Eötvös Loránd Tudományegyetem
Természettudományi Kar

Farkas Dorottya Veronika

Matroidok és szintező algoritmusok

Szakdolgozat

Matematika BSc

alkalmazott matematikus szakirány

Témavezető:

Bérczi-Kovács Erika

Budapest, 2023

Tartalomjegyzék

Köszönetnyilvánítás

Elsősorban szeretném megköszönni témavezetőmnek, Bérczi-Kovács Erikának a téma megismertetését, minden szakmai kérdésemre adott válaszát, a figyelmes kommentjeit és állandó bátorítását és motiválását az elmúlt évek során.

Szeretném továbbá megköszönni összes hajdani tanáromnak a tőlük tanult ismereteket, szaktársaimnak pedig a közös tanulásokat, egymás segítését.

Végül szeretném megköszönni családomnak és barátaimnak a tanulányaim alatt nyújtott állandó támogatást.

1. fejezet

Bevezetés

Mennyi víz tud átfolyni egy csatornarendszerben 2 pont között, ha minden szakasznak ismerjük a kapacitását? Híres gráfelméleti feladat a megengedett folyam keresése, mely megválaszolja az előbbi kérdést. Ford és Fulkerson egy segédgráfokat használó javító utas algoritmust adott a problémára [2], amit Edmonds és Karp, valamint Dinits is tovább javított, így $O(n^2m)$ futásidejű algoritmust mutatott a problémára [9]. Ennél is hatékonyabbnak bizonyult a Goldberg és Tarjan által felfedezett szintező algoritmus: ebben egy lépés során csak lokális javításokat hajtunk végre, és $O(n^3)$ lépés elég hozzá [8].

Maximális folyam keresésén kívül a szintező megközelítéssel még sok más gráfelméleti feladatot meg tudunk oldani: például adott egy irányítatlan gráf és az éleket szeretnénk megirányítani úgy, hogy minden csúcsra adva van egy-egy alsó vagy felső korlát arra, hogy mennyi él futhat bele és mennyi indulhat ki belőle. Továbbá páros gráfban tudunk párosítást keresni, amely fedi az egyik vagy akár mindkét csúcshalmazt. Minden esetben lesz az algoritmusnak néhány közös vonása: minden feladatnál lesznek többletes vagy aktív csúcsok, amelyeket lokálisan javítani szeretnénk, illetve invariáns tulajdonságok, amelyek az algoritmus futása során végig teljesülnek. Az eljárásoksorán végig adott egy szintezése a csúcsoknak, és ha egy aktív csúcsot nem tudunk lokálisan javítani, akkor megemeljük a szintjét.

A dolgozat a szintező algoritmusnak olyan változataival foglalkozik, melyek matroidokhoz kapcsolódó optimalizálási kérdéseket oldanak meg. A matroid a lineáris algebrai függetlenség egy absztrakciója, melyben számos optimalizálási feladat jól kezelhető.

A 2. fejezet motivációja az a feladat, hogy k db diszjunkt feszítőfával legfeljebb hány élét tudjuk lefedni egy gráfnak. Ezt a problémát nemcsak gráfokon, hanem matroidokon fogjuk vizsgálni. A 2.1 részben megismerkedünk a matroid pontos fogalmával. A 2.2 alfejezetben a kiindulási problémát is megválaszoló 2.2.3 Edmonds-Fulkerson tételt mutatjuk be, amelynek bizonyítását a szintező algoritmussal fogjuk befejezni. A 2.3 pontban egy olyan algoritmussal fogunk dolgozni, ahol azok az elemek lesznek az aktívak, amelyek többször le vannak fedve. Ezzel szemben a 2.4 pontban a fedetlen elemek lesznek aktívak. Mindkét algoritmus ugyanazt bizonyítja be, de kicsit más szemlélettel. A 2.5 részben ezt a két algoritmust fogjuk összehasonlítani.

A 3. fejezetben arra keresünk választ, hogy minimális költségű feszítőfák között hogyan tudunk olyat találni, amely csúcsainak a maximális fokszáma a lehető legkisebb. Erre a kérdésre nem célunk pontos értéket keresni, mert NP-nehéz probléma, ezért csak egy approximációs algoritmust fogunk rá bemutatni. A 3.3 pontban kitekintünk arra, hogy ezt hogyan lehet bizonyos matroidokra kiterjeszteni.

2. fejezet

Az összegmatroid rangfüggvénye

A bevezetőben láttunk példákat hálózati folyamfeladatok megoldására szintező algoritmussal. Most ezt a szintező szemléletet szeretnénk átültetni egy általánosabb struktúrára, a matroidokra.

2.1. A matroid

A kombinatorikus optimalizálásban talán az egyik legismertebb és legegyszerűbb algoritmus a mohó-algoritmus: minden lépésben egy aktuálisan legkedvezőbb döntést hozunk, nem foglalkozunk vele, hogy a későbbiekben ezzel nem rontunk-e el valamit. Természetesen felmerül a kérdés, hogy ez milyen problémákra ad optimális megoldást? Vizsgáljuk $(\mathcal{S}, \mathcal{F})$ párost, ahol $\mathcal{S}$ egy véges alaphalmaz, $\mathcal{F}$ egy nem üres, leszálló halmazarendszere. Legyen egy tetszőleges c súlyfüggvényünk az alaphalmaz elemeire és terjesszük ki lineárisan $\mathcal{F}$ elemeire is: egy halmaz súlya legyen a benne szereplő elemek súlyainak összege. Ekkor $\mathcal{F}$ elemei közül a maximális súlyút pontosan akkor tudjuk mohó-algoritmussal megtalálni minden $c : \mathcal{S} \to \mathbb{R}$ súlyfüggvény mellett, ha $(\mathcal{S}, \mathcal{F})$ páros egy matroid [4].

Nézzünk meg egy konkrét példát, a Kruskal-algoritmust: adott egy gráf az éleivel, csúcsaival és egy az éleken definiált költségfüggvénnyel. Ezen szeretnénk keresni minimális összköltségű feszítőfát, ha összefüggő a gráf, és feszítő erdőt, ha nem. Ekkor optimális megoldást kapunk, ha mindig beveszünk egy legolcsóbb élet, amivel még nem keletkezik kör a kiválasztott élek által meghatározott gráfban. Mohónak nevezzük, mert minden él beválasztásánál csak arra figyelünk, hogy a legkisebb költségű élet vegyük be és nem vesszük számításba, hogy összességében ez optimális lesz-e. De könnyen belátható, hogy ez az algoritmus valóban optimális megoldását adja a feladatnak [2].

Egy másik megközelítési lehetőség a matroidok definiálására, amit mi is használni fogunk, ha a függetlenséget szeretnénk általánosítani: egy közös struktúrát adunk például egy vektortér vektorain értelmezett lineáris függetlenségnek és a gráfokon vett független éleknek (erdőknek).

Az alapfogalmak tisztázásához Frank András Matroidelmélet jegyzetének fejezeteit használom fel [6].

2.1.1. Definíció (Matroid). *Adott $\mathcal{S}$ véges halmaz és részhalmazainak egy $\mathcal{F}$ rendszere. Ekkor $M = (\mathcal{S}, \mathcal{F})$ páros **matroid**, ha $\mathcal{F}$ teljesíti a következő függetlenségi axiómákat:*

- *(I1) $\emptyset \in \mathcal{F}$*
- *(I2) ha $X \subseteq Y$ és $Y \in \mathcal{F} \Rightarrow X \in \mathcal{F}$*
- *(I3) $\forall\ X \subseteq \mathcal{S}$ részhalmazra az $\mathcal{F}$-nek X-ben fekvő, X-ben tartalmazásra nézve maximális részhalmazai azonos méretűek.*

Hogyan kapcsolódnak ide a gráfok? Minden gráfhoz tudunk természetes módon társítani egy matroidot, ahogy azt a 2.1 Kruskal-példában is láttuk:

2.1.2. Definíció (Grafikus matroid). *A $G = (V, E)$ gráfhoz tartozó **grafikus matroid** az $M = (\mathcal{S}, \mathcal{F})$ pár, ha $\mathcal{S} = E$ és $\mathcal{F} = \{G$ erdői$\}$.*

Vezessük be a matroidok használatához legszükségesebb fogalmakat: egy részük a lineáris algebrából, egy másik részük pedig a gráfelméletből származik.

2.1.3. Definíció. *$\mathcal{F}$ elemeit nevezzük **függetleneknek**, $\mathcal{S} \setminus \mathcal{F}$ elemeit pedig **függőnek**.*

*Egy $X \in \mathcal{S}$ halmaz maximális méretű független részhalmazának a mérete legyen az X **rangja**. Jel: $r(X)$. A független elemek rangja megegyezik a méretükkel.*

*Egy $X \subseteq \mathcal{S}$ halmaz **feszíti** $Y \subseteq \mathcal{S}$ halmazt, ha $r(X \cup Y) = r(X)$.*

*A matroid egy legnagyobb méretű független elemét **bázisnak** hívjuk. Jel: B egy bázis, $\mathcal{B}$ a matroid bázisainak halmaza. Ekkor $r(B) = r(S)$.*

*Egy tartalmazásra nézve minimális halmaz **vágás**, ha minden bázist metsz.*

*$C \in \mathcal{S}$ **kör**, ha*

- *$C \notin \mathcal{F}$*
- *$C \setminus \{x\} \in \mathcal{F}\ \forall\ x \in C$.*

Jel: C egy kör, $\mathcal{C}$ a matroid köreinek halmaza.

2.1.4. Megjegyzés. *Nézzük meg mit jelentenek ezek a fogalmak grafikus matroidokon! A rang egy adott élhalmazban a maximálisan kiválasztható független élek számát jelenti. A bázisok összefüggő gráfból képzett grafikus matroid esetén feszítő fák, több összefüggőségi komponens esetén pedig feszítő erdők lesznek. Vágásoknak a gráf elemi vágásait mondjuk, körök pedig egyszerűen az eredeti gráf körei lesznek (angolban jellemzően a "circuit" kifejezést használjuk a matroidokon vett körökre és "cycle" kifejezést a gráfok köreire).*

Egy matroid független halmazait a függetlenségi axiómáktól eltérő módon is lehet definiálni.

