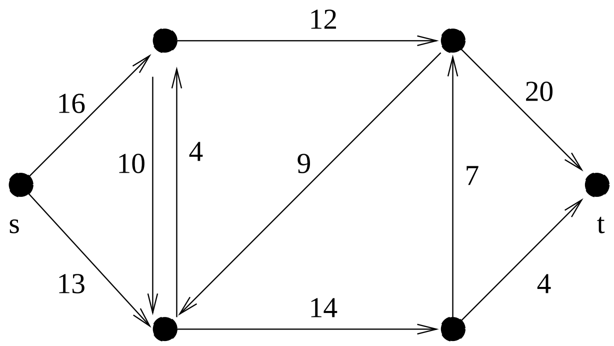
Föreläsning 6: Ford-Fulkersons flödesalgoritm

Maximalt flöde i graf

Vi har en riktad graf G med tal $c(e) \geq 0$ definierat för varje kant e i grafen. Vi har en **källa** s och en **sänka** t.
Exempel:



Med $f(e)$ menar vi ett **flöde** som skickas som går i kanten e. För flödet skall följande gälla:

$$0 \leq f(e) \leq c(e) \text{ för alla } e \in E$$

Låt sedan $In(v)$ betyda mängden av kanter som går in till v och $Ut(v)$ mängden kanter som går ut från v. Då gäller

$$\sum_{e \in In(v)} f(e) = \sum_{e \in Ut(v)} f(e) \text{ för alla } v \text{ utom } s, t$$

Vi kallar $v(f) = \sum_{e \in Ut(s)} f(e)$ för storleken på flödet.

Lite notation: Om $e = (u, v)$ skriver vi $f(e) = f(u, v)$.

Om $X \subseteq V, Y \subseteq V$ så är $f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y)$

Ett **snitt** är en partition av V i två disjunkta delar X och Y sådana att $s \in X$ och $t \in Y$.

Om (X, Y) är ett snitt så är $v(f) = f(X, Y) - f(Y, X)$

Bevis: Om x är en nod och e är en kant sätter vi

$$\begin{cases} \phi(x,e) = 1 & \text{om } e \text{ går ut från} x \\ \phi(x,e) = -1 & \text{om } e \text{ går in till} x \\ \phi(x,e) = 0 & \text{annars} \end{cases}$$

Då gäller

$v(f) = \sum_e \phi(s, e) f(e)$

$\sum_e \phi(x,e)f(e) = 0$ för alla x utom s och t.

$v(f) = \sum_{x\in X}\sum_e \phi(x,e)f(e)$
Men om $e = (x_1, x_2)$ så ger e bidrag $f(e)$ från $\sum_e \phi(x_1,e)f(e)$ och bidrag $-f(e)$ från $\sum_e \phi(x_2,e)f(e)$.

Så en kant $e_1 = (x,y)$ ger bidrag $f(e_1)$ från $\sum_{x\in X}\sum_e \phi(x,e)f(e)$
och en kant $e_2 = (y,x)$ ger bidrag $-f(e_2)$ från $\sum_{x\in X}\sum_e \phi(x,e)f(e)$.
Det ger $v(f) = f(X,Y) - f(Y,X)$.

Specialfall:
$v(f) = f(s, V - s) = f(s, V)$.
$v(f) = f(V - t, t) = f(V, t)$.

Hur hittar man maxflöde $s \to t$?

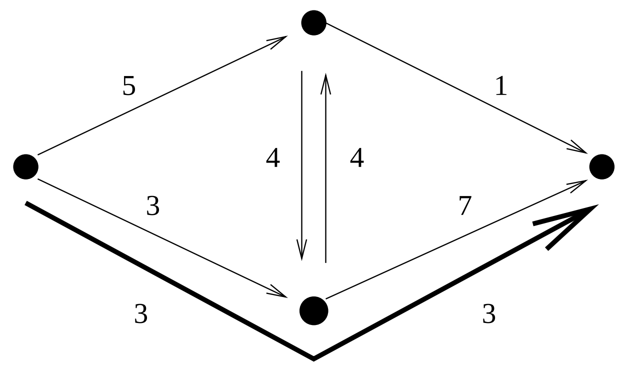
Idé (Ford-Fulkersson): Om flödet inte är maximalt så hittar vi en stig $s \to t$ som det går att öka längs. Öka sedan. Men hur hittar man sådana stigar?

