Laufzeitanalyse

Lemma G

Es gibt von jedem überfließenden Knoten u einen Pfad nach s in G_f.

Beweis

Nehmen wir an, es gäbe keinen solchen Pfad.

Sei $U = \{\, v \mid v \text{ von } u \text{ in } G_f \text{ erreichbar} \,\}$.

$$\begin{aligned} 0 &\geq f(V-U,U) \text{ sonst Residualkante aus } U \text{ heraus} \\ &= f(V-U,U) + f(U,U) \\ &= f(V,U) = e(U) \end{aligned}$$

Also ist $e(u) = 0$. Widerspruch. $\square$

Laufzeitanalyse

Lemma

Es werden nur $O(|V|^2)$ Lift-Operationen ausgeführt.

Beweis

- Eine Lift-Operation erhöht $h(u)$ um mindestens 1
- Sie wird nur auf überfließende Knoten angewendet
- $h(u) < 2|V|$, da Pfad in G_f zu s mit $h(s) = |V|$ (Lemma G)

Laufzeitanalyse

Lemma

Es werden nur $O(|V| \cdot |E|)$ saturierte Push-Operationen durchgeführt.

Beweis

- Wird eine Push-Operation auf (u, v) angewandt, dann führt (u, v) um 1 nach unten
- (u, v) verschwindet dann aus G_f (weil saturierter Push)
- Vor der nächsten Push-operation auf (u, v) muß ein Push auf (v, u) durchgeführt werden, dabei führt (u, v) um 1 nach oben
- Zwischen zwei saturierten Pushs auf (u, v) erhöht sich $h(u)$

$\Rightarrow$ Nur $O(|V|)$ saturierte Pushs auf (u, v) weil $h(u) = O(|V|)$.

Laufzeitanalyse

Lemma

Es werden nur $O(|V|^2 \cdot |E|)$ nicht-saturierte Push-Operationen durchgeführt.

Beweis

Sei $\Phi = \sum_{u \in X} h(u)$ mit $X = \{ v \mid e(v) > 0 \}$.

- Eine Lift-Operation erhöht Φ höchstens um $|V|$
- Ein saturierter Push erhöht Φ höchstens um $2|V|$
- Ein nicht-saturierter Push erniedrigt Φ um mindestens 1

Insgesamt wird Φ höchstens um $O(|V|^2|E|)$ erhöht.

$\Rightarrow$ höchstens $O(|V|^2|E|)$ nicht-saturierte Pushs.

Der einfache Preflow-Push-Algorithmus

Theorem

Ein maximaler Fluß kann in $O(|V|^2|E|)$ Schritten berechnet werden.

Beweisskizze

Der einfache Preflow-Push-Algorithmus muß so implementiert werden, daß

- in konstanter Zeit ein überfließender Knoten u gefunden wird, falls noch einer existiert,
- in konstanter Zeit bestimmt wird, ob eine Push- oder Lift-Operation anwendbar ist,
- eine Lift-Operation in $O(|V|)$ Schritten durchgeführt werden kann, und
- eine Push-Operation in konstanter Zeit durchgeführt wird.

Minimum Cost Flow Problem

Eingabe:

Ein s-t-Netzwerk $G = (V, E)$, eine Zahl $\bar{f}$ und eine *Kostenfunktion* $b\colon E \to \mathbf{R}_\mathbf{0}^+$.

