# Heuristic Analysis

All problems are in the Air Cargo domain. They have the same action schema defined, but different initial states and goals. All three problems were solved using three different non-heuristic search algorithms and two executions of the A* algorithm with two heuristics: *ignore preconditions* and *planning graph level sum*.

## 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

### Definition

#### Initial state

```
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

```
Goal(At(C1, JFK) ∧ At(C2, SFO))
```

### Optimal plan

The optimal plan for the problem 1 has 6 steps which can be the following:

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

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

### Search algorithms results

| Algorithm | Expansions | Goal Tests | New Nodes | Plan length | Time elapsed (secods) |
|---|---|---|---|---|---|
| Breadth First Search | 43 | 56 | 180 | 6 | 0.046311 |
| Depth First Search | 21 | 22 | 84 | 20 | 0.022239 |
| Uniform Cost Search | 55 | 57 | 224 | 6 | 0.049400 |
| A* Search with Ignore Preconditions | 55 | 57 | 224 | 6 | 0.055052 |
| A* Search with PG Level Sum | 11 | 13 | 50 | 6 | 1.373335 |

### Analysis

From the non-heuristic search algorithms used to solve the planning problem, only Depth First Search was not able to get an optimal plan. Breadth First Search did less node expansions and goal tests resulting in a slightly smaller execution time than Uniform Cost Search. Both heuristics arrived at an optimal plan, as expected, and, although the PG Level Sum heuristic executed the least goal tests and expanded the least amount of nodes and new nodes, it had by far the longest execution time.

It is worth noting that both, the Uniform Cost Search and the Ignore preconditions heuristic, expand the same amount of nodes and have the same amount of goal tests but the Uniform Cost Search is faster because it does not have to spend extra time computing the heuristic value.

The best algorithm for this problem is Bread First Search.

## Problem 2

### Definition

#### Initial state

```
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**

```
Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))
```

# Optimal plan

The optimal plan for the problem 2 has 9 steps which can be the following:

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

## Search algorithms results

| Algorithm | Expansions | Goal Tests | New Nodes | Plan length | Time elapsed (secods) |
|---|---|---|---|---|---|
| Breadth First Search | 3343 | 4609 | 30509 | 9 | 15.329773 |
| Depth First Search | 624 | 625 | 5602 | 619 | 3.827281 |
| Uniform Cost Search | 4853 | 4855 | 44041 | 9 | 52.683208 |
| A* Search with Ignore Preconditions | 4853 | 4855 | 44041 | 9 | 53.042080 |
| A* Search with PG Level Sum | 13 | 15 | 123 | 9 | 30.826120 |

## Analysis

From the non-heuristic search algorithms used to solve the planning problem, only Depth First Search was not able to get an optimal plan. Breadth First Search did less node expansions and goal tests resulting in a smaller execution time than Uniform Cost Search. Both heuristics arrived at an optimal plan, as expected, and the PG Level Sum heuristic executed the least goal tests and expanded the least amount of nodes and new nodes, and it had a smaller execution time than the Ignore Preconditions heuristic.

Again both, the Uniform Cost Search and the Ignore preconditions heuristic, expanded the

same amount of nodes and have the same amount of goal tests but the Uniform Cost Search was faster by a small margin because it does not have to spend extra time computing the heuristic value.

The best algorithm for this problem is, again, Bread First Search.

# Problem 3

## Definition

### Initial state

```
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

```
Goal(At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))
```

# Optimal plan

The optimal plan for the problem 3 has 12 steps which can be the following:

```
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)
```

## Search algorithms results

| Algorithm | Expansions | Goal Tests | New Nodes | Plan length | Time elapsed (secods) |
|---|---|---|---|---|---|
| Breadth First Search | 14663 | 18098 | 129631 | 12 | 110.269349 |

| Algorithm | Expansions | Goal Tests | New Nodes | Plan length | Time elapsed (secods) |
|---|---|---|---|---|---|
| Depth First Search | 408 | 409 | 3364 | 392 | 1.923002 |
| Uniform Cost Search | 18223 | 18225 | 159618 | 12 | 442.549738 |
| A* Search with Ignore Preconditions | 18223 | 18225 | 159618 | 12 | 473.942290 |
| A* Search with PG Level Sum | 23 | 25 | 209 | 12 | 81.589832 |

### Analysis

From the non-heuristic search algorithms used to solve the planning problem, only Depth First Search was not able to get an optimal plan. Breadth First Search did less node expansions and goal tests resulting in a smaller execution time than Uniform Cost Search. Both heuristics arrived at an optimal plan, as expected, and the PG Level Sum heuristic executed the least goal tests and expanded the least amount of nodes and new nodes, having the smallest execution time of all the algorithms.

Again both, the Uniform Cost Search and the Ignore preconditions heuristic, expanded the same amount of nodes and have the same amount of goal tests but the Uniform Cost Search was faster by a small margin because it does not have to spend extra time computing the heuristic value.

The best algorithm for this problem is, by far, the A* Search with PG Level Sum.

## Conclusion

For small problems the Bread First Search is the best option as it always finds an optimal solution in the fastest way. For larger problems the A* Search with PG Level Sum is the best option. In all cases this algorithm does the least number of evaluations and expansions but generating the Planning Graph is an expensive process so the benefits does not show up until the graph becomes large enough.