Online jegyzet az Egészértékű Programozás I. és II. tárgyhoz

Király Tamás, Kis Tamás és Szegő László

2024. december 9.

Egészértékű programozás I. vizsgatematika

2020. tavasz

1. Az egészértékű lineáris programozási feladat definíciója, néhány alapvető típus (termelési feladat, hátizsák-feladat, szolgáltató-elhelyezés, szabási feladat, stb.). Vegyes programozási feladat. Egészértékű változókkal kifejezhető: indikátor változó, rácsból vett értékek, poliéderek uniója (1).
2. LP relaxáció, egész pontok konvex burka, Meyer tétele (2.1).
3. Hilbert bázisok, unimoduláris, és teljesen unimoduláris mátrixok, Graver féle teszthalmaz (2.2, 5.6).
4. Teljes duális egészértékűség definíciója, jellemzése Hilbert bázisokkal (2.27. lemma). A 2.25 tétel bizonyítása Hilbert bázisok segítségével (azaz a 2.27. lemmát használva).
5. Nem megoldhatóság, korlátok nem megoldható rendszer méretére, valalmint az optimális megoldást adó IP méretére(2.3)
6. A diszjunktív elv, Gomory-féle egész, és vegyes vágás, metszet vágások (3.5, 3.6, 3.7, 3.9).
7. LP alapú kolátozás és szétválasztás (branch and bound) módszer. Alkalmazás a hátizsák-feladatra (5.2).
8. Lagrange relaxáció. A Lagrange duális feladat, Geoffrion tétele. Szubgradiens módszer a Lagrange duális feladat megoldására (5.3).
9. Az utazó ügynök probléma definíciója, komplexitása. Alsó korlátok: 1-fa korlát, Held-Karp korlát, lineáris programozási alsó korlát. Az utóbbi kettő megegyezik (7.1,7.2).
10. Az utazó ügynök feladt megoldása heurisztikus módszerekkel. Lokális keresés, túranövelő módszerek, túrajavító módszerek, Christofides heurisztikája, Lin-Kernighan algoritmus (6,6.4,6.5).
11. Oszlopgenerálás és alkalmazásai. Egydimenziós szabási feladat (5.5.2), Dantzig-Wolfe dekompozíció.
12. Benders dekompozíció, alkalmazás a szolgáltató elhelyezési problémára (5.4).
13. Dinamikus programozás: hátizsákfeladat különböző variánsai, maximális súlyú stabil halmaz fában, maximális pozitív súlyú r-gyökerű részfa r-gyökerű fában. (4).

Egészértékű programozás II. vizsgatematika

2024. ősz

1. Teljes duális egészértékűség, Gomory-Chvátal-vágások, Chvátal-rang (3.1, 3.2, 3.3; a bizonyítások a 3.3-ból kellenek)
2. Vágások az utazóügynök feladatra: virág-egyenlőtlenségek és eldöntésük (7.4, 11.6)
3. Fedési vágások és klikk-vágások. Vágások felemelése bináris feladatoknál, alkalmazás a stabil halmaz poliéderre és fedési vágásokra (3.4, 3.13)
4. Felemelés és vetítés (3.14)
5. Szemidefinit programozás. Lovász-féle ϑ függvény. MAXCUT közelítése: Goemans és Williamson algoritmusa (`http://homepages.cwi.nl/~monique/files/laurent.pdf`)
6. Rácsok, Minkowski tétel (8.1, 8.1.1 nem kell). Racionális számok közelítése kis nevezőjű törtekkel (8.4).
7. Redukált bázisok, LLL algoritmus (8.2).
8. Legközelebbi vektor közelítése. Egészértékű programozási feladat megoldása fix dimenzióban (8.3).
9. Jain iteratív kerekítő algoritmusa az irányítatlan hálózattervezési feladatra (9.3). Lau és Singh iteratív relaxációs algoritmusa a fokszámkorlátos feszítő fa feladatra (9.4)
10. Diszkrepancia: Spencer és Gluskin tételei, Lovett-Meka algoritmus, Rothvoss algoritmusa (9.5)

Tartalomjegyzék

1. fejezet

Példák egészértékű optimalizálásra

1.1. Hozzárendelési feladat

Adott n darab munka és elvégzésükhöz n ember. Tudjuk, hogy az i-edik ember a j-edik munkát c_{ij} összegért végzi el. Célunk a munkák olyan kiosztása, amely a lehető legkisebb költségráfordítással jár.

Vezessük be az x_{ij} vátozókat, amelyek értéke 1, ha az i-edik ember a j-edik munkát kapja, és 0, ha nem.

A feladatkiosztást leíró feltételek ekkor a következők:
egyrészt minden munkát el kell végeztetnünk:

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, 2, \dots n,$$

másrészt minden ember csak egy munkát kaphat (a számosságok miatt pontosan egy munkát kap):

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, 2, \dots n.$$

A célfüggvény pedig a következő.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

1.2. Hátizsák feladat

Többféle hátizsák feladatot is ismerünk, amelyekkel bővebben majd a 4.2 fejezetben foglalkozunk. Most az úgynevezett bináris hátizsák feladatot tekintjük. Adott egy b költségvetésű pénzügyi keret (magyarán b Forint), és n darab befektetési lehetőség, amelyek költségigénye a_i összeg az $i = 1, 2, \dots n$ értékekre. A befektetések várható hozama c_i $(i = 1, \dots, n)$. Célunk a lehető legnagyobb hozam elérése.

Az x_i változó legyen 1, ha befektetünk pénzt az i-edik lehetőségbe, és 0, ha nem.

Mivel az elköltendő pénzösszeg nem haladhatja meg a rendelkezésünkre állót:

$$\sum_{i=1}^{n} a_i x_i \le b.$$

A maximalizálandó függvény pedig:

$$\sum_{i=1}^{n} c_i x_i.$$

1.3. Halmazfedési feladat

Adott egy alaphalmaz és annak néhány részhalmaza, amelyeknek költsége is van. Válasszuk ki a legkisebb összeköltséggel részhalmazok egy olyan rendszerét, amely az alaphalmaz minden elemét tartalmazza.

A feladat egy lehetséges motivációja: tekintsük egy nagyvárosban a lehetséges tűzoltó-állomások halmazát azok megépítési költségeivel együtt. Továbbá tekintsük a város térképét, amelyen bejelöltük, hogy az egyes lehetséges állomásokról mely területek érhetők el a helyi szabályozás szerinti maximális kiszállási időn belül. Célunk a legkisebb költséggel egy olyan tűzoltó-állomás hálózat létrehozása, amely a város minden részébe a megadott időkorlát alatt egységeket tud küldeni.

Az x_j változó legyen 1, ha kiválasztjuk a j-edik részhalmazt (példánkban a j-edik lehetséges állomásról elérhető területet), és 0, ha nem.

Tegyük fel, hogy m darab részhalmazunk van, az alaphalmaz pedig n elemű. Az a_{ij} érték legyen 1, ha az alaphalmaz i-edik eleme benne van a j-edik részhalmazban. Ekkor, mivel minden elemnek benne kell lennie valamelyik kiválasztandó részhalmazban:

$$\sum_{j=1}^{m} a_{ij} x_j \geq 1 \qquad i = 1, 2, \ldots, n.$$

A minimalizálandó költségfüggvény pedig:

$$\sum_{j=1}^{m} c_j x_j.$$

1.4. Az utazóügynök feladata

Adott a $G = (V, E)$ teljes gráf és a $c : E \to \mathbb{R}_+$ élhossz-függvény. Feladatunk a legrövidebb olyan körnek a megkeresése, amely a gráf minden pontján áthalad.

Az alapgráf minden élének feleltessünk meg egy változót, és tekintsük az alábbi egyenlőtlenség-rendszert.

$$x(\delta(v)) = 2 \quad \forall v \in V$$
$$x_e \in \{0, 1\} \quad \forall e \in E.$$

Ennek megoldásai a gráfot fedő diszjunkt körök karakterisztikus vektorai. A következő feltételek hozzávétele után azonban már csak a Hamilton-körök lesznek a megoldások.

$$x(\delta(S)) \geq 2 \quad \forall S \subset V, S \neq \emptyset.$$

A célfüggvény pedig a következő:

$$\min \sum_{e \in E} c_e x_e.$$

1.5. Szolgáltató-elhelyezési feladat (facility location)

Adott potenciális szolgáltatóhelyek $N = \{1, 2, \ldots, n\}$ és megrendelők $M = \{1, 2, \ldots, m\}$ halmaza. Legyen a j szolgáltatóhely megnyitásának fix költsége f_j, kapacitása pedig u_j. Az i megrendelőnek d_i igénye van; ha egységnyi igényét a j szolgáltatóhely elégíti ki, akkor az c_{ij} költséget jelent. Döntsük el, hogy mely szolgáltatóhelyeket használjuk (amelyek feltevés szerint elegendően nagy kapacitásúak), és az egyes megrendelők igényeinek mekkora részét teljesítsék az egyes szolgáltatóhelyek.

Vezessük be az y_j bináris változókat, amelyek értéke 1, ha használjuk a j szolgáltatóhelyet, és 0, ha nem. Az x_{ij} nemnegatív változó jelölje azt, hogy mekkora igényét teljesíti az i megrendelőnek a j szolgáltatóhely.

Mivel minden megrendelő teljes igényét ki kell elégítenünk, ezért:

$$\sum_{j \in N} x_{ij} = d_i \qquad \forall i \in M.$$

Ha a j raktárból szállítunk valahová, akkor a következő egyenlőtlenségek biztosítják, hogy valóban $y_j = 1$ legyen a $j \in M$ raktárra:

$$x_{ij} \le u_j y_j \qquad \forall i \in M.$$

A célfüggvényünk pedig a következő.

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j.$$

1.6. Termelésütemezési feladat

Egy adott termék termeléséről kell döntenünk egy n periódusú időszakban. Az alábbi adatokkal rendelkezünk:

f_t: a t-edik periódusban a termelés fix költsége,

p_t: a t-edik periódusban az egységnyi mennyiség termelésének költsége,

h_t: a t-edik periódusban az egységnyi mennyiség tárolásának költsége,

d_t: ennyi mennyiségű termékre van igény a t-edik periódusban.

Definiáljuk a következő változókat.

x_t: a t-edik periódusban termelt mennyiség,

z_t: a t-edik periódus végén tárolt mennyiség,

$y_t = 1$, ha a t-edik periódusban termelünk, 0, ha nem.

A célfüggvény és a feltételek:

$$\min \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t z_t + \sum_{t=1}^{n} f_t y_t,$$

$$z_{t-1} + x_t = d_t + z_t \qquad \forall t = 1, 2, \ldots, n,$$

$$x_t \le M y_t \qquad \forall t = 1, 2, \ldots, n,$$

$$z_0 = 0, z_t, x_t \ge 0, y_t \in \{0, 1\} \qquad \forall t = 1, 2, \ldots, n,$$

ahol M egy megfelelően nagy szám ahhoz, hogy ne korlátozza a megtermelendő mennyiségeket. Amennyiben $z_n = 0$-nak is teljesülnie kell, akkor M választható például $\sum_{i=t}^{n} d_i$ értéknek az y_t-hez tartozó sorban.

1.7. Rácsból vett értékek

Egészértékű feladatként azt is fel tudjuk írni, ha azt szeretnénk, hogy $x \in \mathbb{R}^n$ vektorváltozónk csak egy $\mathcal{L}(B) \subseteq \mathbb{R}^n$ rács pontjait vehesse fel értékként, ahol B egy $n \times d$-es d rangú mátrix (a rács definíciója a 8.1 fejezetben található). Tegyük fel hogy feladatunk a következő:

$$\max\{cx : Ax \le b,\ x \in \mathcal{L}(B)\}.$$

Ezt egészértékű programozási feladatként a következőképpen írhatjuk fel $y \in \mathbb{Z}^d$ változók segítségével:

$$\max\{(cB)y : (AB)y \le b,\ y \in \mathbb{Z}^d\}.$$

1.8. Korlátos poliéderek uniója

Adott k darab korlátos poliéder: $P_i = \{x \in \mathbb{R}^n : A^i x \le b^i,\ -d \le x \le d\}$ $(i = 1, \ldots, k)$. A következő feladatot szeretnénk modellezni:

$$\max\{cx : x \in \cup_{i=1}^k P_i\}.$$

Létezik olyan β szám amire $-d \le x \le d$ esetén $A^i x \le b^i + \beta\mathbf{1}$ minden i-re. Ennek segítségével a felírás vegyes programozási feladatként:

$$\begin{aligned}
&\max cx \\
&-d \le x \le d \\
&A^i x \le b^i + \beta(1 - y_i)\mathbf{1} && (i = 1, \ldots, k) \\
&\sum_{i=1}^k y_i = 1 \\
&y \in \{0,1\}^k
\end{aligned}$$

2. fejezet

Poliéderben lévő egész pontok konvex burka

Az alábbiakban röviden összefoglaljuk a jegyzetben használt fogalmakat és jelöléseket.

Az $\{x \in \mathbb{R}^n : Ax \leq b\}$ poliédert **racionális poliédernek** mondjuk, ha az A mátrix és a b vektor elemei racionális számok. Feltehetjük, hogy ekkor A és b elemei egészek.

Az $x \in \mathbb{R}^n$ vektort **egész vektornak** mondjuk, ha minden komponense egész. A vektor elnevezés mellett használni fogjuk a pont elnevezést is. Pontként többnyire akkor hivatkozunk rá, amikor poliéderes környezetben vagyunk, vektorként pedig akkor, amikor valamely térnek egy elemére gondolunk.

A P poliéder **egész poliéder**, ha minden oldala tartalmaz egész pontot. Egyenesmentes, azaz csúcsos poliéder esetén ez azt jelenti, hogy a csúcsai egész pontok.

A fejezet elején az Olvasó figyelmébe ajánljuk a 11.1 fejezet lineáris programozási emlékeztetőjét.

2.1. Meyer tétele

A továbbiakban P_I jelöli a P poliéder egész pontjainak konvex burkát. (Emlékeztetünk arra, hogy egy poliéder karakterisztikus kúpja a poliéder által tartalmazott irányok halmaza.)

2.1. tétel (Meyer, 1974)**.** *Ha P racionális poliéder, akkor P_I is poliéder. Ha $P_I \neq \emptyset$, akkor P és P_I karakterisztikus kúpjai megegyeznek.*

A fenti tétel a nemracionális poliéderekre nem érvényes. Előfordul, hogy csak végtelen sok féltér metszereként áll elő a P_I „poliéder". Az is előfordulhat, hogy P_I nem zárt halmaz!

Bizonyítás. Tekintsük a P Motzkin-tétel (11.4 tétel) szerinti $Q + C$ előállítását, ahol feltehetjük, hogy a C kúpot az $y_1, y_2, \ldots, y_s$ egész vektorok generálják. Legyen

$$B := \{\sum_{i=1}^{s} \mu_i y_i : 0 \leq \mu_i \leq 1, i = 1, 2, \ldots, s\}.$$

Megmutatjuk, hogy $P_I = (Q + B)_I + C$, ami elegendő, hiszen $Q + B$ korlátos halmaz, azaz $(Q + B)_I$ valóban politóp, s így poliéder is.

Először bebizonyítjuk, hogy $P_I \subseteq (Q + B)_I + C$. Mivel $(Q + B)_I + C$ konvex halmaz, elég belátni, hogy a P minden egész p pontja benne van.

Motzkin tétele szerint p felírható $q + c$ alakban, ahol $q \in Q, c \in C$. Ekkor léteznek olyan nemnegatív μ_i együtthatók, melyekkel:

$$p = q + c = q + \sum_{i=1}^{s} \mu_i y_i =$$

$$= (q + \sum_{i=1}^{s}(\mu_i - \lfloor\mu_i\rfloor)y_i) + \sum_{i=1}^{s}(\lfloor\mu_i\rfloor)y_i) =: (q+b) + c'.$$

$b \in B$ nyilvánvalóan teljesül, továbbá $q+b$ egész, mivel $q+b = p - c'$, azaz előáll két egész vektor különbségeként.

Most bebizonyítjuk, hogy $P_I \supseteq (Q+B)_I + C$.

$$(Q+B)_I + C \subseteq P_I + C = P_I + C_I \subseteq (P+C)_I = P_I,$$

ahol felhasználtuk, hogy $C_I = C$, mivel C racionális. □

2.2. Hilbert-bázisok, unimodularitás

2.2. definíció. Legyen $C \subseteq \mathbb{R}^n$ egy poliéderkúp. Az $a_1, a_2, \ldots, a_m$ C-beli egész vektorok véges halmazát **Hilbert-bázisnak** nevezzük, ha minden C-beli egész vektor előáll az a_i vektorok nemnegatív egész együtthatós lineáris kombinációjaként.

Nincs minden poliéderkúpnak Hilbert-bázisa. $C = \{(x,y) \in \mathbb{R}^2 : x \geq 0, \sqrt{5}x \leq y\}$.

2.3. tétel (Gordan 1873, Hilbert 1890)**.** *Minden racionális poliéderkúpnak létezik Hilbert-bázisa.*

Bizonyítás. Weyl tétele (11.2 tétel) szerint feltehetjük, hogy a C racionális poliéderkúp az $y_1, y_2, \ldots, y_s$ egész vektorok által generált kúpként is előáll. Legyen $H := \{a_1, a_2, \ldots, a_m\}$ a következő korlátos halmazba eső összes egész vektor.

$$V := \{\sum_{i=1}^{s} \lambda_i y_i : 0 \leq \lambda_i \leq 1, i = 1, 2, \ldots, s\}.$$

Megmutatjuk, hogy H a C Hilbert-bázisa.

Legyen b egy tetszőleges C-beli egész pont. Ekkor

$$b = \sum_{i=1}^{s} \mu_i y_i = \sum_{i=1}^{s}(\mu_i - \lfloor\mu_i\rfloor)y_i + \sum_{i=1}^{s}\lfloor\mu_i\rfloor y_i.$$

Az első szumma eredményét jelölje v. $v \in V$ triviálisan teljesül. Mivel az y_i vektorok benne vannak H-ban, elegendő azt megmutatni, hogy v is benne van a H halmazban. $v = b - \sum_{i=1}^{s}\lfloor\mu_i\rfloor y_i$, azaz v megkapható egész vektorok összegeként, ezért v is egész, tehát valóban H-ban van. □

Mikor nevezünk egy kúpot csúcsosnak? Ha van csúcsa, azaz ha nem tartalmaz egyenest. Ez azt jelenti, hogy végesen generált kúp esetén a kúpot generáló nemnulla vektoroknak nem létezik nemtriviális, nemnegatív lineáris kombinációja, amely a 0 vektort adná eredményül. A szemléletünk által elvárható módon most a Farkas-lemma (11.5 tétel) következményeként megmutatjuk, hogy ez azzal ekvivalens, hogy létezik egy vektor, amellyel az összes nemnulla generáló vektor hegyesszöget zár be, azaz érvényes az alábbi, ahol a B mátrix sorai a kúp generáló vektorai.

2.4. lemma. *Az adott B mátrixhoz akkor és csak akkor létezik x megoldása a $Bx > 0$ rendszernek, ha nem létezik megoldása a következőnek:*

$$yB = 0, y \geq 0, y \neq 0.$$

Bizonyítás.

$$\exists x : Bx > 0 \Leftrightarrow \exists x : Bx \geq \mathbf{1} \Leftrightarrow \exists x : Bx \leq -\mathbf{1}.$$

A Farkas-lemma (11.5 tétel) szerint ez azzal ekvivalens, hogy

$$\nexists y : yB = 0, y \geq 0, y(-\mathbf{1}) < 0 \Leftrightarrow \nexists y : yB = 0, y \geq 0, y \neq 0.$$

□

2.5. tétel (van der Corput 1931)**.** *Ha C csúcsos kúp, akkor létezik egyértelmű minden Hilbert-bázis által tartalmazott Hilbert-bázisa.*

Ha C-nek nincs csúcsa, akkor nem létezik egyértelmű minimális Hilbert-bázis. Például $\mathbb{R}$-et mint kúpot tekintve, az összes (p, q) pár Hilbert-bázis, ahol p negatív egész, q pozitív egész, és relatív prímek.

Bizonyítás. Legyen M azon C-beli nemnulla, egész vektorok halmaza, amelyek nem állnak elő két C-beli egész vektor összegeként. M elemeinek nyilván minden Hilbert-bázisban benne kell lennie. M véges halmaz, mivel az előző, 2.3 tétel szerint létezik H Hilbert-bázis. Az állítjuk, hogy M Hilbert-bázis.

Mivel C csúcsos, ezért az előző lemma szerint létezik egy $y \in \mathbb{R}^n$ vektor, amelyre $yx > 0$ teljesül a C minden nemnulla x elemére.

Tegyük fel indirekten, hogy a C-beli c egész vektor nem áll elő M-ből nemnegatív egész együtthatós lineáris kombinációként. Legyen c olyan, amelyre az yc skaláris szorzat minimális! Létezik ilyen c, hiszen a véges H halmazban kell lennie.

Ekkor mivel $c \notin M$, $c = c_1 + c_2$, ahol c_1, c_2 C-beli egész vektorok. Mivel $vc_1 < vc, vc_2 < vc$, ezért c_1 és c_2 előáll M-beliekből, következésképpen c is, ami ellentmondás. □

Legyen most A egy $m \times m$-es nemszinguláris mátrix egész elemekkel. Azt vizsgáljuk, hogy A oszlopai mikor alkotják az általuk generált kúp Hilbert-bázisát.

2.6. tétel. *A oszlopai pontosan akkor alkotják az általuk generált kúp Hilbert-bázisát, ha* $|\det(A)| = 1$.

Bizonyítás. Ha $|\det(A)| = 1$, akkor A^{-1} is egész mátrix. A generált kúp tetszőleges egész b elemére az őt adó lineáris kombináció együtthatóit $A^{-1}b$ adja, tehát ezek egészek, azaz az oszlopok Hilbert-bázist alkotnak.

A másik irányhoz tegyük fel, hogy az oszlopok Hilbert-bázist alkotnak. Ebből következik, hogy ha egy egész b-re $A^{-1}b$ nemnegatív, akkor egész. Azt állítjuk, hogy minden egész b-re $A^{-1}b$ egész. Ha adott b-re ez nem teljesülne, akkor vegyünk egy olyan egész y vektort, amire $y + A^{-1}b \geq 0$, ; legyen $b' := A(y + A^{-1}b) = Ay + b$. Ekkor b' egész, de $A^{-1}b'$ nemnegatív és nem egész, ami ellentmond a Hilbert-bázis tulajdonságnak.

Beláttuk tehát, hogy $A^{-1}b$ egész minden egész b-re, speciálisan minden egységvektorra is. Ezért viszont A^{-1} egész mátrix, tehát $\det(A^{-1})$ egész. A $\det(A^{-1})\det(A) = 1$ összefüggés miatt tehát $|\det(A)| = 1$. □

2.7. definíció. Egy A $m \times n$-es m rangú egész mátrix *unimoduláris*, ha minden bázis determinánsa 1 vagy -1.

2.8. tétel (Veinott, Dantzig, 1968)**.** *Legyen A egy $m \times n$-es m rangú egész mátrix. A akkor és csak akkor unimoduláris, ha a $P_b := \{x : \; Ax = b, \; x \geq 0\}$ poliéder minden egész b-re egész poliéder.*

Bizonyítás. Ha A unimoduláris, akkor tetszőleges egész b-re a P_b poliéder minden bázismegoldása egész, tehát P_b egész poliéder.

A fordított irányhoz legyen B egy bázis. Mivel minden b-re P_b minden bázismegoldása (speciálisan a B-hez tartozó is, ha megoldás) egész, ezért B oszlopai az általuk generált kúp Hilbert-bázisát alkotják. A 2.6. Tétel szerint ekkor $|\det(B)| = 1$. □

2.9. definíció. Egy mátrix *teljesen unimoduláris (TU)*, ha minden négyzetes D részmátrixára $\det(D) \in \{0, 1, -1\}$.

2.10. állítás. *Az A egész mátrix pontosan akkor TU, ha az $[A, I]$ mátrix unimoduláris.*

Bizonyítás. Könnyű. □

2.11. tétel (Hoffman, Kruskal, 1956)**.** *Az A egész mátrix pontosan akkor TU, ha a $Q_b := \{x : \ Ax \leq b, \ x \geq 0\}$ poliéder minden egész b-re egész poliéder.*

Bizonyítás. Q_b egész poliéder minden egész b-re $\Leftrightarrow$ $\{z : \ [A, I]z = b, \ x \geq 0\}$ egész poliéder minden egész b-re $\Leftrightarrow$ $[A, I]$ unimoduláris $\Leftrightarrow$ A TU. □

2.12. tétel (Hoffman, Kruskal, 1956)**.** *Ha az A egész mátrix TU, akkor az $\{x : \ Ax \leq b\}$ poliéder minden egész b-re egész poliéder.*

Bizonyítás. A TU $\Rightarrow$ $[A, -A]$ TU $\Rightarrow$ $\{z : \ [A, -A]z \leq b, \ z \geq 0\}$ egész poliéder minden egész b-re $\Rightarrow$ $\{x : \ Ax \leq b\}$ egész poliéder minden egész b-re. □

Legyen $D = (V, E)$ egy irányított gráf, és legyen $F \subseteq E$ egy olyan kijelölt élhalmaz, ami irányítatlan értelemben feszítő fa. Tekintsük a következő A mátrixot, aminek sorai F éleivel, oszlopai pedig $E - F$ éleivel vannak indexelve. Egy adott $uv \in E - F$ élhez tartozik F-ben egy egyértelmű (nem feltétlenül irányított) út v-ből u-ba. Legyenek ennek élei $e_1, \ldots, e_k$. Az A mártix uv-hez tartozó oszlopa legyen a következő: egy e_i élhez 1 tartozik, ha iránya megegyezik az út irányával, és -1, ha az irány fordított; a többi F-beli élhez pedig 0 tartozzon.

2.13. definíció. Egy mátrixot *hálózati mátrixnak* nevezünk, ha előáll egy irányított gráfból és egy feszítő fából a fenti módon.

2.14. állítás. *Hálózati mátrix részmátrixa is hálózati. Sort vagy oszlopot -1-gyel szorozva is hálózati mátrixot kapunk.*

Bizonyítás. Könnyű. □

2.15. tétel (Tutte, 1965)**.** *Minden hálózati mátrix teljesen unimoduláris.*

Bizonyítás. Elég belátni négyzetes hálózati mátrixra. Használjunk indukciót a mátrix méretére. Legyen $D = (V, E)$ és F a mátrixot adó irányított gráf és fa, és legyen $uv \in F$ a fának egy u levélre illeszkedő éle (feltehető, hogy így van irányítva). Ha uv sorában csak 1 nemnulla elem van, akkor a determináns kifejtési szabály miatt indukcióval kész vagyunk. Különben feltehető, hogy pontosan 1 $su \in E - F$ él lép u-ba, a többi él kilép u-ból. Könnyű ellenőrizni, hogy ha su oszlopát hozzáadjuk egy $ut \in E - F$ él oszlopához, akkor újra hálózati mátrixot kapunk: az ut kihagyásával és egy st él hozzávételével kapott gráfhoz tartozót. A determináns persze nem változik. Ezt a műveletet elvégezve minden $ut \in E - F$ élre olyan mátrixot kapunk, ahol uv sorában csak 1 nemnulla elem van, ilyenről pedig már láttuk hogy determinánsa 0, 1 vagy -1. □

2.3. Nemmegoldhatóság

A Caratheodory-tétel egy érdekes egészértékű analogonját adja a következő tétel.

2.16. tétel (Bell és Scarf, 1977). *Ha az $\{a_i x \le \beta_i, 1 \le i \le m\}$ rendszernek nem létezik egész megoldása, akkor van egy legfeljebb 2^n elemű részrendszere, amelynek már szintén nincs egész megoldása.*

Bizonyítás. Feltehetjük, hogy a rendszer minimálisan nemmegoldható, azaz elhagyva belőle akár egyetlen egyenlőtlenséget is, megoldhatóvá válik az egész rácspontok körében. Ekkor minden j-re létezik x_j egész vektor, amelyre $a_j x_j > \beta_j$, és minden $i \ne j$ indexre $a_i x_j \le \beta_i$, akkor $m \le 2^n$.

Legyen $Z := \mathbb{Z}^n \cap \mathrm{conv}\{x_1, x_2, \ldots, x_m\}$. Ekkor a következő rendszernek nem létezik Z-beli megoldása (de persze az már lehetséges, hogy egész megoldása van).

$$\begin{aligned} a_1 x &< \min_{z\in Z}\{a_1 z : a_1 z > \beta_1\}, \\ &\vdots \\ a_m x &< \min_{z\in Z}\{a_m z : a_m z > \beta_m\}. \end{aligned} \tag{2.1}$$

Ugyanis, ha létezne $z \in Z$ ami teljesíti (2.1) valamennyi feltételét, akkor $a_i z \le \beta_i$, $i = 1, \ldots, m$, ami ellentmond annak, hogy a rendszer nem megoldható.

Válasszuk úgy a $\gamma_j \in \mathbb{R}$, $j = 1, 2, \ldots, m$ számokat, hogy $\gamma_j \ge \min_{z\in Z}\{a_j z : a_j z > \beta_j\}$, és a következő rendszernek ne létezzen Z-beli megoldása, ezen belül pedig $\sum_{j=1}^m \gamma_j$ maximális legyen.

$$\begin{aligned} a_1 x &< \gamma_1, \\ &\vdots \\ a_m x &< \gamma_m. \end{aligned} \tag{2.2}$$

Ilyen γ létezik (lásd (2.1)), γ_j felülről is korlátos ($\gamma_j \le a_j x_j$), a γ vektorok halmaza pedig zárt, hiszen a komplementere nyílt:

$$\{\gamma \in \mathbb{R}^m : \exists z \in Z,\ \text{melyre } a_i z < \gamma_i\ \forall\ i = 1, 2, \ldots m\} = \\ = \bigcup_{z\in Z}\{\gamma \in \mathbb{R}^m : a_i z < \gamma_i\ \forall\ i = 1, 2, \ldots m\},$$

ez utóbbi pedig nyíltak uniója.

Mivel $\sum_{j=1}^m \gamma_j$ maximális, ezért minden j-re létezik $y_j \in Z$, hogy $a_j y_j = \gamma_j$, és $a_i y_j < \gamma_i$ minden $i \ne j$ indexre (hiszen a $\sum_{j=1}^m \gamma_j$ maximalitása miatt bármelyik γ_j-t megnövelve $\varepsilon > 0$-val, már lenne megoldása a (2.2) rendszernek). Mivel indirekten $m > 2^n$, ezért léteznek k és l különböző indexek, amelyeknek a megfelelő komponenseik azonos paritásúak. Ekkor $\frac{y_k + y_l}{2} \in Z$, ami viszont kielégíti (2.2)-at. Ellentmondás. □

Megjegyezzük, hogy a fenti tételben szereplő 2^n éles, amint azt a következő példa mutatja $n \ge 2$ esetben:

$$\sum_{i\in I} x_i - \sum_{i\notin I} x_i \le |I| - 1 \qquad \forall\ I \subseteq \{1, 2, \ldots, n\}.$$

A fenti tétel egészértékű optimalizációs feladatokra vonatkozó következménye az alábbi.

2.17. tétel (Scarf, Todd, 1977). *Ha $\mu := \max\{cx : Ax \le b, x \text{ egész}\}$ véges, akkor létezik egy legfeljebb $2^n - 1$ sorból álló $A'x \le b'$ részrendszer, amelyre $\max\{cx : A'x \le b', x \text{ egész}\} = \mu$.*

Bizonyítás. A következő rendszernek nem létezik egész megoldása semmilyen $t \in \mathbb{Z}_+$ paraméterre.

$$\begin{aligned} Ax &\leq b, \\ cx &\geq \mu + \frac{1}{t}. \end{aligned} \tag{2.3}$$

Az előző, 2.16 tétel szerint minden t-re létezik egy legfeljebb 2^n sorból álló részrendszer, amelynek nem létezik egész megoldása. Mivel mindegyiknek tagja a $cx \geq \mu + \frac{1}{t}$, ezért létezik egy $A'x \leq b'$ keresett méretű rendszer, amely végtelen sok t-re szerepel, ami azt jelenti, hogy az $\{A'x \leq b', cx > \mu\}$ rendszernek nincs egész megoldása. □

2.4. Bonyolultság

Vizvári Béla jegyzetének 45. oldalán szereplő tétel, itt biz. nélkül.

2.18. tétel.

A küszöbgráfok előállítási tétele.

2.19. tétel (Chvátal és Hammer, 1977)**.** *Tetszőleges G gráfra az alábbi három állítás ekvivalens.*

1. *G előáll a ponthozzáadás, és pontszorzás műveletek alkalmazásával.*
2. *G küszöbgráf.*
3. *G-nek nincs feszített $2K_2$, P_4 vagy C_4 részgráfja.*

Bizonyítás. *1*-ből *2*, és *2*-ből *3* egyszerűen következik.

Most megmutatjuk *3*-ból *1*-et. A G pontszámára vonatkozó indukcióval bizonyítunk. Amennyiben G-ben van izolált pont (amelyre definíció szerint nem illeszkedik él), kész vagyunk. Ha nincs ilyen pont, akkor tekintsünk egy maximális fokszámú v pontot. Ha v minden más ponttal össze van kötve, újfent készen vagyunk, ha pedig nem ez a helyzet, tekintsünk egy z pontot, amely nem a szomszédja.

Mivel z nem izolált pont, létezik egy t szomszédja. Mivel v maximális fokú, kell léteznie olyan s szomszédjának, amely t-nek nem szomszédja. Ekkor a vt és uz élek meglététől függően a *3*-ban tiltott feszített részgráfokat kapunk, következésképpen v-nek minden más pont szomszédja. □

2.5. Teljesen duálisan egészértékű rendszerek

A poliéderes kombinatorika a lineáris porgramozás dualitástételét használja fel kombinatorikai feladatokkal kapcsolatos következtetések levonására. Ebben a fejezetben a teljesen duálisan egészértékűség fogalmát ismertetjük, és a vele kapcsolatos legfontosabb eredményeket.

A bevezetendő fogalom alapgondolata a következő. Kombinatorikus struktúrákhoz rendelünk poliédereket, és ezen poliéderek csúcsai ha egészek, akkor az elég sok információt ad nekünk. Belátjuk, hogy ha tetszőleges egész célfüggvény esetén egész az optimum értéke, akkor a poliéder is egész. Ezután egy poliéder egészségét majd úgy próbájuk bebizonyítani, hogy meghatározzuk egy olyan leíró rendszerét, amely duálisának az optimuma minden egész célfüggvényhez egész vektoron is felvétetik. Amint az ki fog derülni, ez egy nagyon gyümölcsöző elgondolás.

2.20. tétel (Hoffman, 1974)**.** *A P korlátos racionális poliéder akkor és csak akkor egész, ha a* $\max\{cx : x \in P\}$ *érték minden egész c vektorra egész.*

Bizonyítás. Világos, hogy egész poliéderre teljesül a feltétel, most lássuk be, hogy a P korlátos racionális poliéderre a feltétel teljesülése maga után vonja, hogy egész.

Legyen $v = (v_1, v_2, \dots, v_n)$ a P egy csúcsa, és legyen $c \in \mathbb{Z}^n$ olyan célfüggvény, melyre a következő maximum egyedül a v ponton vétetik fel: $\max\{cx : x \in P\}$. (Lásd 2. feladat.)

A c esetleges felszorzásával elérhető, hogy $cv > cu + u_1 - v_1$ fennáljon a P minden más u csúcsára. Ekkor viszont a $c' := (c_1 + 1, c_2, \dots, c_n)$ egész célfüggvényre szintén a v-n véteik fel $\max\{c'x : x \in P\}$. Ami viszont azt jelenti, hogy a két egész optimum $cv - c'v = v_1$ különbsége egész. Hasonlóan belátható, hogy a v minden koordinátája is egész, és mivel v tetszőleges csúcsnak választható, beláttuk, hogy a P egész poliéder. □

Edmonds és Giles ezt a tételt kiterjesztette tetszőleges poliéderre 1977-ben (tehát nem kell, hogy korlátos legyen, sőt még csúcsának sem kell lennie.)

2.21. tétel (Edmonds, Giles, 1977)**.** *A P racionális poliéder akkor és csak akkor egész, ha a* $\max\{cx : x \in P\}$ *érték minden egész c vektorra, ha véges, akkor egész.*

Bizonyítás. A bizonyításhoz a következő lemmát használjuk:

2.22. lemma. *Legyen A egész mátrix, és b egész vektor. Pontosan akkor létezik egész x vektor amire $Ax = b$, ha minden y vektorra, amire yA egész, yb is egész.*

Bizonyítás. Ha létezik jó egész x, akkor az $(yA)x = yb$ összefüggésből következik az y-ra vonatkozó tulajdonság. A fordított irányhoz feltehető, hogy $Ax = b$-nek van megoldása, és A sorai lineárisan függetlenek. Könnyen ellenőrizhető, hogy mind x mind y létezése szempontjából ekvivalens rendszert kapunk, ha oszlopokat felcserélünk, illetve egy oszlop egész többszörösét egy másikhoz adjuk. Ilyen műveletekkel a mátrix $[B, 0]$ alakba hozható, ahol B egész háromszögmátrix.

Mivel $B^{-1}[B, 0]$ egész mátrix, a feltevésből következik, hogy $B^{-1}b$ egész vektor. Így az $x := \binom{B^{-1}b}{0}$ vektor jó lesz, hiszen $[B, 0]x = b$. □

A tételhez azt kell bizonyítanunk, hogy a $P = \{x \ : \ Ax \le b\}$ poliéder, ahol A és b egész mátrix, ill. vektor, minden P' minimális oldalának van egész pontja. Tudjuk, hogy P' egy affin altér, ami leírható egy $A'x = b'$ egész együtthatós egyenlet rendszerrel. Erről kell megmutatni, hogy van egész megoldása. A lemma alkalmazásához be kell látnuk, hogy minden olyan y' vektorra, hogy $y'A'$ egész vektor, $y'b'$ egész szám. Először is megmutatjuk, hogy tetszőleges $y' \ge 0$-ra, a $c := y'A'$ (egész) célfüggvény vektor mellett P' pontjai a $\max\{cx \ : \ x \in P\}$ optimális megoldásai. Ugyanis legyen $x' \in P'$ tetszőleges pont. Ekkor $A'x' = b'$ teljesül. Tehát $cx' = (y'A')x' = y'(A'x') = y'b'$ a c definíciója, és x' választása miatt. Mivel P' poliéder a P egy oldala, $x' \in P$ is teljesül. Továbbá y' vektort 0 komponensekkel kiegészítve az $A'x \le b'$ sorain kívül, kapjuk, hogy $(y', 0)A = c$, valamint $(y', 0)b = y'b'$. Ebből az következik, hogy

$$\max\{cx \ : \ Ax \le b\} \ge cx' = y'b' = (y', 0)b \ge \min\{yb \ : \ yA = c, y \ge 0\}. \tag{2.4}$$

Mivel $\max\{cx \ : \ Ax \le b\} \le \min\{yb \ : \ yA = c, y \ge 0\}$ a gyenge LP dualitás tétel miatt, (2.4)-ben mindenütt egyenlőség teljesül. Tehát $x' \in P'$ valóban optimális megoldása $\max\{cx \ : \ Ax \le b\}$ -nek.

A tétel feltétele szerint a $\max\{cx \ : \ Ax \le b\}$ optimum értéke egész amennyiben c egész vektor, tehát

$$\mathbb{Z} \ni \max\{cx \ : \ x \in P\} = cx' = (y'A')x' = y'(A'x') = y'b',$$

ahol $x' \in P'$, tehát $y'b'$ egész.

Az $y' \geq 0$ feltétel elhagyható, ugyanis ha $y' \not\geq 0$ egész vektor, akkor legyen $\lambda > 0$ kellően nagy egész szám, amelyre $y'' = y' + \lambda(1, \dots 1) \geq 0$. Erre az y'' vektorra alkalmazva a fenti gondolatmenetet kapjuk, hogy $c := y''A'$ esetén c egész, és $y''b'$ is egész. De ekkor $y''b = (y' + \lambda(1, \dots, 1))b'$, és mivel $\lambda(1, \dots, 1)b'$ egész, ezért $y'b'$ is egész. Tehát beláttuk, hogy tetszőleges y' esetén, ha $y'A'$ egész vektor, akkor $y'b'$ egész szám. A lemmát alkalmazva kapjuk, hogy létezik egész x vektor amire $A'x = b'$, azaz P' tartalmaz egész pontot. $\square$

Ezen tétel fényében nagyon hasznosnak bizonyul az alábbi fogalom.

2.23. definíció. Az $Ax \leq b$ racionális számokból álló egyenlőtlenség-rendszert *teljesen duálisan egészértékűnek* mondjuk, ha minden olyan egész $c \in \mathbb{Z}^n$ célfüggvényre, melyre az optimum korlátos, a $\min\{by : yA = c, y \geq 0\}$ duális feladat optimuma egész y vektoron is felvétetik.

Az angol elnevezés (total dual integrality) után a fenti tulajdonságot TDI-ségnek rövidítjük.

2.24. tétel (Edmonds, Giles 1977). *Legyen a $P = \{x : Ax \leq b\}$ racionális poliéder, továbbá b egész vektor. Ha az $Ax \leq b$ rendszer teljesen duálisan egészértékű, akkor a P egész poliéder.*

Bizonyítás. Mivel b egész, a duális feladat optimuma egész, tehát a dualitástétel (11.9 tétel) szerint $\max\{cx : Ax \leq b\}$ is egész minden egész c-re, azaz a 2.21 tétel szerint P egész poliéder. $\square$

2.25. tétel (Giles és Pulleyblank, 1979). *Legyen P egy racionális poliéder. Ekkor létezik egy $Ax \leq b$ TDI rendszer, amelyre A egész mátrix, és $P = \{x : Ax \leq b\}$. Továbbá amennyiben P egész poliéder, b is választható egésznek.*

Bizonyítás. Legyen $P = \{x \in \mathbb{R}^n : Mx \leq d\}$, ahol M egész mátrix. Legyen $L := \{l \in \mathbb{Z}^n : l = yM, 0 \leq y \leq 1\}$, azaz azon egész vektorok halmaza, amelyek előállnak az M soraiból 0 és 1 közti együtthatókkal vett lineáris kombinációként. Világos, hogy L véges halmaz. Jelölje $\beta(l) := \max\{lx : x \in P\}$.

Most megadjuk a keresett rendszert. Álljon az $Ax \leq b$ az összes $lx \leq \beta(l)$ sorokból, ahol $l \in L$. Mivel az M minden sora benne van L-ben, ezért $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. Ha pedig P egész, akkor a b is egész.

Most megmutatjuk, hogy az $Ax \leq b$ TDI-rendszer. Legyen $c \in \mathbb{Z}^n$ olyan egész vektor, amelyre a $\max\{cx : x \in P\}$ korlátos. Megadjuk a duális $\min\{yb : yA = c, y \geq 0\}$ egy egész optimális megoldását. A dualitás tétel (11.9 tétel) szerint:

$$\max\{cx : Ax \leq b\} = \min\{yb : yA = c, y \geq 0\} =: z^*.$$

Jelölje y^* a duális egy olyan megoldását, amelyen a minimum felvétetik, és jelölje y^*_M az M soraihoz tartozó részét. A $c' := (y^*_M - \lfloor y^*_M \rfloor)M$ vektor benne van az L halmazban, továbbá $\beta(c') \leq (y^*_M - \lfloor y^*_M \rfloor)d$. Eszerint $z^* \geq \beta(c') + \lfloor y^*_M \rfloor d$, viszont z^* szitén optimumértéke a $\min\{yd : yM = c, y \geq 0\}$ feladatnak, azaz az M soraihoz tartozó duális változóknak $\lfloor y^*_M \rfloor$, a $c'x \leq \beta(c')$ sorhoz tartozó változónak pedig 1 értéket adva a duális egy egész optimális megoldását kapjuk. $\square$

A bizonyításban a racionális poliédert meghatározó egyenlőtlenség-rendszerhez hozzátettünk nagyon sok újabb redundáns feltételt, ami után a duális feladat megváltozott (változóinak a száma nőtt), és lehetővé tette, hogy egész optimum is létezzen. Az előadáson tekintettünk egy példát, amelyben a K_4 gráf 2-párosítási poliéderének minimális leírásáról láttuk, hogy nem TDI, viszont egy redundáns sort hozzávéve már TDI leírást kaptunk.

Leírunk egy másik bizonyítást is Giles és Pulleyblank tételére, ami a TDI tulajdonság Hilbert-bázisok segítségével történő jellemzését használja.

2.26. definíció. Legyen A egész mátrix, és F a $P = \{x: \ Ax \leq b\}$ poliéder egy oldala. Az A mátrix a_i sorát F-re nézve *aktívnak* nevezzük, ha $a_i x = b_i$ minden $x \in F$ vektorra. Jelölje C_F az F-re nézve aktív sorok által generált kúpot. A kiegészítő eltérések tétele értelmében C_F azon c vektorok halmaza, amikre az $\{Ax \leq b, \ \max cx\}$ feladatnak F összes pontja optimális megoldása; így C_F nem függ a P-t definiáló rendszertől.

2.27. lemma. *Legyen A egész mátrix. Az $Ax \leq b$ rendszer akkor és csak akkor TDI, ha $\{x: \ Ax \leq b\}$ minden minimális oldalára igaz, hogy a rá nézve aktív sorok Hilbert-bázist alkotnak.*

Bizonyítás. Ha $Ax \leq b$ TDI, és F a $P = \{x: \ Ax \leq b\}$ poliéder egy oldala, akkor minden egész $c \in C_F$-re a duális $\{yA = c, \ y \geq 0, \ \min yb\}$ feladatnak van olyan egész y megoldása, ahol $y_i = 0$ ha a_i nem aktív F-re nézve. Következésképp az F-re nézve aktív sorok Hilbert-bázist alkotnak.

Másik irány: tegyük fel, hogy egy egész c-re a primál feladat optimuma korlátos, de a duál feladatnak nincs egész megoldása. Létezik egy F minimális oldal, aminek pontjai c-re nézve optimálisak. Ekkor $c \in C_F$, de c nem áll elő F-re nézve aktív sorok nemnegatív egész lineáris kombinációjaként, mert az a duál feladat optimális megoldása lenne. $\square$

A 2.25. tétel alternatív bizonyítása. Legyen $A'x \leq b'$ a P egy definiáló rendszere, és P minden F minimális oldalára legyen H_F a C_F egy Hilbert-bázisa. Mivel C_F racionális kúp, és A' racionális mátrix, ezért F-re nézve aktív sorainak van olyan egész számú többszöröse, ami egész vektor C_F-ben, tehát előáll H_F elemeinek nemnegatív, egész lineáris kombinációjaként. Adott $h \in H_F$-re az $\alpha_h := hx$ érték azonos F minden x pontjára, és a $hx \leq \alpha_h$ egyenlőtlenséget P minden pontja teljesíti.

Az $Ax \leq b$ rendszer álljon a $hx \leq \alpha_h$ egyenlőtlenségekből minden $h \in H_F$-re és minden F minimális oldalra. Ekkor $A'x \leq b'$ minden lényeges egyenlőtlensége előáll ezek nemnegatív lineáris kombinációjaként (hiszen mindegyik aktív P valamelyik minimális oldalra nézve), így $Ax \leq b$ a P definiáló rendszere. A 2.27. Lemma értelmében az $Ax \leq b$ rendszer TDI.

Végül ha P egész poliéder, akkor P minden minimális oldala tartalmaz egész pontot, tehát a fenti α_h értékek egészek. $\square$

2.28. állítás. *Ha P teljes dimenziós racionális poliéder, akkor C_F csúcsos kúp minden F minimális oldalra.*

Bizonyítás. Tegyük fel, hogy $c \in C_F$ és $-c \in C_F$ valamilyen $c \neq 0$ vektorra, és legyen $\alpha := cx$ egy tetszőleges $x \in F$-re. Ekkor $cx \leq \alpha$ és $-cx \leq -\alpha$ is teljesül P minden pontjára, tehát P nem teljes dimenziós. $\square$

2.29. definíció. A P racionális poliédernek egy definiáló $Ax \leq b$ TDI rendszere *minimális*, ha tetszőleges egyenlőtlenség elhagyásával kapott $A'x \leq b'$ rendszer vagy nem TDI, vagy $P \neq \{x: \ A'x \leq b'\}$.

2.30. tétel (Schrijver, 1981)**.** *Legyen P teljes dimenziós racionális poliéder. Ekkor létezik egy egyértelmű minimális $Ax \leq b$ TDI rendszer, amire A egész és $P = \{x: \ Ax \leq b\}$. P akkor és csak akkor egész poliéder, ha b egész.*

Bizonyítás. Az $Ax \leq b$ rendszert a 2.25. Tétel bizonyításához hasonlóan definiáljuk, csak H_F-nek mindig a C_F csúcsos kúp egyértelmű minimális Hilbert-bázisát választjuk. Ekkor A egész és $Ax \leq b$ TDI.

Tegyük fel, hogy van olyan P-t definiáló $A''x \leq b''$ TDI rendszer, amire A'' egész, és P valamely F minimális oldalára és $h \in H_F$-re a $hx \leq \alpha_h$ egyenlőtlenség nem szerepel az

$A''x \leq b''$ rendszerben. Ekkor F minden pontja a $\max\{hx : A''x \leq b''\}$ lineáris program optimális megoldása, tehát a duális feladat minden optimális megoldásaban $y_i = 0$, ha a''_i az A''-nak nem aktív sora F-re nézve. Mivel h nem áll elő mint más C_F-beli egész vektorok nemnegatív egész lineáris kombinációja, a duális feladatnak nincs egész optimális megoldása, ami ellentmond a TDI tulajdonságnak.

Ha P egész poliéder, akkor a 2.25. Tétel bizonyításához hasonlóan látható, hogy b egész. □

Végül bebizonyítunk egy tételt, mely szerint egy TDI rendszerben egy egyenlőtlenséget egyenlőségre változtatva ismét TDI rendszert kapunk.

2.31. tétel. *Legyen $Ax \leq b$ TDI rendszer, és $ax \leq \beta$ ennek a rendszernek egy egyenőtlensége. Ekkor az $ax \geq \beta$ egyenlőtlenséget hozzávéve a rendszerhez TDI rendszert kapunk.*

Bizonyítás. Legyen c egy egész vektor amire az optimum korlátos, erre keresünk egész optimális duális megoldást. Legyen y^* egy optimális megoldás, és legyen az $ax = \beta$-hoz tartozó duális érték $\lambda^* - \mu^*$, ahol $\lambda^*, \mu^* \geq 0$. Legyen $c' = c + Na$, ahol $N \geq \mu^*$, N egész, és Na egész. Ha y^*-ban $\lambda^* - \mu^*$-t lecseréljük $\lambda^* + N - \mu^*$-re, akkor x^*-gal teljesülnek a komplementaritási feltételek, tehát x^* optimális megoldása a $\max\{c'x : \; Ax \leq b\}$ feladatnak. Mivel $Ax \leq b$ TDI rendszer volt, van c'-re egész optimális duális megoldás: y', λ', ami x^*-gal teljesíti a komplementaritási feltételeket. Ebből viszont látszik, hogy $y', \lambda' - N$ optimális duális megoldása az $ax \geq \beta$-val kiegészített feladatnak a c célfüggvényre. □

2.5.1. Feladatok

1. feladat. *Az alábbi bizonyításával mutassuk meg, hogy a TDI-ség használatakor nagyon fontos a jobb oldali b vektor egészértékűsége.*

Tetszőleges racionális $Ax \leq b$ rendszerhez létezik olyan t pozitív egész, amelyre $\frac{1}{t}Ax \leq \frac{1}{t}b$ „teljesen duálisan egészértékű”.

2. feladat. *Mutassuk meg, hogy tényleg létezik a 2.20 tétel bizonyításában használt olyan egész célfüggvény, amely egy korlátos poliéderen kizárólag egy előre megadott ponton veszi fel a maximumát.*

3. fejezet

Érvényes vágások

A lineáris programozás egyik alaperedménye annak meghatározása, hogy egy tetszőleges, megoldható $Ax \leq b$ egyenlőtlenség-rendszer megoldásai teljesítik-e az adott $wx \leq t$ egyenlőtlenséget. A választ a Farkas-lemma (vagy Farkas-tétel) adja meg, amely szerint ez pontosan akkor teljesül, ha létezik egy megfelelő méretű, nemnegatív y vektor, amelyre $yA = w$, $yb \leq t$. Ebben a fejezetben azt a kérdést vizsgáljuk, hogy az $Ax \leq b$ egyenlőtlenség-rendszer *egész* megoldásai mikor teljesítenek egy adott $wx \leq t$ egyenlőtlenséget.

3.1. Gomory-Chvátal-levezetések

3.1. definíció. A $wx \leq t$ vágást a $P = \{x : Ax \leq b\}$ poliéderre vonatkozóan *érvényes*nek nevezzük, ha az összes P-beli egész pont teljesíti.

Az $Ax \leq b$ egyenlőtlenség-rendszer által meghatározott P poliéder *(elsőfajú) Gomory-Chvátal-vágásának* (röviden *GC-vágásának*) nevezzük a $cx \leq d$ vágást, amennyiben c egész, és létezik az y nemnegatív vektor, amelyre $yA = c$, $\lfloor yb \rfloor = d$.

Könnyen látható, hogy a GC-vágások érvényes vágások a poliéder egész pontjaira vonatkozóan. Ám ennek a fordítottja még nem igaz, viszont az is világos, hogy a GC-vágásokkal és az eredeti egyenlőtlenség-rendszer egyenlőtlenségeivel újabb GC-vágásokat generálva szintén érvényes vágásokat kapunk. (A vágás elnevezés onnan származik, hogy a tekintett egyenlőtlenségeket esetleg nem teljesíti az eredeti poliéder minden pontja, így azokat a újonnan tekintett egyenlőtlenséggel levágjuk.)

Azt mondjuk, hogy a $wx \leq t$ vágásnak létezik *Gomory-Chvátal-levezetése* (röviden *GC-levezetése*), ha az eredeti egyelnőtlenség-rendszerhez hozzávett GC-vágásokból előállított GC-vágásokkal, és azoknak a rendszerhez való hozzávételének az ismételgetésével előáll GC-vágásként. Formálisan a definíció a következő:

A $wx \leq t$ vágásnak a $P = \{x : a_i x \leq b_i, i = 1, 2, \dots m\}$ poliédert definiáló rendszerből létezik GC-levezetése, ha léteznek az

$$a_{m+k}x \leq b_{m+k} \quad (k = 1, 2, \dots M)$$

vágások az

$$y_{kj} \quad (1 \leq k \leq M, 1 \leq j \leq m + k - 1)$$

nemnegatív számokkal együtt, amelyekre minden $k = 1, 2, \dots M$ értékre az $a_{m+k}x \leq b_{m+k}$ vágás az

$$a_i x \leq b_i \quad (i = 1, 2, \dots m + k - 1)$$

rendszer GC-vágása, és $a_{m+M} = w$, $b_{m+M} \leq t$.

Egy ilyen levezetés természetszerűleg azt is bizonyítja, hogy a $wx \le t$ vágás érvényes vágás, azaz P minden egész pontja teljesíti.

Az alábbi tétel Chvátaltól származik 1973-ból. (Bár Gomory 1960-as eredményeiből is levezethető.)

3.2. tétel. *Legyen $P = \{x : Ax \le b\}$ egy racionális korlátos poliéder, és $wx \le t$ egy érvényes vágás, melyre w egész vektor. Ekkor $wx \le t$-nek létezik Gomory-Chvátal-levezetése.*

Megjegyezzük, hogy az állítás igaz nem-korlátos racionális poliéderekre és korlátos nem-racionális poliéderekre is. A továbbiakban Schrijver 1980-ból származó bizonyítása szerint haladunk. A fenti tétel bizonyításához szükségünk lesz a következő speciális esetre.

3.3. tétel. *Legyen $P = \{x : Ax \le b\}$ egy korlátos racionális poliéder, amely nem tartalmaz egész pontot. Ekkor létezik a $0x \le -1$ vágásnak Gomory-Chvátal-levezetése.*

Vegyük észre, hogy az előző tétel a Farkas-lemma egészértékű megfelelője, különösképpen, ha a Farkas-lemma alábbi alakját tekintjük.

3.4. lemma (Farkas-lemma)**.** *Az alábbi két rendszer közül pontosan az egyik oldható meg.*

- $Ax \le b$
- $yA = 0, y \ge 0, yb = -1.$

A fenti tételekre a poliéder dimenziójára vonatkozó teljes indukciós bizonyítást adunk, amelyben nagy hasznunkra lesz a következő lemma.

3.5. lemma. *Legyen F a P korlátos racionális poliéder egy oldala. Ha $cx \le d$ az F egy GC-vágása, akkor létezik a P-nek egy $c'x \le d'$ GC-vágása, melyre*

$$F \cap \{x : c'x \le d'\} = F \cap \{x : cx \le d\}.$$

Ha $cx \le d$ GC-levezethető F-ből, akkor létezik egy P-ből GC-levezethető $c'x \le d'$ vágás, melyre

$$F \cap \{x : c'x \le d'\} = F \cap \{x : cx \le d\}.$$

Bizonyítás. Először a GC-vágásra vonatkozó állítást bizonyítjuk. Legyen $P = \{x : A_1x \le b_1, A_2x \le b_2\}$, ahol A_i és b_i $(i = 1, 2)$ egész komponenseket tartalmaz, és $F = \{x : A_1x \le b_1, A_2x = b_2\}$. Mivel $cx \le d$ GC-vágás, létezik y_1 és y_2, melyekre

$$y_1A_1 + y_2A_2 = c,$$

$$\lfloor y_1b_1 + y_2b_2 \rfloor = d.$$

Az y_2-nek lehetnek negatív koordinátái is mivel egyenlőségfeltételekhez tartozik (Egy $ax = \beta$ egyenlőséget az $ax \le \beta, (-a)x \le -\beta$ egyenlőtlenségekkel helyettesítve a két nemnegatív y együttható különbsége nem feltétlenül nemnegatív.).

Legyen c' és d' a következő:

$$c' := y_1A_1 + (y_2 - \lfloor y_2 \rfloor)A_2 = c - \lfloor y_2 \rfloor A_2,$$

$$d' := \lfloor y_1b_1 + (y_2 - \lfloor y_2 \rfloor)b_2 \rfloor = d - \lfloor y_2 \rfloor b_2.$$

Ekkor c' egész, és $c'x \le d'$ a P poliéder egy GC-vágása (mivel $y_2 - \lfloor y_2 \rfloor$ nemnegatív). Továbbá, mivel $d = d' + \lfloor y_2 \rfloor b_2$,

$$F \cap \{x : c'x \le d'\} = F \cap \{x : c'x \le d', \lfloor y_2 \rfloor A_2x = \lfloor y_2 \rfloor b_2\} = F \cap \{x : cx \le d\}.$$

A levezethetőségre vonatkozó állítás bizonyításához figyeljük meg, hogy a fenti bizonyításban a $c'x \le d'$ vágást úgy kaptuk, hogy $cx \le d$-hez az $A_2x \le b_2$ rendszer egyenlőtlenségeinek nemnegatív egész kombinációját adtuk hozzá. Ha most $cx \le d$-nek adott egy GC-levezetése F-ből, akkor a levezetés vágásaihoz sorrendben készíthetünk P-re érvényes vágásokat az $A_2x \le b_2$ rendszer egyenlőtlenségeinek megfelelő nemnegatív egész kombinációinak hozzáadásával. A módosított vágások F-ből ugyanazt vágják le mint az eredetiek; a módosított levezetés utolsó vágása legyen $c'x \le d'$. □

A 3.3. tétel bizonyítása. A P poliéder dimenziójára vonatkozó indukciót használunk. Ha P nulladimenziós, azaz egy pontból áll (amely tehát nem egész), akkor az állítás a dualitástétel egyszerű következménye. Tegyük fel, hogy P dimenziója legalább 1, és minden nála alacsonyabb dimenziós korlátos poliéderre az állítás igaz.

Legyen w egész vektor, és $wx \le t$ egy olyan egyenlőtlenség, amelyre $P \cap \{x : wx = t\}$ a P egy valódi oldala. Jelölje P' a $wx \le \lfloor t \rfloor$ GC-vágással kapott poliédert, azaz $P' := P \cap \{x : wx \le \lfloor t \rfloor\}$.