2.1.5. Állítás. *[6]*

1. *Adott egy $M = (\mathcal{S}, \mathcal{F},)$ matroid, ekkor a független halmazok halmaza a rangfüggvénnyel is megadható:*

$$\mathcal{F} = \{F \subseteq \mathcal{S} : r(F) = |F|\}.$$

2. *A matroidok rangfüggvényére igazak az alábbiak:*

   (R1) üres halmaz rangja nulla: $r(\emptyset) = 0$

   (R2) monoton: $X \subseteq Y \subseteq \mathcal{S} \Rightarrow r(X) \leq r(Y)$

   (R3) szubkardinális: $r(X) \leq |X|$

   (R4) szubmoduláris: $r(X) + r(Y) \geq r(X \cup Y) + r(X \cap Y)$

*Ha egy $r : 2^{|\mathcal{S}|} \to \mathbb{Z}_+$ halmazfüggvény teljesíti $(R1) - (R4)$-et, akkor egy matroidot ír le, amit $M = (\mathcal{S}, r)$-rel jelölünk és ezért ezeket a tulajdonságokat **rangaxiómáknak** hívjuk.*

A rangfüggvényhez hasonlóan, ha a matroid bázisait vagy köreit ismerjük, azokból is tudjuk definiálni a független halmazokat:

2.1.6. Állítás. *[6]*

1. *Adott egy $M = (\mathcal{S}, \mathcal{F},)$ matroid, ekkor a független halmazok halmaza a bázisokkal is megadható:*

$$\mathcal{F} = \{F \subseteq \mathcal{S} : \exists B \in \mathcal{B},\ hogy\ F \subseteq B\}.$$

2. *A matroidok bázisai teljesítik a következő tulajdonságokat:*

(B1) $\mathcal{B}$ *nemüres*

(B2) kicserélési axióma: $B_1, B_2 \in \mathcal{B}$ *és* $x_1 \in B_1 - B_2 \Rightarrow \exists\ x_2 \in B_2 \setminus B_1 :$ $B_1 \setminus \{x_1\} \cup \{x_2\} \in \mathcal{F}$

Ha $\mathcal{B}$ *teljesíti* $(B1), B(2)$*-t, akkor egy matroidot határoz meg, amit* $M = (\mathcal{S}, \mathcal{B})$*-vel jelölünk és ezért ezeket a tulajdonságokat* ***bázisaxiómáknak*** *hívjuk.*

2.1.7. Állítás. *[6]*

1. *Adott egy* $M = (\mathcal{S}, \mathcal{F},)$ *matroid, ekkor a független halmazok halmaza a köreivel is megadható:*

$$\mathcal{F} = \{F \subseteq \mathcal{S} : \nexists C \in \mathcal{C},\ hogy\ C \subseteq F\}.$$

2. *A matroidok körei teljesítik a következő tulajdonságokat:*

(C1) $\emptyset \notin \mathcal{C}$

(C2) $C_1, C_2 \in \mathcal{C} \Rightarrow C_1 \not\subset C_2$

(C3) erős köraxióma: $C_1, C_2 \in \mathcal{C}, e \in C_1 \cap C_2, f \in C_1 \setminus C_2 \Rightarrow \exists\ C \in \mathcal{C},$ *hogy* $f \subseteq C \subseteq C_1 \cup C_2 \setminus e$

Ha egy $\mathcal{C}$ *teljesíti* $(C1) - (C3)$*-t, akkor megad egy matroidot, amit* $M = (\mathcal{S}, \mathcal{C})$*-vel jelölünk és ezeket a tulajdonságokat* ***köraxiómáknak*** *hívjuk.*

A következő könnyen meggondolható állítások későbbi bizonyításoknál lesznek hasznosak:

2.1.8. Állítás. *A maximális / minimális súlyú bázisok teljesítik a bázistulajdonságokat.*

2.1.9. Állítás. *Ha* F *független és* $e \in \mathcal{S} \setminus F$*, akkor* $F \cup \{e\}$ *legfeljebb egy kört tartalmaz. Ezt az* e *elem* F *halmazához tartozó alapkörének hívjuk és* $C(F, e)$*-vel jelöljük.*

Orákulumokkal adott matroid

Hogyan tudunk egy algoritmus inputjában jól megadni egy matroidot? Ha megadjuk az $\mathcal{S}$ alaphalmazt és felsoroljuk $\mathcal{F}$ elemeit az nem mindig szerencsés, mert utóbbi gyakran exponenciális számú az alaphalmaz méretével kifejezve. Ezért nem adjuk meg konkrétan a matroidot, hanem egy **függetlenségi orákulumot** fogunk használni, mely az alaphalmaz bármely részhalmazáról konstans időben megmondja, hogy eleme-e a független elemek halmazának. Korábban láttuk (2.1.5), hogy egy matroidot nem csak a függetlenjei felsorolásával, hanem bázisaival, köreivel vagy rangfüggvényével is megkaphatunk. Ezekben az esetekben érdemes hozzájuk tartozó orákulumokat használni: egy **bázis orákulum** megmondja az alaphalmaz egy tetszőleges részhalmazáról, hogy bázis-e. Bár azt érdemes megjegyezni, hogy ebből nem tudunk polinomiális számú lépésben függetlenségi orákulumot előállítani, ahhoz **erős bázis orákulumra** lesz szükségünk, melyben meg van adva egy adott bázis a döntési szubrutin mellett. A **rang orákulum** ekvivalens a függetlenségivel, de a **kör orákulumból** megint nem tudunk függetlenségit csinálni (a függetlenségiből tudunk kört). Ezentúl amikor azt mondjuk, hogy adott egy matroid mindig arra gondolunk, hogy van egy alaphalmaz és egy orákulum az előbb felsoroltak közül[6].

2.2. Motiváció: az összegmatroid rangja

Képzeljük el a következő gráfokkal modellezhető problémát: néhány város között szeretnénk úthálózatot építeni, összesen k diszjunkt feszítőfát határozzunk meg a városok és utak által generált gráfban. A munkát szeretnénk párhuzamosítani, ezért mindegyik feszítőfára egy külön céget bízunk meg, akik akár különböző költségfüggvényeket is adhatnak az útszakaszok megépítésére. Ekkor ha létezik olyan megoldás, hogy minden cégnél egy legolcsóbb feszítőfát építtetünk és azok éldiszjunktak, akkor meg szeretnénk kapni egy ilyen optimális beosztást. Egyelőre költségfüggvények nélkül keressünk megoldást a feladatra.

2.2.1. Definíció (Összegmatroid)**.** *Legyen $\mathcal{S}$ közös alaphalmaz, $M_1 = (\mathcal{S}, \mathcal{F}_1), ..., M_k = (\mathcal{S}, \mathcal{F}_k)$ matroidok.* ***Partícionálható halmazoknak*** *hívjuk $\mathcal{S}$ azon részhalmazait, amelyek előállnak az M_i matroidokból vett egy-egy független halmaz egyesítéseként. Jel: F_Σ.*

2.2.2. Tétel. *[6] Legyen $M_1, ..., M_k$ matroidok közös alaphalmaza S. Ekkor F_Σ partícionálható halmazok rendszere teljesíti a függetlenségi axiómákat, vagyis matroidot határoz meg.*

Ekkor ha ezeket az M_i matroidokat úgy adjuk meg, hogy az alaphalmaz a gráf éleinek a halmaza és a bázisok a feszítőfák, akkor az alaphalmaz rangja a legnagyobb partícionálható halmaz mérete lesz. Ez pont arra a kérdésre válaszol, hogy legfeljebb hány különböző utat tudunk megépíteni k db feszítőfával. Ha ez a szám $k(|V| - 1)$, ahol V a gráf csúcsainak halmaza, akkor létezik k db diszjunkt feszítőfa.

2.2.3. Tétel (Edmonds – Fulkerson)**.** *[3] Az összegmatroid rangfüggvényét a következő összegformula adja meg:*

$$r_\Sigma(S) = \min_{X \subseteq Z}\{|Z - X| + \sum_i r_i(X)\}.$$

Bizonyítás. Legyen $\mathcal{B}_i$ az M_i matroid bázisrendszere, mind a k matroidból vegyünk egy $B_i \in \mathcal{B}_i$ bázist. Legyen F ezen k bázis uniója, ezen halmaz maximális mérete a kérdés, vagyis $r_\Sigma(S) = \max_F |F|$. Azaz a fenti tétel egy min-max tétel. Most csak a könnyebb irányt, $max \leq min$ irányt fogjuk bizonyítani, a másik irány később, a 2.3 algoritmusból fog adódni. Legyen $Z \in \mathcal{S}$ tetszőleges, ekkor:

$$|F| = |F \cap Z| + |F \setminus Z|$$

$$|F \setminus Z| \leq |S \setminus Z|, \text{ mivel } F \subseteq S$$

$$|F \cap Z| \leq \sum_i |B_i \cap Z|, \text{ hiszen } F = \cup B_i$$

$$\sum_i |B_i \cap Z| \leq \sum_i r_i(Z),$$

mert Z-ben lehetnek még B_i-n kívül elemek, amikkel nagyobb méretű független részhalmazt kaphatunk, mint a metszet (ami biztosan független, hiszen egy bázis részhalmaza). Összességében:

$$|F| = |F \cap Z| + |F \setminus Z| \leq \sum_i |B_i \cap Z| + |S \setminus Z| \leq \sum_i r_i(Z) + |S \setminus Z|,$$

vagyis $max \leq min$ teljesül.

2.2.4. Lemma (Optimalitási feltételek)**.** *[6] Egyenlőséghez, optimalitáshoz a következő szükséges és elégséges feltételek kellenek:*

1. $\mathcal{S} \setminus Z \subseteq \cup_i B_i$
2. $B_i \cap B_j \cap Z = \emptyset$ *minden* $1 \leq i < j \leq k$*-ra*
3. $B_i \cap Z$ *feszíti* M_i*-ben* Z*-t minden* $i = 1, ..., k$*-ra.*

Bizonyítás.

