Heuristic functions analisys

Stanislav Levental

March 20, 2017

Abstract

This document includes comparison of different heuristics I've tried and their performance characteristics.

1 Heuristics

1.1 Problem 1

To exchange cargo between to airports we just need to load it, fly and unload in the other airport, optimal plan will look like:

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

This plan has optimal length of 6 and all approaches except depth-first search were successful in finding this solution. Breadth-first search was the fastest among others - and this is expected because state space is very limited, so simple algorithms may perform better, even if asymptotically they are worse.

Algorithm	Expansions	Goal Tests	Plan (act)	Time (seconds)
breadth-first search	43	56	**6**	**0.142**
depth-first search	12	**13**	12	0.038
uniform-cost search	55	57	**6**	0.174
A^* **(level-sum)**	**11**	**13**	**6**	6.3
A^* **(no preconditions)**	41	43	**6**	0.15

1.2 Problem 2

Problem 2 has bigger search space and it's optimal execution plan is larger, means that we needed to look further in a search space.

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

Optimal plan for *Problem 2* was found by all algorithms except depth-first search, this algorithm doesn't guarantee finding an optimal solution - that's what we can observe in test results as well. Solutions which are using heuristic functions are more optimal in terms of node expanding (if we won't take depth-first search into account) and in term of time execution. However building a planning graph takes more time than I would expect and even with the most optimal solution in terms of node visiting - heuristic calculation of level-sum is very costly.

Algorithm	Expansions	Goal Tests	Plan (act)	Time (seconds)
breadth-first search	3343	4609	**9**	45.8
depth-first search	582	583	575	8.6
uniform-cost search	4853	4855	**9**	93.4
A^* **(level-sum)**	**86**	**88**	**9**	741
A^* **(no preconditions)**	1506	1508	**9**	**30.2**

1.3 Problem 3

Problem 3 is the biggest among three problems researched, since search space growths exponentially even a small increase in a state impacts search space dramatically. Optimal solution will look like this:

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

After the last experiment we can be sure that we are observing degradation in performance of simple solutions which are not using heuristics and less degradation in more optimal solutions - which are faster and more efficient in terms of node visit number. Level-sum heuristic took too much time for the last problem because it uses Planning Graph and builds it every time we need to evaluate the action, this behavior could be optimized using memoization - storing pre-calculated heuristic value for each state or building Planning Graph on-the-fly, until we found the goal. I believe that this may make this heuristic work faster.

Algorithm	Expansions	Goal Tests	Plan (act)	Time (seconds)
breadth-first search	14663	18098	**12**	296
depth-first search	627	628	596	11.04
uniform-cost search	18223	18225	**12**	649
A^* **(no preconditions)**	5118	5120	**12**	**172**