Paraméteres problémák a kombinatorikus optimalizálásban

Szakdolgozat

Írta: Jüttner Alpár

Témavezető: Frank András
egyetemi tanár

ELTE Természettudományi Kar
Operációkutatási Tanszék

1998.

Tartalom

1. Bevezetés

Ha adott egy $\mathcal{G} = (V, E)$ gráf és egy $l : E \longrightarrow \mathbb{R}_+$ hosszfüggvény, akkor az $s - t$ távolságot — az l szerinti legrövidebb $s \rightsquigarrow t$ $(s, t \in V)$ út hosszát — hatékonyan meg tudjuk határozni például a Dijkstra algoritmussal. Természetesen adódik a következő feladat: adott még egy $c : E \longrightarrow \mathbb{R}_+$ költségfüggvény, ami az élek hosszának egy egységgel való csökkentésének költségét adja meg és szeretnénk az $s - t$ távolságot minél olcsóbban egy L érték alá csökkenteni úgy, hogy a gráfban az egyes élek súlyát csökkenthetjük. Vagyis keressük $\min\{cz : 0 \leq z \leq l, f(z) \leq L\}$-t, ahol $f(z) := = \min\{(l - z)(p) : p \text{ egy } s \rightsquigarrow t \text{ út}\}$.

Tehát van egy optimalizálási feladatunk, ahol maga a feladat egy z paramétertől függ, és mi egy olyan paramétert keresünk, melynél az optimum bizonyos tulajdonságoknak eleget tesz. Az ilyen jellegű feladatot nevezzük *paraméteres problémá*nak. Ezek lehetnek önmagukban érdekes feladatok, de többször előfordul, hogy egy (akár diszkrét) optimalizálási feladat egy paraméteres feladat megoldására vezethető vissza.

Dolgozatunkkal célunk bepillantást adni a paraméteres problémák meglehetősen szerteágazó területébe és áttekinteni a megoldásukra szolgáló főbb eszközöket.

A paraméteres problémák két csoportját vizsgáljuk aszerint, hogy az $f(z)$-t optimalizáljuk (például maximalizáljuk) vagy z egy — többnyire lineáris — célfüggvényét optimalizáljuk úgy, hogy $f(z)$-re teszünk megkötéseket. Az előző típusúak például egyes problémák *Lagrange-relaxáció*jának megoldásakor jelennek meg, míg az utóbbiak főleg az úgynevezett *inverz problémák* területén találnak alkalmazásokra.

Sok paraméteres probléma könnyen felírható lineáris programozási feladatként és található valamilyen analitikus módszer a megoldására. Ennél sokkal izgalmasabb feladat e problémákra erősen polinomiális futási idejű kombinatorikus algoritmust találni, a mi vizsgálódásunknak is ez áll a középpontjában. E kérdések megválaszolásához sokszor mélyebb apparátust kell felhasználnunk.

A harmadik fejezetben számos paraméteres problémát megvizsgálunk. Ezek közül néhány közvetlenül visszavezethető más ismert optimalizálási feladatra, máskor egy kicsit többet kell dolgoznunk. Megismerkedünk az úgynevezett *Newton-módszer*rel, ami több esetben is hasznosnak bizonyul paraméteres problémák megoldásakor.

A Newton-módszer egy kifinomult alkalmazása Frederickson és Solis-Oba eredménye, ami egy matroidokon levő inverz feladatot old meg. Ezt ismertetjük a negyedik fejezetben.

Nagyon sok probléma megoldható C. H. Norton, S. A. Plotkin és Tardos Éva szeparációs eljáráson alapuló algoritmusával. Maga az algoritmus igen általános környezetben működik, ennek ára viszont, hogy sok technikai részlettel kell megküzdeni. Ezért csak kimondjuk a megfelelő tételt, be nem bizonyítjuk. Ehelyett nehány speciális esetben bemutatjuk, hogyan működik az algoritmus.

A második fejezetben definiáljuk egy feladat Lagrange-relaxációját és mutatunk két alkalmazást. Az első egy példa arra, amikor egy $\mathcal{NP}$-teljes probléma Lagrange-relaxációja már erősen polinomiális időben megoldható. A második alkalmazásban a Lagrange-relaxáció segítségével adunk hatékony algoritmust egy optimalizálási feladatra.

A negyedik fejezetben néhány megoldatlan problémát vázolunk, a függelékben pedig a felhasznált jelölésekről és tételekről adunk áttekintést.

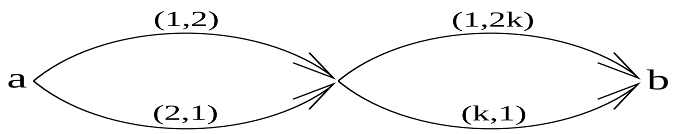
Végezetül a szerző itt szeretne köszönetet mondani témavezetőjének, *Frank András*nak e dolgozat elkészülte során nyújtott rengeteg segítségéért, hasznos tanácsaiért és végtelen türelméért.

2. Lagrange relaxáció

Nézzünk először egy példát:

Legyen adott egy $\mathcal{G} = (V, E)$ irányított gráf (egy úthálózat), c és $t : E \longrightarrow \mathbb{R}_+$ súlyfüggvények (egy költség- és egy időfüggvény) és $a, b \in V$. Az a pontból szeretnénk eljutni b-be a lehető legolcsóbban, de az utazásra legfeljebb T idő áll rendelkezésre. Azaz keressük $\min\{c(p) : p \in \mathcal{P}_{a,b}$ és $t(p) \leq T\}$-t, Ahol $\mathcal{P}_{a,b}$ az $a \rightsquigarrow b$ utak halmaza. Sajnos ez egy $\mathcal{NP}$-teljes probléma, így nem várható bizonyíthatóan gyors algoritmus a megoldására. Természetesen adódik a következő heurisztikus elgondolás: Tekintsük a $c_\lambda := c + \lambda t$ *módosított költségfüggvény*t. Rögzített λ-ra a c_λ minimális utat persze ki tudjuk számolni, jelöljük ezt p_λ-val. Legyen először $\lambda = 0$. Ha erre $t(p_\lambda) \leq T$, akkor ez az eredeti feladatnak is optimális megoldása. Ha nagyobb, akkor növeljük a λ-t. A módosított költségfüggvényben ekkor erőteljesebben érvényesül a t hatása, így a c_λ szerinti optimális megoldás t szerinti súlya csökken. Tehát addig növeljük λ-t, míg a c_λ-optimális megoldás t szerinti súlya T-nél kisebb nem lesz.

Semmi sem biztosítja számunkra, hogy ez az eljárás az optimális megoldást vagy akár csak bizonyíthatóan legfeljebb az optimum valamilyen konstansszorosát adná. Ez nem is igaz, erre mutat példát az 1. ábra. Könnyen ellenőrizhető, hogy itt a fenti módszer a $k + 1$ költségű utat találja meg, az optimális megoldás költsége viszont csak 3. ($k > 2$.)



1. Ábra: $T = 2k + 1$ (Az éleken a c és a t értékei vannak)

Tehető azonban egy fontos észrevétel. Ha $\lambda \geq 0$, és $p \in \mathcal{P}_{a,b}$ olyan, hogy $t(p) \leq T$, akkor

$$f(\lambda) := min\{c_\lambda(p) : p \in \mathcal{P}_{a,b}\} - \lambda T \leq c_\lambda(p) - \lambda T = c(p) + \lambda(t(p) - T) \leq c(p),$$

azaz minden $\lambda \geq 0$-ra $f(\lambda)$ *alsó becslését* adja az eredeti feladat optimális megoldásának, ha az létezik; és nem korlátos, ha nem. Továbbá, ha létezik az eredeti feladatnak megoldása, akkor $c_\lambda(p_\lambda) \leq c_\lambda(p)$ alapján

$$c(p_\lambda) - c(p) \leq \lambda(t(p) - t(p_\lambda)) \leq \lambda(T - t(p_\lambda)),$$

azaz ha valamely λ^*-ra $T - t(p_{\lambda^*}) = 0$, akkor p_{λ^*} az eredeti feladat optimális megoldása és λ^* maximalizálja f-et.
Könnyen látható továbbá, hogy amennyiben $\tilde{\lambda}$ maximalizálja $f(\lambda)$-t, akkor $p_{\tilde{\lambda}}$ választható olyannak, hogy $t(p_{\tilde{\lambda}}) \leq T$ legyen, és ez a $p_{\tilde{\lambda}}$ a fenti heurisztikával kapható legjobb megoldása a feladatnak.

Érdekes tanulság, hogy egy, az optimális megoldást közelítő heurisztikából alsó becslést kapunk a feladatra. Összefoglalva: Az eredeti problémából bizonyos peremfeltételeket elhagytunk (ez a *relaxáció*), és beépítettük a célfüggvénybe. Az eredeti feladat feltételeit kielégítő megoldások persze kielégítik a relaxált feltételeket is, így az értelmesen megválasztott célfüggvény és a kompenzáció miatt egy alsó becslést kaptunk az eredeti feladatra. Ha a peremfeltétel nem teljesül, akkor erősítjük a hatását a célfüggvényben és ezzel kikényszerítjük, hogy a megoldás közeledjen az optimális megoldáshoz, és ezzel együtt az kapott alsó becslés és az eredeti feladat optimuma közti különbség is csökkenjen. Ez a *Lagrange-relaxáció* alapgondolata.

2.1. Egy kicsit pontosabban

Tekintsük a következő feladatot:

Keressük a

$$z^* = \min cx$$

értéket az

$$Ax = b \tag{P}$$

$$x \in X$$

feltételek mellett. Itt cx egy lineáris célfüggvény és $Ax = b$ explicit megadott lineáris feltételek. X általában valamilyen jól számolható poliéder vagy diszkrét halmaz (például megengedett áramok vagy utak, fák, fenyők halmaza).

$$L(\lambda) := \min\{cx + \lambda(Ax - b) : x \in X\}$$

a feladat ***Lagrange-függvénye***, λ elemeit pedig ***Lagrange multiplikátorok***nak nevezzük (λ egy annyi dimenziós vektor, ahány sora van A-nak).
A fejezet elején leírt meggondolás a következő általános eredményt adja:

2.1. Lemma. (Lagrange korlát) *Minden λ vektorra $L(\lambda)$ alsó becslése a (P) probléma optimális célfüggvény-értékének.*

Nyilván a legjobb alsó becslést az

$$L^* = \max_{\lambda} L(\lambda)$$

optimalizálási feladat megoldása adja. Ezt ***Lagrange multiplikátor feladat***nak nevezzük. Szintén általában igaz a következő:

2.2. Lemma.

(a) *Legyen λ egy olyan Lagrange multiplikátor és x egy olyan megengedett megoldása (P)-nek, melyre $L(\lambda) = cx$. Ekkor λ optimális megoldása a Lagrange multiplikátor problémának és x optimális megoldása (P)-nek.*

(b) *Ha valamely λ Lagrange multiplikátorra a feladat Lagrange relaxációjának egy x^* optimális megoldása megengedett megoldása (P)-nek, akkor x^* optimális megoldása (P)-nek és λ optimális megoldása a Lagrange multiplikátor feladatnak.*

Sokszor (a fejezet elején említett példában is) a relaxálandó feltétel nem egyenlőségekből, hanem egyenlőtlenségekből áll. Bár ez az úgynevezett slackváltozók segítségével könnyen visszavezethető az előzőre, talán nem haszontalan külön is megfogalmazni a kapott eredményeket.

Tehát (P)-ben az $Ax = b$ feltétel helyett $Ax \leq b$ szerepel. Ekkor viszont csak $\lambda \geq 0$ esetén kapunk alsó becslést, így a Lagrange multiplikátor feladatot módosítanunk kell:

$$L^* = \max_{\lambda \geq 0} L(\lambda).$$

Ekkor a megfelelő optimalitási kritérium:

2.3. Lemma. *Legyen $(P^{\leq})$ a $\min\{cx : Ax \leq b \text{ és } x \in X\}$ optimalizálási feladat és alkalmazzuk a Lagrange relaxációt az $Ax \leq b$ feltételre. Tegyük fel, hogy valamely λ Lagrange multiplikátorra és a Lagrange relaxáció x^* megoldására (1) x^* megengedett megoldása $(P^{\leq})$-nek és (2) $\lambda(Ax^* - b) = 0$. Ekkor x^* optimális megoldása $(P^{\leq})$-nek.*

Megjegyzés.A Lagrange relaxációs technikát Held és Karp [19, 20] fejlesztette ki, ennek segítségével adtak alsó becslést az utazó ügynök feladatra.

Most — bizonyítás nélkül — ismertetjük Carolyn Haibt Norton, Serge A. Plotkin és Tardos Éva tételét [3].

2.2. A lineáris szeparációs algoritmus

Martin Grötscheltől, Lovász Lászlótól és Lex Schrijvertől [10] származik a következő tétel.

2.4. Tétel. *Ha van egy poliéder és ehhez egy polinomiális futásidejű szeparáló algoritmus, ami a tér egy pontjáról eldönti, hogy a poliéderben van-e, és ha nincs, akkor megad egy félteret, ami a pontot elválasztja a poliédertől, akkor létezik egy polinomiális futásidejű algoritmus, ami egy lineáris célfüggvényt optimalizál a poliéderen.*

Sajnos ez az eredmény nem ad erősen polinomiális algoritmust. Hogy ezt elérjük, megkötésekkel kell élnünk.

2.5. Definíció. *Egy szeparáló algoritmust **lineáris**nak nevezünk, ha olyan korlátozott körülmények között tud működni, hogy a futása során a z input vektort (amiről tehát el kell dönteni, hogy a poliéderben van-e vagy sem) nem kapja meg explicit, hanem csak egy speciális orákulummal. Az algoritmus az orákulumhoz egy tetszőleges általa megadott a vektorral és β számmal fordulhat, az orákulum pedig megválaszolja, hogy $a \cdot z \leq \beta$ teljesül-e vagy sem.*

Megjegyzés. Legyen $\mathcal{C}$ a zárt egységkörlap a síkon, azaz $\mathcal{C} := \{z \in \mathbb{R}^2 : |x| \leq \leq 1\}$. Ehhez tudunk adni szeparáló algoritmust: egy x_0 pontot elfogadunk, ha $|x_0| \leq 1$, elutasítunk, ha nem, és ekkor $-x_0$-t megadjuk, mint szeparáló irányt. Ez viszont nem lineáris. A következő állítás mutatja, hogy nem is adható lineáris szeparáló algoritmus.

2.6. Állítás. *Ha egy $\mathcal{C}$ konvex alakzathoz van korlátos futási idejű lineáris szeparáló algoritmus, akkor az egy poliéder. Viszont a poliéder oldalainak száma exponenciálisan nagy lehet a szeparáló algoritmus futásidejében, már két dimenzióban is.*

Bizonyítás.A korlátos futási idejű lineáris szeparácló algoritmus a működése során a z-ről olyan információkat kap, hogy az általa a futása során megadott véges sok hipersík melyik oldalán van. Tehát miután lefutott, ismer egy $\mathcal{Q}$ poliédert, és azt tudja, hogy $z \in \mathcal{Q}$. Mivel ez alapján el tudja dönteni, hogy z benne van-e $\mathcal{C}$-ben, ez csak úgy lehet, ha vagy $\mathcal{Q} \subseteq \mathcal{C}$ vagy $\mathcal{Q} \cap \mathcal{C} = \emptyset$. Ha az algoritmus a futása során t kérdést tesz fel, akkor a lehetséges $\mathcal{Q}$-k száma 2^t. Ezek lefedik az egész teret, és $\mathcal{C}$ ezek közül néhánynak az egyesítése. $\mathcal{C}$

tehát poliéderek egyesítése és konvex, így maga is poliéder.
Másrészt könnyen látható, hogy a szabályos 2^n-szöglap $n+1$ összehasonlítással szeparálható.

Megjegyzés. Legyen $\mathcal{P}$ a $\{(\sin\frac{2\pi}{n}, \cos\frac{2\pi}{n}) \in \mathbb{R}^2 : n = 1, 2, \ldots\}$ pontok konvex burka. Nem túl nehéz látni, hogy ehhez a konvex halmazhoz létezik polinomiális futási idejű szeparáló algoritmus és létezik lineáris szeparáló algoritmus is, viszont ez utóbbi nem korlátos futási idejű. Az előző állítás alapján nem is létezik ilyen.

A [3]-ban a szerzők a következőt bizonyították be:

2.7. Tétel. *Ha egy poliéderhez van polinomiális futási idejű lineáris szeparáló algoritmus, akkor* ***rögzített dimenzióban*** *egy lineáris célfüggvényt erősen polinomiális időben tudunk optimalizálni a poliéderen.*

Megjegyzés.A tétel igazi megkötése a „rögzített dimenzió”, az algoritmus a dimenzióban exponenciális futásidejű. Mi a tételt egy vagy két dimenzióban használjuk, ekkor valóban hatékony algoritmust ad. Végigtekintve a kombinatorikus algoritmusokon megfigyelhetjük, hogy szinte kivétel nélkül lineárissá tehetők.

A tétel pontosabban:

2.8. Tétel. *Ha a $\mathcal{P} \subseteq \mathbb{R}^d$ konvex halmazhoz van lineáris szeparáló algoritmus, ami t ideig fut és q összehasonlítást végez, akkor $c \in \mathbb{R}^d$-hez létezik egy algoritmus, ami $O(q^d t)$ ideig fut és talál egy cx-et maximalizáló $x \in \mathcal{P}$ pontot vagy megállapítja, hogy cx nem korlátos $\mathcal{P}$-ben.*

Megemlítjük egy érdekes és fontos következményét is az előbbi tételnek:

2.9. Tétel. *Tegyük fel, hogy a $\max\{ax : Ax \leq b\}$ ($A \in \mathbb{R}^{m\times n}$) lineáris program megoldására van egy algoritmus, ami lineáris a-ban és b-ben (azaz az algoritmus inputja a és b, de ezeket a lineáris szeparáló algoritmushoz hasonlóan nem kapja meg explicit, hanem csak az ott megengedett típusú kérdéseket teheti fel róla), t ideig fut és q összehasonlítást végez. Ekkor minden k, l egész, $C \in \mathbb{R}^{k\times l}$, $B \in \mathbb{R}^{m\times l}$, $A' \in \mathbb{R}^{k\times n}$ mátrix, $a' \in \mathbb{R}^l$ és $b' \in \mathbb{R}^k$ esetén létezik egy algoritmus, ami $O(tq^{kl+l+k})$ időben megoldja a $\max\{ax + a'x : Ax + Bz \leq b, A'x + Cz \leq b'\}$ lineáris programot.*

Azaz ha egy LP-feladat olyan, hogy konstans sok sor és/vagy oszlop elhagyásával már erősen polinomiális időben megoldható, akkor az eredeti feladat is megoldható így.

E két kitérő után fordítsuk figyelmünket újra a fejezet elején említett feladatra. Célunk erősen polinomiális időben maximalizálni az $f(\lambda)$ értéket.

2.3. A példa

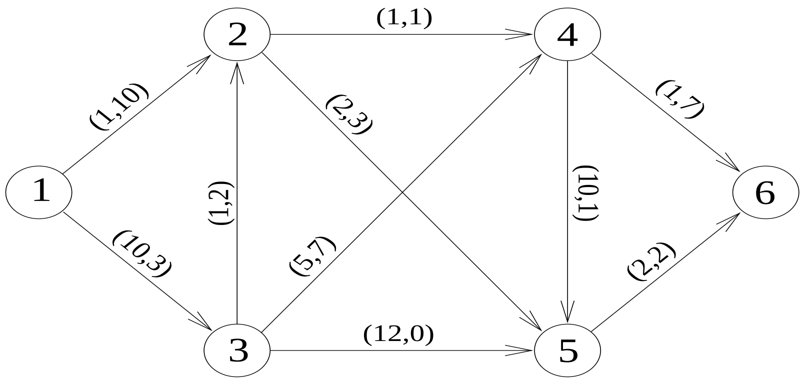
Emlékeztetünk, hogy $f(\lambda) := min\{c(p)+\lambda t(p)-\lambda T : p \in \mathcal{P}_{a,b}\}$ és feladatunk az f-et maximalizáló λ megtalálása.

Legyen $\mathcal{Q} := \{\begin{bmatrix} \lambda \\ \mu \end{bmatrix} \in \mathbb{R}^2 : \mu \leq f(\lambda)\}$. A feladatunk ekkor:

$$max\{(0,1) \cdot x : x \in \mathcal{Q}\}$$

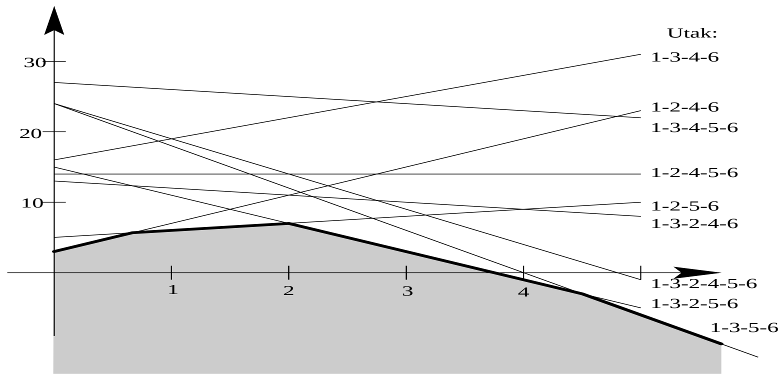
Megjegyzés. Nyilván az f függvény grafikonja éppen a $\mathcal{Q}$ határa lesz. A 2. ábrán bemutatunk egy példagráfot, a zárójelek között az élekre írt számok a c és a t súlyfüggvények. A 3. ábra mutatja az e gráfhoz tartozó f függvényt $T = 14$ esetén és a szürke tartomány a $\mathcal{Q}$. Minden p útra $c_\lambda(p) - \lambda T$ lineáris λ-ban, azaz a képe egy egyenes és f ezen egyenesek alsó burkolója, továbbá

$$\mathcal{Q} := \bigcap_{p \in \mathcal{P}_{a,b}} \left\{ \begin{bmatrix} \lambda \\ \mu \end{bmatrix} \in \mathbb{R}^2 : \mu \leq c_\lambda(p) \right\}.$$



2. Ábra: A gráf, a súly- és a költségfüggvény

A 2.7. tétel alapján elég $\mathcal{Q}$-ra egy lineáris szeparáló algoritmust mutatnunk. Legyen a vektorunk $\begin{bmatrix} \lambda_0 \\ \mu_0 \end{bmatrix}$, erről kell eldöntenünk, vajon $\mathcal{Q}$-ban van-e? Az algoritmus a következő: c_{λ_0}-ra mint súlyfüggvényre futtassuk le például a Dijkstra algoritmust, és a végén a kapott legrövidebb út hosszát hasonlítsuk össze $\mu_0 + \lambda_0 T$-vel. Ha kisebb vagy egyenlő, akkor elfogadunk, ha nagyobb, akkor elutasítunk. Vegyük észre, hogy ez az algoritmus lineáris $\begin{bmatrix} \lambda_0 \\ \mu_0 \end{bmatrix}$-ban. Valóban, a Dijkstra algoritmus a gráfon adott súlyfüggvényt aképpen



3. Ábra: A Lagrange függvény $T = 14$ esetén

használja csak fel, hogy két (a-ból induló) út hosszát hasonlítja össze. Ha most az élek és az utak súlyait szimbolikusan tartjuk nyilván, akkor a p_1 és a p_2 út összehasonlítása a $c(p_1) + \lambda_0 t(p_1) < c(p_2) + \lambda_0 t(p_2)$ kérdés eldöntését jelenti, ami vagy λ_0-tól függetlenül eldönthető, vagy, ha például $t(p_2) > t(p_1)$, akkor a $\frac{c(p_1)-c(p_2)}{t(p_2)-t(p_1)} < \lambda_0$ eldöntésére redukálódik, ami persze lineáris $\begin{bmatrix}\lambda_0\\ \mu_0\end{bmatrix}$-ban. A Dijkstra algoritmus futtatása után tehát kezünkben van egy p^* optimálisnak mondott út, és el kell döntenünk, hogy $c(p^*) + \lambda_0 t(p^*) - \lambda_0 T \leq \mu_0$ teljesül-e? Ez szintén egy lineáris összehasonlítás. Amennyiben úgy találjuk, hogy $\begin{bmatrix}\lambda_0\\ \mu_0\end{bmatrix} \notin \mathcal{Q}$, akkor meg kell adnunk egy szeparáló félteret is. Könnyen látható, hogy $t(p^*) - T$ az f λ_0-beli (egyik) érintőjének a meredeksége, így a $(t(p^*) - T, -1)$ vektor egy szeparáló irányt határoz meg.

Mivel az eljárás működési elvének megértése is célunk, pontosan megfogalmazzuk az — ebben a speciális esetben lényegesen leegyszerűsödő — optimalizáló algoritmust.

Jelöljük az f maximumhelyét λ_{opt}-tal. Ez lehet egyetlen pont vagy egy intervallum.

2.10. Állítás. *Egy adott λ-ról könnyen el tudjuk dönteni, hogy optimális-e illetve ha nem az, akkor azt hogy $\lambda_{opt} < \lambda$ és $\lambda_{opt} > \lambda$ közül melyik teljesül.*

Bizonyítás. A Dijkstra algoritmussal keressünk meg egy c_λ-t minimalizáló p_λ utat. Ha erre $t(p_\lambda) - T = 0$, akkor a fejezet elején tett meggondolás alapján λ optimális. Ha $t(p_\lambda) - T < 0$ és $\lambda' > \lambda$, akkor

$$f(\lambda') \leq c(p_\lambda) + \lambda'(t(p_\lambda) - T) < c(p_\lambda) + \lambda(t(p_\lambda) - T) = f(\lambda),$$

vagyis $\lambda_{opt} \leq \lambda$. Hasonlóan, ha $t(p_\lambda) - T > 0$, akkor $\lambda_{opt} \geq \lambda$. El kell még döntenünk, hogy $t(p_\lambda) - T \neq 0$ esetén λ optimális-e? λ pontosan akkor optimális, ha f-nek a λ-beli jobboldali deriváltja nem nagyobb, mint 0 és a baloldali deriváltja nem kisebb, mint 0. (Ezek a deriváltak léteznek, mert f szakaszonkét lineáris.) Tekintsük ehhez a lefuttatott Dijkstra algoritmus által megadott potenciál szerinti pontos élek gráfját. Ismeretes, hogy a c_λ-t minimalizáló $a \rightsquigarrow b$ utak pontosan a csak ezen éleket használó utak. Keressük meg a pontos élek gráfjában a t-t minimalizáló p_1 és maximalizáló p_2 $a \rightsquigarrow b$ utakat. Ezeket gyorsan meg tudjuk keresni, mert a pontos élek gráfja aciklikus. Nyilván $t(p_1) - T$ lesz az f-nek a λ-beli jobboldali és $t(p_2) - T$ a baloldali deriváltja. ☺

Most kezdjük el futtatni a Dijkstra algoritmust c_λ-ra úgy, hogy a λ-t nem adjuk meg előre, hanem ha az algoritmus rákérdez a λ-ra, azaz megkérdezi, hogy pl. λ kisebb-e mint egy λ_0, akkor meghatározzuk hogy λ_{opt} kisebb-e vagy nagyobb, mint λ_0, és eszerint válaszolunk. Vagyis mindig az optimális λ irányába tereljük az algoritmust. Miután az algoritmus lefutott, megad egy optimális utat. A λ pontos értékét nem ismeri, a válaszaink alapján lehetséges értékek egy I intervallumot alkotnak. Mi olyan válaszokat adtunk az algoritmus kérdéseire, hogy ezen I tartalmazzon optimális λ-t. Mivel létezik egy olyan út, ami az egész intervallumon optimális, ezért f lineáris I-n, így az intervallum valamelyik végpontja maximalizálni fogja f-et.

Az algoritmus erősen polinomiális, hiszen a Dijkstra algoritmus is az. Pontosabban:

2.11. Tétel. *Ha a gráfnak n pontja és m éle van, akkor az f-et maximalizáló λ-t a fent leírt algoritmus $O(m^2 + n^2 \log^2 n)$ lépésben maximalizálja.*

Bizonyítás.A Dijkstra algortimus futási ideje $O(m+n\log n)$, ha a Fibonacci heap-et alkalmazzuk a prioritásos sor kezelésére. Az algoritmus tehát legfeljebb ennyiszer kérdez rá a λ-ra. Minden ilyen kérdésre az adandó válasz a Dijkstra algoritmus egy futtatásával számítható ki. Így a teljes futási idő $O((m + n\log n)^2) = O(m^2 + n^2 \log^2 n)$. ☺

Megjegyzés. Mivel a minimális út probléma megegyezik egy egészértékű folyamfeladattal, érdemes megvizsgálni a feladatunknak egy másik relaxációját is, mégpedig azt, hogy a feladatot felírjuk egészértékű LP feladatként, relaxáljuk az egészértékűségi feltételt, és megoldjuk a kapott LP feladatot. Az alábbiakban megmutatjuk, hogy ezzel a módszerrel ugyanazt az alsó becslést kapjuk és a relaxált feladat megoldása is ugyanazon probléma megoldására vezetődik vissza.

Az egészértékűségi feltételt elhagyva a következő feladatot kapjuk: Keressük

$$\min cx \text{ -et}$$

a

$$\begin{aligned} x &\geq 0 \\ \sum_{e\in\varrho(v)} x_e - \sum_{e\in\delta(v)} x_e &= \begin{cases} -1, & \text{ha } v = a, \\ +1, & \text{ha } v = b \text{ és} \\ 0 & \text{máskor.} \end{cases} \qquad \forall v \in V\text{-re} \\ -\sum_{e\in E} t_e x_e &\geq -T \end{aligned}$$

rendszerre nézve, ahol $x \in \mathbb{R}^E$. Áttérve a duális feladatra keressük

$$\max \pi_b - \pi_a - \lambda T \text{ -t}$$

a

$$\lambda \geq 0 \tag{1}$$

$$\pi_j - \pi_i \leq c_e + \lambda t_e \quad \forall e \in E, e = (i,j)\text{-re} \tag{2}$$

poliéderben ($\pi \in \mathbb{R}^V$ és $\lambda \in \mathbb{R}$). A (2) feltétel éppen azt fejezi ki, hogy π megengedett potenciál a $c + \lambda t$ súlyfüggvényre nézve. Ismeretes, hogy egy pontpárra a megengedett potenciálkülönbség maximuma megegyezik a pontpár között vezető legrövidebb úttal. Ebből már következik az megjegyzésben megfogalmazott állítás.

2.4. Minimális feszítőfa fokszámmegkötéssel

Legyen adott egy $\mathcal{G} = (V, E)$ irányítatlan gráf, rajta egy $c : E \longrightarrow \mathbb{R}$ költségfüggvény. Adott továbbá egy $v \in V$ pont és egy k pozitív egész. A feladatunk keresni egy minimális súlyú olyan feszítőfát, amelynek a v-beli foka k. Tehát keressük

$$\min cx \quad -et$$

a

$$\sum_{e\in\delta(v)} x_e = k \tag{3}$$

$$x \in X$$

feltételek mellett, ahol X a $\mathcal{G}$-beli feszítőfák karakterisztikus vektorainak halmaza. Alkalmazzuk a Lagrange-relaxációt a (3) feltételre. Ekkor a Lagrange-függvény $L(\lambda) := min\{cx + \lambda \sum_{e\in\delta(v)} x_e - \lambda k : x \in X\}$ lesz. Ez egy paraméteres minimális feszítőfa feladat. Mivel rögzített λ mellett λk konstans, így egy adott λ-ra $L(\lambda)$ kiszámítása egy minimális feszítőfa keresését jelenti a

$$c_\lambda(e) := \begin{cases} c(e) & \text{ha } e \notin \delta(v) \text{ és} \\ c(e) + \lambda & \text{ha } e \in \delta(v) \end{cases}$$

súlyfüggvényre nézve. A következőkben belátjuk, hogy amennyiben az eredeti feladatnak van megengedett megoldása, akkor tudunk találni egy λ^* Lagrange multiplikátort és e multiplikátor mellett a relaxált feladatnak egy optimális megoldását, amely az eredeti feladatnak megengedett megoldása. A 2.2. lemma alapján ez *optimális megoldása* lesz az eredeti feladatnak.

Legyen először $\lambda = 0$ és T^0 egy c_0-minimális feszítőfa. Ha T^0-ben pontosan k darab v-ből kiinduló él van, akkor készen vagyunk. Tegyük most fel, hogy kevesebb, mint k v-ből kiinduló T^0-beli él van. Ekkor elkezdjük csökkenteni λ-t és elkészítjük feszítőfáknak egy $T^1, \ldots, T^q$ sorozatát, úgy hogy mindegyik T^j minimális feszítőfa lesz egy λ^j-re, és T^{j+1}-et úgy kapjuk, hogy T^j egy nem $\delta(v)$-beli élét lecseréljük egy $\delta(v)\backslash T^j$-beli élre (így a v-ből kiinduló élek száma egyel növekedik). Legyen $e \in \delta(v)\backslash T^j$. Mivel T^j minimális, ezért az e alapkörében levő súlyok nem nagyobbak, mint $c_{\lambda^j}(e)$. Válasszuk $\lambda_e^{j+1} \leq \lambda^j$-t úgy, hogy $c_{\lambda_e^{j+1}}(e)$ megegyezzen az e alapkörében levő e-n és a másik $\delta(v)$-beli élen kívüli élek súlyának maximumával. Legyen e_{j+1} az az él, amire $\lambda_{e^{j+1}}^{j+1}$ a legnagyobb, legyen $\lambda^{j+1} := \lambda_{e^{j+1}}^{j+1}$ és legyen f_{j+1} az e alapkörén levő e-n kívüli legnehezebb él úgy, hogy $f_{j+1} \notin \delta(v)$. Ekkor $T^{j+1} := T^j + e_{j+1} - f_{j+1}$ megfelel a kívánalmainknak.

Ha a T^0-ban a v-ből kiinduló élek száma nagyobb, mint k, akkor hasonlóan járunk el, a λ növelése mellett egyesével csökkentjük a v-ből kiinduló élek számát.

Megjegyzés. Tekinthetjük azt az általánosítását a feladatnak, hogy nem egy, hanem több ponton is fokszámmegkötést teszünk. Ez egy $\mathcal{NP}$-teljes feladat. Ha ugyanis $a, b \in V$, akkor a-ra és b-re egyet, a többi pontra kettőt írva ez éppen egy $a \rightsquigarrow b$ Hamilton út keresését jelenti. Ha azonban csak egy $F \subseteq V$ *független* ponthalmazon írjuk elő a fokokat, akkor már kezelhetővé válik a feladat, bár egészen más eszközökkel, mint ahogyan az egy pontra tett fokszámmegkötés esetén tettük. Könnyű látni, hogy ekkor az olyan élhalmazok, amelyeknek az F-beli pontokon a foka pontosan az előírt fokszám, egy matroid bázisait alkotják. Mivel a feszítőfák élhalmazai szintén egy matroid bázisai, így a feladatunk egy minimális súlyú közös bázis keresése. Az Edmonds-féle *súlyozott matroid metszet* algoritmus éppen ezt a feladatot oldja meg.

3. Néhány paraméteres feladat

3.1. Konzervatív súlyozás

3.1. Definíció. *Egy (irányított vagy irányítatlan) gráf élsúlyozását **konzervatívnak** hívunk, ha nincs benne negatív (azaz negatív összsúlyú) kör.*

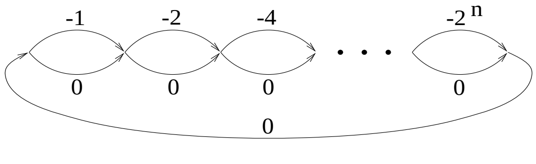
A konzervatív súlyozás igen lényeges szerepet játszik a legrövidebb utak keresésekor, mert a súlyozás konzervatívsága biztosítja, hogy a legrövidebb séta és a legrövidebb út hossza egybeessen.

Tekintsük a következő feladatot: Tegyük fel, hogy a súlyozás nem konzervatív és keressük azt a legkisebb α értéket, melyre a gráf éleit α-val egységesen megnövelve a súlyozás konzervatív lesz. Könnyen látható, hogy α éppen a minimális körátlag. A következőkben ennek meghatározására adunk algoritmust.

3.1.1. Irányított gráfok

Természetesen adódik a következő algoritmus: Keresünk egy negatív kört a gráfban (ez könnyen megoldható dinamikus programozással, például a legrövidebb út kereső algoritmusok is rábukkannak egy negatív körre, ha a súlyozás nem konzervatív), e kör átlagának ellentettjével megemeljük az élek súlyát, majd ezt addig ismételjük, amíg van negatív kör. Végül az emelések összegének -1-szerese adja minimális körátlagot.

Sajnos ez nem ad polinomiális algoritmust. Tekintsük a 4. ábrán látható gráfot. Ebben $2^{n+1} - 1$ különböző súlyú negatív kör található. Ha mindig a legkisebb abszolút értékű negatív körátlag ellentettjével növelünk, akkor exponenciális sok emelést kell végeznünk.



4. Ábra: Egy gráf $2^{n+1} - 1$ különböző súlyú negatív körrel

Tegyük most fel, hogy a legnegatívabb kört gyorsan meg tudjuk találni. Álljon ez k élből. Ekkor e kör átlagának -1-szeresével megemelve minden él súlyát, minden legalább k élből álló kör súlya nemnegatívvá vált, így az emelés után egy negatív kör legfeljebb $k-1$ élből állhat. Tehát a legnegatívabb kör átlagának ellentettjével való növelést legfeljebb n-szer iterálva megkapjuk a minimális körátlagot.

Ez még mindig nem ad polinomiális algoritmust, mert a minimális kör probléma tartalmazza a Hamilton kör problémát, így $\mathcal{NP}$-teljes.

A segítséget a körséták jelentik számunkra. A legfeljebb n hosszú körséták közül a minimális súlyút már dinamikus programozással meg tudjuk keresni. Mivel egy gráfban pontosan akkor van negatív legfeljebb n hosszú körséta, ha van negatív kör, továbbá — mivel egy körséta körök (multiplicitással vett) uniója — minden negatív körséta átlaga nem kisebb mint a minimális körátlag, így a minimális körséta negatív átlagával növelve az élek súlyát, az előbbi meggondolás alapján legfeljebb n iteráció után elérjük, hogy ne legyen negatív kör. Kaptuk:

3.2. Tétel. *Egy gráfban a minimális körátlagot $O(n^2m)$ lépésben ki tudjuk számolni.*

Algoritmusunkon javíthatunk a következő észrevétellel. Az előbb említettük, hogy a legfeljebb n hosszú körséták átlaga nem kisebb, mint a minimális körátlag. Másrészt nem is nagyobb, hiszen a körséták között szerepelnek a valódi körök is. A minimális körsétaátlagot pedig ki tudjuk számolni, hiszen ez egyenlő $\min\limits_{1\leq k\leq n} c_k/k$-val, ahol c_k a legkisebb súlyú pontosan k élű körséta súlya. Ez megfelelő adatstruktúra alkalmazása esetén a következőt adja:

3.3. Tétel. *Az előző algoritmus lépésszáma $O(nm)$-re javítható.*

3.1.2. Irányítatlan gráfok

Irányítatlan gráfok esetén hasonló módon járunk el, mint az irányított esetben. A gondot az jelenti, hogy egy irányítatlan gráfban egy körséta nem okvetlenül áll elő, mint körök uniója (Például, ha $(u,v)\in E$, akkor (u,v,u) egy körséta, de nem kör és nem is körök uniója). Ezért a minimális körséta átlaga helyett egy minimális Euler-részgráf negatív átlagával fogunk emelni.

3.4. Definíció. *Legyen $\mathcal{G} =< V,E >$ egy irányítatlan gráf és $T\subseteq V$. Egy $F\subseteq E$ halmazt T-**kötésnek** nevezünk, ha minden $v\in T$ esetén $d_F(v)$ páratlan és minden $v\notin T$ esetén $d_F(v)$ páros.*

J. Edmonds és E. Johnson [4] bizonyította be a következő tételt:

3.5. Tétel. *A $c : E \longrightarrow \mathbb{R}$ súlyfüggvényre nézve minimális T-kötést $O(n^3)$ lépésben meg tudjuk találni.*

$T = \emptyset$ esetén az algoritmus éppen a minimális súlyú Euler-részgráfot találja meg. Mivel itt is igaz, hogy egy minimális súlyú Euler-részgráf átlagának -1-szeresével növelve, ha az k élből állt, akkor az új gráfban egy negatív Euler-részgráf legfeljebb $k-1$ élet tartalmazhat, ezért legfeljebb m iteráció után ez az algoritmus is befejeződik. Vagyis:

3.6. Tétel. *Egy irányítatlan gráfban a minimális körátlag $O(mn^3)$ lépésben kiszámítható.*

3.1.3. Súlyozott eset

Az előző feladatok egy lehetséges általánosítása, hogy a $\mathcal{G} = (V, E)$ (irányított vagy irányítatlan) gráf és $c, w : E \longrightarrow \mathbb{R}$ súlyfüggvények esetén keressük azt a minimális $\lambda \geq 0$ számot, amire a $w + \lambda c$ súlyfüggvénnyel a gráf nem tartalmaz negatív kört. Az előző két alfejezetben a $c \equiv 1$ esetet vizsgáltuk. Az ott leírt algoritmus itt is alkalmazható (csak a *körséta-átlag*ot kell $\frac{w(C)}{c(C)}$-nek definiálni). Nem tudjuk azonban igazolni, hogy ez polinomiális sok emelés után megtalálja a keresett λ-t. Ezért mélyebb eszközöket kell felhasználnunk.

Itt is a 2.7. tétel szeparációs algoritmusa van segítségünkre. Legyen a poliéderünk $\mathcal{P} := \{\lambda \in \mathbb{R} : \lambda \geq 0$ és $w + \lambda c$-ben nincs negatív kör$\}$. Ekkor $\min\{\lambda : \lambda \in \mathcal{P}\}$-t keressük. $\mathcal{P}$-re pedig tudunk szeparációs algoritmust adni: Adott λ_0-ra keresünk egy negatív kört illetve irányítatlan esetben negatív Euler részgráfot $w + \lambda_0 c$-ben, ha nem találunk, akkor elfogadunk, ha pedig találunk, akkor elutasítunk és visszaadjuk „$\lambda > \lambda_0$"-t mint szeparáló irányt. Ez az algoritmus könnyen láthatóan λ-ban lineárissá alakítható (a minimális Euler részgráf keresése legrövidebb út és minimális súlyú párosítás keresésére vezetődik vissza).

