AIND-Planning heuristic analysis

Problem 1

Optimal plan length : 6
Optimal solution (variants of) :
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Algorithm	Expansions	Goal Tests	New Nodes	Time (seconds)	Plan length
Breadth first search	43	56	180	0.0304	6
Depth first graph search	21	22	84	0.0131	20
Uniform cost search	55	57	224	0.0405	6
A* with ignore preconditions	51	53	208	0.0396	6
A* with levelsum	55	57	224	1.3581	6

Problem 2

Optimal plan length : 9
Optimal solution (variants of) :
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(C1, P1, JFK)
Unload(C3, P3, SFO)
Unload(C2, P2, SFO)

Algorithm	Expansions	Goal Tests	New Nodes	Time (seconds)	Plan length
Breadth first search	3343	4609	30509	13.4323	9
Depth first graph search	624	625	5602	3.1930	619
Uniform cost search	4852	4854	44030	12.7153	9
A* with ignore preconditions	4297	4299	39110	11.6920	9
A* with levelsum	4852	4854	44030	759.8867	9

Problem 3
Optimal plan length : 12
Optimal solution (variants of) :
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(C1, P1, JFK)
Unload(C2, P2, SFO)

Algorithm	Expansions	Goal Tests	New Nodes	Time (seconds)	Plan length
Breadth first search	14663	18098	129631	96.2561	12
Depth first graph search	408	409	3364	1.7435	392
Uniform cost search	18223	18225	159618	53.0608	12
A* with ignore preconditions	16993	16995	149302	53.7762	12
A* with levelsum	18223	18225	159618	4588.3180	12

Performance comparison:

Uninformed planning search:

- **Breadth first search**, in all problems, using the “shallowest nodes in search tree first” logic *could always find **optimal** solution* with relatively average number of expansions. The number of expansions were significantly higher than uninformed graph search because **BFS did not keep track of explored nodes**.
- **Depth first graph** search on the other hand kept track of all explored nodes and hence expanded *substantially lower number of nodes.* Given that it expands the deepest nodes first without any heuristic, it ends up going down a path that is **not optimal** path cost. But at the same time, given that it is implemented using graph and does not use any logic besides goal test it ends up processing a non-optimal path with **extremely fast** speed. Due to the nature of this algorithm, it was also a non-complete search. Without explored nodes , this algorithm would provide substantial space savings.

- Uniform cost search implemented as best first graph search using path cost as the metric also always found the **optimal** path. It expanded significantly higher number of nodes compared to DFS but given that it kept track of explored nodes, it was faster than BFS. Due to the nature of this problem, this algorithm (cheapest path cost/greedy best first search) produces results quite similar to levelsum heuristic, even without any feedback/knowledge. In my comparisons , this seemed like the best uninformed search that guarantees optimal path at relatively good node expansion

Heuristic based A star planning search:

- **A * with ignore preconditions** heuristic (a.k.a. ignore delete lists heuristic) created a relaxed version of the problem. It does not reduce the state space and is still NP hard to find optimal solution, but with no dead ends and no backtracking , **optimal** solution is always guaranteed in polynomial time. And due to the feedback, amount of node expansion is less than the best uninformed search – uniform cost search.
- **A * with levelsum** heuristic uses the subgoal independence assumption and calculates the level cost of each goal clause to create a level sum. While it does produce optimal solution on average, it is inadmissible because periodically it produces a **suboptimal** path. Also, the time cost increases substantially as the conjunctive goal gets bigger, given that it individually calculates heuristic for each goal clause.