Artificial Intelligence Nanodegree

Implement a Planning Search - Heuristic Analysis

Ollie Graham
22nd October 2017

Documentation of non-heuristic planning solution searches.

Air Cargo P1

**** PROBLEM SETUP ****
Initial state for this problem is TTTTFFFFFFFF

Actions for this domain are:
  Load(C1, P1, JFK)
  Load(C1, P2, JFK)
  Load(C1, P1, SFO)
  Load(C1, P2, SFO)
  Load(C2, P1, JFK)
  Load(C2, P2, JFK)
  Load(C2, P1, SFO)
  Load(C2, P2, SFO)
  Unload(C1, P1, JFK)
  Unload(C1, P2, JFK)…
  …Unload(C1, P1, SFO)
  Unload(C1, P2, SFO)
  Unload(C2, P1, JFK)
  Unload(C2, P2, JFK)
  Unload(C2, P1, SFO)
  Unload(C2, P2, SFO)
  Fly(P1, JFK, SFO)
  Fly(P2, JFK, SFO)
  Fly(P1, SFO, JFK)
  Fly(P2, SFO, JFK)

Fluents in this problem are:
  At(C1, SFO)
  At(C2, JFK)
  At(P1, SFO)
  At(P2, JFK)
  At(C2, SFO)
  In(C2, P1)
  In(C2, P2)
  At(C1, JFK)
  In(C1, P1)
  In(C1, P2)
  At(P1, JFK)
  At(P2, SFO)

Goal requirement for this problem are:
  At(C1, JFK)
  At(C2, SFO)

Optimal Solution

An optimal solution for this problem is the following 6 steps:

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

Solving Air Cargo Problem 1 using breadth_first_search...

Expansions	Goal Tests	New Nodes
43	56	180

Plan length: 6 Time elapsed in seconds: 0.030681947944685817

This is an optimal solution taking 6 steps to reach the goal states.

Solving Air Cargo Problem 1 using depth_first_graph_search...

Expansions	Goal Tests	New Nodes
21	22	84

Plan length: 20 Time elapsed in seconds: 0.0162638210458681

This is not an optimal solution taking 20 steps vs. the optimal length of 6. It was the quickest solution to complete of the three I tested however.

Solving Air Cargo Problem 1 using uniform_cost_search...

Expansions	Goal Tests	New Nodes
55	57	224

Plan length: 6 Time elapsed in seconds: 0.04005855100695044

This is an optimal solution given a plan length of 6, the solution returned marginally slower than the solution using breadth first search.

Solving Air Cargo Problem 1 using astar_search with h_ignore_preconditions...

Expansions	Goal Tests	New Nodes
41	43	170

Plan length: 6 Time elapsed in seconds: 0.03826236599979893

This is an optimal solution, taking 6 steps. This heuristic has similar expansions, goal tests and new nodes to breadth first search and performed marginally better than the h_1 fake heuristic.

Solving Air Cargo Problem 1 using astar_search with h_pg_levelsum...

Expansions	Goal Tests	New Nodes
11	13	50

Plan length: 6 Time elapsed in seconds: 0.9867048759997488

This is also an optimal solution with 6 steps to a solution. The planning graph heuristic performs far better than any of the previous search / heuristic combinations with only 11 expansions, 13 goal tests and 50 new nodes. The heuristic took much longer to return a result, returning the solution in just under one second.

