Heuristic Analysis

Yung-Chun, Lu
19/03/2017

Discussion

In non-heuristic search, we could see that the number of expanded nodes is fewest because its space complexity is lowest comparing to other two algorithms. In heuristic search, we could see that level-sum is much better heuristic than ignore-preconditions because of the number of expanded nodes is much less. That’s because ignore-preconditions does not consider some actions may undo the effects of others.

I want to use following four different aspects to evaluate the performance of algorithms

- **Completeness**: Is the algorithm guaranteed to find a solution when there is one?
- **Optimality**: Does the strategy find the optimal solution?
- **Time Complexity**: How long does it take to find a solution?
- **Space Complexity**: How much expanded node is needed to perform the search?

For non-heuristic search:

- Completeness: All algorithms are guaranteed to find a solution in these 3 problems.
- Optimality: Based on the length of plan, only breadth-first and uniform-cost can give the optimal solution.
- Time Complexity: Based on the time elapsed, depth-first outperforms than other 2 algorithms in these 3 problems.
- Space Complexity: Based on the number of node expansions, depth-first still outperforms than other 2 algorithms in these 3 problems.

For heuristic search:

- Completeness: All algorithms are guaranteed to find a solution in these 3 problems.
- Optimality: Based on the length of plan, all algorithms can give the optimal solution.
- Time Complexity: Based on the time elapsed, ignore-preconditions can do better than level-sum.
- Space Complexity: Based on the number of node expansions, level-sum can do better than ignore-preconditions.

In my situation, the order of importance of aspects is **Optimality > Time Complexity > Space Complexity**. So these are the algorithms I will use

- For Air_Cargo_P1: Breadth First.
- For Air_Cargo_P2: Breadth First.
- For Air_Cargo_P3: A* with ignore preconditions.

Reference:

- Russell, Stuart and Norvig, Peter. Artificial Intelligence: A Modern Approach 3rd Edition, P. 87.
- Measuring Problem Solving Performance

Table 1 - Non-heuristic Search

Air_Cargo_P1				
Algorithm	Node Expansions	Goal Tests	Time Elapsed(Sec)	Plan Length
Breadth First	43	56	0.045	6
Depth First	21	22	0.025	20
Uniform Cost	55	57	0.068	6

Air_Cargo_P2				
Algorithm	Node Expansions	Goal Tests	Time Elapsed(Sec)	Plan Length
Breadth First	3343	4609	16.805	9
Depth First	624	625	4.042	619
Uniform Cost	4852	4854	48.604	9

Air_Cargo_P3				
Algorithm	Node Expansions	Goal Tests	Time Elapsed(Sec)	Plan Length
Breadth First	14663	18098	125	12
Depth First	408	409	2.092	392
Uniform Cost	18235	18237	625.505	12

Table 2 - Heuristic Search

Air_Cargo_P1

Algorithm	Node Expansions	Goal Tests	Time Elapsed(Sec)	Plan Length
A* with ignore preconditions	41	43	0.060	6
A* with level-sum	11	13	2.581	6

Air_Cargo_P2

Algorithm	Node Expansions	Goal Tests	Time Elapsed(Sec)	Plan Length
A* with ignore preconditions	1506	1508	20.316	9
A* with level-sum	86	88	315.120	9

Air_Cargo_P3

Algorithm	Node Expansions	Goal Tests	Time Elapsed(Sec)	Plan Length
A* with ignore preconditions	5118	5120	113.950	12
A* with level-sum	408	410	1894.661	12

Optimality of Solution for Air Cargo Problems

Problem	**Air_Cargo_P1**	**Air_Cargo_P2**	**Air_Cargo_P3**
Algorithm	Breadth First	Breadth First	A* with ignore preconditions
Plan	1. Load(C1, P1, SFO) 2. Load(C2, P2, JFK) 3. Fly(P2, JFK, SFO) 4. Unload(C2, P2, SFO) 5. Fly(P1, SFO, JFK) 6. Unload(C1, P1, JFK)	1. Load(C1, P1, SFO) 2. Load(C2, P2, JFK) 3. Load(C3, P3, ATL) 4. Fly(P2, JFK, SFO) 5. Unload(C2, P2, SFO) 6. Fly(P1, SFO, JFK)	1. Load(C2, P2, JFK) 2. Fly(P2, JFK, ORD) 3. Load(C4, P2, ORD) 4. Fly(P2, ORD, SFO) 5. Unload(C4, P2, SFO) 6. Load(C1, P1, SFO)

		7. Unload(C1, P1, JFK) 8. Fly(P3, ATL, SFO) 9. Unload(C3, P3, SFO)	7. Fly(P1, SFO, ATL) 8. Load(C3, P1, ATL) 9. Fly(P1, ATL, JFK) 10. Unload(C3, P1, JFK) 11. Unload(C1, P1, JFK) 12. Unload(C2, P2, SFO)