29. Push-Relabel Algorithmus

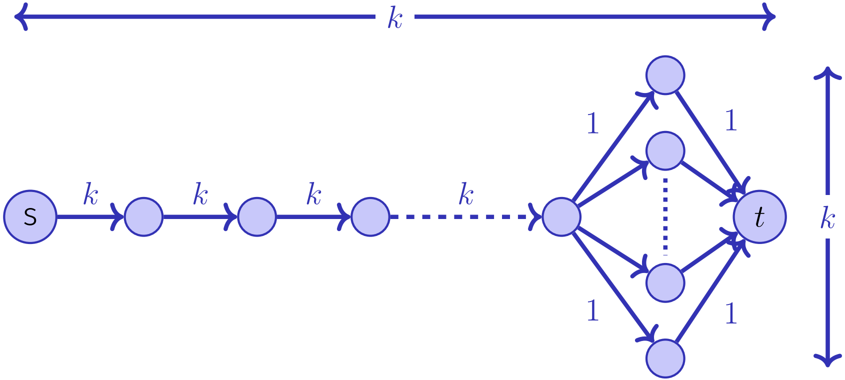
Disclaimer

These slides contain the most important formalities around the Push-Relabel algorithm and its correctness. One example is still missing. We motivate the algorithm in the lectures and give more examples there.

The conception of this lecture taken from Tim Roughgarden (Stanford)

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

Beispiel



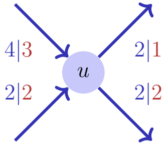
Here, the Ford-Fulkerson algorithm (and Edmonds-Karp) executes $\Omega(k^2)$ steps.

Pre-Flow

A pre-flow $f : V \times V \to \mathbb{R}$ is a flow with a relaxed flow conservation condition:

- **Bounded Capacity**:
  For all $u, v \in V$: $f(u,v) \le c(u,v)$.
- **Skew Symmetry**:
  For all $u, v \in V$: $f(u,v) = -f(v,u)$.
- **Relaxed flow condition**:
  For all $u \in V \setminus \{s,t\}$:

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



node with excess
$\alpha_f(u) = 3 + 2 - 1 - 2 = 2.$

The quantitiy $\alpha_f(u)$ is called **excess** of f at u

Algorithmus $\texttt{Push}(u, v)$

The residual network G_f remains defined for a pre-flow as before for a flow.

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

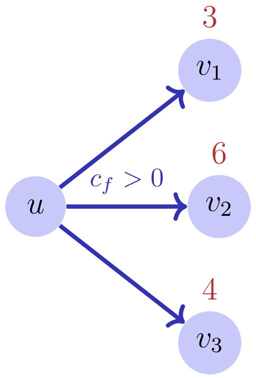
Height Function

A height function $hV \rightarrow \mathbb{N}_0$ on G will make sure that the flow is not pushed infinitely often in circles. Moreover, the following invariants makes sure that s keeps being disconnected from t in the residual network.

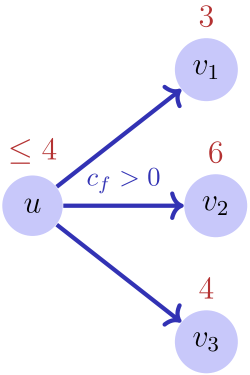
Invariants of the height function

1. $h(s) = n$
2. $h(t) = 0$
3. for each $u, v \in V$ with $c_f(u, v) > 0$ it holds that $h(u) \leq h(v) + 1$.

Beispiel



Beispiel



Edges in the residual network go at most down by one (or stay on the same height or go up)

No Augmenting Path

The length of a path from s to t in the residual network is at most $n-1$. Because for each edge (u,v) with $c_f(u,v)>0$ it holds that $h(v)\geq h(u)-1$ and since $h(s)=n$ and $h(t)=0$ (thus a path from height n to height 0 requires at least n steps), **no augmenting path exists when the invariants are preserved**.

Strategies

Ford-Fulkerson (conservative)

- Invariant: flow conservation
- Steps: augmenting paths
- Goal: separate s from t in the residual network.

Push-Relabel

- Invariant: height invariant (no augmenting path!)
- Steps: push flow
- Goal: achieve flow conservation

Push-Relabel-Algorithmus

Input: Flow graph $G = (V, E, c)$, with source s and sink 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**
    choose u with $\alpha_f(u) > 0$ and maximal $h(u)$
    if $\exists v \in V : c_f(u, v) > 0 \;\wedge\; h(v) = h(u) - 1$ **then**
        $\mathtt{push}(u, v)$ // push
    else
        $h(u) \leftarrow h(u) + 1$ // relabel

Correctness: Invariants Lemma

Lemma 38

During the execution of the Push-Relabel algorithm, the invariants for the height functions are preserved

Immediate conclusion: when the Push-Relabel algorithm terminates, it terminates with a max-flow.

Invariants-Lemma: Proof

Proof:

After initialization, the invariants are fulfilled because only for edges (s, u) the height difference less than -1, but there we have $c_f(s, u) = 0$

Invariants on s and t are preserved because the height of s and t is never changed.

