Diskrete Mathematik

Univ.-Prof. Dr. Goulnara ARZHANTSEVA

SS 2018

Graphen

Definition: Einfacher Graph $\mathbf{G} = \mathbf{G}(V, E) = (V(\mathbf{G}), E(\mathbf{G})) = (V, E)$

Ein einfacher Graph **G** besteht aus einer (endlichen) Menge V von Knoten (Vertices) und einer Teilmenge $E \subseteq \binom{V}{2}$ von Kanten (Edges).

Notation: $\binom{V}{2} := \{A \subseteq V : |A| = 2\}$ 2–elementigen Teilmengen von V.

Ein einfacher Graph kann nicht Schlingen und mehrfache Kanten haben.

Graphen

Definition: Einfacher Graph $\mathbf{G} = \mathbf{G}(V, E) = (V(\mathbf{G}), E(\mathbf{G})) = (V, E)$

Ein einfacher Graph **G** besteht aus einer (endlichen) Menge V von Knoten (Vertices) und einer Teilmenge $E \subseteq \binom{V}{2}$ von Kanten (Edges).

Notation: $\binom{V}{2} := \{A \subseteq V : |A| = 2\}$ 2–elementigen Teilmengen von V.

Ein einfacher Graph kann nicht Schlingen und mehrfache Kanten haben.

Definition: Graph $\mathbf{G} = \mathbf{G}(V, E) = (V(\mathbf{G}), E(\mathbf{G})) = (V, E)$

Ein Graph **G** besteht aus einer (endlichen) Menge V von Knoten (Vertices) und eine (endliche) Multimenge E der Familie der 2–elementigen Multimengen von V.
E ist eine (endliche) Multimenge von Kanten (Edges).

Ein Graph kann Schlingen und mehrfache Kanten haben.

Graphen: Weg und Kreis

Definition: Weg und Kreis

Sei **G** ein Graph. Eine Wanderung $v_0, v_1, \ldots, v_n$ in G mit der Eigenschaft, daß

- alle Knoten (außer eventuell der erste und der letzte) verschieden sind (d.h.: für $0 \leq i < j \leq n,\ (i,j) \neq (0,n)$ gilt $v_i \neq v_j$),
- und alle Kanten $\{v_{i-1}, v_i\}$ verschieden sind,

nennen wir

- einen Weg von v_0 nach v_n, wenn $v_0 \neq v_n$,
- einen Kreis, wenn $v_0 = v_n$.

Graphen: Weg und Wanderung

Proposition

Sei G ein Graph. Zwei Knoten v, u in G sind genau dann durch einen Weg verbunden, wenn sie durch eine Wanderung verbunden sind.

Beweis:

Jeder Weg ist eine Wanderung, daher ist die eine Richtung klar. Umgekehrt überlegen wir, daß jede Wanderung $v = v_0, v_1, \ldots, v_n = u$ zu einem Weg "verkürzt" werden kann: Solange es einen Knoten w in der Wanderung gibt, der mehrfach vorkommt, schneiden wir das Stück zwischen dem ersten und letzten Vorkommnis von w heraus:

$$\left(v_0, \ldots, v_k, \underbrace{w, \ldots, w}_{\text{ausschneiden!}}, v_m, \ldots, v_n \right) \to (v_0, \ldots, v_k, w, v_m, \ldots, v_n) .$$

Wenn es keinen solchen Knoten (mehr) gibt, haben wir einen Weg vor uns, der v mit u verbindet. ■

Graphen: Adjazenzmatrix

Definition: Adjazenzmatrix

Sei $\mathbf{G}(V, E)$ ein Graph. Die Adjazenzmatrix von $\mathbf{G}$ ist eine $|V| \times |V|$ Matrix $A(\mathbf{G})$, deren Zeilen und Spalten wir uns durch die Knoten aus V bezeichnet denken, und deren Eintrag in Zeile v_i, Spalte v_j gleich der Vielfachheit von $\{v_i, v_j\}$ in E ist.

Graphen: Adjazenzmatrix

Definition: Adjazenzmatrix

Sei $\mathbf{G}(V, E)$ ein Graph. Die Adjazenzmatrix von $\mathbf{G}$ ist eine $|V| \times |V|$ Matrix $A(\mathbf{G})$, deren Zeilen und Spalten wir uns durch die Knoten aus V bezeichnet denken, und deren Eintrag in Zeile v_i, Spalte v_j gleich der Vielfachheit von $\{v_i, v_j\}$ in E ist.

Bemerkung: $A(\mathbf{G})$ ist eine reelle symmetrische Matrix und daher diagonalisierbar: Die Untersuchung ihrer Eigenwerte kann interessante Aussagen über den zugrundeliegenden Graphen liefern.

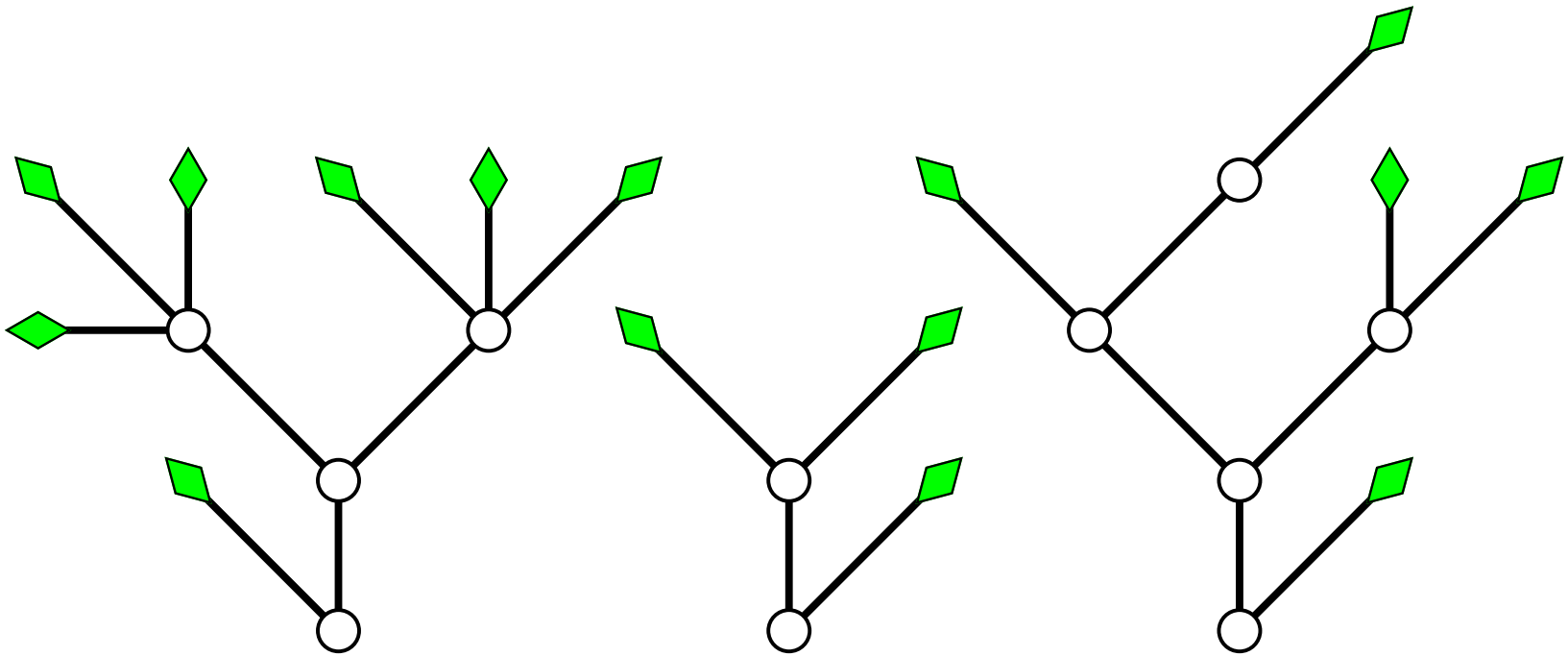
Graphen: Wälder und Bäume

Definition: Wald und Baum

Ein Wald ist ein Graph, der keinen Kreis enthält.
Ein Baum ist ein zusammenhängender Wald.
Ein Knoten vom Grad 1 in einem Wald heißt ein Blatt.

