Heuristic Analysis

Implement a Planning Search: Air Cargo Planning

The following tables show the results gathered after solving the air cargo problems with both uninformed and heuristic based search. The goal of this analysis is to document the results obtained from each search type and find an optimal solution for each air cargo problem, that is; a search algorithm that finds the lowest path among all possible paths from start to goal.

Results

For each set of problems, the row corresponding to the optimal solution has been highlighted.

Problem 1

Search Type	Expansions	Goal Tests	New Nodes	Length	Time (s)	Optimal
Breadth First Search	*43*	*56*	*180*	*6*	*0.033*	*Yes*
Breadth First Tree Search	*1458*	*1459*	*5960*	*6*	*0.991*	*Yes*
Depth First Graph Search	*21*	*22*	*84*	*20*	*0.016*	*No*
Depth Limited Search	*101*	*271*	*414*	*50*	*0.089*	*No*
Uniform Cost Search	*55*	*57*	*224*	*6*	*0.036*	*Yes*
Recursive Best First Search	*4429*	*4230*	*17023*	*6*	*2.875*	*No*
Greedy Best First Graph Search h_1	*7*	*9*	*28*	*6*	*0.004*	*Yes*
A* Search h_1	*55*	*57*	*224*	*6*	*0.042*	*Yes*
A* Search h_ignore_preconditions	*41*	*43*	*170*	*6*	*0.048*	*Yes*
A* Search h_pg_levelsum	*11*	*13*	*50*	*6*	*10.370*	*Yes*

Problem 2

Search Type	Expansions	Goal Tests	New Nodes	Length	Time (s)	Optimal
Breadth First Search	*3343*	*4609*	*30509*	*9*	*17.079*	*Yes*
Breadth First Tree Search						
Depth First Graph Search	*624*	*625*	*5602*	*619*	*4.059*	*No*
Depth Limited Search						
Uniform Cost Search	*4853*	*4855*	*44041*	*9*	*18.235*	*Yes*
Recursive Best First Search						
Greedy Best First Graph Search h_1	*998*	*1000*	*8982*	*15*	*4.986*	*No*
A* Search h_1	*4853*	*4855*	*44041*	*9*	*26.667*	*Yes*
A* Search h_ignore_preconditions	*1450*	*1452*	*13303*	*9*	*11.172*	*Yes*
A* Search h_pg_levelsum	*86*	*88*	*841*	*9*	*597.83*	*Yes*

Problem 3

Search Type	Expansions	Goal Tests	New Nodes	Length	Time (s)	Optimal
Breadth First Search	*14663*	*18098*	*129631*	*12*	*135.890*	*Yes*
Breadth First Tree Search	*18223*	*18225*	*159618*	*12*	*616.23*	*Yes*
Depth First Graph Search	*408*	*409*	*3364*	*392*	*2.300*	*No*
Depth Limited Search						
Uniform Cost Search	*18223*	*18225*	*159618*	*12*	*64.950*	*Yes*
Recursive Best First Search						
Greedy Best First Graph Search h_1	*5578*	*5580*	*49150*	*22*	*18.543*	*No*
A* Search h_1	*18223*	*18225*	*159618*	*12*	*57.27*	*Yes*
A* Search h_ignore_preconditions	*5040*	*5042*	*44944*	*12*	*22.760*	*Yes*
A* Search h_pg_levelsum	*325*	*327*	*3002*	*12*	*3692.360*	*Yes*

Optimal Sequence of Actions

Problem 1 – Breadth First Search

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

Problem 2 – A Search h_ignore_Preconditions*

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

Problem 3 – A Search h_ignore_Preconditions*

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

Discussion of Analysis

As depicted in the tables above, both breadth-first-search as well as uniform-cost-search yield an optimal plan for all three problem spaces with BFS needing less time, less node expansions and goal tests than UCS. BFS and UCS, both always find an optimal plan but the duration of the UCS indicates that the cost assessment of the planning problem is not suited to accelerate the search. Hence, **BFS is the optimal search algorithm**.

Also, a comparison between BFS and DFS shows the clear advantage of BFS in this search space with respect to optimality. While DFS is very fast at finding solutions, the resulting plan is much longer than optimal. Moreover, Depth Limited Search does not find a solution for problems two and three; hence, can be ruled out as a candidate of choice for most problems.

Of all planning approaches that did not guarantee an optimal solution, Greedy Best First Graph search shows the best results with respect to the length of plan. Moreover, number of goal tests as well as node expansions are rather small. The caveat is that in a real life scenario with a much bigger problem space, the plan lengths obtained are still quite distant from optimum.

The ignore_preconditions heuristic adds edges to the graph by ignoring all preconditions which increases the number of possibilities and hence, makes it easier to find a path. level_sum is a heuristic based on the information implemented in the plan graph. The heuristic estimates the expected cost to reach a goal, by counting the levels it takes to reach the goal. In order to not spoil the count, a serial graph is used which does not have more than one action per level. As depicted in the above tables, **A* searches provides an optimal solution.** It's also worth noting that the 'h_pg_levelsum' heuristic did in overall perform poorly, most likely due to the heuristic being too complex.

Choosing the Best Planning Approach

When the goal is just to find a path with optimality not being the priority, DFS is the best candidate as it's both complete and uses less resources. **When optimality is a must BFS or A* search with ignore_preconditions heuristic should be considered**, since they both direct the solver towards to optimal path. A* search with level_sum heuristic seems promising but is computationally expensive.

References

1. http://cs.lmu.edu/~ray/notes/usearch/
2. Quiz: Tree Search, Udacity Artificial Intelligence Nanodegree, Lesson 7 (Search), Video 8.
3. Quiz: Uniform Cost Search, Udacity Artificial Intelligence Nanodegree, Lesson 7(Search), Video 16-21.
4. Quiz: A * Search, Udacity Artificial Intelligence Nanodegree, Video 27-32 of Lesson 7 (Search).
5. Russell, S. and Norvig, P. Artificial Intelligence, A Modern Approach, Third Edition, Sec 10.3.1 Planning graphs for heuristic estimation, LEVEL SUM HEURISTIC
6. Russell, S. and Norvig, P. Artificial Intelligence, A Modern Approach, Third Edition, Sec 10.2.3 Heuristics for planning, IGNORE PRECONDITIONS HEURISTIC