Planning Search Heuristic Analysis

René Straub – AIND Project 3

Introduction

This analysis is showing the results after comparing ten different planning strategies. The problem can be classified as deterministic logistic planning, that needs to be solved in three different increasing search spaces.

Optimal Plan

Optimal plans for all problems were figured out by different strategies. The following table is showing the steps for BFS (Breadth First Search).

Problem 1	Problem 2	Problem 3
Path length: 6	Path length: 9	Path length: 12
Load(C2, P2, JFK)	Load(C2, P2, JFK)	Load(C2, P2, JFK)
Load(C1, P1, SFO)	Load(C1, P1, SFO)	Load(C1, P1, SFO)
Fly(P2, JFK, SFO)	Load(C3, P3, ATL)	Fly(P2, JFK, ORD)
Unload(C2, P2, SFO)	Fly(P2, JFK, SFO)	Load(C4, P2, ORD)
Fly(P1, SFO, JFK)	Unload(C2, P2, SFO)	Fly(P1, SFO, ATL)
Unload(C1, P1, JFK)	Fly(P1, SFO, JFK)	Load(C3, P1, ATL)
	Unload(C1, P1, JFK)	Fly(P1, ATL, JFK)
	Fly(P3, ATL, SFO)	Unload(C1, P1, JFK)
	Unload(C3, P3, SFO)	Unload(C3, P1, JFK)
		Fly(P2, ORD, SFO)
		Unload(C2, P2, SFO)
		Unload(C4, P2, SFO)

Table 1: Optimal paths

Discussion non heuristic search

The tables below is showing the metrics for 5 non heuristic search strategies. As mentioned in the project introduction, a search is canceled if it takes longer then 10 minutes. This was true for Breadth First Tree Search and Depth Limited Search, so there are no results listed.
Among the strategies providing results, Depth First Graph Search is the fastest for all problems, but never provides an optimal path. It needs significantly fewer expansions to find the goal, but unfortunately the path length is similar to the number of expansions.
BFS (Breadth First Search) and UCS (Uniform Cost Search) always find an optimal path. For more complex problems, UCS takes less time comparatively.

Problem 1				
Search Strategy	**Path Length**	**Time (s)**	**Expansions**	**Optimal**
Breadth First Search	6	0.044	43	Yes
Breadth First Tree Search	6	1.337	1458	Yes
Depth First Graph Search	12	0.012	12	No
Depth Limited Search	50	0.126	101	No
Uniform Cost Search	6	0.053	55	Yes

Table 2: Problem 1 without heuristics

Problem 2				
Search Strategy	**Path Length**	**Time (s)**	**Expansions**	**Optimal**
Breadth First Search	9	19.294	3343	Yes
Breadth First Tree Search	-	-	-	-
Depth First Graph Search	575	3.911	582	No
Depth Limited Search	-	-	-	-
Uniform Cost Search	9	16.608	4852	Yes

Table 3: Problem 2 without heuristics

Problem 3				
Search Strategy	**Path Length**	**Time (s)**	**Expansions**	**Optimal**
Breadth First Search	12	128.944	14663	Yes
Breadth First Tree Search	-	-	-	-
Depth First Graph Search	596	4.121	627	No
Depth Limited Search	-	-	-	-
Uniform Cost Search	12	70.977	18223	Yes

Table 4: Problem 3 without heuristics

Discussion heuristic search

The tables below show the metrics for 5 heuristic search strategies. It is important to note, that „h_1" is only returning 1 and therefore can´t be considered as a „real" heuristic. All A* strategies show an optimal path length.
Regarding Time and Expansions, A* h_1 performs worst among all A* strategies. It´s metrics are similar to UCS.
It is interesting to directly compare „ignore_preconditions" and „levelsum". Their metrics for time and expansions seem to be reciprocal. While „ignore_preconditions" is faster it needs more node expansions. Like the name says it ignores preconditions, thus leading to more actions at each level, resulting in more expansions.
Greedy BFS h_1 perfomes like „ignore_preconditions" expect it does´nt lead to an optimal path length.

Problem 1				
Search Strategy	**Path Length**	**Time (s)**	**Expansions**	**Optimal**
Recursive BFS h_1	6	4.194	4229	Yes
Greedy BFS h_1	6	0.008	7	Yes
A* h_1	6	0.053	55	Yes
A* h_ignore_preconditions	6	0.054	41	Yes
A* h_pg_levelsum	6	0.686	11	Yes

Table 5: Problem 1 with heuristics

Problem 2				
Search Strategy	**Path Length**	**Time (s)**	**Expansions**	**Optimal**
Recursive BFS h_1	-	-	-	-
Greedy BFS h_1	17	3.536	990	No
A* h_1	9	16.233	4852	Yes
A* h_ignore_preconditions	9	5.741	1450	Yes
A* h_pg_levelsum	9	60.584	86	Yes

Table 6: Problem 2 with heuristics

Problem 3				
Search Strategy	**Path Length**	**Time (s)**	**Expansions**	**Optimal**
Recursive BFS h_1	-	-	-	-
Greedy BFS h_1	22	21.669	5578	No
A* h_1	12	73.2144	18223	Yes
A* h_ignore_preconditions	12	23.369	5040	Yes
A* h_pg_levelsum	12	298.219	318	Yes

Table 7: Problem 3 with heuristics

Conclusion

After performing tests for ten different strategies, it appears that heuristic search strategies perform better than non heuristic strategies as the search space increases. Under all strategies calculating an optimal path length, A* with the ignore preconditions heuristic performs best regarding the time. UCS seems to be a good choice if you need an optimal path, but can´t use heuristics. The Depth First Graph Search is the fastest, but does a really bad job in calculating short paths. Nonetheless it might be usefull in domains with real time like requirements, were it is most important to deliver a result very fast.

References

Udacity, Artificial Intelligence Nanodegree Lesson 10-14, 2017.

Künstliche Intelligenz: Ein moderner Ansatz, S. Russel & P. Norvig, Pearson Education, Chapter 10, 2012.