Maximum flow i netværk. Del 2.

Martin Paluszewski
Ph.D. studerende
palu@diku.dk

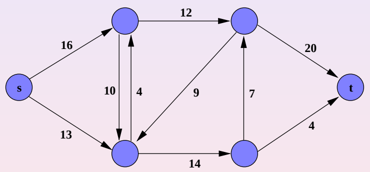
DIKU

28/11-2006

Emner

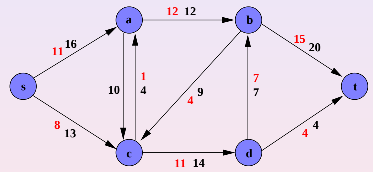
- Opfriskning af flownetværk og maximum flow.
- Opfriskning af Ford-Fulkerson. $\mathcal{O}(E|f^*|)$.
- Ford-Fulkerson -> Edmonds-Karp algoritme. $\mathcal{O}(VE^2)$.
- Maximum bipartite matching (med maximum flow). $\mathcal{O}(VE)$.
- Push-relabel algoritme. $\mathcal{O}(V^2E)$.

Flownetværk



- G=(V,E) er en orienteret og sammenhængende graf.
- Der er præcis en source s og en sink t.
- Alle kanter $(u, v) \in E$ har en ikke-negativ kapacitet.

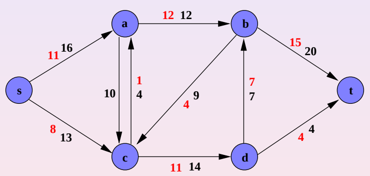
Et flow



Et flow i G er en funktion $f : V \times V \rightarrow \mathcal{R}$, der opfylder følgende:

- **Kapacitet:** For alle $u, v \in V$, skal $f(u, v) \leq c(u, v)$.
- **Skævsymmetri:** For alle $u, v \in V$ skal $f(u, v) = -f(v, u)$.
- **Flowbevarelse:** For alle $u \in V - \{s, t\}$ skal $\sum_{v \in V} f(u, v) = 0$.

Maximum flow



- **Værdien** af et flow er defineret som: $|f| = \sum_{v \in V} f(s, v)$
- **Maximum flow:** Givet et flownetværk. Find et gyldigt flow med den største flowværdi $|f|$.

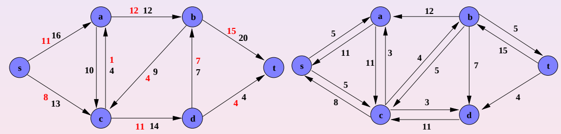
Ford-Fulkersons algoritme

```
FORD-FULKERSON(G, s, t)
for each edge (u, v) ∈ E[G] do
    f[u, v] ← 0
    f[v, u] ← 0
end for
while there exists a path p from s to t in the residual network G_f do
    c_f(p) ← min{ c_f(u, v) : ((u, v) is in p}
    for each edge (u, v) ∈ p do
        f[u, v] ← f[u, v] + c_f(p)
        f[v, u] ← −f[u, v]
    end for
end while
```

Ford-Fulkersons algoritme
residualnetværk

- Givet et flownetværk $G = (V, E)$ og et flow f
- Residualnetværket $G_f = (V, E_f)$, hvor
- Kapaciteten af kanterne er $c_f(u, v) = c(u, v) - f(u, v)$
- $E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}$

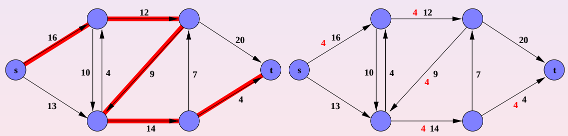
Ford-Fulkersons algoritme
Eksempel på residualnetværk



Flownetværk Residualnetværk

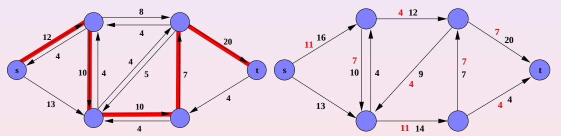
- $c_f(u, v) = c(u, v) - f(u, v)$
- $E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}$