1. $F \setminus Z = \mathcal{S} \setminus Z$, ekkor a méretük is megegyezik: $F \setminus Z \subseteq S \setminus Z$ mindig teljesül, az egyenlőséghez még $\mathcal{S} \setminus Z \subseteq \cup_i B_i$ szükséges és elégséges
2. $|F \cap Z| = \sum_i |B_i \cap Z|$-hez az kell, hogy ne legyenek ismétlődő elemek, ehhez szükséges és elégséges: $B_i \cap B_j \cap Z = \emptyset$ minden $1 \leq i < j \leq k$-ra

3. $|B_i \cap Z| = r_i(Z)$ minden i-re, akkor az összegük is megegyezik, ez pedig akkor és csak akkor lesz igaz, ha $B_i \cap Z$ feszíti (2.1.3) M_i-ben Z-t minden i-re .

□

Ilyen, az optimalitási feltételeknek eleget tevő B_i bázisok és $Z \subseteq S$ halmaz megtalálására szolgál a szintező algoritmus, amit a következő fejezetben vizsgálunk.

□

2.3. A többletes csúcsokat emelő szintező algoritmus

Összefoglalva az eddigieket: van egy $\mathcal{S}$ alaphalmazunk, ezen k matroid. A célunk az összegmatroid rangjának algoritmikus meghatározása a korábban bemutatott 2.2.3 tétel bizonyításával:

$$r_\Sigma(S) = \min_{X \subseteq Z}\{|Z - X| + \sum_i r_i(X)\}.$$

A $max \leq min$ irányt már beláttuk és onnan kaptuk a 2.2.4 lemmában szereplő optimalitási feltételeket. Most a $max \geq min$ irányhoz a következő a feladatunk: adott $\mathcal{S}$ alaphalmaz, ezen $M_1, ..., M_k$ matroidok, és szeretnénk megtalálni B_i bázisokat és $Z \subseteq S$ halmazt, melyek eleget tesznek a 2.2.4 lemmában felsorolt feltételeknek .

2.3.1. Építőkövek

Az algoritmushoz először szedjünk össze néhány fogalmat és kisebb lépést. Az $\mathcal{S}$ alaphalmazon adottak $M_1, ..., M_k$ matroidok a bázisaikkal. Ekkor

2.3.1. Definíció. *Egy $h : \mathcal{S} \to \{0, 1, ..., n\}$ függvényt **szintfüggvénynek** nevezünk, ahol $n = |\mathcal{S}|$.*

*Adott $B_i \in \mathcal{B}_i, 1 \leq i \leq k$ bázisokra nézve egy $t \in \mathcal{S}$ elem legyen **többletes**, ha egynél több bázisban is szerepel. Egy $s \in \mathcal{S}$ elem **fedetlen**, ha egy bázisban sincs benne.*

2.3.2. Definíció (Megengedett csere)**.** *Egy $(s, t) \in \mathcal{S}$ pontpár **megengedett csere** a B_i bázisban egy adott szinthalmazra, ha teljesíti a következő feltételeket:*

- *$s \in \mathcal{S} \setminus B_i$ valamint $t \in C(B_i, s)$, ahol C az s alapköre (2.1.9)*
- *$h(t) = h(s) + 1$*

Egy t ponthoz tartozó megengedett cseréket a B_i bázisban jelöljük $S_t^{B_i}$-vel.

2.3.3. Definíció (megállási feltételek)**.** *Az alábbi eseteket nevezzük* ***megállási feltételeknek****:*

(A) minden elem lefedett

(B) üres szint keletkezik (ekkor felette minden pont fedett (2.3.8))

(C) nincs több többletes elem, vagyis diszjunktak a bázisok

2.3.2. Az algoritmus lépései

Induljunk ki egy adott $\mathcal{S}$ alaphalmazból, $M_1, ..., M_k$ matroidokból, ezeken $B_1, ...B_k$ bázisokból és h azonosan 0 szintfüggvényből.

Ha a bázisok diszjunktak, akkor $Z := \mathcal{S}$, ha pedig $\mathcal{S}$ minden eleme le van fedve, akkor $Z = \emptyset$ teljesíti az optimalitási feltételeket. Különben az alábbiak szerint folytatódik az algoritmus:

Egy általános lépés: legyen t a legalsó többletes elem (2.3.1). Ha valamelyik őt tartalmazó bázisban létezik t-nek megengedett cseréje (2.3.2), vagyis $\exists\ (s, t) \in S_t^{B_i}$, akkor t-t cseréljük ki s-re ebben a bázisban. Ha egyik bázisban sincs megengedett cseréje, vagyis nem tudjuk csökkenteni t többletét, akkor emeljük meg eggyel a szintjét.

Az algoritmus véget ér, ha bekövetkezik valamelyik megállási feltétel (2.3.3), vagyis egy elem emelésekor üres szint keletkezik, ekkor Z legyen az üres szint alatti elemek halmaza, vagy bekövetkezik valamelyik speciális eset: minden elem le van fedve, vagy nincs több többletes pont.

Algorithm 1 Többletes csúcsokat emelő szintező

Input: $\mathcal{S}, M_1, ..., M_k$

Output: (B, Z)

```
B := B_1, ..., B_k tetszőleges
h(v) := 0 minden v ∈ S
while nincs üres szint do
    if nincs többletes pont then
        Z := S
        return (B, Z)
    else if nincs fedetlen pont then
        Z := ∅
        return (B, Z)
    else
        t := legalsó többletes pont
        if létezik B_i bázis és benne megengedett (s,t) csere then
            CSERE(B_i; (s, t))
        else
            h(t)++
Z := {v : h(v) < h(t)}
return (B, Z)
```

2.3.3. Az algoritmus helyessége

Az algoritmus során használjunk egy szubrutint, ami megad egy megengedett cserét. Ennek futásideje legyen γ. Ekkor:

2.3.4. Lemma. *[5] Az algoritmus véges, lépésszáma $O(\gamma n^3)$.*

Bizonyítás. Ha egy elem eléri az n. szintet, akkor keletkezik egy üres szint, mivel $n+1$ szint és n elem van. Ez legfeljebb n^2 szintemelésen belül megtörténik.

Nevezzük fázisnak az algoritmus futásának egy olyan részét, ami alatt nem változik a bázisok uniójának mérete és szintemelés sem történik. Báziscserénél az unió mérete nem változik, ha olyan elemre cseréltünk, ami valamelyik másik bázisban szerepelt, vagy eggyel nő, ha eddig nem fedett elemet veszünk be. Legfeljebb n bázisnövelés történik, hiszen $|\mathcal{S}| = n$. Így legfeljebb $n^2 + n$ fázis van, mivel legfeljebb n^2 szintemelés és n bázisnövelés lehet.

Egy fázis során legfeljebb n báziscsere történhet, hiszen a legalsó többször fedett elem szintje mindig eggyel csökken a fázison belül. Így a báziscserék száma legfeljebb $O(n \cdot (n^2 + n)) = O(n^3)$.

1. sor vizsgálata: kiindulási bázisok megtalálása. Egy adott matroidban mohó módon tudunk találni egy bázist. Mindig beveszünk egy elemet, amivel még független marad, ezt függetlenségi orákulummal tudjuk ellenőrizni (2.1), ez $O(n)$ és k alkalommal kell elvégezni.

3. sor vizsgálata: van-e üres szint. Tároljuk duplán láncolt listákban az egyes szinteken lévő pontokat, szintemeléskor a megemelt elemet vegyük ki a régi listájából és tegyük bele az újba. Legyen t a legutóbbi legalsó többletes elem. Csak szintemeléskor keletkezhet üres szint, így olyankor mindig meg kell nézni, hogy t eredeti szintjének listája üres-e. Ez minden szintemeléshez egy konstans lépést ad hozzá.

7. sor vizsgálata: fedetlen pont van-e még. Az elején minden elemre nézzük meg, hogy fedetlen-e, állítsunk be hozzájuk egy-egy igaz-hamis változót, valamint azt is tároljuk el, hogy összesen mennyi fedetlen elem van. Ha egy elemet egyszer lefedtünk, akkor többet már nem lesz fedetlen. Minden báziscserénél vizsgáljuk meg, hogy fedetlen elemet cserélünk-e be és ha igen, akkor állítsuk át a változóját, a számlálót pedig csökkentsük. Ha az összeg valamikor 0 lesz, akkor nincs több fedetlen elemünk, ekkor az algoritmus leáll. Az elején az n pont vizsgálata $O(n)$, utána viszont csak a báziscsere művelethez adtunk hozzá egy konstans lépést.

4. sor vizsgálata: van-e még többletes pont. Az elején minden elemhez rendeljünk egy duplán láncolt listát, amiben az őket tartalmazó bázisokat soroljuk fel. Kapjanak egy változót arra is, hogy aktívak-e, legyen egy összegünk is a fedetlenhez hasonlóan. Ez $O(n)$ lépés. Minden báziscserénél t listájából kivesszük az adott bázist, s listájába beletesszük. Ha t már nem lesz többletes, akkor állítsuk át a változóját és a számlálót csökkentsük eggyel. Ha s most válik többletessé, akkor pedig növeljük eggyel a számlálót. Ha valamikor 0, akkor álljunk le. Ez a báziscserékhez egy konstans idejű lépést ad hozzá a.

11. sorvizsgálata: legalsó többletes pont megtalálása. Vizsgáljuk meg először, hogy egy fázison belül mennyibe kerül összesen a legalsó többletes elemek megtalálása. Mivel nem nő a bázisok uniójának mérete, ezért mindig már lefedett elemre cseréltünk, vagyis a becserélt elem biztosan többletes lesz, és mivel eggyel lejjebb van, mint t, ezért az lesz az új legalsó többletes elem. Ez minden báziscserénél egy konstans idejű lépés. Fázisok között első esetben akkor váltunk, ha nőtt az unió, vagyis egy addig fedetlen elem kerül be, annak egyszeres lesz a lefedettsége, tehát nem többletes. Ekkor először t-t ellenőrizzük, ha ő már nem többletes akkor az ő

szintjén kell továbbmennünk a listában utána következő elemeken. Ha egyik sem többletes, akkor ugrunk egy szintet és így tovább. Legfeljebb n lépésben megtaláljuk a következő legalsó többletes pontunkat. Fázisváltás másik oka az lehet, ha megemeltük t szintjét. Ekkor az ő eredeti szintjétől kell az előző esethez hasonlóan végignézni a pontokat. Mindkét esetben legfeljebb n lépésben megtaláltuk két fázis között a következő legalsó többletes csúcsot, és összesen n^2+n fázis lehet, így $O(n^3)$ lépés elegendő.

