Implement a Planning Search

Heuristic Analysis

By Mahad Amer Akhter

Overview

The objective of this project is compare different non-heuristic and heuristic search in order to find the optimal plan for an airport cargo system. We are given three problems with different initial states and goals.

The air cargo action schema is described as follows:

```
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))
```

Results

The Following section contains the results of each of the problems

Problem 1:

Non Heuristic Search						
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (s)**	**Optimal**
Breadth First Search	43	56	180	6	0.058	Yes
Breadth First Tree Search	1458	1459	5960	6	1.897	Yes
Depth First Graph Search	21	22	84	20	0.026	No
Depth Limited Search	101	271	414	50	0.169	No
Uniform Cost Search	55	57	224	6	0.07	Yes
Recursive Best First Search	4229	4230	17023	6	5.338	Yes
Greedy Best First Search	7	9	28	6	0.0107	Yes

Heuristic Search						
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (s)**	**Optimal**
A* search with h1	55	57	224	6	0.07	Yes
A* search with h ignore preconditions	41	43	170	6	0.069	Yes
A* search with h pg levelsum	11	13	50	6	1.55	Yes

Five of the non-heuristic searches found the optimal plan, however, Breadth First Search was able to find it in the least amount of time. The optimal suggested by BFS is the following:

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

Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

All the heuristic searches arrived at the optimal plan, however A* search with h1 and A* search with h_ignore_preconditions arrived fastest at the solution

Problem 2:

Non Heuristic Search						
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (s)**	**Optimal**
Breadth First Search	3343	4609	30509	9	10.802	Yes
Breadth First Tree Search	Time Out					
Depth First Graph Search	624	625	5602	619	4.844	No
Depth Limited Search	Time Out					
Uniform Cost Search	4852	4854	44030	9	15.234	Yes
Recursive Best First Search	Time Out					
Greedy Best First Search	990	992	8910	21	3.15	No

Heuristic Search						
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (s)**	**Optimal**
A* search with h1	4852	4854	44030	9	15.487	Yes
A* search with h ignore preconditions	1450	1452	13303	9	5.48	Yes
A* search with h pg levelsum	86	88	841	9	218.779	Yes

Two of the non-heuristic searches arrived at the optimal plan with Breadth First search being the fastest. The optimal path is as follows:

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

All the heuristic searches found the optimal path, however, A* search with h ignore preconditions was the fastest

Problem 3:

Non Heuristic Search						
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (s)**	**Optimal**
Breadth First Search	14663	18098	129631	12	94.836	yes
Breadth First Tree Search	Time Out					
Depth First Graph Search	408	409	3364	392	2.257	No
Depth Limited Search	Time Out					
Uniform Cost Search	18234	18236	159707	12	83.396	yes
Recursive Best First Search	Time Out					
Greedy Best First Search	5605	5607	49360	22	24.303	No

Heuristic Search						
Search Method	**Expansions**	**Goal Tests**	**New Nodes**	**Plan Length**	**Time (s)**	**Optimal**
A* search with h1	18234	18236	159707	12	82.299	yes
A* search with h ignore preconditions	5040	5042	44944	12	29.052	yes
A* search with h pg levelsum	325	327	3002	12	1384.28	yes

Two of the non-heuristic searches found the optimal plan but uniform cost search was faster in this case. The optimal plan for this problem is as follows:

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

Among the heuristic searches, all of them found the optimal plan and similar to the previous problems, A* search with h ignore preconditions

Analysis

Non-Heuristic Searches

Among the non-heuristic searches, Breadth first Search, Depth First Graph Search and Uniform Cost Search were all successful in finding a plan. However, only Breadth First Search and Uniform Cost Search were able to find an optimal solution.

BFS was able to find a solution in the least amount of time in the case of Problem 1 and Problem2. In Problem 3, however, Uniform Cost Search was slightly faster

Heuristic Searches

In all the three problems, all the heuristic searches were able to find the optimal plan. A* search with h ignore preconditions, however did it in the least amount of time. Hence we can conclude that A* search with h ignore preconditions was the best heuristic used