Ford-Fulkersons algoritme
Eksempel på Ford-Fulkersons algoritme



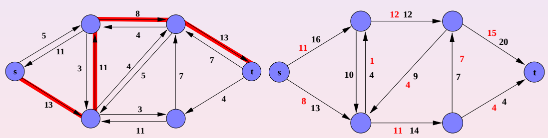
Residualnetværk Flownetværk

Ford-Fulkersons algoritme
Eksempel på Ford-Fulkersons algoritme



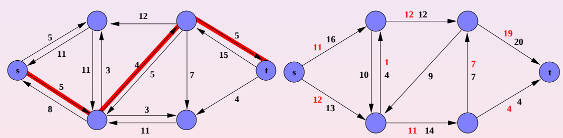
Residualnetværk Flownetværk

Ford-Fulkersons algoritme
Eksempel på Ford-Fulkersons algoritme



Residualnetværk Flownetværk

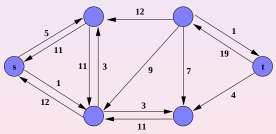
Ford-Fulkersons algoritme
Eksempel på Ford-Fulkersons algoritme

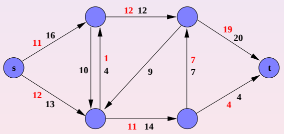


Residualnetværk Flownetværk

Ford-Fulkersons algoritme

Eksempel på Ford-Fulkersons algoritme





Residualnetværk   Flownetværk

Edmonds-Karps algoritme

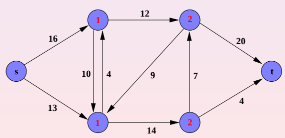
Edmonds-Karps algoritme: Vejen i residualnetværket findes ved bredde-først søgning. Resten som i Ford-Fulkersons algoritme.

Køretider:

- Ford-Fulkerson: $\mathcal{O}(E|f^*|)$
- Edmonds-Karp: $\mathcal{O}(VE^2)$

Theorem

Hvis Edmonds-Karps algoritme køres på et flownetværk $G = (V, E)$ med source s og sink t, så gælder, for alle knuder $v \in V - \{s, t\}$, at den korteste vej $\delta(s, v)$ i residualnetværket G_f forøges monotont ved hver flowforøgelse.



Bevis.

- Antag at der findes en knude $v \in V - \{s, t\}$, hvor flowforøgelsen **reducerer** $\delta(s, v)$, dvs.

$$\delta_{f'}(s, v) < \delta_f(s, v) \qquad (1)$$

- Lad $p = s \rightsquigarrow u \to v$ være den korteste vej fra s til v i $G_{f'}$. Antag at v er den første knude på p, hvor (1) gælder. Vi har derfor:

$$\begin{aligned} \delta_{f'}(s, u) &= \delta_{f'}(s, v) - 1 \quad (2) \\ \delta_{f'}(s, u) &\geq \delta_f(s, u) \quad (3) \end{aligned}$$

Bevis. (fortsat)

$$\begin{aligned} \delta_{f'}(s, u) &= \delta_{f'}(s, v) - 1 \quad (2) \\ \delta_{f'}(s, u) &\geq \delta_f(s, u) \quad (3) \end{aligned}$$

Det gælder at:

$$(u, v) \notin E_f$$

Antag at $(u, v) \in E_f$

$$\begin{aligned} \delta_f(s, v) &\leq \delta_f(s, u) + 1 \quad (\textit{trekantsulighed}) \\ &\leq \delta_{f'}(s, u) + 1 \quad \text{pga. (3)} \\ &= \delta_{f'}(s, v) \quad \text{pga. (2)} \end{aligned}$$

Bevis. (fortsat)

- Vi har $(u, v) \notin E_f$ og $(u, v) \in E_{f'}$.
- **Spørgsmål:** Hvorfor oprettes kant (u,v) i residualnetværket?
- **Svar:** Der har været en flowforøgelse fra v til u

Edmonds-Karps algoritme forøger altid flowet gennem en korteste vej. Så:

$$\begin{aligned} \delta_f(s, v) &= \delta_f(s, u) - 1 \\ &\leq \delta_{f'}(s, u) - 1 \quad \text{pga. (3)} \\ &= \delta_{f'}(s, v) - 2 \quad \text{pga. (2)} \end{aligned}$$

Dette er i modstrid med antagelsen:

$$\delta_{f'}(s, v) < \delta_f(s, v) \quad \square$$

Theorem

Antallet af flowforøgelser i Edmonds-Karp algoritme er $\mathcal{O}(VE)$.

Bevis

- En kant (u,v) i residualnetværket er **kritisk** på en vej p, hvis $c_f(p) = c_f(u, v)$.
- Der er altid mindst 1 kritisk kant.
- Efter en flowforøgelse forsvinder de(n) kritiske kant(er) fra residualnetværket.
- Det vises nu at en kant kan blive kritisk højst $\mathcal{O}(V)$ gange.

