

1 Lösungsidee

Die Lösungsidee dreht sich darum, den Dijkstra Algorithmus so umzubauen, dass er nach dem Weg mit dem wenigsten Abbiegen statt geringster Länge sucht.

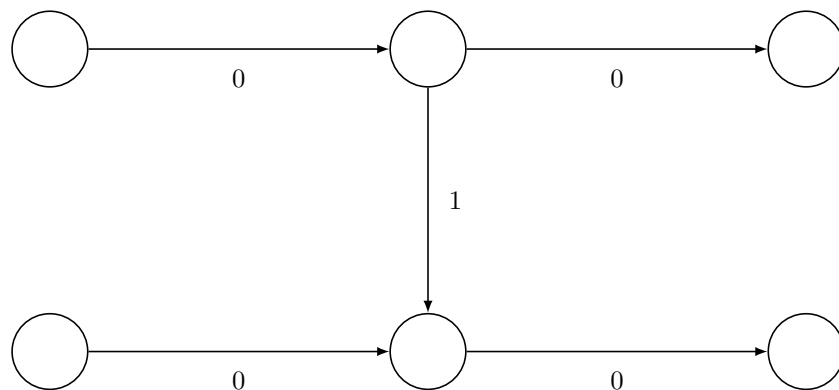
Hierfür müssen die Eingabedaten zuerst in einen für Dijkstra geeigneten Graphen konvertiert werden, also müssen die Kreuzungen ermittelt werden. Hierfür kann der Sweep-Line bzw. Bentley–Ottmann Algorithmus verwendet werden, mit einer durchschnittlichen Komplexität von $n \log n$. Mehr dazu später. Sind die Kreuzungen gefunden, kann der Dijkstra-Algorithmus verwendet werden. Zuerst wird der kürzeste Weg berechnet, woraufhin dessen Länge und die Anzahl an Abbiegungen darin ermittelt.

Daraufhin beginnt die Ermittlung des endgültigen Wegs. Grundsätzlich verläuft diese Ermittlung wie der normale Dijkstra Algorithmus, mit ein paar kleinen Unterschieden. Der erste Unterschied ist, dass der Algorithmus alle Wege mit einer Länge größer der Länge des kürzesten Wegs plus die erlaubte Extrastrecke (also +15% oder +30%) eliminiert werden. Der zweite Unterschied ist, dass alle Wege mit genau so vielen oder mehr Abbiegungen als der kürzeste Weg ebenfalls eliminiert werden. Der dritte und wichtigste Unterschied ist, dass Dijkstra hierbei nicht nach Länge, sondern nach Anzahl an Abbiegungen optimiert.

1.1 Modifizierter Dijkstra im Detail

Um Dijkstra für eine Minimierung von Abbiegungen umzubauen, könnten die Kosten von Kanten des Graphs, anstatt Länge, Existenz von Abbiegungen beschreiben, mit entweder einer 1 für eine Abbiegung oder 0 für eine Gerade. Dabei fällt auf, dass um die Kosten einer Kante des Graphs zu berechnen, die im Weg vorherkommende Kante betrachtet werden muss. Das bedeutet, dass Dijkstra hier nicht in der Form verwendet werden kann, da Dijkstra manchmal Knoten des Graphs von einem anderen Weg aus erneut besucht, wobei sich die Kosten der Kanten aus dem Knoten heraus verändern würden.

Um dieses Problem zu umgehen wird der Graph verändert. Kreuzungen werden im Graphen immernoch als Knoten dargestellt, jedoch gibt es für jede Kreuzung mehrere Knoten, einen für jede geradlinige Straße die in der Kreuzung vorhanden ist. Alle Knoten einer Kreuzung sind dabei verbunden durch Kanten mit Kosten von 1.



1.2 Sweep-Line / Bentley–Ottmann Algorithmus

Der