Bemerkung: Die Zusammenhangskomponenten eines Waldes sind Bäume.

Beispiel: ein Wald mit drei Bäumen (aus Skriptum)



Die Blätter sind hier als Rhomben gezeichnet.

Graphen: Bäume

Lemma 1: Ein Blatt

Ein Baum **T** mit $n > 1$ Knoten hat (mindestens) ein Blatt.

Beweis: Angenommen, es gäbe keinen Knoten vom Grad 1. Da ein Baum zusammenhängend ist, gibt es (außer für $n = 1$) keinen isolierten Knoten (das ist ein Knoten vom Grad 0, der eine eigene Zusammenhangskomponente darstellt). Daher müßte also jeder Knoten Grad mindestens 2 haben. Dies wiederum würde bedeuten, daß es beliebig lange Wanderungen in **T** gibt, bei denen niemals eine Kante zweimal unmittelbar hintereinander durchlaufen wird.

Graphen: Bäume

Es würde also eine solche Wanderung w geben, deren Länge größer ist als n: Darin muß es dann aber notwendigerweise (mindestens) einen Knoten geben, der (mindestens) zweimal vorkommt. Ein minimaler Abschnitt in w[1] von einem solchen Knoten v zu dem nächsten Auftreten von v in der Wanderung

$$\Big(\ldots, \underbrace{v, \ldots, v}_{\text{geschlossener Weg = Kreis!}}, \ldots \Big)$$

würde einen Kreis bilden (denn der Abschnitt kann nicht vom Typ (v, x, v) sein, da niemals eine Kante unmittelbar hintereinander zweimal durchlaufen wird); ein Widerspruch. ■

[1] Genauer: Sei $w = (v_1, v_2, \ldots, v_n)$. Ordne die Menge aller Paare (i, j) mit $1 < i < j < n$ und $v_i = v_j$, durch $(i, j) \leq (k, l) :\Leftrightarrow i \geq k$ und $j \leq l$ ("Intervall–Inklusion"): In dieser teilweise geordneten Menge gibt es minimale Elemente.

Graphen

Ein zusammenhängender Graph hat “relativ viele Kanten”, ein Graph ohne Kreise hat “relativ wenige Kanten”:

Für einen Baum mit n Knoten ist die Anzahl seiner Kanten eindeutig festgelegt, wie das nächste Resultat zeigt.

Graphen: Baum

Lemma 2: $n - 1$ Kanten

Ein Baum **T** mit genau n Knoten hat genau $n - 1$ Kanten.

Beweis:

Wir zeigen die Behauptung mit Induktion. Die Sache ist klar für $n = 1$ (denn die einzigen möglichen Kanten in diesem Fall wären Schlingen, die bilden aber Kreise).

Graphen: Baum

Lemma 2: $n - 1$ Kanten

Ein Baum **T** mit genau n Knoten hat genau $n - 1$ Kanten.

Beweis:
Wir zeigen die Behauptung mit Induktion. Die Sache ist klar für $n = 1$ (denn die einzigen möglichen Kanten in diesem Fall wären Schlingen, die bilden aber Kreise).
Für den Induktionsschritt von $n - 1$ auf n wählen wir ein Blatt, also einen Knoten v vom Grad 1 in **T** (den es nach Lemma 1 geben muß) und betrachten den Graphen $\mathbf{T} - \mathbf{v}$, der aus **T** entsteht, wenn wir den Knoten v zusammen mit der (einzigen) inzidenten Kante entfernen.

Graphen: Baum

Lemma 2: $n-1$ Kanten

Ein Baum **T** mit genau n Knoten hat genau $n-1$ Kanten.

Beweis:
Wir zeigen die Behauptung mit Induktion. Die Sache ist klar für $n = 1$ (denn die einzigen möglichen Kanten in diesem Fall wären Schlingen, die bilden aber Kreise).
Für den Induktionsschritt von $n-1$ auf n wählen wir ein Blatt, also einen Knoten v vom Grad 1 in **T** (den es nach Lemma 1 geben muß) und betrachten den Graphen $\mathbf{T}-\mathbf{v}$, der aus **T** entsteht, wenn wir den Knoten v zusammen mit der (einzigen) inzidenten Kante entfernen. $\mathbf{T}-\mathbf{v}$ hat $n-1$ Knoten, enthält keinen Kreis und ist zusammenhängend (denn kein Weg in **T**, der zwei Knoten $x, y \neq v$ verbindet, kann den Knoten v enthalten): $\mathbf{T}-\mathbf{v}$ ist also ein Baum und hat nach Induktionsvoraussetzung $n-2$ Kanten, **T** hat also $n-1$ Kanten. ■

Graphen: Baum

Korollar 1: Zwei Blätter

Ein Baum $\mathbf{T}$ mit $n > 1$ Knoten hat (mindestens) zwei Blätter.

Beweis: Wir haben (Leonhard Euler'1736):

$$\sum_{v \in V(\mathbf{T})} \deg(v) = 2 \cdot |E(\mathbf{T})| = 2 \cdot (n-1).$$

Angenommen, es gäbe nur ein Blatt b, dann wäre

$$\sum_{v \in V(\mathbf{T})} \deg(v) = 1 + \sum_{v \in V(\mathbf{T}) \setminus \{b\}} \underbrace{\deg(v)}_{\geq 2} \geq 1 + 2 \cdot (n-1),$$

ein Widerspruch. ■

Graphen: Baum

Korollar 2: Wald und Kanten

Ein Wald mit n Knoten und m Zusammenhangskomponenten hat $n - m$ Kanten; insbesondere hat er also höchstens $n - 1$ Kanten (mit Gleichheit dann und nur dann, wenn er ein Baum ist).

Beweis:

Für einen Wald $\mathbf{F}$ mit n Knoten und mit m Komponenten $\mathbf{T}_1, \ldots, \mathbf{T}_m$, die jeweils $a_1, \ldots, a_m$ Knoten enthalten, gilt $\sum_{i=1}^{m} a_i = n$. Jede Komponente $\mathbf{T}_i$ ist ein Baum und hat nach Lemma 2 also $a_i - 1$ Kanten. Also hat $\mathbf{F}$

$$\sum_{i=1}^{m} (a_i - 1) = n - m$$

Kanten. ■

Graphen: Baum

Korollar 3: Zusammenhängender Graph und Kanten

Ein zusammenhängender Graph mit n Knoten hat mindestens $n - 1$ Kanten; er hat genau $n - 1$ Kanten dann und nur dann, wenn er ein Baum ist.

Beweis: Wir betrachten einen beliebigen zusammenhängenden Graphen $\mathbf{G}$, der kein Baum ist. Dann gibt es also einen Kreis C in $\mathbf{G}$. Sei e eine Kante in C, und sei $\mathbf{G} - \mathbf{e}$ der Graph, der aus $\mathbf{G}$ durch das Entfernen von e entsteht. $\mathbf{G} - \mathbf{e}$ ist noch immer zusammenhängend (denn in jeder Wanderung in $\mathbf{G}$, die die Kante e benutzt, kann e durch den Weg $C - e$ ersetzt werden).

Graphen: Baum

Korollar 3: Zusammenhängender Graph und Kanten

Ein zusammenhängender Graph mit n Knoten hat mindestens $n-1$ Kanten; er hat genau $n-1$ Kanten dann und nur dann, wenn er ein Baum ist.

Beweis: Wir betrachten einen beliebigen zusammenhängenden Graphen $\mathbf{G}$, der kein Baum ist. Dann gibt es also einen Kreis C in $\mathbf{G}$. Sei e eine Kante in C, und sei $\mathbf{G}-\mathbf{e}$ der Graph, der aus $\mathbf{G}$ durch das Entfernen von e entsteht. $\mathbf{G}-\mathbf{e}$ ist noch immer zusammenhängend (denn in jeder Wanderung in $\mathbf{G}$, die die Kante e benutzt, kann e durch den Weg $C-e$ ersetzt werden).

