \wedge \text{At}(\text{P2}, \text{JFK})$ $\wedge \text{Cargo}(\text{C1}) \wedge \text{Cargo}(\text{C2})$ $\wedge \text{Plane}(\text{P1}) \wedge \text{Plane}(\text{P2})$ $\wedge \text{Airport}(\text{JFK}) \wedge \text{Airport}(\text{SFO}))$ $\text{Goal}(\text{At}(\text{C1}, \text{JFK}) \wedge \text{At}(\text{C2}, \text{SFO}))$
2	$\text{Init}(\text{At}(\text{C1}, \text{SFO}) \wedge \text{At}(\text{C2}, \text{JFK}) \wedge \text{At}(\text{C3}, \text{ATL})$ $\wedge \text{At}(\text{P1}, \text{SFO}) \wedge \text{At}(\text{P2}, \text{JFK}) \wedge \text{At}(\text{P3}, \text{ATL})$ $\wedge \text{Cargo}(\text{C1}) \wedge \text{Cargo}(\text{C2}) \wedge \text{Cargo}(\text{C3})$ $\wedge \text{Plane}(\text{P1}) \wedge \text{Plane}(\text{P2}) \wedge \text{Plane}(\text{P3})$ $\wedge \text{Airport}(\text{JFK}) \wedge \text{Airport}(\text{SFO}) \wedge \text{Airport}(\text{ATL}))$ $\text{Goal}(\text{At}(\text{C1}, \text{JFK}) \wedge \text{At}(\text{C2}, \text{SFO}) \wedge \text{At}(\text{C3}, \text{SFO}))$
3	$\text{Init}(\text{At}(\text{C1}, \text{SFO}) \wedge \text{At}(\text{C2}, \text{JFK}) \wedge \text{At}(\text{C3}, \text{ATL}) \wedge \text{At}(\text{C4}, \text{ORD})$ $\wedge \text{At}(\text{P1}, \text{SFO}) \wedge \text{At}(\text{P2}, \text{JFK})$ $\wedge \text{Cargo}(\text{C1}) \wedge \text{Cargo}(\text{C2}) \wedge \text{Cargo}(\text{C3}) \wedge \text{Cargo}(\text{C4})$ $\wedge \text{Plane}(\text{P1}) \wedge \text{Plane}(\text{P2})$ $\wedge \text{Airport}(\text{JFK}) \wedge \text{Airport}(\text{SFO}) \wedge \text{Airport}(\text{ATL}) \wedge \text{Airport}(\text{ORD}))$ $\text{Goal}(\text{At}(\text{C1}, \text{JFK}) \wedge \text{At}(\text{C3}, \text{JFK}) \wedge \text{At}(\text{C2}, \text{SFO}) \wedge \text{At}(\text{C4}, \text{SFO}))$

Uninformed/Non-heuristic Planning Solution Search Analysis

Uninformed planning searches were performed on the 3 problems given, and metrics obtained in the different cases tested is documented below. The metrics were obtained using the following commands to ensure the output is written to file for referencing later:

```
python run_search.py -p 1 -s 1 2 3 4 5 6 7 >> uniformed_search_results_p1.txt
```

```
python run_search.py -p 2 -s 1 3 5 7 >> uniformed_search_results_p2.txt
```

```
python run_search.py -p 3 -s 1 3 5 7 >> uniformed_search_results_p3.txt
```

For Problem 1, the first seven search strategies were run as all the results came back in good time. For Problems 2 and 3, searches that took more than 10 minutes were aborted, which is why the outcome of such searches are not available in this report, most especially *Breadth First Tree Search*, *Depth Limited Search*, and *Recursive Best First Search*.

Below is a summary of the results obtained:

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
Breadth First Search	43	56	0.045	6	Yes
Breadth First Tree Search	1458	1459	1.411	6	Yes
Depth First Graph Search	12	13	0.011	12	No
Depth Limited Search	101	271	0.124	50	No
Uniform Cost Search	55	57	0.054	6	Yes
Recursive Best First Search	4229	4230	4.018	6	Yes

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
Greedy Best First Graph Search	7	9	0.007	6	Yes

Problem 1 Results

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
Breadth First Search	3343	4609	14.3301	9	Yes
Breadth First Tree Search	-	-	-	-	-
Depth First Graph Search	582	583	5.3835	575	No
Depth Limited Search	222719	2053741	1485.6577	50	No
Uniform Cost Search	4853	4855	19.5034	9	Yes
Recursive Best First Search	-	-	-	-	-
Greedy Best First Graph Search	998	1000	3.9971	21	No

