Analysis of Heuristic and Non-Heuristic Planning Search Agents

This document analyzes the planning search agents which are used to solve deterministic logistics planning problems for an Air Cargo transport system.

Problem Setup:

Given 3 Problems in Air Cargo Domain with below schema and initial & goal states.

Air Cargo 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))
```

Initial and Goal States:

Problem1

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))

Problem2

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))

Problem3

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))

Analysis of Heuristic and Non-Heuristic Planning Search Agents

Above goal states can be reached using plans with different lengths. But below one is the optimal plan with lengths Problem1 Cargo = 6 ,Problem2 Cargo = 9 and Problem3 Cargo = 12.

Problem1
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem2
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Problem3
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Analysis of Heuristic and Non-Heuristic Planning Search Agents

Performance of Uninformed Non-Heuristic search agents:

We used different uninformed search agents which doesn't hold any additional information about the states. They perform the blind search by generating the successive child nodes ,comparing with the goal states by performing goal state test.

Below is the results of three cargo problems using different uninformed search agents.

	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Problem1						
Breadth_First_Search	43	56	180	0.06	6	Yes
Breadth_First_Tree_Search	1458	1459	5960	1.36	6	Yes
Depth_First_Graph_Search	21	22	84	0.02	20	No
Depth_Limited_Search	101	271	414	0.1	50	No
Uniform_Cost_Search	55	57	224	0.04	6	Yes
Recursive_Best_First_Search H_1	4229	4230	17023	3.34	6	Yes
Greedy_Best_First_Graph_Search H_1	7	9	28	0.005	6	Yes

	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Problem2						
Breadth_First_Search	3343	4609	30509	15.26	9	Yes
Depth_First_Graph_Search	624	625	5602	3.87	619	No
Depth_Limited_Search	222719	2053741	2054119	1036.86	50	No
Uniform_Cost_Search	4853	4855	44041	13.61	9	Yes
Greedy_Best_First_Graph_Search H_1	998	1000	8982	2.83	21	No

Analysis of Heuristic and Non-Heuristic Planning Search Agents

	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Problem3						
Breadth_First_Search	14663	18098	129631	118.77	12	Yes
Depth_First_Graph_Search	408	409	3364	2	392	No
Uniform_Cost_Search	18223	18225	159618	60.66	12	Yes
Greedy_Best_First_Graph_Search H_1	5578	5580	49150	18.03	22	No

Note:For Problem2,we have excluded Breadth_First_Tree_Search ,Recursive_Best_First_Search H_1 and Greedy_Best_First_Graph_Search H_1 as their collection time is exceeding 15min.
Same is the reason for excluding
Breadth_First_Tree_Search,Depth_Limited_Search,Recursive_Best_First_Search H_1 and Greedy_Best_First_Graph_Search H_1 for Problem3.

Desired Criteria of an agent:
Any agent which performs less number of expansions,goal tests ,generates less number of new nodes and takes less time is desirable.

Problem1:

- Breadth_First_Search,Breadth_First_Tree_Search,Uniform_Cost_Search, Recursive_Best_First_Search H_1,Greedy_Best_First_Graph_Search H_1 are optimal.
- Of these, Breadth_First_Search,Uniform_Cost_Search and Greedy_Best_First_Graph_Search H_1 satisfies the desired agent criteria.
- Even though Depth_First_Graph_Search and Depth_Limited_Search takes less time and memory but doesn't provide optimal solution.
- Breadth_First_Tree_Search and Recursive_Best_First_Search H_1 performs more number of expansions,goal tests and generates more number of new nodes even though they provide optimal plan.

Problem2:

- Breadth_First_Search and Uniform_Cost_Search are optimal but they generates more number of expansions ,goal tests and new nodes compared to Depth_First_Graph_Search.

Analysis of Heuristic and Non-Heuristic Planning Search Agents

- Even though Depth_First_Graph_Search takes less time and memory but doesn't provide optimal solution.
- Depth_Limited_Search doesn't satisfy any single criteria of desired agent.
- Even though Greedy_Best_First_Graph_Search H_1 takes less time but it doesn't provide optimal solution.