Dieses "Zerstören von Kreisen" können wir solange wiederholen, bis ein Baum entstanden ist: Angenommen, wir brauchen dafür r Schritte, dann enthält der ursprüngliche Graph $\mathbf{G}$ also $n-1+r$ Kanten, $r > 0$. ■

Teilgraph

Definition: Teilgraph

Sei $\mathbf{G}(V, E)$ ein Graph. Eine Teilmenge $V_H \subseteq V$ zusammen mit einer Teilmenge $E_H \subseteq E$ definiert einen Teilgraphen $\mathbf{H} = \mathbf{H}(V_H, E_H)$, wenn alle Knoten, die zu Kanten aus E_H gehören, in V_H enthalten sind; also wenn $\left(\bigcup_{e \in E_H} e\right) \subseteq V_H$.

Sei $e \in E(\mathbf{G})$: Den Teilgraph $\mathbf{H}(V_H, E_H)$ mit $V_H = V(\mathbf{G})$ und $E_H = E(\mathbf{G}) \setminus \{e\}$ ($\mathbf{H}$ entsteht also aus $\mathbf{G}$ "durch Entfernen der Kante e") bezeichnen wir mit $\mathbf{G} - \mathbf{e}$.

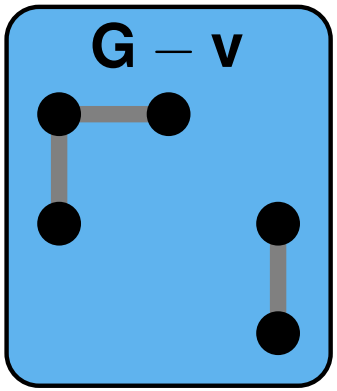
Induzierte Teilgraph

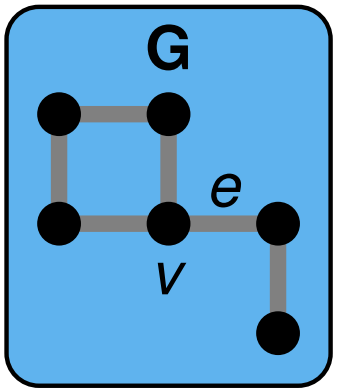
Definition: Induzierte Teilgraph

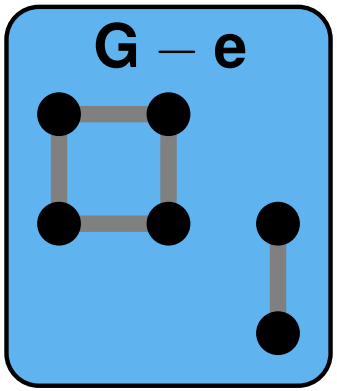
Sei $\mathbf{G}(V, E)$ ein Graph. Eine Teilmenge $V_H \subseteq V$ zusammen mit einer Teilmenge $E_H \subseteq E$ definiert einen (durch V_H) induzierten Teilgraphen $\mathbf{H} = \mathbf{H}(V_H, E_H)$, wenn überdies alle Kanten in $E(\mathbf{G})$, die beide Knoten in V_H haben, auch zu E_H gehören (also $\forall e \in E(\mathbf{G}) : \ e \subseteq V_H \implies e \in E_H$; V_H determiniert dann E_H eindeutig).

Sei $v \in V(\mathbf{G})$: Den induzierten Teilgraph $\mathbf{H}(V_H, E_H)$ mit $V_H = V(\mathbf{G}) \setminus \{v\}$ ($\mathbf{H}$ entsteht also aus $\mathbf{G}$ "durch Entfernen des Knotens v") bezeichnen wir mit $\mathbf{G} - \mathbf{v}$.

Beispiel: Teilgraph und induzierter Teilgraph (aus Skriptum)







induzierter Teilgraph

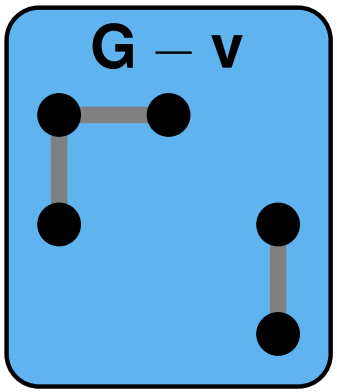
Teilgraph

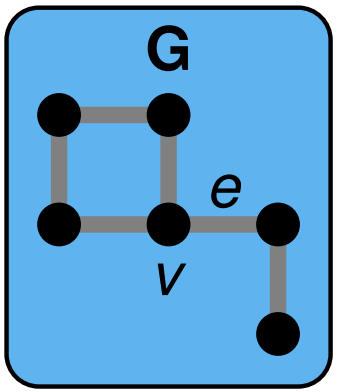
Spannender Teilgraph

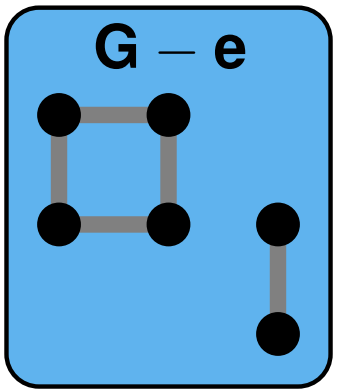
Definition: spannender Teilgraph

Sei $\mathbf{G} = \mathbf{G}(V, E)$ ein Graph. Ein Teilgraph $\mathbf{H} = \mathbf{H}(V_H, E_H)$ von $\mathbf{G}$ heißt spannender Teilgraph, wenn $V_H = V$.
Ein spannender Teilgraph von $\mathbf{G}$, der ein Wald bzw. ein Baum ist, heißt spannender Wald bzw. spannender Baum von $\mathbf{G}$.

Beispiel: Teilgraph und spannender Teilgraph (aus Skriptum)







induzierter Teilgraph

nicht spannender Teilgraph

Teilgraph

spannender Teilgraph

Spannender Baum

Spannbaum = aufspannender Baum = spanning tree

Korollar 4: Spannender Baum

Jeder zusammenhängende Graph hat einen spannenden Baum.

Beweis:
Das folgt an sich sofort aus dem Beweis für Korollar 3. Wir geben hier aber einen zweiten “algorithmischen” Beweis.

Spannender Baum

Sei $\mathbf{G}(V, E)$ ein zusammenhängender Graph.

{Initialisierung:}

1: $S \leftarrow \emptyset$, $\mathbf{T} \leftarrow \mathbf{T}(V, S)$

{Schleife: Wird wiederholt, solange die Bedingung erfüllt ist.}

2: **while** (Bedingung: Der **T** ist nicht zusammenhängend.) **do**

3: Wähle eine beliebige Kante $e \in E$, die Knoten in verschiedenen Zusammenhangskomponenten von **T** verbindet,

4: $S \leftarrow S \cup \{e\}$

5: $\mathbf{T} \leftarrow \mathbf{T}(V, S)$

6: **end while**

Spannender Baum

1: Man findet in jedem Schritt eine geeignete Kante e:
Denn sei **Z** eine Zusammenhangskomponente von **T** und **Y** der "restliche Graph" (also die übrigen Zusammenhangskomponenten).

Seien z bzw. y zwei Knoten aus **Z** bzw. **Y**. In **G** gibt es einen Weg, der z mit y verbindet — der muß eine Kante e enthalten, die "**Z** mit **Y** verbindet" (d.h., einen Knoten aus **Z** und einen aus **Y** enthält). Diese Kante e ist eine geeignete Wahl im Wiederholungsschritt.

Spannender Baum

2: Der Graph **T** enthält keine Kreise:
Denn im Initialisierungsschritt ist das klar. Und durch das Hinzufügen der Kanten e (deren Existenz wir eben gezeigt haben) zu einem Graphen ohne Kreise kann kein Kreis C entstehen, denn dieser müßte die Kante e enthalten — diese verbindet aber nach Konstruktion zwei verschiedene Zusammenhangskomponenten $\mathbf{Z_1}$ und $\mathbf{Z_2}$.

C müßte ja die Kante e “benutzen”, um von $\mathbf{Z_1}$ nach $\mathbf{Z_2}$ zu gelangen — aber für die “Rückkehr” von $\mathbf{Z_2}$ nach $\mathbf{Z_1}$ steht ja wieder nur dieselbe Kante e zur Verfügung. ■

Minimal spannender Baum

Definition: Minimal spannender Baum

Seien $\mathbf{G}(V, E)$ ein zusammenhängender Graph und $\omega : E \to \mathbb{R}^+ \setminus \{0\}$ eine Gewichtsfunktion auf der Menge seiner Kanten.