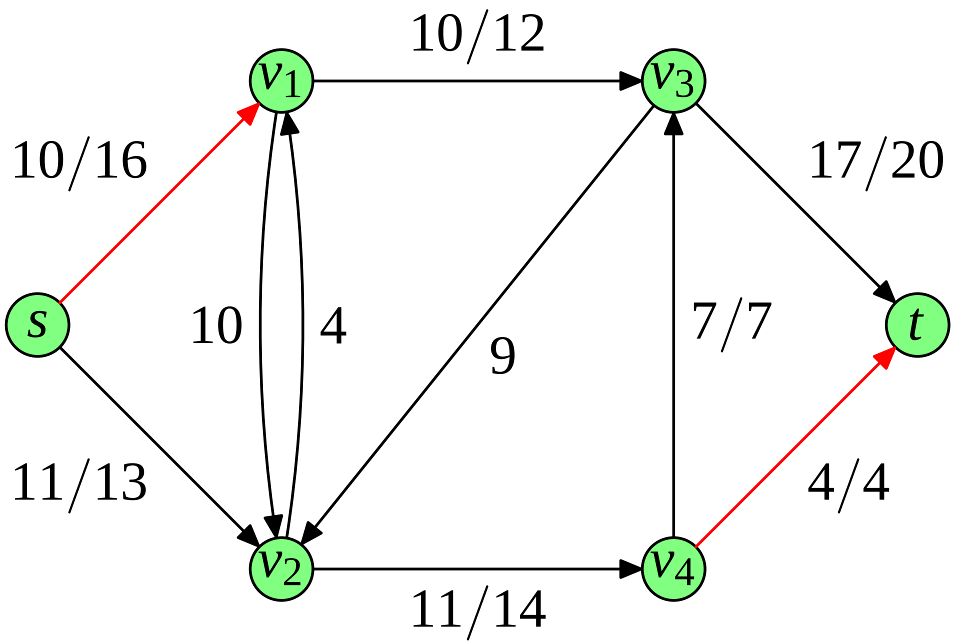
Gesucht:

Ein Fluß f mit $|f| = \bar{f}$ und minimalen Kosten, falls ein solcher existiert.

Die *Kosten* eines Flusses f sind

$$\sum_{e \in E} b(e) f(e).$$

Beispiel



Die schwarzen Kanten haben Kosten 1 und die roten haben Kosten 5.

Die Kosten des Flusses sind 126 und der Betrag 21.

Gibt es einen billigeren Fluß mit Betrag 21?

Minimum Cost Flow Problem

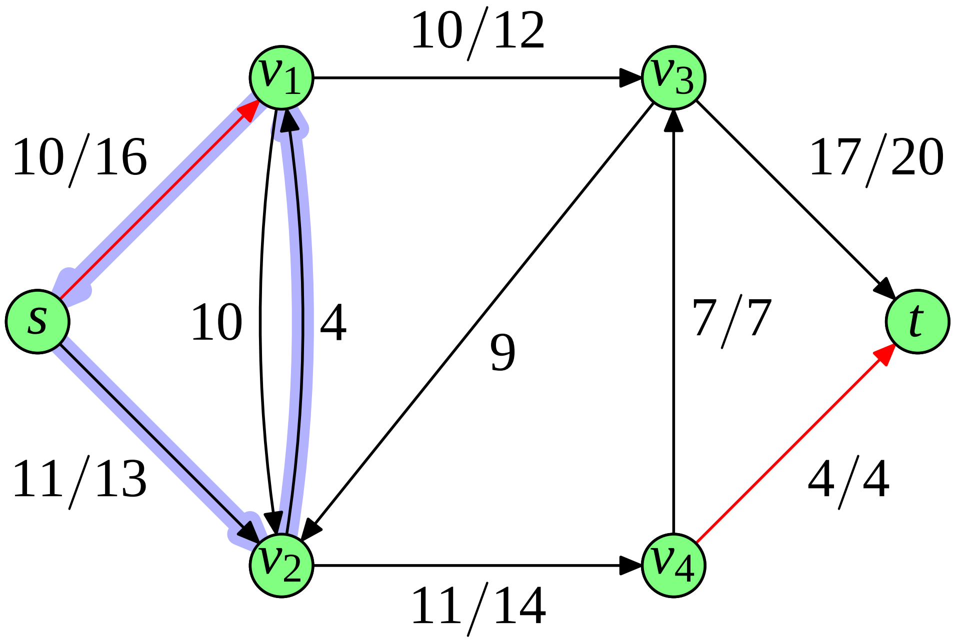
Theorem (Minimum Cost Flow Theorem)

1. Ein Fluß f mit Betrag $|f|$ hat genau dann minimale Kosten, wenn es in G_f keinen Kreis mit negativen Kosten gibt.

2. Seien f ein Fluß mit minimalen Kosten und p ein augmentierender Pfad mit minimalen Kosten. Dann entsteht wieder ein Fluß mit minimalen Kosten, wenn f entlang p augmentiert wird.

