Repetition

Inledningen innehåller saker som förmodligen redan skall vara kända. Det är en genomgång av matchning och flöden. Senare delen innehåller vissa nya fördjupningar och presentation av en variant av matchning som kallas för assignement-spelet.

P. Halls sats

I en bipatit 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 $A \subseteq X$.

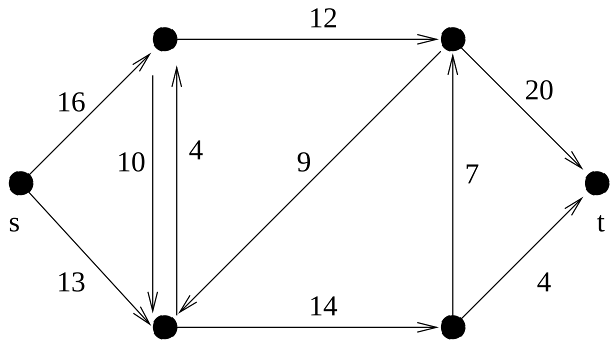
Bevis

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'.

Flödesalgoritmen

Maximalt flöde i graf

Vi har en riktad graf G med tal $c(u, v)$ definierat för varje par av noder så att $0 \leq c(u, v)$. Om $(u, v) \notin E$ måste $c(u, v) = 0$. Vi har en **källa** s och en **sänka** t.
Exempel:



Med $f(u,v)$ menar vi ett **flöde** som skickas mellan u och v. För flödet skall följande gälla:

$$\begin{cases} 1. f(u,v) \leq c(u,v) & \text{för alla } u,v \\ 2. f(u,v) = -f(v,u) & \text{för alla } u,v \\ 3. \sum_{v \in V} f(u,v) = 0 & \text{för alla } u \text{ utom } s,t \end{cases}$$

Vi kallar $|f| = \sum_v f(s,v)$ för storleken på flödet.
Lite notation: Om $X \subseteq V, Y \subseteq V$ så är $f(X,Y) = \sum_x \sum_y f(x,y)$.
Om $x \neq s,t$ så är $f(x,V) = 0$. Dessutom gäller $|f| = f(s,V)$. Det går lätt att visa att $|f| = f(V,t)$.

Bevis

$|f| = f(s,V)$. För varje delmängd X gäller $f(X,X) = 0$. Vi får $f(s,V) = f(V,V) - f(V-s,V) = -f(V-s,V) = f(V,V-s) = f(V,t) + f(V,V-s-t) = f(V,t)$.

Hur hittar man maxflöde $s \to t$?

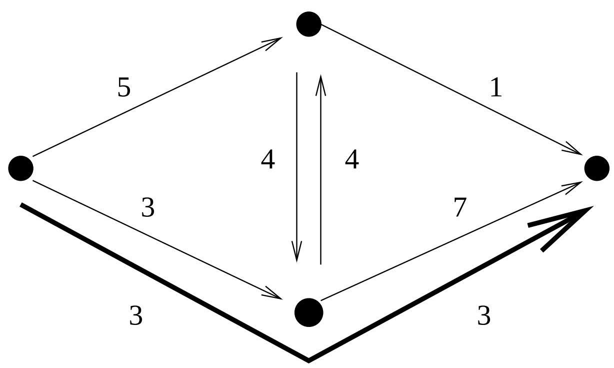
Idé (Ford-Fulkersson): Om flödet inte är maximalt så hittar vi en stig $s \to t$ som det går att öka längs. Öka sedan. Men hur hittar man sådana stigar?

Residualgraf

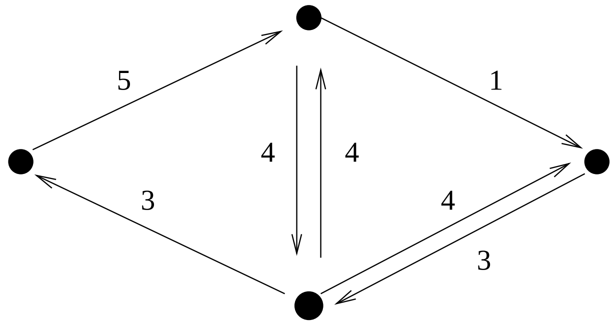
Givet grafen G och ett flöde f så sätter vi $c_f(u,v) = c(u,v) - f(u,v)$.
"Kapaciteten"$c_f(u,v)$ är ett mått på möjligheten att ändra flödet.