Ha $P' = \emptyset$, akkor a Farkas lemma (3.4 lemma) szerint a $0x \le -1$ azonnal megkapható (GC-levezethető) az $Ax \le b,\ wx \le \lfloor t \rfloor$ rendszerből. Tegyük fel tehát, hogy $P' \ne \emptyset$, és $F := \{x \in P' : wx = \lfloor t \rfloor\}$. A $wx \le t$ vágás választása miatt tudjuk, hogy F dimenziója kisebb, mint P dimenziója. Az indukciós feltevés szerint tehát $0x \le -1$ GC-levezethető az $Ax \le b,\ wx = \lfloor t \rfloor$ rendszerből. A 3.5 lemma szerint ekkor az $Ax \le b,\ wx \le \lfloor t \rfloor$ rendszerből valamely $c'x \le d'$ vágásnak létezik GC-levezetése, amelyre $F \cap \{x : c'x \le d'\} = \emptyset$. Következésképpen az $Ax \le b,\ wx \le \lfloor t \rfloor,\ c'x \le d'$ rendszernek a $wx \le \lfloor t \rfloor - 1$ GC-vágása.

Ezután a $P \cap \{x : wx \le \lfloor t \rfloor - 1\}$ poliéderre alkalmazva a fentieket, a P korlátossága miatt szeleteléssel egy olyan $wx \le t'$ vágás GC-levezetését kapjuk véges sok lépésben, amelyre $P \cap \{x : wx \le t'\} = \emptyset$. Ekkor a Farkas-lemma segítségével a $0x \le -1$ levezetését kapjuk. □

A 3.2. tétel bizonyítása. Tegyük fel először, hogy a P nem tartalmaz egész pontot. Ekkor a 3.3 tétel szerint a $0x \le -1$ vágásnak létezik GC-levezetése. Mivel P korlátos, az $l := \max\{wx : x \in P\}$ véges. A dualitástétel szerint tehát a $wx \le \lfloor l \rfloor$ GC-vágás. Ekkor ehhez a $0x \le -1$ alkalmas pozitív számmal való felszorzottját hozzáadva a $wx \le t$ GC-levezetését kapjuk.

Most tegyük fel, hogy P tartalmaz egész pontot. Legyen ismét $l := \max\{wx : x \in P\}$, és $P' := P \cap \{x : wx \le \lfloor l \rfloor\}$. Ha $\lfloor l \rfloor \le t$, akkor készen vagyunk, ha pedig nem, akkor mivel $wx \le t$ érvényes vágás, ezért az $F := \{x \in P' : wx = \lfloor l \rfloor\}$ valódi oldal nem tartalmaz egész pontot. A 3.3 tétel szerint ekkor a $0x \le -1$ vágásnak létezik GC-levezetése az $Ax \le b,\ wx = \lfloor l \rfloor$ rendszerből. A 3.5 lemma szerint ekkor az $Ax \le b,\ wx \le \lfloor l \rfloor$ rendszerből valamely $c'x \le d'$ vágásnak létezik GC-levezetése, amelyre $P' \cap \{x : c'x \le d', wx = \lfloor l \rfloor\} = \emptyset$. Következésképp az $Ax \le b,\ wx \le \lfloor l \rfloor,\ c'x \le \lfloor d' \rfloor$ rendszernek a $wx \le \lfloor l \rfloor - 1$ GC-vágása.

Ennek ismételgetésével a P korlátossága miatt véges sok lépésben a $wx \le t$ vágás GC-levezetését kapjuk. □

3.2. Gomory-Chvátal lezárt és Chvátal-rang

Jelölje P' azon P-beli pontok összességét, amelyek minden GC-vágást teljesítenek. Mivel végtelen sok GC-vágás van, nagyon érdekes az alábbi, Schrijvertől származó tétel.

3.6. tétel. *Ha P racionális poliéder, akkor P' is racionális poliéder.*

Bizonyítás. Feltehetjük, hogy A és b egészek. Belátjuk, hogy P'-t az $Ax \le b$ és azon

$$(yA)x \le \lfloor yb \rfloor \tag{3.1}$$

egyenlőtlenségek határozzák meg, melyekre $0 \le y < 1$ és yA egész vektor. (Ez elég, hiszen (3.1) bal oldalán korlátos együtthatók szerepelhetnek, következésképpen véges sok egyenlőtlenségről van szó.) Most belátjuk, hogy minden GC-vágás előáll az $Ax \le b$ sorainak valamely nemnegatív lineáris kombinációjának és egy (3.1) alatti egyenlőtlenségnek az összegeként.

Legyen $wx \le t$ egy GC-vágás, amely az $Ax \le b$ rendszerből az $\bar{y}$ nemnegatív vektorral vezethető le. Legyen $y' := \bar{y} - \lfloor \bar{y} \rfloor$. Legyen $w' = y'A = w - \lfloor \bar{y} \rfloor A$ és $t' := \lfloor y'b \rfloor = t - \lfloor \bar{y} \rfloor b$. Ekkor w' egész vektor, és a $w'x \le t'$, y'-vel megkapott (3.1)-beli GC-vágás és az $Ax \le b$ soraiból a nemnegatív $\lfloor \bar{y} \rfloor$ vektorral kapott $(\lfloor \bar{y} \rfloor A)x \le \lfloor \bar{y} \rfloor b$ egyenlőtlenség összegeként megkapjuk a $wx \le t$ GC-vágást. □

A P' poliédert a P poliéder *GC-lezártjának* nevezzük. Az az eldöntési probléma, hogy P' üres-e, NP-ben van, hiszen ha üres, akkor a Caratheodory tétel miatt már $n+1$ GC-vágás is mutatja hogy üres. Cornuéjols és Li megmutatták, hogy a probléma NP-teljes, sőt, egy kicsit erősebbet is bizonyítottak.

3.7. tétel. *Van olyan poliéder-osztály, amiben egész csúcsot nem tartalmazó poliéderek vannak, és NP-teljes eldönteni, hogy egy adott, az osztályba tartozó poliéder GC-lezártja üres-e.*

Jelölje $P^{(2)}$ $(P')'$-t, és $P^{(i)}$ $(P^{(i-1)})'$-t. Ekkor a 3.6 tetel szerint poliéderek egy láncát kapjuk:

$$P =: P^{(0)} \supseteq P^{(1)} \supseteq P^{(2)} \supseteq \ldots \supseteq P_I.$$

A 3.2 tételből következik, hogy P_I minden lapja GC-levezethető, tehát létezik olyan k, hogy P_I minden lapja levezethető k lépésben. Ez az alábbi tételt adja.

3.8. tétel. *Legyen $P = \{x : Ax \le b\}$ egy racionális korlátos poliéder, ekkor $P^{(k)} = P_I$ valamely k pozitív egészre.*

A legkisebb ilyen k-t nevezzük a P poliéder *Chvátal-rangjának.* A fogalom kiterjeszthető nemkorlátos racionális poliéderekre, lásd Schrijver [2]. Sőt, a következő is igaz: minden A racionális mátrixhoz létezik egy $k(A)$ szám, hogy tetszőleges racionális b esetén az $Ax \le b$ poliéder Chvátal-rangja legfeljebb $k(A)$. Tehát racionális mátrixok Chvátal-rangjáról is beszélhetünk.

Lényegében a fentiekkel ekvivalens vágásokkal a lexikografikus szimplex módszer használatára épülő egészértékű programozási algoritmust dolgozott ki Gomory az 1950-es években. Lásd Vizvári Béla jegyzetét [5] és a 5.1 fejezetet.

A gyakorlatban az érvényes vágásokat a következőképpen használhatjuk egészértékű programozási feladatok megoldására. Adott a következő feladat $\max\{cx : Ax \le b, x \in \mathbb{Z}^n\}$, és néhány vágás-család, amelyekről tudjuk, hogy érvényes vágások (például megadhatjuk általánosan a feladatosztályra vonatkozó Gomory-Chvátal-levezetéseket). Először oldjuk meg a $\max\{cx : Ax \le b, x \in \mathbb{R}^n\}$ lineáris programozási relaxált feladatot. Amennyiben a kapott x^* optimális megoldás egész, akkor megoldottuk az eredeti feladatunkat. Ha pedig nem, akkor keressünk a vágás-családjainkban egy olyan érvényes vágást, amelyet az x^* nem telesít, azaz $cx^* > d$, de $cx \le d$ minden egész megoldásra igaz. Vegyük hozzá az egyenlőtlenségrendszerünkhöz ezt a vágást, és iteráljuk az eljárást.

Ha szerencsénk van, akkor elég kitartóan alkalmazva a fenti eljárást, megoldjuk az egészértékű programozási feladatot. De ha ezt nem is győzzük kivárni, akkor is az optimális célfüggvényérték egyre jobb felső becsléséhez jutunk, amit például kiválóan tudunk használi egy korlátozás és szétválasztás módszerére épülő algoritmusban (5.2 fejezet).

3.3. A Gomory-Chvátal lezárt és a TDI leírás kapcsolata

Érdekes kapcsolat van a GC-lezárt és a 2.25. illetve 2.30. tételekben szereplő TDI leírások között.

3.9. tétel. *Legyen $Ax \leq b$ a P poliédernek egy TDI leírása, ahol $A \in \mathbb{Z}^{m\times n}$. Ekkor P GC-lezártja a $P' = \{x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor\}$ poliéder.*

Bizonyítás. Az nyilvánvaló, hogy az $Ax \leq \lfloor b \rfloor$ rendszer minden egyenlőtlensége GC-vágás, hiszen A egész mátrix. A másik irányú tartalmazáshoz tegyük fel, hogy $cx \leq d$ egy GC-vágás, azaz c egészértékű és létezik $y \in \mathbb{R}^n_+$, hogy $yA = c$ és $d = \lfloor yb \rfloor$. Mivel az $Ax \leq b$ rendszer TDI, létezik egészértékű $y' \in \mathbb{Z}^n_+$, amire $y'A = c$ és $y'b \leq yb$, következésképp $\lfloor y'b \rfloor \leq \lfloor yb \rfloor = d$. Másrészt y' egészértékűsége miatt $y'\lfloor b \rfloor \leq \lfloor y'b \rfloor$, tehát $y'\lfloor b \rfloor \leq d$. Azt kaptuk, hogy az $Ax \leq \lfloor b \rfloor$ rendszernek lineáris következménye (y' együthatókkal) a $cx \leq d$ egyenlőtlenség, és ez az összes $cx \leq d$ GC-vágásra igaz, tehát $\{x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor\} \subseteq P'$. □

A tételt a 2.25. tétellel kombinálva egyből adódik a 3.6. tétel. Másrészt könnyen kijön az alábbi összefüggés egy poliéder és egy oldala GC-lezártjai között.

3.10. lemma. *Legyen P egy racionális poliéder, és F egy oldala P-nek. Ekkor F GC-lezártja $P' \cap F$, ahol P' a P poliéder GC-lezártja.*

Bizonyítás. A 2.25. tétel szerint P-nek van TDI leírása. Mivel F oldal, ehhez hozzávehetünk egy, az oldalt meghatározó egyenletredszert, amiről feltehető, hogy az együtthatók és a jobboldal is egészek. Tehát a P poliédernek van olyan $A_1x \leq b_1,\ A_2x \leq b_2$ TDI leírása, ahol A_1, A_2 egész mátrixok, b_2 egész vektor, és $F = \{x : A_1x \leq b_1,\ A_2x = b_2\}$. A 2.31. tétel szerint az $A_1x \leq b_1,\ A_2x = b_2$ rendszer is TDI. Így a 3.9. tétel és b_2 egészértékűsége alapján az F oldal GC-lezártja $F' = \{x \in \mathbb{R}^n : A_1x \leq \lfloor b_1 \rfloor,\ A_2x = b_2\}$, ami pont $P' \cap F$. □

A lemmát egymás után többször alkalmazva azt kapjuk, hogy (az előző rész jelölését használva) $F^{(i)} = P^{(i)} \cap F$ minden i-re. Ennek segítségével egyszerű alternatív bizonyításokat adhatunk a 3.3. és a 3.2. tételre.

A 3.3. tétel bizonyítása. Elég belátni, hogy létezik k, hogy $P^{(k)} = \emptyset$, mert az üres poliéderből a Farkas lemma (3.4 lemma) szerint a $0x \leq -1$ GC-levezethető. A P poliéder dimenziója szerinti indukcióval bizonyítunk; 0 dimenzióra igaz az állítás. Legyen w egész vektor, és $wx \leq t$ egy olyan egyenlőtlenség, amelyre $P \cap \{x : wx = t\}$ a P egy valódi oldala. Tegyük fel indirekt, hogy $P^{(k)}$ nem üres semmilyen k-ra. Ekkor létezik egy legkisebb β egész szám, hogy $wx \leq \beta$ GC-levezethető; legyen k olyan, hogy $P^{(k)} \subseteq \{x : wx \leq \beta\}$. Legyen $F = P^{(k)} \cap \{x : wx = \beta\}$; tudjuk, hogy ez nemüres oldala $P^{(k)}$-nak, hiszen ha üres lenne, akkor $P^{(k+1)} \subseteq \{x : wx \leq \beta - 1\}$ teljesülne. Másrészt tudjuk, hogy $\dim(F) < \dim(P)$, hiszen $P \cap \{x : wx = t\}$ valódi oldala P-nek. Indukció szerint létezik ℓ, hogy $F^{(\ell)} = \emptyset$. A 3.10. lemma alapján ekkor $P^{(k+\ell)} \cap \{x : wx = \beta\} = \emptyset$. De ekkor $P^{(k+\ell+1)} \subseteq \{x : wx \leq \beta - 1\}$, ellentmondásban β választásával.

A 3.2. tétel bizonyítása. Tegyük fel indirekt, hogy $wx \leq t$ nem levezethető. Ekkor létezik egy legkisebb β egész szám, hogy $wx \leq \beta$ GC-levezethető; legyen k olyan, hogy $P^{(k)} \subseteq \{x : wx \leq \beta\}$. Legyen $F = P^{(k)} \cap \{x : wx = \beta\}$; tudjuk, hogy ez nemüres oldala $P^{(k)}$-nak, hiszen ha üres lenne, akkor $P^{(k+1)} \subseteq \{x : wx \leq \beta - 1\}$ teljesülne.

Mivel $\beta > t$, F-nek nincs egész pontja, és így a 3.3 tétel szerint szerint létezik ℓ, hogy $F^{(\ell)} = \emptyset$. A 3.10. lemma alapján ekkor $P^{(k+\ell)} \cap \{x : wx = \beta\} = \emptyset$. De ekkor $P^{(k+\ell+1)} \subseteq \{x : wx \leq \beta - 1\}$, ellentmondásban β választásával. □

3.4. Vágások bináris feladatokra

3.4.1. Fedési vágások

A fedési vágásokat a hátizsák-feladatra vezetjük be, de bármilyen bináris feladatra használhatók, ahol szerepel olyan egyenlőtlenség, mint a hátizsák-feladatban. Tekintsük a feladat lineáris relaxációját:

$$P = \{x \in \mathbb{R}^n : 0 \leq x \leq 1, \sum_{j=1}^{n} a_j x_j \leq b\},$$

ahol a_j és b pozitív egészek, és feltesszük, hogy $b \geq a_1 \geq a_2 \geq \cdots \geq a_n$. Jelölés: $[n] = \{1, \ldots, n\}$.

3.11. definíció. Egy $C \subseteq [n]$ halmaz *függő*, ha $\sum_{j \in C} a_j > b$. Jelölés:

$$\bar{C} = C \cup \{k \in [n] \setminus C : a_k \geq a_j \ \forall j \in C\}.$$

3.12. állítás. *A következő egy Gomory-Chvatal vágás:*

$$\sum_{j \in \bar{C}} x_j \leq |C| - 1.$$

Bizonyítás. Könnyű látni, hogy az LP relaxáció minden megoldása teljesíti a $\sum_{j \in \bar{C}} x_j < |C|$ szigorú egyenlőtlenséget. Így létezik $\epsilon > 0$, hogy az LP relaxáció minden megoldása teljesíti a $\sum_{j \in \bar{C}} x_j \leq |C| - \epsilon$ egyenlőtlenséget. □

3.13. definíció. A fenti típusú vágást **fedési vágásnak** (cover inequality) nevezzük. Fedési vágást nem csak hátizsák-feladatnál írhatunk fel, hanem minden olyan feladatnál, ahol az egyenlőtlenségek között szerepel csupa pozitív együtthatós egyenlőtlenség.

Azt akarjuk megvizsgálni, hogy milyen feltételekkel lesznek ezek a vágások lapjai a P_I poliédernek. Tudjuk, hogy P_I n-dimenziós poliéder, hiszen 0 és az egységvektorok benne vannak. Tehát egy $wx \leq t$ vágás pontosan akkor definiál lapot, ha n darab affin független P_I-beli vektor egyenlőséggel teljesíti.

Legyen C egy tartalmazásra minimális független halmaz, és a két legkisebb elemét jelölje $j_1 \leq j_2$.

3.14. lemma. *Tegyük fel hogy a következő két feltétel teljesül:*

$$j \in \bar{C} \setminus C \Rightarrow C \setminus \{j_1, j_2\} \cup \{j\} \text{ független,} \tag{3.2}$$

$$j \in [n] \setminus \bar{C} \Rightarrow C \setminus \{j_1\} \cup \{j\} \text{ független.} \tag{3.3}$$

Ekkor a $\sum_{j \in \bar{C}} x_j \leq |C| - 1$ *egyenlőtlenség a* P_I *poliéder lapját definiálja.*

Bizonyítás. A következő vektorok affin függetlenek a feltételek teljesülése esetén:

- $\chi_{C \setminus \{j\}}$, $j \in C$,
- $\chi_{C \setminus \{j_1, j_2\} \cup \{j\}}$, $j \in \bar{C} \setminus C$,
- $\chi_{C \setminus \{j_1\} \cup \{j\}}$, $j \in [n] \setminus \bar{C}$.

A feltételek szerint ezek P_I-ben vannak. □

3.4.2. Klikk-vágások

Bináris feladatnál literálnak nevezünk egy változót vagy a negáltját. Két literál **tiltott párt alkot**, ha van olyan egyenlőtlenség a rendszerben, ami miatt nem lehet mindkettő igaz. Például $3x_1-2x_2+x_3 \leq 2$ esetén x_1 és $\neg x_2$ tiltott párt alkot. Tekintsük azt a gráfot, aminek csúcsai a literálok, élei pedig a tiltott párok. Ha K ennek a gráfnak egy tartalmazásra maximális klikkje, akkor a

$$\sum_{x_i \in K} x_i + \sum_{\neg x_i \in K} (1 - x_i) \leq 1$$

érvényes vágást **klikk-vágásnak** nevezzük.

Megjegyezzük, hogy a fedési vágásokkal szemben a klikk-vágások nem feltétlenül elsőfajú Gomory-Chvátal vágások.

3.5. A diszjunktív elv

Adott $P_1, P_2 \subseteq \mathbb{R}^n_+$ konvex halmazok, valamint $c^i x \geq c^i_0$ érvényes egyenlőtlenség a P_i halmazra, $i = 1, 2$. [Azaz $\forall \bar{x} \in P_i \;:\; c^i \bar{x} \geq c^i_0$, $i = 1, 2$.] Definiáljuk a következő egyenlőtlenséget: $cx \geq d$, ahol

$$c_j = \max\{c^1_j, c^2_j\},\ j = 1, \ldots, n, \quad c_0 = \min\{c^1_0, c^2_0\}.$$

3.15. állítás. *$cx \geq c_0$ érvényes a $\operatorname{conv}(P_1 \cup P_2)$ halmazra.*

Biz Legyen $\bar{x} \in P_1 \cup P_2$. Ekkor $\bar{x} \in P_i$ valamely $i \in \{1, 2\}$-re. Tehát $c^i \bar{x} \geq c^i_0$, mivel $c^i x \geq c^i_0$ érvényes egyenlőtlenség a P_i halmazra. Továbbá $c\bar{x} \geq c^i \bar{x}$, mivel $\bar{x} \in P_i \subseteq \mathbb{R}^n_+$, és $c \geq c^i$. Végül $c^i_0 \geq c_0$ miatt kapjuk, hogy $c\bar{x} \geq c_0$.

Ha $\bar{x} \in \operatorname{conv}(P_1 \cup P_2) \setminus (P_1 \cup P_2)$, akkor $\bar{x} = \lambda \bar{x}^1 + (1 - \lambda)\bar{x}^2$ valamely $0 < \lambda < 1$ valós számra, és $\bar{x}^i \in P_i$, $i = 1, 2$, vektorokra. Ekkor

$$c\bar{x} = c(\lambda \bar{x}^1 + (1 - \lambda)\bar{x}^2) \geq \lambda c^1 \bar{x}^1 + (1 - \lambda) c^2 \bar{x}^2 \geq \lambda c^1_0 + (1 - \lambda) c^2_0 \geq c_0.$$

□

A $cx \geq c_0$ egyenlőtlenséget *diszjunktív egyenlőtlenségnek* hívjuk. Az elnevezés eredete az, hogy minden olyan vektor teljesíti, ami két konvex halmaz uniójába esik.

