

1. Párosítások

1.1. Alapfogalmak

Emlékeztető. Legyen G egy gráf, $E(G)$ a G élhalmaza, $V(G)$ gráfunk csúcshalmaza. Legyen $F \subseteq E(G)$. Ekkor $V(F) = \{x \in V : x \text{ illeszkedik egy } F\text{-beli élre}\}$.

$V(F)$ egy v eleméről azt mondjuk, hogy F *lefedi* a v csúcsot.

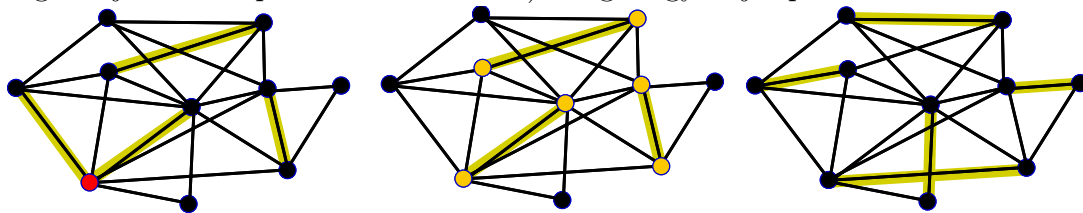
Definíció. M párosítás, ha $|V(M)| = 2|M|$ vagyis M nem-hurok élek végpont-diszjunkt halmaza.

Egy M párosítás által lefedett v csúcsról azt mondjuk, hogy *párosított*. Legyen $\bar{V}(M) = V(G) - V(M)$ az M által nem párosított/ M -párosítatlan csúcsok halmaza. $|\bar{V}(M)| = |V(G)| - |V(M)| = |V(G)| - 2|M|$

Definíció. Ha az M párosítás és $V(M) = V(G)$, akkor *teljes párosításról* beszélünk.

Természetesen csak páros pontszámú gráfoknál lehetséges, hogy létezzen teljes párosítás.

Példa. Egy gráf egy élhalmazzal (sárgával kiemelt élek), ami nem párosítás (pirossal jelzett a csúcs, ahol két eleme összefut). Majd egy párosítás, ami nem teljes párosítás (sárgával jeleztük a párosított csúcsokat). Végül egy teljes párosítás.



Definíció. $\nu(G)$ a G -beli párosítások között a legnagyobb méret.

Ekkor $2\nu(G)$ a legtöbb csúcs, amit párosítani tudunk. $|V(G)| - 2\nu(G)$ a legkevesebb csúcs, ami kimarad egy párosításból.

1.2. Mohó algoritmus

Az algoritmus mohó jelzője onnan ered, hogy nem vonjuk vissza soha a korábbi döntésünket, vagyis ha egy élt egyszer betettünk a párosításba, már nem vesszük ki később. Azzal, hogy korábbi döntésünket nem bíráljuk felül egy nagyon hatékony, egyszerű eljárást kapunk. Sajnos nincs garancia, hogy az output optimális.

Mohó párosítási algoritmus:

(Inicializálás) Kiindul egy M párosításból

AMÍG létezik $e \in E(G) - M$ él úgy, hogy $M \cup \{e\}$ is párosítás

[(Mohó növelés/bővítés) M -et lecseréljük $M \cup \{e\}$ -re.]

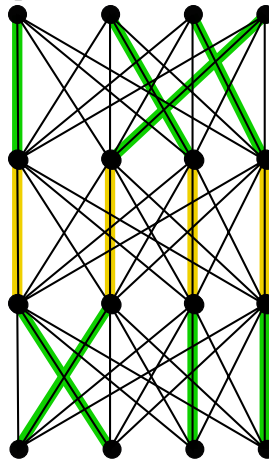
(Elakadás) Az aktuális párosítás az output

//Ekkor minden M -en kívüli él összefut valamelyik M -beli éllel

Megjegyezzük, hogy nincs ciklizálási veszély: bármely javítás garantáltan növeli a párosítás élszámát, ezért az algoritmus szükségszerűen leáll.

Elakadás esetén tudjuk, hogy párosításunk egy speciális módon, a mohó módon nem javítható. Ez nem jelenti azt, hogy ettől eltérő módon nem tudunk nagyobb párosításhoz jutni.

Példa. Gráfunknak négy ugyanannyi csúcsot (legyen ez n , az ábrán $n = 4$) tartalmazó „emelete” van. Két szomszédos emelet között minden élt behúztunk, további élek nincsenek. A sárga élek egy teljes párosítást alkotnak a két középső szint között. Ha a mohó algoritmus ezeket választja ki először, akkor elakad: n élt tartalmaz outputja. A zöld élek egy teljes párosítást alkotnak ($2n$ darab él).