Ein Spannender Baum $\mathbf{T}$ ist minimal, wenn kein anderer spannender Baum in demselben Graphen mit geringerem Gewicht existiert, d.h.

$$\omega(\mathbf{T}) := \sum_{e \in E(\mathbf{T})} \omega(e)$$

ist minimal.

Kruskals Greedy Algorithmus

Proposition: Kruskals Greedy Algorithmus (Joseph Kruskal'1956)

Sei $\mathbf{G}(V, E)$ ein zusammenhängender Graph mit einer Gewichtsfunktion $\omega : E \to \mathbb{R}^+ \setminus \{0\}$ auf der Menge seiner Kanten. Der folgende "greedy algorithm" liefert einen minimalen spannenden Baum:

{Initialisierung:}

1: $S \leftarrow \emptyset$, $\mathbf{T} \leftarrow \mathbf{T}(V(\mathbf{G}), S)$.

{Schleife: Wird wiederholt, solange die Bedingung erfüllt ist.}

2: **while** (Bedingung: Der $\mathbf{T}$ ist nicht zusammenhängend.) **do**

3: Wähle unter den Kanten in $E(\mathbf{G})$, die Knoten in verschiedenen Zusammenhangskomponenten von $\mathbf{T}$ verbinden, eine von minimalem Gewicht (deshalb heißt das Verfahren "gieriger" Algorithmus), e,

4: $S \leftarrow S \cup \{e\}$

5: $\mathbf{T} \leftarrow \mathbf{T}(V(\mathbf{G}), S)$.

6: **end while**

Kruskals Greedy Algorithmus

Beweis:
Daß der Algorithmus einen spannenden Baum liefert, wissen wir bereits: Zu zeigen ist also, daß das Resultat ein minimaler spannender Baum ist.

Sei n die Anzahl der Knoten von $\mathbf{G}$, und sei $e_1, \ldots, e_{n-1}$ die Folge der Kanten von $\mathbf{T}$ in der Reihenfolge, wie sie vom greedy algorithm gewählt werden. Es gilt:

$$\omega(e_1) \leq \cdots \leq \omega(e_{n-1}).$$

Kruskals Greedy Algorithmus

Angenommen, es gäbe einen spannenden Baum mit Kanten $f_1, \ldots, f_{n-1}$, die auch nach dem Gewicht geordnet seien, also

$$\omega(f_1) \leq \cdots \leq \omega(f_{n-1}),$$

der ein kleineres Gesamtgewicht der Kanten hat, also

$$\sum_{i=1}^{n-1} \omega(f_i) < \sum_{i=1}^{n-1} \omega(e_i).$$

Nun wähle k minimal, sodaß

$$\sum_{i=1}^{k} \omega(f_i) < \sum_{i=1}^{k} \omega(e_i).$$

Es ist sicher $k > 1$, denn der greedy algorithm wählt im ersten Schritt eine Kante von minimalem Gewicht. Es gilt also nach Wahl von k

$$\sum_{i=1}^{k-1} \omega(f_i) \geq \sum_{i=1}^{k-1} \omega(e_i),$$

Kruskals Greedy Algorithmus

daher muß gelten:

$$\omega(f_1) \leq \cdots \leq \omega(f_k) < \omega(e_k).$$

Das wiederum heißt: Der greedy algorithm wählt im k-ten Schritt Kante e_k und keine der Kanten $f_1, \ldots, f_k$, die kleineres Gewicht haben.

Also kann keine der Kanten $f_1, \ldots, f_k$ verschiedene Zusammenhangskomponenten im Graphen

$$\mathbf{T}^e = \mathbf{T}(V, \{e_1, \ldots, e_{k-1}\})$$

verbinden.

Kruskals Greedy Algorithmus

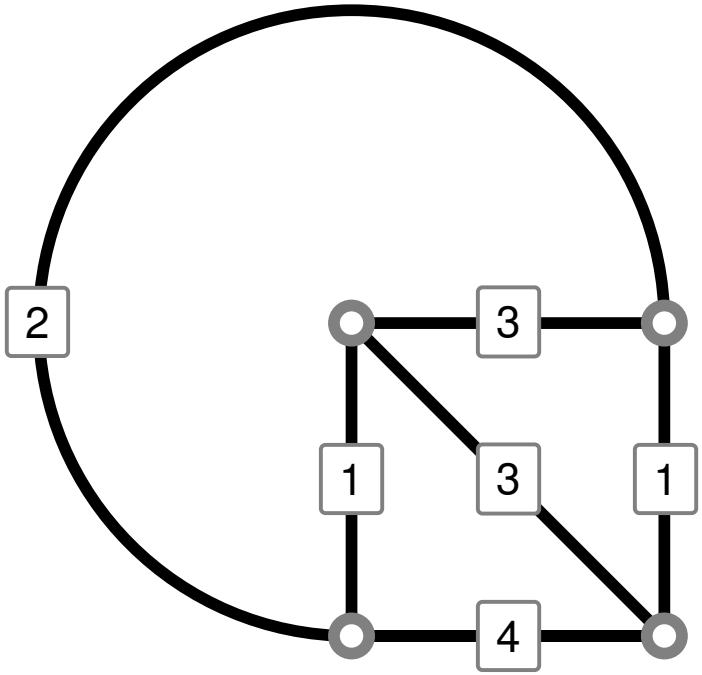
Das bedeutet aber, daß die Knotenmenge jeder Zusammenhangskomponente von

$$\mathbf{T}^f = \mathbf{T}(V, \{f_1, \ldots, f_k\})$$

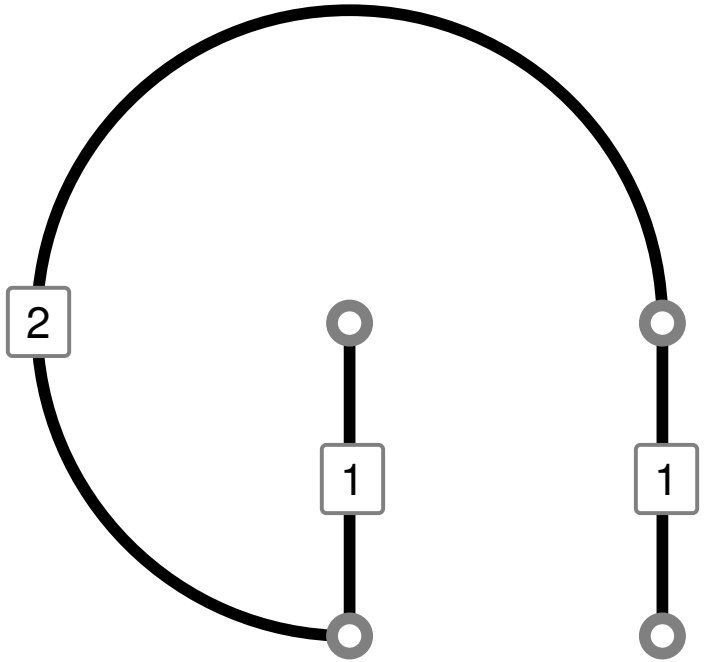
in der Knotenmengen einer Zusammenhangskomponente von $\mathbf{T}^e$ enthalten ist, insbesondere hat also $\mathbf{T}^f$ mindestens so viele Zusammenhangskomponenten wie $\mathbf{T}^e$.

Dies ist aber ein Widerspruch, denn $\mathbf{T}^e$ und $\mathbf{T}^f$ sind Wälder, und die Anzahl ihrer Komponenten ist $n - (k-1) > n - k$. ■

Beispiel: Minimal spannender Baum (aus Skriptum)



Graph mit Gewichtsfunktion



Minimal spannender Baum

Travelling Salesman Problem

Wir müßten eine Rundreise durch n Städte planen und dabei jede Stadt genau einmal aufsuchen. Die Frage ist, wie diese Rundreise am billigsten geschehen kann.

Travelling Salesman Problem

Wir müßten eine Rundreise durch n Städte planen und dabei jede Stadt genau einmal aufsuchen. Die Frage ist, wie diese Rundreise am billigsten geschehen kann.

