MATHE-BRIEF

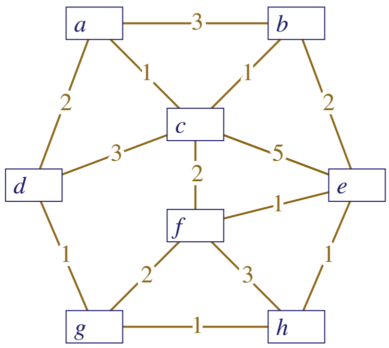
Juni 2010 — Nr. 4

Herausgegeben von der Österreichischen Mathematischen Gesellschaft
http://www.oemg.ac.at/Mathe–Brief ——— mathe–brief@oemg.ac.at

Optimale Wege — der Dijkstra-Algorithmus.

Eine Aufgabe der Praxis der Routenplanung, Telekommunikation, Internet-Organisation usw., die nach mathematischer Behandlung ruft, ist die folgende: Eine Anzahl von Örtlichkeiten ist durch Wege miteinander verbunden. Eine Benützung jedes dieser Wege ist aber mit einem bestimmten Aufwand an Zeit oder Kosten verbunden (wir werden kurz *Kosten* dazu sagen). Wie muss die Route geplant werden, auf der ich mit dem geringsten Kostenaufwand von meinem Startort zu meinem Zielort gelange?

Das passende mathmatische Modell für diese Situation ist ein Graph, d.h. eine Konfiguration, die jede Örtlichkeit durch einen Punkt (*Knoten*) darstellt, und jeden Verbindungsweg durch eine Linie (*Kante*), die die entsprechenden Punkte verbindet — sie braucht nicht gerade zu sein.[1] Ausserdem ist zu jeder Kante ein Zahlenwert notiert, der die Kosten für die Wegbenützung angibt. In Figur 1 ist ein Beispiel für einen solchen Graphen angegeben, wobei die Knoten mit den Buchstaben a bis h bezeichnet sind. Weil wir später zu jedem Knoten noch die Kosten der optimalen Verbindung mit dem Start dazuschreiben werden, ist jeder Knoten durch ein Rechteck markiert, in dem noch Platz freigelassen ist. Wie sollen wir vorgehen, Wenn wir mit dem geringsten Kostenaufwand von a nach h kommen wollen?

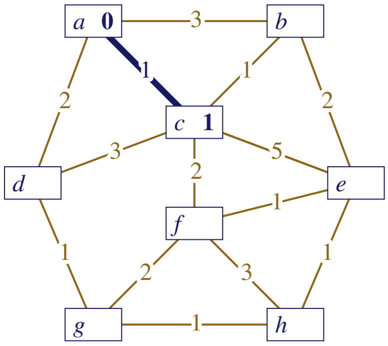


Figur 1

Die Antwort gibt ein Algorithmus, der von dem holländischen Mathematiker Edsger Dijkstra[2] (1930-2002) entwickelt wurde, und der von verblüffender Einfachheit ist. Beim Knoten a können wir gleich 0 eintragen, weil ja noch keine Kosten entstanden sind, wenn wir dort starten. Von a gehen drei Kanten aus, die wir mit ab (3), ac (1) und ad (2) bezeichnen können. In Klammern haben wir immer die entsprechenden Wegkosten für diese Kanten notiert. Die Kante mit den geringsten Kosten, nämlich 1, führt zum Knoten c. Wir tragen diesen Wert 1 beim Knoten c ein (Figur 2) und markieren die Knoten a, c sowie die Kante ac.

[1] Natürlich müssen wir verlangen, dass je zwei Punkte durch einen Weg von aufeinanderfolgenden Kanten verbunden werden können, d.h. dass der Graph zusammenhängend ist.

[2] ausgesprochen „eds-cher daikstra“



Figur 2

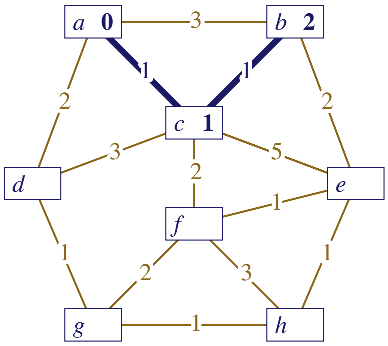
Von den beiden bisher markierten Knoten a und c gehen sechs weiterführende Kanten aus, also solche, deren Endpunkte noch nicht markiert wurden:

$$ab\,(3),\quad ad\,(2),\quad cb\,(1),\quad cd\,(3),\quad ce\,(5)\quad \text{und}\quad cf\,(2).$$

Addieren wir jedesmal die Kosten der Kante zu den eingetragenen Kosten des Anfangspunktes, so ergeben sich die Werte

$$ab\,[3],\quad ad\,[2],\quad cb\,[2],\quad cd\,[4],\quad ce\,[6]\quad \text{und}\quad cf\,[3].$$

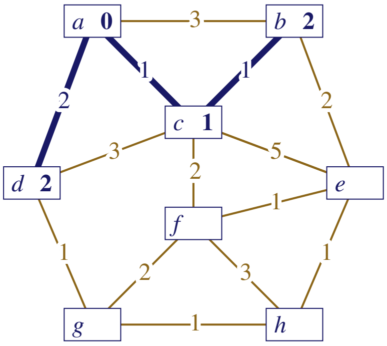
Die kleinste Summe, nämlich 2, ergibt sich für die Kanten ad und cb. Im dem Fall, dass mehrere Kanten dieselbe Summe ergeben so wie hier, wählen wir zufällig eine aus, nämlich cb.[3] Hier wird also beim Knoten b der Wert 2 eingetragen und die Kante cb markiert. 2 sind sicher die minimalen Gesamtkosten für einen von a nach b führenden Weg, da ein solcher Weg rückwärts entweder direkt nach a oder über c nach a führen muss. Jeder andere Weg würde über einen Knoten führen, der den Knoten a mit mehr Kosten erreicht, als der Weg über c.



