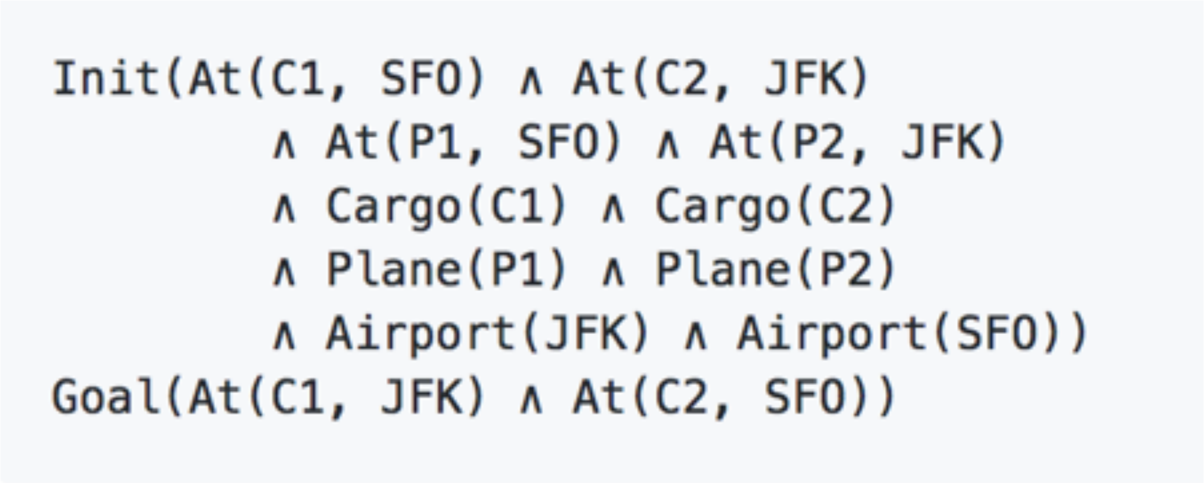
VERDU SANJAY
08/02/2018

Air Cargo Problem : Heuristic Analysis

Optimal Solutions for Problems 1,2,3:

Problem 1 :

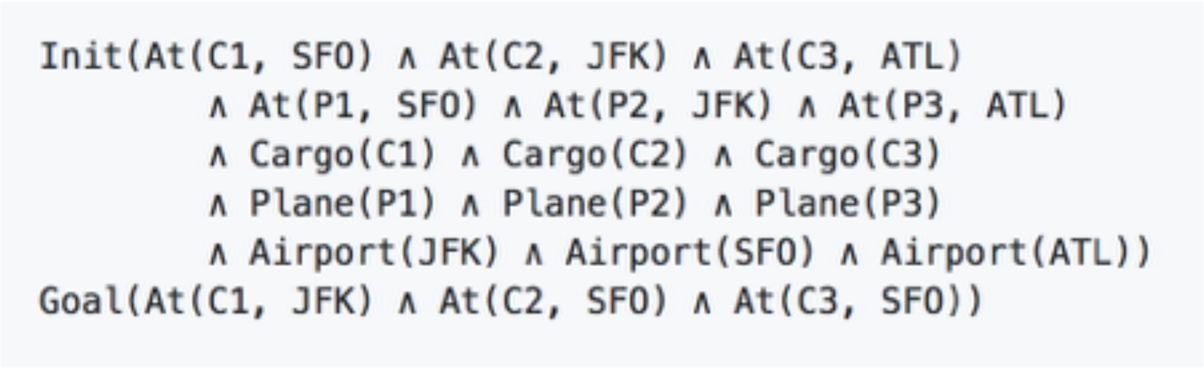


```
Init(At(C1, SFO) ∧ At(C2, JFK)
        ∧ At(P1, SFO) ∧ At(P2, JFK)
        ∧ Cargo(C1) ∧ Cargo(C2)
        ∧ Plane(P1) ∧ Plane(P2)
        ∧ Airport(JFK) ∧ Airport(SFO))
Goal(At(C1, JFK) ∧ At(C2, SFO))
```

Problem 1 Optimal Plan (BFS):

```
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 :

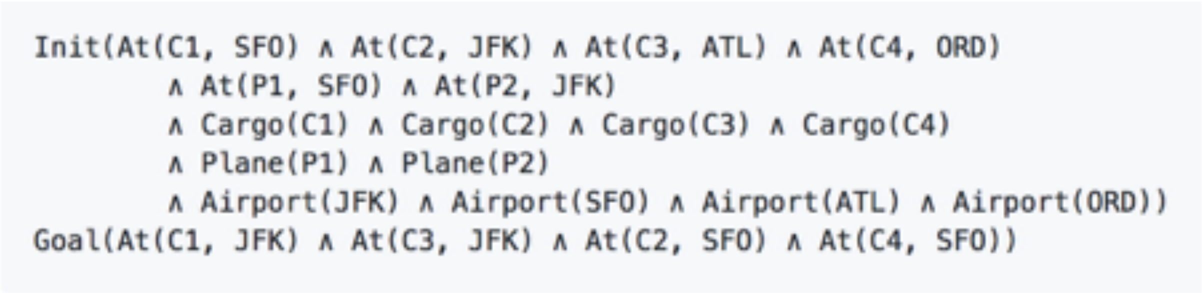


