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	0.029244
Depth First Search	21	22	84	20	0.013205
Uniform Cost Search	55	57	224	6	0.041654
A* Search with Ignore Preconditions	41	43	170	6	0.028614
A* Search with PG Level Sum	11	13	50	6	1.348029
A* Search with PG Level Sum w/o mutex	11	13	50	6	0.565533

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, it had by far the longest execution time. On the other hand, the Ignore Preconditions heuristic expanded less nodes than all the strategies that found the optimal plan allowing it to have the smallest execution time followed closely by Breadth First Search.

The best strategy for this problem is A* Search with Ignore Preconditions.

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	14.359861
Depth First Search	624	625	5602	619	3.253226
Uniform Cost Search	4853	4855	44041	9	44.414789
A* Search with Ignore Preconditions	1506	1508	13820	9	13.523730
A* Search with PG Level Sum	86	88	841	9	154.981159
A* Search with PG Level Sum w/o mutex	86	88	841	9	38.256941

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, although the PG Level Sum heuristic executed the least goal tests and expanded the least amount of nodes, it had by far the longest execution time. On the other hand, the Ignore Preconditions heuristic expanded less nodes than all the strategies that found the optimal plan allowing it to have the smallest execution time followed closely by Breadth First Search.

The best strategy for this problem is, again, A* Search with Ignore Preconditions.

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)

Algorithm	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed (secods)
Breadth First Search	14663	18098	129631	12	112.262875
Depth First Search	408	409	3364	392	2.113631
Uniform Cost Search	18223	18225	159618	12	481.331630
A* Search with Ignore Preconditions	5118	5120	45650	12	90.308010
A* Search with PG Level Sum	414	416	3818	12	1084.156915
A* Search with PG Level Sum w/o mutex	414	416	3818	12	252.465691

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, although the PG Level Sum heuristic executed the least goal tests and expanded the least amount of nodes, it had by far the longest execution time. On the other hand, the Ignore Preconditions heuristic expanded less nodes than all the strategies that found the optimal plan allowing it to have the smallest execution time followed closely by Breadth First Search.

The best algorithm for this problem is, by far, the A* Search with Ignore Preconditions.

Conclusion

For all problems the A* Search with Ignore Preconditions strategy is the best option as it always finds an optimal solution in the fastest way. It is worth noting that the current implementation of this heuristic assumes that that no action can satisfy more than one legal goal [2], then it is sufficient to calculate many of the goal states are not yet satisfied in the current state instead of calculating the whole covering set of actions.

The A* Search with PG Level Sum in all cases does the least number of evaluations and expansions but generating the Planning Graph is an expensive process so the benefits simply does not show up. It is also worth nothing that is possible to optimize the GP generation by removing the calculation of mutex [3] that are not used to calculate the PG Level Sum heuristic, this change makes the execution time of the strategy better to the execution time of the Uniform Cost Search strategy and suggests that more optimizations can be done to the graph generation code to bring it closer to the Ignore Preconditions results.

References

1. Norvig, P. and Russel, S. (2010). Artificial Intelligence: A Modern Approach. Third Edition.
2. Lapollo, C. (2017) Understanding ignore precondition heuristic. Answer on Udacity forums. Recovered from https://discussions.udacity.com/t/understanding-ignore-precondition-heuristic/225906 (https://discussions.udacity.com/t/understanding-ignore-precondition-heuristic/225906)
3. Helen-470486. (2017) Do we actually need mutex links for level_sum heuristic?. Udacity Forums. Recovered from https://discussions.udacity.com/t/do-we-actually-need-mutex-links-for-level-sum-heuristic/235870 (https://discussions.udacity.com/t/do-we-actually-need-mutex-links-for-level-sum-heuristic/235870)