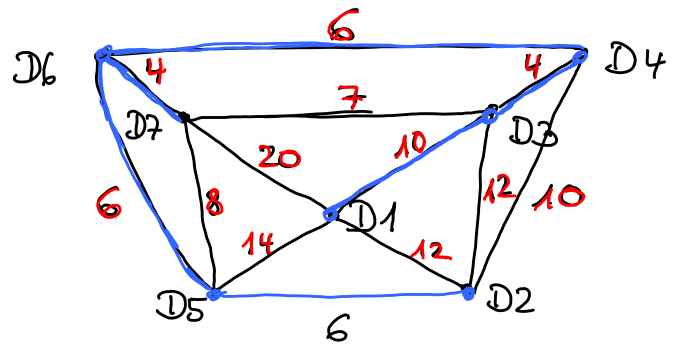
Vorlesung Mathematik 2

17.7.2017

Bp. für den Algorithmus von Kruskal

Sieben Dörfer D1 – D7

	D1	D2	D3	D4	D5	D6	D7
D1	0	12	10	0	14	0	20
D2	12	0	12	10	6	0	0
D3	10	12	0	4	0	0	7
D4	0	10	4	0	0	6	0
D5	14	6	0	0	0	6	8
D6	0	0	0	6	6	0	4
D7	20	0	7	0	8	4	0



Minimalen Kosten: $6+4+6+4+6+10 = 36$

Bestimmung von kürzesten Wegen in Graphen

von einem Knoten zu allen andern Knoten

"Single source shortest paths"

kürzeste Wege zwischen je zwei Knoten

" all pairs shortest paths"

Gegeben ist $G(M, K, v, d)$ — d ist die Bewertungsfunktion

$d(k) \geq 0$ für alle $k \in K$

Algorithmus von Dijkstra

(1) Markiere den Startknoten x_0 mit

$$D(x_0) = 0$$

Setze $i = 1$ $\quad M_1 := \{x_0\}$

weiter mit (2)

(2) Sei M^* die Menge aller Knoten x_s der Kanten (x_r, x_s), deren Knoten x_r in M_i liegt, d.h.

$$M^* = \{x_s \in M \setminus M_i \mid \text{es gibt eine Kante } (x_r, x_s) \text{ mit } x_r \in M_i\}$$

Falls $M^* = \emptyset$, weiter mit (6)

Falls $M^* \neq \emptyset$, weiter mit (3)

(3) Bestimme zu jedem $x_s \in M^*$ eine vorläufige Markierung $D^*(x_s) = \min \left(D(x_r) + d(x_r, s)\right)$

weiter mit (4)

(4) Markiere den Knoten $x_s^* := x_i$ endgültig mit $D(x_i) = D^*(x_s^* := \min D^*(x_s))$

Markiere Kante (x_r, x_i)

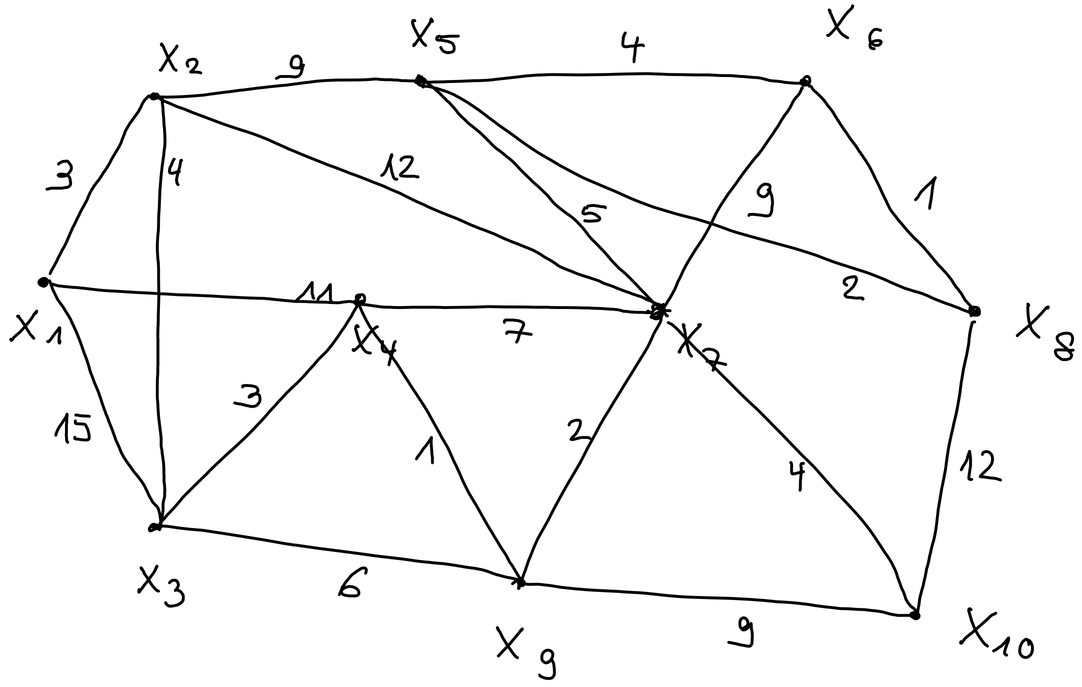
weiter mit (5)

(5) Setze $M_{i+1} := M_i \cup \{x_i\}$ und $i := i+1$

Weiter mit (2)

(6) Ende

Bp:



Aufgabe: Es sollen die kürzesten Wege von x_7 zu allen anderen Knoten bestimmt werden!

