LEHRBUCH

Thomas Ottmann
Peter Widmayer

Algorithmen und Datenstrukturen

6. Auflage

Springer Vieweg

wollen auch zulassen, dass Kanten/Pfeile eine negative Länge haben. Dann können wir Gewinne und Verluste modellieren, aber auch längste Wege durch kürzeste Wege ausdrücken, nämlich mit negativ gemachten Längen der einzelnen Kanten/Pfeile.

Ein ungerichteter Graph $G = (V,E)$ mit einer reellwertigen Bewertungsfunktion $c : E \to \mathbb{R}$ (englisch: cost) heißt *bewerteter Graph*. Für eine Kante $e \in E$ heißt $c(e)$ *Bewertung* (*Länge, Gewicht, Kosten*) der Kante e. Die Länge $c(G)$ des Graphen G ist die Summe der Längen aller Kanten, also $c(G) = \sum_{e \in E} c(e)$. Damit ist für einen Weg $p = (v_0, v_1, \ldots, v_k)$ die Länge dieses Wegs gerade $c(p) = \sum_{i=0}^{k-1} c((v_i, v_{i+1}))$. Für Graphen ohne Bewertung, die wir bisher betrachtet haben, haben wir die Länge von Wegen so definiert, als sei $c \equiv 1$. Die *Entfernung d* (Distanz; englisch: *distance*) von einem Knoten v zu einem Knoten v' ist definiert als $d(v,v') = min\{c(p) \mid p$ ist Weg von v nach $v'\}$, falls es überhaupt einen Weg von v nach v' gibt; sonst ist $d(v,v') = \infty$. Ein Weg p zwischen v und v' mit $c(p) = d(v,v')$ heißt *kürzester Weg* (englisch: *shortest path*) zwischen v und v'; wir bezeichnen ihn mit $sp(v,v')$.

Ganz entsprechend heißt ein Digraph $G = (V,E)$ mit Bewertungsfunktion $c : E \to \mathbb{R}$ *bewerteter Digraph*; wenn er keine Knoten ohne inzidente Pfeile hat, heißt er *Netzwerk*. Die übrigen Begriffe sind entsprechend definiert.

Ist die Länge jeder Kante nicht negativ, also $c : E \to \mathbb{R}_0^+$, so heißt $G = (V,E)$ mit c *Distanzgraph* (in Abschnitt 6.1.1 haben wir Distanzgraphen betrachtet und die Kosten einer Kante entsprechend als Länge bezeichnet). Die Berechnung kürzester Wege in Distanzgraphen ist einfacher und kann schneller ausgeführt werden als in beliebigen bewerteten Graphen, weil sich in Distanzgraphen Wege durch Hinzunahme weiterer Kanten nicht verkürzen können. Algorithmen für das Finden kürzester Wege zwischen gegebenen Knoten in ungerichteten Distanzgraphen operieren nach dem Grundmuster der Breitensuche. Als Folge davon werden beim Berechnen eines kürzesten Weges von einem gegebenen *Anfangsknoten* zu einem gegebenen *Endknoten* auch kürzeste Wege vom Anfangsknoten zu vielen anderen Knoten des Graphen ermittelt. Das Verfahren zur Berechnung eines kürzesten Weges zwischen Anfangs- und Endknoten (*one-to-one shortest path, single pair shortest path*) unterscheidet sich vom Verfahren zur Berechnung der kürzesten Wege von einem Anfangsknoten zu allen anderen Knoten des Graphen (*one-to-all shortest paths, single source shortest paths*) nur durch das Abbruchkriterium. Im schlimmsten Fall haben beide Verfahren dieselbe Laufzeit; wir werden daher im Folgenden das Problem kürzester Wege von einem zu allen anderen Knoten zunächst für Distanzgraphen und dann für beliebige bewertete Graphen betrachten. Dem Problem, zu jedem Paar von Knoten einen kürzesten Weg zu finden, werden wir uns am Schluss dieses Abschnitts zuwenden.

9.5.1 Kürzeste Wege in Distanzgraphen

Wir betrachten das Problem zu einem gegebenen Distanzgraphen $G = (V,E)$ mit $c : E \to \mathbb{R}_0^+$ je einen kürzesten Weg von einem gegebenen Anfangsknoten s (englisch: source) zu jedem anderen Knoten des Graphen zu finden. Abbildung 9.18 zeigt ein Beispiel für einen ungerichteten Distanzgraphen; neben jeder Kante ist deren Länge