Die Kosten eines Pfads (oder Kreises) im Residualnetzwerk sind die Kosten der Kanten in Vorwärtsrichtung minus die Kosten der Kanten in Rückwärtsrichtung.

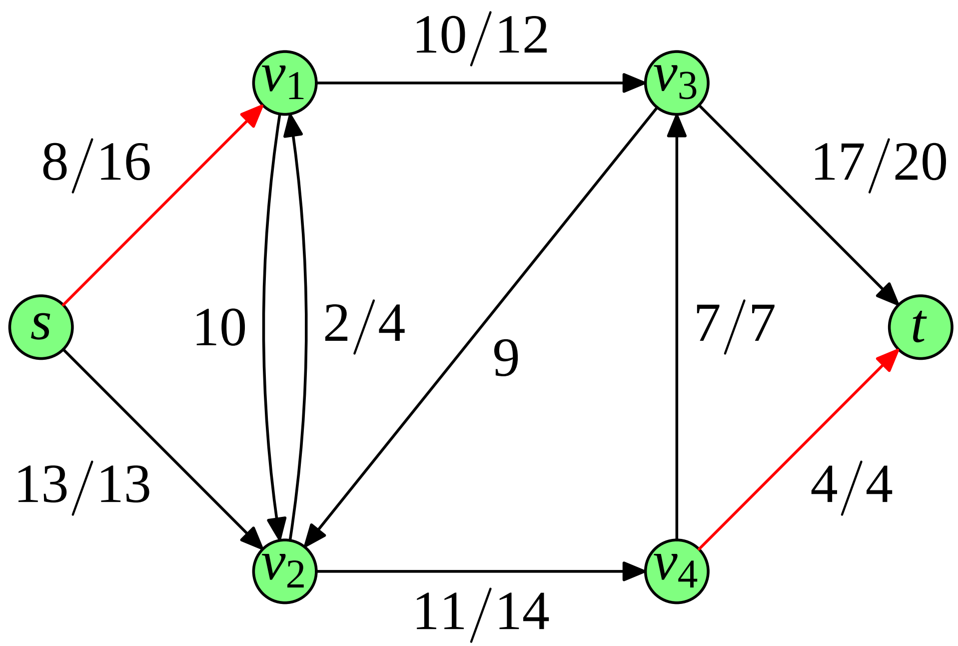
Beispiel



Da der Fluß nicht minimale Kosten hat, muß es einen Kreis in G_f mit negativen Kosten geben.

Seine Restkapazität ist 2 und seine Kosten sind -3.