Algorithmus von Dijkstra

Schritt 1: $D(x_7) = 0 \quad M_1 := \{x_7\}$

Iteration 1

$M^* = \{x_2, x_4, x_5, x_6, x_9, x_{10}\}$

die Menge der zu x_7 benachbarten Knoten

Ermitteln der Entfernungen $D(x_s)$, $x_s \in M^*$

x_s	x_2	x_4	x_5	x_6	x_9	x_{10}
$D^*(x_s)$	12	7	5	9	2	4

$D(x_s) = d(x_7, x_s)$

$D^*(x_s) = \min\left(D(x_r) + d(x_r, x_s)\right) = 2$

Damit $D(x_9) = 2$

Markiere die Kante (x_7, x_9)

Iteration 2

$M^* := \{x_2, x_3, x_4, x_5, x_6, x_{10}\}$

Ermittlung der Entfernungen $D(x_s)$, $x_s \in M^*$ direkt von x_7 oder über x_9

x_s	x_2	x_3	x_4	x_5	x_6	x_{10}
$D^*(x_s)$	12	8	3	5	9	4
		(2+6)	min(7, 2+1)			

Markiere Kante (x_9, x_4)

$$M_3 := M_2 \cup \{x_4\} = \{x_7, x_9, x_4\}$$

$$D(x_4) = 3$$

Iteration 3

$$M^* = \{x_1, x_2, x_3, x_5, x_6, x_{10}\}$$

alle zu x_4, x_7 und x_9 benachbarte Knoten

x_s	x_1	x_2	x_3	x_5	x_6	x_{10}
$D^*(x_s)$	14	12	6	5	9	4

$$D(x_{10}) = 4$$

Markiere die Kante (x_7, x_{10})

$$M_4 = M_3 \cup \{x_{10}\} = \{x_7, x_9, x_4, x_{10}\}$$

Iteration 4

$$M^* = \{x_1, x_2, x_3, x_5, x_6, x_8\}$$

x_s	x_1	x_2	x_3	x_5	x_6	x_8
$D^*(x_s)$	14	12	6	5	9	16

$$D(x_5) = 5$$

$$M_5 = \{x_7, x_9, x_4, x_{10}, x_5\}$$

Iteration 5

$M^* = \{x_1, x_2, x_3, x_6, x_8\}$

x_s	x_1	x_2	x_3	x_6	x_8
$D^*(x_s)$	14	12	6	9	7

$D(x_3) = 6$

Markiere (x_4, x_3)

Iteration 6 $\quad M^* = \{x_1, x_2, x_6, x_8\}$

x_s	x_1	x_2	x_6	x_8
$D^*(x_s)$	14	10	9	7

$D(x_8) = 7$ Markiere (x_5, x_8)

Iteration 7 $\quad M^* = \{x_1, x_2, x_6\}$

x_s	x_1	x_2	x_6
$D^*(x_s)$	14	10	8

$D(x_6) = 8$ Markiere (x_6, x_8)

Iteration 8 $\quad M^* = \{x_1, x_2\}$

x_s	x_1	x_2
$D^*(x_s)$	14	10

$D(x_2) = 10$ Markiere (x_3, x_2)

Iteration 9 $\quad M^* = \{x_1\}$

x_s	x_1
$D^*(x_s)$	13

Markiere (x_2, x_1)

Kürzeste Wege von x_7 zu allen anderen Knoten

$D(x_1) = 13$
$D(x_2) = 10$
$D(x_3) = 6$
$D(x_4) = 3$
$D(x_5) = 5$
$D(x_6) = 8$
$D(x_7) = 0$
$D(x_8) = 7$
$D(x_9) = 2$
$D(x_{10}) = 4$

Flüsse und Schnitte in Netzwerken

Def: Netzwerk $N = (\underset{\text{Knoten}}{M}, \underset{\text{Kanten}}{K}, \underset{\text{Quelle}}{s}, \underset{\text{Senke}}{t}, \underset{\text{Kapazitätsfunktion}}{c})$

auch keine Mehrfachkanten!

Def: Quelle

Knoten x_i in einem gerichteten Graphen, der nur Anfangsknoten von "Pfeilen" ist.

Def: Senke

Knoten x_i, der nur Endknoten von Pfeilen ist.

Def: s-t-Fluss

Belegung der Kanten mit nicht neg. reellen Zahlen mit folgenden Bedingungen

1) Keine Kante darf mit einem höheren Wert als ihre Kapazität belegt sein

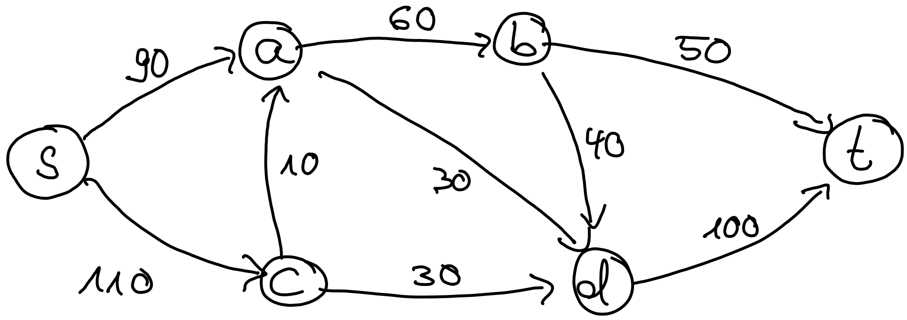
2) In jedem Knoten (außer Quelle und Senke) muss genausoviel hinein- wie hinausfließen
"Flusserhaltung"

Def: Wert eines s-t-Flusses

Def: Schnitt

Algorithmus von Ford u. Fulkerson
findet in einem Netzwerk einen maximalen Fluss

Bp: Paketzustelldienst



Ziel: tägliche mögliche maximale Zahl von Paketen von s nach t