Matchning i allmänna grafer

En naturlig generalisering av matchningsproblemet är att gå från bipartita grafer till godtyckliga grafer. Definition: En matchning i en godtycklig graf är ett urval av kanter som inte har några gemensamma noder. En maximal matchning är en matchning av optimal storlek. Obs: En matchning kan vara sådan att den inte går att utvidga men kan ändå vara sådan att den inte har optimal storlek. Om vi har en matchning som täcker in alla noder säger vi att vi har en **total matchning**.

Krav för en total matchning?

Det går lätt att hitta nödvändiga krav för att en total matchning skall existera: 1. Antalet noder måste vara jämnt. 2. Det får inte finnas några isolerade noder. Men det går att se att vi kan formulera ett krav som täcker in dessa krav som specialfall: 3. Det får inte finnas några udda komponenter. Men det går att vidareutveckla kraven.

Ett P. Hall-liknande krav

Låt G vara en graf. Låt $q(G)$ betyda antalet udda komponenter i G. (Så ett krav för matchning är som vi sett att $q(G) = 0$.) Följande krav är nödvändigt för att en total matchning skall finnas: $q(G - S) \leq |S|$ för alla delmängder S av $V(G)$. Bevis för nödvändighet: (Kortfattat) Om det finns en total matchning och en delmängd S av noderna och $q(G - S)$ komponenter i grafen som återstår när S tagits bort så vet vi att i en udda komponent kan inte alla nodor vara matchade med kanter som ligger i komponenten. Minst en nod måste vara matchad med en nod som ligger i S. Dessa noder i S måste räcka till". Det ger kravet $q(G-S) \leq |S|$. Det går att visa att kravet är tillräckligt också. Men beviset är ganska svårt.

Utökande stigar

Det finns ett annat sätt att se på matchningsproblemet. Antag att G är en graf och att M är en matchning (inte nödvändigtvis optimal). Om vi har en stig $(v_1, v_2), (v_2, v_3), \ldots\ (v_{k-1}, v_k)$ där varannan kant ingår i M så har vi en **alternerande** stig. Om v_1 och v_k är omatchade säger vi att stigen är **utökande**. "Utökningen fås genom att byta matchade kanter mot omatchade kanter i stigen.

Ett kriterium på optimalitet

En matchning är optimal om och endast om den inte har någon utökande stig. Bevis: Självklart är en mtchning med utökande stig inte optimal. Vi visar att en matchning som inte är optimal måste ha en utökande stig. Antag att M inte är optimal och att M' är optimal. Titta på grafen som består av kanterna i $M' \cup M - M' \cap M$. Grafen behöver inte vara sammanhängande men kanterna i grafen består omväxlande av kanter från M och M'. Tag bort alla cykler (som har jämn längd). Tag bort alla stigar som börjar med en kant i M och slutar med en kant i M'. Nu finns bara stigar som börjar och slutar med samma typ av kanter. Men $|M'| > |M|$. Därför måste det finnas minst en stig som börjar och slutar med en kant i M'. Denna stig är en utökande stig i G.

Algoritm

För att hitta en maximal matchning kan vi leta efter utökande stigar. Dessa är inte så lätta att hitta som i det bipartita fallet. Det finns en algoritm av Edmonds som löser problemet i tid $O(|V|^3)$.

Ett annat slags problem

Vi har sett att assignementproblmet är en variant av bipartit matchning. Problemet där är inte att avgöra om en matchning existerar utan att hitta en i en viss mening optimal matchning. Vi skall se på ett liknande problem.

Äktensskapsmatchningar

Vi har n st män $m_1, m_2, \ldots m_n$ och n st kvinnor $w_1, w_2, \ldots w_n$ som skall matchas ihop i äktenskap. Vi antar att varje person har en linjär (fullständig) preferensordning över personerna av motsatt kön. För enkelhets skull antar vi att alla preferenser är strikta. Vi låter $m_j >_{w_i} m_k$ betyda att kvinna w_i föredrar man m_j framför man m_i. Vi skall nu försöka hitta en äktenskapsmatchning som är bra. Vad menar vi med bra?

