Analysis of Heuristics for Planning Problems

Summary

This report analyzes three planning problems in the air cargo transportation domain along with the analysis of the time and space efficiency of the search algorithms that use both informed and non-informed techniques to determine the optimal solution.

In both problems 2 and 3, we see that A* Search (with ignore preconditions) perform better out of both strategies in revealing the optimal solution that is both time and space efficient compared to the rest. In problem 1, we Greedy Best First Graph Search as the overall best strategy while A* Search (with ignore preconditions) offers a similarly optimal solution that is not far off in terms of time and space efficiency. Given these observations, it would be possible to say that A* Search (with ignore preconditions) appears to be the overall best strategy for these problems. In addition, informed search strategies offer better results than uninformed search strategies for these class of problems.

Planning Problems

The three problems provided are modeled around optimally transporting air cargo between a set of airports over a provided fleet of planes.

Action Schema

The problem definition includes three actions:

1. Load Cargo into a plane at an airport
2. Unload Cargo from a plane at an airport
3. Fly a plane from one airport to another

Below is the action schema for the three actions:

```
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

Problem Schema

The initial state and the desired end goal is given in the following schema:

```
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))
```

Optimal Solution

This problem has an optimal plan length of 6 steps:

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

Metrics & Analysis

The table below captures the optimality, time and space characteristics of the search strategies used to derive the optimal solution. Both uninformed and informed search techniques are used to search through the state space to discover the right solution. Uninformed search techniques are highlighted in PURPLE and informed/heuristic/planning graph driven searches are highlighted in ORANGE. The key attributes being compared are:

- the time characteristics measured in seconds of how long the search took to discover a plan that meets the desired goal state
- space characteristics measured by the number of expansions of the states, goal tests, and new nodes
- optimality of solution as indicated by the plan length. If the plan length equals the optimal length for the problem, then it's marked as optimal

Strategy	**Optimal**	**Plan length**	**Expansions**	**Goal tests**	**New nodes**	**Time (s)**
Breadth first search	**Yes**	**6**	**43**	**56**	**180**	**0.033**
Breadth first tree search	Yes	6	1458	1459	5960	0.862
Depth first graph search	No	12	12	13	48	0.007
Depth limited search	No	50	101	271	414	0.080
Uniform cost search	Yes	6	55	57	224	0.037

Recursive best first search (h_1)	Yes	6	4229	4230	17029	2.843
Greedy best first graph search (h_1)	**Yes**	**6**	**7**	**9**	**28**	**0.006**
A* search (h_1)	Yes	6	55	57	224	0.048
A* search (ignore preconditions)	Yes	6	41	43	170	0.036
A* search (h_pg_levelsum)	Yes	6	55	57	224	0.995

From the uninformed search strategies, the most optimal strategy is Breadth First Search for this problem. Depth First Search is time and space efficient, even though it yields a plan that is less optimal than BFS by 6 steps. Breadth first tree search is the most expansive and least efficient from a time and space standpoint, although it yields an optimal solution.

From the informed search strategies, the Greedy Best First Graph Search with the h_1 heuristic offers the most time and space efficient solution in additional to being optimal, while A* Search (with ignore preconditions heuristic) comes in a close second. The Recursive Best First Search is the least time and space efficient. All informed strategies revealed the most optimal solution.

Overall, Greedy Best First Graph Search is the most time and space efficient from the two strategies. A* Search (with ignore preconditions) comes in second in space efficiency, although losing marginally to BFS in time efficiency.

Problem 2

Problem Schema

The initial state and the desired end goal is given in the following schema:

```
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))
```

Optimal Solution

This problem has an optimal plan length of 9 steps:

```
Load(C2, P2, JFK)
Load(C1, P1, SFO)
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)
```

Metrics & Analysis

Strategy	Optimal	Plan length	Expansions	Goal tests	New nodes	Time (s)
Breadth first search	**Yes**	**9**	**3343**	**4609**	**30509**	**14.507**
Breadth first tree search	No	-	-	-	-	> 10m
Depth first graph search	No	575	582	583	5211	3.054
Depth limited search	No	-				> 10m
Uniform cost search	**Yes**	**9**	**4853**	**4855**	**44041**	**12.141**
Recursive best first search (h_1)	No	-	-	-	-	> 10m
Greedy best first graph search (h_1)	No	17	998	1000	8982	2.473
A* search (h_1)	Yes	9	4853	4855	44041	12.371
A* search (ignore preconditions)	**Yes**	**9**	**1450**	**1452**	**13303**	**4.508**
A* search (h_pg_levelsum)	Yes	9	4853	4855	44041	545.817

From the uninformed search strategies, Breadth First Search is the most time space efficient, while Uniform Cost Search is the most time efficient. Both Breadth First Tree Search and Depth Limited Search took more than 10 minutes to search and so have been ruled out due to excessive execution time. Depth First Graph Search, while being space and time efficient compared the rest of the uninformed search strategies, is very sub optimal and therefore is ruled out.

From the informed search strategies, A* Search (with ignore preconditions heuristic) is the most efficient optimal solution. While Greedy Best First Graph Search yields a more space and time efficient solution, it's less optimal than A* Search (with ignore preconditions).

Overall, A* Search (with ignore preconditions heuristic) is the better approach from both strategies that yields the most time and space efficient optimal solution to this problem.

Problem 3

Problem Schema

The initial state and the desired end goal is given in the following schema:

```
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))
```

Optimal Solution

This problem has an optimal plan length of 12 steps:

```
Load(C2, P2, JFK)
Load(C1, P1, SFO)
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)
```

Metrics & Analysis

Strategy	Optimal	Plan length	Expansions	Goal tests	New nodes	Time (s)
Breadth first search	**Yes**	**12**	**14663**	**18098**	**129631**	**105.789**
Breadth first tree search	No	-	-	-	-	>10m
Depth first graph search	No	596	627	628	5176	3.341
Depth limited search	No	-	-	-	-	>10m
Uniform cost search	**Yes**	**12**	**18223**	**18225**	**159618**	**54.260**

Recursive best first search (h_1)	No	-	-	-	-	>10m
Greedy best first graph search (h_1)	No	18	5578	5580	49150	17.340
A* search (h_1)	Yes	12	18223	18225	159618	56.260
A* search (ignore preconditions)	**Yes**	**12**	**5040**	**5042**	**44944**	**17.674**
A* search (h_pg_levelsum)	No	-	-	-	-	>10m

From the uninformed strategies, both Breadth First Search and Uniform Cost Search provide optimal solutions, although the former is more space efficient than the latter and the latter is more time efficient than the former. Both of them are very space and time inefficient compared to Depth First Graph Search, which doesn't provide an optimal solution. Breadth First Tree Search and Depth Limited Search took longer than 10 minutes and so have been ruled out due to excessive execution time.

From the informed strategies, both A* Search (with h1 and ignore preconditions heuristics) generate optimal solutions, while the latter is more space and time efficient than the former. Recursive Best First Search and A* Search (with h_pg_levelsum heuristic) didn't complete in the deadline of 10 minutes and have been ruled out.

Overall, A* Search (with ignore preconditions heuristic) performed better amongst both strategies in giving an optimal solution with the best time and space efficiency characteristics.

Conclusion

In both problems 2 and 3, we see that A* Search (with ignore preconditions) perform better out of both strategies in revealing the optimal solution that is both time and space efficient compared to the rest. In problem 1, we see Greedy Best First Graph Search as the overall best strategy while A* Search (with ignore preconditions) offering a similarly optimal solution that is not far off in terms of time and space efficiency. Given these observations, it would be possible to say that A* Search (with ignore preconditions) appears to be the overall best strategy for these problems. In addition, informed search strategies offer better results than uninformed search strategies for these class of problems.

References

Russell, Stuart J. et al. "Artificial Intelligence A Modern Approach."