Összességében tehát az algoritmus lépészáma $O(\gamma n^3)$.

$\square$

2.3.5. Definíció (Invariáns tulajdonságok)**.**

(α) $u \in \mathcal{S} \setminus B_i, v \in C(B_i, u)$ esetén $h(v) \leq h(u) + 1$

(β) $u \in \mathcal{S} \setminus \cup_i B_i$ esetén $h(u) = 0$, azaz minden fedetlen pont szintje 0

2.3.6. Tétel. *[6] Az algoritmus során az invariáns tulajdonságok (2.3.5) végig teljesülnek.*

Bizonyítás. (α) $u \in \mathcal{S} \setminus B_i, v \in C(B_i, u)$ esetén $h(v) \leq h(u) + 1$ tulajdonság végig teljesül. Először nézzük meg szintemelésnél mi történik. Egy v többletes csúcsot akkor emelünk fel, ha nincs megengedett cseréje (2.3.2), vagyis egy tetszőleges $u \in \mathcal{S} \setminus B_i, v \in C(B_i, u)$-ra $h(u) \neq h(v) - 1$, de mivel (α) eddig teljesült, ezért $h(u) \geq h(v) - 1$, így összességében $h(u) \geq h(v)$, ekkor ha v szintjét megemelejük eggyel, akkor is teljesül, hogy $h(v) + 1 \leq h(u) + 1$.

Ahhoz, hogy báziscsere után is fennálljon (α) a következő lemmát lássuk be:

2.3.7. Lemma. *[6] Legyen B az $M = (\mathcal{S}, \mathcal{B})$ bázisaival adott matroid (2.1.6) egy bázisa, h szintfüggvény (2.3.1), amely teljesíti a következő tulajdonságot:*

$$u \in \mathcal{S} \setminus B, v \in C(B, u) \text{ esetén } h(v) \leq h(u) + 1 \qquad (\alpha)$$

Legyenek $s \in \mathcal{S} \setminus B$ és $t \in C(B, s)$ olyan elemek, melyekre $h(t) = h(s) + 1$ teljesül. Ekkor az (s, t) cserével kapott bázisra, vagyis $B' := B \setminus \{t\} \cup \{s\}$-re is teljesülni fog ($\alpha$).

Bizonyítás. Legyen $C_u := C(B, u)$ és $C'_u := C(B', u)$ egy tetszőleges $u \in \mathcal{S} \setminus (B \cup B')$ elemhez tartozó alapkörök (2.1.9) B és B' bázissal. Egy $X \subseteq B$ halmaz szintje legyen $h(X) := \max_{v \in X}\{h(v)\}$. Ekkor ($\alpha$) átfogalmazható a következőre:

$$h(C_u) \leq h(u) + 1.$$

Ha $u = t$, akkor $C'_u = C'_t = C_s$, vagyis

$$h(C'_u) = h(C'_t) = h(C_s) = h(t) < h(u) + 1.$$

Különben ha $u \in \mathcal{S} \setminus B'$, $u \neq t$, ekkor ha $t \in C_u$, akkor

$$h(C_s) = h(t) \leq h(u) + 1, \text{ vagyis } h(C_u \cup C_s) \leq h(u) + 1.$$

Az erős köraxióma (2.1.7) miatt létezik egy olyan C' kör, ami tartalmazza u-t és $C' \subseteq (C_s \cup C_u) \setminus t$, ahol $C'_u = C'$ szükséges, tehát

$$h(C'_u) \leq h(C_u \cup C_s) \leq h(u) + 1.$$

Ha pedig $t \notin C_u$, akkor $C'_u = C_u$, vagyis

$$h(C'_u) = h(C_u) \leq h(u) + 1.$$

□

Tetszőleges B_i bázisra igaz lesz a lemma, így (α) teljesül báziscsere után is.

(β) $u \in \mathcal{S} \setminus \cup_i B_i$ esetén $h(u) = 0$, azaz minden fedetlen pont szintje 0 tulajdonság bizonyítása:

Kezdetben minden pont szintje 0. Egy elemet csak akkor emelünk fel, ha többletes és nem tudjuk csökkenteni a lefedettségét. Csak addig csökkentjük, amíg többletes, vagyis ha egyszer már lefedtük, utána soha nem lesz lefedetlen. Így (β) teljesül.

□

2.3.8. Állítás. *[6] Leálláskor az optimalitási feltételek (2.2.4) teljesülni fognak.*

Bizonyítás. Az áttekinthetőség kedvéért az optimalitási feltételek újra:

1. $\mathcal{S} \setminus Z \subseteq \cup_i B_i$
2. $B_i \cap B_j \cap Z = \emptyset$ minden $1 \leq i < j \leq k$-ra
3. $B_i \cap Z$ feszíti (2.1.3) M_i-ben Z-t minden $i = 1, ..., k$-ra.

Először is a speciális eseteket vizsgáljuk. Ha a bázisok diszjunktak, vagyis nincs többletes (2.3.1) elem, akkor az algoritmus leáll és $Z := \mathcal{S}$. Ekkor:

1. $\emptyset \subseteq \cup_i B_i$ $\surd$
2. $B_i \cap B_j \cap \mathcal{S} = \emptyset$ $\surd$, mivel diszjunktak

3. $B_i \cap \mathcal{S}$ feszíti M_i-ben $\mathcal{S}$-t minden $i = 1, ..., k$-ra $\surd$.

Ha minden elem le van fedve, akkor az algoritmus leáll és $Z := \emptyset$. Ekkor:

1. $\mathcal{S} \subseteq \cup_i B_i$ $\surd$, hiszen minden elem le van fedve
2. $B_i \cap B_j \cap \emptyset = \emptyset$ $\surd$, mivel üres halmaz metszete mindennel üres halmaz
3. $B_i \cap \emptyset$ feszíti M_i-ben $\mathcal{S}$-t minden $i = 1, ..., k$-ra, $\surd$ hiszen üres halmaz rangja mindenképp 0 (2.1.5).

Most nézzük meg általános helyzetben, amikor t legalsó többletes csúcs emelésével a j. szint üressé válik. Legyen j üres szint mellett $Z := \{u : h(u) \leq j\}$. Ekkor:

1. $\mathcal{S} \setminus Z \subseteq \cup_i B_i$ $\surd$, mivel a lefedetlen pontok a 0. szinten vannak (β) (2.3.5) miatt, az üres szint felett minden fedve lesz.
2. $B_i \cap B_j \cap Z = \emptyset$ minden $1 \leq i < j \leq k$-ra $\surd$: mivel t volt a legalsó többletes, ezért alatta nincsen másik és mellette sem lehet, hiszen az ő felemelésével üres szint keletkezett, szóval senki nem lehetett vele egy szinten. Így az üres szint alatt minden pont legfeljebb egy bázisban lehet.
3. $B_i \cap Z$ feszíti M_i-ben Z-t minden $i = 1, ..., k$-ra $\surd$, vagyis kell: $r(B_i \cap Z) = r(Z)$

   (α) (2.3.5) miatt minden $u \in \mathcal{S} \setminus B_i$-re igaz, hogy $C(B_i, u)$ minden eleme legfeljebb eggyel van u fölött. Vegyünk egy u-t, ő az üres szint alatt van, ezért az egész alapkörének is alatta kell lennie, különben valamelyik eleme legalább 2-vel lenne fölötte. Így $C(B_i, u) \subseteq Z$ $\forall$ $u \in Z \setminus B_i$, ez azt jelenti, hogy $Z \setminus B_i$ egyik tagja sem tudja növelni Z rangját.

$\square$

2.3.9. Megjegyzés. *Speciálisan ha mind a k matroidon különböző költségfüggvényeket definiálunk és nem csak bázisokat, hanem minimális költségű bázisokat keresünk, arra is alkalmazható ez az algoritmus.*

A 2.1.8 állítás alapján a minimális költségű bázisok is teljesítik a bázis tulajdonságokat. A megengedett cserénél (2.3.2) azt is el kell várnunk, hogy a két elem költsége ugyanannyi legyen a bázison belül, de ezen kívül más változtatásra nincs szükség. Így azt a speciális feladatot is meg tudjuk oldani, amit a 2.2 fejezet elején vetettünk fel. Szeretnénk építeni k db diszjunkt feszítőfát k különböző céggel úgy, hogy mindegyik külön megmondja nekünk, hogy egy-egy útszakasz mennyibe kerül. Természetesen minden cégnek a lehető legkevesebbet szeretnénk fizetni, de minél több különböző utat szeretnénk megépíttetni. Ekkor az algoritmusunk megadja, hogy melyik céggel melyik utakat építtessük meg. Ha nem tudunk k db diszjunkt legolcsóbb feszítőfát építeni, akkor az összegmatroid rangja kevesebb lesz $k(|V| - 1)$-nél, ahol V a gráf csúcsainak halmaza.

2.4. A fedetlen elemeket emelő szintező algoritmus

Ebben a fejezetben az előzőhöz nagyon hasonló, néhány pontban mégis különböző algoritmust fogunk vizsgálni, amit Frank András és Miklós Zoltán Simple Push-Relabel Algorithms For Matroids And Submodular Flows című cikkükben írnak le [5]. Továbbra is Edmonds és Fulkerson matroid partíciós tételét (2.2.3) szeretnénk bizonyítani. Emlékeztető:

[Edmonds – Fulkerson] Az összegmatroid rangfüggvényét a következő összegformula adja meg:

$$r_\Sigma(S) = \min_{X \subseteq Z}\{|Z - X| + \sum_i r_i(X)\}.$$

Azt már megállapítottuk, hogy egyenlőség pontosan akkor van, ha a B_i bázisok és Z halmaz teljesíti a következő optimalitási tulajdonságokat (2.2.4):

1. $\mathcal{S} \setminus Z \subseteq \cup_i B_i$
2. $B_i \cap B_j \cap Z = \emptyset$ minden $1 \leq i < j \leq k$-ra
3. $B_i \cap Z$ feszíti M_i-ben Z-t minden $i = 1, ..., k$-ra.