```
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL)
        ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL)
        ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3)
        ∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3)
        ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL))
Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))
```

Problem 2 Optimal Plan (A* search ignoring preconditions) :

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

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))
```

Problem 3 Optimal Plan (A* search ignoring 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)
```

Comparing uniformed non heuristic searches:

Problem 1 :

ALGORITHM	OPTIMAL	PLAN LENGTH	RUN TIME	EXPANSIONS
BFS	YES	6	0.039	43
DFS	NO	20	0.016	21
UNIFORM COST SEARCH	YES	6	0.053	55

Algorithm highlighted in yellow colour is optimal.

For this problem it is evident that optimal plan length is 6 which we got in the case of **BFS** as well as **UNIFORMED COST SEARCH.**But if you look at the RUN times as well as memory the algorithms are using, it pretty clear **BFS** is doing better job here.Its RUN time is only 0.039

DFS is not good here because it has a path length of 20 which is way large than that of **BFS.**This is because of the reason that DFS goes deep through first branch first and then second etc which is of course not optimal.

Problem 2 :

ALGORITHM	OPTIMAL	PLAN LENGTH	RUN TIME	EXPANSIONS
BFS	YES	9	19.81	3343
DFS	NO	619	6.97	624
UNIFORM COST SEARCH	YES	9	22.85	4853

Algorithm highlighted in yellow colour is optimal.

For Problem 2,from the above table it is evident that the optimal length is 9.This is quite a complex problem compared to the first one.Because the RUN times, expansions etc are greater than that of problem 1 for every algorithm we used.

BFS and **UNIFORM COST SEARCH** algorithms are using a lot of space while **DFS** is using a way lesser.Its always your choice to choose if you want fastest algorithm or optimal algorithm.In general we always vote for optimal though.

I choose the same.Of the both optimal solutions above i.e DFS and UNIFORM COST SEARCH,i would choose BFS because its RUN time and memory used is less compared to UNIFORM COST SEARCH.

Problem 3 :

ALGORITHM	OPTIMAL	PLAN LENGTH	RUN TIME	EXPANSIONS
BFS	YES	12	73.49	14663
DFS	NO	392	2.41	408
UNIFORM COST SEARCH	YES	12	83.46	18223

Algorithm highlighted in yellow colour is optimal.

Again,its **BFS** for the same reasons as stated above for problem 2.

Its surprising that, for all 3 problems, **BFS** and **UNIFORM COST SEARCH** produced optimal path lengths.It was RUN time and memory that where BFS had upper hand over **UNIFORM COST SEARCH.**

BFS always finds for shortest path by just expanding one layer after other.**UNIFORM COST SEARCH** first expands path with low cost and this is probably the reason it reached goal state late and used more memory compared to BFS.

So,i would like to conclude for this problem that, among **non heuristic searches, BFS** is the best algorithm.

Comparing heuristic searches:

Problem 1 :

ALGORITHM	OPTIMAL	PLAN LENGTH	RUN TIME	EXPANSIONS
A* with ignore preconditions heuristic	YES	6	0.042	41
A* with PG level sum heuristic	YES	6	0.567	11

Algorithm highlighted in yellow colour is optimal.

From the above table, we can see that both algorithms are optimal with an optimal length of 6.There are both doing quite well with respect to time but first one is faster than the second as well as its using less memory compared to second. So, i go with the first one for this problem.

If you compare this to the optimal **BFS** solution (non heuristic search) you can observe that **BFS was as good as heuristic search.**

So,if i have to choose between non heuristic search and heuristic search, for this problem will go for **Both.**

Problem 2 :

ALGORITHM	OPTIMAL	PLAN LENGTH	RUN TIME	EXPANSIONS
A* with ignore preconditions heuristic	YES	9	5.83	1450
A* with PG level sum heuristic	YES	9	61.510	86

Algorithm highlighted in yellow colour is optimal.

From the above table, we can see that both algorithms are optimal with an optimal length of 9.But if you compare RUN time, first is doing a way better than second.If you compare EXPANSIONS , second is doing a way better than first.Generally we prefer faster solution, so i choose first one.

If you compare this to the optimal **BFS** solution (non heuristic search) you can observe that **Heuristic search** did better than **non – heuristic** w.r.t time (faster solution) even though they have same optimal plan length.

So,if i have to choose between non heuristic search and heuristic search, for this problems will go for **Heuristic search.**

Problem 3 :

ALGORITHM	OPTIMAL	PLAN LENGTH	RUN TIME	EXPANSIONS
A* with ignore preconditions heuristic	YES	12	21.53	5040
A* with PG level sum heuristic	YES	12	346.77	325

Algorithm highlighted in yellow colour is optimal.

From the above table, we can see that both algorithms are optimal with an optimal length of 12.But if you compare RUN time, first is doing a way better than second.If you compare EXPANSIONS , second is doing a way better than first.Generally we prefer faster solution, so i choose first one.

If you compare this to the optimal **BFS** solution (non heuristic search) you can observe that **Heuristic search** did better than **non – heuristic** w.r.t time (faster solution) even though they have same optimal plan length.

So,if i have to choose between non heuristic search and heuristic search, for this problems will go for **Heuristic search.**

Conclusion:

Well,after observing all the readings from tables it is clearly evident that Heuristic search is better than non heuristic search.Its the **A* search ignoring preconditions heuristic** that did the best(according to time).If you prefer memory efficiency **then A* search level sum heuristic is best.**