Air Cargo P2

**** PROBLEM SETUP ****
Initial state for this problem is TTTTTTFFFFFFFFFFFFFFFFFFFFFF

Actions for this domain are:
Load(C1, P1, JFK)
Load(C1, P2, JFK)
Load(C1, P3, JFK)
Load(C1, P1, SFO)
Load(C1, P2, SFO)
Load(C1, P3, SFO)
Load(C1, P1, ATL)
Load(C1, P2, ATL)
Load(C1, P3, ATL)
Load(C2, P1, JFK)
Load(C2, P2, JFK)
Load(C2, P3, JFK)
Load(C2, P1, SFO)
Load(C2, P2, SFO)
Load(C2, P3, SFO)
Load(C2, P1, ATL)
Load(C2, P2, ATL)
Load(C2, P3, ATL)
Load(C3, P1, JFK)
Load(C3, P2, JFK)
Load(C3, P3, JFK)
Load(C3, P1, SFO)
Load(C3, P2, SFO)
Load(C3, P3, SFO)
Load(C3, P1, ATL)
Load(C3, P2, ATL)
Load(C3, P3, ATL)
Unload(C1, P1, JFK)
Unload(C1, P2, JFK)
Unload(C1, P3, JFK)
Unload(C1, P1, SFO)
Unload(C1, P2, SFO)
Unload(C1, P3, SFO)
Unload(C1, P1, ATL)
Unload(C1, P2, ATL)
Unload(C1, P3, ATL)…
…Unload(C2, P1, JFK)
Unload(C2, P2, JFK)
Unload(C2, P3, JFK)
Unload(C2, P1, SFO)
Unload(C2, P2, SFO)
Unload(C2, P3, SFO)
Unload(C2, P1, ATL)
Unload(C2, P2, ATL)
Unload(C2, P3, ATL)
Unload(C3, P1, JFK)
Unload(C3, P2, JFK)
Unload(C3, P3, JFK)
Unload(C3, P1, SFO)
Unload(C3, P2, SFO)
Unload(C3, P3, SFO)
Unload(C3, P1, ATL)
Unload(C3, P2, ATL)
Unload(C3, P3, ATL)
Fly(P1, JFK, SFO)
Fly(P2, JFK, SFO)
Fly(P3, JFK, SFO)
Fly(P1, JFK, ATL)
Fly(P2, JFK, ATL)
Fly(P3, JFK, ATL)
Fly(P1, SFO, JFK)
Fly(P2, SFO, JFK)
Fly(P3, SFO, JFK)
Fly(P1, SFO, ATL)
Fly(P2, SFO, ATL)
Fly(P3, SFO, ATL)
Fly(P1, ATL, JFK)
Fly(P2, ATL, JFK)
Fly(P3, ATL, JFK)
Fly(P1, ATL, SFO)
Fly(P2, ATL, SFO)
Fly(P3, ATL, SFO)

Fluents in this problem are:
At(C1, SFO)
At(C2, JFK)
At(C3, ATL)
At(P1, SFO)
At(P2, JFK)
At(P3, ATL)

In(C1, P1)
In(C1, P2)
In(C1, P3)
In(C2, P1)
In(C2, P2)
In(C2, P3)
In(C3, P1)
In(C3, P2)
In(C3, P3)
At(C1, JFK)
At(C1, ATL)…
…At(C2, SFO)
At(C2, ATL)
At(C3, SFO)
At(C3, JFK)
At(P1, JFK)
At(P1, ATL)
At(P2, SFO)
At(P2, ATL)
At(P3, SFO)
At(P3, JFK)

Goal requirement for this problem are:
At(C1, JFK)
At(C2, SFO)
At(C3, SFO)

Optimal Solution

An optimal solution for this problem is the following 9 steps:

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

Solving Air Cargo Problem 2 using breadth_first_search...

Expansions	Goal Tests	New Nodes
3343	4609	30509

Plan length: 9 Time elapsed in seconds: 14.866952317068353

This is an optimal plan with a length of 9 steps. Again this search returned the solution much slower than depth first search, but still within a reasonable amount of time.

Solving Air Cargo Problem 2 using depth_first_graph_search...

Expansions	Goal Tests	New Nodes
624	625	5602

Plan length: 619 Time elapsed in seconds: 3.7977887109154835

This is not an optimal plan with a length of 619 steps. This is again the quickest solution to complete in 3.8 seconds.

