Part 1 - Planning problems

Experiment and document metrics for non-heuristic planning solution searches

1) Results of uninformed search heuristics for air_cargo_p1, air_cargo_p2, air_cargo_p3

a) air_cargo_p1

Method	Optimality (Path length)	time elapsed (s)	node expansions
BFS	Yes (6)	0.0459	43
DFS	No (20)	0.0272	21
UCS	Yes (6)	0.064586	55

b) air_cargo_p2

Method	Optimality (Path length)	time elapsed (s)	node expansions
BFS	Yes (9)	22.44368	3343
DFS	No (619)	5.4828	624
UCS	Yes (9)	18.035	4853

c) air_cargo_p3

Method	Optimality (Path length)	time elapsed (s)	node expansions
BFS	Yes (12)	168.3366	14663
DFS	No (392)	2.7318	408
UCS	Yes (12)	79.327	18223

Part 2 - Domain-independent heuristics

Experiment and document: metrics of A* searches with these heuristics

a) air_cargo_p1

Method	**Optimality (Path length)**	**time elapsed (s)**	**node expansions**
h_ignore_preconditions	Yes (6)	0.0549	170
h_level_sum	Yes (6)	0.9971	11

b) air_cargo_p2

Method	**Optimality (Path length)**	**time elapsed (s)**	**node expansions**
h_ignore_preconditions	Yes (9)	5.017	1450
h_level_sum	Yes (9)	85.4798	86

c) air_cargo_p3

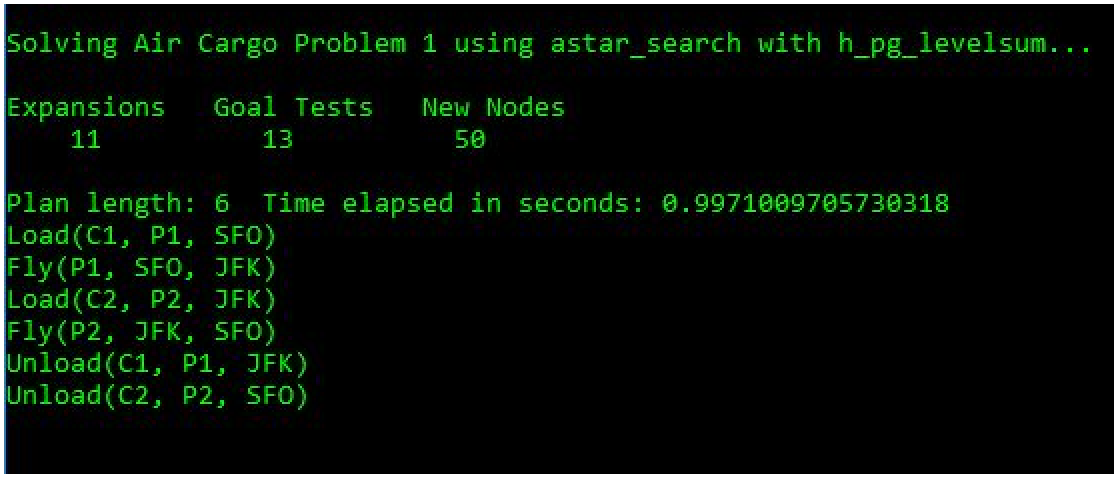
Method	**Optimality (Path length)**	**time elapsed (s)**	**node expansions**
h_ignore_preconditions	Yes (12)	25.234	5040
h_level_sum	Yes (12)	464.01	325

Part 3: Written Analysis

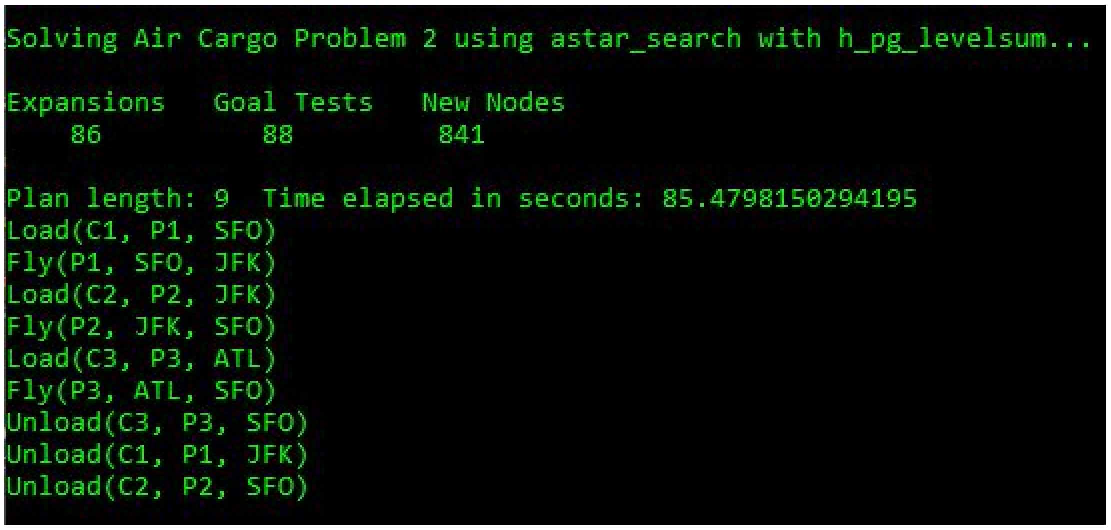
Include the following in your written analysis

