Heuristic Analysis

Optimal plan:

Although all optimal plans shown below are from Breadth First Search, Uniform Cost Search and A* searches also give optimal plans, which are different than the plans shown below but has the same length.

Optimal plan for problem 1:

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

Optimal plan for 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)

Optimal plan for problem 3:

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

Non heuristic search results:

Problem 1:

	Expansions	Goal tests	New Nodes	Plan length	Time
Breadth First Search	43	56	180	6	0.051
Depth First Graph Search	21	22	84	20	**0.024**
Uniform Cost Search	55	57	224	6	0.071

Problem 2:

	Expansions	Goal tests	New Nodes	Plan length	Time
Breadth First Search	3343	4609	30509	9	25.309
Depth First Graph Search	624	625	5602	619	**9.920**
Uniform Cost Search	4852	4854	44030	9	110.637

Problem 3:

	Expansions	Goal tests	New Nodes	Plan length	Time
Breadth First Search	14663	18098	129631	12	205.334
Depth First Graph Search	408	409	3364	392	**5.812**
Uniform Cost Search	18235	18237	159716	12	892.647

Analysis for non-heuristic search result:

1. **Expansions:**

   BFS expands the shallowest unexpanded node, so it expands more nodes than DFS. Also, BFS expands less nodes than Uniform cost search, because uniform cost search continues to search to make sure that there is no shorter path, while BFS stops afer a path is found.

   Rank in order:

   i. DFS
   ii. BFS
   iii. Uniform cost search

2. **Time:**

   With more expansion, more time is taken. Uniform cost also takes more time as it also calculates path cost for each node on the frontier and find the one with shortest path to apply goal test on, which makes uniform cost search even more time-consuming.

   Rank in order:

   i. DFS
   ii. BFS
   iii. Uniform cost search

3. **Optimality:**

   BFS is optimal when all step costs are equal because it always expand the shallowest unexpanded node. Uniform cost search expands the frontier node with lowest path and it makes sure that there are no better solution even after finding 1 solution. DFS always expands to the deepest node and stops right after it finds a path, so it does not guarantee to find an optimal solution.

   Rank in order:

   i. BFS / Uniform cost search
   ii. DFS

Ref: Russel, Stuart and Norvig, Peter. Artificial Intelligence: A Modern Approach 3rd Edition, P.83-85

Ref: Russel, Stuart and Norvig, Peter. Artificial Intelligence: A Modern Approach 3rd Edition, P.85-87

Heuristic search results:

Problem 1:

	Expansions	Goal tests	New Nodes	Plan length	Time
A* with h_1	55	57	224	6	**0.081**
A* with h_ignore_preconditions	41	43	170	6	0.163
A* with h_pg_levelsum	11	13	50	6	6.253

Problem 2:

	Expansions	Goal tests	New Nodes	Plan length	Time
A* with h_1	4852	4854	44030	9	109.442
A* with h_ignore_preconditions	1506	1508	13820	9	**24.109**
A* with h_pg_levelsum	86	88	841	9	412.008

Problem 3:

	Expansions	Goal tests	New Nodes	Plan length	Time
A* with h_1	18235	18237	159716	12	860.882
A* with h_ignore_preconditions	5118	5120	45650	12	**199.930**
A* with h_pg_levelsum	414	416	3818	12	2897.44

Analysis for heuristic search result:

1. **Expansions:**

   H_1 makes the most expansions as it is not truly a heuristic search and is equivalent to Uniform cost search. Level sum makes a more accurate estimation of path length and thus finds a path with less expansions.

   Rank in order:
   i. Level Sum
   ii. Ignore preconditions
   iii. H1

2. **Time:**

   With more expansion, more time is taken. Ignore preconditions takes less time than H_1. The mechanism of LevelSum to calculate path length is very costly, due to this even though Level sum expansion is less, it takes more time.

   Rank in order:
   i. Ignore preconditions
   ii. H1
   iii. Level Sum

3. **Optimality:**

   Both H_1 and A* search are designed to to always find a path and make sure that no shorter path exists. This means an optimal path is guaranteed.

   Rank in order:
   i. Level Sum = Ignore preconditions = H1

Ref: Russel, Stuart and Norvig, Peter. Artificial Intelligence: A Modern Approach 3rd Edition, P.95

Conclusion:

In the experimentation above, there were 3 problems where a series of search methods were applied to find an optimal plan.

Ignore preconditions is the best heuristic which we have used in these problems, primarily because it finds the best solution in minimum time.

Breadth first search was also a candidate, but Ignore preconditions is a better choice, as it showed better results in terms of time and memory. Also ignore preconditions will be better candidate when problems gets more complex.