Definition: Hamiltonscher Kreis

Ein Kreis in einem Graphen **G**, der alle Knoten von **G** enthält, heißt Hamiltonscher Kreis.

Sei $\mathbf{K}_n$ der vollständige Graph mit n Knoten, und sei eine Gewichtsfunktion $\omega : E \to \mathbb{R}$ auf der Menge seiner Kanten gegeben.

In der Familie aller Hamiltonschen Kreise des $\mathbf{K}_n$ betrachten wir jenen, H, für den das Gesamtgewicht seiner Kanten minimal ist: Dieser Kreis ist eine Lösung für das mit $\mathbf{K}_n$ und Gewichtsfunktion ω verbundene Travelling Salesman Problem (die Lösung des Problems ist also ein minimaler Hamiltonscher Kreis).

Gewichtsfunktion: Beispiele

Sei $\mathbf{K}_n$ der vollständige Graph mit n Knoten, und sei eine Gewichtsfunktion $\omega : E \to \mathbb{R}$ auf der Menge seiner Kanten gegeben.

Gewichtsfunktion ω_1: €€€

Gewichtsfunktion ω_2: Zeit

Gewichtsfunktion ω_3: zurückgelegte Strecke

Andere Gewichtsfunktion $\longrightarrow$ anderer minimaler Hamiltonscher Kreis

Gewichtsfunktion: Dreiecksungleichung

Definition: Dreiecksungleichung

Sei $\mathbf{K}_n$ der vollständige Graph mit n Knoten, und sei eine Gewichtsfunktion $\omega : E \to \mathbb{R}$ auf der Menge seiner Kanten gegeben. Wir sagen, die Gewichtsfunktion erfüllt die Dreiecksungleichung, wenn für je drei paarweise verschiedene Knoten a, b und c gilt:

$$\omega(\{a, b\}) + \omega(\{b, c\}) \geq \omega(\{a, c\})$$

Spannende Baume versus Hamiltonsche Kreis

Satz: Spannende Baume versus Hamiltonsche Kreis

Sei $\mathbf{K}_n$ der vollständige Graph mit n Knoten, und sei eine Gewichtsfunktion $\omega : E \to \mathbb{R}$ auf der Menge seiner Kanten gegeben, die die Dreiecksungleichung erfüllt.

Sei m bzw. M das Gesamtgewicht der Kanten eines minimalen spannenden Baumes bzw. eines minimalen Hamiltonschen Kreises. Dann gilt:

$$m \leq M \leq 2m.$$

Spannende Baume versus Hamiltonsche Kreis

Satz: Spannende Baume versus Hamiltonsche Kreis

Sei $\mathbf{K}_n$ der vollständige Graph mit n Knoten, und sei eine Gewichtsfunktion $\omega : E \to \mathbb{R}$ auf der Menge seiner Kanten gegeben, die die Dreiecksungleichung erfüllt.

Sei m bzw. M das Gesamtgewicht der Kanten eines minimalen spannenden Baumes bzw. eines minimalen Hamiltonschen Kreises. Dann gilt:

$$m \leq M \leq 2m.$$

Definition: Eulerscher Graph

Ein Graph, in dem jeder Knoten geraden Grad hat, heißt ein Eulerscher Graph

Spannende Baume versus Hamiltonsche Kreis

Beweis: Daß $m \leq M$ ist, folgt einfach daraus, daß ein Hamiltonscher Kreis natürlich zusammenhängend ist (und ein minimaler spannender Baum ist ein zusammenhängender Graph mit minimalem Gesamtgewicht der Kanten).

Spannende Baume versus Hamiltonsche Kreis

Beweis: Daß $m \leq M$ ist, folgt einfach daraus, daß ein Hamiltonscher Kreis natürlich zusammenhängend ist (und ein minimaler spannender Baum ist ein zusammenhängender Graph mit minimalem Gesamtgewicht der Kanten).

Die zweite Ungleichung sieht man wie folgt: Zunächst konstruieren wir einen minimalen spannenden Baum $\mathbf{T}$ in $\mathbf{K}_n$. Diesen Baum machen wir zu einem Eulerschen Graphen (mit mehrfachen Kanten), indem wir alle seine Kanten “verdoppeln”.

In diesem Eulerschen Graphen konstruieren wir eine Eulersche Wanderung. Um nun einen Hamiltonschen Kreis zu konstruieren, folgen wir dieser Wanderung — aber immer, wenn wir einen Knoten erreichen, den wir schon besucht hatten, setzen wir mit dem nächsten Knoten auf der Wanderung fort, der noch nicht besucht wurde.

Spannende Baume versus Hamiltonsche Kreis

Es ist klar, daß so ein Hamiltonscher Kreis C entsteht — die Frage ist, wie groß das Gewicht der Kanten von C ist. Nach Konstruktion haben wir immer "Abkürzungen gemacht", also Abschnitte $v_i, v_{i+1}, \ldots, v_j$ der Eulerschen Wanderung durch einfache Kanten (v_i, v_j) ersetzt: Wegen der Dreiecksungleichung[2] ist aber

$$\omega\left(\{v_i, v_{i+1}\}\right) + \cdots + \omega\left(\{v_{j-1}, v_j\}\right) \geq \omega\left(\{v_i, v_j\}\right),$$

daher ist das Gewicht der Kanten von $C \leq$ dem Gewicht der Kanten der Eulerschen Wanderung, also $\leq 2m$. ■

[2] . . . und Induktion.

Digraphen und Netzwerke

Definition: Gerichteter Graph

Ein gerichteter Graph (englisch: directed graph, oder kurz Digraph) $\mathbf{D} = \mathbf{D}(V, E)$ (mit mehrfachen Kanten) ist gegeben durch eine Menge V von Knoten und eine Multimenge E der Familie der geordneten Paare von V.

Die Kanten eines gerichteten Graphen $\mathbf{D}$ besitzen also eine Orientierung, d.h., eine Kante $e = (v, u)$ verbindet nicht einfach Knoten v mit Knoten u, sondern führt von v nach u: v ist der Anfangsknoten der Kante, bezeichnet mit $\mathbf{i}(e)$, und u ist der Endknoten der Kante, bezeichnet mit $\mathbf{t}(e)$.

Der Eingangsgrad (englisch: in–degree) eines Knoten v ist die Anzahl der Kanten e mit $\mathbf{t}(e) = v$ (die also nach v führen), und der Ausgangsgrad (englisch: out–degree) ist die Anzahl der Kanten e mit $\mathbf{i}(e) = v$ (die also von v zu anderen Knoten führen).

Digraphen

Definition: Wanderungen und Wege in gerichteten Graphen

Wanderungen und Wege in gerichteten Graphen sind genauso definiert wie für "normale" Graphen — nur daß die Kanten "in der richtigen Richtung benutzt" werden müssen. D.h., wenn

$$v_0, v_1, \ldots, v_n$$

eine Wanderung in **D** ist, dann müssen die geordneten Paare (v_{i-1}, v_i) für $i = 1, \ldots, n$ Kanten aus E sein.

Wenn man in einem Digraphen **D** die Orientierung "vergißt", also jede gerichtete Kante (v, w) durch eine "normale" Kante $\{v, w\}$ ersetzt, dann erhält man einen "normalen" Graphen **G** (um den Gegensatz zu einem gerichteten Graphen zu betonen, spricht man auch von einem ungerichteten Graphen): Dieser wird der zugrundeliegende Graph von **D** genannt.

Netzwerke

Definition: Netzwerk

Ein Netzwerk ist ein Digraph $\mathbf{N}(V, E)$ mit zwei ausgezeichneten Knoten, einer Quelle $\mathbf{q}$ und einer Senke $\mathbf{s}$, $\mathbf{q} \neq \mathbf{s}$, und mit einer Gewichtsfunktion $\mathbf{c} : E \to \mathbb{R}^+$ (die also nur nichtnegative Werte annimmt) und die in diesem Zusammenhang Kapazität heißt.

