29. Push-Relabel Algorithms

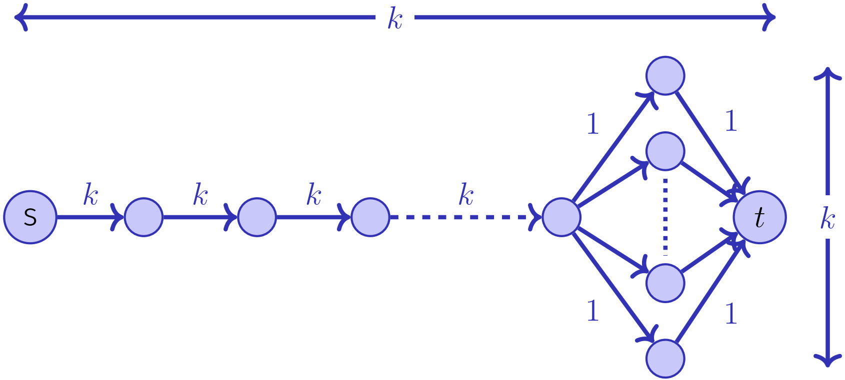
Disclaimer

Diese Folien beinhalten die wichtigsten Formalien zum Push-Relabel Algorithmus und dessen Korrektheit. Es fehlt noch ein Beispiel. In der Vorlesung wird der Algorithmus motiviert und mit Beispielen unterlegt.

Die Konzeption dieser Vorlesung ist übernommen von Tim Roughgarden (Stanford)

`https://www.youtube.com/watch?v=0hI89H39USg`

Beispiel



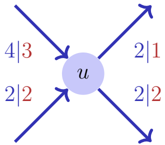
Der Ford-Fulkerson Algorithmus (und Edmonds Karp) führt hier $\Omega(k^2)$ viele Schritte aus.

Pre-Flow

Ein Pre-Flow $f : V \times V \to \mathbb{R}$ ist ein Fluss mit einer abgeschwächten Flusserhaltung:

- **Kapazitätsbeschränkung**:
  Für alle $u, v \in V$: $f(u,v) \le c(u,v)$.
- **Schiefsymmetrie**:
  Für alle $u, v \in V$: $f(u,v) = -f(v,u)$.
- **Abgeschächte Flusserhaltung**:
  Für alle $u \in V \setminus \{s,t\}$:

$$\alpha_f(u) := \sum_{v \in V} f(v,u) \ge 0.$$



Knoten mit Exzess
$\alpha_f(u) = 3 + 2 - 1 - 2 = 2.$

Der Wert $\alpha_f(u)$ heisst **Exzess** von f in u

Algorithmus `Push`(u, v)

Das Erweiterungsnetzwerk $G_f = (V, E_f, c_f)$ ist auf einem Pre-Flow f definiert wie vorher auf einem Fluss.

if $\alpha_f(u) > 0$ **then**
  if $c_f(u, v) > 0$ in G_f **then**
    $\Delta \leftarrow \min\{c_f(u, v), \alpha_f(u)\}$
    $f(u, v) \leftarrow f(u, v) + \Delta$.

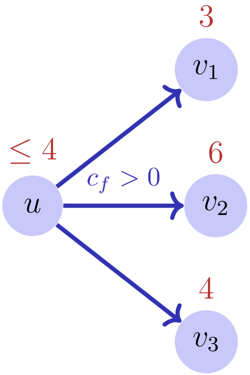
Höhenfunktion

Eine Höhenfunktion $h : V \to \mathbb{N}_0$ auf G wird dafür sorgen, dass der Fluss nicht unendlich oft im Kreis weitergegeben wird. Ausserdem ssorgen die folgenden Invarianten dafür, dass im Erweiterungsnetzwerk s von t getrennt bleibt.

Invarianten der Höhenfunktion

1. $h(s) = n$
2. $h(t) = 0$
3. Für alle $u, v \in V$ mit $c_f(u, v) > 0$ gilt $h(u) \leq h(v) + 1$.

Beispiel



Kanten im Erweiterungsnetzwerk gehen höchstens einen Schritt bergab (oder bleiben gleich oder gehen bergauf).