Ett stabilitetskriterium

Antag att vi har en äktenskapsmatchning f. Det betyder att om m_i är matchad med w_j så är $f(m_i) = w_j$ och $f^{-1}(w_j) = m_i$. Antag att m_1 är matchad med w_1 och att m_2 är matchad med m_2. Antag vidare att $w_1 >_{m_2} w_2$ och $m_2 >_{w_1} m_1$. Då finns det skäl för m_2 och w_1 att vilja förändra äktenskapsmatchningen så att $f(m_2) = w_1$. Vi säger att m_2, w_1 är ett blockerande par och att f är instabil. En matchning utan blockerande par är stabil.

Stabila äktenskaps-satsen

För varje uppsättning preferensordningar existerar det en matchning som är stabil.
Vi skall bevisa det. Terminiologin med män och kvinnor är klassisk. Men vi kommer nu att gå över till en annan terminiologi. Vi kommer att tala om arbetssökare m_i och arbetsgivare w_i. De skall paras ihop i en arbetssituation. Varje aktör har preferensordningar på aktörerna av motsatt slag. Detta uttryckssätt stämmer bättre överens med de verkliga tillämpningar teorin har haft.

Bevis för att stabila matchningar finns

Vi visar existens med hjälp av en algoritm som hittar en stabil matchning. I första steget söker varje arbetare sitt favoritjobb d.v.s. hos sin favoritarbetsgivare. Varje arbetsgivare gör följande: Väljer ut applikant som är bäst av de som dykt upp och ger de övriga avslag. Den utvalde får ett temporärt ja. I nästa steg besöker de avvisade arbetssökarna nästa arbetsgivare i sin preferensordning. Arbetsgivarna gör nu som i det första steget. Enda skillnaden är att en person som fått ett temporärt ja kommer att bli avvisad om någon högre rankad applikant dykt upp. Processen fortsätts enda till alla arbetssökarna har fått ett temporärt jobb. Denna arbetsfördelning utgör då matchningen.

Stannar alltid algoritmen?

Varje arbetsgivare som har besökts av en arbetssökare kommer i fortsättningen att ha en sådan temporärt anställd. Antalet rörliga arbetssökare kommer därför att minska i varje steg eftersom vi arbetar oss neråt i deras preferensordningar. Därför måste algoritmen stanna.

Ger den alltid en stabil äktenskapsfördelning?

Antag att det finns ett blockerande par m, w. Antag att $f(m) = w'$ och $f^{-1}(w) = m'$ där f är matchningen som ges av algoritmen. Eftersom paret är blockerande måste $w >_m w'$ och $m >_w m'$. Då måste m ha besökt w före w'. Då måste w ha avvisat honom för någon m''. Dessutom måste då $m' = m''$ eller $m' >_w m''$. Men w byter inte ner sig". Det betyder att $m >_w m'$ är omöjligt. Något blockerande par kan inte finnas.

Algoritmen är optimal för arbetssökarna

Vi säger att ett arbete w är **passande** för m om det finns en stabil matchning f så att $f(m) = w$. Vi sätter $p(m) = \{w : w \text{ är passande för } m\}$. Vi sätter också $\operatorname{opt}(m) =$ det arbete i $p(m)$ som m rankar högst. Det går att visa algoritmen ger en matchning f så att $f(m) = \operatorname{opt}(m)$ för alla m! Det betyder att f är den unikt optimala matchningen för arbetssökarna.

Bevis

Vi gör först en iaktagelse. Vi tittar på första omgången. Om m blir avvisad av w så är w inte passande för m. För bevis, antag att $w \in p(m)$. Då finns det en stbil matchning g med $g(m) = w$. Men det finns m' (w:s utvalde applikant i 1:a omgången) som rankar w högst av alla jobb. Dessutom gäller $m' >_w m$. Det betyder att paret m', w blockerar g. Därför gäller $w \notin p(m)$.