Ezeknek a halmazoknak a megtalálására szolgált előbb a szintező algoritmus. Most erre szeretnénk egy másik változatot megnézni. Kezdjük megint az építőkövekkel, amikből összerakjuk az egész algoritmust. *-gal jelöljük, ahol valamilyen eltérés van az előzőhöz (2.3.1) képest.

2.4.1. Építőkövek

A 2.3.1 definíció fogalmain nem változtatunk:

Egy $h : \mathcal{S} \to \{0, 1, ..., n\}$ függvényt **szintfüggvénynek** nevezünk, ahol $n = |\mathcal{S}|$.

Adott $B_i \in \mathcal{B}_i, 1 \leq i \leq k$ bázisokra nézve egy $t \in \mathcal{S}$ elem legyen **többletes**, ha egynél több bázisban is szerepel. Egy $s \in \mathcal{S}$ elem **fedetlen**, ha egy bázisban sincs benne.

2.4.1. Definíció (megengedett csere*)**.** *Egy (s, t) pontpár* ***megengedett csere**** *a B_i bázisban, ha teljesíti a következő feltételeket:*

- $s \in \mathcal{S} \setminus B_i$ *valamint* $t \in C(B_i, s)$*, ahol C a s alapköre (2.1.9)*
- $h(t) = h(s) - 1$

Egy s ponthoz tartozó megengedett cseréket a B_i bázisban jelöljük $S_s^{*B_i}$-vel.*

2.4.2. Definíció. *[Megállási feltételek*] Az algoritmus leáll, ha az alábbiak közül bármelyik bekövetkezik:*

(A) minden elem lefedett, vagyis $S = \cup_i B_i$

(B) üres szint keletkezik (ekkor alatta minden pont fedett)*

(C) nincs több többletes elem, vagyis diszjunktak a bázisok

2.4.3. Megjegyzés. *A [5] cikkben valójában nem szerepel megállási feltételként, hogy nincs több többletes elem, de nem rontja el az algoritmusunkat, hiszen ha már nincs többletes elem, akkor diszjunktak a bázisok, így az unió maximális. A futásidőn sem változtat lényegileg, viszont így könnyebben össze tudjuk hasonlítani a két algoritmust a 2.5 fejezet 2.5.5 állításában.*

2.4.2. Az algoritmus lépései

Induljunk ki $B_1^*, ...B_k^*$ bázisokból és h azonosan 0 szintfüggvényből.

Ha a bázisok diszjunktak, akkor $Z^* := \mathcal{S}$, ha pedig $\mathcal{S}$ minden eleme le van fedve, akkor $Z^* = \emptyset$ teljesíti az optimalitási feltételeket. Különben az alábbiak szerint folytatódik:

Egy általános lépés: válasszunk ki egy s legalacsonyabb szinten lévő fedetlen (2.3.1) elemet. Ha van olyan B_i^* bázis, amiben van (s, t) megengedett cseréje* (2.4.1), akkor t-t cseréljük ki s-re ebben a bázisban. Ha egyik bázisban sincs megengedett cseréje*, vagyis nem tudjuk lefedni s-t, akkor emeljük meg eggyel a szintjét.

Az algoritmus véget ér, ha bekövetkezik valamelyik megállási feltétel, vagyis ha az egész alaphalmaz le van fedve, diszjunktak a bázisok vagy keletkezik egy üres szint, ekkor Z legyen az üres szint feletti elemek halmaza.

Algorithm 2 Fedetlen elemeket emelő szintező

Input: $\mathcal{S}, M_1, ..., M_k$

Output: (B, Z^*)

$B := B_1, ..., B_k$ tetszőleges
$h(v) := 0$ minden $v \in \mathcal{S}$
while nincs üres szint **do**
  if nincs többletes pont **then**
    $Z := S$
    return (B, Z)
  else if nincs fedetlen pont **then**
    $Z := \emptyset$
    return (B, Z)
  else
    $s :=$ legalsó fedetlen elem
    if létezik B_i bázis és benne megengedett (s,t) csere* **then**
      CSERE$(B_i; (s, t))$
    else
      h(s)++
$Z := \{v : h(v) > h(s)\}$
return (B, Z)

2.4.3. Az algoritmus helyessége

A 2.4.2 algoritmusnál bizonyított állításokkal analóg feltételekre van szükségünk, amelyek belátása teljesen azonos gondolatokon alapul, így nem fejtjük ki őket még egyszer.

A megengedett cserék * megtalálásához ismét használjunk egy szubrutint, aminek futásideje γ.

2.4.4. Állítás. *[5] Az algoritmus* véges, lépésszáma $O(\gamma n^3)$.*

Bizonyítás. Mivel az algoritmus lépései nagyon hasonlóak a 2.4.2 algoritmushoz, ezért a 2.3.4 állítás alapján analóg módon belátható. □

2.4.5. Definíció (Invariáns tulajdonságok*)**.**

(α)* $u \in \mathcal{S} \setminus B_i,\ v \in C(B_i, u)$ *esetén* $h(v) \geq h(u) - 1$

(β) $h(u) = 0$, minden $u \in \mathcal{S}$ többletes pontra*

2.4.6. Állítás. *[5] Az algoritmus* során az invariáns tulajdonságok* (2.4.5) végig teljesülnek.*

Bizonyítás. A 2.3.6 állítással analóg módon bizonyítható. □

2.4.7. Állítás. *[5] Leálláskor az optimalitási feltételek* (2.2.4) teljesülni fognak.*

Bizonyítás. Ha a 2.4.2 (A) megállási feltétel miatt állunk meg, az egybeesik a 2.3.8 állítás bizonyításának középső esetével. Ha pedig (B^*) miatt állunk le, akkor azt kell belátnunk, hogy az üres szint alatt minden pont fedett. Mivel üres szint úgy keletkezhet, hogy egy legalsó fedetlen elem szintjét megemeljük, így alatta már nem lehet másik fedetlen, hiszen ha vele egy szinten lett is volna másik, akkor nem keletkezett volna üres szint. A bizonyítás többi része analóg az előbb említett 2.3.8 állítás bizonyításának utolsó részével. □

Látjuk, hogy a második algoritmus is helyes. De milyen kapcsolatban áll az első algoritmussal?

2.5. A két algoritmus összehasonlítása

A 2.3 és a 2.4 részekben láttunk ugyanarra a problémára, az összegmatroid rangfüggvényének meghatározására kétféle algoritmust. Most azt szeretnénk megvizsgálni, hogy ez a két algoritmus milyen kapcsolatban áll egymással.

Vezessünk be egy új fogalmat:

2.5.1. Definíció (Duális matroid)**.** *Legyen $M = (\mathcal{S}, \mathcal{B})$ bázisaival adott matroid. Ekkor az M **duális matroidja** $M^* = (\mathcal{S}, \mathcal{B}^*)$, ahol $B^* = \{\mathcal{S} \setminus B : B \in \mathcal{B}\}$.*

Tehát a duális matroidban a bázisok pont az eredeti bázisok komplementerei lesznek. Belátható, hogy ez a struktúra szintén matroidot definiál [6]. Amikor azt mondjuk, hogy adott $M_1, ..., M_k$ matroid $B_1, ..., B_k$ bázisokkal, akkor ezen matroidok duálisához azt is értsük hozzá, hogy megkapjuk a $B_i^* = \mathcal{S} - B_i$ bázisokat is.

A 2.3 és a 2.4 algoritmusok futása közben egy tetszőleges pillanatban van k darab bázis az alaphalmazon és a csúcsok szintfüggvénye teljesíti a megfelelő invariáns tulajdonságokat (2.3.5, 2.4.5). Mit állíthatunk $k = 2$ esetén? Valójában az fog teljesülni, hogy a többletes csúcsokat emelő algoritmus futása megfeleltethető a fedetlen elemeket emelő algoritmus egy futásával. Azt is beláttuk az algoritmusok vizsgálata során, hogy teljesítik az optimalitási feltételeket (2.2.4), de mit jelent az optimális megoldás az eredeti és a duális bázisokban? Azt szeretnénk, hogy $|B_1 \cup B_2|$ a lehető legnagyobb legyen, de ez ugyanazt jelenti, mint hogy $|B_1^* \cup B_2^*|$ a lehető legnagyobb. Összefoglalva, ha meg szeretnénk kapni az optimális megoldást, akkor bedobhatom a kiindulási bázisokat az egyik algoritmusba, vagy a duálisait a másikba és mindkét módszerrel helyes eredményt fogok kapni.

2.5.2. Tétel. *A 2.3 többletes csúcsokat emelő algoritmus futásának lépései az $\mathcal{S}$ alaphalmaz, M_1, M_2 matroidok és B_1, B_2 kiindulási bázisok inputtal megfeleltethető a 2.4 fedetlen elemeket emelő algoritmus egy futásának lépéseivel az $\mathcal{S}$ alaphalmaz, M_1^*, M_2^* matroidok és B_1^*, B_2^* kiindulási bázisok inputtal.*

Bizonyítás. Először is nézzük meg hogyan tudjuk megfeleltetni egymásnak a 2.3.1 és 2.4.1 építőköveket:

2.5.3. Állítás. *Az egyik algoritmusban a fedetlen elemek a másik algoritmusban a többletes elemek.*

Bizonyítás. Ha egy s elem nincs benne sem B_1-ben, sem B_2-ben, akkor fedetlen lesz, viszont ekkor benne van B_1^*-ban és B_2^*-ban is, vagyis a duálisok által meghatározott bázisok szerint többletes lesz, mert egynél több bázisban szerepel. Ha pedig pontosan egyben szerepel, akkor a duálisok közül is pont egyben fog szerepelni. □

2.5.4. Állítás. *Ha (s, t) megengedett csere az egyik algoritmusban, akkor $s = t^*$ és $t = s^*$ jelölésekkel (s^*, t^*) megengedett csere a másik algoritmusban.*

