AIR - Homework 6

Maximilian Mensing
Torsten Jandt

November 7, 2016

1 Uniform-Cost Search

Uniform-cost search (*UCS*), breadth first search (*BFS*) and depth first search (*DFS*) require no heuristics. In UCS nodes are only distinguished by $g(n)$. This means, when expanding to a new node, UCS expands to the node, which has the lowest path cost so far.

BFS is a Case of UCS BFS is a special version of UCS. In case the expansion of any node costs the same, UCS behaves like BFS.

UCS, DFS and BFS are Special Cases of Greedy Best-First Search BFS is a case of Greedy Best-First Search, when the cost to reach any node in the next layer is higher than the cost to reach a nodes in the same layer. This way the algorithm traverses the search tree layer wise.

Since greedy search algorithms do not use backtracking, the first node in the next layer, which is reachable from a current node, has to be reachable at the lowest path cost. In this case, DFS and Greedy Best-First Search would behave alike until the goal has been reached.

Greedy Best-First search always chooses the node which seems to be best next. UCS does basically the same, but is able to backtrack. If there is no need to backtrack their behavior is alike.

UCS is a Case of A* In general, UCS is the uninformed version A*. While UCS has no information at any point how close it is to the goal, this information if provided to A* by heuristics.

2 A*

Completeness of A* A search algorithm is complete, if it can tell whether the goal is reachable or not. If it is reachable it is able to find this path. If the algorithms gets stuck in a loop it is not complete. To avoid loops, A* needs an admissible heuristic, like the manhattan distance.

Termination of A* A* ends the search, once the goal has been found, or it has been discovered tat the goal is unreachable

A* with Consistent Heuristics If a consistent heuristic is used with the A* algorithm, then A* expands nodes in order of increasing $f(n)$ values.

3 Implementation of the Algorithms

Performance of IDA on three maps

	Greedy + Misplaced	Greedy + Manhattan	A* + Misplaced	A* + Manhattan
Iterations	658	373	38470	4254
Fringe	427	243	15740	2278
Visited	657	372	38469	4253
Depth	44	56	26	26

Table 1

Heuristics The misplaced tiles heuristic only considers but not the distance of a tile to it's designated position. This distance is used by the Manhattan distance heuristic. Misplaced tiles is therefore less expressive in regard to the distance of the goal state. Another problem which arises when misplaced tiles is used, is that it is likely that several states which are reachable have the same fitness value. This might happen too when the Manhattan distance is used but due to the lager fitness space it is less likely. This makes is easier for an algorithm to choose a new state which is closer to the goal.

A* and Greedy search A* always finds an optimal path, but doing so it is usually slower than other algorithms. It obtains a global optimum. Greedy search on the other hand always expands to the node which increases the path cost as little as possible. These characteristics allow greedy search to find solutions faster, while A* is able to find better solutions, at the cost of a higher execution time.