Bevis (fortsat)

- Det vises nu at en kant kan blive kritisk højst $\mathcal{O}(V)$ gange.
- Når en kant (u,v) er kritisk første gang gælder:

$$\delta_f(s, v) = \delta_f(s, u) + 1 \qquad (1)$$

- Herefter forsvinder den fra residualnetværket. Den bliver først indsat igen hvis flowet (f') langs (v,u) forøges. Da gælder:

$$\begin{aligned} \delta_{f'}(s, u) &= \delta_{f'}(s, v) + 1 \\ &\geq \delta_f(s, v) + 1 \quad \text{(Her bruges thm 1.)} \\ &= \delta_f(s, u) + 2 \quad \text{pga. (1)} \end{aligned}$$

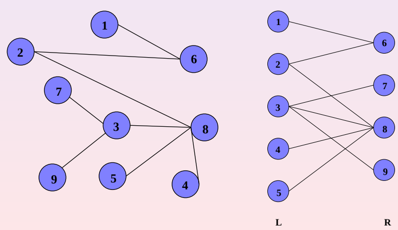
Bevis (fortsat)

$$\delta_{f'}(s, u) = \delta_f(s, u) + 2$$

- Den korteste vej fra s til u forøges altså med mindst 2 hver gang (u,v) bliver kritisk.
- Den korteste vej fra s til u kan højst besøge V knuder.
- Så (u,v) kan blive kritisk højst $\mathcal{O}(V)$ gange.
- I hver flowforøgelse bliver altid mindst 1 kant kritisk, så antallet er flowforøgelser er $\mathcal{O}(EV)$ $\square$.
- Hver iteration af Edmonds-Karps algoritme tager $\mathcal{O}(E)$ tid. Så køretiden er: $\mathcal{O}(VE^2)$

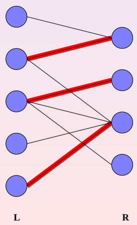
Todelt graf

En graf $G = (V, E)$ er **todelt**, hvis knuderne kan inddeles i to dele $V = L \cup R$ og alle kanter i E går mellem L og R.



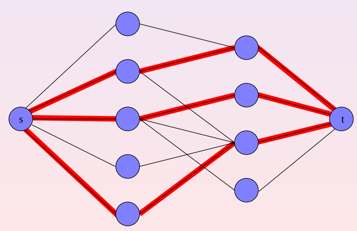
Maximum matching i en todelt graf

En **matching** er en delmængde af kanter $M \subseteq E$, hvor hver knude $v \in V$ højst er forbundet til 1 kant i M. En **maximum matching** er en matching med flest kanter.



Ford-Fulkerson

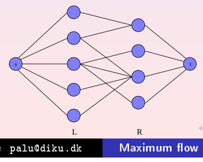
Man kan bruge Ford-Fulkersons algoritme to at finde en maximum matching



Todelt graf til flownetværk

Lad $G = (V, E)$ være en ikke-orienteret todelt graf, og lad $G' = (V', E')$ være det tilsvarende flownetværk.

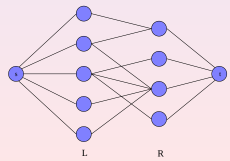
$$\begin{aligned} V' &= V \cup \{s, t\} \\ E' &= \{(s, u) : u \in L\} \\ &\cup \{(u, v) : u \in L, v \in R, \text{ og } (u, v) \in E\} \\ &\cup \{(v, t) : v \in R\} \\ c(u, v) &= 1 \quad (u, v) \in E' \end{aligned}$$



Antallet af kanter

E er kanterne i den **todelte** graf. **E'** er kanterne i det nye **flownetværk**.

$$\begin{aligned} |E| &\geq |V|/2 \\ |E'| &= |E| + |V| \\ |E| \leq |E'| &\leq 3|E| \\ |E'| &= \Theta(E) \end{aligned}$$



Lemma 26.10

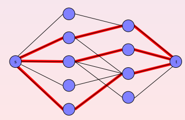
- Lad G=(V,E) være en todelt graf med $V = L \cup R$ og lad $G' = (V', E')$ være det tilsvarende flownetværk.
- Hvis **M** er en matching i G, så findes der et heltalligt flow **f** i G' med værdien |f| = |M|.
- Hvis **f** er et heltalligt flow i G', så findes der en matching **M** i G med |M| = |f|.

Lemma 26.10, del 1

- Hvis **M** er en matching i G så findes der et heltalligt flow **f** i G' med $|f| = |M|$.

