Analysis of Different Search Algorithms on the Air Cargo Problem

The Problem

The Air Cargo Schema in PDDL is given as

```
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

PDDL representation:

```
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))
```

Result

Search	Expansions	Goal Tests	New nodes	Plan Lengh	Time elapsed
Breadth first search	43	56	180	6	0.036
Breadth first tree search	1458	1459	5960	6	1.310
Depth first graph search	12	13	48	12	0.010627

Search	Expansions	Goal Tests	New nodes	Plan Lengh	Time elapsed
Depth limited search	101	271	414	50	0.10
A* with h_1	55	57	224	6	0.045
A* with h_ignore preconditions	55	57	224	6	0.041
A* with h_pg_levelsum	41	43	170	6	3.2977

Optimal solution:

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

Analysis:

This problem is relatively simple. It only has two cargos, two planes, and two airports. Straightforward breadth first graph search (BFS) and depth first graph search (DFS) work best in this case.

Problem 2

PDDL representation

```
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))
```

Search	Expansions	Goal Tests	New nodes	Plan Length	Time elapsed
Breadth first search	3346	4612	30534	9	16.257
Breadth first tree search	Inf	Inf	Inf	Inf	Inf
Depth first graph search	1124	1125	10017	1085	9.088
Depth limited search	213491	1967093	1967471	50	1128.21
A* with h_1	4605	4607	41839	9	85.72

Search	Expansions	Goal Tests	New nodes	Plan Length	Time elapsed
A* with h_ignore preconditions	4605	4607	41839	9	75.02
A* with h_pg_levelsum	1138	1140	10303	9	1769.60

Optimal solution:

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)

Analysis

This problem has three cargos, three planes, and three airports. The result shows that heuristic algorithms do not lead to less number of node expansion except h_pg_levelsum. Because the optimal solution does not have too many steps, BFS and DFS work better than relatively naive heuristics.

Problem 3

PDDL representation

```
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))
```

Results

Search	Expansions	Goal Tests	New nodes	Plan Length	Time elapsed
Breadth first search	14120	17673	124926	12	133.493
Breadth first tree search	Inf	Inf	Inf	Inf	Inf
Depth first graph search	677	678	5608	660	18.099
Depth limited search	Inf	Inf	Inf	Inf	Inf
A* with h_1	16961	16963	149117	12	818.56

Search	Expansions	Goal Tests	New nodes	Plan Length	Time elapsed
A* with h_ignore preconditions	16961	16963	149117	12	777.74
A* with h_pg_levelsum	2622	2624	23148	12	5852.74

Analysis

This problem has four cargos, two planes, and four airports. The search space is then much more larger than the previous two problems. DFS and BFS then becomes less effective. BFS found an optimal solution but explored too many nodes, whereas DFS expanded less nodes but returned a non-optimal plan. A* search with h_pv_levelsum as heuristic function works best in this case.

Optimal Solution

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)