Netzwerke: Fluß

Definition: Fluß

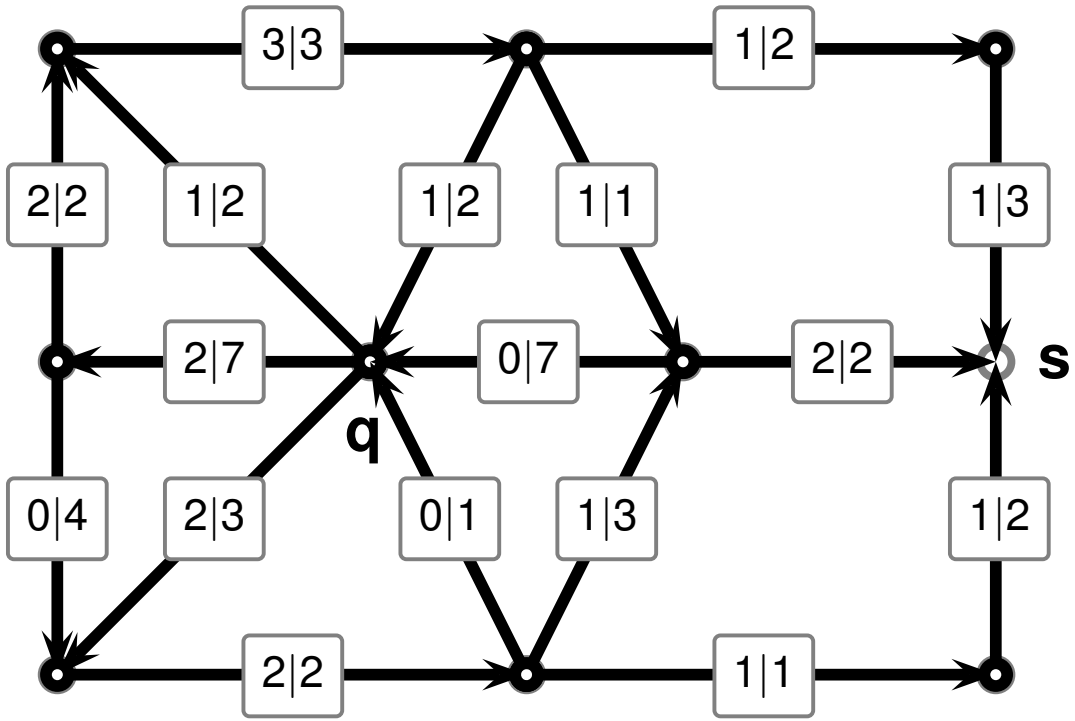
Ein Fluß (englisch: Flow) in einem Netzwerk $\mathbf{N}(V, E)$ (mit Quelle $\mathbf{q}$, Senke $\mathbf{s}$ und Kapazität $\mathbf{c}$) ist eine Funktion $f : E \to \mathbb{R}$ mit den Eigenschaften

- $0 \leq f(e) \leq \mathbf{c}(e)$ für alle Kanten $e \in E$,
- $\sum_{\mathbf{i}(e)=v} f(e) = \sum_{\mathbf{t}(e)=v} f(e)$ für alle Knoten $\neq \mathbf{q}, \mathbf{s}$.

Die Stärke $\mathrm{val}(f)$ eines Flusses f ist definiert als

$$\mathrm{val}(f) = \sum_{\mathbf{i}(e)=\mathbf{q}} f(e) - \sum_{\mathbf{t}(e)=\mathbf{q}} f(e).$$

Beispiel: Netzwerk (aus Skriptum)



Fluss|Kapazität der Kanten sind in kleinen Kästchen eingetragen:
Ersichtlich hat der Fluss Stärke 4

Netzwerke: Lemma

Für eine beliebige Teilmenge $S \subseteq V$ von Knoten eines Netzwerks $\mathbf{N}(V, E)$ führen wir folgende Notation ein:

- $S^{\rightarrow} := \{e \in E : \mathbf{i}(e) \in S, \mathbf{t}(e) \notin S\}$ ("Netto–Output"),
- $S^{\leftarrow} := \{e \in E : \mathbf{t}(e) \in S, \mathbf{i}(e) \notin S\}$ ("Netto–Input").

Lemma: Die Stärke

Sei f ein Fluß in einem Netzwerk $\mathbf{N}(V, E)$ mit Quelle **q** und Senke **s**. Sei S eine beliebige Teilmenge von V, die **q** enthält, aber nicht **s**. Dann gilt:

$$\sum_{e \in S^{\rightarrow}} f(e) - \sum_{e \in S^{\leftarrow}} f(e) = \mathrm{val}(f).$$

Netzwerke: Die Stärke

Beweis: Wir betrachten

$$\sum_{v \in S} \left(\sum_{\mathbf{i}(e)=v} f(e) - \sum_{\mathbf{t}(e)=v} f(e) \right).$$

Einerseits ergibt diese Summe $\mathrm{val}(f)$, denn alle Summanden für $v \neq \mathbf{q}$ sind Null nach Definition eines Flusses, und der Summand $v = \mathbf{q}$ liefert genau $\mathrm{val}(f)$.

Netzwerke: Die Stärke

Beweis: Wir betrachten

$$\sum_{v \in S} \left(\sum_{\mathbf{i}(e)=v} f(e) - \sum_{\mathbf{t}(e)=v} f(e) \right).$$

Einerseits ergibt diese Summe $\mathrm{val}(f)$, denn alle Summanden für $v \neq \mathbf{q}$ sind Null nach Definition eines Flusses, und der Summand $v = \mathbf{q}$ liefert genau $\mathrm{val}(f)$.

Andrerseits können wir die Summe auch als Summe über Kanten interpretieren: Sei $e = (v, w)$ eine Kante. Wenn $v \in S$, dann kommt $+f(e)$ in der Summe vor; wenn auch $w \in S$, dann kommt $-f(e)$ in der Summe vor — diese Terme heben sich also gegenseitig auf.

Netzwerke: Die Stärke

Beweis: Wir betrachten

$$\sum_{v \in S} \left(\sum_{\mathbf{i}(e)=v} f(e) - \sum_{\mathbf{t}(e)=v} f(e) \right).$$

Einerseits ergibt diese Summe $\mathrm{val}(f)$, denn alle Summanden für $v \neq \mathbf{q}$ sind Null nach Definition eines Flusses, und der Summand $v = \mathbf{q}$ liefert genau $\mathrm{val}(f)$.

Andrerseits können wir die Summe auch als Summe über Kanten interpretieren: Sei $e = (v, w)$ eine Kante. Wenn $v \in S$, dann kommt $+f(e)$ in der Summe vor; wenn auch $w \in S$, dann kommt $-f(e)$ in der Summe vor — diese Terme heben sich also gegenseitig auf.

Daher tragen nur solche Kanten etwas zur Summe bei, die genau einen Knoten in S haben, das sind genau jene in $S^{\rightarrow}$ (die entsprechenden Terme werden addiert) bzw. in $S^{\leftarrow}$ (die entsprechenden Terme werden subtrahiert). ■

Netzwerke: Die maximale Stärke

Was ist die maximale Stärke eines Flusses in einem gegebenen Netzwerk?’

Netzwerke: Schnitt

Definition: Schnitt

Ein Schnitt (englisch: Cut) C in einem Netzwerk $\mathbf{N}(V, E)$ ist eine Teilmenge $C \subseteq E$ von Kanten, sodaß jeder Weg von der Quelle $\mathbf{q}$ zur Senke $\mathbf{s}$ mindestens eine Kante aus C enthält.

Die Kapazität $\mathbf{c}(C)$ des Schnitts ist die Summe der Kapazitäten der Kanten in C, also

$$\mathbf{c}(C) = \sum_{e \in C} \mathbf{c}(e).$$

Netzwerke: Fluß und Schnitt

Lemma: Fluß und Schnitt

Sei $\mathbf{N}(V, E)$ ein Netzwerk mit Quelle **q**, Senke **s** und Kapazitätsfunktion **c**.

Dann ist die Stärke eines beliebigen Flusses kleiner oder gleich der Kapazität eines beliebigen Schnitts.

Netzwerke: Fluß und Schnitt

Beweis: Sei C ein Schnitt, und sei S die Menge der Knoten in $\mathbf{N}$, die von der Quelle $\mathbf{q}$ aus durch Wege erreichbar sind, die keine Kanten aus C enthalten. Dann ist $S^{\rightarrow} \subseteq C$.