Execution of $\texttt{push}(u, v)$ can at most yield a new edge (v, u) in the residual network with $h(v) > h(u)$

Execution of relabel takes place only when there is no downward edge. Thus after a relabel it holds that $h(u) \geq h(v) - 1$ for all edges (u, v)

■

Termination and Running Time

Theorem 39

The Push-Relabel algorithm terminates after

- $\mathcal{O}(n^2)$ *relabel operations, and*
- $\mathcal{O}(n^3)$ *push operations.*

The proof is conducted in the following separately for relabel and push.

Key Lemma

Lemma 40

Let f be a pre-flow in G If $\alpha_f(u) > 0$ holds for some node $u \in V - \{s, t\}$, then there is some path $p : u \rightsquigarrow s$ in the residual network G_f

Key Lemma: Proof

Proof: Let $A := \{u \in V : \exists p : s \rightsquigarrow u$ mit $f(e) > 0 \; \forall \; e \in p\}$ and $B := V \setminus A$. For each $u \in A$ there is a path from s with positive flow. Therefore in the residual network there is a path from u to s.

Let $u \in B$. Then $\sum_{v \in V} f(v,u) \geq 0$, because f is a pre-flow.

But also $\sum_{v \in V} \sum_{u \in B} f(v,u) = \underbrace{\sum_{v \in A} \sum_{u \in B} f(v,u)}_{\leq 0} + \underbrace{\sum_{v \in B} \sum_{u \in B} f(v,u)}_{=0} \leq 0$ because there cannot be an edge with postiive weight from A to B and for each edge within B it holds that $f(u,v) = -f(v,u)$. $\Rightarrow \alpha_f(u) = 0 \; \forall \; u \in B$. Thus $\alpha_f(u) > 0$ implies that $u \in A$.

■

Maximum Node Height

Corollary 41

During the execution of the Push-Relabel algorithm it holds that $h(u) < 2n$ for all $u \in V$.

Proof:

Mainlemma: for each node t with $\alpha_f(u) > 0$ there is a path $p : u \leadsto s$ in residual network

Height invariants: edges in G_f go down by at most one step. , $h(s) = n$.

Maximal length of $p : u \leadsto s$ (no cycles!) is $n - 1$. $\Rightarrow$ Maximum height of node is $n + n - 1 = 2n - 1$.

■

Number Relabels

From the previous corollary immediately follows

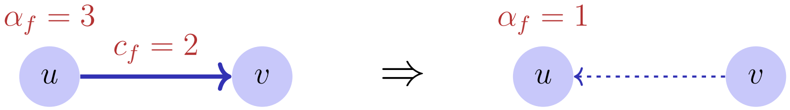
Corollary 42

The Push-Relabel algorithm executes $\mathcal{O}(n^2$ *relabel operations.*

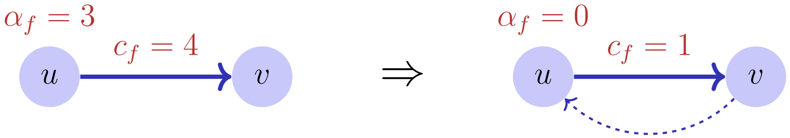
(Non-)Saturating Pushes

push(u, v) is called

- **saturating**, if $c_f(u, v) \leq \alpha_f(u)$



- **non-saturating**, if $c_f(u, v) > \alpha_f(u)$



Number Saturating Pushes

Lemma 43

Between two non-saturing pushes an the same edge (u, v), the Push-Relabel algorithm executes at least two relabel operations.

Immediate conlusion: there are $\mathcal{O}(n^3)$ saturating push operations overal because for each node by corollary 41 there are at $\mathcal{O}(n)$ relabels.

Proof: Number Saturating Pushes

Proof:

After a saturing **push**(u, v) (with $h(u) = h(v) + 1$) edge (u, v) disappears from the residual network.

In order to (u, v) to reappear on the residual network, **push**(v, u) (reverse edge) has to be executed. But before it must hold that $h(v) = h(u) + 1$ therefore to relabels of v are required.

Two more relabels are required on u before a call to **push**$(u, v")$

■

Number Non-Saturating pushes

Lemma 44

Between two relabel-operations, the Push-Relabel algorithm executes at most n non-saturating pushes.

Immediate conlusion: there are $\mathcal{O}(n^3)$ non-saturating push operations overal because by corollary 42 there are $\mathcal{O}(n^2)$ relabel operations.

Proof: Number Non-saturating pushes

Proof:

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

Choice of u for push: $u \in A_f$ with $h(u) \geq h(v)$ for all $v \in A_f$.

During a non-saturating push u disappears from A_f. During this push and following pushes only $v \in A_f$ with $h(v) < h(u)$ are added to A_f- Before a new relabel has been executed, it holds thus that $u \notin A_f$.

Because this argument holds for all chosen u, until the next relabel operation at most n non-saturating pushes can be executed.

■