Heuristic Analysis

PLANNING SEARCH FOR AIR CARGO TRANSPORTATION SYSTEM

Roberto Talamas | Artificial Intelligence, Udacity

Introduction

The project aims to develop a planning agent to solve deterministic logistic planning problems for an air cargo transportation system. Throughout the project I used a planning graph and a variety of automatic domain-independent heuristics using A* search and compare the performance against several uninformed non-heuristic search methods.

Planning problems

As part of the project we were given three planning problems within the air cargo logistic domain that share the same actions schema:

```
Action(Load(c, p, a),
      PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
      EFFECT: ¬ At(c, a) ∧ In(c, p))
Action(Unload(c, p, a),
      PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
      EFFECT: At(c, a) ∧ ¬ In(c, p))
Action(Fly(p, from, to),
      PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
      EFFECT: ¬ At(p, from) ∧ At(p, to))
```

The three air cargo problems and their initial states were the following:

```
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))


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))


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))
```

Non-heuristic planning solution searches

This section focuses on the analysis and comparison of uniformed search strategies. These type of search strategies only use information about the states that is provided in the problem definition. Given this information, they generate successor states and are able to distinguish if a given state is a goal state or a non-goal state.

As part of the analysis, three uninformed search strategies were compared: *Breadth First Search*, *Depth First Search* and *Uniform Cost Search.* Metrics measuring the speed, memory usage and optimality (yes, if a solution of optimal length is found, no otherwise) were collected for each one of the strategies for comparison purposes. The tables below summarize the results obtained for the three logistic problems

Problem 1 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
Bread First Search	43	56	0.1	6	Yes
Depth First Search	12	13	0.03	12	No
Uniform Cost Search	55	57	0.1	6	Yes

Problem 2 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
Bread First Search	3343	4609	11.7	9	Yes
Depth First Search	582	583	4.12	575	No
Uniform Cost Search	4853	4855	16.8	9	Yes

Problem 3 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
Bread First Search	14663	18098	58.2	12	Yes
Depth First Search	627	628	4.32	596	No
Uniform Cost Search	18223	18225	71.57	12	Yes

Results – Non-heuristic planning solutions

Given these three problem set, Breath First Search and Uniform Cost Search are the only two strategies that yield optimal solutions. In terms of speed, Depth First Search is the best strategy but it fails to generate an optimal path. *If finding the optimal path is critical*, Breath First Search would be the best option because it is able to find an optimal path and is slightly faster than Uniform Cost Search.

Heuristic planning solutions

In this sections we explore informed search strategies, strategies that uses problem-specific knowledge beyond what is defined in the initial problems, in order to find solutions for the problems states above. A* Search is used with three different heuristics. Speed, memory usage and optimality metrics are collected in order to compare the three heuristics. The tables below summarize the results of the analysis. Data could not be collected for A* Search with Level Sum heuristic for problem 3 since its execution time exceeded the ten minute threshold.

Problem 1 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
A* Search with h1 heuristic	55	57	.12	6	Yes
A* Search with Ignore Preconditions heuristic	41	43	.12	6	Yes
A* Search with Level Sum heuristic	11	13	1.21	6	Yes

Problem 2 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
A* Search with h1 heuristic	4853	4855	16.40	9	Yes
A* Search with Ignore Preconditions heuristic	1450	1452	6.06	9	Yes
A* Search with Level Sum heuristic	86	88	200.22	9	Yes

Problem 3 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
A* Search with h1 heuristic	18223	18225	74.85	12	Yes
A* Search with Ignore Preconditions heuristic	5040	5042	23.64	12	Yes
A* Search with Level Sum heuristic	-	-	-	-	-

Results – Heuristic planning solutions

Based on the results, A* Search with Ignore Preconditions heuristic is the best performing search strategy. Although, all three heuristics were able to find an optimal solutions to the problems, A* Search with Ignore Preconditions was able to find it faster and required less memory usage compared to the other two solutions. A* Search with Level Sum performed better in terms of memory usage but was unable to find an optimal solution for problem three within the allotted time.

Heuristic vs Non-heuristic solutions

Throughout the analysis the planning strategies that yielded optimal paths for all three problems are: Breath First Search, Uninformed Cost Search and A* Search with all three heuristics. In order to choose the best solutions for the air cargo problems it is worth comparing the best search strategies. As stated in the previous sections, Breath First Search was chosen as the best non-heuristic strategy and A* Search with Ignore Preconditions heuristics was chosen as the best heuristic solution. The two strategies are compared in the tables below against the three problems.

Problem 1 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
Breath First Search	43	56	0.1	6	Yes
A* Search with Ignore Preconditions heuristic	41	43	.12	6	Yes

Problem 2 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
Breath First Search	3343	4609	11.7	9	Yes
A* Search with Ignore Preconditions heuristic	1450	1452	6.06	9	Yes

Problem 3 results

Strategy	Node Expansions	Goal tests	Time elapsed (sec.)	Path Length	Optimal
Breath First Search	14663	18098	58.2	12	Yes
A* Search with Ignore Preconditions heuristic	5040	5042	23.64	12	Yes

Conclusion

According to the results shown above, it is evident that using informed search strategies yields better results in terms of optimality, speed and memory usage when compared to the uniformed search strategies. To conclude, A* Search with Ignore Preconditions heuristics would be the best choice overall to solve the three air cargo problems.