3.6. Gomory egészértékű vágás

Tekintsük a következő egészértékű programozási feladatot.

$$\max cx \tag{3.4}$$

$$Ax = b$$

$$x \geq 0,$$

$$x \text{ egész.} \tag{3.5}$$

Amennyiben nem kötjük ki (3.5)-ot, akkor a feladat *lineáris programozási relaxáltjáról* beszélünk.

Tegyük fel, hogy a megoldott lineáris porgramozási relaxált egy optimális bázismegoldása x^*, és jelölje N a bázisban nem szereplő változók indexhalmazát. Könnyen látható, hogy ekkor

$$\sum_{i \in N} x_i \geq 1$$

érvényes vágás, azaz minden egész megoldás teljesíti. Viszont x^* nem. Algoritmikus szempontból azonban ez a vágás nem túl jó, ezért Gomory egy másfajta érvényes vágást vezetett be, aminek segítségével véges algoritmust is adott, lásd a 5.1. Fejezetet.

A Gomory-féle érvényes vágáshoz a következőképpen jutunk. Legyen a lineáris programozási relaxált egy optimális bázismegoldását mutató egyenletrendszer a következő:

$$x_B + B^{-1}A_N x_N = B^{-1}b.$$

Ennek a felírásnak az együtthatóira használjuk a megszokott jelöléseket: $d_{ij} := (B^{-1}A_j)_i$ és $d_{i0} := (B^{-1}b)_i$. Legyen t a legkisebb olyan index, hogy az optimális bázismegoldásban a t-edik változó nem egész érték. Ekkor a hozzá tartozó egyenlet:

$$x_t + \sum_{j\in N} d_{tj}x_j = d_{t0}. \tag{3.6}$$

Ebből következik, hogy

$$x_t + \sum_{j\in N} \lfloor d_{tj}\rfloor x_j \le x_t + \sum_{j\in N} d_{tj}x_j = d_{t0},$$

mivel $x_j \ge 0$ teljesül minden j-re. Ebből az következik az egészértékű x_j megoldásokra, hogy

$$x_t + \sum_{j\in N} \lfloor d_{tj}\rfloor x_j \le \lfloor d_{t0}\rfloor. \tag{3.7}$$

Ezt az egyenlőtlenséget tehát a feladatunk minden egész megoldása teljesíti, de a lineáris programozási relaxált aktuális optimális megoldása nem. Ugyanis a bázismegoldásban $x_t = d_{t0}$ és $x_j = 0$ miden $j \in N$ indexre, tehát a balodal értéke d_{t0}, ami nagyobb, mint a jobboldal, hiszen d_{t0} nem egész a feltevésünk szerint. A (3.7) egyenlőtlenségből levolna a (3.6 regyenletet a következő ekvialensen egyenlőtlenséget kapjuk:

$$\sum_{j\in N} \{d_{tj}\}x_j \ge \{d_{t0}\}, \tag{3.8}$$

ahol $\{r\} = r - \lfloor r\rfloor$. Utóbbi egyenlőtlenséget *Gomory-vágásnak* nevezzük.

3.7. Gomory-féle vegyes vágás

Vegyes programozási feladatnál a fentiekben szereplő vágások nem használhatók, mert nem feltétlenül érvényesek. Az alábbiakban bemutatunk egy szintén Gomory nevéhez fűződő, kicsit bonyolultabb vágást, ami a vegyes esetre is működik. Az érvényességet kétféle módon is igazoljuk, az első Gomory eredeti levezetésén alapul, a második pedig a diszjunktív elven.

Tekinstsük a következő vegyes programozási feladatot:

$$\begin{aligned}
&\max cx\\
&Ax = b\\
&x \ge 0\\
&x_i \in \mathbb{Z}, \quad i \in I.
\end{aligned}$$

ahol I az egészértékű változók indexhalmaza, és J a többi változóé, tehát amikre nincs egészértékűségi megkötés.

3.7.1. Első levezetés

Ahogy az előző fejezetben, nézzünk egy x^* optimális bázismegoldást. Legyen $t \in I$ a legkisebb olyan I-beli index, hogy $x_t^* \notin \mathbb{Z}$. Ekkor a hozzá tartozó egyenlet:

$$x_t + \sum_{j \in N} d_{tj} x_j = d_{t0}.$$

Vezessük be az $f_j = d_{tj} - \lfloor d_{tj} \rfloor$ jelölést ($j \in N \cap I$ illetve $j = 0$).

3.16. tétel. *A vegyes programozási feladat minden megoldása teljesíti a következő egyenlőtlenséget:*

$$\sum_{j \in N \cap I, f_j \le f_0} \frac{f_j}{f_0} x_j + \sum_{j \in N \cap I, f_j > f_0} \frac{1 - f_j}{1 - f_0} x_j + \sum_{j \in N \cap J, d_{tj} \ge 0} \frac{d_{tj}}{f_0} x_j + \sum_{j \in N \cap J, d_{tj} < 0} \frac{-d_{tj}}{1 - f_0} x_j \ge 1.$$

Bizonyítás. Legyen x egy tetszőleges megoldása a vegyes programozási feladatnak. Definiáljuk az α számot a következőképpen:

$$\alpha = x_t + \sum_{j \in N \cap I, f_j \le f_0} \lfloor d_{tj} \rfloor x_j + \sum_{j \in N \cap I, f_j > f_0} \lceil d_{tj} \rceil x_j.$$

Mivel x I-be tartozó koordinátái egészek, α egész szám. Az $x_t + \sum_{j \in N} d_{tj} x_j = d_{t0}$ egyenletből következik, hogy

$$\sum_{j \in N \cap I, f_j \le f_0} f_j x_j + \sum_{j \in N \cap I, f_j > f_0} (f_j - 1) x_j + \sum_{j \in N \cap J, d_{tj} \ge 0} d_{tj} x_j + \sum_{j \in N \cap J, d_{tj} < 0} d_{tj} x_j = d_{t0} - \alpha.$$

Két esetet különböztetünk meg aszerint, hogy α és d_{t0} hogy viszonyul egymáshoz.

Első eset: $\alpha \le \lfloor d_{t0} \rfloor$. Ekkor egyrészt nyilvánvalóan

$$\sum_{j \in N \cap I, f_j > f_0} (f_j - 1) x_j + \sum_{j \in N \cap J, d_{tj} < 0} d_{tj} x_j \le 0,$$

másrészt

$$\sum_{j \in N \cap I, f_j \le f_0} f_j x_j + \sum_{j \in N \cap J, d_{tj} \ge 0} d_{tj} x_j \ge d_{t0} - \alpha \ge d_{t0} - \lfloor d_{t0} \rfloor = f_0.$$

Ebből a kettőből már következik a tételbeli egyenlőtlenség, az elsőt $\frac{1}{f_0 - 1}$, a másodikat $\frac{1}{f_0}$ szorzóval összeadva.

Második eset: $\alpha \ge \lceil d_{t0} \rceil$. Ekkor egyrészt nyilvánvalóan

$$\sum_{j \in N \cap I, f_j \le f_0} f_j x_j + \sum_{j \in N \cap J, d_{tj} \ge 0} d_{tj} x_j \ge 0,$$

másrészt

$$\sum_{j \in N \cap I, f_j > f_0} (f_j - 1) x_j + \sum_{j \in N \cap J, d_{tj} < 0} d_{tj} x_j \le d_{t0} - \alpha \le d_{t0} - \lceil d_{t0} \rceil = f_0 - 1.$$

Ebből a kettőből már következik a tételbeli egyenlőtlenség, az elsőt $\frac{1}{f_0}$, a másodikat $\frac{1}{f_0 - 1}$ szorzóval összeadva. $\square$

Könnyű ellenőrizni, hogy maga az x^* vektor nem teljesíti a tételbeli egyenlőtlenséget, hiszen $j \in N$ esetén $x_j^* = 0$. Így az egyenlőtlenség használható a vágósíkos eljárásban. Megjegyzendő, hogy egészértékű feladat esetén ez a vágás nem feltétlenül a Gomory-féle egészértékű vágást adja vissza.

3.7.2. Második levezetés

A második levezetés a diszjunktív elvet használja. Először egy gyengébb egyenlőtlenséget vezetünk le a módszer bemutatása érdekében, majd pedig új bizonyítást adunk a Gomory vegyes vágás érvényességének igazolására.

3.17. lemma. *Legyen $f_0 = d_{i0} - \lfloor d_{i0} \rfloor$. Az (3.9) egyenlőtlenséget az egészértékű program minden megengedett megoldása teljesíti, de a B bázishoz tartozó bázismegoldás nem.*

$$\sum_{j \in N : d_{ij} \geq 0} \frac{d_{ij}}{f_0} x_j + \sum_{j \in N : d_{ij} < 0} \frac{-d_{ij}}{1 - f_0} x_j \geq 1. \tag{3.9}$$

Biz Definiáljunk két konvex halmazt:

$$\begin{aligned} P_1 &= \{x \in \mathbb{R}^n \ : \ Ax = b, x \geq 0, x_i \leq \lfloor d_{i_0} \rfloor\} \\ P_2 &= \{x \in \mathbb{R}^n \ : \ Ax = b, x \geq 0, x_i \geq \lceil d_{i_0} \rceil\} \end{aligned}$$

Tetszőleges $\bar{x}$ megengedett megoldására az egészértékű programnak teljesül, hogy $\bar{x} \in P_1 \cup P_2$. Most levezetünk egy-egy érvényes egyenlőtlenséget a P_1 és P_2 halmazokra. Nézzük P_1-et: mivel $\bar{x}_i \leq \lfloor d_{i_0} \rfloor$ miden $\bar{x} \in P_1$ vektorra, a szimplex tábla x_i változóhoz tartozó sorát használva kapjuk, hogy $x_i \leq \lfloor d_{i0} \rfloor$ akkor és csak akkor, ha $d_{i0} - \sum_{j \in N} d_{ij} x_j \leq \lfloor d_{i0} \rfloor$. Átrendezéssel kapjuk a

$$\sum_{j \in N} d_{ij} x_j \geq d_{i0} - \lfloor d_{i0} \rfloor = f_0$$

egyenlőtlenséget, ami érvényes P_1-re. Tovább alakítva kapjuk az ezzel ekvivalens

$$\sum_{j \in N} \frac{d_{ij}}{f_0} x_j \geq 1 \tag{3.10}$$

egyenlőtlenséget. Hasonlóan levezethetjük a

$$\sum_{j \in N} \frac{-d_{ij}}{1 - f_0} x_j \geq 1 \tag{3.11}$$

érvényes egyenlőtlenséget P_2-re. Alkalmazva az 3.15. Propozíciót a (3.10) és (3.11) egyenlőtlenségekre (a feltételei teljesülnek!) kapjuk az

$$\sum_{j \in N} \max\left\{\frac{d_{ij}}{f_0}, \frac{-d_{ij}}{1 - f_0}\right\} x_j \geq 1$$

egyenlőtlenséget, ami érvényes $\mathrm{conv}(P_1 \cup P_2)$-re. Vegyük észre, hogy $\bar{a}_{ij}$ előjele határozza meg a maximumot, azaz

$$\max\left\{\frac{d_{ij}}{f_0}, \frac{-d_{ij}}{1 - f_0}\right\} = \begin{cases} \frac{d_{ij}}{f_0}, & \text{ha } d_{ij} \geq 0 \\ \frac{-d_{ij}}{1 - f_0} & \text{ha } d_{ij} < 0 \end{cases}$$

Ezt behelyettesítve (3.9)-et kapjuk.

Végül vegyük észre, hogy a B bázishoz tartozó x^* bázismegoldásban $x_j^* = 0$ minden $j \in N$ indexre, ezért (3.9) levágja a bázismegoldást. □

Most rátérünk a Gomory vegyes vágás levezetésére. Az alapötlet az, hogy ügyesen válasszuk meg a diszjunkciót.

Legyen $\bar{m} \in \mathbb{Z}^N$ olyan vektor, amit a nem-bázis oszlopok indexelnek, és $\bar{m}_j = 0$, ha $j \in N \setminus (N \cap I)$, és $\bar{m}_j$ egy később megválasztandó egész szám, ha $j \in N \cap I$. Cálunk a nem-bázis egész változók együtthatóinak csökkentése, az $\bar{m}_j$ értéekek alkalmas megválasztásával. Tekintsük az $x_i + \bar{m}x_N \leq \lfloor d_{ij} \rfloor$ és $x_i + \bar{m}x_N \geq \lceil d_{ij} \rceil$ egyenlőtlenségeket. Definálunk két konvex halmazt:

$$P_1(\bar{m}) = \{x \in \mathbb{R}^n \ : \ Ax = b, x \geq 0, x_i + \bar{m}x_N \leq \lfloor d_{i_0} \rfloor\}$$
$$P_2(\bar{m}) = \{x \in \mathbb{R}^n \ : \ Ax = b, x \geq 0, x_i + \bar{m}x_N \geq \lceil d_{i_0} \rceil\}$$

A $P_1(\bar{m})$ és $P_2(\bar{m})$ halmazok függenek $\bar{m}$ választásától, de azt feltesszük, hogy $\bar{m}$ egész vektor, és $\bar{m}_j \neq 0$ csak akkor, ha $j \in N \cap I$.

3.18. állítás. *Minden $\bar{x}$ megengedett megoldására a vegyes egészértékű programnak teljesül, hogy $\bar{x} \in P_1(\bar{m}) \cup P_2(\bar{m})$.*

A bizonyítást az olvasóra bízzuk.

Legyen $N_I = N \cap I$ a nem-bázis egész változók indexhalmaza, $N_J = N \setminus N_I$ a nem-bázis folytonos változók indexhalmaza.

3.19. lemma. *Legyen $f_0 = d_{i0} - \lfloor d_{i0} \rfloor$, és $f_j = d_{ij} - \lfloor d_{ij} \rfloor$, $j \in N$. A (3.12) egyenlőtlenséget az egészértékű program minden megengedett megoldása teljesíti, de a B bázishoz tartozó bázismegoldás nem.*

$$\sum_{j \in N_I : f_j \leq f_0} \frac{f_j}{f_0} x_j + \sum_{j \in N_I : f_j > f_0} \frac{1 - f_j}{1 - f_0} x_j + \sum_{j \in N_J : d_{ij} \geq 0} \frac{d_{ij}}{f_0} x_j + \sum_{j \in N_J : d_{ij} < 0} \frac{-d_{ij}}{1 - f_0} x_j \geq 1. \tag{3.12}$$

Biz Az (3.9) diszjunktív vágás levezetése során megmutattuk, hogy (3.10) és (3.11) érvényes egyenlőtlenségek a P_1 és P_2 halmazokra. A levezetés lépéseit követve kapjuk, hogy

$$\sum_{j \in N} \frac{d_{ij} - \bar{m}_j}{f_0} x_j \geq 1 \tag{3.13}$$

$$\sum_{j \in N} \frac{-(d_{ij} - \bar{m}_j)}{1 - f_0} x_j \geq 1 \tag{3.14}$$

érvényes egyenlőtlenségek a $P_1(\bar{m})$ és $P_2(\bar{m})$ halmazokra. Csak a $j \in N_I$ indexű változókkal foglalkozunk, mivel $\bar{m}_j = 0$ ha $j \in N_J$. Hogyan válasszuk meg $\bar{m}_j$ értékét, hogy x_j $(j \in N_I)$ együtthatója a legkisebb legyen? x_j együtthatója a diszjunktív vágásban:

$$\max\left\{\frac{d_{ij} - \bar{m}_j}{f_0}, \frac{-(d_{ij} - \bar{m}_j)}{1 - f_0}\right\}.$$

Legyen $\bar{m}_j := \lfloor d_{ij} \rfloor$, ha $f_j \leq f_0$, illetve $\bar{m}_j := \lceil d_{ij} \rceil$, ha $f_j > f_0$. Ezen választás mellet x_j együtthatója a diszjunktív vágásban:

$$\max\left\{\frac{f_j}{f_0}, \frac{-f_j}{1 - f_0}\right\} = \frac{f_j}{f_0} \qquad \text{ha } f_j \leq f_0.$$

$$\max\left\{\frac{f_j - 1}{f_0}, \frac{-(f_j - 1)}{1 - f_0}\right\} = \frac{1 - f_j}{1 - f_0} \qquad \text{ha } f_j > f_0.$$

Ezeket a kifejezéseket felhasználva kapjuk a (3.12) vágást. Vegyük észre, ha m_j értékét a fentinél kisebbnek, vagy nagyobbnak választanánk, azzal az együtthatók csak növekednének.

A bizonyítás hátralevő része ugyanaz, mint a 3.17. lemmáé. □

3.8. Dominancia

Azt mondjuk, hogy az $\alpha x \geq \alpha_0$ vágás *dominálja* a $\beta x \geq \beta_0$ vágást, ha $\alpha_j \leq \beta_j$ minden $j \in \{1, \ldots, n\}$-re, és $\alpha_0 \geq \beta_0$.

3.20. állítás. *A Gomory vegyes egészértékű vágás (3.12) dominálja az (3.9) diszjunktív vágást.*

A bizonyítást az olvasóra bízzuk.

A dominancia fogalom haszna a következő.

3.21. lemma. *Ha $\alpha x \geq \alpha_0$ és $\beta x \geq \beta_0$ érvényes egyenlőtlenségek P_I-re, és $\alpha x \geq \alpha_0$ dominálja $\beta x \geq \beta_0$ egyenlőtlenséget, akkor*

$$P \cap \{x \ : \ \alpha x \geq \alpha_0\} \subseteq P \cap \{x \ : \ \beta x \geq \beta_0\}.$$

Biz Legyen $\bar{x} \in P \cap \{x \ : \ \alpha x \geq \alpha_0\}$, ekkor

$$\beta \bar{x} \geq \alpha \bar{x} \geq \alpha_0 \geq \beta_0,$$

ahol az első és harmadik egyenlőtlenség a dominanciából, és $\bar{x} \geq 0$-ból következik, a második pedig $\bar{x}$ választásából. □

Geometriailag a lemma azt jelenti, hogy $\alpha x \geq \alpha_0$ legalább annyit vág le a P poliéderből, mint $\beta x \geq \beta_0$.

3.9. Metszet vágások

Legyen $K \subset \mathbb{R}^n$ konvex poliéder (amely ráadásul valódi részhalmaza $\mathbb{R}^n$-nek). A K poliéder *belseje*, amit $int(K)$-val jelölünk, azoknak a pontoknak a halmaza, amelyek K-hoz tartozank, de nem K lapjain vannak.

Legyen B egy megengedett bázisa az LP relaxációnak, x_i egész változó, amely tört értéket vesz fel a bázismegoldásban, azaz $i \in B \cap I$, és d_{i0} nem egész. A B bázishoz tartozó bázismegoldás $x^* = \begin{pmatrix} x_B^* \\ x_N^* \end{pmatrix} = \begin{pmatrix} d_0 \\ 0_n \end{pmatrix}$. Tekintsük a B bázishoz tartozó szimplex táblát:

$$x_B + \sum_{j \in N} d_j x_j = d_0.$$

A P_B *bázispoliéder*:

$$P_B = \{x \in \mathbb{R}^n \ : \ x_B + \sum_{j \in N} d_j x_j = d_0, x_N \geq 0\}.$$

Vegyük észre, hogy x_B komponensei lehetnek negatívak is, de $x_N \geq 0$ minden $x \in P_B$ vektorra, és ez az egy különbség P és P_B között, azaz

3.22. állítás. $P \subset P_B$.

Biz $\bar{x} \in P$ akkor és csak akkor, ha $\bar{x} \geq 0$, és $A\bar{x} = b$, akkor és csak akkor, ha $\bar{x} \geq 0$, és $(A_B)^{-1} A\bar{x} = (A_B)^{-1} b$, vagy ekvivalensen, $\bar{x}_B + \sum_{j \in N} d_j \bar{x}_j = d_0$. Tehát az $x_B \geq 0$ elhagyásával P-nél bővebb poliédert kapunk. □

Tegyük fel, hogy A mátrixnak m sora van, és hogy a rangja is m. Megmutatjuk, hogy $P_B \subseteq R^n$ egy $n - m$ dimenziós poliéder kúp. Mivel P_B poliédert n egyenlőtlenség írja le, amelyek között m egyenlet van, azt kapjuk, hogy P_B dimenziója legfeljebb $n - m$. Legyen

$$r^k = \begin{pmatrix} -d_k \\ e_k \end{pmatrix}, \quad k \in N$$

ahol $e_k \in \mathbb{R}^N$ egységvektor, amelyet a nem-bázis változókkak indexelünk, és e_k vektor k indexű koordinátája 1, a többi 0.