Bizonyítás. Ha t-nek van (s, t) megengedett cseréje (2.3.2), akkor t-t kivesszük a B_i bázisból és helyette betesszük az eggyel lentebbi s elemet, ami eddig nem szerepelt ebben a bázisban. A másik algoritmusban ha $t = s^*$-ot kivesszük B_i-ből, az pont azt jelenti, hogy betesszük B_i^*-ba, $s = t^*$-ot pedig kivesszük a duális bázisból. A szintfüggvénnyel kapcsolatban az egyik esetben azt várjuk, hogy $h(t) = h(s) - 1$, a másik esetben pedig $h(s) = h(t) - 1$. Mivel a duálisban szerepet cserél a két elem, ez is rendben lesz. Fordítva is elmondható, hogy ha a fedetlent emelőben keresnénk egy megengedett cserét (2.4.1), ugyanaz a lépés jó lenne a többletest emelő algoritmusban is, vagyis egyszerre tudunk mindkét algoritmusban megengedett cserét végrehajtani ugyanazokkal az elemekkel. □

2.5.5. Állítás. *Az egyik algoritmusban az (A) megállási feltétel bekövetkeztekor a másik algoritmusban előáll a (C) eset, és ugyanez igaz a (B) és (B^*) feltételekre.*

Bizonyítás. Emlékeztető:

Megállási feltételek a többletes pontokat emelő algoritmusban (2.3.3):

(A) minden elem lefedett

(B) üres szint keletkezik (ekkor felette minden pont fedett)

(C) nincs több többletes elem, vagyis diszjunktak a bázisok

Megállási feltételek a fedetlen pontokat emelő algoritmusban (2.4.2) :

(A) minden elem lefedett, vagyis $S = \cup_i B_i$

(B*) üres szint keletkezik (ekkor alatta minden pont fedett)

(C) nincs több többletes elem, vagyis diszjunktak a bázisok

Ha B_1 és B_2 szerint minden elem le van fedve, akkor mindegyik szerepel legalább az egyikben. Ekkor viszont B_1^* és B_2^* közül legfeljebb csak az egyikben szerepelhetnek, tehát a duális bázisokban nem lehet többletes elem. Ugyanez igaz fordítva is: ha B_1 és B_2 szerint nincs több többletes elem, akkor B_1^* és B_2^* szerint minden elem le van fedve. Így párba tudtuk állítani az (A) és (C) feltételeket.

Mit mondhatunk (B) és (B^*) feltételek kapcsolatáról? Ha üres szint keletkezik a többletes csúcsokat emelő algoritmusban, akkor a legalsó t többletes elemnek nem volt megengedett cseréje, ezért megemeltük. A 2.5.4 tétel miatt ekkor a fedetlent emelő $t = s^*$ legalsó fedetlen elemnek sem lehet megengedett cseréje, ezért őt is meg kellett emelnünk, vagyis üres szint is egyszerre keletkezik. A többletest emelőben minden többletes pont az üres szint felett lesz, hiszen a legalsó megemelésével keletkezett az üres szint, illetve az összes fedetlen az 0. szinten van a (β) invariáns tulajdonság szerint (2.3.5), így az üres szint felett minden fedett. De ekkor a duálisban az üres szint felett az összes fedetlen lesz, a 0. szinten pedig a többletesek, így azt tudjuk mondani, hogy az üres szint alatt fedett minden pont és ezt is vártuk. Vagyis, ha a többletest emelő algoritmus futása során bekövetkezik az egyik megállási feltétel, akkor a fedetlent emelőben az ezekhez a lépésekhez megfeleltető párhuzamos lépések során ugyanakkor fog bekövetkezni valamelyik ottani megállási feltétel. □

Ezekből már következik, hogy ha futtatjuk az egyik algoritmust, akkor azok a lépések egy helyes futtatását jelentik a másik algoritmusnak, ha ott a kiindulási bázisok duálisát adtuk meg.

Az elején minden elem a 0. szinten van. A többletest emelő szerint veszünk egy többleteset és arra szeretnénk vagy megengedett cserét találni, vagy megemelni a szintjét. Mivel minden megengedett cserének eggyel lentebbi szinten kell lennie, ezért csak megemelni tudjuk. Ez a pont a duálisok szerint egy fedetlen elem volt, aminek a megengedett cseréi szintén eggyel lejjebb vannak, szóval őt is csak megemelni tudtuk volna. Most tegyük fel, hogy egy ideig megfeleltethető a két futtatás és nézzük meg mi történik egy általános helyzetben.

A 2.3 algoritmusban veszek egy t legalsó többletes pontot. Eddig meg tudtuk feleltetni ezt a 2.4 algortimus egy futásával, így ott ezt az elemet jelöljük $t = s^*$-gal, ami ott az egyik legalsó fedetlen a duális bázisok szerint, hiszen ha lenne lentebbi, akkor az többletes lenne az eredeti B_1 és B_2 bázisok szerint. Így ez a s^* egy jó legalsó fedetlen elem a fedetlent emelő algoritmusban. A 2.5.4 tétel miatt tudjuk, hogy egyszerre létezik megenngedett cseréjük, ha van, végezzük el. Ha pedig nincs, akkor mindkettőben meg kell emelnünk eggyel a kiválasztott $t = s^*$ elem szintjét. □

3. fejezet

Egy approximációs algoritmus az MDMST problémára

Az előző fejezetben egy matroidokon definiált problémára láttunk optimális megoldást, most pedig egy approximációs algoritmust fogunk adni arra a feladatra, hogy hogyan lehet egy gráfban olyan minimális költségű feszítőfát találni, aminek a legkisebb a legnagyobb fokszáma. Ennek a problémának bizonyos matroidokra vett általánosítására kitekintünk a 3.3 pontban. A fejezet során Kamalika Chaudhuri, Satish Raob, Samantha Riesenfeld és Kunal Talwar A push-relabel approximation algorithm for approximating the minimum-degree MST problem and its generalization to matroids című cikkét fogjuk alapul venni [1].

3.1. Motiváció

Adott egy $G = (V, E)$ gráf az éleivel és csúcsaival, valamint egy $c : E \to \mathbb{R}$ költségfüggvény az éleken. Ekkor a következő problémákról lesz szó:

- **MST** (minimum spanning tree): minimális költségű feszítőfa
- **MDMST** (minimum-degree minimum spanning tree): olyan MST, melyben a legnagyobb fokszám a lehető legkisebb

Egy MST-t tudunk találni mohó Kruskal-algoritmussal[2], ahogy azt említettük már a 2.1 fejezetben is. Azonban ha azt is meg szeretnénk adni feltételnek, hogy a feszítőfa csúcsainak a legnagyobb fokszáma a lehető legkisebb legyen, akkor már egy NP-nehéz feladatot kapunk (költségfüggvény nélkül is), mert a Hamilton-út probléma visszavezethető rá: ha 2 a legnagyobb fokszám, akkor pont egy Hamilton-utat kapunk, ha pedig legalább 3 lenne, akkor nincs Hamilton-út. Ezért erre a két feltételes feladatra egy polinomiális futásidejű approximációs algoritmust keresünk. Itt a kimeneti MST maximális fokszáma legfeljebb $2\Delta_{OPT}(G) + O(\sqrt{\Delta_{OPT}(G)})$ lesz, ahol $\Delta_{OPT}(G)$ egy olyan G-beli MST-nek lesz a maximális fokszáma, ahol ez a legkisebb. Ha csak a fokszámot szeretnénk minimalizálni, akkor arra létezik lényegesen erősebb, $\Delta_{OPT}(G)+1$ kimenetű algoritmus, de abban a megoldásban a költség nem feltétlenül minimális [7]. Ebben lesz segítségünkre a szintező algoritmusokra jellemző szemlélet. Ezt a problémát a 3.3 részben kiterjesztjük bizonyos matroidokra is.

3.2. Az algoritmus

Tehát a feladatunk a következő: adott egy $G = (V, E)$ gráf az éleken értelmezett c költségfüggvénnyel. A minimális költségű feszítőfák között keressünk olyat, amelyiknek a maximális fokszáma a lehető legkisebb (MDMST). Kezdjük megint az építőkövek megértésével. A 2.3.1 és 2.4.1 fejezetekben látott fogalmakat szeretnénk alkalmazni a jelenlegi problémára, illetve új fogalmakat is bevezetünk. A könnyebb olvasás kedvéért használjuk a következő halmazműveleti jelöléseket: egy elemből álló $\{x\}$ halmaz és az x elem is legyen egyszerűen x, illetve egy másik halmazzal vett unióját és különbségét jelöljük $+$ és $-$ szimbólumokkal. Szükségünk lesz továbbá a következő, gráfelméletben gyakran használt jelölésekre is. Legyen $F \subseteq E$ az élek egy részhalmaza, $U \subseteq V$ a csúcsoké, ekkor legyen

az U által feszített F-beli élek halmaza $\mathbf{i_F}(\mathbf{U}) := \{uv|uv \in F \text{ és } u, v \in U\}$,
az U által érintett F-beli élek halmaza $\mathbf{e_F}(\mathbf{U}) := \{uv|uv \in F \text{ és } u \text{ vagy } v \in U\}$,
egy u csúcsra illeszkedő F-beli élek halmaza $\delta_\mathbf{F}(\mathbf{u}) := \{uv|uv \in F\}$.

3.2.1. Építőkövek

Legyen $G = (V, E)$ gráf, $\mathbf{c}$ költségfüggvény az éleken és T ebben egy MST, melynek maximális fokszáma legyen $\mathbf{\Delta}(\mathbf{T})$.

3.2.1. Definíció. *Egy v csúcsot **többletesnek** nevezünk, ha fokszáma $\Delta(T)$.*

3.2.2. Definíció. *Legyen $h : V \to \{0, 1, ..., n\}$ **szintfüggvény**, ahol $n = |\mathcal{S}|$.*
*A p. szinten lévő csúcsok halmaza legyen: **p. szint** $:= \{u \in V|\ h(u) = p\}$.*
A p. szinten és fölötte lévő csúcsok halmaza legyen: $\mathbf{W_p} := \{u \in V|\ h(u) \le p\}$.
Egy e él szintje legyen: $\mathbf{h}(\mathbf{e}) := max\{h(u), h(v)\}$, ahol u és v az él végpontjai.