1) *Optimal plan for problems 1, 2, and 3*

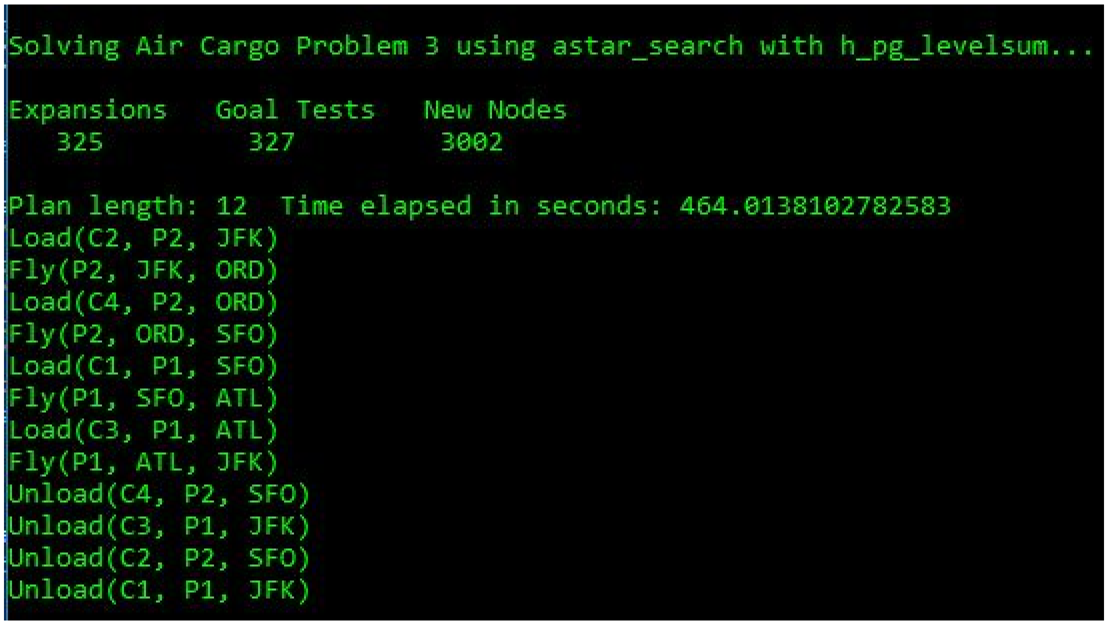
The snapshot below shows optimal plans for problem 1, 2, 3



```
Solving Air Cargo Problem 1 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
    11          13           50

Plan length: 6  Time elapsed in seconds: 0.9971009705730318
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```



```
Solving Air Cargo Problem 2 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
    86          88          841

Plan length: 9  Time elapsed in seconds: 85.4798150294195
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```



```
Solving Air Cargo Problem 3 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
   325          327         3002

Plan length: 12  Time elapsed in seconds: 464.0138102782583
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)
```

2) *Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3. Include breadth-first, depth-first, and at least one other uninformed non-heuristic search in your comparison; Your third choice of non-heuristic search may be skipped for Problem 3 if it takes longer than 10 minutes to run, but a note in this case should be included.*

It can be seen from tables in part 1 that BFS and UCS returns an optimal path whereas DFS returns a longer path which isn't optimal. The time taken and nodes expanded is lowest for DFS.

The reason DFS returns a non optimal path is because it explores the entire depth of the tree for solution whereas the optimal path may lie above the deepest node of the tree. BFS and UCS on the other hand return optimal path if the step cost is assumed to be 1.

From time perspective, DFS doesn't have to explore all the branching options thereby reducing the search time. Time taken for BFS is O(b^d) where b is the branching factor and d is the depth of tree, DFS is O(b^m) where m is the maximum depth of any node. If the solution for DFS is found early without expanding several branches, this time will typically be less than BFS.

Since the cost of each path is one, UCS expands more nodes (O(b^(d+1))) in comparison to BFS (O(b^(d))) as it tries to find the path with minimum cost.Space complexity for DFS is (O(bm)) so it expands the least number of nodes.

3) *Compare and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.*

It can be seen from part 2 that **h_ignore_preconditions** expands more nodes in comparison to **h_level_sum** while taking lesser time to find the solution. This is because it is computationally less intensive to calculate **h_ignore_preconditions** heuristics as it is a simpler heuristic than **h_level_sum** heuristics. On the other hand **h_level_sum** gives a more accurate heuristic for estimation of the goal leading to lesser nodes expansion. Both cases return optimal solution.

4) *What was the best heuristic used in these problems? Was it better than non-heuristic search planning methods for all problems? Why or why not?*

h_level_sum was the best heuristic as it lead to the least expansion of nodes while returning optimal path. It is better than non-heuristic search of node expansions is taken as criteria for determining better planning method. This is because it can guide the problem to goal state faster by estimating the distance from goal and taking that into account while searching.