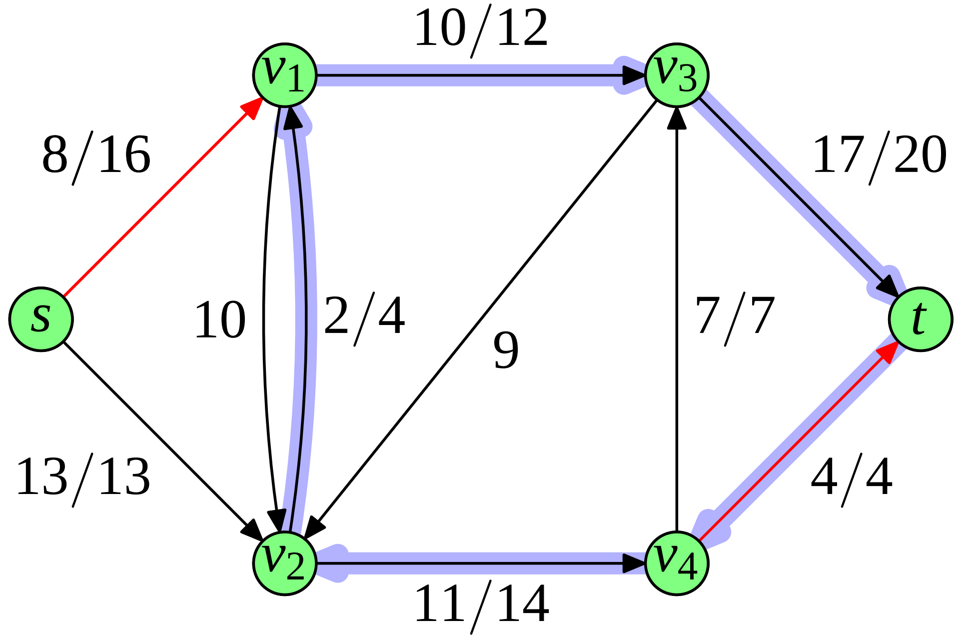
Beispiel



Augmentiert man den Fluß entlang des Kreises, erhalten wir einen neuen, billigeren Fluß.

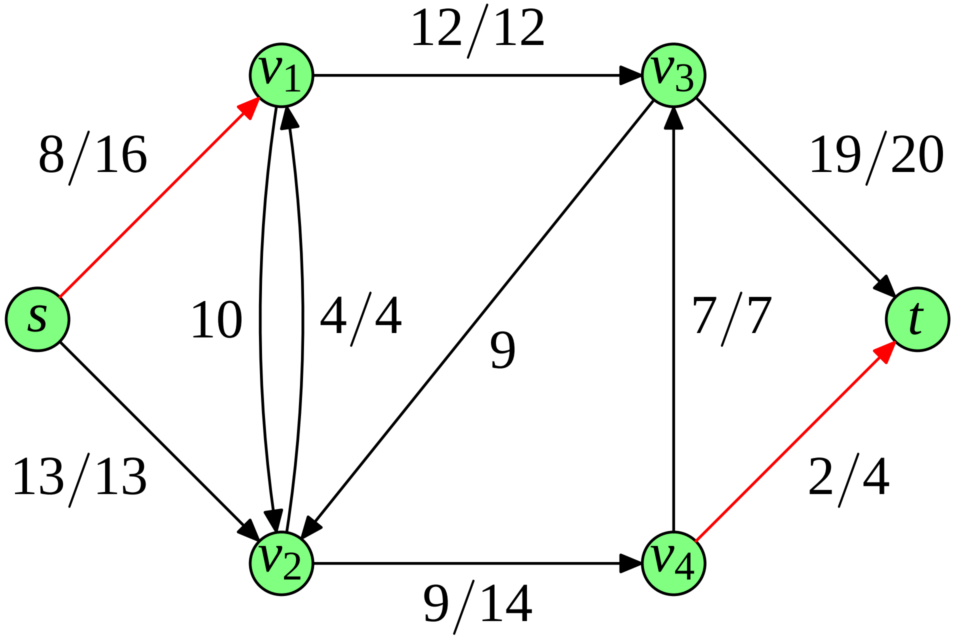
Die Kosten sind jetzt 120. Gibt es einen noch billigeren?

Beispiel



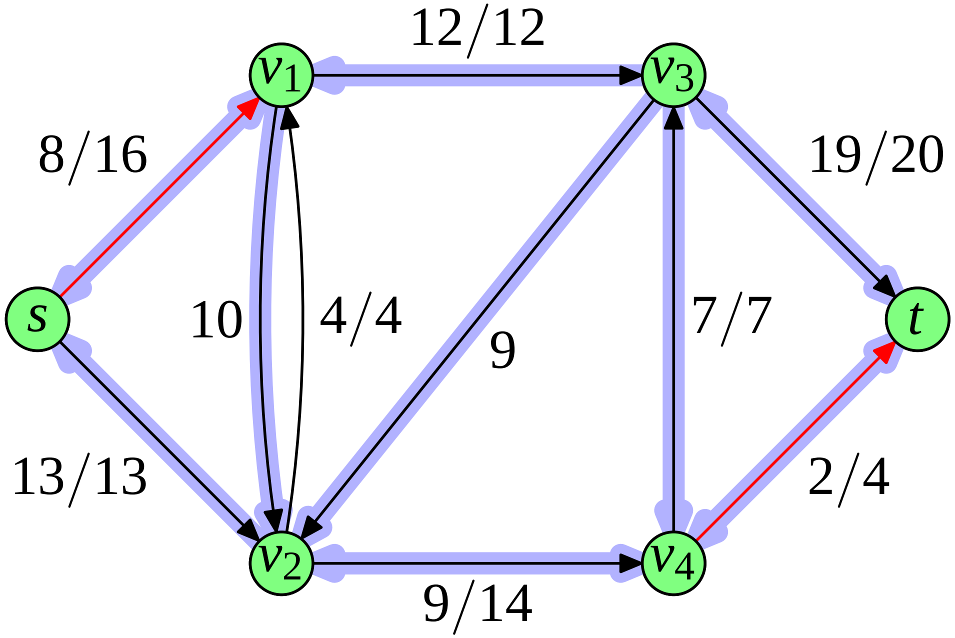
Da es wieder einen augmentierenden Kreis mit negativen Kosten gibt, kann der Fluß noch nicht minimale Kosten haben.

