David Tsukiyama

Implement a Planning Search

In this project we solve deterministic logistics planning problems for an Air Cargo transport system using a planning search agent. We are given 3 classical PDDL problems. We are given an action schema and three initial state and goals.

Task 1

Uninformed planning searches for `air_cargo_p1`, `air_cargo_p2`, and `air_cargo_p3`; provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm.

Task 2

Run A* planning searches using the heuristics you have implemented on `air_cargo_p1`, `air_cargo_p2` and `air_cargo_p3`. Provide metrics on number of node expansions required, number of goal tests, time elapsed, and optimality of solution for each search algorithm

Task 1 results: Non-heuristic planning solution searches

Breadth First Search is the recommended strategy. Both Uniform Cost Search and Breadth First Search obtain optimal results in regards to plan length. However, it seems on balance Breadth First Search uses less memory and is about as quick as Uniform Cost search. Greedy Best First Graph Search is also a good candidate if we disregard the fact it does not produce the optimal plan length. It makes up for this by using significantly less memory and is significantly faster than both Breadth First and Uniform Cost Search.

Air Cargo Problem 1

search	expansion	time	Plan length	Optimal
breadth_first_search	43	0.024	6	Yes
breadth_first_tree_search	1458	0.701	6	Yes
depth_first_graph_search	21	0.012	20	No
depth_limited_search	101	0.094	50	No
uniform_cost_search	55	0.029	6	Yes

recursive_best_first_search h_1	4229	2.058	6	Yes
greedy_best_first_graph_search h_1	7	0.004	6	Yes

Air Cargo Problem 2

search	expansion	time	Plan length	Optimal
breadth_first_search	3342	10.416	9	Yes
breadth_first_tree_search	-	-	-	-
depth_first_graph_search	624	2.754	619	No
depth_limited_search	222719	758.731	50	No
uniform_cost_search	4853	9.234	9	Yes
recursive_best_first_search h_1	-	-	-	-
greedy_best_first_graph_search h_1	998	1.903	15	No

Air Cargo Problem 3

search	expansion	time	Plan length	Optimal
breadth_first_search	14663	79.218	12	Yes
breadth_first_tree_search	-	-	-	-
depth_first_graph_search	408	1.354	392	No
depth_limited_search	-	-	-	-
uniform_cost_search	18223	41.094	12	Yes
recursive_best_first_search h_1	-	-	-	-
greedy_best_first_graph_search h_1	5578	12.766	22	No

Task 2 results: Heuristic planning solution searches

The optimum heuristic planning search solution would be A-Star Search with Ignore Preconditions. A-Star with Level-sum uses less memory, but takes significantly more time; took longer than 10 minutes in the case of problem 3. Indeed Ignore Preconditions is the quickest here.

Air Cargo Problem 1

search	expansion	time	Plan length	Optimal
astar_search with h_1	55	0.085	6	Yes
astar_search with h_ignore_preconditions	41	0.033	6	Yes
astar_search with h_pg_levelsum	11	0.737	6	Yes

Air Cargo Problem 2

search	expansion	time	Plan length	Optimal
astar_search with h_1	4853	9.247	9	Yes
astar_search with h_ignore_preconditions	1450	3.245	9	Yes
astar_search with h_pg_levelsum	86	136.059	9	Yes

Air Cargo Problem 3

search	expansion	time	Plan length	Optimal
astar_search with h_1	18223	41.29267481	12	Yes
astar_search with h_ignore_preconditions	5040	12.7893497	12	Yes
astar_search with h_pg_levelsum	-	-	-	-

Conclusion: Best Overall Search Solution

In our uninformed search we decided the optimal solution was Breadth First Search. Breadth-First search expands shortest paths first so wherever the goal is it will find it by examining no longer paths, making it optimal [1]. In our informed search we decided that A-Star Search with Ignore Preconditions was our optimal solution. Head to head, A-Star Search with Ignore Preconditions outperforms Breadth First Search, it is significantly quicker and uses less memory. This is because informed search, the heuristics, supplies information on the states, which allows informed search to be more efficient. Given an optimistic h function A-Star search will find the lowest cost path, which seems to be the case here [2].

Uninformed vs Informed Search

search	expansion	time	Plan length	Optimal	Problem 1
breadth_first_search	43	0.024	6	Yes	1
astar_search with h_ignore_preconditions	41	0.033	6	Yes	1
breadth_first_search	3342	10.416	9	No	2
astar_search with h_ignore_preconditions	1450	3.245	9	Yes	2
breadth_first_search	14663	79.218	12	Yes	3
astar_search with h_ignore_preconditions	5040	12.7893497	12	Yes	3

References

1. Peter Norvig, Artificial Intelligence Nanodegree, Lesson 8 Search, video 23: Search Comparison 1.

2. Peter Norvig, Artificial Intelligence Nanodegree, Lesson 8 Search, video 32: A* Search 5.

Addendum

Action 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))
```

Problem 1

```
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))
```

Problem 2

```
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)
```

Problem 3

```
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))
```

Air Cargo Problem 1 Optimum Path

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

Air Cargo Problem 2 Optimum Path

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

Air Cargo Problem 3 Optimum Path

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)