A konkrét algoritmus megfogalmazása sem okoz nehézséget a 2.3. fejezet alapján, így ettől eltekintünk.

3.1.4. Inhomogén növelés

Egy másik természetesen adódó kérdés, hogy a negatív kört úgy akarjuk megszüntetni, hogy az élek súlyát nem csak egységesen, hanem egymástól függetlenül is növelhetjük. Azaz egy $w : E \longrightarrow \mathbb{R}$ súly- és egy $c : E \longrightarrow \mathbb{R}_+$ költségfüggvény mellett keressük a

$$\min\{cz : z \in \mathbb{R}^E, z \geq 0 \text{ és } w + z \text{ nem tartalmaz negatív kört}\}$$

értéket. (Megjegyzendő, hogy ez nem adja semelyik előző alfejezetben tárgyalt probléma általánosítását.) A következőkben erre adunk erősen polinomiális algoritmust irányított gráfok esetén.

Ismert tény, hogy egy súlyozott irányított gráf pontosan akkor nem tartalmaz negatív kört, ha létezik a súlyozásra nézve megengedett potenciál. Ez alapján a feladat ekvivalens a következő LP feladattal:

Keressük

$$\min cz \quad \text{-t}$$

a

$$\begin{aligned} z &\geq 0 \\ \pi_j - \pi_i &\leq w_e + z_e \quad \forall e \in E, e = (i,j)\text{-re} \end{aligned}$$

feltételekre nézve, ahol $\pi \in \mathbb{R}^V$ és $z \in \mathbb{R}^E$. Tekintve a feladat duálisát a következőt kapjuk. Keressük

$$\min yw \quad \text{-t}$$

a

$$y \geq 0 \tag{4}$$

$$\sum_{e \in \varrho(v)} y_e - \sum_{e \in \delta(v)} y_e = 0 \quad \forall v \in V\text{-re} \tag{5}$$

$$y \leq c \tag{6}$$

feltételek mellett, ahol $y \in \mathbb{R}^E$. A (5) feltétel azt mondja ki, hogy y egy áram, vagyis a feladat ekvivalens egy w-minimális költségű y áram keresésével, ahol $0 \leq y \leq c$. Erre ismeretes erősen polinomiális algoritmus, ami megadja a duális megoldást is.

Megjegyzés.A 3.3. fejezetben szükségünk lesz arra, hogy bizonyos éleken ne engedjük meg az emelést. Ez könnyen láthatóan a duális feladatban úgy jelenik meg, hogy a megfelelő élekhez tartozó duális változóra (6)-ban felső korlátot nem adunk meg (azaz $+\infty$ a felső korlát). A minimális költségű áramot találó algoritmus pedig tudja kezelni ezt az esetet is.

Megjegyzés.A fenti módszer nem alkalmazható irányítatlan esetben, mivel a megengedett potenciál létezése csak irányított gráfok esetén alkalmazható a súlyozás konzervatívságának karakterizálása. Irányítatlan gráfok esetében nem is ismerünk hatékony algoritmust erre a feladatra.

3.1.5. Inhomogén növelés felső korláttal

Tekintsük megint az előző feladatot (irányított esetben), azzal a különbséggel, hogy korlátozzuk az éleken való növelést, azaz megkötjük még, hogy

$$z \leq a,$$

ahol $a \in \mathbb{R}^E$ és $a \geq 0$. Ekkor a feladat duálisa a következő lesz.
Keressük

$$\min yw + ua \ \text{ -t}$$

a

$$\begin{aligned} 0 &\leq y && (7) \\ 0 &\leq u && (8) \\ \sum_{e \in \varrho(v)} y_e - \sum_{e \in \delta(v)} y_e &= 0 \qquad \forall v \in V\text{-re} && (9) \\ y &\leq c + u && (10) \end{aligned}$$

feltételek mellett, ahol $y, u \in \mathbb{R}^E$. Legyenek $y', y'' \in \mathbb{R}^E$ új változók és cseréljük le y-t $y' + y''$-re, úgy, hogy (7) és (10) helyett a

$$0 \leq y' \leq c$$
$$0 \leq y'' \leq u$$

megkötéseket tesszük, a célfüggvényt pedig lecseréljük

$$\min y'w + y''w + ua \ \text{ -ra.}$$

Ez a probléma természetesen ekvivalens az előzővel. Tekintsünk most egy optimális megoldást. Mivel $a \geq 0$, ezért u komponenseit csökkentve a célfüggvény nem nő, így feltehető, hogy $y'' = u$. Ekkor tehát

$$\min y'w + y''(w + a) \ \text{ -t}$$

keressük a

$$\begin{aligned} 0 \leq y' &\leq c \\ 0 &\leq y'' \\ \sum_{e \in \varrho(v)} (y'_e + y''_e) - \sum_{e \in \delta(v)} (y'_e + y''_e) &= 0 \qquad \forall v \in V\text{-re} \end{aligned}$$

feltételek mellett. Ez pedig szintén egy minimális költségű áram feladat: Minden e élet megduplázunk, az egyikre c_e a felső korlát és w_e a költség, a másikon nincs felső korlát, de a költsége $w_e + a_e$. Az alsó korlát minden élen 0.

3.2. A Newton módszer

Egy $f : \mathbb{R} \longrightarrow \mathbb{R}$ függvény egy gyökhelyének meghatározására szolgál a következő — Newton módszer néven ismert — eljárás. Induljunk el egy $x_0 \in \mathbb{R}$ számból, és ebből iteratívan elkészítünk egy (x_n) sorozatot a következőképpen:

$$x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)}$$

Az f-re tett bizonyos megkötések mellett ez értelmes és konvergálni fog egy gyökhelyhez. Talán a legegyszerűbb ezek közül:

3.7. Állítás. *Ha f deriválható, konvex, létezik gyöke és $f'(x_0) \neq 0$, akkor x_n konvergál valamelyik gyökhöz.*

Mi szakaszonként lineáris konvex függvényekkel foglalkozunk, ekkor a függvény deriváltján a pontbeli *valamelyik* (például a legkevésbé meredek) érintő meredekségét értjük. Ha még a függvény véges sok törésponttal rendelkezik, akkor így véges időn belül megtalálunk egy gyököt.

A következőkben azt mutatjuk be, hogy miként lehet ezt a módszert bizonyos típusú függvények minimalizálására felhasználni.

Keressük

$$\sigma := \min\left\{\frac{s(X)}{c(X)} : X \in \mathcal{A}, c(X) > 0\right\} - t,$$

ahol $\mathcal{A}$ egy véges halmaz és $s, c : \mathcal{A} \longrightarrow \mathbb{R}$ nemnegatív függvények. Tekintsük most a következő paraméteres problémát:

$$a(\lambda) := \min\{s(X) - \lambda c(X) : X \in \mathcal{A}\}$$

Legyen $X_\lambda \in \mathcal{A}$ olyan, ahol a fenti minimum felvétetik, azaz amire $a(\lambda) = s(X_\lambda) - \lambda c(X_\lambda)$ és X^*, amire $c(X^*) > 0$ és $\sigma = \frac{s(X^*)}{c(X^*)}$. Ekkor, ha $\lambda \leq \sigma$, akkor

$$a(\lambda) = s(X_\lambda) - \lambda c(X_\lambda) \geq s(X_\lambda) - \sigma c(X_\lambda) \geq s(X_\lambda) - \frac{s(X_\lambda)}{c(X_\lambda)} = 0$$

és ha $\lambda \geq \sigma$, akkor, mivel $c(X^*) \neq 0$,

$$a(\lambda) \leq s(X^*) - \lambda c(X^*) < s(X^*) - \sigma c(X^*) = 0.$$

Ezzel kaptuk, hogy

$$\sigma = \max\{\lambda : a(\lambda) \geq 0\},$$

és ha $a(\lambda)$-t ki tudjuk számolni úgy, hogy közben egy X_λ-t is találunk, akkor — tekintve, hogy $a(\lambda)$ konkáv függvény — adódik a következő algoritmus: Legyen először $\lambda_0 := \frac{s(X_0)}{c(X_0)}$ valamelyik $A \in \mathcal{A}$-ra, amire $c(A) \neq 0$. Az algoritmus során végig $\lambda_n \geq 0$. Ezután kiszámítjuk $a(\lambda_n)$-t és X_{λ_n}-t. Ha $a(\lambda_n) < 0$, akkor legyen $\lambda_{n+1} := \frac{s(X_{\lambda_n})}{c(X_{\lambda_n})}$, és ezt folytassuk, amíg $a(\lambda_n) \geq 0$ nem lesz. Ekkor viszont az elöbbiek szerint $\lambda_n \leq \sigma$, így $\lambda_n = \sigma$, tehát készen vagyunk.

Most megadunk két — Cunninghamtől [8] származó — esetet, amikor ez az algoritmus gyorsan megtalálja σ-t.

3.8. Lemma. *Tegyük fel, hogy $\lambda \geq \sigma$, $A \in \mathcal{A}$ minimalizálja $s - \lambda c$-t, $\lambda' := = \frac{s(A)}{c(A)}$ és $B \in \mathcal{A}$-ra $s(B) - \lambda' c(B) < 0$. Ekkor $c(B) < c(A)$.*

Bizonyítás.

$$\begin{aligned} 0 &> s(B) - \lambda' c(B) \\ &= s(B) - \lambda c(B) + \lambda c(B) - \lambda' c(B) \\ &\geq s(A) - \lambda c(A) + \lambda c(B) - \lambda' c(B) = (c(B) - c(A))(\lambda - \lambda'), \end{aligned}$$

amiből $c(B) < C(A)$ következik. ☺

Tehát ha c értékkészlete kicsi, akkor ez biztosítja a gyors konvergenciát.

Megjegyzés.A 3.1. fejezetben tárgyalt minimális körátlag is egy ilyen alakú probléma, $\mathcal{A}$ a legfeljebb n hosszú körséták halmaza, egy $C \in \mathcal{A}$-ra $s(C)$ a körséta súlya és $c(C)$ a körséta élszáma. Ekkor $a(\lambda)$ kiszámítása egy minimális körséta keresését jelenti az $s - \lambda 1$ súlyozás mellett. Tehát ez a módszer éppen a közölt minimális körátlagot kiszámító egyszerűbb eljárást adja.

Megjegyzés.A 3.5. fejezet valamint a negyedik fejezet algoritmusa is ezen az eljáráson alapul.

Legyen most E egy alaphalmaz és $\mathcal{A} = \mathcal{P}(E)$.

3.9. Lemma. *Legyen s szubmoduláris, $c \geq 0$ pedig olyan szupermoduláris függvény, amire $c(\emptyset) = 0$. Tegyük fel, hogy $\lambda \geq \sigma$, $A \in \mathcal{A}$ minimalizálja $s - \lambda c$-t, $\lambda' := \frac{s(A)}{c(A)}$, $B \in \mathcal{A}$ minimalizálja $s - \lambda' c$-t és mindkét minimum negatív. Ekkor $A \cap B \neq A$ és $A \cap B$ szintén minimalizálja $s(X) - \lambda' c(X)$-et.*

Bizonyítás. Mivel $s(X) - \lambda' c(X)$ szubmoduláris ($\lambda' \geq 0$) és $s(A) - \lambda' c(A) = = 0$, ezért

$$s(B) - \lambda' c(B) \geq s(A \cup B) - \lambda' c(A \cup B) + s(A \cap B) - \lambda' c(A \cap B).$$

$s(A \cup B) - \lambda' c(A \cup B) \geq 0$, mert különben a 3.8. lemmát alkalmazva azt kapnánk, hogy $c(A \cup B) < c(A)$, ami lehetetlen. Így viszont $s(B) - \lambda' c(B) \geq$ $\geq s(A \cap B) - \lambda' c(A \cap B)$, amint azt állítottuk.

Ha tehát az algoritmust úgy módosítjuk, hogy $X_{\lambda_{n+1}}$-et $X_{\lambda_n} \cap A$-nak választjuk, ahol A az $s - \lambda_n c$-t minimalizáló halmaz, akkor az algoritmus legfeljebb $|E|$ iteráció után végetér.

Megjegyzés. Ez a lemma fontos szerephez jut a 3.5. fejezet irányított megfelelőjének, a *gyökeres erősség*nek a számításakor [8]. Dolgozatunkban ez nem szerepel.

3.3. Inverz útproblémák

Kétféle értelmes inverz út feladat is megfogalmazható aszerint, hogy az $a,b \in V$ pontra az $a \rightsquigarrow b$ út hosszát akarjuk egy meghatározott értékre beállítani vagy kijelölünk egy $a \rightsquigarrow b$ utat és olyan megváltoztatását keressük a súlyoknak, hogy a kijelölt út minimális legyen.

Tegyük fel, hogy a gráf irányított, az éleken adott egy $w : E \longrightarrow \mathbb{R}_+$ súly- és egy $c : E \longrightarrow \mathbb{R}_+$ költségfüggvény és keressük azt a minimális költségű $z \in \mathbb{R}^E$-t, amire a $w + z$ súlyfüggvény a kívánt tulajdonságú. Mindkét feladatot visszavezethetjük a 3.1.4. pontban megfogalmazott problémára.

Legyen tehát p_0 egy rögzített $a \rightsquigarrow b$ út. Keressük tehát azt a z-t, amelyre p_0 egy legrövidebb $a \rightsquigarrow b$ út a $w + z$ súlyozásra nézve és amelyre $\sum_{e \in E} c_e |z_e|$ minimális (azaz a változtatás kerül pénzbe).

3.10. Állítás. *Legyen z a feltételeknek megfelelő optimális megoldás. Ekkor z választható olyannak, hogy minden $e \in p_0$-ra $z_e \leq 0$ és minden $e \notin p_0$-ra $z_e \geq 0$.*

Bizonyítás. Ha z megengedett megoldás, akkor a fenti feltételt sértő e élekhez tartozó z_e-ket nullára változtatva a $w + z$ továbbra is eleget tesz a követelményeknek és cz nem nő.

3.11. Állítás. *Legyen $\mathcal{G} = (V, E)$ egy súlyozott irányított gráf és tegyük fel, hogy a súlyozás konzervatív. Legyen továbbá $a, b \in V$ és p egy $a \rightsquigarrow b$ út. Ez esetben p pontosan akkor legrövidebb $a \rightsquigarrow b$ út, ha p éleit megfordítva és a súlyaikat negálva a kapott gráfban nincs negatív kör.*

Mivel z a p_0 élein nempozitív, ezért ez a negálás után nemnegatív lesz, így a p_0 éleit megfordítva és a w-t negálva a p_0 élein éppen a 3.1.4. pontban megfogalmazott problémát kapjuk.

Legyen most $T \in \mathbb{R}_+$ olyan, ami nagyobb az a—b távolságnál és keressük azt a $z \geq 0$ vektort, melyre a $w + z$-minimális út hossza legalább T és ezen belül cz minimális.

Ennek megtalálásához húzzunk be egy új e élet b-ből a-ba és legyen $w_e = -T$. Ekkor megint olyan z-t keresük, amire a gráfban a $w + z$ súlyozásban nincs negatív kör és cz minimális, de úgy, hogy az új e él súlyát nem növelhetjük. Ilyen z vektort a 3.1.4. pont végén tett megjegyzés alapján szintén tudunk találni.

3.4. Vágások és folyamok

E rövid alfejezetet az inverz folyamfeladatoknak szenteljük. Itt az irányítatlan esetet nem érdemes külön vizsgálni, mert az éleket mindkét irányban irányítva behúzva a feladatokat visszavezethetjük az irányított esetre.

Legyen tehát $\mathcal{G} := (V, E)$ egy irányított gráf, élein egy $c \geq 0$ kapacitásfüggvény és $s, t \in V$. Az élek kapacitásait szeretnénk úgy megnövelni, hogy az $s - t$ pontok között létezzen F értékű folyam vagy ami ezzel ekvivalens, a minimális $s - t$ vágás súlya legalább F legyen.

A *homogén eset* ugyanúgy kezelhető, mint a konzervatív súlyozás keresésénél láttuk. *Súlyozatlan eset*ben a Newton módszer ad gyors eljárást (a minimális vágás súlyát F-re felemelve az F-nél kisebb vágások élszáma kisebb, mint a felemelt vágás élszáma), *súlyozott eset*ben pedig a 2.2. fejezet lineáris szeparációs eljárását használva kapunk erősen polinomiális algoritmust.

Vizsgáljuk most az *inhomogén eset*et. Ekkor adott a c kapacitásfüggvény mellett egy d költségfüggvény is, ami megadja egy él kapacitásának egy egységgel való növelésének költségét és keressük azt a $z \in \mathbb{R}^E$, $z \geq 0$ vektort, amelyre a $c + z$ élkapacitások mellett létezik F értékű $s - t$ folyam és dz minimális. Ezt LP-feladatként felírva éppen a 3.1.5. fejezet utolsó pontjában megfogalmazott probléma duálisát kapjuk. Az ottani megfontolások e problémának is megoldását adják. Tehát az algoritmus a következő. Minden e élet megduplázunk, az egyikre c_e kapacitást és 0 költséget, a másikra ∞ kapacitást és d_e költséget írunk és ebben keresünk egy minimális költségű F értékű $s - t$ folyamot. A ∞ kapacitású éleken levő folyamértékek adják az egyes éleken szükséges emelés mértékét. Az emelés költsége éppen a folyam költsége lesz.

3.5. Gráfok „erőssége”

3.12. Definíció. *Legyen $\mathcal{G} := (V, E)$ egy irányítatlan összefüggő gráf és $s : E \longrightarrow \mathbb{R}_+$ egy nemnegatív súlyfüggvény. $A \subseteq E$ esetén jelölje $\kappa(A) + 1$ az A élhalmaz elhagyásával keletkező komponensek számát a gráfban. A*

$$\sigma := \min\{\frac{s(A)}{\kappa(A)} : A \subseteq E, \kappa(A) \geq 1\}$$

*számot a gráf **erősség**ének nevezzük.*

A gráf erősségének meghatározása az egyik első alkalmazása a Newton módszernek, az alább közölt eljárás Cunningham nevéhez fűződik [8].

A 3.2. fejezet alapján az

$$a(\lambda) := \min\{s(A) - \lambda\kappa(A) : A \subseteq E\}$$

értéket és az ezt minimalizáló A_λ halmazt kell legfeljebb $|V|$ darab λ-ra kiszámítani, hiszen $\kappa(A)$ egy 0 és $|V| - 1$ közé eső egész szám. Tekintsük a $\mathcal{G}$ gráf grafikus matroidját és legyen $rk : \mathcal{P}(E) \longrightarrow \mathbb{Z}$ a rangfüggvény. Ekkor könnyen láthatóan minden $A \subseteq E$-re

$$\kappa(A) = rk(E) - rk(E - A),$$

így a feladatunk a

$$\min\{\frac{1}{\lambda}s(A) + rk(E - A) : A \subseteq E\}$$

megkeresésére redukálódik, ami az I.24. tétel alapján az $\frac{1}{\lambda}s$ vektor egy $\mathcal{P}(rk)$-bázisának keresését jelenti. ☺

Vizsgáljuk most a megfelelő inverz problémát. A $\mathcal{G} = (V, E)$ gráf és az $s : E \longrightarrow \mathbb{R}_+$ súlyfüggvény mellett legyen adott egy $c : E \longrightarrow \mathbb{R}_+$ költségfüggvény és egy $\sigma_0 > 0$ szám és keresünk egy olyan $z \in \mathbb{R}_+^E$ vektort, hogy az $s + z$ súlyozással a gráf erőssége legalább σ_0, ezen belül pedig cz minimális legyen. s-et (és z-t) σ_0-lal elosztva feltehetjük, hogy $\sigma_0 = 1$.

Így egy $z \geq 0$ pontosan akkor lesz megengedett megoldás, ha minden $A \subseteq E$-re $(s + z)(A) \geq \kappa(A)$ teljesül. Mivel $\kappa(A) = rk(E) - rk(E - A)$, ezért ez ekvivales azzal, hogy minden $A \subseteq E$-re $(s + z)(A) + rk(E - A) \geq rk(E)$.

Vegyük észre, ha létezik egy $y \in \mathcal{P}(rk)$, amire $y \leq s + z$ és $y(E) = rk(E)$, akkor minden $A \subseteq E$-re

$$(s + z)(A) \geq y(A) = y(E) - y(E - A) \geq rk(E) - rk(E - A),$$

azaz z megengedett megoldás. Megfordítva, ha $(s+z)(A) + rk(E-A) \geq$ $\geq rk(E)$ igaz minden $A \subseteq E$-re, akkor az I.24. tétel alapján

$$\max\{y(E) : y \leq s+z, y \in \mathcal{P}(rk)\} =$$
$$\min\{(s+z)(A) + rk(E-A) : A \subseteq E\} \geq rk(E),$$

azaz a következő állítást kaptuk.

3.13. Állítás. *Egy $z \geq 0$ vektor pontosan akkor megengedett megoldása a fenti inverz feladatnak, ha létezik egy $y \in \mathcal{P}(rk)$, amire $y \leq s+z$ és $y(E) = rk(E)$.*

Ez az y persze $\mathcal{P}(rk)$-bázisa lesz $s+z$-nek. Ha most z optimális megoldás, akkor — mivel $c \geq 0$ — feltehető, hogy $s_j + z_j = y_j$, ha $z_j > 0$. Azaz y előáll $y = w + z$ alakban, ahol w az s egy $\mathcal{P}(rk)$ bázisa. Az eljárásunk ezek alapján a következő. Megkeressük az s egy tetszőleges w $\mathcal{P}(rk)$-bázisát, ezután a $\mathcal{P}_w := \{z \geq 0 : w + z \in \mathcal{P}(rk)\}$ halmazban keresünk egy olyan elemet, amire $y(E)$ maximális és ezen belül minimális költségű. Az I.26. tétel alapján $\mathcal{P}_w$ független a w választásától, így az algoritmus helyesen működik, továbbá ez egy polimatroid, tehát a minimális költségű maximális $\mathcal{P}_w$-beli z vektort (azaz az 1 vektor minimális költségű $\mathcal{P}_w$-bázisát) a súlyozott mohó algoritmussal hatékonyan meg tudjuk keresni.

Megjegyzés.A feladatot tetszőleges matroidra általánosíthatjuk, ha κ-t a matroid ko-rang függvényének definiáljuk. Az algoritmus ugyanúgy működik ebben az általánosabb környezetben is (lásd a 4.1. fejezetet).

Megjegyzés. Hasonlóképpen ugyanez az eljárás alkalmazható, ha nem $\frac{s(A)}{\kappa(A)}$-t, hanem $\frac{s(A)+\alpha}{\kappa(A)+\beta}$-t akarjuk minimalizálni valamilyen α és β konstansokra. Különösen egyszerű a helyzet az $\alpha = 0$, $\beta = 1$ esetben, ekkor a minimum éppen a minimális vágásérték fele lesz.

3.6. Ha van 100 Forintunk...

Az előbbiekben több olyan problémát tárgyaltunk, ahol egy a gráf adataiból számolható mérőszámot kívántunk egy adott értékre növelni úgy, hogy a változtatás költsége minimális legyen. Feladatunk lehet azonban az is, hogy van F forintunk és ebből a pénzből szeretnénk minél jobban megnövelni az illető mérőszámot. Sokszor nem triviális a két típusú feladat közötti ekvivalencia. Szerencsére a 2.2. fejezet szeparációs algoritmusa megint a segítségünkre siet.

Nézzük például az $s-t$ folyamok esetét. Jelölje $f(a)$ annak a költségét, hogy a maximális folyam értékét a-ra növeljük. Legyen $\mathcal{P} := \{a \in \mathbb{R} : f(a) \leq F\}$.

Erre adható lineáris szeparációs algoritmus: egy adott a_0-ra kiszámítjuk $f(a_0)$-t, ha ez legfeljebb F, akkor elfogadunk, ha nagyobb, akkor elutasítunk, és az „$a < a_0$"-t adjuk vissza, mint szeparáló irányt. Ez lineáris lesz, így $A := \max\{a : a \in \mathcal{P}\}$-t ki tudjuk számolni, ez pedig éppen a keresett érték.

A konkrét algoritmus a következőképpen működik. A 3.4. fejezet alapján van egy algoritmusunk, ami meghatározza $f(a)$-t. Ez úgy működik, hogy létrehoz egy $\mathcal{G}' := (V, E')$ segédgráfot megfelelő költség és kapacitásértékekkel és azon keres egy minimális költségű a értékű x folyamot. Tegyük fel, hogy ez úgy működik, hogy először keres egy a értékű folyamot (például a legrövidebb javító utak módszerével), majd ezt javítjuk (a minimális átlagú javító körökkel [11]). Ez az persze nem lineáris a-ban, mert az eredmény, az a értékű folyam nem lehet független a-tól, ha a lehetséges a-k halmaza nem egy pont, hanem egy intervallum. Viszont ha úgy módosítjuk az algoritmust, hogy (mint a legrövidebb utak esetén) az a értéket szimbolikusan kezeli (azaz az eredményül kapott folyamot $x + ax'$ alakban kapjuk, ahol $x, x' \in \mathbb{R}_+^{E'}$), akkor már valóban megvalósítható úgy az algoritmus, hogy a-val csak lineáris összehasonlítást végezzen (és semmi mást).

Most kezdjük el futtatni ezt az algoritmust anélkül, hogy a-t valóban megadnánk. Ha az algoritmus megkérdezi, hogy egy adott a_0-nál nagyobb-e a vagy kisebb, akkor számítsuk ki $f(a_0)$-t. Ha ez nem nagyobb, mint F, akkor válaszoljuk azt, hogy $a \geq a_0$, különben pedig azt, hogy $a < a_0$. Miután az algoritmus lefutott, a kérdéseire általunk adott válaszok alapján lehetséges a-k egy I intervallumot adnak. Az algoritmus ad egy x és egy x' vektort, hogy minden $a \in I$-re $x + ax'$ egy a értékű minimális költségű folyam a segédgráfban, így ennek értéke pontosan a eredeti feladatban a folyamérték a-ra növelésének a költsége. A kérdésekre a válaszokat úgy adtuk, hogy a lehetséges a-k halmaza (az algoritmus lefutása után az I) mindig tartalmazza az általunk keresett számot (azaz a legnagyobb olyan a értéket, amire az eredeti feladatunkban a folyamértéket F forintból a-ra tudjuk növelni) továbbá $I \subseteq \mathcal{P}$, vagyis minden $a \in I$ esetén a maximális folyamérték a-ra növelésének költsége nem nagyobb, mint F. Így a keresett szám $\max\{a : a \in I\}$ lesz.

E módszer előnye, hogy egy általános módszert ad az ilyen problémák megoldására. Hasonló kérdés tehető fel például, ha nem a legnagyobb $s - t$ folyam értékét, hanem a legrövidebb $s \rightsquigarrow t$ út hosszát akarjuk F forintból minél jobban megnövelni. Az előbbi eljárás erre az esetre is alkalmazható. Megjegyezzük, hogy erre feladatra Fulkerson és Harding [13, 14] adott először hatékony eljárást. Az ő algoritmusuk teljesen más elvek alapján működik.

4. Inverz feladat matroidokon

Feladatunk most a következő: Adott egy $M = (E, \mathcal{I}, w)$ súlyozott matroid, egy $c : E \longrightarrow \mathbb{R}$ költségfüggvény és egy $B \in \mathbb{R}$ szám. Keresünk egy olyan $z \in \mathbb{R}^E$-t, amire $cz \leq B$ és $\min\{(w + z)(B) : B \in \mathcal{B}_M\}$ maximális. Vagyis adott egy c költségfüggvény, ami megadja, hogy a elem súlyának egy egységgel való növelése mennyibe kerül és B forintból szeretnénk a minimális súlyú bázis súlyát minél nagyobbá tenni.

Erre a feladatra Frederickson és Solis-Oba adott hatékony algoritmust, először speciálisan grafikus matroidokra (azaz a minimális súlyú feszítőfa súlyának növelésére) [16]-ban, majd ezt terjesztették ki teljesen általános matroidra [17, 18]. Ezt ismertetjük az alábbiakban.

A feladatot először egy valamilyen értelemben homogénebb minimalizálási feladatra vezetjük vissza, amit a Newton módszer segítségével oldunk meg a W. H. Cunningham [6] és H. Narayannan [7] által kifejlesztett algoritmus felhasználásával.