Induktion

Antag att vi har visat att efter k omgångar har ingen arbetssökare m avvisats från ett jobb w som är passande för m. Vi visar att detta då gäller också efter $k + 1$ omgångar. Antag att m söker jobb w och blir avvisad. Antag att det finns en stabil matchning g med $g(m) = w$. Men vi vet att det finns en m' som w föredrar (som slår ut m i algoritmen). Det kan hända att m' inte rankar w högst av alla jobb. Men inget jobb som m' rankar högre är passande för m' enligt induktionsantagandet. Det betyder att $w >_{m'} g(m')$. Så $m'w$ är ett blockerande par till g och g kan inte vara stabil.

Slutsats

När algoritmen är klar kan ingen arbetssökare ha avvisats från ett för denne passande jobb. Om $f(m) = w$ gäller $w \in p(m)$ (eftersom f är stabil). Eftersom jobben söks i preferensordning måste $f(m) = \operatorname{opt}(m)$

Algoritmen ur arbetsgivarnas perspektiv

Låt $p(w) =$ mängden av arbetssökare m som passar för w, d.v.s. det finns stabil matchning g så att $g(m) = w$. Sätt $\operatorname{sub}(w) =$ den passande arbetssökare som w rankar lägst. De gäller att algoritmen ger en matchning f så att $f^{-1}(w) = \operatorname{sub}(w)$. Algoritmen ger sämsta möjliga resultat för arbetsgivarna.

Bevis

Det går att se som i föregående bevis att ingen arbetsgivare avvisar någon passande arbetssökare. Dessutom gäller förstås att $f^{-1}(w) \in p(w)$. Antag att det finns m' så att $m' \in p(m)$ och $m >_w m$ och att det finns en stabil matchning g så att $g^{-1}(w) = m'$. Då måste $g(m) = w'$ uppfylla $w >_m w'$ (eftersom $w = \mathrm{opt}(m)$). Men då är m', w ett blockerande par till g.

Äktenskapsmatchning i generella grafer

Antag att vi har en komplett graf K_n där varje nod har en preferensordning på övriga noder. Vi antar att antalet noder är jämnt. (Om vi startar med ett udda antal noder kan vi lägga till en nod n^* som placeras sist i alla preferensordningar och som själv ges en godtycklig preferensordning.) En stabil matchning är en funktion f så att $(x) \neq x$ och om $f(x) = y$ så är $f(y) = x$. Dessutom skall det gälla att om x, y är ett matchat par så får det inte finnas z så att $x >_z f(z)$ och $z >_x$. Den stora frågan är om det alltid finns stabila matchningar. Svaret är nej!

Ett spel på mängder

Vi har en grundmängd $M = \{1, .., n\}$. Till varje delmängd F tilldelas ett värde a_F. Elementen i M tolkas som **spelare**. Ett urval av delmängder till M utgör en **partition** av M om varje spelare ligger i exakt en delmängd. Delmängderna kallas för **koalitioner**. Vi säger att spelarna i en koalition F drar in värdet a_F. Antag att vi har en partition P. Produktiviteten för partitionen är $v(P) = \sum_{F \in P} A_F$. Givet P skall varje spelare få en utdelning (lön") u_i. Vi ställer 2 krav på dessa utdelningar. 1. $\sum_i u_i = v(P)$. 2. För varje delmängd F' (i P eller inte) måste vi ha $\sum_{i \in F'} u_i \geq a_{F'}$. Detta verkar vara nödvändiga krav för stabilitet hos en partition och utdelning. Om det sista kravet inte är uppfyllt lönar det sig för personerna i F' att bryta upp ur sina koalitioner och bilda nya koalitionen F'.

Går stabilitet alltid att få?

Givet en mängd M och en värdefunktion a_F på delmängderna, går det att hitta en partition P och en motsvarande mägd med utdelningar som ger stabilitet? Svaret är nej!