Artificial Intelligence Practicum 1
Path planning with A*

Garben Tanghe
Olivier Van den Nest
Group 16

October 20, 2019

Question 1

The number of vertices is 264,346 and the number of arcs is 733,846. They are both verified to be correct.

Question 2

The A* search algorithm avoids taking paths which were already expensive at the time of searching a new node. More concretely, it calculates a function $f(n)$ for each new explored node n, where $f(n) = g(n) + h(n)$. $g(n)$ indicates the cost to reach node n from the start, whereas $h(n)$ is a chosen heuristic function to approximate the remaining distance from node n to the goal. This heuristic is **admissible**, meaning that $0 \leq h(n) \leq h^*(n)$, with $h^*(n)$ the true cost from node n to the goal. $h(n)$ never overestimates the true cost from node n to the goal. A non-admissible heuristic overestimates the true cost from the goal, but it may expend much fewer nodes than an admissible heuristic, resulting in a lower total cost (search cost + path cost).

At each iteration of the A* algorithm, the node with the lowest f value is chosen. Then, new neighboring nodes are explored and the total cost (f) is calculated for them. The algorithm ends if the node with the lowest cost is equal to the goal.

In graph search, A* needs a stronger requirement than admissibility to assure optimality. **Consistent** heuristics make sure that the A* algorithm finds the optimal path in graph search. It means that the heuristic function (h) meets the following condition for all nodes n: $h(n) \leq c(n, a, n') + h(n')$, where n' is a new explored node, and $c(n, a, n')$ the cost from node n to node n' following arc a. This results in f being non-decreasing along any path.

Note: Admissibility is a necessary condition for consistency.

There can be concluded that an admissible heuristic is sufficient to assure optimality of A* in tree search, whereas a consistent heuristic is needed to make sure A* is optimal in graphs.

Question 3

The total distance, found by the implemented A* algorithm is 191 998.0 cm, which is the correct and optimal distance To go from node 189543 to 246008, 132 nodes are passed through. This excludes the starting node, but includes the goal.

Question 4

The A* algorithm with $h(n) = 0$ will explore a lot more nodes since it has no intuition on the remaining distance. Instead, all neighboring nodes from a node which has the lowest cost yet will be queried. It prefers a path with 10 nodes and current cost 9 over a path of 1 node with current cost 10. This definitely leads to the optimal solution, but the absence of a meaningful heuristic will cause the runtime to rise due to the extended search space.

Note: The A algorithm with this heuristic is equivalent to **Dijkstra's Shortest Path (DSP)** algorithm.*

Question 5

When using the Manhattan distance as the heuristic cost, the total distance becomes 220 463.0 cm, which is far from optimal. This heuristic is not longer admissible, which causes the algorithm to take the wrong path. The length of the path that was found is now 184, which is 52 nodes more than when the Euclidean distance was used.

The complexity of the algorithm remains unchanged. Although there must be said that calculating the Manhattan distance is much faster on a computer than the Euclidean distance, as the square root operator is quite costly.

Note: If all nodes lay on a grid, the Manhattan distance will be closer to the actual cost of an arc.

Question 6

If an admissible, but inconsistent heuristic is used, optimality is not guaranteed. A path with length 194 734.0 cm and 145 nodes is found when using the following heuristic: when the sum of the 2 nodes is even, the heuristic is 0. In case it is uneven, the Euclidean distance is used.

Question 7

When using Euclidean distance, Manhattan distance or no heuristic, the same path with length 269 987.0 s and 132 steps is found. This path is again the optimal path. As discussed in question 3, the latter is equivalent to Dijkstra's Shortest Path algorithm and is a lot slower than when using Euclidean or Manhattan distance. This is because the latter has no idea in what direction to search and will thus move in all directions until the goal is found ‘by accident’.

When using the admissible, inconsistent heuristic, the optimal path is not found, but it is faster than when using Euclidean distance. Here, there is thus a trade-off between speed and cost. The found path was (293 206.0 s and 104 steps).

Question 8

The Euclidean heuristic was chosen, since it is consistent and still runs very fast for this specific problem. However, other algorithms have specific advantages as well. Non-admissible or inconsistent heuristics may have a significant speedup compared to consistent ones, and $h = 0$ (DSP algorithm) is still widely used in practice because of its simplicity and optimality.