Megemlítünk egy, a fenti algoritmussal kapcsolatos alaptételt. Ez azt garantálja, hogy a nagyon egyszerű algoritmus outputja nem olyan rossz.

1. Tétel. Legyen $\nu_{\text{mohó}}(G)$ a mohó algoritmus egy tetszőleges futásának mérete. Legyen $\nu(G)$ a legnagyobb párosítás mérete. Ekkor

$$\frac{\nu(G)}{2} \leq \nu_{\text{mohó}}(G) \leq \nu(G).$$

Bizonyítás. (BSc anyag) A második egyenlőtlenség nyilvánvaló abból, hogy a mohó algoritmus egy párosítást számol ki.

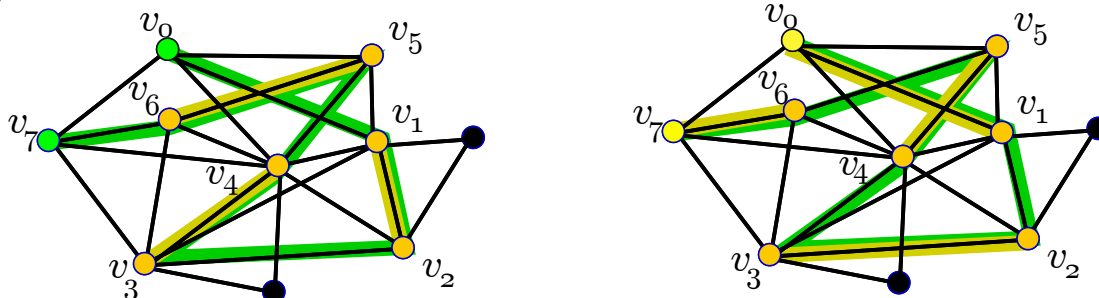
Az első egyenlőtlenség igazolása: Legyen $M_{\text{mohó}}$ a mohó algoritmus outputja, $L = V(M_{\text{mohó}})$ a mohó algoritmus outputja által párosított pontok halmaza. Nyilvánvaló, hogy L lefogó ponthalmaz és $|L| = 2\nu_{\text{mohó}}(G)$. Így L mérete minden párosítás méretét felülről becsli, speciálisan $\nu(G) \leq |L| = 2\nu_{\text{mohó}}(G)$. ■

A továbbiakban maximális elemszámú párosításokat megkereső algoritmusokkal fogunk foglalkozni.

1.3. Javító utas algoritmusok

Definíció. Legyen G gráf M párosítás G -ben, $P : v_0, e_1, v_1 \dots e_k v_k$ egy út. P út javító út M -re nézve, ha v_0 és v_k nem párosítottak, k páratlan és a páros sokadik élek elemei M -nek.

Példa. A sárga párosításra nézve a zöld út javító út. A második ábrán a javított párosítás látható:



A P javító út segítségével kapott $M' = (M \setminus E(P)) \cup (E(P) \setminus M) = M \Delta E(P)$ élhalmaz párosítás. Azaz M' -be P mentén pontosan azon P -beli éleket rakjuk, melyek eredetileg nem voltak M -ben, P -n kívül a párosítás nem változik. Könnyen látható, hogy így mindig eggyel nagyobb élszámú párosítást kapunk. Ezt nevezzük M javító utas javításának.

Ezzel sémát kaptunk négy párosításokat kereső algoritmusokra, amiket javító utas algoritmusoknak nevezünk. Ezek általános vázlata:

Javító utas algoritmusok sémája:

Adott egy G gráf és benne egy M párosítás

AMÍG találunk M -re vonatkozó javító utat

[(Javító utas növelés) M -et lecseréljük $M \Delta E(P)$ -re.]

(Elakadás) Az aktuális párosítás az output

// Ekkor nem létezik M -re vonatkozó javító út.

Megjegyzés. Itt sincs ciklizálási veszély, mert bármely javítás garantáltan növeli a párosítás élszámát, ezért az algoritmus szükségszerűen leáll.

Megjegyzés. Az 1 hosszú javító út menti javítás a mohó javítás.

Azaz a javító utas algoritmusok a mohó algoritmus kiterjesztései. A mohó algoritmus elakadására korábban láttunk példát. Könnyű ellenőrizni, hogy azon futva a javító utas algoritmusok megtalálják a teljes párosítást. Ez nem véeltlen.