Problem3:

- Breadth_First_Search and Uniform_Cost_Search are optimal but they generates more number of expansions ,goal tests and new nodes compared to Depth_First_Graph_Search.
- Even though Depth_First_Graph_Search takes less time and memory but doesn't provide optimal solution.
- Even though Greedy_Best_First_Graph_Search H_1 takes less time and memory compared to Breadth_First_Search and Uniform_Cost_Search but it doesn't provide optimal solution.

Recommendation:

Based on above results,if memory and time is the criteria,use Greedy_Best_First_Graph_Search H_1 as against Depth_First_Graph_Search as it performed well in solving Problems 1 and 2 and Depth_First_Graph_Search performed well in solving only Problem3.
If optimal solution is the criteria,use Breadth_First_Search as against Uniform_Cost_Search because Breadth_First_Search takes less memory and time compared to Uniform_Cost_Search.

Below is from section 3.47 of Artificial Intelligence: A Modern Approach(3rd edition)

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes^a	Yes^{a,b}	No	No	Yes^a	Yes^{a,d}
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes^c	Yes	No	No	Yes^c	Yes^{c,d}

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: ^{a} complete if b is finite; ^{b} complete if step costs $\geq \epsilon$ for positive ϵ; ^{c} optimal if step costs are all identical; ^{d} if both directions use breadth-first search.

Analysis of Heuristic and Non-Heuristic Planning Search Agents

Performance of Informed Heuristic Search Agents:

	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Problem1						
H_1	55	57	224	0.04	6	Yes
H_Ignore_Preconditions	41	43	170	0.04	6	Yes
H_Pg_Levelsum	11	13	50	1.09	6	Yes
Problem2						
H_1	4853	4855	44041	13.15	9	Yes
H_Ignore_Preconditions	1450	1452	13303	4.84	9	Yes
H_Pg_Levelsum	86	88	841	170.16	9	Yes
Problem3						
H_1	18223	18225	159618	58.7	12	Yes
H_Ignore_Preconditions	5040	5042	44944	18.86	12	Yes
H_Pg_Levelsum	325	327	3002	1048.05	12	Yes

H_1 :A* Search with constant 1 heuristic.
H_Ignore_Preconditions:A* Search with ignored Preconditions heuristic.
This 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.
H_Pg_Levelsum:A* Search with level sum.
This heuristic uses a planning graph representation of the problem state space to estimate the sum of all actions that must be carried out from the current state in order to satisfy each individual goal condition.

From the above results,it is clear that all informed heuristic search agents provide optimal solutions.
H_Ignore_Preconditions performed well compared to H_Pg_Levelsum.
H_1 and H_Ignore_Preconditions almost take same time in solving Problem1.
H_Pg_Levelsum takes less memory but more time compared to H_Ignore_Preconditions.
H_Ignore_Preconditions takes more memory but less time compared to H_Pg_Levelsum.

Analysis of Heuristic and Non-Heuristic Planning Search Agents

Uninformed Vs Informed Heuristic Search Agents:

Problem1	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Breadth_First_Search	43	56	180	0.06	6	Yes
Uniform_Cost_Search	55	57	224	0.04	6	Yes
Greedy_Best_First_Graph_Search H_1	7	9	28	0.005	6	Yes
H_Ignore_Preconditions	41	43	170	0.04	6	Yes
H_Pg_Levelsum	11	13	50	1.09	6	Yes

Problem2	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Breadth_First_Search	3343	4609	30509	15.26	9	Yes
Uniform_Cost_Search	4853	4855	44041	13.61	9	Yes
H_Ignore_Preconditions	1450	1452	13303	4.84	9	Yes
H_Pg_Levelsum	86	88	841	170.16	9	Yes

Problem3	Expansions	Goal Tests	New Nodes	Time(seconds)	Plan Length	Optimal
Breadth_First_Search	14663	18098	129631	118.77	12	Yes
Uniform_Cost_Search	18223	18225	159618	60.66	12	Yes
H_Ignore_Preconditions	5040	5042	44944	18.86	12	Yes
H_Pg_Levelsum	325	327	3002	1048.05	12	Yes

From the above results,it is clear that A* Search with ignored Preconditions heuristic outperforms the uninformed Breadth First Search both in terms of memory and time.

Therefore, A* Search with ignored Preconditions heuristic is the best overall solution for the Air Cargo problem.

Conclusion:
Above results clearly mentions that the informed search agents with well designed custom heuristics out performs the uninformed search agents both in terms of memory and time.