Föreläsning 13 i ADK

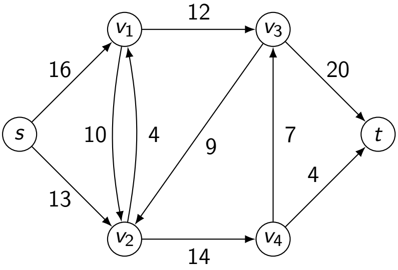
Grafer: maximala flöden

Viggo Kann

KTH

Maximalt flöde i graf

- **Indata:** Digraf G med kantvikter $c(u,v) \geqslant 0$, två speciella hörn i G: källan s, utloppet t
- **Utdata:** Ett maximalt flöde genom grafen från s till t så att högst $c(u,v)$ flödar genom kanten (u,v) för varje kant
- **Exempel:**



Maximalt flöde i graf

Algoritmidé: (Ford-Fulkerson) [8 i Goodrich, 7.1 - 7.3 i Kleinberg, 18 (förr 11) i Biggs, 27.2 i CLR, 33 i Sedgewick, 13.3 i Grimaldi]

Starta med flöde 0
while $\exists$ stig från s till t längs vilken flödet kan öka **do**
    Öka flödet längs denna stig så mycket det går

Tidskomplexitet: $\mathcal{O}(|V|^3)$ (för bästa implementationen)

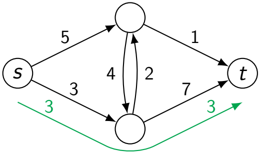
Sats

Om $c(u,v) \in \mathbb{N}$ (heltalskapaciteter) så producerar algoritmen ett maximalt flöde med heltalsflöden i varje kant

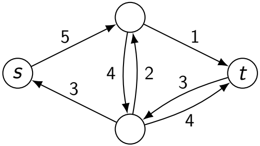
Restflödesgrafen

Ett enkelt sätt att hitta stigar som ökar flödet är att använda restflödesgrafen G_f som har samma hörn som G och en kant från u till v om flödet lokalt från u till v kan ökas; kantens kapacitet, *restkapaciteten*, är $c_f(u,v) = c(u,v) - f(u,v)$, där $f(u,v) = -f(v,u)$.

Exempel: Det utritade flödet 3 i grafen:



Ger restflödesgrafen:



Edmonds-Karp algoritm för flöde

Ford-Fulkersons metod där den stig som har minst antal kanter alltid väljs.

Implementation:

- Hitta kortaste stigen i restflödesgrafen med hjälp av breddenförstsökning från s.

Komplexitetsanalys:

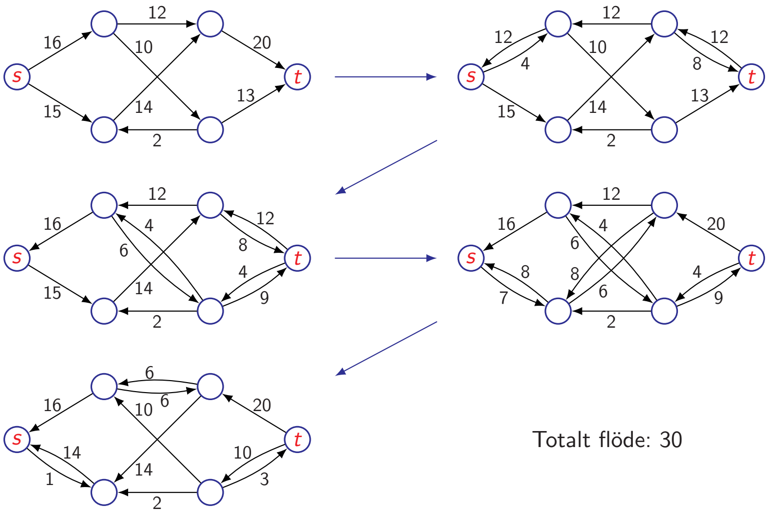
- Att hitta kortaste stigen tar tid $\mathcal{O}(|E|)$
- Att uppdatera flödet längs stigen tar tid $\mathcal{O}(|V|)$

Lemma (beviset ingår inte i kursen)

Om den kortaste stigen väljs i Ford-Fulkersons metod hittas det maximala flödet efter högst $|V||E|$ varv i slingan.

- Total tidskomplexitet: $\mathcal{O}(|V||E| \cdot |E|) = \mathcal{O}(|V||E|^2)$

Flödesexempel



Totalt flöde: 30

Maximalt flöde = minimalt snitt

- Givet ett snitt $(S, V - S)$ i flödesgrafen så att $s \in S$ och $t \in V - S$
- Låt $c(S, V - S)$ vara summan av vikterna på dom kanter som korsar snittet från S till $V - S$

Idé: Flödet är högst $c(S, V - S)$ för alla snitt

Sats

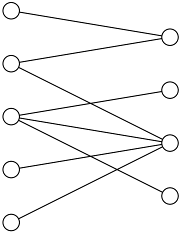
Det maximala flödet $= \min c(S, V - S)$

Bevis:

- Låt f vara ett maximalt flöde
- Då finns det ingen stig från s till t i restflödesgrafen G_f längst vilken flödet kan öka
- Låt $S = \{v \in V : \exists \text{ stig från } s \text{ till } v \text{ i } G_f\}$
- $(S, V - S)$ är ett snitt
- För varje kant (u,v) som korsar snittet från S till $V - S$ gäller $f(u,v) = c(u,v)$
- $\Rightarrow$ flödet $= c(S, V - S)$

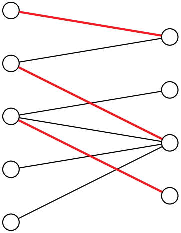
Bipartit matchning

- **Indata:** Bipartit graf $\langle U \cup V, E \rangle$
- **Utdata:** En matchning $M \subseteq E$ av maximal storlek
- M är en matchning om inga kanter i M har någon gemensam ändpunkt
- Exempel:



Bipartit matchning

- **Indata:** Bipartit graf $\langle U \cup V, E \rangle$
- **Utdata:** En matchning $M \subseteq E$ av maximal storlek
- M är en matchning om inga kanter i M har någon gemensam ändpunkt
- Exempel:



Bipartit matchning, reduktion till flöde

Reduktion (transformation) av problemet bipartit matchning till flödesproblemet.
Vi löser alltså flödesproblemet och får en lösning till bipartit matchning.

function $\textsc{BipartiteMatching}(U, V, E)$
- $V' \leftarrow U \cup V \cup \{s, t\}$
- $E' \leftarrow E \cup \{(s,u) : u \in U\} \cup \{(v,t) : v \in V\}$
- **for** $e \in E'$ **do**
  - $c(e) \leftarrow 1$
- **return** $\textsc{FordFulkerson}(V', E') \cap E$

Bipartit matchning till flöde – exempel

function $\textsc{BipartiteMatching}(U,V,E)$

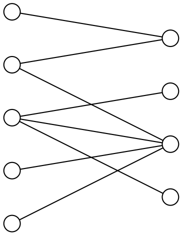
  $V' \leftarrow U \cup V \cup \{s,t\}$

  $E' \leftarrow E \cup \{(s,u) : u \in U\} \cup \{(v,t) : v \in V\}$

  for $e \in E'$ **do**

    $c(e) \leftarrow 1$

  return $\textsc{FordFulkerson}(V', E') \cap E$



Bipartit matchning till flöde – exempel

function $\textsc{BipartiteMatching}(U, V, E)$

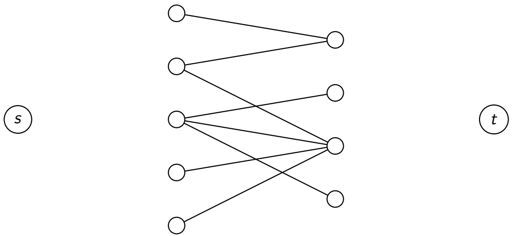
$V' \leftarrow U \cup V \cup \{s, t\}$

$E' \leftarrow E \cup \{(s,u) : u \in U\} \cup \{(v,t) : v \in V\}$

for $e \in E'$ **do**

$c(e) \leftarrow 1$

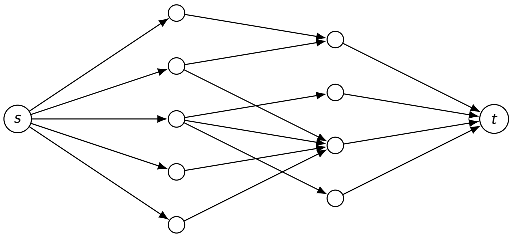
return $\textsc{FordFulkerson}(V', E') \cap E$



Bipartit matchning till flöde – exempel

function $\textsc{BipartiteMatching}(U, V, E)$

- $V' \leftarrow U \cup V \cup \{s, t\}$
- $E' \leftarrow E \cup \{(s,u) : u \in U\} \cup \{(v,t) : v \in V\}$
- **for** $e \in E'$ **do**
  - $c(e) \leftarrow 1$
- **return** $\textsc{FordFulkerson}(V', E') \cap E$



Bipartit matchning till flöde – exempel

function $\textsc{BipartiteMatching}(U,V,E)$

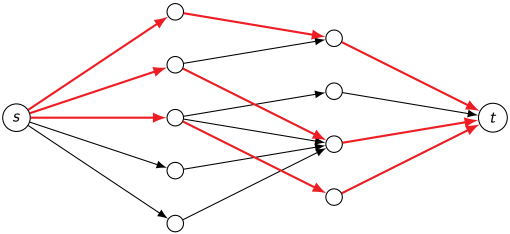
$V' \leftarrow U \cup V \cup \{s,t\}$

$E' \leftarrow E \cup \{(s,u) : u \in U\} \cup \{(v,t) : v \in V\}$

for $e \in E'$ **do**

$c(e) \leftarrow 1$

return $\textsc{FordFulkerson}(V', E') \cap E$



Bipartit matchning till flöde – exempel

function $\textsc{BipartiteMatching}(U,V,E)$
  $V' \leftarrow U \cup V \cup \{s, t\}$
  $E' \leftarrow E \cup \{(s,u) : u \in U\} \cup \{(v,t) : v \in V\}$
  for $e \in E'$ **do**
    $c(e) \leftarrow 1$
  return $\textsc{FordFulkerson}(V', E') \cap E$