Solving Air Cargo Problem 2 using uniform_cost_search...

Expansions	Goal Tests	New Nodes
4605	4607	41839

Plan length: 9 Time elapsed in seconds: 14.665838729008101

This is an optimal solution, taking 9 steps to complete. A* search returned the solution marginally quicker than breadth first search for this problem, but was much slower than depth first search.

Solving Air Cargo Problem 2 using depth_limited_search...

Expansions	Goal Tests	New Nodes
222719	2053741	2054119

Plan length: 50 Time elapsed in seconds: 1041.550559644005

I ran this search out of interest for this problem to see how it would perform – in short, it performed terribly, so I did not elect to run it for the other problems.

This is NOT an optimal solution taking 50 steps to complete with continual loading and unloading of C1 onto P1 at SFO. It also takes a very long time (>1,000 secs) to compute the solution relative to the other searches.

Solving Air Cargo Problem 2 using astar_search with h_ignore_preconditions...

Expansions	Goal Tests	New Nodes
1311	1313	11989

Plan length: 9 Time elapsed in seconds: 4.165279638999891

This is an optimal solution taking 9 steps. The solution also performs better than all but the depth first search solution thus far.

Solving Air Cargo Problem 2 using astar_search with h_pg_levelsum...

Expansions	Goal Tests	New Nodes
74	76	720

Plan length: 9 Time elapsed in seconds: 153.60466069699942

This is an optimal solution and outperformed the other search / heuristic combinations by a large margin for this problem. The downside is a longer, but still reasonable time to compute the solution.

Air Cargo P3

**** PROBLEM SETUP ****
Initial state for this problem is TTTTTTFFFFFFFFFFFFFFFFFFFFFFFFFF

Actions for this domain are:

Load(C1, P1, JFK)
Load(C1, P2, JFK)
Load(C1, P1, SFO)
Load(C1, P2, SFO)
Load(C1, P1, ATL)
Load(C1, P2, ATL)
Load(C1, P1, ORD)
Load(C1, P2, ORD)
Load(C2, P1, JFK)
Load(C2, P2, JFK)
Load(C2, P1, SFO)
Load(C2, P2, SFO)
Load(C2, P1, ATL)
Load(C2, P2, ATL)
Load(C2, P1, ORD)
Load(C2, P2, ORD)
Load(C3, P1, JFK)
Load(C3, P2, JFK)
Load(C3, P1, SFO)
Load(C3, P2, SFO)
Load(C3, P1, ATL)
Load(C3, P2, ATL)
Load(C3, P1, ORD)
Load(C3, P2, ORD)
Load(C4, P1, JFK)
Load(C4, P2, JFK)
Load(C4, P1, SFO)
Load(C4, P2, SFO)
Load(C4, P1, ATL)
Load(C4, P2, ATL)
Load(C4, P1, ORD)
Load(C4, P2, ORD)
Unload(C1, P1, JFK)
Unload(C1, P2, JFK)
Unload(C1, P1, SFO)
Unload(C1, P2, SFO)
Unload(C1, P1, ATL)
Unload(C1, P2, ATL)
Unload(C1, P1, ORD)
Unload(C1, P2, ORD)
Unload(C2, P1, JFK)…
…Unload(C2, P2, JFK)
Unload(C2, P1, SFO)
Unload(C2, P2, SFO)
Unload(C2, P1, ATL)
Unload(C2, P2, ATL)
Unload(C2, P1, ORD)
Unload(C2, P2, ORD)
Unload(C3, P1, JFK)
Unload(C3, P2, JFK)
Unload(C3, P1, SFO)
Unload(C3, P2, SFO)
Unload(C3, P1, ATL)
Unload(C3, P2, ATL)
Unload(C3, P1, ORD)
Unload(C3, P2, ORD)
Unload(C4, P1, JFK)
Unload(C4, P2, JFK)
Unload(C4, P1, SFO)
Unload(C4, P2, SFO)
Unload(C4, P1, ATL)
Unload(C4, P2, ATL)
Unload(C4, P1, ORD)
Unload(C4, P2, ORD)
Fly(P1, JFK, SFO)
Fly(P2, JFK, SFO)
Fly(P1, JFK, ATL)
Fly(P2, JFK, ATL)
Fly(P1, JFK, ORD)
Fly(P2, JFK, ORD)
Fly(P1, SFO, JFK)
Fly(P2, SFO, JFK)
Fly(P1, SFO, ATL)
Fly(P2, SFO, ATL)
Fly(P1, SFO, ORD)
Fly(P2, SFO, ORD)
Fly(P1, ATL, JFK)
Fly(P2, ATL, JFK)
Fly(P1, ATL, SFO)
Fly(P2, ATL, SFO)
Fly(P1, ATL, ORD)
Fly(P2, ATL, ORD)