Nach Lemma “Die Stärke” gilt für einen beliebigen Fluß f:

$$\mathrm{val}\,(f) = \sum_{e \in S^{\rightarrow}} f\,(e) - \sum_{e \in S^{\leftarrow}} f\,(e) \leq \sum_{e \in S^{\rightarrow}} \mathbf{c}\,(e) \leq \sum_{e \in C} \mathbf{c}\,(e) = \mathbf{c}\,(C)\,.$$

Max–Flow–Min–Cut–Theorem

Satz: Max–Flow–Min–Cut–Theorem, Satz von Ford–Fulkerson

Sei $\mathbf{N}(V, E)$ ein Netzwerk mit Quelle **q**, Senke **s** und Kapazitätsfunktion **c**.

Dann ist die maximale Stärke eines Flusses gleich der minimalen Kapazität eines Schnitts.

Max–Flow–Min–Cut: Lemma

Lemma:

Sei $\mathbf{N}(V, E)$ ein Netzwerk mit Quelle $\mathbf{q}$, Senke $\mathbf{s}$ und Kapazitätsfunktion $\mathbf{c} : E \to \mathbb{Z}^+$ (d.h., alle Kapazitäten sind ganze nichtnegative Zahlen).

Dann gibt es einen Fluß f in $\mathbf{N}$, der in jeder Kante ganzzahlig ist (d.h., $f(e) \in \mathbb{Z}^+$ für alle $e \in E$: Wir sagen, f ist ein ganzzahliger Fluß), und einen Schnitt C, sodaß

$$\mathrm{val}(f) = \mathbf{c}(C).$$

Max–Flow–Min–Cut: Lemma

Lemma:

Sei $\mathbf{N}(V, E)$ ein Netzwerk mit Quelle $\mathbf{q}$, Senke $\mathbf{s}$ und Kapazitätsfunktion $\mathbf{c} : E \to \mathbb{Z}^+$ (d.h., alle Kapazitäten sind ganze nichtnegative Zahlen).

Dann gibt es einen Fluß f in $\mathbf{N}$, der in jeder Kante ganzzahlig ist (d.h., $f(e) \in \mathbb{Z}^+$ für alle $e \in E$: Wir sagen, f ist ein ganzzahliger Fluß), und einen Schnitt C, sodaß

$$\mathrm{val}(f) = \mathbf{c}(C).$$

Insbesondere ist f ein Fluß maximaler Stärke und C ein Schnitt minimaler Kapazität.

Satz “Max–Flow–Min–Cut” ist damit also für ganzzahlige Kapazitätsfunktionen gezeigt

Max–Flow–Min–Cut: Lemma

Beweis: Das Kernstück unseres Beweises ist die Behauptung: Für einen ganzzahligen Fluß f gibt es

- entweder einen ganzzahligen Fluß f' mit $\mathrm{val}(f') = \mathrm{val}(f) + 1$
- oder einen Schnitt C mit $\mathbf{c}(C) = \mathrm{val}(f)$.

Max–Flow–Min–Cut: Lemma

Beweis: Das Kernstück unseres Beweises ist die Behauptung: Für einen ganzzahligen Fluß f gibt es

- entweder einen ganzzahligen Fluß f' mit $\mathrm{val}(f') = \mathrm{val}(f) + 1$
- oder einen Schnitt C mit $\mathbf{c}(C) = \mathrm{val}(f)$.

Wenn wir das zeigen können, können wir algorithmisch vorgehen und mit dem "Nullfluß" $f \equiv 0$ starten, den wir sukzessive "verstärken" (solange die erste Alternative der Behauptung zutrifft), bis die zweite Alternative der Behauptung zutrifft und wir einen maximalen Fluß konstruiert haben.
Zum Beweis dieser Behauptung konstruieren wir algorithmisch zwei Mengen $S \subseteq V$ und $E_S \subseteq E$:

Max–Flow–Min–Cut: Lemma

{Initialisierung:}

1: $S \leftarrow \{\mathbf{q}\}$, $E_S \leftarrow \emptyset$.

{Schleife: Wird wiederholt, solange die Bedingung erfüllt ist.}

2: **while** (Bedingung: $\mathbf{s} \notin S$ und es existiert eine Kante $e = (v, w)$, für die entweder $f(e) < \mathbf{c}(e)$ und $v \in S$, aber $w \notin S$ (i.e.: $e \in S^{\rightarrow}$) oder $f(e) > 0$ und $v \notin S$, aber $w \in S$ (i.e.: $e \in S^{\leftarrow}$)) **do**

3: $S \leftarrow S \cup \{v, w\}$ (i.e.: gib den Knoten von e, der noch nicht in S enthalten ist (also v oder w), zu S dazu.)

4: $E_S \leftarrow E_S \cup \{e\}$ (i.e.: gib die Kante e zu E_S dazu.)

5: **end while**

Max–Flow–Min–Cut: Lemma

Die so konstruierte Knotenmenge S ergibt zusammen mit der Kantenmenge E_S einen (gerichteten) Teilgraphen $\mathbf{N}'$ von $\mathbf{N}$, dessen zugrundeliegender (ungerichteter) Graph (in dem man also “die Orientierung der Kanten vergißt”) ein Baum ist.

Es gibt nun zwei Möglichkeiten:

Max–Flow–Min–Cut: Lemma

Fall 1: $\mathbf{s} \in S$. Dann gibt es eine Folge von paarweise verschiedenen Knoten $\mathbf{q} = v_0, v_1, \ldots, v_n = \mathbf{s}$, sodaß für alle $i = 1, \ldots, n$

(a) entweder $(v_{i-1}, v_i) =: e_i \in E_S$ ist eine der Kanten mit $f(e_i) < \mathbf{c}(e_i)$,

(b) oder $(v_i, v_{i-1}) =: e_i \in E_S$ ist eine der Kanten mit $f(e_i) > 0$

gilt[3]. Die Menge dieser Kanten $\{e_i : 1 \leq i \leq n\}$ zerfällt also in zwei disjunkte Teilmengen

- $A \subseteq E_S$: für $e_i \in A$ trifft Alternative (a) zu,
- $B \subseteq E_S$: für $e_i \in B$ trifft Alternative (b) zu.

[3] Im zugrundeliegenden Graphen von $\mathbf{N}'$ (wo also "auf die Orientierung der Kanten vergessen wird") entspricht diese Folge dem eindeutigen Weg von **q** nach **s**.

Max–Flow–Min–Cut: Lemma

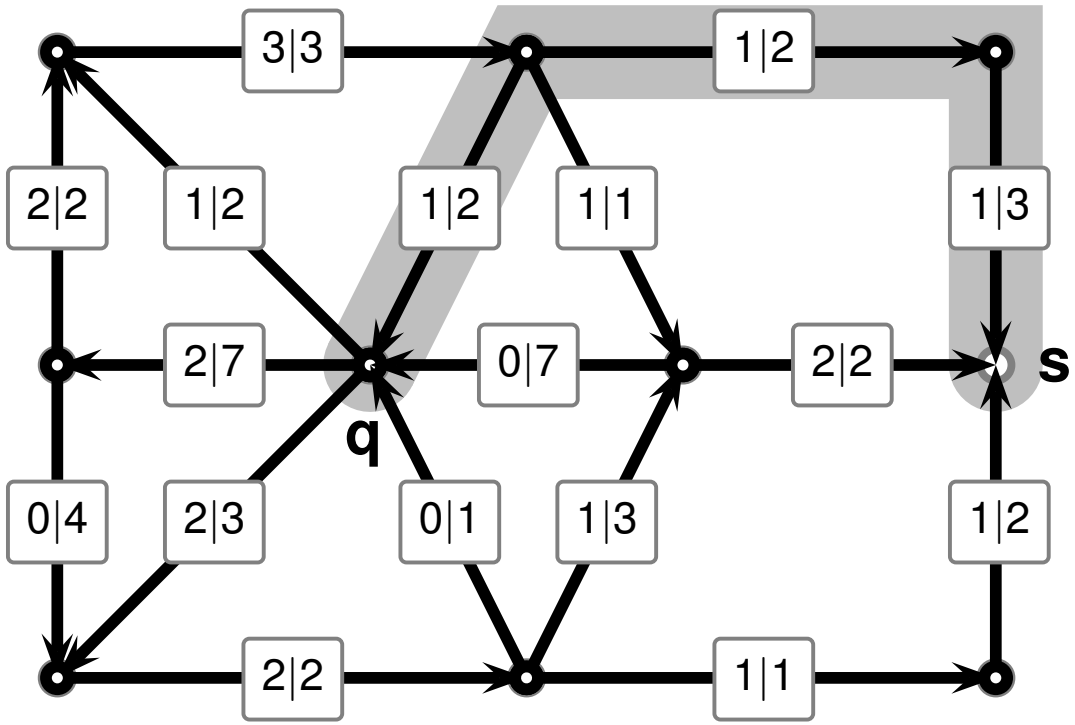
Nun definieren wir einen Fluß f' wie folgt:

$$f'(e) = \begin{cases} f(e) + 1 & \text{falls } e \in A, \\ f(e) - 1 & \text{falls } e \in B, \\ f(e) & \text{sonst.} \end{cases}$$

