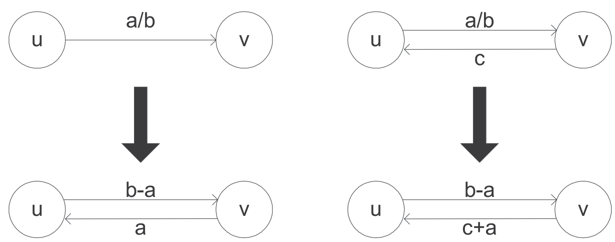
Algorithmen für Fortgeschrittene

Jan Sebastian Siwy   Martin Spickermann

2. Vorlesung vom 15. April 2004

Nachtrag: Bildung des Restnetzes aus der letzten Vorlesung.

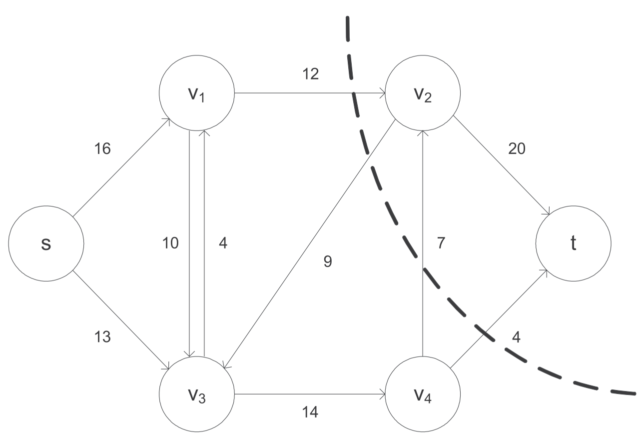


Kanten mit der Kapazität 0 können weggelassen werden.

Korollar: Der Wert jedes Flusses im Netz G ist durch die minimale Kapazität aller denkbarer Schnitte nach oben beschränkt, denn

$$f(S,T) \leq c(s,t).$$

Im Beispiel kann der Fluss nicht größer sein als 23 (Schnittverlauf laut Skizze):



Satz 1: Sei f Fluss im Netz G_f, dann sind folgende Aussagen äquivalent:

1. f ist maximal.
2. Es gibt keine augmentierenden Wege bzgl. G, c, f.
3. Es gibt einen Schnitt S, T mit $|f| = c(S, T)$.
   (Bemerkung: Dieser Schnitt hat minimale Kapazität.)

Beweis:

- $1 \Rightarrow 2$:

  trivial

- $2 \Rightarrow 3$:

  Es gibt in G keinen augmentierenden Weg, d.h. in G_f gibt es keinen Weg von s nach t.

  Sei $S = \{v \in V \mid \exists \text{ Weg von } s \text{ nach } v \text{ in } G_f\}$ und $T = V \backslash S$.

  Betrachte Schnitt S, T, bei dem $f(u, v) = c(u, v)$ für alle $u \in S$ und $v \in T$.

  Daraus folgt:

  $$\begin{aligned} |f| &= f(S, T) && \text{(nach Lemma 2)} \\ &= c(S, T) \end{aligned}$$

- $3 \Rightarrow 1$:

  Nach dem Korollar gilt $|f| \leq c(S, T)$ für alle Flüsse und Schnitte, also auch für diesen.

  Da $|f| = c(S, T)$ gilt, ist f maximal.

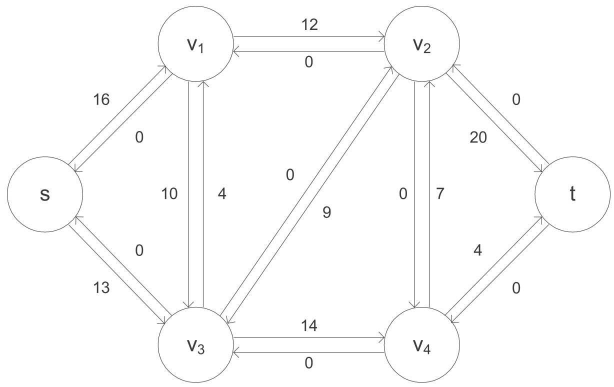
$\square$

Ford-Fulkerson-Algorithmus (Schema):

