Heuristics Analysis for the Air Cargo Problems

Adwaith Gupta

PROBLEM DESCRIPTION

The Air Cargo problems are defined below, the action schema is common to all the three problems, but the initial and goals states vary.

Air Cargo 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))
```

Problem-1 (P1)

```
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 (P2)

```
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 (P3)

```
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 PLANS

P1 (Plan length=6)	P2 (Plan length=9)	P3 (Plan length=12)
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK)	Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P3, ATL, SFO) Unload(C3, P3, SFO)	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(P1, ATL, JFK) Unload(C1, P1, JFK) Unload(C3, P1, JFK) Fly(P2, ORD, SFO) Unload(C2, P2, SFO) Unload(C4, P2, SFO)

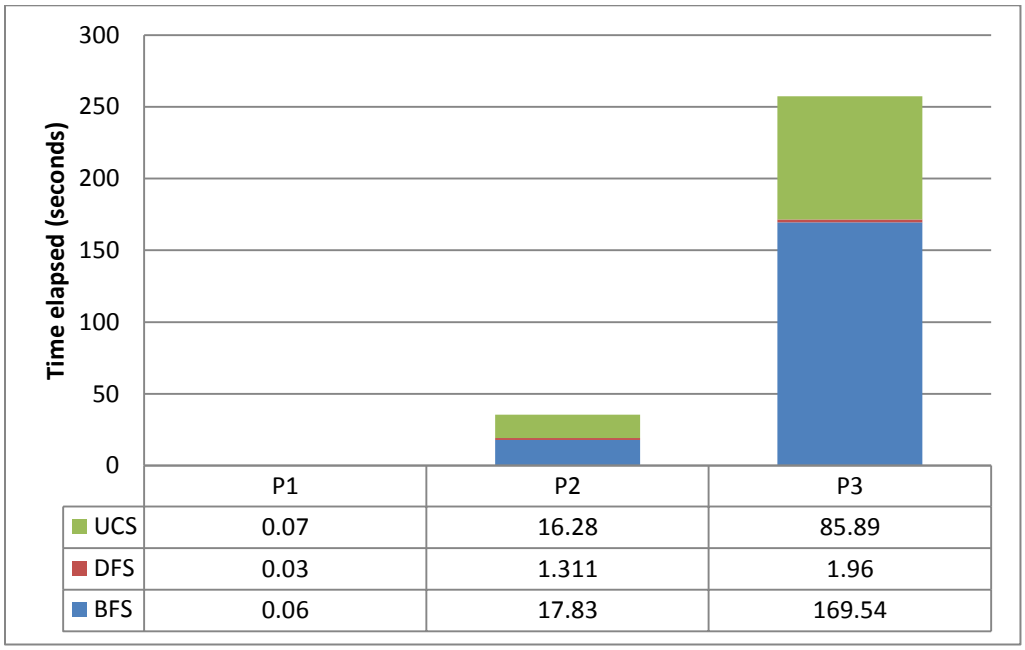
NON-HEURISTIC SEARCH RESULTS

Optimality

The depth-first graph search (DFS) as expected does not give an optimal solution. Breadth-first search (BFS) does give an optimal solution because each step in plan has equal cost, for example, the step cost of flying from SFO to ATL is same as flying from JFK to ORD. Uniform-cost search (UCS) gives the optimal solution. This optimality behavior was observed for all the three problems.

Time elapsed

DFS takes the least amount of time; however it is useless because it provides very long non-optimal solutions. P1 results are hardly visible because the times scale is much smaller than P2 and P3. Deciding time complexity from P1 may not be good idea because the coding overheads might have caused those differences.



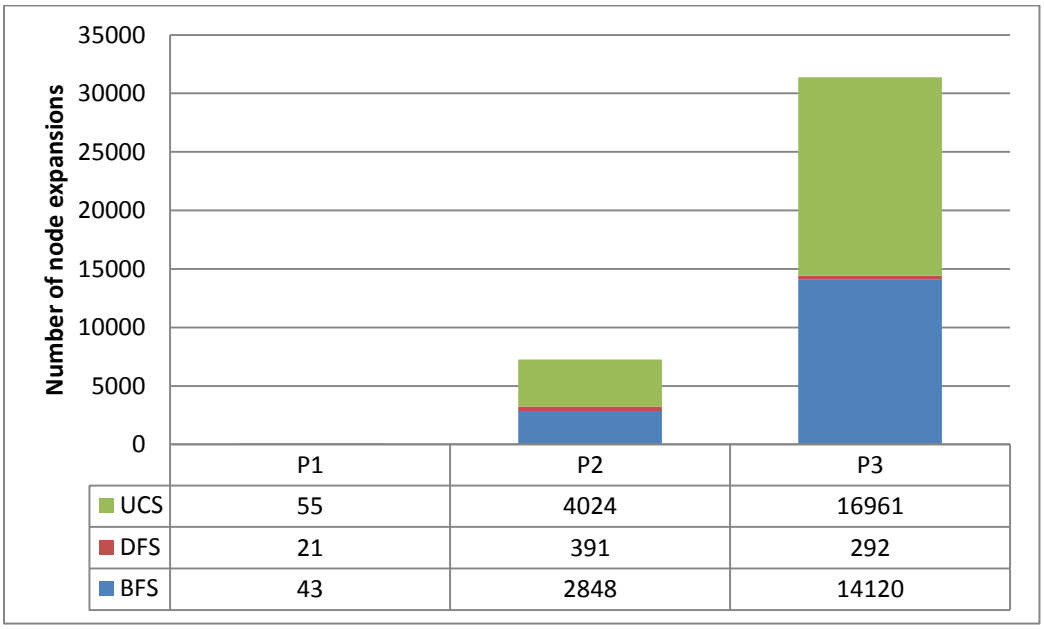
	P1	P2	P3
UCS	0.07	16.28	85.89
DFS	0.03	1.311	1.96
BFS	0.06	17.83	169.54