Fly(P1, ORD, JFK)
Fly(P2, ORD, JFK)
Fly(P1, ORD, SFO)…
…Fly(P2, ORD, SFO)
Fly(P1, ORD, ATL)
Fly(P2, ORD, ATL)

Fluents in this problem are:

At(C1, SFO)
At(C2, JFK)
At(C3, ATL)
At(C4, ORD)
At(P1, SFO)
At(P2, JFK)
In(C1, P1)
In(C1, P2)
In(C2, P1)
In(C2, P2)
In(C3, P1)
In(C3, P2)
In(C4, P1)
In(C4, P2)
At(C1, JFK)
At(C1, ATL)…
…At(C1, ORD)
At(C2, SFO)
At(C2, ATL)
At(C2, ORD)
At(C3, SFO)
At(C3, JFK)
At(C3, ORD)
At(C4, SFO)
At(C4, JFK)
At(C4, ATL)
At(P1, JFK)
At(P1, ATL)
At(P1, ORD)
At(P2, SFO)
At(P2, ATL)
At(P2, ORD)

Goal requirement for this problem are:

At(C1, JFK)
At(C2, SFO)
At(C3, JFK)
At(C4, SFO)

Optimal Solution

An optimal solution for this problem is the following 12 steps:

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)

Solving Air Cargo Problem 3 using breadth_first_search...

Expansions	Goal Tests	New Nodes
14663	18098	129631

Plan length: 12 Time elapsed in seconds: 121.44572576403152

This is an optimal solution taking 12 steps. This is the slowest search to return a solution for this problem.

Solving Air Cargo Problem 3 using depth_first_graph_search...

Expansions	Goal Tests	New Nodes
408	409	3364

Plan length: 392 Time elapsed in seconds: 1.9219510019756854

This is not an optimal solution, taking 392 steps to complete. It remains the quickest search to return a result however.

Solving Air Cargo Problem 3 using uniform_cost_search...

Expansions	Goal Tests	New Nodes
16961	16963	149117

Plan length: 12 Time elapsed in seconds: 57.008438766002655

This is an optimal solution taking 12 steps to complete. A* search returned the solution in the shortest time for problem 3 thus far.

Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions	Goal Tests	New Nodes
4444	4446	39227

Plan length: 12 Time elapsed in seconds: 14.541172215999723

This is an optimal solution for the problem and the best performance of all except for depth first search thus far.

Solving Air Cargo Problem 3 using astar_search with h_pg_levelsum...

Expansions	Goal Tests	New Nodes
229	231	2081

Plan length: 12 Time elapsed in seconds: 157.690003058

This is an optimal solution. The heuristic had much lower expansions, goal tests and new nodes than the other search / heuristic combinations, but took much longer to (order of 10x) as long to return the solution.