Es ist leicht zu sehen, daß $0 \leq f'(e) \leq \mathbf{c}(e)$ für alle Kanten $e \in E$ gilt, und daß für alle Knoten v_i, $i = 1, \ldots, n-1$ (andere Knoten — außer **q** und **s** — sind von der Änderung ja gar nicht betroffen!) nach wie vor $\sum_{\mathbf{i}(e)=v_i} f'(e) = \sum_{\mathbf{t}(e)=v_i} f'(e)$ gilt: f' ist also tatsächlich ein Fluß. Ebenso ist leicht zu sehen, daß $\mathrm{val}(f') = \mathrm{val}(f) + 1$.

Beispiel (aus Skriptum)

Die folgende Graphik zeigt eine Menge S (grau unterlegt), die sich mit diesem Schritt ergeben würde: Die Senke **s** gehört zu S, daher läßt sich der Fluß um 1 erhöhen.



Max–Flow–Min–Cut: Lemma

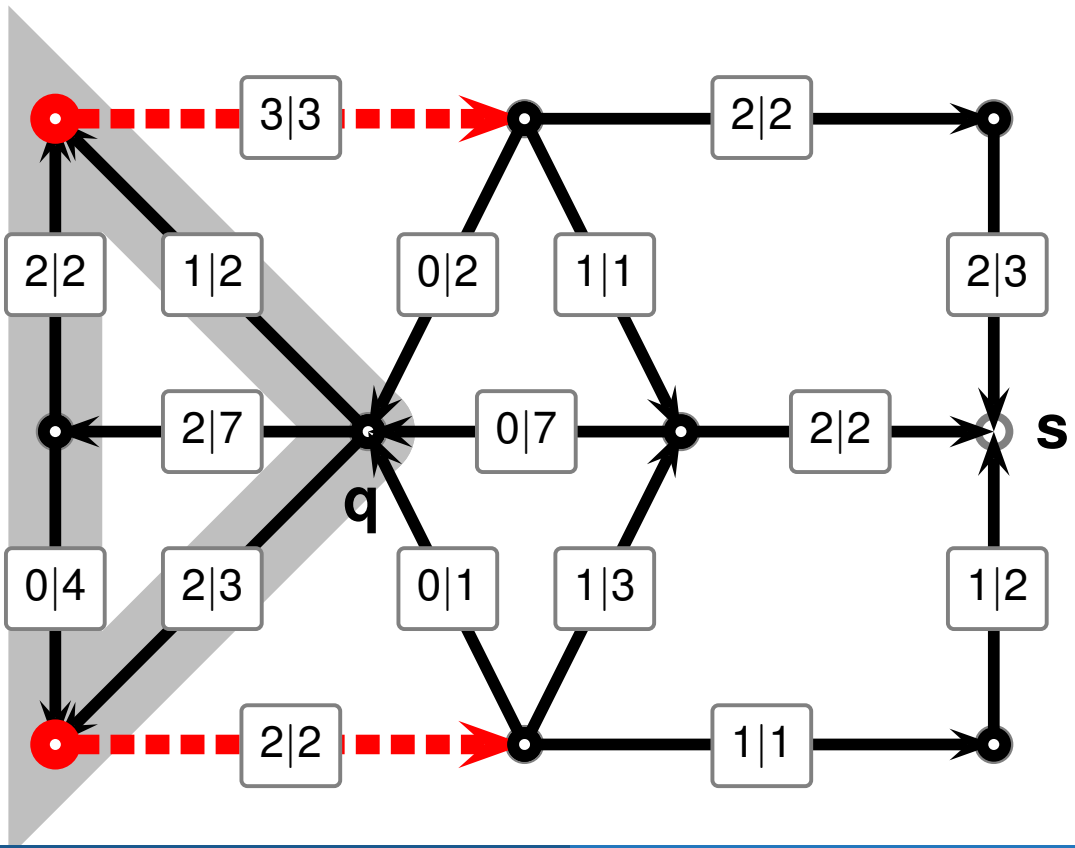
Fall 2: $\mathbf{s} \notin S$. Dann ist aber $S^{\rightarrow}$ ein Schnitt, und nach Konstruktion von S gilt für jede Kante $e \in S^{\rightarrow}$: $f(e) = \mathbf{c}(e)$, und es gilt für jede Kante $e \in S^{\leftarrow}$: $f(e) = 0$. Daher ist

$$\mathrm{val}(f) = \sum_{e \in S^{\rightarrow}} f(e) - \sum_{e \in S^{\leftarrow}} f(e) = \sum_{e \in S^{\rightarrow}} \mathbf{c}(e) = \mathbf{c}(S^{\rightarrow}).$$

Damit ist die obige Behauptung (und unser Lemma) gezeigt. ■

Beispiel (aus Skriptum)

Die folgende Graphik zeigt die Situation, die sich nach der Erhöhung des Flusses aus der vorigen Graphik ergibt: Die Menge S (wieder grau unterlegt) enthält nun die Senke **s** nicht, und die beiden aus S herausführenden Kanten (die strichlierten roten Kanten links oben und links unten) bilden einen Schnitt.



Max–Flow–Min–Cut–Theorem: Beweis

Beweis: Nach Lemma ist die Behauptung für ganzzahlige Kapazitätsfunktionen richtig. Dann gilt sie aber auch für Kapazitätsfunktionen $\mathbf{c} : E \to \mathbb{Q}^+$: Dazu multiplizieren wir alle Kapazitäten $\mathbf{c}(e)$ mit dem gemeinsamen Nenner m und erhalten so ein Netzwerk mit ganzzahligen Kapazitäten, für dieses gilt der Satz, und durch "Division aller Kapazitäten und Flüsse durch m" sehen wir die Gültigkeit des Satzes im ursprünglichen Netzwerk **N**.

Max–Flow–Min–Cut–Theorem: Beweis

Für reellwertige Kapazitätsfunktionen können wir alle Kapazitäten $\mathbf{c}(e)$ beliebig genau von unten durch rationale Zahlen approximieren, und durch Anwendung unseres Satzes für rational–wertige Kapazitätsfunktionen finden wir Flüsse, die der Kapazität eines minimalen Schnittes beliebig nahe kommen: Die Aussage folgt dann durch Übergang zum Grenzwert. ■

Max–Flow–Min–Cut: Bemerkung

Strenggenommen ist dieses Argument noch nicht ausreichend — wir müssen ja “konvergente Flüsse und Schnitte” haben, nicht nur konvergente Stärken und Kapazitäten.

Dazu müßten wir folgendermaßen vorgehen: Unsere Konstruktion liefert ja eine Folge von Flüssen/Schnitten, deren Stärken/Kapazitäten konvergieren. Nun betrachten wir eine Kante k_1: Die Fluß–Werte auf dieser Kante müssen nicht konvergieren, da sie aber beschränkt sind, muß es einen Häufungspunkt und somit eine konvergente Teilfolge von Flüssen/Schnitten geben.

Danach wendet man dasselbe Argument für diese Teilfolge und Kante k_2 an, u.s.w: Da wir einen endlichen Graphen haben, bricht dieses Verfahren ab.