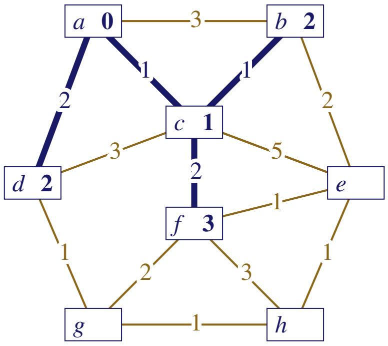
Figur 3

Der Dijkstra-Algorithmus (er heißt auch kürzeste-Wege-Algorithmus) sieht nun den folgenden wesentlichen Schritt vor. Angenommen, wir haben bereits eine Anzahl von Knoten markiert und wissen, auf welchem Weg wir diese mit minimalen Kosten von a aus erreichen können. Dann schauen wir uns alle Kanten an, die von diesen Knoten aus *weiter* (zu einem noch nicht markierten Knoten) führen, addieren die Kosten jeder Kante zu den Kosten ihres jeweiligen Anfangsknotens und markieren den Endknoten mit der minimalen Kostensumme. Ist die Wahl nicht eindeutig, wenden wir die obige Regel an. Wir können dabei sicher sein, dass die eingetragene Kostensumme und der markierte Weg zu diesem Knoten optimal sind. (Figuren 4, 5 und 6).

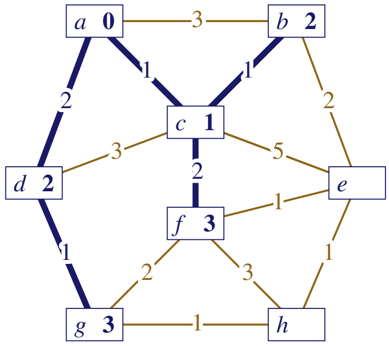
[3]Wir verwenden die folgende Regel: Wir wählen denjenigen Endpunkt aus, der im Alphabet früher kommt, und bei gleichen Endpunkten markieren wir die Kante mit dem Anfangspunkt, der im Alphabet früher kommt.



Figur 4

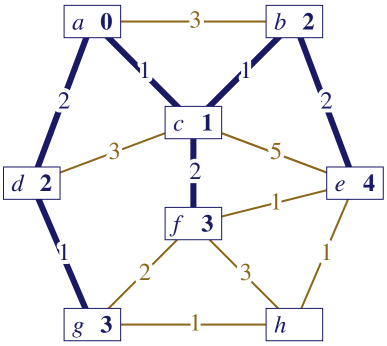


Figur 5

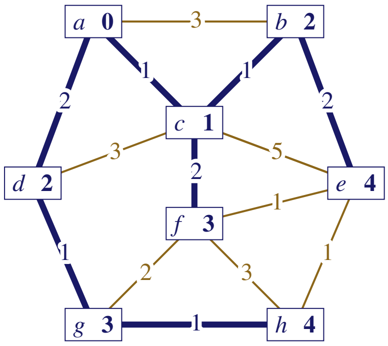


Figur 6

Wir führen diese Schritte durch, bis wir beim Knoten *h* angelangt sind. In unserem Beispiel ist *adgh* der optimale Weg, er hat den Kostenwert 4 und wird in sieben Schritten bestimmt (Figuren 7 und 8).



Figur 7



Figur 8

Ein paar charakteristische Züge des Dijkstra-Algorithmus seien zum Schluss noch einmal festgehalten:

- Bei jedem Schritt des Algorithmus werden ein weiterer Knoten und eine weitere Kante markiert.
- Die Kostenwerte der so erzeugten Verbindungswege mit dem Startknoten bilden dabei eine monoton aufsteigende Folge (die streckenweise auch konstant bleiben kann).
- Fortgesetzt kann die Prozedur werden, solange noch ein Knoten nicht markiert ist.
- Wenn alle Knoten markiert sind, wissen wir für jeden Knoten des Graphen, auf welchem Wege und mit welchen Kosten er von *a* aus erreicht werden kann.

Bei echten Problemstellungen in der Praxis sind die auftretenden Graphen wesentlich größer, und es muss die Art der Datenspeicherung, der Suchprozeduren und der Prozesssteuerung zweckmäßig gewählt werden. Die wesentlichen Schritte des Dijkstra-Algorithmus sind aber dieselben wie in unserem Beispiel.

Literatur: W. D. WALLIS.: *A Beginners' Guide to Graph Theory*. Birkhäuser Verlag, Boston 2000.