Planning Search Results Summary

Problem	Search & Heuristic Type	Expansions	Goal Tests	New Nodes	Plan Length	Time (secs)
Air Cargo Problem 1	breadth_first_search	43	56	180	6	0.030682
	depth_first_graph_search	21	22	84	20	0.016264
	uniform_cost_search	55	57	224	6	0.040059
	astar_search_with_h_ignore_preconds	41	43	170	6	0.038262
	astar_search_with_h_pg_levelsum	11	13	50	6	0.986705
Air Cargo Problem 2	breadth_first_search	3,343	4,609	30,509	9	14.866952
	depth_first_graph_search	624	625	5,602	619	3.797789
	depth_limited_search	222,719	2,053,741	2,054,119	50	1041.55056
	uniform_cost_search	4,605	4,607	41,839	9	14.665839
	astar_search_with_h_ignore_preconds	1,311	1,313	11,989	9	4.165280
	astar_search_with_h_pg_levelsum	74	76	720	9	153.604660
Air Cargo Problem 3	breadth_first_search	14,663	18,098	129,631	12	121.445726
	depth_first_graph_search	408	409	3,364	392	1.921951
	uniform_cost_search	16,961	16,963	149,117	12	57.008439
	astar_search_with_h_ignore_preconds	4,444	4,446	39,227	12	14.541172
	astar_search_with_h_pg_levelsum	322	324	2,964	12	192.921529

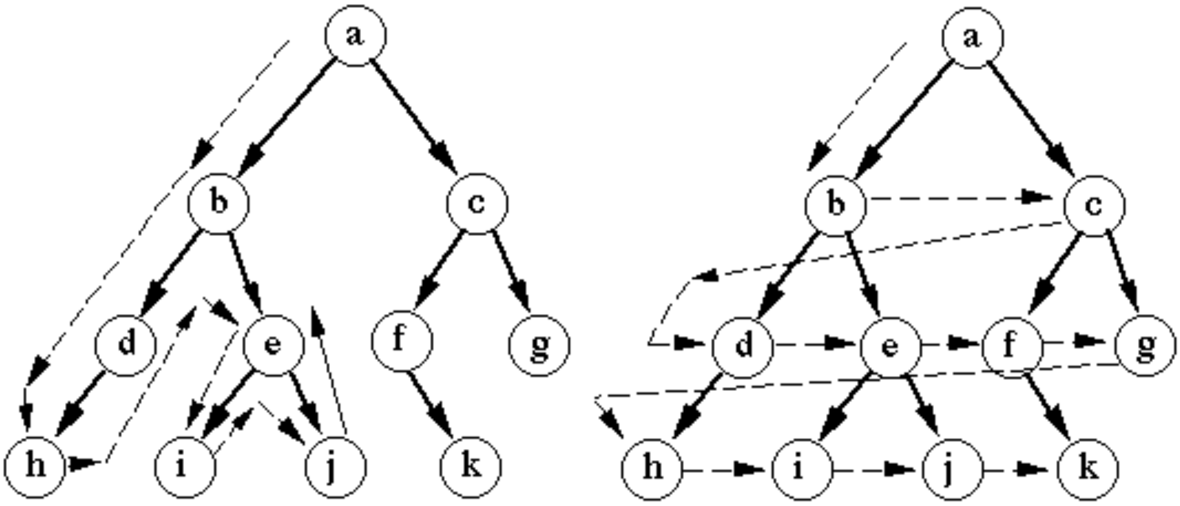
Comparison of Uninformed Planning Algorithm Results

For uninformed search I selected breath first, depth first graph and uniform cost searches for each of the Air Cargo problems. Please refer to the planning search results summary table for results of each problem / search scenario.

In terms of time to return a solution depth first search was the quickest for each of the three problems for any of the searches. Depth first search also had the fewest expansions, goal tests and new nodes of any of the searches.

Where depth first search falls down however, is that it does not generally return an optimal solution, as is evident in the table, returning solutions that were many steps longer than the optimal solution.

The following diagram shows how depth first search progresses through the state nodes compared to breadth first search.

