Planning Heuristic Analysis

By Alessandro Bersia

The three different planning problems were given used the same action 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))
```

The following are the initial states and goals of the three problems:

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))
```

Each problem has been analyzed with several algorithms in order to find the lowest path from the start to the goal; sample plans with optimal length are shown below:

PROBLEM 1 (optimal plan length):

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

PROBLEM 2 (optimal plan length):

```
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)
```

PROBLEM 3 (optimal plan length):

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

NON-HEURISTIC SEARCHES

(preferred solution highlighted in green, over 10 minutes of time elapsed in red)

Air Cargo Problem 1

	Number of Node Expansion Required	Number of Goals Tested	Time Elapsed	Plan Length	Optimal
Breadth First Search	43	56	0.03101	6	Yes
Breadth First Tree Search	1459	1459	0.8642	6	Yes
Depth First Graph Search	21	22	0.01285	20	No
Depth Limited Search	101	271	0.0802	50	No
Uniform Cost Search	55	57	0.03380	6	Yes

Air Cargo Problem 2

	Number of Node Expansion Required	Number of Goals Tested	Time Elapsed	Plan Length	Optimal
Breadth First Search	3343	4609	12.5785	9	Yes
Breadth First Tree Search			>10 min		
Depth First Graph Search	386	387	1.5831	380	No
Depth Limited Search			>10 min		
Uniform Cost Search	4853	4855	11.2749	9	Yes

Air Cargo Problem 3

	Number of Node Expansion Required	Number of Goals Tested	Time Elapsed	Plan Length	Optimal
Breadth First Search	**14663**	**18098**	**92.5705**	**12**	**Yes**
Breadth First Tree Search			**>10 min**		
Depth First Graph Search	**408**	**409**	**1.9882**	**392**	**No**
Depth Limited Search			**>10 min**		
Uniform Cost Search	**18223**	**18225**	**50.2203**	**12**	**Yes**

COMPARE NON-HEURISTIC SEARCH

Non-heuristic searches that run for more than 10 minutes will not be chosen, therefore Breadth First Tree Search and Depth Limited Search will not be used, since they required more than 10 minutes to be processed for Problem 2 and Problem 3.

Breadth first search always considers the shortest path first [1][1] and we can observe that the optimal solution is found in a reasonable amount of time

Depth First Graph Search is not optimal because it explores the node as deep as possible in the graph, even if the goal is yet to its right [1].

In the problem 1 the optimal plan length is 6, and considering the computing time and the explored note, BFS is preferred (see tables above for the details).

For problem 2 and 3 the situation is similar, BFS achieved the optimal plan length even though Depth First Graph Search gets the lowest number of node expanded and goal tested. Nevertheless, DFS is not optimal because the achieved path lengths are dramatically higher.

DFS can be used when a fast execution is needed, due to its lower time elapsed (shown in the tables above), but it will not return the shortest route.

[1] Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition)

HEURISTIC SEARCHES

(preferred solution highlighted in green, over 10 minutes of time elapsed in red)

Air Cargo Problem 1

	Number of Node Expansion Required	Number of Goals Tested	Time Elapsed	Plan Length	Optimal
Recursive Best First Search h_1	4229	4230	2.5217	6	Yes
Greedy Best First Search h_1	7	9	0.0046	6	Yes
A* Search h_1	55	57	0.0345	6	Yes
A* Search h_ignore_precondition	55	57	0.0391	6	Yes
A* Search h_pg_levelsum	39	41	0.8427	6	Yes

Air Cargo Problem 2

	Number of Node Expansion Required	Number of Goals Tested	Time Elapsed	Plan Length	Optimal
Recursive Best First Search h_1			>10 min		
Greedy Best First Search h_1	998	1000	2.2613	21	No
A* Search h_1	4853	4855	11.1344	9	Yes
A* Search h_ignore_precondition	4853	4855	12.3814	9	Yes
A* Search h_pg_levelsum			>10 min		

Air Cargo Problem 3

	Number of Node Expansion Required	Number of Goals Tested	Time Elapsed	Plan Length	Optimal
Recursive Best First Search h_1			**>10 min**		
Greedy Best First Search h_1	**6943**	**6945**	**18.4924**	**22**	**No**
A* Search h_1	**18223**	**18225**	**48.9885**	**12**	**Yes**
A* Search h_ignore_precondition	**18223**	**18225**	**54.7164**	**12**	**Yes**
A* Search h_pg_levelsum			**>10 min**		

COMPARE HEURISTIC SEARCH

A* with constant heuristic and A* with ignore precondition achieve the same number of node expanded and the same number of goal tested for all the 3 problems analyzed. Due to its slightly lower elapsed time, A* with constant heuristic is the preferred choice for all the 3 problems.

A* with Levelsum Heuristic achieved lower number of expanded node and goal tested and the optimal plan length for problem 1, but scored higher time elapsed. Especially in problem 2 and 3, the computation time of A* with Levelsum Heuristic is over 10 minutes. Therefore, it is not efficient in terms of computational time, because it needs to generate the graphplan's graph [1]. Hence, the advised algorithm for the heuristic search is A* with constant heuristic (see tables above for more details)

CONCLUSION

According to the result achieved, the Bread First Search algorithm can solve planning problems quickly and efficiently, therefore it is a suitable candidate for solving this kind of problems. Nevertheless, as the complexity of the problems increases, a heuristic algorithm (e.g. A* Search with constant heuristic) can outperform the Bread First Search algorithm and should be preferred.

REFERENCES

[1] Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition)