Answer sheet homework 1, 4SC000, TU/e, 2018-2019

Answer to problem 3 (1/3)[1]

Group 2	
Student	**Student number**
Bob Clephas	1271431
Tom van de Laar	1265938
Job Meijer	1268155
Marcel van Wensveen	1253085

In this document two different path planning algorithms are evaluated. Those algorithms (Dijkstra and A*) determine the ideal path given a map split in nodes, a starting point and a goal. In this document we will prove that we can use the Dijkstra algorithm to obtain the same result as when we use the A*, although the two algorithms have different input arguments. Both algorithms iterate over an list of $OPEN$ nodes and decide every iteration which node they will evaluate during that iteration. The difference between the two algorithms is the equation to decide which node will be evaluated. In this document we will first explain both algorithms and secondly prove that we can use the Dijkstra algorithm with adjusted inputs to obtain the same path as when we use A*.

The initial conditions for both the Dijkstra and the A* algorithm are equal to

$$d_i = \infty for\ i \in \mathcal{V} - \{p\}, d_p = 0,\ and\ OPEN = \{p\},\ where\ p\ is\ the\ initial\ node \tag{1}$$

In the Dijkstra algorithm the next node i that will be evaluated is the node with the minimum estimate d_i in the $OPEN$ list. For this node all the possible transitions to surrounding nodes are evaluated in terms of the cost w_{ij} to reach that node from node i. If $d_i + w_{ij} < d_j$ where d_j is the iterated cost to reach node j, the cost d_j will be updated and the transition is stored in $\beta(j) = i$.

For the A* algorithm the next node that will be opened is the node i that has the minimal cost of $d_i + h(i)$ in the $OPEN$ list, where d_i is the cost to reach the node and $h(i)$ is the estimated cost from node i to the final node t. For the chosen node all the possible transitions to the surrounding nodes are evaluated in the same way as in the Dijkstra algorithm and the way to store possible transitions with lower costs are also the same as in the Dijkstra algorithm.

[1] You can print, write down your answers and scan or you can create a similar pdf document with the same number of pages in word or latex with your answers.

The goal is to proof that the Dijkstra and A* algorithms output the same optimal path and $OPEN$ list in the situation that specific (adjusted) weights are used in the Dijkstra algorithm and the weights of A* are the original weights. Because the way of calculating the transition costs and the way of storing the best transitions are equal in both algorithms it is sufficient to proof that the criteria to decide which node has to be opened are equal in both algorithms. The specific weights for the Dijkstra algorithm are noted as $\bar{w}_{ij}$ and are equal to $w_{ij} + h(j) - h(i)$ where $h(i)$ and $h(j)$ are the estimated costs from node i and j respectively to the final node.

If the next node that is evaluated is equal in both algorithm, the node with the lowest cost must be the same for both algorithms. The cost of node k is expressed as $d(k)$.

In the case of the changed Dijkstra algorithm the cost is equal to the sum of the previous (adjusted) costs $\bar{d}(k)$ to reach that node. For a node reached by a path expressed by $\{p, ..., k\}$ this is equal to

$$\begin{aligned}
\bar{d}(k) &= \sum_{i=p}^{k-1}(\bar{w}_{ik}) \\
&= \sum_{i=p}^{k-1}(w_{ij} + h(j) - h(i)) \\
&= w_{p1} + h(1) - h(p) + w_{12} + h(2) - h(1) + ... + w_{k-1k} + h(k) - h(k-1) \\
&= w_{p1} + w_{12} + ... + w_{k-1k} - h(p) + h(k) \\
&= \sum_{i=p}^{k-1}(w_{ik}) - h(p) + h(k)
\end{aligned} \tag{2}$$

$$= d(k) - h(p) + h(k) \tag{3}$$

The term of $h(p)$ in (3) is present in every node, is constant and thus has no effect on the decision which node has the lowest cost. The decision of which node had the minimum cost can therefore be simplified to

$$d(k) + h(k) \tag{4}$$

For the A* algorithm the cost is equal to the actual cost $d(k)$ plus the estimated cost to the goal $h(k)$. For a path from node p to k this is by definition equal to

$$d(k) + h(k) \tag{5}$$

From (4) and (5) it is easily seen that for both algorithms the same equation is used to determine which node has the minimum cost and thus which node is evaluated. Therefore both algorithms open the same node each iteration and produce the same $OPEN$ list and thus the same optimal path.

Answer to problem 3 (3/3)