Residualgraf

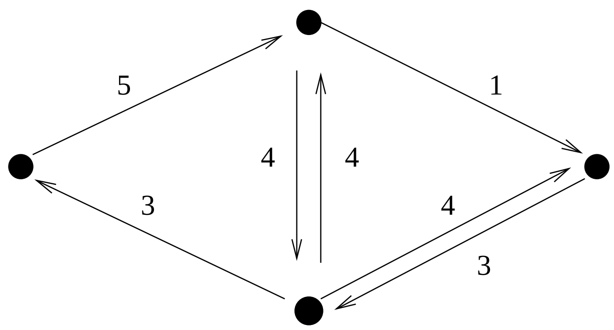
Givet grafen G och ett flöde f så definierar vi en ny graf G_f. G_f har samma noder som G. Om e är en kant i G så finns e med i G_f också. Vi $c_f(e) = c(e) - f(e)$. Om $e = (u,v)$ är en kant i G med $f(e) > 0$ och $e' = (v,u)$ inte finns med redan så lägger vi till kanten. Sedan sätts $c_f(e') = f(e)$. Kanter som nu har $c_f(e) = 0$ tas inte med i G_f.
Kapaciteten $c_f(e)$ är ett mått på möjligheten att ändra flödet.

Observera att $c_f(e) \geq 0$ för alla $e \in G_f$.

Exempel:
Flödet av värde 3 i grafen



ger restflödet



Residualgrafen G_f anger hur mycket flödet kan ändras på de olika kanterna.

Utökande stig

En utökande stig p är en stig $s \to t$ i G_f. Låt $\delta_p = min\{c_f(u,v)$ där (u,v) är kanter i $p\}$.

$\Delta_p f$ är en ändring i flödet som ges av

$$\Delta_p f(u,v) = \begin{cases} \delta_p & \text{om}(u,v) \in p \\ -\delta_p & \text{om}(v,u) \in p \\ 0 & \text{annars} \end{cases}$$

Vi bildar nytt flöde genom att sätta $f'(u,v) = f(u,v) + \Delta_p f(u,v)$. Det ger $v(f') = v(f) + \delta_p$.

Ford-Fulkersons algoritm

FORD-FULKERSON($G = \langle V, E \rangle$,c)
    foreach $e \in E$
        $f(e) \leftarrow 0$
    while det finns väg p sådan att $s \to t$ i G_f
        $\delta \leftarrow min\{c_f(u,v) : (u,v) \in p\}$
        foreach $(u,v) \in p$
            if $(u,v) \in E$
                $f(u,v) \leftarrow f(u,v) + \delta$
            if $(v,u) \in E$
                $f(u,v) \leftarrow f(u,v) - \delta$

Om utökande stig finns är f inte maximalt. Det omvända gäller : Om det inte finns utökande stig är f maximalt. Hur visar man det?

Snitt

Kapaciteten $C(X,Y)$ är definierad som $\sum_{x \in X} \sum_{y \in Y} c(x,y)$.

Viktig olikhet: $v(f) \leq C(X,Y)$ för alla snitt X, Y.

Bevis: Vi har $v(f) = f(s,V) = f(s,V) + f(X - s, V) = f(X,V) =$

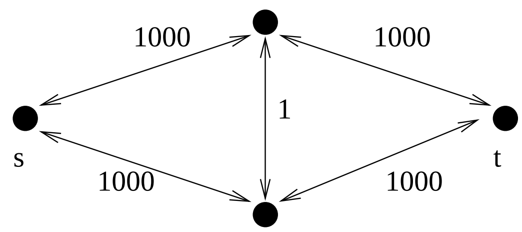
$f(X,X) + f(X,Y) = f(X,Y)$.
Men $f(X,Y) = \sum_x \sum_y f(x,y) \leq \sum_x \sum_y c(x,y) = C(X,Y)$.

Antag att f inte har någon utökande stig.
Sätt $X_f = \{v$ så att det finns en stig $s \to v$ i $G_f\}$. Sätt $Y_f = V - X_f$. Då är (X_f, Y_f) ett snitt. Om $u \in X_f$ och $v \in Y_f$ och $(u,v) \in E$ måste $f(u,v) = c(u,v)$. Om $(v,u) \in E$ måste $f(v,u) = 0$. Vi har därför $|f| = C(X_f, Y_f)$ Vi inser också att $C(X_f, Y_f)$ måste vara ett snitt av minimal storlek.

Ford-Fulkersons sats: Värdet på det maximala flödet i en graf är lika med värdet på det minimala snittet.

Effektivare flödesalgoritmer

Det finns grafer för vilka Ford-Fulkersons algoritm fungerar dåligt:



Edmonds och Karp hittade följade förbättring: Öka alltid flödet längs den kortaste stigen i residualgrafen (hittas med BFS). Högst $|V||E|$ iterationer behövs. Detta ger tidskomplexiteten $O(|V||E|^2) \in O(|V|^5)$ eftersom uppdatering av flödet tar tid $O(|E|)$.
Den bästa kända algoritmen för problemet har tidskomplexiteten $O^*(\min(|V|^{2/3}, |E|^{1/2})|E|)$ (Goldberg och Rao, 1997).
$f(n) \in O^*(g(n))$ om $f(n) \in O(g(n) \log^k n)$ för något k.