2. Tétel (Berge-tétel). G gráf, M párosítás. Ha M nem optimális, akkor létezik rá javító út. Vagyis elakadás esetén az aktuális párosítás optimális.

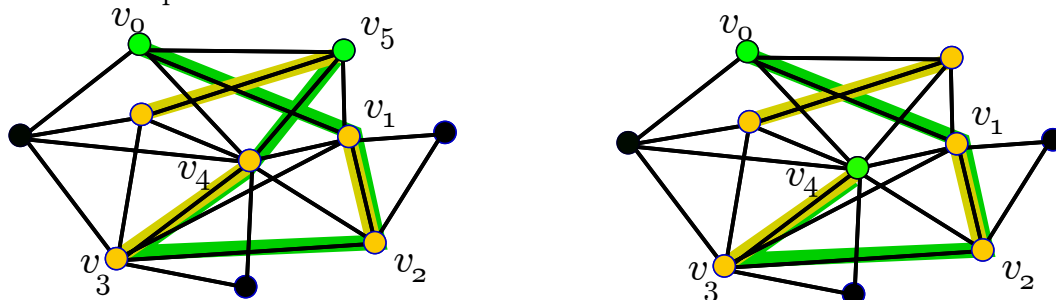
Megjegyzés. A javító utak hatékony keresése nem egyszerű! Keresésünknek olyanak kell lenni, ha sikertelen, akkor ne is legyen javító út. Egy gráfban az utak teljes sokasága hatalmas lehet. Így ennek végignézése általában túl sokáig tart.

A következőkben a javító út keresésre mutatunk egy mohó változatot.

1.4. Mohó javító út keresés

Definíció. P javító út kezdemény, ha $P : v_0 e_1 v_1 \dots e_k v_k$ út, és $v_0 \notin V(M)$, valamint a páros indexű élek elemei M -nek.

Példa. Egy öt hosszú javító út kezdemény, ami nem javító út, mert v_5 párosított. A második ábrán egy négy hosszú javító út kezdemény, ami nem lehet javító út, hiszen hossza páros.



Javító út kezdeményeket keresünk és ezeket párhuzamosan növeljük annak reményében, hogy javító úttá terjesztjük ki őket. Keresésünk megvalósítása egy címkézéssé lesz: A csúcsokra címkéket helyezünk, aminek jelentése: oda vezető javító út kezdeményt találtunk.

Ezen címkék egy kicsit több információt tartalmaznak mint az odavezető javító út kezdemény megtalálásának ténye. Ha a megtalált javító út kezdeménye páros hosszú akkor a csúcs külső címkét kap, ha pedig páratlan hosszú, akkor **belső** címkét kap.

Legyen C a címkézett csúcsok halmaza és K a külső, B a belső címkézett pontok halmaza. Nyilván $C = K \dot{\cup} B$.

A mohóság abban nyilvánul meg, hogy a címkehalmaz folyamatosan bővül, és nincs felülírás. Azaz, ha azt tapasztaljuk, hogy már címkézett pontba egy eddig nem látott javító út kezdemény halad, akkor ezt elvetjük („későn vettük észre”). Az eredeti javító út kezdemény (amire a címkézést korábban alapítottuk) lesz az amit folytatni próbálunk.

Kiinduló címkehalmazt választunk (ezek pontok lesznek, amikbe egy 0 hosszú javító út kezdeményt találtunk, azaz $\bar{V}(M)$ egy részhalmaza) és a címkézést az alábbi módon bővítjük:

Mohó javító út kereső algoritmus:

(Inicializálás) Kiindulunk $\bar{V}(M)$ egy R részhalmazából.

A kezdeti $K = R$,

A kezdeti $B = \emptyset$,

// Így a kezdeti $C = K = R$.

AMÍG találunk $k \in K$ csúcsot és s címkézetlen szomszédját

[Ha s nem párosított csúcs, akkor

(Sikeres keresés) a k -ba vezető javító út kezdemény s -be meghosszabbítva egy javító utat kapunk.

Ha s párosított csúcs, akkor

(Mohó címkénövelés) legyen s' az M -beli párja.