Kezdjük néhány definícióval:

4.1. Definíció. *Legyen $M = (E, \mathcal{I}, w)$ egy súlyozott matroid és rögzítsünk le egy $c : E \longrightarrow \mathbb{R}_+$ költségfüggvényt. Ekkor*

$$cov_M(S) := \min\{|S \cap I| : I \in \mathcal{B}_M^{min}\}^1.$$

Ez nem más, mint a minimális súlyú bázisok által definiált matroid ko-rang függvénye.
*Azt mondjuk, hogy egy S halmazt α-val **emelünk**, ha elemeinek súlyát egységesen α-val megnöveljük.*

$$tol_M(S) := \min\{\alpha \in \mathbb{R}_+ : S \text{ elemeit } \alpha\text{-val emelve } cov_M(S) \text{ csökken}\}^1$$

és végül legyen

$$rate_M(S) := \frac{c(S)}{cov_M(S)}.$$

Ha egy S halmazon való emeléssel a minimális bázis súlyát egy $\delta < tol_M(S)$ értékkel növeljük, akkor $\alpha := \frac{\delta}{cov_M(S)}$-sel kell emelnünk és ennek költsége $c(S)\frac{\delta}{cov_M(S)} = rate_M(S) \cdot \delta$. Ez alapján megfogalmazható egyfajta „mohó" eljárás az optimális emelés megtalálásához, miszerint mindig a $rate_M(S)$-et

[1] A *coverage* és a *tolerance* angol szavak alapján.

minimalizáló S-et kell emelni $tol_M(S)$-sel, amíg a pénzünk el nem fogy. A 4.1. fejezetben algoritmust adunk a $rate_M(S)$-et minimalizáló halmaz keresésére, utána pedig belátjuk, hogy az előbbi eljárás jó, azaz valóban az optimális emeléshez jutottunk és gyors, vagyis az előbbi emelést nem kell túl sokszor iterálni.

4.1. Minimális rátájú halmaz keresése

Rögzítsünk egy $M = (E, \mathcal{I}, w)$ súlyozott matroidot és egy $c : E \longrightarrow \mathbb{R}_+$ költségfüggvényt. Legyen $w_1, w_2, \ldots, w_p$ a w értékkészlete. $S \subseteq E$-re legyen $S_{<w_i} := \{x \in S : w(x) < w_i\}$. Az $S_{=w_i}, S_{\leq w_i}$ és $S_{>w_i}$ definíciója hasonló.

Most legyen $M_i := (M|E_{\leq w_i})/E_{<w_i}$. Bebizonyítjuk, hogy a $rate_{M_i}$-t minimalizáló S_i-k közül legalább az egyik minimalizálja $rate_M$-et.

4.2. Lemma. *Ha B egy minimális súlyú bázis M-ben és valamely w_i-re $B_{=w_i} \neq \emptyset$, akkor $B_{=w_i}$ minimális súlyú bázisa az $M_i := (M|E_{\leq w_i})/E_{<w_i}$ matroidnak.*

Bizonyítás.$B_{<w_i}$ bázisa $(M|E_{\leq w_i})|E_{<w_i}$-nek, így az összehúzás definíciója miatt $B_{=w_i}$ bázisa $(M|E_{\leq w_i})/E_{<w_i}$-nek. M_i homogenitása miatt persze minimális súlyú is lesz.

4.3. Lemma. *Legyen B minimális súlyú bázis M-ben, w_i olyan, hogy $B_{=w_i} \neq \emptyset$ és A egy minimális súlyú bázisa M_i-nek. Ekkor $B_{<w_i} \cup A \cup B_{>w_i}$ is minimális súlyú bázis M-ben.*

Bizonyítás. Egészítsük ki A-t az $M|E_{\leq w_i}$ egy A' bázisává. Ekkor minden $y \in B_{=w_i} - A$-hoz van $x \in A' - B_{\leq w_i}$, amire $S = B_{\leq w_i} - y + x$ bázisa $M|E_{\leq w_i}$-nek. Mivel $B_{\leq w_i}$ minimális súlyú bázisa $M|E_{\leq w_i}$-nek, ezért $w(x) \geq w(y)$, így $x \in A$ és S szintén minimális súlyú bázis. Ezt ismételve $B_{=w_i} - A$ elemeire az állítást bebizonyítottuk.

4.4. Lemma. *Ha $S \subseteq E$ minimalizálja $rate_M$-et és w_i olyan, hogy $S_{=w_i} \neq \emptyset$, akkor*

(i) $S_{=w_i}$ minimalizálja $rate_M$-et és

(ii) Minden $rate_{M_i}$-t minimalizáló halmaz minimalizálja $rate_M$-et is.

Bizonyítás. Minden $j = 1, 2, \ldots, p$-re legyen B^j egy olyan minimális súlyú bázisa M-nek, amire $cov_M(S_{=w_j}) = |B^j \cap S_{=w_j}|$. A 4.3. lemma alapján $B := \cup_{j=1}^p B^j_{=w_j}$ minimális súlyú bázisa M-nek. Így $cov_M(S) = |S \cap B| = \sum_{j=1}^{p} cov_M(S_{=w_j})$, amiből — mivel S minimalizálja $rate_M$-et —

$$rate_M(S) = \frac{\sum_{j=1}^{p} c(S_{=w_j})}{\sum_{j=1}^{p} cov_M(S_{=w_j})} = \frac{c(S_{=w_i})}{cov_M(S_{=w_i})} = rate_M(S_{=w_i})$$

adódik minden i-re, amire $S_{=w_i} \neq \emptyset$, amivel (i) bizonyítást nyert.

A 4.2. és 4.3. lemmák alapján M_i minimális súlyú bázisai éppen az $A_{=w_i}$-k, ahol A az M minimális súlyú bázisa. Így minden $T \subseteq E_{=w_i}$-re $cov_{M_i}(T) = cov_M(T)$ és $rate_{M_i}(T) = rate_M(T)$, amiből (ii) következik. ☺

A 4.4. lemma alapján a $rate_M$-et minimalizáló S-et megtalálhatjuk úgy, hogy minden M_i-ben keresünk egy $rate_{M_i}$-t minimalizáló S_i halmazt és ezek közül kiválasztuk azt, amelyre $rate_M(S_i)$ minimális.

Mivel M_i-ben valamennyi elem súlya egyforma, így most már csak egy azonosan 1 súlyozású matroidban kell tudnunk egy $rate_{M_i}$-t minimalizáló halmazt találnunk. Ehhez a 3.2. fejezetben tárgyalt Newton-módszert alkalmazzuk.

Megjegyzés. Ha M grafikus matroid, akkor M_i is az, és ekkor $rate_{M_i}$ minimalizálása éppen a megfelelő gráf 3.5. fejezetben definiált erősségének kiszámítását igényli a c súlyfüggvényre nézve. Az itt közölt algoritmus az erősséget számító eljárás egyszerű kiterjesztése.

Vizsgáljuk tehát az $M_i := (E_i, \mathcal{I}_i)$ súlyozatlan matroidot. Jelölje n_i az M_i rangját és legyen $m_i := |E_i|$. Legyen $\sigma_i := \min_{S \subseteq E_i} rate_{M_i}(S)$ és $\lambda \geq 0$-ra $g_i(\lambda) := \min\{c(T) - \lambda cov_{M_i}(T) : T \subseteq E_i\}$. Ekkor $g_i(\lambda) \leq 0$ és $\sigma_i = \max\{\lambda : g_i(\lambda) = 0\}$. Könnyen látható, hogy g_i egy nem növő, konkáv, szakaszonként lineáris függvény, továbbá minden lineáris szakasz meredeksége megegyezik $cov_{M_i}(S)$-sel valamely $S \subseteq E_i$-re és így legfeljebb n_i töréspontja lehet.

A fentiek alapján a Newton módszer segítségel ki tudjuk számolni σ_i-t. Az algoritmus a következő. Mivel $\sigma_i \leq \frac{c(E_i)}{n_i}$, ezért $\lambda = \frac{c(E_i)}{n_i}$-vel indulunk. Az algoritmus során végig $\lambda \geq \sigma_i$. Ezután kiszámítjuk $g_i(\lambda)$-t és azt a T_λ-t, ami a $c(T_\lambda) - \lambda cov_{M_i}(T_\lambda)$-t minimalizálja. Ha $g_i(\lambda) = 0$, akkor $\lambda \leq \sigma_i$, így $\lambda = \sigma_i$, tehát készen vagyunk. Ha $g_i(\lambda) < 0$, akkor az új λ legyen $\frac{c(T_\lambda)}{cov_{M_i}(T_\lambda)}$.

Ez legfeljebb n_i iteráció után végetér.
Tehát a
$$\min\left\{\frac{c(T)}{\lambda} - cov_{M_i}(T) : T \subseteq E_i\right\}$$
paraméteres problémát kell (legfeljebb n_i-szer) megoldanunk. Mivel M_i súlyozatlan, ezért minden $S \subseteq E_i$-re $cov_{M_i}(S) = rk_{M_i}(E_i) - rk_{M_i}(E_i - S, M_i)$. Így a feladatunk
$$\min\{d(T) + rk_{M_i}(E_i - T) : T \subseteq E_i\} \tag{11}$$
alakú, ahol $d(T) := \frac{c(T)}{\lambda}$. Az I.24. tétel alapján ez a feladat ekvivalens a d vektor $\mathcal{P}(M_i)$-bázisának keresésével. Erre Narayanan algoritmusát használva a következőt kaptuk:

4.5. Lemma. *Az $M_i = (E_i, \mathcal{I}_i)$ súlyozatlan matroidban a $rate_M$-et minimalizáló halmaz $O(m_i^4 n_i + m_i^3 n_i^3 \tau)$ lépésben megtalálható, ahol τ egy halmaz független voltának eldöntéséhez szükséges idő.*

4.2. Az algoritmus

Az algoritmusunk tehát a következő: Keresünk az *előző fejezet eljárásával* egy $rate_M$-et minimalizáló S halmazt, S-et emeljük Δ-val, ahol $\Delta :=$ $= \min\{tol_M(S), B/rate_M(S)\}$, majd B-t csökkentjük $\Delta * rate_M(S)$-sel. A továbbiakban ezt ismételjük, amíg B nulla nem lesz.

Annak igazolásához, hogy ez az algoritmus helyesen működik, vizsgáljunk meg egy w^* optimális B költségű emelést. Tegyük fel, hogy ezt egy $\mathcal{A}$ algoritmus találta meg. Legyen $\mathcal{A}$ olyan, ami szintén úgy működik, hogy egy lépésében egy halmazt kiválaszt és azt emeli egy bizonyos értékkel. Tekintsük ezen emeléseket folytonosnak és legyen egy $0 \leq b \leq B$ értékre $M^*(b) =$ $= (E, \mathcal{I}, w_b^*)$ a matroid abban a pillanatban, amikor az $\mathcal{A}$ algoritmus éppen b egyéget költött az emelésre és legyen S_b^* az utoljára emelt halmaz. A w^*-ot megtaláló $\mathcal{A}$ algoritmus többféleképpen működhet, ezt egyenlőre szabadon hagyjuk. Ennek megfelelően $M^*(b)$ és S_b^* is többféle lehet, csak $\mathcal{A}$ lerögzítése után lesz egyértelmű. $M(b) = (E, \mathcal{I}, w_b)$ és S_b ugyanezeket jelöli a mi algoritmusunkra.

Az algoritmusunk helyes működéséhez elég belátni, hogy $rate_{M(b)}(S_b) \leq$ $\leq rate_{M^*(b)}(S_b^*)$ minden $0 \leq b \leq B$ értékre. Pontosabban, ha van olyan $\mathcal{A}$ algoritmus, melyre ez teljesül, és a w^* optimális megoldást találja meg, akkor ez igazolja a mi algoritmusunk helyességét. Tehát most úgy fogjuk rögzíteni $\mathcal{A}$-t hogy a fenti egyenlőtlenség fennálljon.

Ehhez az $M = (E, \mathcal{I}, w)$ súlyozott matroidra jelölje $M_{\leq \alpha} := \{e \in E : w(e) \leq \leq \alpha\}$ és tekintsük a következő lemmát.

4.6. Lemma. *Legyen $M = (E, \mathcal{I}, w)$ és $M' = (E, \mathcal{I}, w')$ két súlyozott matroid (ugyanazon matroid két különböző súlyozással) és legyen $S \subseteq E$. Ha minden $e \in S$ elemre $M'_{\leq w'(e)} \subseteq M_{\leq w(e)}$, akkor $cov_M(S) \leq cov_{M'}(S)$.*

Bizonyítás. Legyen $B \in \mathcal{B}_M^{min}$ és $S' := S \cap B$. S' persze bővíthető w-minimális bázissá, így az I.17. állítás alapján minden $e \in S'$ elemre és minden C körre, amire $e \in C$ teljesül, hogy $\max_{f \in C - S'} w(f) \geq w(e)$. Mivel ekkor a lemma feltétele alapján $\max_{f \in C - S'} w'(f) \geq w'(e)$ is igaz minden C körre, amire $e \in C$, így szintén az I.17. állítás miatt S' kibővíthető w'-minimális bázissá, amiből $cov_M(S) \leq cov_{M'}(S)$ már következik.

Ez már sugallja számunkra, hogy mik lesznek a megfelelő S_b^*-ok. Azt követeljük meg, hogy $\mathcal{A}$ mindig úgy válassza meg S_b^*-t, hogy minden $e \in S_b^*$ elemére $M(b)_{\leq w_b(e)} \subseteq M^*(b)_{\leq w_b^*(e)}$ legyen. Ha ez lehetséges, akkor az előző lemma alapján

$$rate_{M(b)}(S_b) \leq rate_{M(b)}(S_b^*) \leq rate_{M^*(b)}(S_b^*),$$

ami bizonyítja az algoritmusunk helyességét.

Most már pontosan előírhatjuk, hogyan válassza ki $\mathcal{A}$ az S_b^*-ket. Legyen

$$w_b^*(e) := \min\{w^*(e), w^b(e) + \Delta_b\}$$

minden $e \in E$-re, ahol Δ_b-t úgy választjuk meg, hogy a w súlyfüggvény w_b^*-re való emelésének költsége éppen b legyen. Ekkor $w_b^*(e) \leq w^*(e)$ és így $w_B^*(e) = w^*(e)$ minden $e \in E$-re. w_b^* definíciójából adódóan legyen

$$S_b^* := \{e \in E : \forall 0 \leq \varepsilon \leq b\text{-re } w_{b-\varepsilon}^*(e) < w^*(e)\}$$

Legyen most $0 \leq b \leq B$ tetszőleges. Legyen $e_1 \in S_b^*$ és $e_2 \in E$. S_b^* és w_b^* definíciójából $w_b^*(e_1) = w_b(e_1) + \Delta_b$ és $w_b^*(e_2) \leq w_b(e_2) + \Delta_b$. Így, ha $w_b(e_2) \leq w_b(e_1)$, akkor $w_b^*(e_2) \leq w_b^*(e_1)$, azaz $M(b)_{\leq w_b(e_1)} \subseteq M^*(b)_{\leq w_b^*(e_1)}$, amint azt megkívántuk.