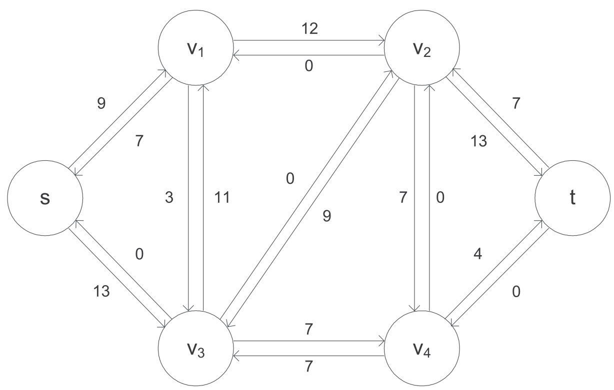
Initialisiere den Fluss f auf 0.
while $\exists$ augmentierender Weg p von s nach t im Restnetz G_f **do**
  $\forall$ Kante $e \in p$ erhöhe den Fluss f um die Kapazität $c_f(p)$ dieses Weges, wobei $c_f(p) = \min c_f(e)$
end while

In unserem Beispiel:

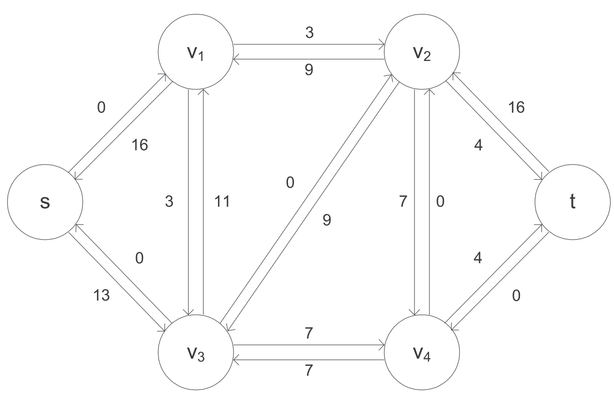
- Initialisierung:



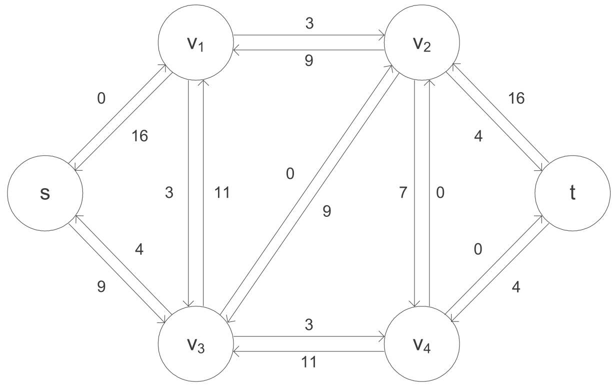
- Schritt 1 (Weg über s, v_1, v_3, v_4, v_2, t mit Minimum 7):



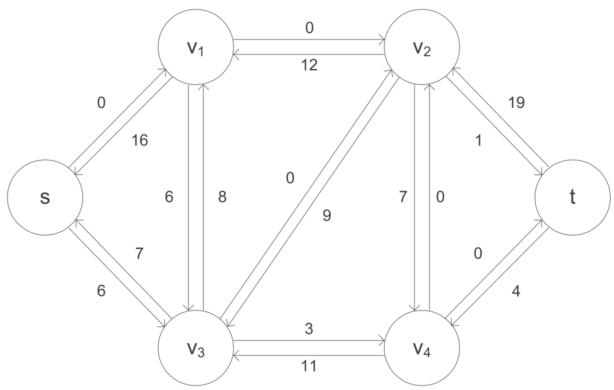
- Schritt 2 (Weg über s, v_1, v_2, t mit Minimum 9):



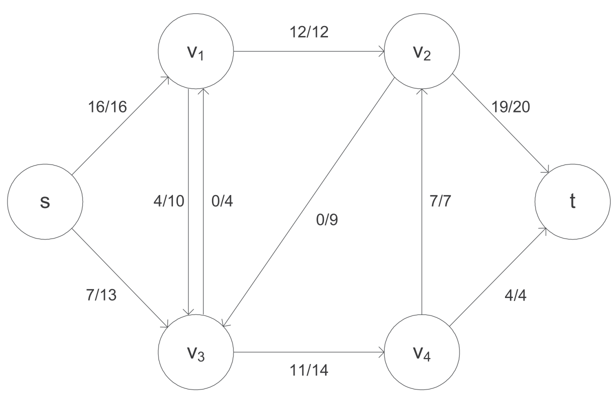
- Schritt 3 (Weg über s, v_3, v_4, t mit Minimum 4):



- Schritt 4 (Weg über s, v_3, v_1, v_2, t mit Minimum 3):



- Lösung:



Einzelheiten des Algorithmus:

- Schritt 1:

  Variable für Fluss f definieren und diese auf 0 setzen.

- Schritt 2:

  Finden eines Weges. Konstruiere dazu als Datenstruktur einen Graphen G':

  $$G' = (V, E') \quad \text{mit} \quad E' = E \cup \{(u, v) \mid (v, u) \in E\}$$

  Jedes Restnetz ist Teilgraph von G'. Kanten mit Rest 0 können ignoriert werden.