$K \leftarrow K \cup \{s'\}$,

$B \leftarrow K \cup \{s\}$,

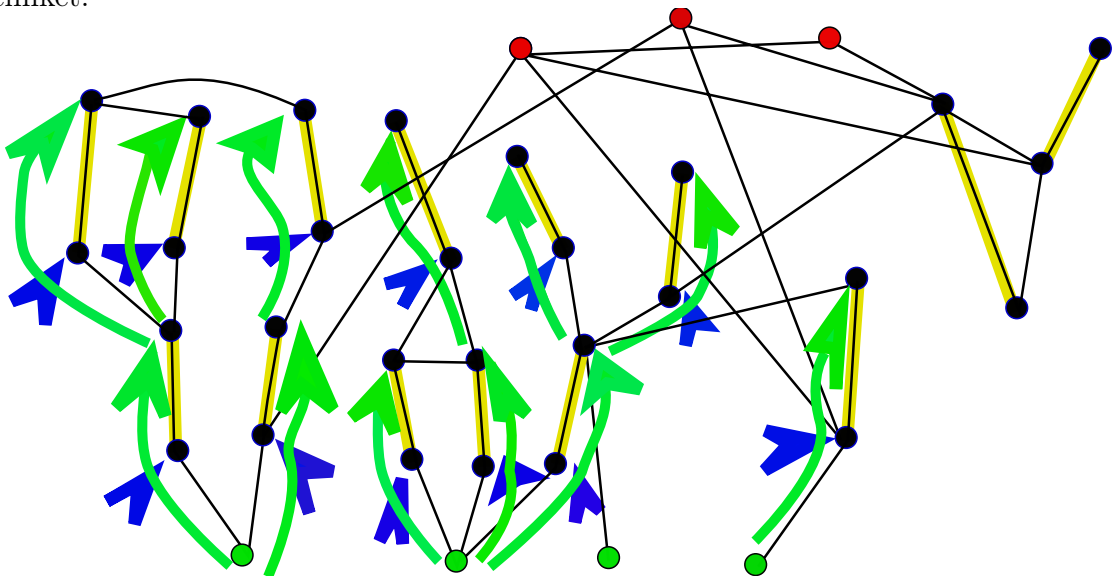
$C \leftarrow C \cup \{s, s'\}$.]

(Elakadás) Ha a fenti ciklusból nem 'sikeres kereséssel' jöttünk ki, akkor 'sikertelen kereséssel' állunk le.
 // Sikertelen keresés esetén minden külső pont
 // összes szomszédja címkézett.

A sikeres keresés esete „tiszta”. A sikertelen keresés esetén azonban célszerű vizualizálni mi is történt.

A címkenövelések minden címkekiosztásához egy „felelős” rendelhető: s k miatt kapja címkéjét, s' pedig s miatt. A megfelelő élek mentén terjed ki a címke. Így a címketerjedést/keresést gyökeres erdő írja le. A gyökerek az inicializálás során kiválasztott R pontjai. A címkék további terjedése dupla ághajtásokkal történik. A kettő hosszú ágak belső pontjai kapják a belső címkét. A végső pontjai a külső címkét kapott pontok, ahonnan a keresés/címkekiosztás továbbléphet.

Példa. A sárga élek alkotják párosításunkat. A zöld csúcsok azok a párosítatlan csúcsok, amelyek kiinduláskor külső címkét kaptak (R elemei). A piros csúcsok azok a párosítatlan csúcsok, amelyek kiinduláskor nem lettek megcímkézve ($\bar{V}(M) - R$, a célcsúcsok, amiket szeretnénk keresésünkkel elérni). A zöld nyilak a dupla ághajtások (a külső címke terjedése). Közben a kék nyíl mutatja a kiosztott belső címkét.

