Planning Search: Written Analysis

Udacity AI Nanodegree - Term 1: Project 3

Author: Stephen Rawson

Problem 1:

```
Init(At(C1, SFO) ∧ At(C2, JFK)
       ∧ At(P1, SFO) ∧ At(P2, JFK)
       ∧ Cargo(C1) ∧ Cargo(C2)
       ∧ Plane(P1) ∧ Plane(P2)
       ∧ Airport(JFK) ∧ Airport(SFO))
Goal(At(C1, JFK) ∧ At(C2, SFO))
```

The Optimal Plan:

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

Note: The Optimal Plan was taken from Greedy Best First Graph Search.

Compare And Contrast Non-Heuristic Algorithms

Table 1, provides insight into the performance of seven search algorithms. Results label with green highlight the optimal solutions for the problem. The number of expansions and new nodes varied greatly amongst the optimal search algorithms. Greedy Best First Graph Search was the top performing algorithm for this problem. It had the lowest number of expansions and new nodes which lead to fast overall time. Depth First Graph Search also had the second lowest expansions and new nodes, and saw a fast overall time, but resulted in Plan Length of 20 which is more than double that of optimal Plan Length of 6.

Problem 1 Non-Heuristic Search Results					
Search Algorithm	Expansions	New Nodes	Plan Length	Time (s)	Optimal
Breadth First Search	43	180	**6**	0.030	**Yes**
Breadth First Tree Search	1458	5960	**6**	0.945	**Yes**

Depth First Graph Search	21	84	20	0.017	No
Depth Limited Search	101	414	50	0.089	No
Uniform Cost Search	55	224	6	0.035	Yes
Recursive Best First Search	4229	17023	6	2.702	Yes
Greedy Best First Graph Search	7	28	6	0.005	Yes

Table 1. Problem non-heuristic search result metrics.

Compare and Contrast Heuristic Algorithms

Table 2, provides an insight into the performance of A* Search with 3 different heuristics. All 3 algorithms achieved the optimal plan length of 6. The level sum heuristic was the slowest of all 3 by more than a factor of 2, but it only expanded to 11 nodes versus the next closets of 41.

Problem 1 Heuristic Search Results					
Search Algorithm	Expansions	New Nodes	Plan Length	Time (s)	Optimal
A* Search with h_1	55	224	6	0.03487534	Yes
A* Search (ignore precondition)	41	170	6	0.037547264	Yes
A* Search (levelsum)	11	50	6	0.887974935	Yes

Table 2. Problem 1 heuristic search result metrics.

Problem 2:

```
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL)
        ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL)
        ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3)
        ∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3)
        ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL))
Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))
```

The Optimal Plan:

```
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)
```

Note: The Optimal Plan was taken from A* Ignore Preconditions Search.

Compare And Contrast Non-Heuristic Algorithms

Table 3 are the results of all seven non-heuristic search algorithms. Breadth First Tree Search, Depth Limited Search, and Recursive Best First Search timed out after 10 minutes. Greedy Best First Graph Search (GBFGS) had the lowest time, number of expansions and new nodes. Its performance was 7x faster than Depth First Search's result which did not yield the optimal plan length. GBFGS had 6x less expansions then next optimal algorithm, Breadth First Search (BFS).

Problem 2 Non-Heuristic Search Results					
Search Algorithm	Expansions	New Nodes	Plan Length	Time (s)	Optimal
Breadth First Search	3401	31049	9	16.1887	Yes
Breadth First Tree Search	-	-	-	> 600	-
Depth First Graph Search	1192	10606	1138	8.3968	No
Depth Limited Search	-	-	-	> 600	-
Uniform Cost Search	4761	43206	9	11.4653	Yes
Recursive Best First Search	-	-	-	> 600	-
Greedy Best First Graph Search	550	4950	9	1.2882	Yes

Table 3. Problem non-heuristic search result metrics.

Compare and Contrast Heuristic Algorithms

Table 4 results for A* search with various heuristics show that levelsum only expanded to 86 and utilized only 841 new nodes. Ignore precondition heuristic expanded to 1460 with 13303 new nodes. Both heuristics achieved the optimal plan length but the ignore precondition took only 4.94 seconds vs. 169 seconds.

Problem 2 Heuristic Search Results					
Search Algorithm	Expansions	New Nodes	Plan Length	Time (s)	Optimal
A* Search with h_1	4761	43206	9	13.3419	Yes

A* Search (ignore precondition)	1450	13303	9	4.9447	Yes
A* Search (levelsum)	86	841	9	169.8756	Yes

Table 4. Problem 2 heuristic search result metrics.

Problem 3:

```
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD)
       ∧ At(P1, SFO) ∧ At(P2, JFK)
       ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4)
       ∧ Plane(P1) ∧ Plane(P2)
       ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD))
Goal(At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))
```

The Optimal Plan:

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

Note: The Optimal Plan was taken from Uniform Cost Search.

Compare And Contrast Non-Heuristic Algorithms

Table 5 provides insight into 4 search algorithm results. BFS and Uniform Cost search both achieved the optimal plan length of 12. Uniform Cost's performance time was 50% less than BFS. BFS had a lower number of expansions 14491 versus Uniform Cost's 17783.

Problem 3 Non-Heuristic Search Results					
Search Algorithm	Expansions	New Nodes	Plan Length	Time (s)	Optimal
Breadth First Search	14491	128184	12	100.658	Yes
Breadth First Tree Search	-	-	**-**	**> 600**	**-**
Depth First Graph Search	2099	17558	2014	21.286	No

Depth Limited Search	-	-	**-**	**> 600**	**-**
Uniform Cost Search	17783	155920	**12**	54.121	**Yes**
Recursive Best First Search	-	-	**-**	**> 600**	**-**
Greedy Best First Graph Search	4031	35794	**22**	11.370	**No**

Table 5. Problem 3 non-heuristic search result metrics.

Compare and Contrast Heuristic Algorithms

Table 6 results for A* search with various heuristics show that levelsum only expanded to 323 and utilized only 2983 new nodes. Ignore precondition heuristic expanded to 5003 with 44586 new nodes. Both heuristics achieved the optimal plan length but the ignore precondition took only 16.16 seconds vs. 944.37 seconds.

Problem 3 Heuristic Search Results					
Search Algorithm	Expansions	New Nodes	Plan Length	Time (s)	Optimal
A* Search with h_1	17783	155920	**12**	49.26727635	**Yes**
A* Search (ignore precondition)	5003	44586	**12**	16.19452596	**Yes**
A* Search (levelsum)	323	2983	**12**	944.3700594	**Yes**

Table 6. Problem 3 heuristic search result metrics.

Conclusion

The Best Heuristic used in these problems really depends on two things: performance and space. If you require the optimal plan quickly and don't mind how much memory space it takes then the A* Search with ignore precondition is your solution. If memory space is limited and time requirements aren't stressed then the level sum heuristic would be ideal. Overall, though I would say that **ignore precondition** was ideal planning search method because tt nearly performed as fast or faster than all 7 non-heuristic algorithms in all three problems. Also, ignore precondition in problems two and three had significantly less expansions and new nodes then any non-heuristic methods. In problem one, ignore preconditions slower, had higher expansions and new nodes then by Greedy Best First Graph Search. However, Greedy Best First Results resulted in non optimal plan lengths in problems two and three. This lead me to conclude that ignore precondition heuristic would be best used in these problems.