Matchning

I en bipartit graf G med $V = X \cup Y$ är en matchning ett urval av kanter så att inga kanter har gemensamma noder. Om m är matchningens storlek och $m = |X|$ har vi en fullständig matchning $X \to Y$. Om $m = |X| = |Y|$ har vi en perfekt matchning.

En beteckning: Om $A \subseteq X$ så är $\Gamma(A) \subseteq Y$ de noder som är förbundna med noder i A.

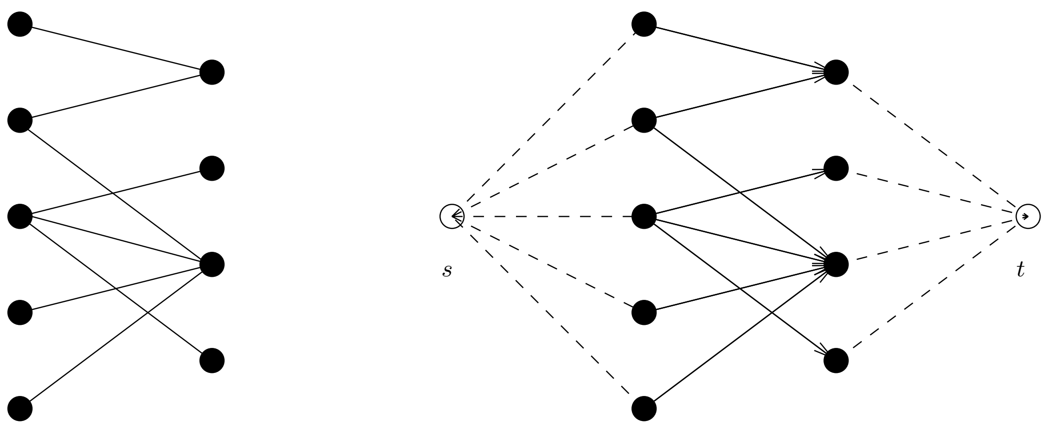
P. Halls sats: Det finns en fullständig matchning $X \to Y$ om och endast om $|A| \leq |\Gamma(A)|$ för alla $A \subseteq X$.

Bevis: Endast om går lätt att se. Vi visar det finns en fullständig matchning om villkoret är uppfyllt. Vi antar att $|A| \leq |\Gamma(A)|$ alltid gäller. Använd induktion över storleken på X. Om $|X| = 1$ så finns det förstås matchning.

Fall1: $|A| < |\Gamma(A)|$ gäller för alla $A \subset X$. Välj $x_1 \in X$. $|\Gamma(x_1)| > 1$. Matcha x_1 med någon granne y_1. Sätt $X' = X - x_1$, $Y' = Y - y_1$. Låt Γ' beteckna grannmängder i nya grafen. Låt $A' \subseteq X'$. Villkoret $|A'| \leq |\Gamma'(A')|$ måste då gälla eftersom $|A'| < |\Gamma(A')|$ Enligt induktionsantagandet kan X' matchas in i Y'.
Fall2: Det finns A så att $|A| = |\Gamma(A)|$. Enligt induktionen kan A matchas med en mängd $Y_A \subseteq Y$. Sätt $X' = X - A$, $Y' = Y - Y_A$. Låt $A' \subseteq X'$. Vi vill visa att $|A'| \leq |\Gamma'(A')|$ Vi vet att $|\Gamma(A' \cup A)| \geq |A'| + |A|$. $\Gamma'(A' \cup A) = \Gamma(A') - \Gamma(A)$. $|\Gamma'(A')| = |\Gamma(A' \cup A)| - |\Gamma(A)| \geq |A' \cup A| - |A| = |A'|$. Induktionen visar att X' kan matchas in i Y'.

Maximal matchning som flödesproblem

Problemet att hitta en maximal bipartit matchning (d.v.s. maximalt antal ingående kanter) kan lösas genom att bilda ett flödesproblem och sedan lösa det:



Alla kanter har kapacitet 1 riktad åt höger. Konstruera ett maxflöde.

Sats: Om alla kanter i en graf G har heltalskapaciteter så kommer maxflödesalgoritmen att ge ett flöde som har heltalsvärde i alla kanter.

I vår graf kommer kanterna att få flöde 0 eller 1. Låt M vara mängden av kanter som har flöde 1. Då kommer M att bilda en maximal matchning.