Szükségünk van még a $tol_M(S)$ meghatározására. Ez szerencsére nem okoz gondot, mert csak olyan S-ekre kell kiszámítani, hogy minden $e \in S$-re $w(e)$ megegyezik. Az I.16. állítás alapján egy minimális súlyú bázis az emelés során csak akkor szűnhet meg minimálisnak lenni, amikor az S-beliek súlya átlép egy másik, a matroidban létező súlyt. Ha tehát az S-beliek súlya w_i, akkor $tol_M(S) = w_j - w_i$, ahol w_j az a legkisebb w_i-nél nagyobb súly, amire emelve S-et $cov_M(S)$ csökken. (Ha ilyen nem létezik, akkor $tol_M(S) = +\infty$). Ehhez $cov_M(S)$-et kell tudnunk kiszámolni. Szintén S homogenitása miatt

$$cov_M(S) = cov_{M_i}(S) = rk_{M_i}(E_{=w_i}) - rk_{M_i}(E_{=w_i} - S).$$

Ennek kiszámításához legfeljebb $|E_{=w_i}|$ halmaz függetlenségét kell eldönteni M_i-ben. Egy $T \subseteq E_{=w_i}$ halmaz M_i-beli függetlenségének eldöntéséhez keresnünk kell egy B minimális súlyú bázisát M-nek és $(B - B_{=w_i}) \cup T$ függetlenségét kell meghatároznunk M-ben. Így $cov_{M_i}(S)$ $O(|E|\tau)$, $tol_M(S)$ pedig $O(|E|^2\tau)$ lépésben meghatározható.

Mivel az egyes halmazok emelése során új súly nem jelenik meg, így az algoritmus legfeljebb $|E|rk(M)$ iteráció után végetér. (Ha i_e az a szám, amire $w(e) = w_{i_e}$, akkor a $\sum_{e \in B} i_e$ érték minden B minimális súlyú bázisra egyenlő és minden emeléskor legalább egyel nő, továbbá nem nagyobb, mint $|E|rk(M)$.)

Összegezve az eddigieket, a következő tételt kaptuk:

4.7. Tétel. *A keresett minimális költségű emelés*

$$O(|E|^5 rk(M)^2 + |E|^4 rk(M)^4 \tau)$$

lépésben megtalálható, ahol τ az egy halmaz függetlenségét eldöntő orákulum futási ideje.

Megjegyzés. Szintén Frederickson és Solis Oba [17, 18] megmutatta, hogy a fenti algoritmus több speciális esetben leegyszerűsödik. Például egy páros gráf transzverzális matroidja esetén (azaz amikor a minimális súlyú teljes párosítás súlyát kívánjuk megnövelni) a megfelelő feladat

$$O(mn(m + n^2)|E| \log(m^2/|E| + 2))$$

lépésben megoldható, ahol E a páros gráf élhalmaza.
Partíciós matroid esetén pedig mindössze $O(|S| \log |S|)$ idő szükséges (S a matroid alaphalmaza).

5. Amit nem tudunk

Ebben a fejezetben — a teljeség igénye nélkül — felvázoljuk a témakör néhány olyan kérdését, amire nem ismerjük a választ.

- Kezdjük mindjárt az elején. A szó legszorosabb értelmében, ugyanis a dolgozatunk legelején, a bevezetésben említett problémára — amikor az s—t távolságot akarjuk csökkenteni egy súlyozott gráf két adott pontjára — sajnos nem tudunk hatékony algoritmust adni. A feladat a „szintaktikai” hasonlóság ellenére meglehetősen eltér a 3.3. fejezetben tárgyalt problémától. Annyit mindenesetre megállapíthatunk, hogy egy optimális z csökkentő vektor úgy néz ki, hogy egy $s \rightsquigarrow t$ út (amelyik a legrövidebb út lesz) kivételével mindenütt 0, ezen az úton pedig z sorban a legolcsóbb élek hosszát csökkenti (egy él kivételével nullára) úgy, hogy ezen út hossza éppen L legyen. A legjobb ilyen utat viszont nem tudjuk megtalálni.

- A 2.2. fejezet tétele egy nagyon hatékony eszköz a kezünkben, de az általa adott algoritmusok struktúrája meglehetősen áttekinthetetlen. Sokszor érdekes lenne az ilyen módon megoldott problémákra közvetlen megoldást találni. Tekintsük például a 2.3. fejezetben tárgyalt problémát. Már az sem nyilvánvaló, hogyan lehet véges futásidejű (tehát nem közelítő) algoritmust adni a λ pontos értékének meghatározására, bár ez még megoldható egy a Newton eljáráshoz hasonló módszerrel: Legyártunk egy $p_0, p_1, \ldots$ és egy $q_0, q_1, \ldots$ $a \rightsquigarrow b$ útsorozatot úgy, hogy minden p_i-re $t(p_i) \geq T$ és minden q_i-re $t(q_i) \leq T$ teljesül valamint egy $\lambda_1, \lambda_2, \ldots$ számsorozatot a következőképpen. Legyen p_0 egy c-minimális és q_0 egy t-minimális út. Ha egy i-re $t(p_i) = t(q_i)$, akkor $t(p_i) = T$, így p_i egy optimális megoldás, tehát készen vagyunk. Ha nem, akkor legyen $\lambda_{i+1} := \frac{c(q)-c(p)}{t(p)-t(q)}$. Ha $f(\lambda_{i+1}) = c(p_i) + \lambda_{i+1}t(p_i) - \lambda_{i+1}$ akkor λ_{i+1} optimális, tehát megint készen vagyunk. Ha nem, és az $f(\lambda_{i+1})$-et minimalizáló p^* útra $t(p^*) \geq T$, akkor legyen $p_{i+1} := p^*$ és $q_{i+1} := q_i$, ha pedig $t(p^*) \leq T$, akkor legyen $p_{i+1} := p_i$ és $q_{i+1} := p^*$. Könnyen látható, hogy ez az algoritmus exponenciális időben megtalálja az optimális λ-t. Azt, hogy vajon ez az eljárás erősen polinomiális futásidejű-e, egyelőre nem tudjuk sem cáfolni sem bizonyítani.

- Hasonlóképpen nem tudjuk a választ arra, hogy a 3.1.3. pontban adott problémára a Newton módszer (azaz amikor a minimális átlagú körséta negatív átlagával emelünk ameddig van negatív kör) erősen polinomiális algoritmust ad-e?

- Természetesen adódó kérdés a 3.1.4. pontban vizsgált probléma irányítatlan esetben. Erre sem ismerünk gyors algoritmust. (Láthatóan a konzervatív súlyozás sokkal nehezebben kezelhető irányítatlan, mint irányított esetben. A negatív kör létezésének tesztelése, a minimális körátlag kiszámítása is lényegesen bonyolultabb irányítatlan gráfokban és egy irányítatlan gráfban a súlyozás konzervatívságának Sebő-féle karakterizációja is sokkal körülményesebb a megengedett potenciál létezésével való karakterizációnál.)

I. Jelölések és alapvető tételek

E fejezetben ismertetjük az általunk használt jelöléseket és alapvető tételeket. Fő céljunk rövid áttekintést adni a felhasznált és ismertnek feltételezett témakörökről, így az állításokat általában bizonyítás nélkül közöljük, legyen az egy trivialitás, vagy akár komoly tétel. A jelölések terén kisebb-nagyobb eltéréssel igyekszünk az általános konvenciókat követni.

I.1. Definíció. *Az egyesítés és kivonás jelölésére az $\cup$ és $\setminus$ mellett használjuk a $+$ és $-$ jeleket, kicsit tágabb értelemben. Például $A + B$ jelentheti $A \cup B$ és $A \cup \{B\}$ bármelyikét (főleg az utóbbit). Ez félreértesre nem ad alkalmat. $\mathcal{P}(A)$ az A halmaz hatványhalmaza.*

I.2. Definíció. *A valós számhalmazt $\mathbb{R}$-rel jelöljük, $\mathbb{R}_+$, a nemnegatív számok halmaza. Az $\mathbb{R}^E$ vektortéren az $E \longrightarrow \mathbb{R}$ függvények halmazát értjük, és vegyesen használjuk egy $x \in \mathbb{R}^E$ elem vektor és függvény tulajdonságát. Az egyszerűség kedvéért használjuk az $x_e := x(e)$ jelölést is. A 0-t és az 1-et használjuk az azonosan 0 illetve azonosan 1 vektor jelölésére is. Ha $x, y \in \mathbb{R}^E$, akkor $x \geq y$ azt jelenti, hogy minden $e \in E$-re $x_e \geq y_e$. $R_+^E := \{x \in \mathbb{R}^E : x \geq \geq 0\}$. Egy $A \subseteq E$ halmazra $\chi_A \in \mathbb{R}^E$ jelöli az A halmaz karakterisztikus vektorát. $x \in \mathbb{R}^E$ esetén $x(A) := \chi_A \cdot x = \sum_{a \in A} x(a)$.*

Egy vektor értelem szerint lehet sor- vagy oszlopvektor, a traszponálás jelét vektorok esetén nem írjuk ki.

Sokszor használjuk (és ismertnek tételezzük fel) a lineáris programozás fogalmait, jelöléseit; többször alkalmazzuk *Farkas Gyula* lemmáját és a dualitástételt.

I.3. Definíció. *Ha $\mathcal{G} = (V, E)$ egy irányított gráf és $A \subseteq E$, akkor $\varrho(A)$ illetve $\delta(A)$ jelöli az A halmazba be-, illetve az onnan kilépő élek halmazát. $e \in E$ esetén az $e = (i, j)$ jelölés azt jelenti, hogy az e él „feje" a j, a „farka" pedig az i pont. Irányítatlan gráf esetén $\delta(A)$ jelöli azon élek halmazát, amelynek pontosan az egyik vége van A-ban. Egy $p = (v_0, v_1, \ldots, v_k)$ pontsorozat* ***séta****, ha minden $i \in \{1, \ldots, k\}$-ra (v_{i-1}, v_i) az (irányított vagy irányítatlan) gráf egy éle és* ***körséta****, ha $v_0 = v_k$. Egy séta vagy körséta* ***hosszán*** *a k számot (a benne levő élek számát) értjük. Egy séta* ***út****, ha valamennyi pontja különböző. Hasonlóan, egy körséta* ***kör****, ha legalább három hosszú és csak az első és az utolsó pontja egyezik meg. $\mathcal{P}_{a,b}$-vel jelöljük az a-ból b-be vezető utak halmazát.*

Ismertnek tételezzük fel az alapvető algoritmuselméleti fogalmakat és eljárásokat, mint például az $\mathcal{NP}$ osztály, az $\mathcal{NP}$-teljesség vagy a dinamikus programozás fogalmát. Szintén hivatkozás nélkül felhasználjuk gráfokkal kapcsolatos alapvető algoritmusokat, mint például a szélességi-, a mélységi keresést vagy a legrövidebb utakat találó eljárásokat. Használjuk az áram, a folyam, a pontenciál fogalmát, az ezekkel kapcsolatos tételeket, és az ezeket találó algoritmusokat.

Sokszor lesz segítségünkre a matroidelmélet és a polimatroidok elmélete, a következőkben erről adunk rövid áttekintést.

I.4. Definíció. *Egy $M = (E, \mathcal{I})$ halmazpárt **matroid**nak nevezünk, ha $\mathcal{I} \subseteq \mathcal{P}(E)$ és*

(i) $\emptyset \in \mathcal{I}$,

(ii) ha $X \subseteq Y \in \mathcal{I}$, akkor $X \in \mathcal{I}$ és

(iii) minden $X \subseteq E$ részhalmazra az $\mathcal{I}$ tartalmazásra nézve maximális X-ben fekvő tagjai azonos elemszámúak.

*Az $\mathcal{I}$ elemeit **független halmaz**oknak nevezzük. Az $\mathcal{I}$ maximális elemszámú tagjai az M **bázisa**i. Az M bázisainak halmazát $\mathcal{B}_M$-mel vagy $\mathcal{B}$-vel jelöljük.*

Megjegyzés. Az $\mathcal{I}$ mérete lehet exponenciális $|E|$-ben, így $\mathcal{I}$-t nem lehet hatékonyan explicite megadni. Szerencsére a matroid struktúrája miatt általában erre nincs is szükség. Egy matroidot adottnak tekintünk, ha van egy ***orákulum***, amely E-nek egy részhalmazáról eldönti, hogy független-e vagy sem.

Talán legalapvetőbb példa a matroidokra a következő. Legyen $\mathcal{G} := (V, E)$ egy irányítatlan gráf. Ekkor $M_{\mathcal{G}} := (E, \{A \subseteq E : A \text{ nem tartalmaz kört}\})$ egy matroid, amit a $\mathcal{G}$ ***grafikus matroid***jának nevezünk.

Fontos tulajdonsága a bázisoknak a

I.5. Tétel. (Kölcsönös kicserélési tulajdonság) *Ha $A, B \in \mathcal{B}$ és $x \in A - B$, akkor létezik $y \in B - A$, hogy $A - x + y \in \mathcal{B}$ és $B - y + x \in \mathcal{B}$.*

Megjegyzés. Ez a tulajdonság azzal együtt, hogy minden bázis elemszáma egyenlő (és hogy $\mathcal{B} \neq \emptyset$) karakterizálja is a bázisokat abban az értelemben, hogy ha egy nemüres halmazrendszer rendelkezik ezzel a tulajdonsággal, akkor az egy alkalmas matroid bázisát alkotja.

