Cargo Planning Problem Analysis

This project tries to solve deterministic logistics planning problems for an Air Cargo transport system using a planning search agent. We use uninformed search and heuristic-based search and compare the results for 3 given planning problems.

Problems

There are 3 problems to solve. The initial and goal states for the problems are:

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))

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))

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))

The Action schema for all 3 problems is:

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))

Optimal solutions

Possible optimal (based on plan length) solution for problem 1:

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

Optimal plan length is 6.

Possible optimal (based on plan length) solution for problem 2:

LOAD(C1,P1,SFO)
FLY(P1,SFO,JFK)
UNLOAD(C1,P1,JFK)
Load(C2,P2,JFK)
FLY(P2,JFK,SFO)
UNLOAD(C2,P2,SFO)
LOAD(C3,P3,ATL)
FLY(P3,ATL,SFO)
UNLOAD(C2,P3,SFO)

Optimal plan length is 9.

Possible optimal (based on plan length) solution for problem 3:

LOAD(C1,P1,SFO)
FLY(P1,SFO,ATL)
LOAD(C3,P1,ATL)
FLY(P1,ATL,JFK)
UNLOAD(C1,P1,JFK)
UNLOAD(C3,P1,JFK)
LOAD(C2,P2,JFK)
FLY(P2,JFK,ORD)
LOAD(C4,P2,ORD)
FLY(P2,ORD,SFO)
UNLOAD(C2,P2,SFO)
UNLOAD(C4,P2,SFO)

Optimal plan length is 12.

Uninformed searches

I ran the 7 uninformed search strategies for the 3 problems and listed the results in the following tables to compare the search results based on the metrics. For the searches which took more than 10 minutes there are no results listed. Also the searches which have **Plan length** equal to the optimal solution are considered **Optimal** searches in the table.

To compare the results, we can consider the **Plan length** as a metric of optimality, **Time** as a metric of performance and **Expansions** as a metric of memory usage. Depending on how we assign importance to each metric, different searches can be considered better for a given application.

For these problems, I believe that **Plan length** is the most important metric because it directly affects physical operations of the field. **Time** and **Expansion** are computational resources and between them I give more weight to **Time** assuming planning time could be critical and will also assume enough amount of computer memory is available for the application as long as it is reasonable. The result could be different if one considers different importance weights for metrics.

Comparing values for Problem 1, among the optimal searches, the best searches are 1- Greedy best first search, 2- Breadth first search and 3- Uniform cost search.

Comparing values for Problem 2, among the optimal searches, although they are very close, but the best searches are 1- Uniform cost search and 2- Breadth first search and.

Comparing values for Problem 3, among the optimal searches, the best searches are 1- Uniform cost search and 2- Breadth first search.

Voting among the best performing searches for the 3 problems, **Uniform cost search** wins the best performing search.

Based on the course videos, among the available searches we used here, only Uniform cost search and Breadth first search are Complete and Optimal, meaning they are guaranteed to find the goal with shortest path. Our results confirms this, because only these 2 searches find the optimal path in all cases.

Problem 1 - Uninformed searches

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
Breadth first search	43	56	180	6	0.03111	Yes
Breadth first tree search	1458	1459	5960	6	0.9585	Yes
Depth first graph search	12	13	48	12	0.0080	No
Depth limited search	101	271	414	50	0.09673	No
Uniform cost search	55	57	224	6	0.0385	Yes
Recursive best first search	4229	4230	17029	6	2.8106	Yes
Greedy best first graph search	7	9	28	6	0.0050	Yes

Problem 2 - Uninformed searches

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
Breadth first search	2844	40333	23627	9	11.7368	Yes
Breadth first tree search	?	?	?	?	?	?
Depth first graph search	44	45	275	38	0.0903	No
Depth limited search	?	?	?	?	?	?
Uniform cost search	4281	4283	35054	9	10.0260	Yes
Recursive best first search	?	?	?	?	?	?
Greedy best first graph search	478	480	3646	19	1.1182	No

Problem 3 - Uninformed searches

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
Breadth first search	14663	18098	129631	12	104.0059	Yes
Breadth first tree search	?	?	?	?	?	?
Depth first graph search	627	628	5167	596	3.2801	No
Depth limited search	?	?	?	?	?	?
Uniform cost search	18150	18152	159029	12	59.1173	Yes
Recursive best first search	?	?	?	?	?	?
Greedy best first graph search	5389	5391	47596	26	17.6213	No

Heuristic based searches

I ran the 3 heuristic based search strategies for the 3 problems and listed the results in the following tables to compare the search results based on the metrics. The searches which have **Plan length** equal to the optimal solution are considered **Optimal** searches in the table.

To compare the results, we can consider the **Plan length** as a metric of optimality, **Time** as a metric of performance and **Expansions** as a metric of memory usage.

Comparing values for Problem 1, among the optimal searches, the best searches are 1- A* search with ignore preconditions, 2- A* search with h1 and 3- A* search with levelsum.

Comparing values for Problem 2, among the optimal searches, the best searches are 1- A* search with ignore preconditions, 2- A* search with h1 and 3- A* search with levelsum .

Comparing values for Problem 3, among the optimal searches, the best searches are 1- A* search with ignore preconditions and 2- A* search with h1 .

Based on the course material, A* search is guaranteed to find shortest cost path if the heuristics is admissible which is the case for ignore precondition and levelsum heuristics. So it makes sense that both of the searches find the optimal path (except the case which we aborted the search because it took long time).

Problem 1- Heuristic searches

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
A* search with h1	55	57	224	6	0.05131	Yes
A* search with ignore preconditions	41	43	170	6	0.0382	Yes
A* search with levelsum	11	13	50	6	0.9733	Yes

Problem 2 - Heuristic searches

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
A* search with h1	4281	4283	35054	9	11.1479	Yes
A* search with ignore preconditions	1272	1274	10732	9	3.8083	Yes
A* search with levelsum	172	174	1314	9	182.9656	No

Problem 3 - Heuristic searches

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
A* search with h1	18150	18152	159029	12	60.7601	Yes
A* search with ignore preconditions	5038	5040	44926	12	17.0623	Yes
A* search with levelsum	?	?	?	?	?	?

Comparing best of uninformed and heuristic searches

Comparing metrics in tables below, **A* search with ignore preconditions** wins in all 3 problems.

Problem 1

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
Uniform cost search	55	57	224	6	0.0385	Yes
A* search with ignore preconditions	41	43	170	6	0.0382	Yes

Problem 2

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
Uniform cost search	4281	4283	35054	9	10.0260	Yes
A* search with ignore preconditions	1272	1274	10732	9	3.8083	Yes

Problem 3

Search type	Expansions	Goal tests	New nodes	Plan length	Time (seconds)	Optimal?
Uniform cost search	18150	18152	159029	12	59.1173	Yes
A* search with ignore preconditions	5038	5040	44926	12	17.0623	Yes

Conclusion

As the tests show, **A* search with ignore preconditions** has the best results among the others. Going back to theory, uninformed searches do their search by searching the state space looking for the goal state without considering any additional information from domain.
On the other hand heuristic based searches which in our case will be obtained by relaxing the problem description, gives us an admissible heuristic which can be used in better search algorithm like A* to find the goal state more efficiently.
The results matches what theory taught us.
Remembering the course material on search, A* search combines the best parts of greedy search, which explores a small number of nodes in many cases, and uniform cost search, which is guaranteed to find the shortest path. The result of our tests confirm what learned from the course.