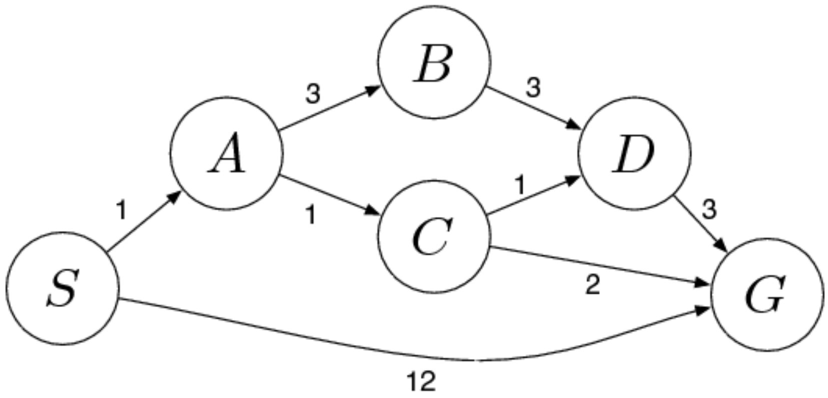


Depth-first search Breadth-first search

Once depth first search reaches a goal state it returns the solution without exploring further nodes on the same level, as is the case with breadth first search. This is why depth first search will not return an optimal solution.

Breadth first search will as a result expand many more nodes than depth first search as result, but it will always return an optimal solution, as is evidenced in the results table.

Uniform cost search (UCS) finds the lowest “cost” path from a source state (S) to the goal state (g) as illustrated in the following diagram.



UCS is the best non - heuristic search method and utilizes [Dijkstra's algorithm](#) to find the shortest path between nodes in a graph.

UCS expands more nodes than breadth first search because even though it might find a solution on the current frontier it continues to search until the solution returned is the lowest cost solution of all the remaining paths on the frontier. This is also evidenced in the results table.

For these problems breadth first search performed the best, but in reality, where there would certainly be associated costs for each leg of flying a plane with cargo uniform cost search would be the best non - heuristic search method to apply.

Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

A* search is essentially the same as UCS, but with the addition of an heuristic which estimates the cheapest path from the current node to the goal state.

Specifically, A* selects the path that minimizes:

$$f(n) = g(n) + h(n)$$

The ignore preconditions heuristic estimates the minimum number of actions that must be carried out from the current state in order to satisfy all of the goal conditions by ignoring the preconditions required for an action to be executed.

The level-sum heuristic uses a planning graph representation of the state space to estimate a sum of all actions that must be carried out from the current state in order to satisfy each individual goal condition.

For each of the Air Cargo problems we can see that the level sum heuristic expands fewer nodes, has fewer goal tests and has fewer new nodes than the ignore preconditions heuristic. This is because the level-sum heuristic is a much better approximation of the true cost of reaching the goal state.

Level-sum is a better approximation of the true cost of reaching the goal state precisely because it doesn't ignore preconditions which the planning algorithm actually has to consider.

Ingoring preconditions does outperform the level-sum heuristic in terms of speed of returning a solution in all three problem cases, by up to a factor of ~65 in the most complex problem, problem 3. So, if time were a consideration then this heuristic might be preferred.

What was the best heuristic used for these problems?

The best heuristic I used for these problems is the level-sum planning graph heuristic which outperformed the ignore preconditions heuristic by some margin in terms of exploration of the graph, only failing to beat it in the time required to reach the solution.

It also outperformed all of the non - heuristic searches in terms exploration of the graph, but again failed to outperform any of them in terms of time taken to return a solution.

The non - heuristic searches are not guided towards a path in the graph which is the most promising in terms of reaching a goal state, and are thus termed uninformed searches. This means that they inherently need to explore more of the graph to reach the optimal solution (if at all). By using the level-sum heuristic A* star search has a very good approximation of which path will lead to the optimal goal state without having to expand much of the graph.