Time elapsed comparison for the BFS, DFS and UCS on three Air Cargo problems

The difference between UCS and BFS becomes significant for P3, a relatively large problem. UCS takes almost half the time taken by BFS. Since UCS and BFS are both optimal, it is fair to say that for these problems UCS is the best solution in terms of time complexity.

Number of node expansions

Again, the number of node expansions for DFS is the lowest, but it gives non-optimal solution, rendering it useless for these cargo planning problems. Between BFS and UCS, BFS shows lower number of expansions. The space complexity for UCS is almost similar to BFS for equal step costs. Strictly, the complexity is slightly more than the BFS because UCS examines all the nodes at the goals' depth. This is consistent with the results obtained below.



	P1	P2	P3
UCS	55	4024	16961
DFS	21	391	292
BFS	43	2848	14120

Number of node expansions comparison for the BFS, DFS and UCS on three Air Cargo problems

HEURISTIC SEARCH RESULTS

Optimality

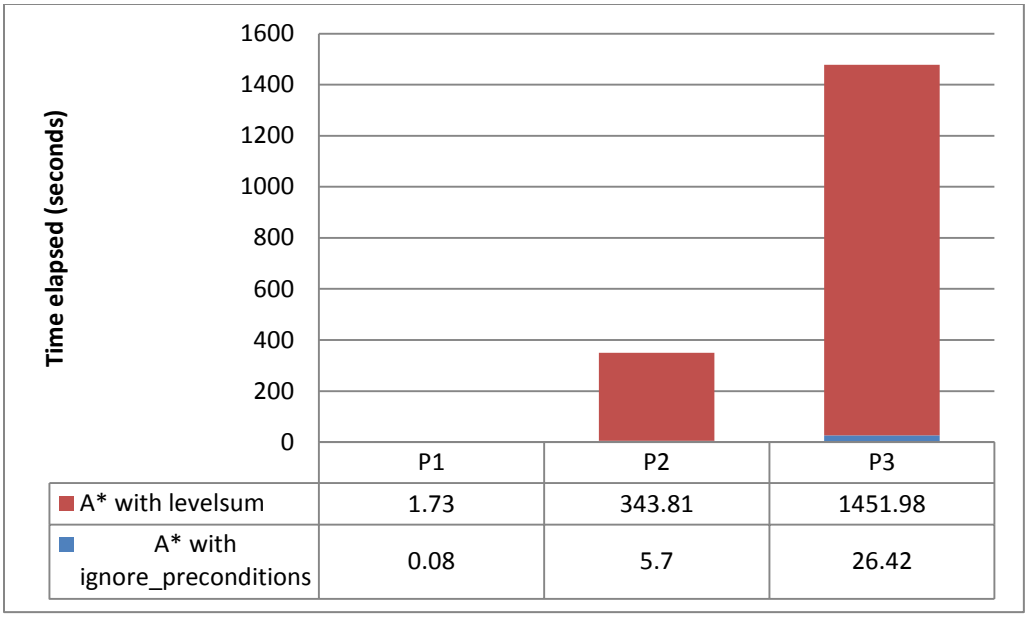
Both ignore_preconditions and levelsum heuristics give optimal solutions for P1 and P2. For P3 however, levelsum heuristic's plan length is 13 compared to the optimal plan length of 12.