Problem 2 Results

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
Breadth First Search	14663	18098	66.0125	12	Yes
Breadth First Tree Search	-	-	-	-	-
Depth First Graph Search	627	628	4.3186	596	No
Depth Limited Search	-	-	-	-	-
Uniform Cost Search	18223	18225	77.7799	12	Yes

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
Recursive Best First Search	-	-	-	-	-
Greedy Best First Graph Search	5578	5580	20.5859	22	No

Problem 3 Results

According to the results obtained from the 3 problems analyzed, it was observed that **Breadth First Search** and **Uniform Cost Search** are the only two uninformed search strategies that yield *optimal plans* within the 10-minute time limit. Optimal plans, in the case of the cargo problem means the goal is achieved with the least actions possible.

In Problems 1 & 2, **Greedy Best First Graph Search** performs the fastest based on the time elapsed when the search was performed; however, this wasn't the case in Problem 3, as **Depth First Graph Search** performed fastest in that situation. Overall, if we take the *execution speed* (time elapsed), and *memory usage* (number of node expansions) into consideration as a single unit, in the three searches, **Depth First Graph Search** seems to be the best choice across board.

In conclusion, if the determination of the optimal path is critical, **Breadth First Search** is the best strategy to use, as it performs faster and uses less memory than Uniform Cost Search.

Informed/Heuristic Planning Solution with A* Search Analysis

This section brings to fore the outcome of the metrics obtained when the A* planning search was adopted, with three different heuristics:

- A* search with h1 heuristic
- A* search with Ignore Preconditions heuristic
- A* search with Level Sum heuristic

Below is a summary of the results obtained:

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
A* search with h1	55	57	0.0624	6	Yes
A* search with Ignore Preconditions	41	43	0.0579	6	Yes
A* search with Levelsum	39	41	1.1893	6	Yes

Problem 1 Results

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
A* search with h1	4853	4855	14.006	9	Yes
A* search with Ignore Preconditions	1450	1452	5.146	9	Yes
A* search with Levelsum	1129	1131	316.816	9	Yes

Problem 2 Results

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length	Optimal?
A* search with h1	18223	18225	67.029	12	Yes
A* search with Ignore Preconditions	5040	5042	22.897	12	Yes
A* search with Levelsum	-	-	-	-	-

Problem 3 Results

As mentioned earlier in the uninformed search analysis, search strategies that take more than 10-minutes are aborted, thereby leading to no metrics obtained for such searches. This is evident in the *A* search with Levelsum*.

All the A* heuristics yield an optimal plan, however, a close comparison of the three heuristics reveals that **A* search with Ignore Preconditions heuristic** comes up as the fastest, in addition to coming up with an optimal plan. It should also be mentioned that *A* search with Levelsum heuristic* uses the least memory (number of node expansions).

Comparison of Informed and Uninformed Search Strategies

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length
Breadth First Search	43	56	0.045	6
A* search with Ignore Preconditions	41	43	0.0579	6

Problem 1 Results

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length
Breadth First Search	3343	4609	14.3301	9
A* search with Ignore Preconditions	1450	1452	5.146	9

Problem 2 Results

Search Strategy	# Node Expansions	# Goal Tests	Time Elapsed	Plan Length
Breadth First Search	14663	18098	66.0125	12
A* search with Ignore Preconditions	5040	5042	22.897	12

Problem 3 Results

The search strategies that generate optimal plans are *Breadth First Search*, *Uniform Cost Search*, and *A* Search* with all three heuristics.

As discussed earlier, *Depth First Graph Search* is faster and uses less memory than Uniform Cost Search for Uninformed search strategies. For informed search strategies, *A* Search with Ignore Preconditions heuristic* is the fastest and it uses minimal memory, therefore, the ultimate choice is between *Depth First Graph Search* and *A* Search with Ignore Preconditions heuristic*.

Looking at the tables above for the three problems under evaluation, it can be concluded that *A* Search with Ignore Preconditions heuristic* is the best overall choice for the Air Cargo problem.

Conclusion

The results provided above in the comparison of uninformed search strategies and informed search strategies with custom heuristics clearly illustrates the benefits of using the latter over the former, when searching for an optimal plan. The benefits are significant in terms of speed (time elapsed) and memory usage (number of node expansions).