

Review of Oskar Andersson's HW1 Part B

Mattias Mikkola mmikkola@kth.se

Peter Boström pbos@kth.se

2010-09-13

1 Review

1.1

I think depth first will find a solution faster than breadth first will. This is due to that all goal states are in the deepest possible level of the search tree. Breadth first search will have to expand all levels above before it may generate a solution, while depth first search will have the possibility to find a solution after just a few steps in the tree.

Good (2), the section explains perfectly why DFS is a better choice.

1.2

4112 nodes expanded.

Good (2). The algorithm was a classic DFS.

1.3

18365 nodes expanded.

Good (2). Textbook BFS.

1.4

6855 nodes expanded. The evaluation function used was a function returning the number of dirty cells surrounding the current position of the robot. A lot of dirty cells yields a high cost.

Good (2). The heuristic prefers some moves over others, according to specs. However the heuristic generally doesn't perform well at all. For larger boards that we tried, DFS completely blows best-first out of the water. Observing the answers given by both DFS and BFS would give that a sequence of "folding" moves would've been better. The task as we interpreted it however, was only to implement a heuristic, not necessarily a good one.

1.5

No, it won't. For example, testing the different algorithms on a 7x7 board leaving a maximum of 8 cells dirty, the best-first algorithm is better than depth first. Breadth first search is the worst in any case though.

Good (2), a case which proves the ranking to differ is presented. We tested the algorithm for that example, and best-first expanded 10 nodes, while BFS expanded 18. Note that this is an edge case which depends on the move order of the DFS. Generally best-first is worse.