Hao Chen

Udacity Artificial Intelligence

Heuristic Analysis

There are several optimal plans for problems 1 to 3. As an example shown below, the optimal plan lengths for problem 1, 2, 3 are 6,9,12 actions respectively.

Problem1:

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

Problem2:

```
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)
```

Problem3:

```
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)
```

Next, I compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1, 2, and 3. The searches I analyzed are breadth first search, depth first search, and uniformed cost search. Detail performances are shown below.

Problem1:

	Path Length	Time	Node Expansions
Breadth first search	6	0.043	43
Depth first graph search	20	0.02	21
Uniform cost search	6	0.05	55

Problem2:

	Path Length	Time	Node Expansions
Breadth first search	9	17.18	3343
Depth first graph search	619	4.2	624
Uniform cost search	9	16.1	4761

Problem 3:

	Path Length	Time	Node Expansions
Breadth first search	12	130.59	14663
Depth first graph search	392	2.37	408
Uniform cost search	12	68.5	17783

Lastly, I compared and contrast heuristic search result metrics using A* with the "ignore preconditions" and "level-sum" heuristics for Problems 1, 2, and 3.

Problem1:

	Path Length	Time	Node Expansions
A* w/ ignore preconditions	6	0.05	41
A* w/ level sum	6	1.3	11

Problem2:

	Path Length	Time	Node Expansions
A* w/ ignore preconditions	9	5.75	1450
A* w/ level sum	9	266.03	86

Problem3:

	Path Length	Time	Node Expansions
A* w/ ignore preconditions	12	22	5003
A* w/ level sum	12	1387	311

In conclusion, for non-heuristic search planning, breadth first search and uniform cost search yield an optimal action plan. However, depth first graph search is the fastest and uses the least memory (node expansions). For heuristic search planning, both heuristics yield to an optimal action. A* search with ignore precondition heuristic is faster. On the other hand, A* search with level sum heuristic uses the least amount of memory. By comparing all the heuristics, we see that A* search with ignore precondition would be the best solution for the air cargo problem.

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Time	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Space	b^d	b^d	bm	bl	bd	$b^{d/2}$
Optimal?	Yes	Yes	No	No	Yes	Yes
Complete?	Yes	Yes	No	Yes, if $l > d$	Yes	Yes

Figure 3.18 Evaluation of search strategies. b is the branching factor; d is the depth of solution; m is the maximum depth of the search tree; l is the depth limit.

[1]

[1] Artifical Intellegence: A Modern Approach. S. Russel. P. Norvig. Chapter 3: Solving problems by searching