Beispiel



Hat der Fluß jetzt minimale Kosten?

Beispiel



Der Fluß hat minimale Kosten und es gibt keinen Kreis mit negativen Kosten im Residualnetzwerk.

Beweis (1. Teil)

Wenn ein Kreis mit negativen Kosten existiert, dann hat der Fluß nicht minimale Kosten:

Sei K ein Kreis in G_f mit negativen Kosten und Restkapazität $c_f(K)$.

Ändern wir den Fluß indem wir ihn auf jeder Kante von K um $c_f(K)$ erhöhen, dann sinken die Kosten.

$\Rightarrow$ Dies widerspricht der Annahme, daß f minimale Kosten hat.

Andere Richtung:

Angenommen f hat nicht minimale Kosten. Dann existiert ein Fluß f^* mit kleineren Kosten, aber $|f| = |f^*|$.

Sei $\Delta = f^* - f$. Da $|f| = |f^*|$, gilt $\Delta(V, s) = \Delta(V, t) = 0$.

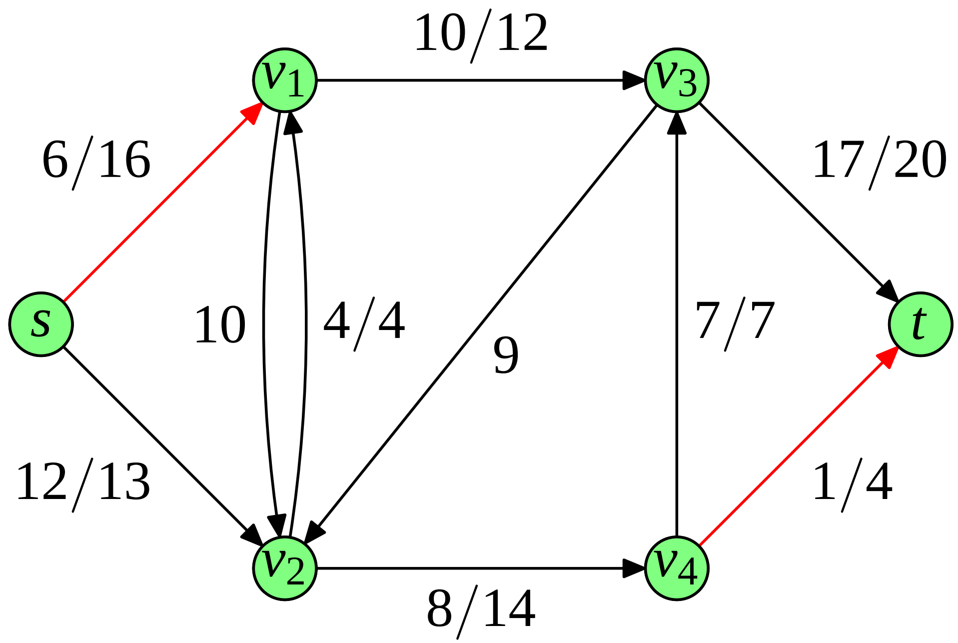
Es gilt also $\Delta(V, u) = 0$ für alle $u \in V$.

$\Rightarrow$ $\Delta(u, v) > 0$ ist eine Vereinigung von Flüssen auf Kreisen.