Time elapsed

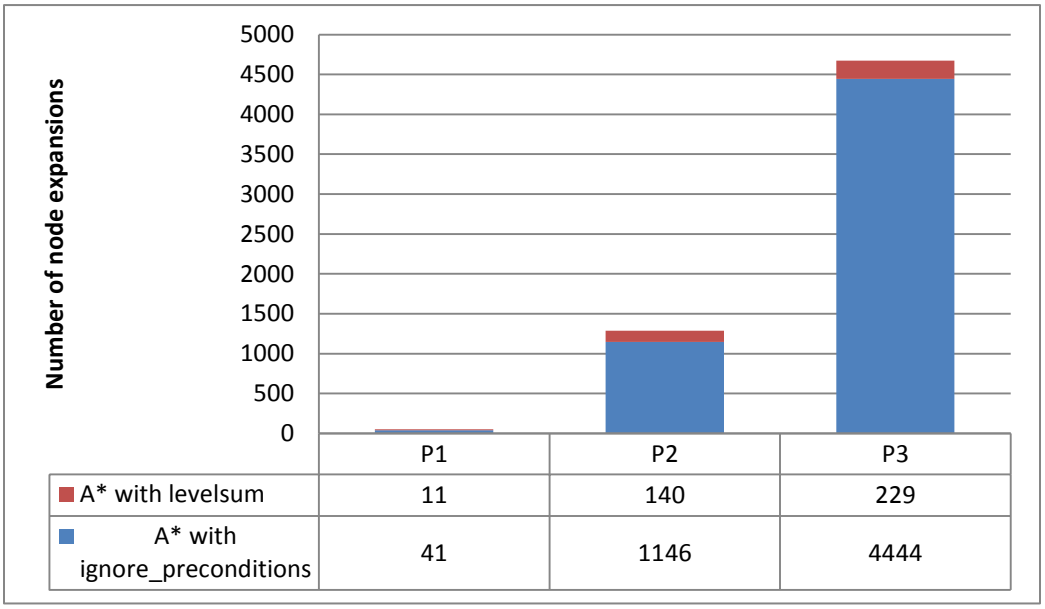
Clearly, ignore_preconditions heuristic takes significantly less time compared to the levelsum heuristic. Given that ignore_preconditions heuristic gives optimal plan for all the three problems, it appears to be the best choice for solving such problems.

Number of node expansions

Ignore_preconditions heuristic basically ignores all preoncditions from actions. Every action becomes applicable in every state, which means a lot of nodes are possible to be expanded. The theory is consistent with the results. Ignore_preconditions heuristic expands significantly large number of nodes compared to the levelsum heuristic in all the three cargo problems.



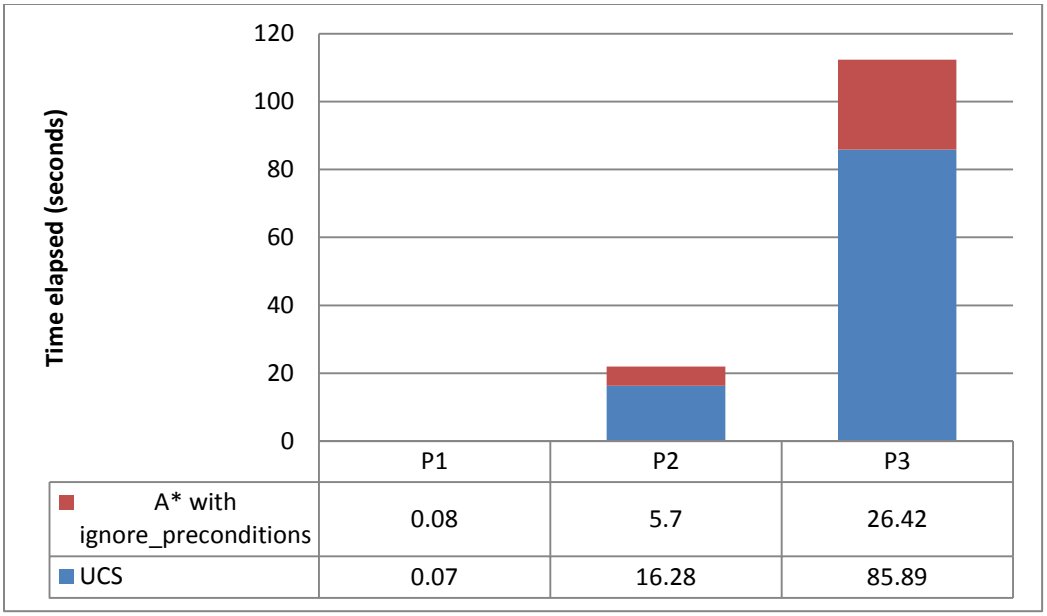
Time elapsed comparison for the A* search with ignore_preconditions heuristic and A* search with levelsum heuristic on three Air Cargo problems



Number of node expansions comparison for the A* search with ignore_preconditions heuristic and A* search with levelsum heuristic on three Air Cargo problems

CONCLUSIONS

Ignoring the space complexity (number of node expansions) makes A*with ignore_preconditions and UCS as the best planners from heuristic and non-heuristic planning schemes respectively. A comparison of time elapsed results is shown below. A* with ignore_precondtions heuristic performs better than UCS.



Time elapsed comparison for the A* search with ignore_preconditions heuristic and UCS on three Air Cargo problems

As far as the number of node expansions go, A* with ignore_preconditions starts to perform much better as the problem becomes larger (shown below). From the results, it can be said that for such air cargo type problems A* search with ignore_preconditions heuristic is the best planner.