Även om $c(u,v) = 0$ kan $c_f(u,v) > 0$ om $f(u,v) < 0$. Men $c_f(u,v) \geq 0$ för alla u,v.

Vi låter kanterna $E_f = \{(u,v) : c_f(u,v) > 0\}$ bilda kanterna i **residualgrafen** G_f som har kapaciteterna c_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 $|f'| = |f| + \delta_p$.

Ford-Fulkersons algoritm

Ford-Fulkerson($G = \langle V, E \rangle$,c)
    foreach $(u,v) \in E$
        $f(u,v) \leftarrow 0$
        $f(v,u) \leftarrow 0$
    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 $(u,v) \in p$
            $f(u,v) \leftarrow f(u,v) + \delta$
            $f(v,u) \leftarrow -f(u,v)$

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

Ett snitt är en partition av V i två disjunkta delar X och Y sådana att $s \in X$ och $t \in Y$. Kapaciteten $C(X,Y)$ är definierad som $\sum_{x \in X} \sum_{y \in Y} c(x,y)$.

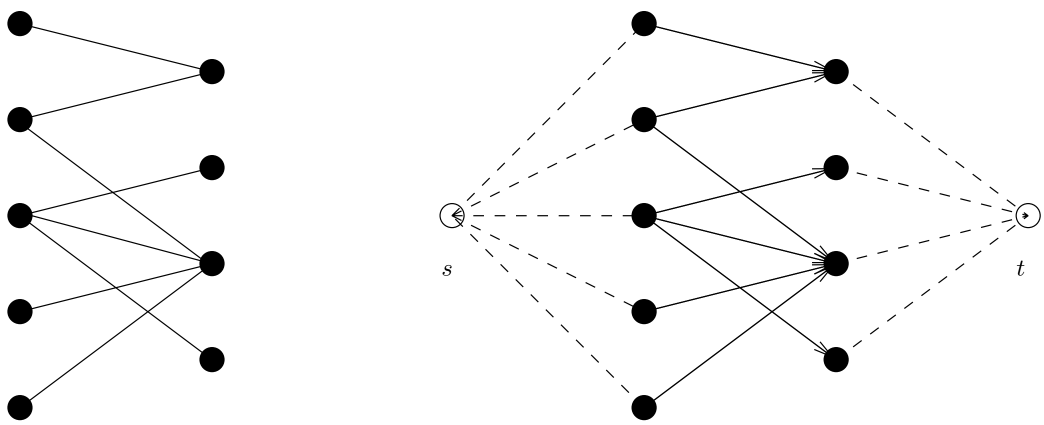
Viktig olikhet: $|f| \leq C(X,Y)$ för alla snitt X, Y.
Bevis: Vi har $|f| = f(s,V) = f(s,V) + f(X - s, V) = f(X,V) = f(X,X) + f(X,Y) = f(X,Y)$. Men $f(X,Y) = \sum_x \sum_y f(x,y) \leq \sum_x \sum_y c(x,y) = C(X,Y)$.

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.

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.

Mengers sats

Antag att vi har en oriktad sammanhängande graf med två noder s och t markerade. Två stigar från s till t är **kantdisjunkta** om de inte har några gemensamma kanter. Vi säger att en mängd $A \subseteq E(G)$ är s, t-separerande om alla stigar som går från s till t måste innehålla minst en kant i A.
Mengers sats: Storleken på en maximal mängd av kantdisjunkta stigar från s till t är lika med storleken på en minimal s, t-spererande kantmängd.

Bevis

Tolka grafen som ett nätverk med s som källa och t som sänka. Vi tänker oss att alla kanter är dubbelriktade och har kapacitet 1. Då motsvarar ett maximalt flöde en mängd kantdisjunkta stigar. På samma sätt motsvarar ett minimalt snitt ett minimalt val av s, t-separerande kanter. Ford-Fulkersons maxflödessats visar nu att dessa tal är lika.

En annan variant

Det finns ett lite annat sätt att uttrycka Mengers sats på. Två stigar från s till t är **noddisjunkta** om de inte har några gemensamma noder. Vi säger att en mängd $A \subseteq V(G) - \{s,t\}$ är s,t-separerande om alla stigar som går från s till t måste gå genom minst en nod i A.
Mengers sats: Storleken på en maximal mängd av noddisjunkta stigar från s till t är lika med storleken på en minimal s,t-spererande nodmängd.
Beviset är en modifikation av det andra beviset. Det lämnas som övning.