Kein Erweiterungspfad

Die Länge eines Pfades von s nach t im Erweiterungsnetzwerk ist maximal $n-1$. Da für jede Kante (u,v) mit $c_f(u,v)>0$ gilt, dass $h(v)\geq h(u)-1$ und da $h(s)=n$ und $h(t)=0$ (der Pfad von s nach t müsste also mindestens Länge n haben), kann es **bei Einhalten der Höheninvarianten keinen Erweiterungspfad** geben.

Strategien

Ford-Fulkerson (konservativ)

- Invariante: Flusserhaltung
- Schritte: Erweiterungspfade
- Ziel: Trenne s von t im Erweiterungsnetzwerk G_f.

Push-Relabel

- Invariante: Höheninvariante (kein Erweiterungspfad!)
- Schritte: Push Flow
- Ziel: Stelle Flusserhaltung her.

Push-Relabel-Algorithmus

Input: Flussnetzwerk $G = (V, E, c)$, Quelle s und Senke t. $n := |v|$

$h(s) \leftarrow n$
foreach $v \neq s$ **do** $h(v) \leftarrow 0$
foreach $(u, v) \in E$ **do** $f(u, v) \leftarrow 0$
foreach $(s, v) \in E$ **do** $f(s, v) \leftarrow c(s, v)$

while $\exists u \in V \setminus \{s, t\} : \alpha_f(u) > 0$ **do**
    wähle u mit $\alpha_f(u) > 0$ und maximalem $h(u)$
    if $\exists v \in V : c_f(u, v) > 0 \;\wedge\; h(v) = h(u) - 1$ **then**
        `push`(u, v) // push
    else
        $h(u) \leftarrow h(u) + 1$ // relabel

Korrektheit: Invarianten-Lemma

Lemma 38

Während der Ausführung des Push-Relabel Algorithmus bleiben die Invarianten zur Höhenfunktion erhalten.

Unmittelbare Folgerung: Wenn der Push-Relabel-Algorithmus terminiert, terminiert er mit einem maximalen Fluss

Invarianten-Lemma: Beweis

Beweis:

- Nach der Initialisierung sind die Invarianten erhalten, denn nur für Kanten (s, u) ist die Höhendifferenz kleiner als -1, dort gilt $c_f(s, u) = 0$
- Invarianten für s und t bleiben erhalten, da die Höhe von s und t nicht verändert wird.
- Ausführung von **`push`**(u, v) erzeugt im Restnetzwerk hochstens die Kante (v, u), mit $h(v) > h(u)$
- Ausführung von Relabel findet nur statt, wenn keine Abwärtskante vorhanden ist. Damit gilt danach $h(u) \geq h(v) - 1$ für alle Kanten (u, v)

■

Terminierung und Laufzeit

Theorem 39

Der Push-Relabel Algorithmus terminiert nach

- $\mathcal{O}(n^2)$ *Relabel-Operationen und*
- $\mathcal{O}(n^3)$ *Push-Operationen.*

Der Beweis wird im Folgenden für Relabel und Push-Operationen separat geführt.

Hauptlemma

Lemma 40

Sei f ein Pre-Flow in G. Wenn $\alpha_f(u) > 0$ gilt für einen Knoten $u \in V - \{s,t\}$, dann gibt es einen Pfad $p : u \rightsquigarrow s$ im Erweiterungsnetzwerk G_f

Hauptlemma: Beweis

Beweis: Sei $A := \{u \in V : \exists p : s \rightsquigarrow u$ with $f(e) > 0 \; \forall \; e \in p\}$ und $B := V \setminus A$. Für jedes $u \in A$ gibt es einen Pfad von s mit positiven Fluss. Daher gibt es im Erweiterungsnetzwerk einen Pfad von u nach s.

Sei $u \in B$. Dann $\sum_{v \in V} f(v, u) \geq 0$, denn f ist pre-flow.

Aber auch $\sum_{v \in V} \sum_{u \in B} f(v, u) = \underbrace{\sum_{v \in A} \sum_{u \in B} f(v, u)}_{\leq 0} + \underbrace{\sum_{v \in B} \sum_{u \in B} f(v, u)}_{=0} \leq 0$ denn es kann keine Kante mit positivem Fluss von A nach B geben und für jede Kante innerhalb von B gilt $f(u, v) = -f(v, u)$. $\Rightarrow \alpha_f(u) = 0 \; \forall \; u \in B$. Also impliziert $\alpha_f(u) > 0$ dass $u \in A$.