3.2.3. Definíció. *Legyen T egy adott MST, ekkor **T-beli csere** egy olyan (e, e') élpár lehet, amire igaz, hogy*

- $e' \notin T, e \in C(T, e')$
- $c(e) = c(e')$.

3.2.4. Állítás. *Ha (e, e') csere volt T-ben, akkor $T - e + e'$ szintén MST lesz G-n.* □

3.2.5. Definíció. *Az u-ra hasznos T-beli **megengedett cserék** halmaza legyen $\mathbf{S^T_u} := \{(e, e')\ T\text{-beli csere}\}$, ha (e, e') teljesíti a következőket:*

- $e \in \delta_T(u), e' \notin \delta_T(u)$
- $h(u) \geq h(e') + 1$

3.2.6. Megjegyzés. *Valójában $h(u) = h(e') + 1$ is teljesül a 3.2.10 (α) tulajdonság miatt.*

3.2.7. Definíció. *Legyen $\mu \geq 1 \in \mathbb{R}$, ekkor a p. szint **$\mu$-ritka**, ha: $|W_p| \leq \mu \cdot |W_{p+1}|$.*

Vegyük észre, hogy ez $\mu = 1$ esetén pont az üres szintet jelenti, amivel a 2.3.2 és 2.4.2 algoritmusok dolgoztak. Ez a fogalom fogja segíteni az algoritmus approximációs jellegét.

3.2.2. Az algoritmus lépései

Először is keressünk egy tetszőleges T minimális költségű feszítőfát a gráfon, ezt megtehetjük például a mohó Kruskal-algoritmussal[2], ahogy azt említettük már a 2.1, 3.1 részekben is.

Ezután fázisokra osztjuk az algoritmust: minden fázis végére vagy eggyel csökken T maximális fokszáma, vagy kapunk egy tanút, ami bizonyítja, hogy közel vagyunk az optimális megoldáshoz. Ha a fázis elején a maximális fokszám Δ volt, akkor ezt a fázist Δ-fázisnak hívjuk.

Egy általános fázis elején határozzuk meg T max fokszámát, ez legyen $\Delta(T)$. A h szintfüggvény értéke legyen 0 minden csúcsra és vezessünk be egy t többletes függvényt is a csúcsokra: ha fokszáma $\Delta(T)$, vagyis többletes a csúcs (3.2.1), akkor 1, különben 0.

Amíg van többletes csúcs és nincs μ-ritka szint (3.2.7), addig egy általános lépés a fázison belül: legyen p a legalacsonyabb szint, ahol van többletes csúcs. Legyen U_p a p. szinten lévő többletes csúcsok halmaza. Ha van olyan $u \in U_p$, aminek van megengedett (e, e') cseréje $\in S^T_U$, akkor T-ből töröljük e-t és vegyük be e'-t. Ezután az e él végpontjainak többletességét állítsuk 0-ra, és ha e' valamelyik végpontja többletessé vált, akkor ott t legyen 1. Ha egyik p. szinten lévő többletes csúcsnak sincs megengedett cseréje, akkor emeljük fel az egész U_p halmazt a $p+1$. szintre.

Álljon le az algoritmus, ha bármikor μ-ritka szint keletkezik. Ennek a szintnek az indexe legyen p^*.

Az algoritmus kimenete egy T fa és $\mathcal{W}^{p^*} = (F, W_{p^*})$ páros, ahol F a p^*. szint alatti T-beli élek halmaza, $\mathcal{W}^{p^*}$ pedig a p^*. szint feletti pontok halmaza (3.2.2).

Algorithm 3 Approximációs szintező

Input: (G, μ)
Output: T, $\mathcal{W}^{p^*}$

T := tetszőleges MST-je G-nek
while nincs p^* μ-ritka szint **do**
  $\Delta := T$ max fokszáma
  $\forall v \in V:\ h(v) := 0$ és $t(v) := 1$, ha v fokszáma Δ, különben $t(v) := 0$
  while van többletes csúcs és nincs μ-ritka szint **do**
    p := legalacsonyabb többletes csúcsot tartalmazó szint indexe
    $U_p := \{$többletes csúcs $|$ szintje $p\}$
    if van olyan csúcs U_p-ben, aminek van megengedett cseréje (e, e') **then**
      T := $T - e + e'$
      for v in {e végpontjai} **do**:
        t(v) := 0
      for v in {e' végpontjai} **do**:
        if v fokszáma most Δ **then**
          t(v) := 1
    else
      for v in U_p: h(v) := p+1
F := {T-beli élek | nem W_{p^*+1}-beli}
return $(T; \mathcal{W}^{p^*} := (F, W_{p^*}))$

3.2.3. Az algoritmus helyessége

A bizonyítás során ellenőrizni szeretnénk, hogy az algoritmusunk valóban polinomiális idejű, valamint megnézzük, hogy most melyek lesznek az invaráns tulajdonságok és belátjuk, hogy azok teljesülnek is. Az utóbbi bizonyítása nagyon hasonló lesz a

2.3.3 és 2.4.3 fejezetekben látott tételekhez. Most azt is meg kell néznünk, hogyan válasszuk jól μ-t, hogy a kívánt approximációt kapjuk.

3.2.8. Állítás. *[5] Az algoritmus polinomiális futásidejű.*

Bizonyítás. Ha valamelyik csúcs felér az n. szintre, akkor biztosan van egy üres, vagyis 1-ritka szintünk (skatulya-elv). Tehát minden fázisban minden csúcs legfeljebb n alkalommal lesz újraszintezve bármilyen $\mu \geq 1$-re. Egy fázison belül így legfeljebb n^2 lépésre kerülhet sor. Mivel az utolsón kívül minden fázis eggyel csökkenti T fokszámát, így legfeljebb n fázis van. □

3.2.9. Megjegyzés. *[5] Ha $\mu > 2$, akkor egy csúcsot legfeljebb $\log_2 n$-szer nevezünk át, így egy gyorsabb algoritmust kapunk.*

3.2.10. Definíció (Invariáns tulajdonságok)**.** [1]

(α) $\forall u \in V : \forall (e, e')$ *csere* $\in S_u^T$*-re:* $h(e) \leq h(e') + 1$

(β) egy Δ fázisban minden csúcs fokszáma legfeljebb Δ

3.2.11. Állítás. *[5] Az algoritmus futása során végig teljesülnek a 3.2.10 invariáns tulajdonságok.*

Bizonyítás. Az (α) tulajdonság teljesülésének belátása teljesen analóg a 2.3.6 bizonyítással, hiszen egy gráfban a feszítőfák vagy minimális költségű feszítőfák azok a grafikus matroidban a bázisok, minimális költségű bázisok (2.1.2, 2.1.4), így használható a 2.3.7 lemma.

Most vizsgáljuk meg a (β) tulajdonságot. Egy Δ-fázis elején a legnagyobb fokszám Δ. Báziscserénél egy U_p-beli elemnek csökkentjük eggyel a fokszámát és egy alatta lévő elem fokszámához ad hozzá egyet. Mivel alatta legfeljebb $\Delta - 1$ lehetett a fokszám és egy megengedett csere végrehajtásakor 1-gyel változnak a fokszámok, így valóban nem lesz Δ-nál nagyobb fokszámunk.

□

3.2.12. Tétel. *Az algoritmus leállásakor $2\Delta_{OPT}(G)+O(\sqrt{\Delta_{OPT}(G)})$ optimális megoldást kapunk.*

[1] A cikk ezeket a tulajdonságokat nem hívja invariáns tulajdonságnak, a dolgozat egységesítése miatt írtam így. Az alfa tulajdonságot lehetséges (feasible) szintezésnek nevezi. Külön-külön mindkét állítást kimondja, csak nem így hivatkozik rájuk.

Bizonyítás. Ha azért áll le az algoritmus belső ciklusa, mert nincs több többletes csúcs, akkor a fának a maximum fokszáma legfeljebb $\Delta - 1$. Ha azért áll le, mert μ-ritka szint keletkezett, akkor megmutatjuk, hogy Δ közel van az optimális megoldáshoz.

Először is szükségünk lesz a tanú fogalmára:

3.2.13. Definíció (Tanú)**.** *Legyen $F \subseteq E$ erdő, amit tartalmaz G egy MST-je és $X \subset V$. Ekkor (F, X)* ***tanú****, ha teljesíti a következő tulajdonságot:*

minden F-et tartalmazó T MST-re : minden $e \in T \setminus F : e \in e(X)$.

3.2.14. Állítás. *[5] Egy $\mathcal{W} = (F, X)$ tanú a $G = (V, E)$ gráfon. Ekkor minden G-beli MST maximum fokszáma legalább $\frac{|V|-|F|-1}{|X|}$.*

Bizonyítás. Legyen T és T' a G gráf egy-egy MST-je úgy, hogy T' tartalmazza F-et és maximális a metszetük. A báziscserélési axióma miatt (2.1.6) T tartalmazza $T' \setminus F$-et. A tanú definíciójából következik, hogy minden $T' \setminus F$-beli él érinti X-et. Mivel $|V| - |F| - 1$ db él van $T' \setminus F$-ben, ezért X-ben az átlagos fokszám $T' \setminus F$-en és ebből következően T-n is legalább $\frac{|V|-|F|-1}{|X|}$. □

Az algoritmus leállásakor kapunk egy tanút:

3.2.15. Állítás. *[5] Amikor az algoritmus leáll, legyen T az MST és h az aktuális szintfüggvény. Bármilyen p egészre $F := e(W_{p+1}) \setminus T$ és $X := W_p$. Ekkor $\mathcal{W}^p = (F, X)$ tanú.*

Bizonyítás. Indirekt tegyük fel, hogy T' egy olyan MST-je a G gráfnak, ami tartalmazza F-et és tartalmaz egy $e' \notin F$ élet, ami nem érinti W_p-t. A báziscserélési axióma (**??**) miatt van egy olyan $e \in T \setminus F$ él, hogy (e, e') csere T-ben. Mivel $e \in T$, ezért érinti W_{p+1}-et, ezért $h(e) \geq p + 1$. Másrészről e nem érinti W_p-t, ezért $h(e') \leq p - 1$. Ez ellentmond a 3.2.10 (α) tulajdonságnak. □

Legyen p^* a μ-ritka szint indexe. A 3.2.14 és 3.2.15 állításokból következik, hogy bármelyik MST fokszáma legalább $\frac{|V|-|F|-1}{|W_{p^*}|}$. Vegyük észre, hogy a számláló pont $e_T(W_{p^*+1})$ méretével egyezik meg. Ekkor

$$\frac{|V| - |F| - 1}{|W_{p^*}|} = \frac{|V| - |F| - 1}{|W_{p^*+1}|} \cdot \frac{|W_{p^*+1}|}{|W_{p^*}|} \geq \frac{1}{\mu} \cdot \frac{|e_T(W_{p^*+1})|}{|W_{p^*+1}|}.$$

Ebből következik a következő lemma:

3.2.16. Lemma. *[5] Amikor az algoritmus leáll, legyen T az MST, h az aktuális szintfüggvény és p^* a μ-ritka szint indexe. A T-beli, W_{p^*+1}-t érintő csúcsok halmaza $e_T(W_{p^*+1})$. Ekkor minden G-beli MST fokszáma legalább $\frac{1}{\mu} \cdot \frac{|e_T(W_{p^*+1})|}{|W_{p^*+1}|}$.*

Tehát $\Delta_{OPT}(G)$ alsó korlátjához $e_T(W_{p^*+1})$ alsó korlátját kell csökkentenünk.