Mer detaljera analys av matchning

Fråga: Om inte hela X går att matcha, hur stor är då en matchning av maximal storlek.
Definition: Om storleken på en maximal matchning är $|X| - d$ så säger man att d är **defekten** hos grafen. Kan d beskrivas på något annat sätt? Om X inte kan matchas så finns det en delmängd $A \subseteq X$ så att $|A| > |\Gamma(A)|$. Vi säger att A har den lokala defekten $|A| - |\Gamma(A)|$. Definition: Givet $A \subseteq X$ är den lokala defekten $\delta(A) = \max(|A| - |\Gamma(A)|, 0)$. Det går att se att om det finns A med $\delta(A) > 0$ så måste $d \geq \delta(A)$. Faktiskt gäller $d = \max\limits_{A} \delta(A)$.

Bevis

Låt $m = \max\limits_{A} \delta(A)$. Vi visar att det finns en matchning av storlek $|X| - m$. Detta innebär att m måste vara lika med defekten d eftersom $m \leq d$. Lägg till m nya noder till Y. För varje ny nod drar vi en kant till varje nod i X. (Det ger $|X|m$ nya noder.) Den nya grafen uppfyller P. Halls villkor. Det finns en total matchning av X. Ta bort de m st noder i X som matchats med en ny nod i Y. Det ger en matchning av storlek $|X| - m$ i den gamla grafen.

Strukturen hos en maximal matchning

Vi säger att $Z \subseteq X$ är en maximal matchad mängd (MMM) om Z går att matcha och om Z inte går att utvidga med något element och fortfarande vara matchningsbar.
Påstående: Alla MMM har storlek $|X| - d$.
Bevis: Antag att Z är en MMM med $|Z| < |X| - d$. Antag att $x \notin Z$. Eftersom Z inte kan utökas med x måste det finnas en delmängd $A_x \subset Z$ så att $|A_x \cup \{x\}| > |\Gamma(A_x \cup \{x\})|$. Det betyder att $|A_x| = |\Gamma(A_x)|$ och att $x \in \Gamma(A_x)$. Låt nu $Z^* = \cup_{x \notin Z} A_x$. Då gäller $\Gamma(X - Z) \subseteq \Gamma(Z^*)$.
Om A och B är två mängder som går att matcha så är $|\Gamma(A \cup B)| = |\Gamma(A)| + |\Gamma(B)| - |\Gamma(A \cap B)| \leq |A| + |B| - |A \cap B| = |A \cup B|$. Det betyder, genom induktion, att $|\Gamma(Z^*)| = |Z^*|$. Då gäller $|\Gamma((X - Z) \cup Z^*)| = |\Gamma(Z^*)| = |Z^*|$. $|(X - Z) \cup Z^*| = |X| - |Z| + |Z^*|$. Mängden $(X - Z) \cup Z^*$ har den lokala defekten $\delta = |X| - |Z|$. Men $|X| - |Z| > |X| - (|X| - d) = d$. Så $\delta > d$. Men det är omöjligt!

Algoritmisk konsekvens

Varje mängd A som går att matcha kan utvidgas till en MMM med $|X| - d$ element. Det finns alltså inga ”dåliga val” av startmängd.

Königs sats

Vi har en bipartit graf $G = X \cup Y$. En maximal matchning är ett maximalt urval av kanter utan gemensamma noder. Låt ν vara storleken på en maximal matchning. En hörntäckning är ett urval av noder så att varje kant innehåller minst en av noderna i mängden. Låt τ vara storleken på en minimal hörntäckning. Man ser lätt att $\nu \leq \tau$. Königs sats: $\nu = \tau$.

Bevis

Använd Mengers sats. Lägg till en källa s och förbind den med alla noder i X. Lägg till en sänka t och förbind den med alla noder i Y. Värdet på $\nu =$ värdet på antalet noddisjunkta stigar från s till t. Värdet på $\tau =$ värdet på minimala antalet noder som separerar s från t. Enligt Mengers sats är då $\tau = \nu$.