Bevis

- Hvis $(u, v) \in M$ så lad $f(s, u) = f(u, v) = f(v, t) = 1$,
- lad $f(u, s) = f(v, u) = f(t, v) = -1$,
- lad $f(u, v) = 0$ for alle andre kanter.

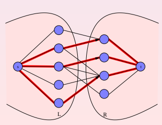


Lemma 26.10, del 1

- Hvis **M** er en matching i G så findes der et heltalligt flow **f** i G' med $|f| = |M|$.

Bevis (fortsat)

- f overholder kapacitet, skævsymmetri og flowbevarelse.
- Flowet over **snittet** $(L \cup \{s\}, R \cup \{t\})$ er |M|.
- Ifølge lemma (26.5) har flowet over snittet samme værdi som flowet i netværket. Så |f| = |M|.



Lemma 26.10

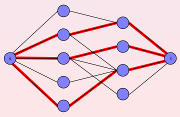
- ~~Hvis **M** er en matching i G så findes der et heltalligt flow **f** i G' med |f| = |M|.~~
- Hvis **f** er et heltalligt flow i G', så findes der en matching **M** i G med |M| = |f|.

Lemma 26.10, del 2

- Hvis **f** er et heltalligt flow i G', så findes der en matching **M** i G med $|M| = |f|$.

Bevis

- Lad $M = \{(u, v) : u \in L, v \in R, \quad og\ f(u, v) > 0.$
- M er en matching pga. flowbevarelse og heltallig flow.
- $|M| = f(L, R) = f(L \cup s, R \cup t) = |f|$



Theorem

- Hvis **M** er en matching i G så findes der et heltalligt flow **f** i G' med |f| = |M|.
- Hvis **f** er et **heltalligt** flow i G', så findes der en matching **M** i G med |M| = |f|.

- Kan vi altid være sikre på at Ford-Fulkerson returnerer et **heltalligt** flow, når kapaciterne er heltallige?
- Ja. Øvelse 26.3-2. (induktionsbevis).

Korollar

Kardinaliteten af en maximum matching **M** i en todelt graf G er værdien af et maximum flow **f** i det tilsvarende flownetværk G'.

Bevis

- Antag at M er en maximum matching i G og det tilsvarende flow f i G' **ikke** er maximum flow.
- Kald maximum flow i G' **f'** ($|f'| > |f|$)
- f' svarer til en matching M' i G hvor $|M'| = |f'|$
- Så $|M'| > |M|$ (modstrid) $\square$

Køretiden

- En matching har højst $\min(|L|, |R|) = \mathcal{O}(V)$ kanter
- Værdien af et maximum flow i G' er derfor $|f^*| = |M^*| = \mathcal{O}(V)$
- Køretiden for Ford-Fulkerson er $\mathcal{O}(E'|f^*|)$
- Køretiden for at finde maximum matching med Ford-Fulkerson er $\mathcal{O}(VE)$, da $|E'| = \Theta(E)$

Push relabel algoritme

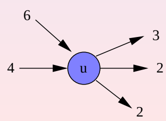
- Hurtigste algoritme både teoretisk og i praksis
- **Push relabel** $\mathcal{O}(V^2E)$.
- **Edmonds-Karps algoritme** $\mathcal{O}(VE^2)$

Begreber

Preflow

- Overholder kapacitet og skævsymmetri men **ikke** nødvendigvis flowbevarelse.

$$f : V \times V \to \mathcal{R}$$
$$f(V, u) \geq 0$$

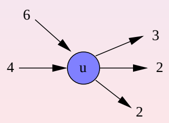


Begreber

Flowoverskud (excess flow)

$$e(u) = f(V, u)$$

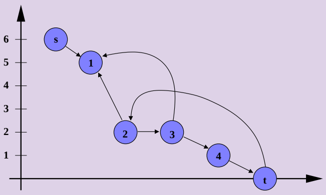
En knude **flyder over** hvis $e(u) > 0$.



Højden af en knude

$h : V \to \mathcal{N}$ er en **højdefunktion** hvis:

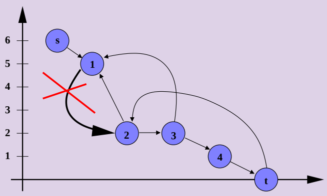
- h(s) = |V|
- h(t) = 0



Højden af en knude

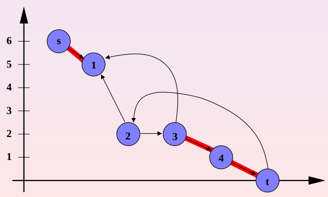
$h : V \to \mathcal{N}$ er en **højdefunktion** hvis:

- h(s) = |V|
- h(t) = 0
- $h(u) \leq h(v) + 1$ for alle kanter $(u, v) \in E_f$



Push operation

- Kan anvendes på kanter $(u, v) \in E_f$, hvor u flyder over, og hvor $h(u) = h(v) + 1$.
- **push(u,v)** skubber/forøger flowet fra u til v med $\min(e(u), c_f(u, v))$.