- Schritt 3:

  Konstruktion bzw. Aktualisierung des Restnetzes.

Laufzeitanalyse:

- Schritt 1:

  Kostet $\mathcal{O}(|E|)$.

- Schritt 2:

  Kostet $\mathcal{O}(|E|)$ mit Breiten- oder Tiefensuche pro Durchlauf.

- Schritt 3:

  Kostet $\mathcal{O}(|E|)$ pro Durchlauf.

  Jede Kante auf p und die Gegenkante muss aktualisiert werden.

Wie viele Durchläufe benötigt nun der Algorithmus insgesamt?

- Bei jedem Durchlauf wird der Fluss erhöht.

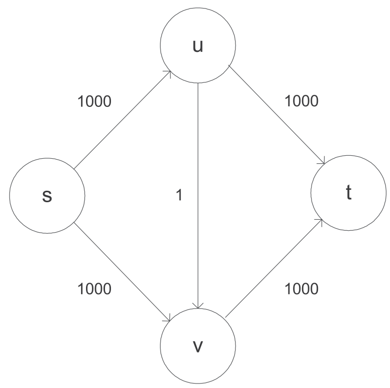
- Wenn wir annehmen, dass die Kapazitäten ganze Zahlen sind, erhöht sich der Fluss um mindestens 1 je Durchgang.

Wenn f^* der maximale Fluss ist, so gibt es höchstens $|f^*|$ Durchläufe. Die Laufzeit des Ford-Fulkerson-Algorithmus ist also $\mathcal{O}(|E| \cdot |f^*|)$.

Bemerkungen durch Laufzeitanalyse: Diese Aussage zur Laufzeit ist unbefriedigend, weil

- wir angenommen haben, dass die Kapazitäten ganze Zahlen sind und
- der maximale Fluss $|f^*|$ exponentiell zur Eingabegröße sein kann.

Der maximale Fluss $|f^*|$ ist tatsächlich möglich!



Es werden 2000 Durchläufe erreicht, wenn abwechselnd die Pfade s, u, v, t und s, v, u, t gewählt werden.

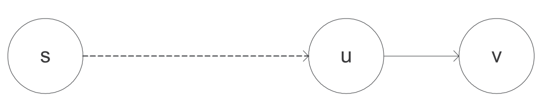
Edmond-Karp-Algorithmus: Dieser findet immer den kürzesten Weg durch Breitensuche in Schritt 2.

Lemma 3: Sei $\delta_f(u, v)$ der Abstand $u, v \in V$ im Restnetz G_f.

Beim Edmond-Karp-Algorithmus gilt für alle Knoten $v \in V \backslash \{s, t\}$, dass in jedem augmentierendem Schritt der Abstand $\delta_f(u, v)$ monoton wächst.

Beweis: Angenommen das Lemma gilt nicht, d.h. es existieren ein augmentierender Schritt und ein Knoten v, so dass $\delta_{f'}(s, v) < \delta_f(s, v)$ gilt mit f Fluss vor und f' Fluss nach dem augmentierendem Schritt.

O.b.d.A. sei v der Knoten mit der Eigenschaft, dass der Abstand $\delta_f(s, v)$ minimal ist. Der kürzeste Weg von s nach v in $G_{f'}$ sei p'. Der Knoten u sei der vorletzte Knoten auf diesem Weg p'.



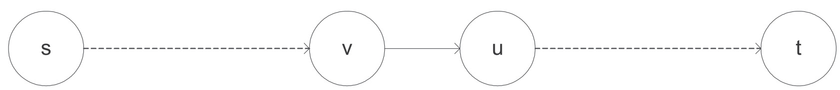
Wegen der Minimalität von v gilt $\delta_{f'}(s,v) \geq \delta_f(s,u)$. Betrachte den Fluss $f(u,v)$ vor dem augmentierenden Schritt:

- Fall 1: $f(u,v) < c(u,v)$, d.h. (u,v) ist eine Kante in G_f.

$$\begin{aligned} \delta_f(s,v) &\leq \delta_f(s,u) + 1 \\ &\leq \delta_{f'}(s,u) + 1 \\ &= \delta_{f'}(s,v) && \text{(Widerspruch!)} \end{aligned}$$

- Fall 2: $f(u,v) = c(u,v)$, d.h. (u,v) ist keine Kante in G_f.

  Damit muss der augmentierende Weg p die Kante (u,v) enthalten haben.



$$\begin{aligned} \delta_f(s,v) &= \delta_f(s,u) - 1 \\ &\leq \delta_{f'}(s,u) - 1 \\ &= \delta_{f'}(s,v) - 2 \\ &< \delta_{f'}(s,v) && \text{(Widerspruch!)} \end{aligned}$$

$\square$