Bevis av P.Halls sats från Königs sats

Antag att G uppfyller P. Halls villkor. Vi visar att $\nu = |X|$ genom att visa att X är en optimal hörntäckning. Antag att en optima hörntäckning består av $A \subseteq X$ och $B \subseteq Y$. Sätt $A' = X - A$ och $B' = Y - B$. Eftersom A, B är optimalt valda måste $\Gamma(A') = B$. Enligt P. Halls villkor är $|A\ | \leq |\Gamma(A')| = |B|$. Vi kan alltså ersätta B med A' och får då $\tau = |X|$.

Assignement-problemet

Vi har n st arbetsgivare (jobb) och n st arbetare. Om jobb nr i kombineras med arbetare j får vi en produktivitet a_{ij}. Vi vill hitta en matchning $(i, f(i))$ så att $\sum a_{i,f(i)}$ är maximal. Hur hittar man en sådan matchning?

En annan formulering

Vi har en $n \times n$-matris med element a_{ij}. En **generaliserad diagonal** är ett val av n element så att vi har exakt ett elemnet från varje rad och varje kolumn. Vi vill hitta en generaliserad diagonal med maximal elementsumma.

Algoritm

Vi använder en speciell teknik. Vi försöker hitta tal $u_1, u_2, \dots u_n, v_1, v_2, \dots v_n$ som uppfyller $u_i + v_j \geq a_{ij}$ för alla i, j. Vi ser att om A är en generaliserad diagonalsumma måste $\sum_i u_i + \sum_j v_j \geq A$. Vi kommer att visa att det minsta valet av $\sum_i u_i + \sum_j v_j$ ger värdet på en maximal diagonalsumma. Vi antar för enkelhets skull att alla a_{ij} är positiva heltal.
Vi konstruerar u_i, v_j-värden i flera steg. Vi sätter först $u_i = \max_j a_{ij}$ och $v_j = 0$. Då är $u_i + v_j \geq a_{ij}$ för alla i, j. Vi minskar u_i-värdena och ökar v_j-värdena. Om det går att hitta en matchning $(i, f(i))$ så att $u_i + v_{f(i)} = a_{i,f(i)}$ så är $\sum_i u_i + \sum_j v_j = \sum_i a_{i,f(i)}$ en optimal diagonalsumma.
Låt I vara en indexmängd. Låt $\Gamma(I) = \{j :\ \text{det finns } i \in I \text{ så att } u_i + v_j = a_{ij}\}$. Om det inte finns en matchning finns det I med $|\Gamma(I)| < |I|$. Minska alla u_i med $i \in I$ med 1. Öka alla v_j med $j \in \Gamma(I)$ med 1.
Vi ser att
1. Totalsumman $\sum_i u_i + \sum_j v_j$ minskar med minst 1.
2. Alla olikheter $u_i + v_j \geq a_{ij}$ bevaras. Men processen måste stanna eftersom totalsumman inte kan bli lägre än en diagonalsumma. Men när processen stannar har vi en matchning och en optimal diagonalsumma.

En spelteoretisk tolkning

Antag att n arbetsgivare och n arbetare skall dela in sig i par. Om arbetsgivare i och arbetare j samarbetar genererar de utdelningen a_{ij}. Vi tolkar detta som pengar. Varje arbetsgivare skall sedan få en lön"u_i och varje arbetare en lön"v_j. Det finns naturliga krav som man kan ställa på dessa löner.
1. $\sum_i u_i + \sum_j v_j = A$ där A är diagonalsumman för matchningen som är bildad.
2. För varje par (i, j) skall $u_i + v_j \geq a_{ij}$.

Analys av sista kravet

Antag att i och j inte samarbetar. Om $u_i + v_j < a_{ij}$ skulle de kunna bryta upp sina nuvarande partnerskap och samarbeta med varandra. De skulle då kunna dela på a_{ij} och fördela lönerna u_i', v_j' så att $u_i' + v_j' = a_{ij}$ och $u_i' > u_i, v_j' > v_j$. På samma sätt kan vi se att om i och j samarbetar och $u_i + v_j < a_{ij}$ så kan de dela a_{ij} och bilda nya, bättre löner u_i', v_j'. Det arrangemang som vår algoritm ger ger också något som kallas **stabilitet**. Denna stabilitet verkar vara ett nödvändigt krav för att en lönefördelning skall fungera. Algoritmen kan modifieras så att man kan få andra typer av stabila fördelningar. Det går att visa att algoritmen, som den är beskriven här, ger en fördelning som är optimal för arbetsgivarna.