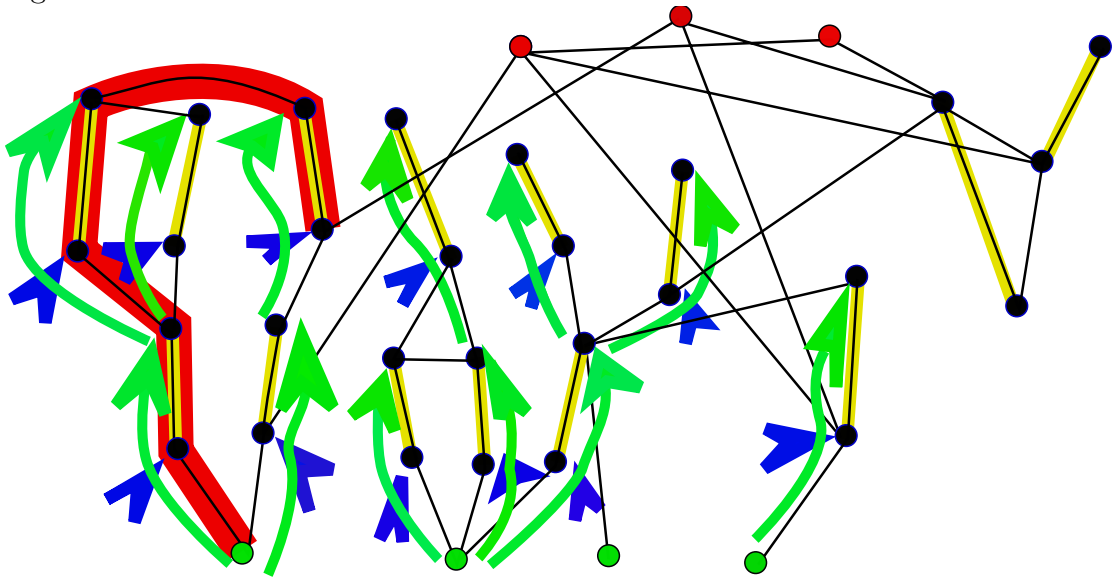


Észrevétel. Kezdetben a címkézett pontok nem párosított csúcsok és mind külső címkét kap, majd egyszerre két párosított csúcs kap címkét, egyik külsőt, másik belsőt. Ennek két következményét emeljük ki.

- (1) s -et úgy választottuk, hogy ne legyen címkézett. Így automatikusan teljesül, hogy s' se címkézett. Azaz korábbi ígéretünk (nincs újracímkézés) teljesül.
- (2) A címkézés minden címkekiterjesztése után $|K| - |B| = |R|$. Valóban: A kezdetben ez teljesült, majd mindig eggyel növeltük $|K|$ -t és $|B|$ -t is, azaz különbségük azonos marad.

Példa. Előfordulhat, hogy a keresés belső pontként ér el egy x pontot és ekkor a keresés x -nek az x' párja felé megy, és x más szomszédja felé nem. Egy másik javító út kezdemény viszont külső pontként érné el x -et. Az ábrán egy ilyen rossz fázisban

elért javító út kezdeményt emel ki a piros út. Ezt be is fejezhetnénk javító úttá, ha megtaláltuk volna.



3. Tétel (Kőnig Dénes - Egerváry Jenő / Magyar módszer). Ha G páros gráf (A alsó és F felső színosztályokkal), M párosítás. Legyen $R = A \cap \bar{V}(M)$. Ekkor ha a mohó javító út kereső algoritmus sikertelen kereséssel ér véget, akkor M optimális, azaz nincs rá vonatkozó javító út.

Bizonyítás. Kezdetben $K \subseteq A$, $B \subseteq F$. Címkekiosztásnál külső címkét egy belső címkéjű pont szomszédja kap és belső címkét egy külső címkéjű pont szomszédja kap. Így a fenti tulajdonság öröklődik. Mindig (így az algoritmus végén is) $K \subseteq A$, $B \subseteq F$. Azaz a külső és alsó kategóriák ugyanazok. Hasonlóan a belső és felső kategóriák ugyanazok.

Feltevésünk szerint sikertelen kereséssel állunk le. Így a külső pontok szomszédai mind címkézettek.

Jelölés. K szomszédsága: $N(K) = \{s \in V, s \text{ szomszédos valamely } K\text{-belivel}\}$.

Tehát az algoritmus végén $N(K) \subset C$.

A tétel nagyon fontos, két bizonyítást is közlünk rá.

I. Bizonyítás: Néhány általános megjegyzéssel kezdünk.

Definíció. Legyen $S \subset A$. Legyen $\epsilon(S) = |S| - |N(S)|$. Azaz S elemszámának többlete szomszédainak számához képest (ami persze lehet negatív is).

Definíció. $\delta_A(P) = |A| - |P|$ (nem párosított pontok száma A -ban).

Észrevétel. Legyen S egy tetszőleges alsó pontokat tartalmazó halmaz, P egy párosítás. Ekkor

$$\delta_A(P) \geq \epsilon(S).$$

Tetszőleges P párosítás esetén S -beli csúcsok párvai $N(S)$ -ből kerülnek ki. Így legalább $|S| - |N(S)| = \epsilon(S)$ csúcs párosítatlan marad A -ban (már csak S -et figyelembe véve is). Ezek után az észrevételbeli egyenlőtlenség nyilvánvaló.

Persze az észrevétel csak akkor nem semmitmondó, ha $\epsilon(S) > 0$.

4. Következmény. Legyen S alsó pontok egy halmaza, amelyre $\epsilon(S) > 0$. Ekkor G -ben nincs teljes párosítás.

A fenti tulajdonságú S halmazokat hasznos külön névvel ellátni.

Definíció. Alsó pontok egy S halmaza *Kőnig-akadály*, ha $\epsilon(S) > 0$, azaz S szomszédainak száma kisebb mint elemeinek száma.