3.23. állítás. *A P_B poliédernek $n-m$ darab 1-dimenziós oldala van, amelyeket az r^k vektorok generálnak. A P_B 1-dimenziós oldalai:*

$$R_k = \left\{ x \in \mathbb{R}^n \ : \ \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} d_0 \\ 0 \end{pmatrix} + \lambda r^k, \lambda \geq 0 \right\}, \quad k \in N.$$

Biz A P_B 1-diemenziós oldalait úgy kapjuk, hogy egy kivételével minden definiáló egyenlőtlenséget egyenlőségre változtatunk. Tehát legyen $L_k = \{x \in P_B \ : \ x_j = 0 \text{ minden } j \in N \setminus \{k\}\}$ egy 1-dimenziós oldal.

Megmutatjuk, hogy $L_k = R_k$. Először megmutatjuk, hogy $R_k \subseteq P_B$. Legyen $\bar{x} \in R_k$, azaz van olyan $\lambda \geq 0$, melyre

$$\begin{pmatrix} \bar{x}_B \\ \bar{x}_N \end{pmatrix} = \begin{pmatrix} d_0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} -d_k \\ e_k \end{pmatrix} \tag{3.15}$$

Helyettesítsük be $\bar{x}$-et a P_B-t definiáló egyenletekbe:

$$\bar{x}_B + \sum_{j \in N} d_j \bar{x}_j = (d_0 + \lambda(-d_k)) + \sum_{j \in N \setminus \{k\}} d_j \cdot 0 + \lambda d_k = d_0,$$

ahol az első egyenlet az (3.15) definícióból következik, a második pedig triviális. Mivel $\bar{x}_N \geq 0$, $\bar{x} \in P_B$ teljesül.

Végül vegyük észre, hogy minden $\bar{x} \in R_k$ vektorra $\bar{x}_j = 0$, ha $j \in N \setminus \{k\}$. Tehát $R_k \subseteq L_k$. Mivel L_k 1-dimenziós, ezért $L_k = R_k$ következik. □

Az R_k, $k \in N$, halmazokat a *bázismegoldásból kiinduló sugaraknak* nevezzük.

3.24. állítás.

$$P_B = \left\{ \bar{x} \in \mathbb{R}^n \ : \ \exists \lambda_j \geq 0, j \in N, \ \text{ melyre } \bar{x} = \begin{pmatrix} d_0 \\ 0 \end{pmatrix} + \sum_{j \in N} r^j \lambda_j \right\}.$$

Biz A 3.23. Állítás szerint az egyenlőség jobb oldalán definiált halmaz része P_B-nek. Tekintsünk tetszőleges $\bar{x} \in P_B$ pontot. Ekkor $\bar{x}_B + \sum_{j \in N} d_j \bar{x}_j = d_0$ teljesül, és ebből következik, hogy $\bar{x}_B = d_0 - \sum_{j \in N} d_j \bar{x}_j$. Legyen $\lambda_j = \bar{x}_j$ minden $j \in N$ indexre. Ekkor a jobboldali halmaz tartalmazza a

$$\begin{pmatrix} d_0 - \sum_{j \in N} d_j \bar{x}_j \\ \bar{x}_N \end{pmatrix}$$

pontot, ami éppen $\bar{x}$. □

3.25. definíció. Adott $P \subseteq \mathbb{R}^n$ konvex poliéder, és $I \subseteq \{1, \ldots, n\}$ indexhalmaz. Ekkor $Z(P, I)$ azon P beli x pontok halmaza, amelyekre $x_j \in \mathbb{Z}$ teljesül minden $j \in I$ indexre. Továbbá $P_I := \overline{\text{conv}}(Z(P, I))$.

3.26. definíció. Legyen K konvex halmaz, amely a *belsejében tartalmazza* a B bázishoz tartozó x^* bázismegoldást, de K belseje az egészértékű program egyetlen megengedett megoldását sem tartalmazza, azaz $int(K) \cap Z(P, I) = \emptyset$. Legyen $\lambda_k \geq 0$, $k \in N$, az a szám, melyre $x^* + \lambda_k r^k$ a K poliéder határán van, feltéve, hogy R_k halmaznak van közös pontja a K határával, különben $\lambda_k = \infty$. Legyen $\alpha_k = 1/\lambda_k$. $\alpha_k = 0$, ha $\lambda_k = \infty$. A

$$\sum_{j \in N} \alpha_j x_j \geq 1 \tag{3.16}$$

egyenlőtlenséget *metszet vágásnak* nevezzük.

Számljuk ki a metszet vágást egy speciálisan választott konvex halmaz esetén. Legyen $K = \{x \in \mathbb{R}^n \quad \lfloor d_{i0} \rfloor \le x_i \le \lceil d_{i0} \rceil\}$. Ekkor K konvex poliéder, és a belsejében tartalmazza az x^* bázismegoldást, hiszen d_{i0} nem egész a kiinduló feltételünk szerint. Továbbá K nem tartalmaz egyetlen megengedett egész megoldást a belsejében, legfeljebb a határoló lapjain.

Határozzuk meg a λ_k értékeket:

- Ha $d_{ik} = 0$, akkor $\lambda_k = \infty$.
- Ha $d_{ik} > 0$, akkor a $x^* + \lambda r^k$, $\lambda \ge 0$, félegyenes metszi K határát, és λ_k kielégíti az $d_{i0} + \lambda_k(-d_{ik}) = \lfloor d_{i0} \rfloor$ egyenletet, azaz $\lambda_k = \frac{d_{i0} - \lfloor d_{i0} \rfloor}{d_{ik}} > 0$.
- Ha $d_{ik} < 0$, akkor a $x^* + \lambda r^k$, $\lambda \ge 0$, félegyenes metszi K határát, és λ_k kielégíti az $d_{i0} + \lambda_k(-d_{ik}) = \lceil d_{i0} \rceil$ egyenletet, azaz $\lambda_k = \frac{\lceil d_{i0} \rceil - d_{i0}}{-d_{ik}} > 0$.

Legen $N^+ = \{j \in N \ : \ \lambda_j < \infty\}$. Legyen $\alpha_k = 1/\lambda_k$ minden $k \in N^+$-re, és $\alpha_k = 0$, ha $k \in N \setminus N^+$.

3.27. lemma. *A K halmaz fenti választása mellett a $\sum_{j \in N} \alpha_j x_j \ge 1$ metszet vágás érvényes a P_I poliéderre.*

Biz Írjuk ki részletesen a kapott metszett vágást:

$$\sum_{j \in N \ : \ d_{ij} \ge 0} \frac{d_{ij}}{d_{i0} - \lfloor d_{i0} \rfloor} x_j + \sum_{j \in N \ : \ d_{ij} < 0} \frac{-d_{ij}}{\lceil d_{i0} \rceil - d_{i0}} x_j \ge 1$$

Vegyük észre, hogy az egyenlőtlenség ugyanaz, mint az (3.9) diszjunktív vágás. $\square$

A fenti bizonyítás azt is igazolja egyben, hogy az egyszerű diszjunktív vágás egy speciális metszet vágás.

Most nézzük az általános esetet.

3.28. lemma. *Legyen K konvex halmaz, amire teljesül, hogy (i) belsejében tartalmazza az aktuális bázismegoldást, és (ii) a belseje nem tarlmazza a vegyes-egészértékű program egyetlen megengedett megoldását sem. Ekkor a (3.16) metszet vágás érvényes P_I-re.*

Biz Megmutatjuk, hogy tetszőleges $x' \in P$ vektorra, ha $\sum_{j \in N} \alpha_j x'_j < 1$ teljesül, akkor $x' \in int(K)$. Mivel $int(K) \cap Z(P, I) = \emptyset$ a K választása szerint, ezért $x' \notin Z(P, I)$.

Legyen $x' \in P$ olyan, hogy $\sum_{j \in N} \alpha_j x'_j < 1$. Ekkor léteznek egyértelműen $\delta_j \ge 0$, $j \in N$, együtthatók, amelyekre $x' = x^* + \sum_{j \in N} \delta_j r^j$, mivel $x^* = \begin{pmatrix} x^*_B \\ x^*_N \end{pmatrix} = \begin{pmatrix} d_0 \\ 0 \end{pmatrix}$, $P \subset P_B$, és az r^j vektorok lineárisan függetlenek. Ezt felhasználva, és r^j definíciója alapján kapjuk, hogy $x'_j = \delta_j$, $j \in N$.

Vegyük észre, hogy $x^* + \lambda_j r^j \notin int(K)$, a λ_j választása miatt, de $x^* + \lambda r^j \in int(K)$ minden $0 \le \lambda < \lambda_j$ -re. Tehát ha $\beta_j \ge 0$, $j \in N^+$, olyan együtthatók, amelyekre $\sum_{j \in N^+} \beta_j < 1$, akkor $x^* + \sum_{j \in N^+} \beta_j(\lambda_j r^j) \in int(K)$.

Következésképpen, ha $\sum_{j \in N^+} x'_j/\lambda_j < 1$, akkor $\beta_j := x'_j/\lambda_j$ választással kapjuk, hogy $x^* + \sum_{j \in N^+} \beta_j(\lambda_j r^j) \in int(K)$. Mivel $\beta_j(\lambda_j r^j) = (x'_j/\lambda_j)(\lambda_j r^j) = x'_j r^j$, kapjuk, hogy $x' = x^* + \sum_{j \in N^+} x'_j r^j \in int(K)$.

Végül vegyük észre, hogy $\sum_{j \in N^+} x'_j/\lambda_j < 1$ ekvivalens az $\sum_{j \in N} \alpha_j x'_j < 1$ egyenlőtlenséggel. $\square$

3.10. Szeparáció és optimalizálás

Ebben a fejezetben röviden ismertetjük a Grötschel, Lovász, Schrijver szerzőhármastól 1986-ból származó eredményt a szeparáció és optimalizálás ekvivalenciájáról. Hacsijan 1979-ben közzétette az ellipszoid-módszerként ismeretes polinomiális algoritmusát a lineáris programozási feladatra. Ennek a módszernek a következménye Grötschel et al. tétele, amelynek elméleti jelentősége bizonyos problémák polinomiális megoldhatóságának poliéderes eszközökkel való bizonyítása.

Tekintsük a következő feladatot. Adott egy $G = (V, E)$ gráf, két kitüntetett s és t pontja, és az élein egy nemnegatív c súlyfüggvény. Keressük meg a minimális súlyú s-t-vágást, azaz azt az élhalmazt, amelynek a gráfból való törlésével s és t különböző komponensekbe kerül.

Ezen feladat lp-relaxáltja a következő.

$$\min \sum_{e \in E} c_e x_e$$

$$\begin{aligned} \sum_{e \in K} x_e &\geq 1 \quad \forall\, K \in \mathcal{K} \\ 0 \leq x_e &\leq 1 \quad \forall\, e \in E \end{aligned}$$

ahol $\mathcal{K}$ az összes s-ből t-be vezető út halmaza. Könnyen látható, hogy a 0-1-megoldások éppen az s-t-vágások, és az is világos Ford és Fulkerson maximális folyamokra vonatkozó tételéből, hogy a fenti poliéder minden csúcsa egész.

Igaz ugyan, hogy az ellipszoid-módszer polinomidőben megoldaná a fenti feladatot, csakhogy a gráfhoz képest a feladat mérete exponenciális, tekintve, hogy rengeteg s-ből t-be vezető út létezhet. Ám ha adott egy $0 \leq x^* \leq 1$ vektor, akkor annak ellenőrzése a feltételek nagy számát tekintve is gyorsan megy, hogy teljesíti-e az összeset. Nevezetesen az x^*_e nemnegatív élsúlyok mellett csak azt kell ellenőriznünk, hogy a legrövidebb s-ből t-be vezető út hosszabb-e 1-nél. ha igen, akkor minden feltételt teljesít x^*, ha pedig nem akkor a kezünkben van egy feltétel, azaz egy (legrövidebb) út, amelyhez tartozó feltételt megsérti x^*.

Ennek fényében nagyon érdekes lesz a megfelelően értelmezett poliéderosztályokon a szeparáció és optimalizálás ekvivalenciáját kimondó igen mély tétel, amit itt nem bizonyítunk.

Általánosan a *szeparációs probléma* a következő. Amennyiben adott a $P \subseteq \mathbb{R}^n$ korlátos racionális poliéder és egy racionális $v \in \mathbb{R}^n$, akkor döntsük el, hogy a v pont benne van-e P-ben, ha pedig nem, akkor adjunk meg egy racionális $c \in \mathbb{R}^n$ vektort, amelyre $cx < cv$ teljesül a P minden x pontjára.

Az *optimalizálási probléma* pedig így fest. Adott a $P \subseteq \mathbb{R}^n$ korlátos racionális poliéder és egy racionális $c \in \mathbb{R}^n$ (célfüggvény) vektor. Döntsük el, hogy a P poliéder üres-e, és ha nem, akkor adjunk meg egy $x^* \in P$ pontot, melyre $cx \leq cx^*$ teljesül a P minden x pontjára.

A fent említett tétel szerint ezen két feladat polinomiálisan ekvivalens a megfelelően értelmezett feladatosztályokon. A feladatosztályok korrekt értelmezése helyett példaként megemlítjük, hogy a gráfok teljes párosítási poliéderét tekinthetjük egy poliéder osztálynak, amin tehát ekvivalens az optimalizálás és a szeparálás. Mivel ismert az Edmondstól származó minimális súlyú teljes párosítási algoritmus, az optimalizáció polinomiális. Grötschel et al. tétele szerint tehát létezik polinomiális algoritmus a szeparációs problémára is.

Míg Edmonds algoritmusa a hatvanas évek végéről való, az első teljes párosítási szeparáló algoritmust csak 1982-ben adta meg Padberg és Rao, amit általánosabb formában megtalálhatunk a 11.6 fejezetben. Mivel ez egy igen egyszerű algoritmus, megtörténhetett volna, hogy először ez születik meg, és csak utána az optimalizációs problémára vonatkozó polinomiális algoritmus, amelynek tehát a Grötschel et al. tétel szerint léteznie kell.

Ezen alfejezetben ünnepelt tételnek azonban egyelőre csak elvi jelentősége van, algoritmikus szempontból még nem sikerült hasznot húzni belőle. Elvi jelentősége szerint ha sikerül egy problémának olyan poliéderes leírását adni, amely segítségével a szeparációs feladatot polinomidőben meg tudjuk oldani, akkor joggal várhatjuk, hogy egy direkt kombinatorikai jellegű optimalizálásis algoritmus is kidolgozásra kerül. Így történt ez példának okáért az úgynevezett útpárosítási feladattal.

3.11. Csoport-feladat

A csoport-feladat az egészértékű programozási feladat egy olyan relaxációja, ahol nem az egészértékűségből engedünk, hanem bizonyos feltételeknél egyenlőség helyett csak azonos maradékosztályba esést követelünk. Innen jön a „csoport-feladat" elnevezés – a maradékosztályok csoportjáról van szó.

Legyen az egészértékű programozási feladatunk $\max\{cx : Ax = b, x \in \mathbb{Z}_+^n\}$ alakban adva, ahol A egy $m \times n$-es egész mátrix, b és c pedig egész vektorok. Legyen K egy k rangú $m \times k$-as egész mátrix, és $\beta \in \mathbb{R}^k$. A csoport-feladat a következő egészértékű optimalizálási feladat:

$$\begin{aligned} z_K(b) &= \max cx - \beta\lambda \\ Ax - K\lambda &= b \\ x &\in \mathbb{Z}_+^n \\ \lambda &\in \mathbb{Z}^k \end{aligned}$$

Hogyan válasszuk meg a β vektort? Az eredeti feladat LP-relaxációjának duálisa $\min\{yb : yA \geq c, y \in \mathbb{R}^m\}$. Feltesszük hogy ennek van optimális megoldása: y^*. Válasszuk a $\beta = y^*K$ vektort. Így a csoport-feladat a következő alakba írható:

$$\begin{aligned} z_K(b) &= y^*b + \max(c - y^*A)x \\ x &\in S_K(b), \end{aligned}$$

ahol

$$S_K(b) = \{x \in \mathbb{Z}_+^n : Ax - K\lambda = b \text{ valamilyen } \lambda \in \mathbb{Z}^k\text{-ra}\}.$$

Figyeljük meg, hogy $c - y^*A \leq 0$ és y^*b az LP-relaxáció optimuma, tehát itt a csoport-feladat optimum-értéke legfeljebb az LP-relaxáció optimum-értéke.

A továbbiakban az $S_K(b)$ halmaz egy szebb reprezentációját akarjuk megadni. Ehhez a K mátrix Smith-féle normálalakját használjuk:

3.29. tétel. *Ha K egy k rangú $m \times k$-as egész mátrix, akkor létezik R és C unimoduláris egész mátrix a következő tulajdonságokkal: R $m \times m$-es, C $k \times k$-as, és $RKC = \Delta$, ahol a Δ $m \times k$-as mátrixnak csak a $\delta_{i,i}$ $(i = 1, \ldots, k)$ elemei nemnullák, ezek pozitív egészek, és $\delta_{i,i}$ osztója $\delta_{i+1,i+1}$-nek.*

A Δ mátrixra nézve már értelmezhetőek a maradékosztályok: egy $d \in \mathbb{Z}^m$ vektorra

$$\phi_\Delta(d)_i = \begin{cases} d_i \mod \delta_{i,i} & \text{ha } i \leq k, \\ d_i & \text{egyébként.} \end{cases}$$

Ennek segítségével már megadhatjuk $S_K(b)$ kanonikus alakját.

3.30. állítás.

$$S_K(b) = \{x \in \mathbb{Z}_+^n : \sum_{j=1}^n \phi_\Delta(Ra_j)x_j \equiv \phi_\Delta(Rb) \mod \Delta\}.$$

Bizonyítás.

$$\begin{aligned} S_K(b) &= \{x \in \mathbb{Z}_+^n : Ax - K\lambda = b \text{ valamilyen } \lambda \in \mathbb{Z}^k\text{-ra}\} \\ &= \{x \in \mathbb{Z}_+^n : RAx - RK\lambda = Rb \text{ valamilyen } \lambda \in \mathbb{Z}^k\text{-ra}\}. \end{aligned}$$

Mivel C unimoduláris, $C\lambda$ pontosan akkor egész vektor ha λ is az. Ezért

$$\begin{aligned} S_K(b) &= \{x \in \mathbb{Z}_+^n : RAx - RKC\lambda = Rb \text{ valamilyen } \lambda \in \mathbb{Z}^k\text{-ra}\} \\ &= \{x \in \mathbb{Z}_+^n : RAx = Rb + \Delta\lambda \text{ valamilyen } \lambda \in \mathbb{Z}^k\text{-ra}\}. \end{aligned}$$

□

A fentiek alapján a csoport-feladat a következő alakra hozható:

$$z_K(b) = y^*b + \max \sum_{j=1}^n (c_j - y^*a_j)x_j$$

$$\sum_{j=1}^n \phi_\Delta(Ra_j)x_j \equiv \phi_\Delta(Rb) \mod \Delta$$

$$x \in \mathbb{Z}_+^n.$$

Tekintsük most azt az esetet, amikor $k = m$, azaz K egy $m \times m$-es nem-szinguláris egész mátrix. Ekkor összesen $\prod_{i=1}^m \delta_{i,i} = |\det(K)|$ maradékosztályunk van. Mivel $c_j - y^*a_j \le 0$ minden j-re, a csoport-feladat itt megoldható minimális költségű út keresési feladatként egy $|\det(K)|$ csúcsszámú irányított gráfon.

