Planning Heuristic Analysis

The results are compared based on time complexity, space complexity, and optimality. Clock time is used as a proxy for time complexity and space omplexity is measured by the number of explored and evaluated nodes.

Optimal plans

Problem 1

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

Problem 2

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

Problem 3

```
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Unload(C3, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)
```

No heuristics

Problem	Algorithm	Expansion	Goals	New Nodes	Time	Optimal
P1	DFS	21	22	85	0.02	False

Problem	Algorithm	Expansion	Goals	New Nodes	Time	Optimal
P1	BFS	43	56	180	0.03	True
P1	UCS	55	57	224	0.04	True
P2	DFS	624	625	5602	3.5	False
P2	BFS	3343	4609	30509	13	True
P2	UCS	4840	4842	43918	44	True
P3	DFS	408	409	3364	1.7	False
P3	BFS	14663	18098	129631	97	True
P3	UCS	16963	16965	149136	3.7e+02	True

Across all problems DFS is the fastest (time complexity) but is not resulting in optimal results. BFS is resulting in less expansions and new nodes relative to UCS and it is faster so it's superior to UCS.

BFS is more appropriate for trees that are not very wide since this will result in inefficient memory usage. Time and memory complexity for BFS is O(b^d) were b is the branching factor and d the depth of the most shallow solution. For very deep trees DFS becomes impractical and it's not complete in the tree version.

Using heuristics

Problem	Algorithm	Expansion	Goals	New Nodes	Time	Optimal
P1	A* with h1	55	57	224	0.04	True
P1	A* ignore precond	41	43	170	0.04	True
P1	A* pg levelsum	11	13	50	2.3	True
P2	A* ignore precond	1504	1506	13800	13	True
P2	A* with h1	4840	4842	43918	42	True
P2	A* pg levelsum	85	87	831	2.8e+02	True
P3	A* ignore precond	4723	4725	41835	84	True
P3	A* with h1	16963	16965	149136	3.8e+02	True
P3	A* pg levelsum	378	380	3461	1.8e+03	False

All solutions except for levelsum for problem 3 are optimal. Levelsum also has the highest time complexity but the lowest space complexity. h1 has the highest space complexity that really degrades with a depth of 12 in problem 3. Ignore preconditions seems to be a happy medium with much less time complexity as the depth increases and a relatively reasonable usage of space.

Conclusions

Overall the use of heuristics is helpful with both time and space complexity as evident from the results. However a single heuristics is not appropriate for all problems. Depending on the depth of problem we may want to choose a heuristic that provides optimal solutions and achieves the lowest time and space which is different for the three problems explored here.

Ignore preconditions is a heuristic that adds edges (actions) to the graph in an attempt to make it easier to find a path between the initial and the goal state. This is accomplish by dropping the conditions on a state and assuming that all actions are permissible. Even though this is an instance

of the set covering problem that is NP-hard problem a greedy solution returns a set of logN size where N is the number of literals in the goal, there's no guarantee of admissibility though. Level sum assumes subgoal independence (otherwise inadmissible) and returns the sum of level costs of goals. To calculate the cost we need to sum the values of the first appearance of each goal that is more expensive than counting the number of goal states minus the non-overlapping clauses.

References

- Artificial intelligence: a modern approach (3rd edition). SJ Russell, P Norvig. Prentice Hall, 2009
- When is it practical to use Depth-First Search (DFS) vs Breadth-First Search (BFS)? https://stackoverflow.com/questions/3332947/when-is-it-practical-to-use-depth-first-search-dfs-vs-breadth-first-search-bf
- Completeness of depth-first search https://stackoverflow.com/questions/9250630/completeness-of-depth-first-search