5. Következmény. Legyen S alsó pontok egy halmaza és P egy párosítás. Ha $\epsilon(S) = \delta_A(P)$, akkor P egy optimális párosítás.

A fenti esetben azt mondjuk, hogy S egy bizonyító alsó-Kőnig-halmaz P optimalitására.

Térjünk vissza a G páros gráfunkhoz, amelyben az M párosítással a magyar módszert futtatva sikertelen keresést folytattunk. Tudjuk, hogy leálláskor $N(K) \subset C$. Páros gráfban azt is tudjuk, hogy $K \subset A$, így $N(K) \subset F$. Azaz $N(K) \subset C \cap F = B$. Igazából belső címkét csak úgy kaphat egy csúcs, hogy egy már külső címkéjű csúcs szomszédja. Azaz $N(K) = B$.

Tehát

$$\epsilon(K) = |K| - |N(K)| = |K| - |B| = |R| = |A \cap \bar{V}(M)| = \delta_A(M).$$

Azaz K egy bizonyító alsó-Kőnig-halmaz M optimalitására.

II. Bizonyítás: Ismét néhány általános megjegyzéssel kezdünk.

Emlékeztető. Ey L csúcshalmaz lefogó ponthalmaz, ha minden él legalább egy L -beli pontra illeszkedik.

Egy a fogalmat megvilágító „mese”: Legyen a gráf egy múzeum tervrajza, amiben képek vannak a folyosókon (éleken). Ezeket örökkel őriztetjük, akik csúcspontokban ülhetnek, ahol az ott összefutó összes folyosót ellenőrzik. Minél kevesebb őrrel akarjuk az összes élt/folyosót ellenőrizni.

Észrevétel. Ha L lefogó ponthalmaz, és P párosítás: $|L| \geq |P|$.

A „mese” szerepkiosztását használó magyarázat: P össze nem futó folyosórendszer, így mindegyik eleme külön őrt követel. Azaz tetszőleges jó őrelhelyezésben legalább $|P|$ darab őr szerepel.

Kiemeljük két fontos következményét az észrevételnek:

6. Következmény. Legyen L egy lefogó ponthalmaz és M egy párosítás. Ha $|L| = |P|$, akkor P a lehető legnagyobb párosítás G -ben és L a lehető legkisebb lefogó ponthalmaz G -ben.

A következményben szereplő L, P pár esetén azt mondjuk, hogy L egy bizonyító lefogó ponthalmaz P optimalitására és M egy bizonyító párosítás L optimalitására.

Most térjünk vissza G páros gráfunkhoz benne egy M párosítással, ahol a magyar módszer sikertelen kereséssel állt le.

Az M -beli élek két csoportra, címkézett és címkézetlen élekre osztható. (A címkézett élek mindkét végpontja, a címkézetleneknek egyik végpontja sem címkézett.) Jelöljük a címkézett felső végpontok halmazát L_C -vel, a címkézetlen alsó végpontok halmazát $L_{\bar{C}}$ -vel. Vegyük észre, hogy $L_C = B$. Legyen $L = L_C \cup L_{\bar{C}}$. Ekkor $|L| = |M|$, hiszen minden M -beli él egy csúccsal járul hozzá L -hez, amit ezek a hozzájárulások adnak ki.

Észrevétel. L lefogó ponthalmaz G -ben.

Valóban: Legyen $e = af$ egy tetszőleges él ($a \in A, f \in F$).

(1) Ha f címkézett, akkor $f \in B$ és így $f \in L_C$.

(2) Ha f nem címkézett, akkor a sem címkézett (hiszen, ha a címkézett, azaz külső pont lenne, akkor szomszédja nem lehetne címkézetlen az címkekiosztás elakadásánál). Speciálisan $a \notin C_0 \Rightarrow a$. Azaz a párosított alsó pont címkézetlen M -beli élen: $a \in L_{\bar{C}}$.

Tehát $L = L_C \cup L_{\bar{C}}$ valóban lefogó ponthalmaz.

Az észrevétel azt adja, hogy az L, M pár olyan, hogy L bizonyítja M optimalitását (természetesen M is bizonyítja L optimalitását). ■

Megjegyzés. Mindkét bizonyítás többet állít mint az M optimalitása. Az is adódott, hogy M optimalitására van bizonyító lefogó ponthalmaz és bizonyító alsó-Kőnig-halmaz. Ezek konstruktív módon adódnak a magyar módszer melléktermékeként.

Megjegyzés. Az is adódott, hogy a második bizonyításban kiolvasott L lefogó halmaz optimális. Azaz a magyar módszer alkalmazható a legkisebb méretű lefogó ponthalmaz meghatározására páros gráfokban.