3.12. Sarokpoliéderek és metszet vágások

Az előző fejezetben $\max\{cx : Ax = b, x \in \mathbb{Z}_+^n\}$ alakú feladatot néztünk, és egy K mátrix segítségével írtuk fel a csoport-feladatot. Nézzük meg, hogy mi történik, ha $K = B$, ahol B az LP-relaxáció optimális bázisa! Feltesszük, hogy A és b egészek. Legyen y^* a duális feladat egy optimális megoldása; a csoportfeladat így

$$\max\{cx - y^*B\lambda : x \in \mathbb{Z}_+^n,\ \lambda \in \mathbb{Z}^m,\ B^{-1}Ax - \lambda = B^{-1}b\}.$$

Mivel a bázisváltozók oszlopaira megszorítva $B^{-1}A$ egységmátrix, és $y^*B = c_B$, ez ekvivalens a következővel (N jelöli a bázisban nem szereplő változók indexhalmazát):

$$\max\{cx : Ax = b,\ x \in \mathbb{Z}^n,\ x_j \ge 0 \text{ ha } j \in N\}.$$

Tehát a csoportfeladatnak az a relaxáció felel meg, hogy a bázisváltozókra elhagyjuk a nemnegativitási feltételt. A továbbiakban ennek a relaxációnak a megoldáshalmazával foglalkozunk.

Vezessük be az $\overline{A} = B^{-1}A$, $\overline{b} = B^{-1}b$ és $S = S_B(b)$ jelöléseket, legyen $\overline{x}$ a bázismegoldás, és az egyszerűség kedvéért tegyük fel, hogy $B = \{1, 2, \dots, m\}$. Ezekkel

$$S = \{x \in \mathbb{Z}^n : x_i = \overline{b}_i - \sum_{j \in N} \overline{a}_{ij}x_j \ (i \in B),\ x_j \ge 0 \ (j \in N)\}.$$

3.31. definíció. A conv(S) halmazt a B bázishoz tartozó *sarokpoliédernek* nevezzük, és corner(B)-vel jelöljük.

Ha S definíciójában elhagyjuk az egészértékűségi feltételt, a következő poliédert kapjuk:

$$P(B) = \{x \in \mathbb{R}^n : x_i = \bar{b}_i - \sum_{j \in N} \overline{a}_{ij} x_j \ (i \in B),\ x_j \geq 0 \ (j \in N)\}.$$

3.32. állítás. *A $P(B)$ poliédernek egyetlen csúcsa van: $\overline{x}$, és $n-m$ extrém iránya: r^j $(j \in N)$ ahol*

$$r_i^j = \begin{cases} -\overline{a}_{ij} & \textit{ha } i \in B \\ 1 & \textit{ha } i = j \\ 0 & \textit{egyébként.} \end{cases}$$

Bizonyítás. Könnyen látható. □

3.33. állítás. *Ha aff$(P(B))$ tartalmaz egész pontot, akkor corner(B) $n-m$ dimenziós.*

Bizonyítás. Belátjuk, hogy corner(B) nemüres. Ebből már következik az állítás, hiszen Meyer tétele (2.1. tétel) miatt ekkor $P(B)$ és corner(B) karakterisztikus kúpja megegyezik. Legyen $x \in \operatorname{aff}(P(B)) \cap \mathbb{Z}^n$, és $N^- = \{j \in N : x_j < 0\}$. Ha $N^- = \emptyset$, akkor kész vagyunk. Különben legyen k olyan pozitív egész szám, hogy $k\overline{a}_{ij} \in \mathbb{Z}$ minden $i \in [m], j \in N^-$-ra. Tekintsük a következő x' vektort: $x'_j = x_j$ ha $j \in N \setminus N^-$, $x'_j = x_j - k\lfloor x_j/k \rfloor$ ha $j \in N^-$, végül $x'_i = \bar{b}_i - \sum_{j \in N} \overline{a}_{ij} x'_j$ ha $i \in B$. Könnyű ellenőrizni, hogy $x' \in \operatorname{corner}(B)$. □

A továbbiakban corner(B)-re érvényes vágásokat fogunk keresni. Ez azért hasznos, mert ezek a vágások nyilvánvalóan érvényesek az eredeti feladatra is, így használhatjuk őket vágósíkként. Jegyezzük meg, hogy ha $\bar{b}$ egész, akkor corner$(B) = P(B)$, tehát ekkor nincsenek nemtriviális vágások. Ezért mostantól feltesszük, hogy $\overline{x}$ nem egész, és őt leválasztó vágást keresünk. Ráadásul mivel a bázisváltozók kifejezhetők a többi változó lineáris kombinációjaként, elég olyan egyenlőtlenségeket nézni, amikben csak a nembázis-változók szerepelnek.

3.34. lemma. *Ha corner(B) nemüres, akkor minden nemtriviális vágás $\sum_{j \in N} \gamma_j x_j \geq 1$ alakú, ahol $\gamma_j \geq 0$ $(j \in N)$.*

Bizonyítás. Tekintsünk egy $\sum_{j \in N} \gamma_j x_j \geq \beta$ vágást. Ha valamilyen $l - re$ $\gamma_l < 0$, akkor $\sum_{j \in N} \gamma_j r_j^l = \gamma_l < 0$, ellentmondásban azzal, hogy r^l végtelen iránya corner(B)-nek. Tehát $\gamma_j \geq 0$ minden $j \in N$-re, és persze $\beta > 0$ ha a vágás nemtriviális, tehát az általánosság megszorítása nélkül feltehetjük, hogy $\beta = 1$. □

Az alábbiakban Egon Balas módszerét ismertetjük ilyen vágások konstruálására.

3.35. definíció. Egy $C \subseteq \mathbb{R}^n$ konvex zárt halmaz *rácspontmentes*, ha a belsejében nincs egész vektor. Legyen C rácspontmentes konvex zárt halmaz aminek $\overline{x}$ a belsejében van, és legyen

$$\alpha_j = \sup\{\alpha : \overline{x} + \alpha r^j \in C\} \ (j \in N).$$

Ekkor a C által meghatározott *metszet vágás* a $\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1$ vágás (itt α_j értéke lehet $+\infty$, ekkor $\frac{1}{\alpha_j} = 0$).

3.36. tétel (Balas). *Tetszőleges C rácspontmentes konvex zárt halmazra, aminek $\overline{x}$ a belsejében van, az általa meghatározott metszet vágás érvényes corner(B)-re.*

Bizonyítás. Megmutatjuk, hogy $P(B)$-nek az a része, ami nem teljesíti az egyenlőtlenséget, C belsejében van. Ebből következik az állítás, hiszen C belsejében nincs egész vektor.

Legyen $Q = P(B) \cap \{x \in \mathbb{R}^n : \sum_{j\in N} \frac{x_j}{\alpha_j} < 1\}$ (azaz $P(B)$-nek az a része, ami nem teljesíti az egyenlőtlenséget) és legyen $\overline{Q} = P(B) \cap \{x \in \mathbb{R}^n : \sum_{j\in N} \frac{x_j}{\alpha_j} \le 1\}$. A $\overline{Q}$ halmaz egy $n-m$ dimenziós poliéder, aminek csúcsai $\overline{x}$, $\overline{x} + \alpha_j r^j$ (azon j indexekre, amikre α_j véges), extrém irányai pedig r^j (azon j indexekre, amikre α_j végtelen). Mivel a $\sum_{j\in N} \frac{x_j}{\alpha_j} = 1$ lapon az $\overline{x} + \alpha_j r^j$ csúcsok vannak, Q minden pontja kifejezhető mint az $\{\overline{x} + \alpha r^j\ (0 \le \alpha < \alpha_j)\}$ félig nyílt szakaszok pontjainak konvex kombinációja plusz az extrém irányok nemnegatív kombinációja. Mivel ezek a félig nyílt szakaszok C belsejében vannak, Q is része C belsejének. □

3.37. tétel. *Minden, corner(B)-re érvényes vágás metszet vágás valamilyen C-re.*

Bizonyítás. Legyen $\sum_{j\in N} \gamma_j x_j \ge 1$ egy nemtriviális vágás; ekkor a 3.34. lemma szerint $\gamma_j \ge 0$ minden $j \in N$-re. Vegyük $P(B)$-nek egy olyan $Dx \le d$ leírását, ahol D egész mátrix és d egész vektor. Legyen

$$C = \{x \in \mathbb{R}^n : Dx \le d + \mathbf{1},\ \sum_{j\in N} \gamma_j x_j \le 1\}.$$

Figyeljük meg, hogy C-nek nincs egész vektor a belsejében, hiszen arra $Dx \le d$ teljesülne, így $P(B)$-beli egész vektor lenne amire $\sum_{j\in N} \gamma_j x_j < 1$, ellentmondva a vágásunk érvényességének.

Tehát C egy rácspontmentes konvex zárt halmaz, aminek $\overline{x}$ a belsejében van, és könnyű ellenőrizni, hogy az általa adott metszet vágás pont $\sum_{j\in N} \gamma_j x_j \ge 1$. □

A lényeges vágások megtalálásához elég tartalmazásra maximális rácspontmentes konvex zárt halmazokat nézni. Ezek struktúráját Lovász tétele írja le; csak a korlátos esetet bizonyítjuk.

3.38. tétel (Lovász)**.** *Ha C tartalmazásra maximális teljes dimenziós rácspontmentes konvex zárt halmaz, akkor $C = P + L$, ahol P poliéder, L lineáris altér,* $\dim(P) + \dim(L) = n$, *és C minden lapjának relatív belsejében van egész pont.* □

Bizonyítás. (Korlátos eset.) Feltesszük, hogy C része a $B = \{x \in \mathbb{R}^n : l \le x \le u\}$ téglának. Minden $z \in B \cap \mathbb{Z}^n$ ponthoz létezik a_z és b_z, hogy $a_z x \le b_z\ \forall x \in C$, de $a_z z \ge b_z$. A $P = \{x \in \mathbb{R}^n \cap B : a_z x \le b_z\ \forall z \in B \cap \mathbb{Z}^n\}$ korlátos poliéder rácspontmentes, és $C \subseteq P$, tehát $C = P$.

Ha egy lapnak a relatív belsejében nem lenne egész pont, akkor a lapot meghatározó egyenlőtlenség jobboldalát kicsit megnövelve rácspontmentes poliédert kapnánk, ellentmondásban a maximalitással. □

3.39. állítás. *Ilyen C-nek legfeljebb 2^n lapja van.*

Bizonyítás. Legyen x^F az F lap belsejében lévő egész pont. Ha 2^n-nél több lap lenne, akkor a skatulya-elv szerint lenne két olyan lap (F_1, F_2), hogy minden koordinátában x^{F_1} és x^{F_2} paritása megegyezik. Ekkor viszont $\frac{x^{F_1}+x^{F_2}}{2}$ is egész vektor, és C belsejében van, ellentmondásban a rácspontmentességgel. □

2 dimenzióban ezek szerint legfeljebb 4 lap lehet. További esetszétválasztással belátható, hogy a tartalmazásra maximális rácspontmentes konvex zárt halmazok a következő kategóriákba sorolhatók:

- $\{(x_1, x_2) : b \le a_1 x_1 + a_2 x_2 \le b + 1\}$, ahol a_1, a_2, b egészek, és $(a_1, a_2) = 1$.

- Háromszög, ezen belül 3 típus. Első típus: minden csúcs egész, minden oldal belsejében 1 egész pont. Második típus: legalább 1 tört csúcs; az erre illeszkedő oldalak belsejében 1 rácspont, a harmadik oldal belsejében legalább 2. Harmadik típus: Összesen 3 rácspont a háromszög határán.

- Négyszög, minden oldal belsejében egyetlen rácspont.

Megjegyezzük, hogy a fentiekhez hasonló tételeket lehet kimondani vegyes programozási feladatokra is, és a módszer igazán ilyen feladatoknál hasznos. Belátható, hogy minden $(\mathbb{Z}^p \times \mathbb{R}^{n-p})$-mentes maximális konvex zárt halmaz előáll, mint egy maximális $\mathbb{Z}^p$-mentes poliéder és $\mathbb{R}^{n-p}$ direkt szorzata. Ezért a vágásokat annyi dimenziós rácspontmentes poliéderek határozzák meg, ahány egész változó van.

3.13. Vágások felemelése (lifting)

A 3.5. Lemmában láttuk, hogy ha a P racionális poliéder egy F oldalára van egy G-C levezethető $cx \leq d$ vágás, akkor P-re van olyan G-C-levezethető $c'x \leq d'$ vágás, amire $F \cap \{x : c'x \leq d'\} = F \cap \{x : cx \leq d\}$. Az alábbiakban azt a speciális esetet nézzük, amikor $P \subseteq [0,1]^n$, és $F = \{x \in P : x_1 = 0\}$. Azt vizsgáljuk, hogy egy F-re érvényes vágás P-re való felemelése mikor határozza meg P_I egy lapját.

Példaként tekintsük egy $G = (V, E)$ gráf stabil halmaz poliéderét. Legyen $P = \{x \in \mathbb{R}^V : x \geq 0,\ x_u + x_v \leq 1\ \forall\ uv \in E\}$. Ekkor P_I a stabil halmazok karakterisztikus vektorainak konvex burka, ezt nevezzük stabil halmaz poliédernek. Könnyű látni, hogy P_I teljes dimenziós.

3.40. tétel (Padberg)**.** *Legyen $C \subseteq V$ egy klikk. A $\sum_{v \in C} x_v \leq 1$ vágás érvényes, és pontosan akkor határozza meg lapját P_I-nek, ha C tartalmazásra nézve maximális klikk.*

Bizonyítás. Ha C nem tartalmazásra nézve maximális, akkor létezik $C' \supsetneq C$ klikk. A $\sum_{v \in C'} x_v \leq 1$ vágás szigorúan dominálja a $\sum_{v \in C} x_v \leq 1$ vágást, tehát utóbbi nem lehet lap.

Ha C tartalmazásra nézve maximális, akkor meg tudunk adni $|V|$ darab P_I-beli affin független vektort, ami az egyenlőtlenséget egyenlőséggel teljesíti:

- $v \in C$-re χ_v,

- $v \notin C$-re $\chi_{\{v,w\}}$, ahol w egy olyan C-beli csúcs, amire $vw \notin E$ (ilyen van, mert C maximális).

□

Ismert tétel, hogy P_I-nek pontosan akkor nincs más lapja, ha G perfekt gráf. Tegyük fel most, hogy G-ben van egy H páratlan lyuk, azaz egy $2k+1$ elemű feszített kör, ahol $k \geq 2$. Ekkor $\sum_{v \in H} x_v \leq k$ érvényes vágás. Ha $V = H$, akkor persze ez lapja P_I-nek, hiszen van $2k+1$ affin független megoldás ami egyenlőséggel teljesíti, más esetben viszont nem feltétlenül. Például ha $V = H + z$, és $vz \in E$ minden $v \in H$-ra, akkor nem lap (az ilyen gráfot nevezzük wheel-nek, a vz élek a küllők). Ahhoz, hogy lapot csináljunk belőle, meg kell vizsgálnunk, mi az a maximális α, amire

$$\alpha x_z + \sum_{v \in H} x_v \leq k$$

érvényes vágás. Látható, hogy $\alpha = k$-ra még érvényes, de nagyobb α-ra már nem, a $\{z\}$ egyelemű stabil halmaz miatt. Így a

$$k x_z + \sum_{v \in H} x_v \leq k$$

érvényes vágás, ráadásul lapja P_I-nek, mert a χ_z vektor is egyenlőséggel teljesíti, és affin független az eddigi $2k+1$ vektorral. Ezt a vágást nevezzük a $\sum_{v\in H} x_v \le k$ vágás **felemeltjének**.

Általánosan a következő tételek mondják ki a felemelt vágás létezését. Legyen $P \subseteq [0,1]^n$ racionális poliéder, $P^0 = P \cap \{x : x_n = 0\}$, és $P^1 = P \cap \{x : x_n = 1\}$.

3.41. tétel (Wolsey)**.** *Tegyük fel, hogy $\sum_{j=1}^{n-1} c_j x_j \le d$ érvényes P_I^0-re, és k dimenziós oldalt határoz meg. Tegyük fel továbbá, hogy P_I^1 nem üres. Ekkor az $\alpha = d - \max\{\sum_{j=1}^{n-1} c_j x_j : x \in P_I^1\}$ értékre az*

$$\alpha x_n + \sum_{j=1}^{n-1} c_j x_j \le d$$

vágás érvényes P_I-re, és $k+1$ dimenziós oldalt határoz meg.

Bizonyítás. Az érvényességet könnyű ellenőrizni α definíciója alapján. A dimenzió növekedése abból adódik, hogy P_I^1-ben is van vektor ami egyenlőséggel teljesíti. □

Megjegyezzzük, hogy ha c és d egész, akkor α is egész lesz; ha pedig $\sum_{j=1}^{n-1} c_j x_j \le d$ érvényes P_I^1-re is, akkor α nemnegatív. A tétel kimondható arra az esetre is, amikor P^0 és P^1 szerepét felcseréljük:

3.42. tétel (Wolsey)**.** *Tegyük fel, hogy $\sum_{j=1}^{n-1} c_j x_j \le d$ érvényes P_I^1-re, és k dimenziós oldalt határoz meg. Tegyük fel továbbá, hogy P_I^0 nem üres. Ekkor a $\beta = -d + \max\{\sum_{j=1}^{n-1} c_j x_j : x \in P_I^0\}$ értékre a*

$$\beta x_n + \sum_{j=1}^{n-1} c_j x_j \le d + \beta$$

vágás érvényes P_I-re, és $k+1$ dimenziós oldalt határoz meg.

Bizonyítás. Hasonlóan az előzőhöz. □

A fentiekből például következik, hogy ha egy gráfban van egy H páratlan lyuk, és P a stabil halmaz poliéder, akkor a $\sum_{v\in H} x_v \le (|H|-1)/2$ egyenlőtlenséget lépésenként egy csúcsot hozzávéve fel tudjuk emelni P_I lapjává. Megjegyzendő, hogy a kapott lap függhet a csúcsok hozzávételének sorrendjétől.

Egy másik fontos alkalmazás a 3.4.1 fejezetben szereplő fedési vágások felemelése. Tekintsünk egy tetszőleges J tartalmazásra minimális függő halmazt, és a $\sum_{j\in J} x_j \le |J|-1$ alakú fedési vágást. Az egyszerűség kedvéért tegyük fel, hogy $J = \{1,\dots,k\}$. Egyenként határozzuk meg az $x_{k+1},\dots,x_n$ változók $\alpha_{k+1},\dots,\alpha_n$ együtthatóit, a következőképpen:

$$\alpha_l = |J| - 1 - \max\{\sum_{j=1}^{k} x_j + \sum_{j=k+1}^{l-1} \alpha_j x_j : (x,1,0,..,0) \in P_I\}.$$

A 3.41 tétel utáni megjegyzés értelmében minden α_l nemnegatív egész szám. Mivel J tartalmazásra minimális függő halmaz, a végén kapott $\sum_{j=1}^{k} x_j + \sum_{j=k+1}^{n} \alpha_j x_j \le |J|-1$ vágás lapot határoz meg a 3.41 tétel szerint. Első ránézésre úgy tűnhet, hogy α_l meghatározása maga is egy hátizsák-feladat. Azonban megmutatjuk, hogy valójában az α_l értékék dinamikus programozással nagyon gyorsan kiszámolhatók. Adott $l \in \{k+1,\dots,n\}$ és $s \in \{0,\dots,|J|-1\}$ értékekre legyen

$$\begin{aligned} f(l,s) &= \min\{\sum_{j=1}^{l} a_j x_j :\ x \in \{0,1\}^l,\ x_l = 1,\ \sum_{j=1}^{k} x_j + \sum_{j=k+1}^{l-1} \alpha_j x_j = s\} \\ &= a_l + \min\{\sum_{j=1}^{l-1} a_j x_j :\ x \in \{0,1\}^{l-1},\ \sum_{j=1}^{k} x_j + \sum_{j=k+1}^{l-1} \alpha_j x_j = s\}. \end{aligned}$$

Tetszőleges s-re $f(k+1,s)$ könnyen kiszámolható, mert a legkönnyebb s darab tárgyat kell venni. Ha adott l-re ismerjük $f(l,s)$ értékét minden s-re, akkor α_l így kapható meg:

$$\alpha_l = |J| - 1 - \max\{s :\ f(l,s) \leq b\},$$

Az $f(l+1,s)$ értékek pedig a következők:

$$f(l+1,s) = a_{l+1} + \min\{f(l,s) - a_l, f(l, s-\alpha_l)\}.$$

3.14. Felemelés és vetítés

A felemelés és vetítés (lift and project) módszere bináris egészértékű programozási feladatoknál használható megengedett vágások keresésére. A módszer két fázisból áll: a felemelési fázisban felírunk egy kvadratikus programozási feladatot, ami erősebb az LP relaxáltnál, majd ezt a kvadratikus feladatot linearizáljuk újabb változók bevezetésével (tehát felemeljük egy magasabb dimenziós térbe, innen a név). A vetítési fázisban ezt a magasabb dimenziós poliédert levetítjük az eredeti térbe.