Push operation

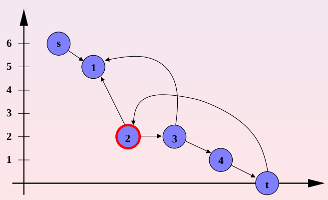
- Kan anvendes på **kanter** $(u, v) \in E_f$ der flyder over, og hvor $h(u) = h(v) + 1$.
- **push(u,v)** skubber/forøger flowet fra u til v med $\min(e(u), c_f(u, v))$.

Push pseudokode

1: $d_f(u, v) \leftarrow \min(e[u], c_f(u, v))$
2: $f(u, v) \leftarrow f[u, v] + d_f(u, v)$
3: $f[v, u] \leftarrow -f[u, v]$
4: $e[u] \leftarrow e[u] - d_f(u, v)$
5: $e[v] \leftarrow e[v] + d_f(u, v)$

Relabel operation

- Kan anvendes på **knuder** u der flyder over, og hvis $h[u] \leq h[v]$ for alle $(u, v) \in E_f$.
- Relabel forøger højden af u.
- $h[u] \leftarrow 1 + \min\{h[v] : (u, v) \in E_f\}$



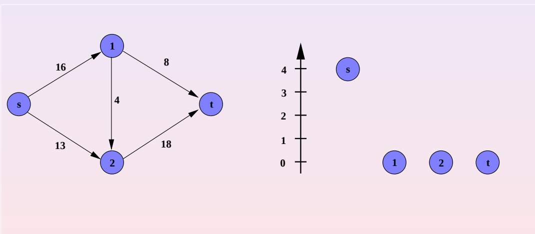
Initialisering pseudokode

for each vertex $u \in V[G]$ **do**
  $h[u] \leftarrow 0$
  $e[u] \leftarrow 0$
end for
for each edge $(u,v) \in E[G]$ **do**
  $f[u,v] \leftarrow 0$
  $f[v,u] \leftarrow 0$
end for
$h[s] \leftarrow |V[G]|$
for each vertex $u \in \mathrm{Adj}[s]$ **do**
  $f[s,u] \leftarrow c(s,u)$
  $f[u,s] \leftarrow -c(s,u)$
  $e[u] \leftarrow c(s,u)$
  $e[s] \leftarrow e[s] - c(s,u)$
end for

Push relabel pseudokode

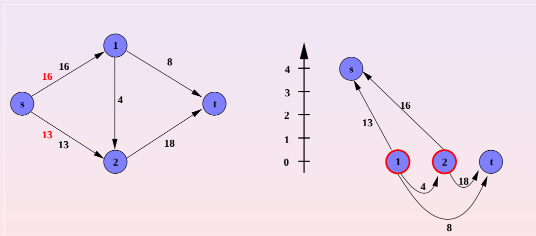
INITIALIZE
while there **exists** an applicable push or relabel operation **do**
  Select an applicable push or relabel operation and perform it
end while