$\Rightarrow$ Einer der Kreise muß negative Kosten haben

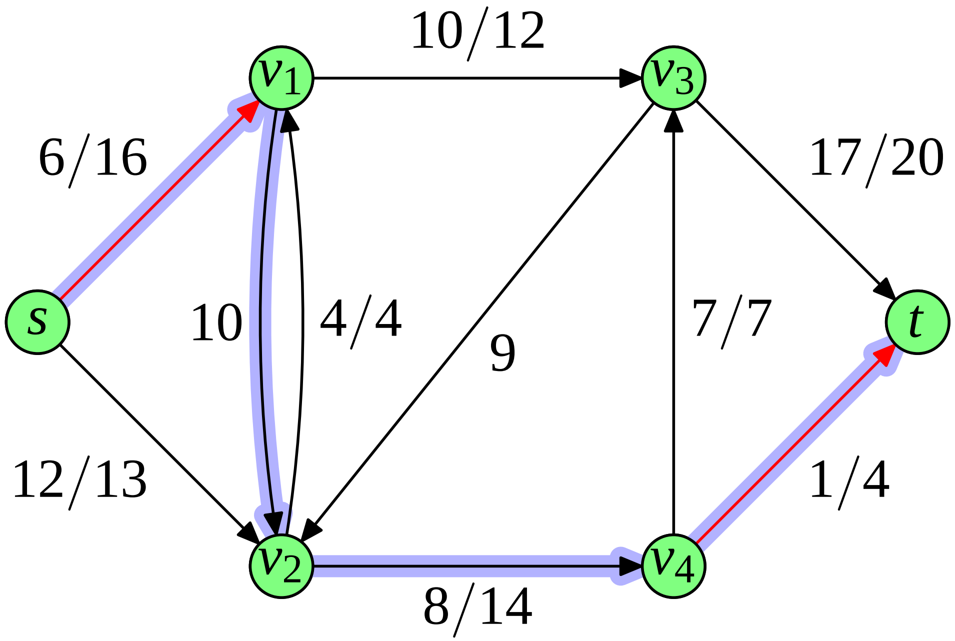
(Damit ist der erste Teil bewiesen.)

Beispiel



Der Fluß hat minimale Kosten. Was passiert, wenn er mit einem augmentierenden Pfad augmentiert wird?

Beispiel



Dieser augmentierende Pfad hat nicht minimale Kosten, denn von s nach v_2 gibt es eine billigere Abkürzung.

Geht man die Abkürzung und dann den übersprungenen Teil rückwärts, erhalten wir einen Kreis mit negativen Kosten.

Beweis (2. Teil)

Damit durch eine Augmentierung ein Kreis im Residualnetzwerk entsteht, muß der augmentierende Pfad durch eine Kante des Kreises rückwärts gehen.

Sei p ein augmentierender Pfad, durch den ein Kreis K mit negativen Kosten entsteht. Die Kante (u, v) liege auf p und die Kante (v, u) liege auf K.

Sei p' folgender Pfad: Von s entlang p bis u, dann entlang K bis v, dann entlang p bis t.

Die Kosten von p' sind dann kleiner als von p. Also kann p kein augmentierender Pfad mit minimalen Kosten sein.

Ein einfacher Algorithmus

Gesucht: Fluß mit Betrag B und minimalen Kosten

```
Initialisiere Fluß f zu 0
while es gibt einen augmentierenden Pfad p do
    finde einen augmentierenden Pfad p
        mit minimalen Kosten
    augmentiere f entlang p mit min{B − |f|, c_f(p)}
return f
```

Durch das Minimum Cost Flow Theorem ist garantiert, daß der gefundene Fluß minimale Kosten hat.

Die Edmonds–Karp–Variante

for each edge $(u, v) \in E$ **do**
  $f(u, v) := 0$
  $f(v, u) := 0$
while there exists a path from s to t in G_f **do**
  $p :=$ a shortest, minimum cost path from s to t in G_f
  $c_f(p) := \min\{\, c_f(u, v) \mid (u, v) \text{ is in } p \,\}$
  for each edge (u, v) in p **do**
    $f(u, v) := f(u, v) + \min\{B - |f|, c_f(p)\}$
    $f(v, u) := -f(u, v)$
return f