Implementing a project on Planning Search

-Sujit Khanna

The aim of this project is to solve a set of planning problems for an Air Cargo Transport System and compare and contrast heuristic and non heuristic search results based on optimality, time elapsed, number of nodes of expansion and such. Non-Heuristic Search Algorithms include breadth first search, breadth first tree search, depth first graph search, depth limited search, uniform cost search, recursive best first search , greedy best first graph search and heuristic search algorithms included variant of A* search discussed in more detail later in the paper.

The three planning problems in Air Cargo Domain are represented below:

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))
```

Comparisons between varied heuristic and non-heuristic strategies are represented in tables below:

Planning Problem 1						
Search Algorithm	**Plan length**	**Expansion**	**New nodes**	**Time(sec)**	**Optimality**	**Type**
1. breadth_first_search	6	43	180	0.036	Yes	Non-Heuristic
2. breadth_first_tree_search	6	1458	5960	0.902	Yes	Non-Heuristic
3. depth_first_graph_search	12	12	48	0.010	No	Non-Heuristic
4. depth_limited_search	50	101	414	0.093	No	Non-Heuristic
5. uniform_cost_search	6	55	224	0.046	Yes	Non-Heuristic
6. recursive_best_first_search h_1	6	4429	17029	2.600	Yes	Non-Heuristic
7. greedy_best_first_graph_search h_1	6	7	28	0.006	Yes	Non-Heuristic
8. astar_search h_1	6	55	224	0.042	Yes	Heuristic
9. astar_search h_ignore_preconditions	6	41	170	0.045	Yes	Heuristic
10. astar_search h_pg_levelsum	6	11	50	1.511	Yes	Heuristic

Optimal Plan for problem 1:

greedy_best_first_graph_search with h_1

Plan length: 6 Time elapsed in seconds: 0.006889881987067249
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Planning Problem 2						
Search Algorithm	**Plan length**	**Expansion**	**New nodes**	**Time(sec)**	**Optimality**	**Type**
1. breadth_first_search	9	3343	30509	17.110	Yes	Non-Heuristic
2. breadth_first_tree_search	-	-	-	-	No	Non-Heuristic
3. depth_first_graph_search	575	582	5211	3.850	No	Non-Heuristic
4. depth_limited_search	-	-	-	-	No	Non-Heuristic
5. uniform_cost_search	9	4853	44041	12.836	Yes	Non-Heuristic
6. recursive_best_first_search h_1	-	-	-	-	No	Non-Heuristic
7. greedy_best_first_graph_search h_1	15	998	8982	3.010	Yes	Non-Heuristic
8. astar_search h_1	9	4853	44041	18.440	Yes	Heuristic
9. astar_search h_ignore_preconditions	9	1450	13303	4.590	Yes	Heuristic
10. astar_search h_pg_levelsum	9	86	841	165.930	Yes	Heuristic

Optimal Plan for problem 2:

breadth_first_search.

Plan length: 9 Time elapsed in seconds: 13.569492993179649
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)

Planning Problem 3						
Search Algorithm	**Plan length**	**Expansion**	**New nodes**	**Time(sec)**	**Optimality**	**Type**
1. breadth_first_search	12	14663	129631	118.810	yes	Non-Heuristic
2. breadth_first_tree_search	-	-	-	-	No	Non-Heuristic
3. depth_first_graph_search	596	627	5176	3.484	No	Non-Heuristic
4. depth_limited_search	-	-	-	-	No	Non-Heuristic
5. uniform_cost_search	12	18223	159618	51.630	yes	Non-Heuristic
6. recursive_best_first_search h_1	-	-	-	-	No	Non-Heuristic
7. greedy_best_first_graph_search h_1	22	5578	49150	16.030	No	Non-Heuristic
8. astar_search h_1	12	18223	159618	56.090	Yes	Heuristic
9. astar_search h_ignore_preconditions	12	5040	44944	17.640	Yes	Heuristic
10. astar_search h_pg_levelsum	-	-	-	-	No	Heuristic

Optimal Plan for problem 3:

astar_search with h_ignore_preconditions

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)

Performance Analysis of Planning Problem 1:

The first table shows the performance metrics of 10 search strategies (7 non heuristic and 3 heuristic).

Compare and Contrast Non Heuristic Strategies: In case of non heuristic strategies all algorithms apart from depth_first_graph_search and depth_limited_search were optimal. For all the methods the number of nodes expansion varied a lot. Greedy Best First Graph Search performed the best as it had the lowest number of nodes expansion which led to lowest execution time.

Compare and Contrast Heuristic Strategies: For heuristic strategies all algorithms provided optimal performance levelsum strategy had the lowest number of nodes expansion but was the slowest of all the heuristic strategies

Performance Analysis of Planning Problem 2:

Compare and Contrast Non Heuristic Strategies: For non heuristic strategies breadth first search, depth limited search and recursive best first search exceeded execution time of 600 seconds and of the remaining strategies only breadth first search, uniform cost search, and greedy best first graph search were optimal. The best performing optimal strategy was greedy best first graph search with lowest execution time and fewest number of nodes expansion amongst optimal strategies.

Compare and Contrast Heuristic Strategies: All heuristic strategies yielded optimal performance. Ignore_precondition strategy had the best execution time where as levelsum strategy had fewest number of nodes expansion with slowest execution of all heuristic strategies.

Performance Analysis of Planning Problem 3:

Compare and Contrast Non Heuristic Strategies: For non heuristic strategies breadth first search and uniform cost search were optimal with strategies breadth first search, depth limited search and recursive best first search timing out (>600 sec) and depth first graph search being suboptimal (large plan length). The fastest strategy was uniform cost search and breadth first search had fewest node expansions of all optimal strategies.

Compare and Contrast Heuristic Strategies: A* search and ignore preconditions achieved optimal plan length with levelsum timing out (>600 sec). Of the two optimal strategies ignore preconditions was better as it had lower number of nodes expansion and faster execution time.

Analysis of Search Strategies:

When looking at non-heuristic strategies breadth first strategy and uniform cost strategy were the only ones that could find optimal path for all planning problems within 10 minutes. In terms of execution speed depth first graph search is the fastest and uses least amount of memory (fewest nodes expansion) however it is unable to find the optimal action plan. Non-heuristic strategies performed better in planning problems 1 and 2 as they were simpler than problem 3, this indicates that for simpler problems it is optimal to use non-heuristic strategies, however as the complexity increases as seen in planning problem 3 heurist strategies performed better as evident from table 3 where ignore precondition strategy had the fastest execution time amongst optimal strategies. This method provides a very quick estimate of how close any given state is to the goal state according to **Section 10.2.3: AIMA 3rd Edition.**