Az előző tétel páros gráfokra működik csak. Mire az általánosabb, nem páros gráfokra is hasonló tételek kialakultak, majdnem 10 év telt el. A következő részben az általános esetben vizsgáljuk a problémát, azaz nem szükségszerűen páros gráfokban keresünk javító utat egy adott párosításra nézve.

1.5. Edmonds-algoritmus

A magyar módszerben az alsó párosítatlan pontokból indítottuk a keresésünket. Általában nincs ilyen fogalmunk, mint „alsó pontok”. Most kiindulásnak $K = \bar{V}(M)$, $B = \emptyset$ címkézést választjuk. Azaz az összes párosítatlan pontot 0 hosszú javító út kezdeményeknek tekintjük és alkalmazzuk a mohó címkenövelő eljárást elakadásig. Ez biztosan elakad, hiszen a növelés mohón történik és kezdetben minden lehetséges javító út végpont „lehetséges javító út kezdőpont” címkét kap.

Edmonds-algoritmus:

(Inicializálás) Legyen $R = \bar{V}(M)$.

A kezdeti $K = R$,

A kezdeti $B = \emptyset$,

//Így a kezdeti $C = K = R$.

(Címkenövelés) Mohó címkenövelés elakadásig

// Ekkor a külső csúcsok mindegyik szomszédja címkézett. Kialakul egy

// F kereső erdő. A kereső erdő mindegyik komponense egy-egy R -beli

// pontban gyökerezett.

AMÍG találunk $k, k' \in K$ csúcsot, amelyek egy e él mentén szomszédosak

// Legyen r és r' azon R -beli elemek, amelyekből induló/amelyekben

// gyökerező keresések megcímkézik k -t és k' -t.

[Ha $r \neq r'$, akkor

(Sikeres keresés) a k -ba vezető javító út kezdemény után e -n

keresztül k -ba lépve, majd bejárva az ehhez vezető javító

út kezdeményt fordítva r és r' közötti javító utat találtunk.

Ha $r = r'$, akkor

(Edmonds-eset) \star // Később tárgyaljuk.]

(Elakadás) Ha a fenti ciklusból nem 'Sikeres kereséssel' jöttünk ki, akkor 'Sikertelen kereséssel' állunk le.

Mielőtt az „Edmonds-esetet” részletesen tárgyaljuk néhány fogalmat kialakítunk. Legyen P az r -ből k -ba vezető javító út kezdemény, ami a címkézéséért felel. Hasonlóan legyen P' az $r(=r')$ -ből k' -be vezető javító út kezdemény, ami a címkézéséért felel. Legyen a_e az elágazás pontja, azaz az utolsó csúcs, amíg a két út közösen halad.

Észrevétel. a_e -ben a kereső erdő szétágazik vagy P és P' közül valamelyik út leáll (azaz $a_e \in \{k, k'\}$). Mindkét esetben a_e egy külső pont a felépített kereső erdőben.

P -n a_e -től k -ig, illetve P' -n a_e -től k' -ig dupla ághajtásokkal ért el a keresés. Azaz az odavezető két út mindegyike páros hosszú, amit e egy páratlan körré, C_e fűz össze.

r komponense az F kereső erdőben egy r gyökerű F_r kereső fa. Ebben e nem egy él. Így F_r -hez hozzáadva az e élt egyetlen kör keletkezik. Ezt írjuk le előbb.

Észrevétel. C_e összes pontjába vezet páros hosszú javító út kezdemény is.