3.2.17. Definíció. *__Önkéntesnek__ hívunk egy fokszámcsökkentést, ha a csúcs többletes volt, és __kényszerűnek__, ha nem.*

Szeretnénk meghatározni, hogy legfeljebb hány kényszerű fokszámcsökkenés történik az algoritmus futása során, ehhez szükségünk lesz még a következő fogalmakra.

3.2.18. Definíció. *Jelöljük __zászlóval__ azokat a csúcsokat, amelyeket eggyel fentebbi szintre emeltünk, de nem csökkentettük a lefedettségét. Továbbá minden csúcshoz rendeljünk egy __többletes csere__ címkét, amelyben azt a cserét tároljuk, amely miatt többletessé vált a csúcs.*

Kezdetben nincsenek zászlók és üres az összes címke. A Δ-fázis belső ciklusában mindig keresünk egy nemüres U_p-t, vagyis lesz egy p szintű többletes elem. Ha létezik rá (e, e') megengedett csere, akkor végrehajtjuk, e végpontjain töröljük a zászlókat és e' végpontjainak a címkéjébe beírjuk (e, e') cserét, ha most valamelyik többletessé vált. Ha nincs megengedett csere, akkor U_p minden elemét megemeljük eggyel, beállítjuk a zászlókat és többletes csere címkéket kiürítjük. Tehát egy zászlós pont címkéje mindig üres. Mivel a fázis során egyik csúcsnak sem lehet Δ-nál több a fokszáma (3.2.10), ezért minden csúcsnak legfeljebb 1 többlete van, vagyis egyik címkét sem írjuk át mielőtt kiürítenénk.

3.2.19. Definíció. *Egy (e, e') __csere szintje__ legyen e szintje, vagyis $h(e, e') := h(e)$. Egy cserét __gyökér cserének__ hívunk, ha hasznos egy zászlós csúcsra.*

3.2.20. Definíció. *Az (e, e') cserét azért hajtottuk végre, hogy e-nek egy u végpontjának többletét csökkentsük. Ekkor legyen (f, f') az u címkéjében szereplő többletes csere, amikor végrehajtottuk az (e, e') cserét. Így (f, f') volt az utolsó csere, ami növelte u fokszámát, ezért nevezzük az (f, f') cserét az (e, e') __szülő cseréjének__.*

A zászlózási technika biztosítja, hogy minden nem-gyökér cserének legyen szülő cseréje.

Egy p szintű csere definíció szerint csökkenti egy p szintű pont fokszámát. Mivel a cserék során a többlet mindig egy magasabb szintű csúcsból terhelődik át egy alacsonyabban lévő csúcsba, ezért egy nem-gyökér csere szintje mindig kisebb, mint a szülő cseréjének a szintje. A szülő kapcsolat definiál egy D irányított gráfot a cserék halmazán.

3.2.21. Definíció. *A D irányított gráf egy komponensét nevezzük __szakasznak__.*

Vagyis egy szakasz azokból a cserékből áll, amelyek ugyanabból a gyökér cseréből következnek. Egy szakaszban lévő cserék nem feltétlenül egymás után következnek az algoritmus futása során.

3.2.22. Definíció. *Egy **szakasz szintje** legyen a gyökér csere szintje.*

Mivel minden csere legfeljebb 2, nála alacsonyabb szinten lévő csere szülője, ezért teljesül a következő állítás.

3.2.23. Állítás. *[5] Egy p szintű szakaszban legfeljebb 2^{p-q} db q szintű csere van.* □

Azt mondjuk, hogy egy kényszerű fokszámcsökkentést tartalmaz egy szakasz, ha valamelyik szakaszbeli csere okozta. Mivel minden cseréből legfeljebb egy kényszerű fokszámcsökkenés következik:

3.2.24. Következmény. *[5] Egy p szintű szakaszban legfeljebb $2^{p-q+1}-1$ kényszerű fokszámcsökkenés van.*

Bizonyítás. Egy r szintű csúcs kényszerű fokszámcsökkenése egy r szintű csere miatt következhet be, így a p szintű szakaszban lévő cserék a p. és q. szint között lehetnek.

$$\sum_{r=q}^{p} 2^{r-q} = 2^{p-q+1} - 1.$$

□

3.2.25. Állítás. *[5] Legyen $p > p^*$. Ekkor $|W_p| > \mu|W_{p+1}|$.*

Bizonyítás. Az algoritmus futása során mindig legfeljebb egy p-re csökken a p. szint mérete és ekkor ugyanennyivel nő a $p+1$. szint mérete. Vagyis az egyetlen szint, ami nem μ-ritkából μ-ritkábá tud változni a p. szint. Mivel az algoritmus leáll, ha μ-ritka szint keletkezett, ezért pontosan egy μ-ritka szint lesz. □

3.2.26. Állítás. *[5] A kényszerű fokszámcsökkenések száma W_{p^*+1}-ben legfeljebb $2|W_{p^*+1}|(\frac{\mu}{\mu-2})$.*

Bizonyítás. A 3.2.24 és 3.2.25 állításokból következik. □

Ebből már levezethető az approximációs hányados egy adott μ mellett:

3.2.27. Lemma. *[5] Adott G gráf, $\mu > 2$, a 3.2.2 algoritmus polinomiális idejű és ad egy Δ fokszámú MST-t, ahol $\Delta \leq \mu\Delta_{OPT}(G) + 2 + \frac{2\mu}{\mu-2}$.* □

Ha ismerjük az optimális Δ értéket és $\mu = 2\Delta_{OPT}(G) + 4\sqrt{\Delta_{OPT}(G)} + 4$-t helyettesítünk be a lemmába, akkor azt kapjuk, hogy $\Delta \leq 2\Delta_{OPT}(G) + O(\sqrt{\Delta_{OPT}(G)})$, vagyis bebizonyítottuk a tételt.

□

3.3. Általánosítás matroidokra

A 2. fejezetben a szintező algortimust matroidokra alkalmaztuk, próbáljuk meg ezt a feladatot is általánosítani. Vegyünk egy $H = (V, E)$ hipergráfot, ahol egy él mérete legfeljebb k. Ehhez a hipergráfhoz definiáljunk egy matroidot úgy, hogy az alaphalmaz az élek halmaza: $M = (E, \mathcal{F})$. A matroidhoz adjunk meg egy c költségfüggvényt is. Ekkor M matroid minimális költségű bázisai (MCB) között keresünk olyat, amely H-ban minimális fokszámú (MDMCB). Jelöljük $\Delta(H, M)$-mel ennek a bázisnak a fokszámát.

Az algoritmus általánosításához még néhány fogalomról nézzük meg, hogyan változik a 3.2.1 fejezethez képest. Adott T MCB, ekkor (e, e') csere, ha $e \in T, e' \notin T$ és $T - e + e'$ szintén MCB. A szintfüggvény és többletesség pontosan ugyanaz maradhat, és amikor élekre szeretnénk kiterjeszteni, akkor $h(e) := \max_{e \in \delta_u} h(u)$. A tanút is ki tudjuk terjeszteni: MST helyett MCB-ben keressük. Ezek után a 3.2.2 algoritmus alapján tudjuk definiálni az általános algoritmust annyi különbséggel, hogy az inputba H és M tartozik. Az algoritmus polinomiális idejű és teljesülnek rá az invariáns tulajdonságok [5]. Ebben az esetben egy gyengébb approximációt tudunk bizonyítani, mert az élszámra nem feltétlenül teljesül egy feltétel amit felhasználtunk a 3.2.12 tétel bizonyításában. Így az algoritmus leállásakor kapott MCB H-beli fokszámáról a következőt tudjuk mondani [5]:

$$\Delta \leq k^2 \Delta_{OPT}(H, M) + 2k^{\frac{3}{2}} \sqrt{\Delta_{OPT}(H, M)} + k + 1.$$

Irodalomjegyzék

[1] A push–relabel approximation algorithm for approximating the minimum-degree mst problem and its generalization to matroids. *Theoretical Computer Science*, 410(44):4489–4503, 2009.

[2] Ronald L Rivest Clifford Stein Cormen, Thomas; Charles E Leiserson. Introduction to algorithms, third edition. pages 631–633, 735–736, 2009.

[3] J. Edmonds and D.R. Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards*, pages 147–153, 1965.

[4] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming 1*, pages 127–136, 1971.

[5] András Frank and Zoltán Miklós. Simple push-relabel algorithms for matroids and submodular flows. *Japan journal of industrial and applied mathematics*, 29:419–439, 2012.

[6] András Frank. Matroidelméleti alapok. *ELTE TTK, Operációkutatási Tanszék*, 2007.

[7] Raghavachari B. Furer M. Approximating the minimum-degree steiner tree to within one of optimal. *Journal of Algorithms*, 17:409–423, 1994.

[8] Robert E. Goldberg, Andrew V.; Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM.*, 35(4):921–940, 1988.

[9] Herbert S. Wilf. Algorithms and complexity. *University of Pennsylvania*, page 65, 1994.