I.6. Definíció. *$rk_M(S) := \max\{|I| : I \subseteq S, I \in \mathcal{I}\} = \max\{|S \cap B| : B \in \mathcal{B}_M\}$ az $S \subseteq E$ halmaz **rang**ja. $rk(M) := rk_M(E)$ a **matroid rangja**. A $t_M(S) := \min\{|S \cap B| : B \in \mathcal{B}_M\}$ szám az $S \subseteq E$ halmaz **ko-rang**ja.*

I.7. Állítás. *$t_M(S) := rk_M(E) - rk_M(E - S)$.*

I.8. Definíció. *Ha $M = (E, \mathcal{I})$ egy matroid, akkor $\mathcal{B}^* := \{E - B : B \in \mathcal{B}_M\}$ egy matroid bázisát alkotja. Ezt az M **duális matroid**jának nevezzük és M^*-gal jelöljük.*

I.9. Állítás. *Minden $S \subseteq E$ halmazra $t_M(S) + rk_{M^*}(S) = |S|$.*

I.10. Definíció. *Ha $M = (E, \mathcal{I})$ egy matroid és $S \subseteq E$, akkor*

$$M|S := (S, \{I : I \in \mathcal{I} \text{ és } I \subseteq S\}) \text{ és}$$
$$M \backslash S := (E - T, \{I : I \subseteq E - T \text{ és } S \cup Y \in \mathcal{I}\}),$$

*ahol Y az $M|S$ egy bázisa. (A kapott matroid nem függ Y választásától) $M|S$ neve az M **megszorítása** S-re, $M \backslash S$ pedig az S **összehúzásával** keletkező matroid.*

I.11. Definíció. *Legyen $M = (E, \mathcal{I})$ egy matroid. Egy $C \subseteq E$ halmaz **vágás**, ha minden bázisba belemetsz. Egy $K \subseteq E$ halmaz **kör**, ha minimális összefüggő, azaz nem független, de bármely elemét elhagyva már független.*

I.12. Állítás. *Ha $B \in \mathcal{B}$ bázis, akkor minden $e \notin \mathcal{B}$ elemre $B + e$-ben pontosan egy kör van. Ezt az e elem B-hez tartozó **alapkör**ének nevezzük, és $C(B, e)$-vel jelöljük.*

I.13. Definíció. *Egy $M = (E, \mathcal{I}, w)$ hármast **súlyozott matroid**nak nevezünk, ha $(E, \mathcal{I})$ matroid és $w : E \longrightarrow \mathbb{R}$ függvény. Az M-nek a w-re nézve minimális súlyú bázisait $\mathcal{B}_M^{min}$-nel, $\mathcal{B}_w^{min}$-nel vagy $\mathcal{B}^{min}$-nel jelöljük.*

I.14. Állítás. *$\mathcal{B}_M^{min}$ teljesíti a (kölcsönös) kicserélési axiómát, így egy matroid bázisát alkotja.*

A következő algoritmus a jólismert minimális súlyú feszítőfát találó algoritmus általánosítása.

I.15. Állítás. *A súlyozott matroid egy minimális súlyú bázisát adja a következő* ***mohó algoritmus****. Legyen először $A = \emptyset$ és $E = \{e_1, \ldots, e_n\}$, úgy, hogy $i < j$ esetén $w(i) \leq w(j)$. Vegyük e sorrend szerint az E elemeit, és vegyük hozzá A-hoz, ha A ezután is független marad. Az eljárás végén kapott A lesz a keresett bázis.*

Dolgozatunkban felhasználjuk a következő két minimalitási kritériumot.

I.16. Állítás. *Egy $B \in \mathcal{B}$ bázis pontosan akkor minimális súlyú, ha minden $e \in E - B$ és minden $f \in C(C, e)$ elemre $w(e) \geq w(f)$.* ☺

I.17. Állítás. *Egy $I \in \mathcal{I}$ halmaz pontosan akkor bővíthető minimális súlyú bázissá, ha minden C körre és minden $f \in I \cap C$ elemre $\max\limits_{e \in C - I} w(e) \geq w(f)$.* ☺

I.18. Definíció. *Egy M matroid független elemei incidenciavektorának konvex burkát az M* ***matroid-poliéderének*** *nevezzük és $\mathcal{P}(M)$-mel jelöljük.*

I.19. Állítás. $\mathcal{P}(M) = \{z \in \mathbb{R}_+^E \mid \forall S \subseteq E\text{-re} \sum\limits_{e \in S} z(e) \leq rk(S, M)\}$ ☺

I.20. Definíció. *Egy $\mathcal{P} \subseteq \mathbb{R}_+^E$ kompakt halmazt* ***polimatroid****nak nevezünk, ha*

(i) $0 \in \mathcal{P}$,

(ii) *Ha $y \leq x \in \mathcal{P}$, akkor $y \in \mathcal{P}$ és*

(iii) *Minden rögzített $x \in \mathbb{R}_+^E$-re minden maximális $y \in \mathcal{P}$, $y \leq x$-re $y(E)$ független y választásától.*

A (iii)-beli y-t az x ***$\mathcal{P}$-bázisá****nak nevezzük.*

A polimatroid fogalmát Edmonds [15] vezette be és a témakör sok mély és alapvető tétele is tőle származik. Bebizonyította például, hogy egy $\mathcal{P}$ polimatroidban egy vektor $\mathcal{P}$-bázisa mohó módon megtalálható, még súlyozott esetben is.

I.21. Tétel. (Edmonds) *Minden $c \in \mathbb{R}^E$-re egy $x \in \mathbb{R}^E_+$ vektornak a cy-t minimalizáló $\mathcal{P}$-bázisa megkapható a következő módon. Legyen $E = \{e_1, \dots, e_m\}$ az E egy felsorolása, amire $i < j$ esetén $c(e_i) \leq c(e_j)$. Először legyen $y = 0$, ezután végigmegyünk E elemein a fenti sorrendben és a megfelelő komponenst megnöveljük amennyire lehet, úgy, hogy $y \leq x$ és $x \in \mathcal{P}$ továbbra is teljesüljön. Azaz $y \leftarrow y + \min(x(e_i), max\{\varepsilon : y + \varepsilon\chi_{\{e_i\}} \in \mathcal{P}\})\chi\{e_i\}$-t számítjuk ki $\forall i \in \{1, \dots, n\}$-re.*

I.22. Definíció. $f : \mathcal{P}(E) \longrightarrow \mathbb{R}$ ***polimatroid függvény***, ha normalizált, *monoton és szubmoduláris, azaz*

(i) $f(\emptyset) = 0$,

(ii) $f(A) \leq f(B)$, *ha* $A \subseteq B \subseteq E$ *és*

(iii) $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$, *ha* $A, B \subseteq E$.

I.23. Definíció. $\mathcal{P}(f) := \{x \in \mathbb{R}^E_+ : x(A) \leq f(A)\ \forall A \subseteq E\text{-re}\}$ *az f* ***által generált polimatroid****.*

I.24. Tétel. *Legyen f egy polimatroid függvény az E részhalmazain. Ekkor $P(f)$ egy polimatroid és egy $x \in \mathbb{R}^E_+$-nek minden y $P(f)$-bázisára*

$$y(E) = \min\{x(A) + f(E - X) : A \subseteq E\}. \tag{12}$$

Bizonyítás.$P(f)$ polimatroid volta következik (12)-ből, így elegendő azt igazolni.

$$y(E) = y(A) + y(E - A) \leq x(A) + f(E - A)$$

alapján elegendő egy A-t találni, ahol egyenlőség áll fenn. Egy y-ra nézve egy $B \subseteq E$ halmazt nevezzünk *szoros*nak, ha $y(B) = f(B)$. Ha B és C két szoros halmaz, akkor

$$\begin{aligned} f(B) + f(C) = y(B) + y(C) &= y(B \cap C) + y(B \cup C) \\ &\leq f(B \cap C) + f(B \cup C) \leq f(B) + f(C), \end{aligned}$$

ami csak úgy lehet, ha mindenütt egyenlőség áll, amiből következik, hogy $B \cap C$ és $B \cup C$ szintén szoros halmaz. Mivel y bázis, ezért minden $j \in E$-re vagy $y_j = x_j$ teljesül vagy j eleme egy szoros halmaznak. Válasszuk most $E - A$-t az összes szoros halmaz egyesítésének.

$$y(E) = y(A) + y(E - A) = x(A) + f(E - A)$$

alapján ez az A megfelelő.

I.25. Következmény. *Minden $y \in P(f)$-re és $j \in E$-re*

$$\max\{\varepsilon : y + \varepsilon\chi_{\{j\}} \in P(f)\} = \min\{f(A) - y(A) : j \in A \subseteq E\}.$$

A I.21. tétel alkalmazásához lényeges a $\max\{\varepsilon : y + \varepsilon\chi_{\{j\}} \in P(f)\}$ meghatározása $\mathcal{P}(f)$-ben. Cunningham [8]-ban megmutatta, hogy ha f egy grafikus matroid rangfüggvénye, akkor ez egy folyamfeladatra vezethető vissza. Szintén Cunningham [6] adott először tetszőleges matroid esetén hatékony algoritmust egy x vektor $\mathcal{P}(M)$-bázisának megtalálására. Maga az algoritmus a matroid-partíció feladat és a folyamprobléma egy érdekes kombinációján alapul. A legjobb ismert algoritmus egy vektor $\mathcal{P}(M)$-bázisának megtalálására Narayanantól származik [7]. Ennen futási ideje $O(|E|^4 + |E|^3 rk(M)^2\tau)$, ahol τ a függetlenségi orákulum futási ideje.

A következő tétel a matroid összehúzás általánosítása.

I.26. Tétel. *Legyen f egy polimatroid függvény az E részhalmazain, $s \in \mathbb{R}_+^E$ és w az s egy $\mathcal{P}(f)$-bázisa. Ekkor $\mathcal{P}_w := \{z \in \mathbb{R}_+^E : w + z \in \mathcal{P}(f)\}$ egy polimatroid és független a w választásától.*

Bizonyítás. Legyen $x \in \mathbb{R}_+^E$ tetszőleges és legyen y egy maximális olyan vektor, amire $y \leq x$ és $y \in \mathcal{P}_w$. Belátjuk, hogy ekkor $y + w$ az $x + s$ $\mathcal{P}(f)$-bázisa. Indirekt tegyük fel, hogy nem. Ekkor van egy $j \in E$ elem és egy $\varepsilon > 0$ szám, amire $y + w + \varepsilon\chi_{\{j\}} \in \mathcal{P}(f)$ és $y_j + w_j + \varepsilon \leq x_j + s_j$. Így $y + \varepsilon\chi_{\{j\}} \in \mathcal{P}_w$, amiből y maximalitása miatt $y_j = x_j$ következik. Ebből viszont $w_j + \varepsilon \leq s_j$ és $w + \varepsilon\chi_{\{j\}} \in \mathcal{P}(f)$ adódik, ami ellentmond a w választásának.
$y(E) = (y + w)(E) - w(E)$ független az y és a w választásától, mivel $y + w$ az $x + s$-nek, a w pedig az s-nek egy $\mathcal{P}(f)$-bázisa. Ez bizonyítja, hogy $\mathcal{P}_w$ polimatroid. Továbbá $x \in \mathcal{P}_w$ pontosan akkor, ha $x(E) = y(E)$, függetlenül a w-től, vagyis $\mathcal{P}_w$ nem függ a w választásától.

Bibliográfia

[1] Ravindra K. Ahuja, Tomas L. Magnanti, James B. Orlin: *Network Flows* 1993, PRENTICE HALL, Upper Saddle River, New Jersey 07458.

[2] E. Cohen, N. Meggido. *Strongly polynomial-time and NC algorithms for detecting cycles in dynamic graphs*, Journal of the Association for Computing Machinery 40 (1993) 791-832.

[3] Carolyn Haibt Norton, Serge A. Plotkin, Éva Tardos. *Using Separation Algorithms in Fixed Dimension*, Journal of Algorithms 13 (1992), no 1, pp.79-98.

[4] J. Edmonds és E. Johnson. *Matching: a well-solved class of linear programs*, Combinatorial Structures and Their Applications, R. K. Guy, E. Milner, and N. Sauer, Eds. Gordon and Beach, New York, 1970, pp.89-92.

[5] A. Frank. *A Survey on T-joins, T-cuts, and Conservative Weightings*, Combinatorics, Paul Erdős is Eighty (Volume 2) Keszthely (Hungary), 1993 Budapest, 1996, pp.213-252.

[6] W.H. Cunningham. *Testing membership in matroid polyhedra*, Journal of Combinatorial Theory, Series B, 36 (1984), pp. 161-188.

[7] H. Narayanan. *A rounding technique for the polymatroid membership problem*, Linear Algebra and its Applications, 221 (1995), pp. 41-57.

[8] W.H. Cunningham. *Optimal attack and reinforcement of a network*, Journal of the Association for Computing Machinery, 32 (1985), pp. 549-561.

[9] Eddie Cheng, William H. Cunningham. *A faster algorithm for computing the strength of a network*, Information Processing Letters 49 (1994) 209-212

[10] M.Grötschel, L. Lovász, and A. Schrijver. *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1:169-197,1981.

[11] Éva Tardos. *A strongly polynomial minimum cost circulation algorithm*, Combinatorica, 5:247-255,1985.

[12] D. R. Fulkerson, Gary C. Harding. *Maximizing the minimum source-sink path subject to a budget constraint*, Math. Program. 13. 116-118, 1975.

[13] D. R. Fulkerson. *Increasing the capacity of a network: then parametric budget problem*, Management Science 5 (1959) 472-483.

[14] G. C. Harding. *Some budgeted optimization problems and the edge disjoint branchings problem* Dissertation, Cornell University, 1977.

[15] J. Edmonds *Submodular functions, matroids, and certain polyhedra*, Combinatorial Structures and Their Applications, R. K. Guy, E. Milner, and N. Sauer, Eds. Gordon and Beach, New York, 1970, pp.69-87.

[16] G. N. Frederickson, R. Solis-Oba. *Increasing the weight of minimum spanning trees*, Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, 1996, pp. 539-546.

[17] G. N. Frederickson, R. Solis-Oba, *Efficient algorithms for robustness in matroid optimization*, Proceedings of the Eight Annual ACM-SIAM Symposium on Discrete Algorithms, 1997, pp. 659-668.

[18] G. N. Frederickson, R. Solis-Oba, *Algorithms for Robustness in Matroid Optimization*, Combinatorica, megjelenés alatt.

[19] M. Held, R. Karp. *The traveling salesman problem and minimum spanning trees*, Operations Research 18, 1138-1162.

[20] M. Held, R. Karp. *The traveling salesman problem and minimum spanning trees, Part II.*, Mathematical Programming 6, 62-68.