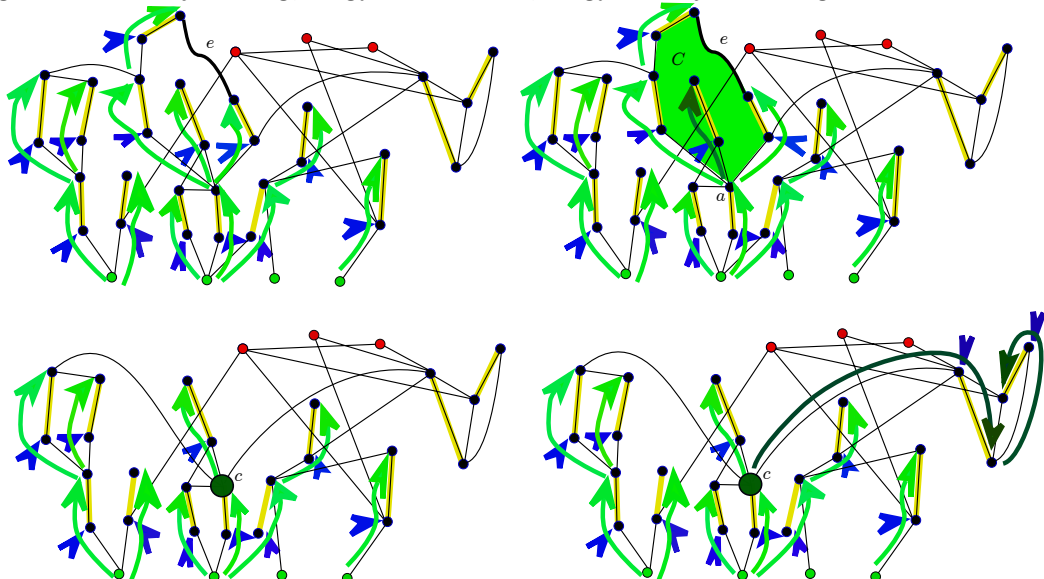
Persze C_e összes pontjába a_e -től különböző pontjába vezet páratlan hosszú javító út kezdemény is. Számunkra azonban a páros hosszúak értékesebbek. Ezek azok, amelyek olyan fázisban érik el a csúcst, hogy a keresés nem determinált/szétágaz/-hat.

Definíció. G -ben a C_e kör zsugorítása (a kör ponthalmazát egy pontba húzzuk össze) a \tilde{G} gráfhoz vezet. Formálisan: $V(\tilde{G}) = V(G) \setminus V(C_e) \cup \{c\}$ (c egy új pont, ami a kör össze csúcsát reprezentálja). $E(\tilde{G}) = E(G) \setminus \{C_e\}$ -n belüli élek}. $I(\tilde{G})$ természetes módon definiált.

$$\tilde{M} = M \cap E(\tilde{G}).$$

\tilde{F} : a C_e -n kívüli dupla ághajtásokkal felépülő kereső erdő.

Példa. Az első ábra a keresést és az e élt mutatja. A második ábra a C kört és annak a pontját mutatja. A harmadik ábrán a zsugorítás utáni helyzet látható. Végül azt mutatjuk meg, hogy a címkézés, hogyan terjed a zsugorítás után.



7. Lemma. (i) \tilde{M} párosítás \tilde{G} -ben

(ii) \tilde{F} keresőerdő \tilde{M} -ra

(iii) c külső pont \tilde{F} -ben.

A lemma egyszerű, nem bizonyítjuk. Ezek után már tárgyalhatjuk az Edmonds-algoritmus kihagyott esetét.

(Edmonds-eset) A megfelelő e él megtalálása után határozzuk meg a C_e kört. Zsugorítsuk ezt. A zsugorítás egy \tilde{G} gráfhoz, \tilde{M} párosításhoz és az erre vonatkozó \tilde{F} kereső erdőhöz vezet. Ezzel térjünk vissza a (Címkenövelés) lépéshez

Hol is tartunk? Hogyan néz ki az eddig leírt lépések végrehajtása? Az algoritmusunk „generikus futása” során egy

$$(G, M) = (G_0, M_0) \rightarrow (G_1, M_1) \rightarrow \dots \rightarrow (G_\ell, M_\ell)$$

zsugorítás sorozatot hajt végre, majd sikeres vagy sikertelen kereséssel leáll. A sikeres keresés esete sem világos. A látható javító út egy M_ℓ -re vonatkozó javító út G_ℓ -ben. Nekünk egy M -re vonatkozó javító út kell G -ben.

A teljességhez szükségesek a következők.

8. Tétel-hiány. *Legyen P javító út M_ℓ -re G_ℓ -ben. Ekkor létezik javító út M -re G -ben.*

Ez persze adódik abból, ha egy zsugorítás esetén a javító utat vissza tudjuk vetíteni. (A tételt a lemma ℓ -szeres iterált alkalmazása adja.)

9. Lemma-hiány. *Legyen P javító út \tilde{M} -ra \tilde{G} -ben. Ekkor létezik javító út M -re G -ben.*

10. Tétel-hiány. *Ha az Edmonds-algoritmus „Sikertelen kereséssel” áll le, akkor M optimális, azaz nem létezik rá vonatkozó javító út.*

Igazából a hiányzó lemma konstruktív bizonyítása szükséges. Azaz a lemma által garantált javító utat meg is kell konstruálnunk (és a programozónak ezt implementálni is kell az Edmonds-algoritmus megvalósításakor). A hiányzó tételekkel a következő órán foglalkozunk.