Eksempel



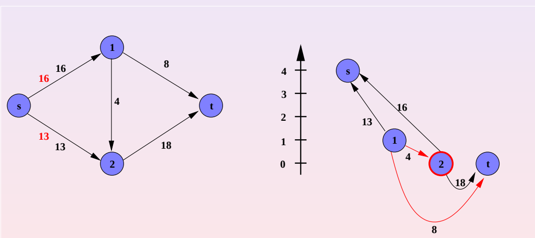
Flownetværk Residualnetværk + højde

Eksempel



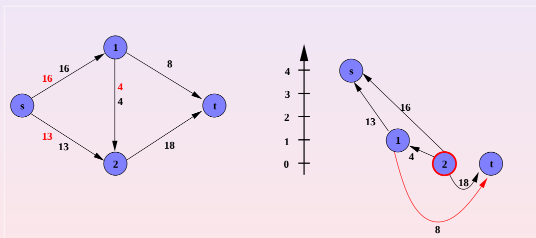
Flownetværk Residualnetværk + højde

Eksempel



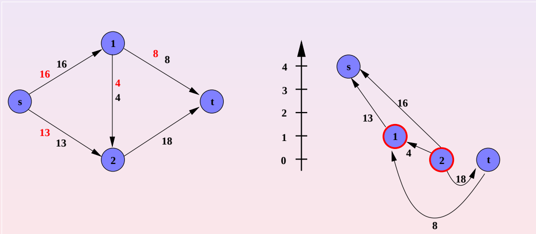
Flownetværk Residualnetværk + højde

Eksempel



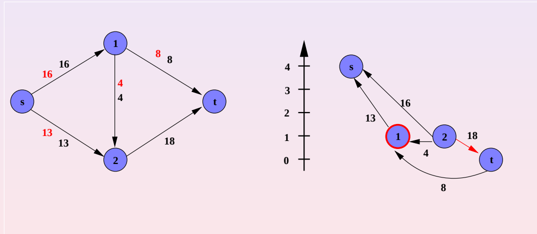
Flownetværk Residualnetværk + højde

Eksempel



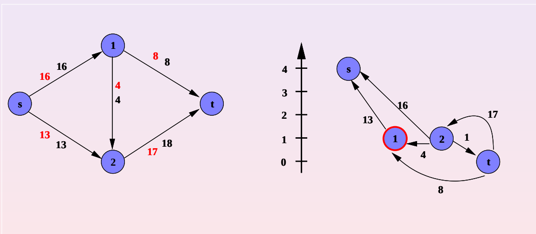
Flownetværk Residualnetværk + højde

Eksempel



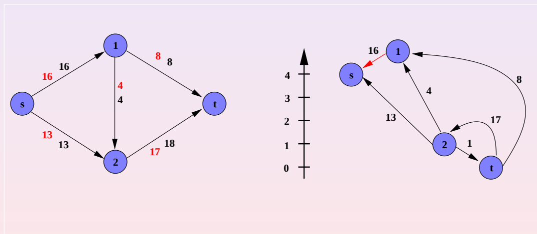
Flownetværk Residualnetværk + højde

Eksempel



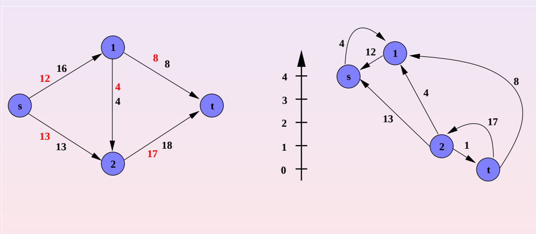
Flownetværk Residualnetværk + højde

Eksempel



Flownetværk Residualnetværk + højde

Eksempel



Flownetværk Residualnetværk + højde

Ikke pensum

Spørgsmål

- Vil algoritmen terminere?
- Hvad er køretiden?
- Har vi et maximum flow når/hvis algoritmen terminerer?

Ikke pensum

Lemma 26.18

Lad $G = (V, E)$ være et flownetværk med source s og sink t. Lad f være et preflow i G, og lad h være en højdefunktion af V. Så er der ingen vej fra s til t i residualnetværket G_f.

Ikke pensum

Lemma 26.18

Lad $G = (V, E)$ være et flownetværk med source s og sink t. Lad f være et preflow i G, og lad h være en højdefunktion af V. Så er der ingen vej fra s til t i residualnetværket G_f.

Bevis

- Antag at der er en simpel vej $p =< v_0, v_1, \cdots, v_k >$ fra s til t i G_f, hvor $v_0 = s$ og $v_k = t$. Så $k < |V|$
- Da h er en højdefunktion gælder $h(v_i) \leq h(v_{i+1}) + 1$ for $i = 0, 1, \cdots, k - 1$
- Dette giver $h(s) \leq h(t) + k$
- Men $h(t) = 0$, så $h(s) \leq k < |V|$
- Dette er i modstrid med kravet om at $h(s) = |V|$

Ikke pensum

Theorem 26.19 (korrekthedsbevis)

Hvis PUSH-RELABEL terminerer på $G = (V, E)$ så er preflowet f et maximum flow i G.

Bevis

- f er et **preflow** efter initialisering.
- f er et **preflow** efter hver push og relabel operation.
- Når algoritmen terminerer flyder ingen knuder over. (lemma 26.15).
- Så f er et flow.
- Ifølge lemma (26.18) er der ingen vej fra s til t i G_f.
- Så ifølge max-flow min-cut thm. 26.7 er flowet f et maximum flow.