■

Maximale Knotenhöhe

Corollary 41

Während der Ausführung des Push-Relabel Algorithmus gilt stets $h(u) < 2n$ für alle $u \in V$.

Beweis:

Hauptlemma: für jeden Knoten u mit $\alpha_f(u) > 0$ gibt es Pfad $p : u \rightsquigarrow s$ in Restnetzwerk.

Höheninvarianten: Kanten in G_f gehen maximal einen Schritt abwärts , $h(s) = n$.

Maximale Länge von $p : u \rightsquigarrow s$ (zyklenfrei!) ist $n - 1$. $\Rightarrow$ Maximale Höhe Knoten ist $n + n - 1 = 2n - 1$.

$\blacksquare$

Anzahl Relabels

Aus dem vorherigen Korollar folgt direkt

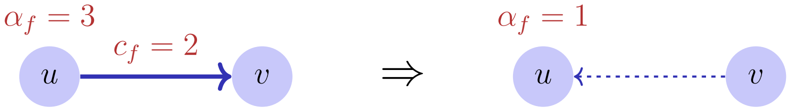
Corollary 42

Der Push-Relabel Algorithmus führt $\mathcal{O}(n^2)$ Relabel-Operationen aus.

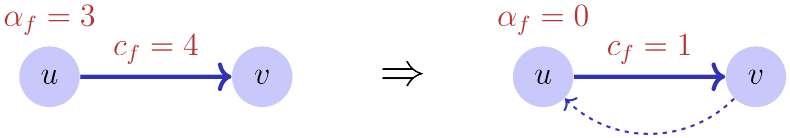
(Nicht) sättigende Push-Operationen

push(u, v) heisst

- **sättigend**, wenn $c_f(u,v) \leq \alpha_f(u)$



- **nicht sättigend**, wenn $c_f(u,v) > \alpha_f(u)$



Anzahl sättigende Push-Operationen

Lemma 43

Zwischen zwei sättigenden Push-Operationen auf derselben Kante (u, v) führt der Push-Relabel Algorithmus auf u und v jeweils mindestens zwei Relabel- Operationen aus

Unmittelbare Folgerung: es finden insgesamt $\mathcal{O}(n^3)$ sättigende Push-Operationen statt, denn für jeden Knoten werden nach Korollar 41 $\mathcal{O}(n)$ Relabel-Operationen ausgeführt.

Beweis: Anzahl sättigende Push-Operationen

Beweis:

Nach sättigendem $\texttt{push}(u, v)$ (mit $h(u) = h(v) + 1$) verschwindet Kante (u, v) vom Restnetzwerk.

Damit Kante (u, v) wieder im Restnetzwerk erscheint, muss $\texttt{push}(v, u)$ (Gegenkante) ausgeführt werden. Dafür muss $h(v) = h(u) + 1$ gelten, also sind mindestens zwei Relabel Operationen auf v nötig.

Nachfolgend sind mindestens zwei Relabel-Operationen auf u nötig, damit $\texttt{push}(u, v)$ ausgeführt werden kann.

■

Anzahl nichtsättigende Push-Operationen

Lemma 44

Zwischen zwei Relabel-Operationen führt der Push-Relabel Algorithmus maximal n nicht-sättigende Push-Operationen aus.

Unmittelbare Folgerung: es finden insgesamt $\mathcal{O}(n^3)$ nichtsättigende Push-Operationen statt, denn nach Korollar 42 finden insgesamt $\mathcal{O}(n^2)$ Relabel-Operationen statt

Beweis: Anzahl nichtsättigende Push-Operationen

Beweis:

Sei $A_f := \{v \in V : \alpha_f(v) > 0\}$

Wahl von u bei push: $u \in A_f$ mit $h(u) \geq h(v)$ für alle $v \in A_f$.

Bei nichstättigender push-Operation verschwindet u von A_f. Bei diesem und jedem weiteren Push kommen höchstens $v \in A_f$ hinzu mit $h(v) < h(u)$. Bis eine erneute Relabel-Operation stattgefunden hat, gilt also stets $u \notin A_f$.

Dieses Argument gilt für jedes gewählte u und es können bis zur nächster Relabel-Operation höchstens n nichtsättigende Push-Operationen ausgeführt werden.

■