Nézzük először, hogyan kell vetíteni. Egy $P = \{(x,z) \in \mathbb{R}^{n+d} : Ax + Bz \leq b\}$ poliéder vetülete az x változók terére $P_x = \{x \in \mathbb{R}^n : \exists z \in \mathbb{R}^d : (x,z) \in P\}$.

3.43. állítás. *Legyen $C = \{y : yB = 0, y \geq 0\}$, és legyen E a C kúp extrém irányainak halmaza. Ekkor $P_x = \{x \in \mathbb{R}^n : (yA)x \leq yb\ \forall y \in E\}$.*

Bizonyítás. Legyen $Q = \{x \in \mathbb{R}^n : (yA)x \leq yb\ \forall y \in E\} = \{x \in \mathbb{R}^n : (yA)x \leq yb\ \forall y \in C\}$.

Először belátjuk, hogy $P_x \subseteq Q$. Ha $x \in P_x$, akkor létezik z, hogy $(x,z) \in P$, és ekkor $y \in E$-re

$$(yA)x = (yA)x + (yB)z = y(Ax + Bz) \leq yb.$$

Most belátjuk, hogy $Q \subseteq P_x$. Legyen $x \in Q$; a definíció szerint $y(b - Ax) \geq 0$ minden $y \in C$-re, és ez a Farkas lemma értelmében azzal ekvivalens, hogy létezik $z \in \mathbb{R}^d$ amire $Bz \leq b - Ax$. Azaz $(x,z) \in P$, tehát $x \in P_x$. □

Az állítás értelmében P_x leírásához a C kúp extrém irányait kell ismerni; ez persze nem mindig áll rendelkezésre, sőt akár exponenciálisan sok extrém irány is lehet. De bizonyos szerencsés esetekben meg tudjuk határozni a vetületet.

A felemelés és vetítés algoritmus egy kijelölt változót próbál meg egészértékűvé tenni. Később látni fogjuk, hogy egymás után többször alkalmazva meg is oldja a bináris programozási feladatot (persze nem polinom időben).

Felemelés és vetítés algoritmus

Feltesszük, hogy a feladatunk megengedett tartománya $\mathcal{F} = \{x \in \{0,1\}^n : Ax \le b\}$, A és b egészek, és a $0 \le x \le 1$ egyenlőtlenségek benne vannak az $Ax \le b$ rendszerben.
Input: $A, b, j \in \{1, \dots, n\}$
Output: egy P_j poliéder

- Vegyük azt a kvadratikus rendszert, amit úgy kapunk, hogy az $Ax \le b$ egyenlőtlenségeit megszorozzuk x_j-vel, illetve $(1 - x_j)$-vel:

$$(Ax)x_j \le bx_j, \tag{3.17}$$
$$(Ax)(1 - x_j) \le b(1 - x_j). \tag{3.18}$$

  Figyeljük meg, hogy az így kapott egyenlőtlenségek érvényesek $\mathcal{F}$ minden elemére.

- (Lift) Vegyünk fel minden $i \in \{1, \dots, n\} \setminus \{j\}$-re egy új y_{ij} változót. A fenti egyenlőtlenségekben helyettesítsük $x_i x_j$-t y_{ij}-vel, és helyettesítsük x_j^2-et x_j-vel (az utóbbit az motiválja, hogy $x_j \in \{0,1\}$ esetén tényleg $x_j = x_j^2$). Legyen $L_j(P)$ az így kapott poliéder az (x, y) térben.

- (Project) Vetítsük $L_j(P)$-t vissza az x változók terére: $P_j = (L_j(P))_x$.

3.44. tétel. *Legyen $P = \{x \in \mathbb{R}^n : Ax \le b\}$. Ekkor a Lift-and-Project algoritmussal kapott poliéder:*

$$P_j = \mathrm{conv}(P \cap \{x \in \mathbb{R}^n : x_j \in \{0,1\}\}).$$

Bizonyítás. A $P_j \supseteq \mathrm{conv}(P \cap \{x \in \mathbb{R}^n : x_j \in \{0,1\}\})$ irány bizonyításához legyen $\bar{x} \in P \cap \{x \in \mathbb{R}^n : x_j \in \{0,1\}\}$, és legyen $\bar{y}_{ij} = \bar{x}_i \bar{x}_j$ $(i \ne j)$. Mivel $\bar{x}$ teljesíti a (3.17-3.18) egyenlőtlenségeket és $\bar{x}_j = \bar{x}_j^2$, ezért $(\bar{x}, \bar{y}) \in L_j(P)$, tehát $\bar{x} \in P_j$.

A másik irány bizonyítását több esetre osztjuk. Az mindenesetre világos, hogy $P_j \subseteq P$, hiszen az egymásnak megfelelő (3.17)-(3.18) egyenlőtlenség-párokat összeadva azt kapjuk, hogy $Ax \le b$ érvényes tetszőleges $(x, y) \in L_j(P)$-re.

1. eset: $P \cap \{x \in \mathbb{R}^n : x_j = 0\} = \emptyset$. Ekkor a Farkas Lemma következtében létezik $y \ge 0$, amire $yA = -e_j$ és $yb = -\varepsilon$ valamilyen $\varepsilon > 0$-ra (itt e_j a j-edik egységvektor). Mivel minden x-re, amire a (3.17-3.18) egyenlőtlenségek teljesülnek, teljesül $yAx(1 - x_j) \le yb(1 - x_j)$ is, így az ilyen x-ekre $-e_j x(1 - x_j) = x_j(1 - x_j) \le -\varepsilon(1 - x_j)$. A linearizáláskor ebből az lesz, hogy $0 \le -\varepsilon(1 - x_j)$, tehát ez egy érvényes egyenlőtlenség $L_j(P)$-re, de mivel $x_j \le 1$ benne van a rendszerben, ezért $x_j = 1$ is érvényes $L_j(P)$-re, tehát P_j-re is.

2. eset: $P \cap \{x \in \mathbb{R}^n : x_j = 1\} = \emptyset$. Ez hasonlóan bizonyítható mint az előző eset.

3. eset: $P \cap \{x \in \mathbb{R}^n : x_j = 0\} \ne \emptyset$ és $P \cap \{x \in \mathbb{R}^n : x_j = 1\} \ne \emptyset$. Azt látjuk be, hogy minden $\mathrm{conv}(P \cap \{x \in \mathbb{R}^n : x_j \in \{0,1\}\})$-re érvényes $ax \le \beta$ egyenlőtlenség P_j-re is érvényes. Először belátjuk, hogy ekkor létezik $\lambda \le 0$, hogy $ax + \lambda x_j \le \beta$ minden $x \in P$-re. Valóban, $x \in P$ és $x_j = 0$ esetén bármilyen λ jó, tehát, kihasználva hogy P-nek véges sok csúcsa van, kell lenni λ-nak ami minden $x \in P$-re jó.

Hasonlóan: létezik $\nu \le 0$, hogy $ax + \nu(1 - x_j) \le \beta$ minden $x \in P$-re. Itt $x \in P$ és $x_j = 1$ esetén bármilyen ν jó, tehát, mivel P-nek véges sok csúcsa van, kell lenni ν-nak ami minden $x \in P$-re jó.

Ezek szerint ha egy x kielégíti a (3.17-3.18) egyenlőtlenségeket, akkor $(1 - x_j)(ax + \lambda x_j) \le (1 - x_j)\beta$ és $x_j(ax + \nu(1 - x_j)) \le x_j\beta$. Ezeket összeadva:

$$ax + (\lambda + \nu)(x_j - x_j^2) \le \beta,$$

Tehát az $x_j = x_j^2$ átírással $ax \leq \beta$ érvényes $L_j(P)$-re, tehát P_j-re is. $\square$

Az algoritmust persze alkalmazhatjuk egymás után a különböző változókra, mondjuk $j_1, \ldots, j_t$ sorrendben. Jelölje $P_{j_1,\ldots,j_t}$ az így kapott poliédert.

3.45. tétel.

$$P_{j_1,\ldots,j_t} = \mathrm{conv}(P \cap \{x \in \mathbb{R}^n : x_{j_i} \in \{0,1\}, i = 1, \ldots, t\}).$$

Bizonyítás. A tételt t szerinti indukcióval bizonyítjuk. A 3.44 Tétel szerint $t = 1$ re az állítás igaz, úgyhogy tegyük fel hogy $t \geq 2$. Legyen $Q = \{x : x_{j_i} \in \{0,1\}, i = 1, \ldots, t-1\}$. Ekkor

$$\begin{aligned}
P_{j_1,\ldots,j_t} &= (\mathrm{conv}(P \cap Q))_{j_t} && (3.19)\\
&= \mathrm{conv}(\mathrm{conv}(P \cap Q) \cap \{x : x_{j_t} \in \{0,1\}\}) && (3.20)\\
&= \mathrm{conv}((\mathrm{conv}(P \cap Q) \cap \{x : x_{j_t} = 0\}) \cup (\mathrm{conv}(P \cap Q) \cap \{x : x_{j_t} = 1\})) && (3.21)\\
&= \mathrm{conv}(\mathrm{conv}(P \cap Q \cap \{x : x_{j_t} = 0\}) \cup \mathrm{conv}(P \cap Q \cap \{x : x_{j_t} = 1\})) && (3.22)\\
&= \mathrm{conv}((P \cap Q \cap \{x : x_{j_t} = 0\}) \cup (P \cap Q \cap \{x : x_{j_t} = 1\})) && (3.23)\\
&= \mathrm{conv}(P \cap \{x : x_{j_i} \in \{0,1\}, i = 1, \ldots, t\}). && (3.24)
\end{aligned}$$

Itt (3.20) a 3.44 Tétel következménye; (3.22) onnan jön, hogy egy hipersíkot be lehet vinni a conv-on belülre; (3.23) pedig annak köszönhető, hogy $\mathrm{conv}(\mathrm{conv}(A) \cup \mathrm{conv}(B)) = \mathrm{conv}(A \cup B)$. $\square$

A tételből következik, hogy a végeredmény szempontjából mindegy milyen sorrendben csináljuk a változókra a Lift-and-Project-et, és ha az összes változóra megcsináljuk, akkor éppen az egész pontok konvex burkát kapjuk. Persze ebből nem lesz polinomiális algoritmus, hiszen a vetítést nem feltétlenül tudjuk polinom időben megcsinálni.

A fentinél erősebb módszert kapunk, ha egy lépésben nem csak egyetlen x_j változóra írjuk fel az $(Ax)x_j \leq bx_j$ és $(Ax)(1-x_j) \leq b(1-x_j)$ egyenlőtlenségeket, hanem az összes változóra. A fent bizonyítottak alapján az így kapott P^1 poliéderre teljesül, hogy

$$P^1 = \cap_{j=1}^n \mathrm{conv}(P \cap \{x \in \mathbb{R}^n : x_j \in \{0,1\}\}).$$

Szemidefinit programozás segítségével még jobb vágásokat kaphatunk. Tudjuk, hogy az $(1,x)(1,x)^T$ $(n+1) \times (n+1)$-es szimmetrikus mátrix pozitív definit. Ezért a lienarizálásnál extra feltételként köthetjük ki, hogy ha ebben a mátrixban lecseréljük x_ix_j-t y_{ij}-re és x_j^2-et x_j-re, akkor szemidefinit mátrixot kapunk. Tehát a feladatunk egy LP lesz egy olyan extra feltétellel, hogy a változókból alkotott mátrix pozitív szemidefinit; ez pedig egy szemidefinit programozási feladat, ami megoldható polinom időben. A módszer Lovász és Schrijver nevéhez fűződik, akik a következőt is bizonyították:

3.46. tétel. *Legyen S^t a fenti szemidefinit felemelés és vetítés t-szer való ismétlésével kapott megoldáshalmaz. Konstans t-re S_t-n polinom időben tudunk optimalizálni.*

3.15. Feladatok

3. feladat. *Mekkora a Chvátal-rangja a következő poliédernek?*

$$P := \mathrm{conv}\{(1/2, 1), (3/2, 3), (5/2, 1)\}.$$

4. feladat. *Legyen k egy pozitív egész szám. Mutassuk meg, hogy az alábbi poliéder Chvátal-rangja legalább k.*

$$P := \mathrm{conv}\{(0,0), (1,0), (1/2, k)\}.$$

5. feladat. *Bizonyítsuk be a 3.3 tételt egypontú poliéder esetére.*

6. feladat. *Mekkora a törtpárosítás poliéder Chvátal-rangja?*

7. feladat. *Bizonyítsuk be, hogy a 3.8 tétel tetszőleges korlátos poliéderre is kiterjeszthető.*

8. feladat. *Legyen az $Ax \leq b$ totálisan duálisan egészértékű rendszer, amelyre A egész mátrix. Bizonyítsuk be, hogy $P' = \{x : Ax \leq \lfloor b \rfloor\}$, ahol $P' := \{x : Ax \leq b\}$.*

9. feladat. *Az összefüggő $G = (V, E)$ gráfhoz definiáljuk a következő P poliédert, ahol $x \in \mathbb{R}^V$.*

$$x_u + x_v \leq 1 \quad \forall \quad uv \in E \tag{3.25}$$

$$x_v \geq 0 \quad \forall \quad v \in V$$

Világos, hogy a G stabil ponthalmazai karaktarisztikus vektorainak a konvex burka ($S(G)$) benne van a fenti poliéderben, továbbá $P_I = S(G)$. Adjuk meg a Gomory-Chvátal-levezetését a következő érvényes vágásoknak.

(i) *Legyen $C \subseteq V$ a G egy páratlan körének ponthalmaza.*

$$x(C) \leq (|C| - 1)/2.$$

(ii) *Legyen $W \subseteq V$ a G egy* 5*-kerekének ponthalmaza (*5*-keréknek nevezünk egy gráfot, ha egy 5 hosszú körből áll és egy hatodik pontból, amely az össze többivel össze van kötve). Jelölje r az* 5*-kerék középontját (a másik öt ponttal öszekötött pontot).*

$$2x_r + x(W - r) \leq 2.$$

(iii) *Legyen $K \subseteq V$ a G egy teljes részgráfjának ponthalmaza.*

$$x(K) \leq 1.$$

4. fejezet

Dinamikus programozás

A dinamikus programozás elnevezés a matematikai programozásban egy bizonyos fajta algoritmusszervezési eljárást takar. Az általános definíciótól eltekintünk, ehelyett konkrét példákon mutatunk be néhány dinamikus programozási algoritmust, amelyekből minden Olvasó képet alkothat magának arról, miben is áll ezen elgondolás, és mikor is alkalmazható. Az alábbiakban [6]-re támaszkodva bemutatunk négy-öt alapfeladatot és azok dinamikus programozási megoldását. A dinamikus programozás nem más, mint a rekurzív megoldhatóság lehetősége, a rekurzió maga.

4.1. Legrövidebb utak

Adott egy irányított $D = (V, A)$ gráf, és minden irányított e élén egy c_e nemnegatív költség (vagy hossz). Adott továbbá egy s kiindulási pont, és szeretnénk meghatározni s-ből az összes többi pontba vezető legrövidebb irányított utat.

Tegyük fel, hogy a p pont rajta van az s-ből a t pontba vezető legrövidebb úton, amely utat jelöljünk P-vel.

4.1. megfigyelés. A P s-ből p-ig vezető része egy legrövidebb s-ből p-be vezető út, a p-től t-ig vezető rész pedig egy legrövidebb p-ből t-be vezető út.

Ha ez nem volna igaz, akkor egy rövidebb részúttal egy rövidebb $s - t$-utat kaphatnánk. (Vegyük észre, hogy *leghosszabb* utakra bár hosszabb élsorozatot kapnánk, de nem feltétlenül utat.) A következő összefüggés adódik.

4.2. állítás. *Jelölje $d(v)$ az s-ből v-be vezető legrövidebb út hosszát. Ekkor*

$$d(v) = \min_{u \in V^-} \{d(u) + c_{uv}\}, \tag{4.1}$$

ahol V^- azon pontok halmazát jelöli, amelyekből vezet él v-be.

Ez azt jelenti, hogy ha ismerjük v-ből az összes V^--beli pontba vezető legrövidebb út hosszát, akkor könnyen meg tudjuk határozni a v-be vezető legrövidebb út hosszát is.

Azonban ez a megfigyelés közvetlenül nem ad még algoritmust, hiszen elképzelhető, hogy $d(z)$ kiszámításához szükségünk van $d(w)$-re, ugyanakkor $d(w)$ kiszámításához pedig $d(z)$ ismeretére volna szükség. Ámde irányított köröket nem tartalmazó, úgynevezett *aciklikus* gráfokra mégis kaptunk egy egyszerű algoritmust.

4.3. megfigyelés. Legyen adva a $D = (V, A)$ aciklikus irányított gráf és a V ponthalmazának egy olyan (ún. topologikus) sorrendje, melyre minden él korábbi pontból későbbi pontba mutat. Ekkor ezen sorrend szerint a pontokon az (4.1) rekurzió szerint végighaladva s-ből az összes többi pontba vezető legrövidebb út meghatározható. Ezen algoritmus futási ideje $O(|A|)$.

Vegyük észre, hogy a fenti algoritmus tetszőleges élsúlyozás mellett is helyes eredményt ad, azaz nem kell megkövetelni az élhosszak nemnegativitását.

Tetszőleges irányított gráfra viszont másképp kell valamiféle alkalmas sorrendet meghatároznunk. Ennek egy módja azon $D_k(v)$ függvény bevezetése, ahol $D_k(v)$ az s-ből v-be vezető legrövidebb legfeljebb k darab élből álló út hosszát jelöli. Ekkor igaz a következő:

$$D_k(v) = \min\{D_{k-1}(v), \min_{u \in V^-} [D_{k-1}(u) + c_{uv}]\}. \tag{4.2}$$

Ha a k-t 1-től $|V| - 1$-ig növeljük, akkor a $D_k(v)$ értékeket az összes $v \in V$ pontra a (4.2) rekurzió szerint kiszámolhatjuk. Ez így egy $O(|V||A|)$ futási idejű algoritmust szolgáltat, és persze $d(v) = D_{n-1}(v)$.

4.2. A hátizsákfeladat

Ebben a fejezetben a hátizsákfeladat néhány különböző változatát oldjuk meg dinamikus programozási módszerrel. Amennyiben a feladatban szereplő mennyiségek nem túl nagyok, ez a módszer nagyon hatékonynak bizonyul.

4.2.1. A 0-1 hátizsákfeladat

Tekintsük az alábbi 0-1 hátizsákfeladatot.

$$z := \max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x \in \{0,1\}^n$$

ahol az a_j együtthatók és a b pozitív egész számok.

Tekintsük az alábbi *kisebb* feladatokat, ahol a λ szám a $0, 1, \ldots b$ értékeket veszi fel, az r paraméter pedig az $1, 2, \ldots n$ értéket. Jelölje $P_r(\lambda)$ a következő feladatot.

$$f_r(\lambda) := \max \sum_{j=1}^{r} c_j x_j$$

$$\sum_{j=1}^{r} a_j x_j \leq \lambda$$

$$x \in \{0,1\}^r$$

Ekkor $z = f_n(b)$ adja az eredeti feladat maximumát. Most megadunk egy rekurziót, amely az $f_r(\lambda)$ értékeket azon $f_s(\mu)$ értékekből számolja, melyekre $s \leq r$ és $\mu < \lambda$.

Mit mondhatunk a $P_r(\lambda)$ probléma x^* optimális megoldásáról? Nyilván vagy $x_r^* = 0$, vagy $x_r^* = 1$.

- Ha $x_r^* = 0$, akkor könnyen látható, hogy $f_r(\lambda) = f_{r-1}(\lambda)$.

- Ha $x_r^* = 1$, akkor $f_r(\lambda) = c_r + f_{r-1}(\lambda - a_r)$.

Ezek szerint a következő rekurziót kapjuk:

$$f_r(\lambda) = \max\{f_{r-1}(\lambda), c_r + f_{r-1}(\lambda - a_r)\}.$$

Világos, hogy $f_0(\lambda) = 0$ minden $\lambda \geq 0$ értékre, vagy inkább lépjünk eggyel tovább: $f_1(\lambda) = 0$ minden $0 \leq \lambda < a_1$ esetén, és $f_1(\lambda) = \max[c_1, 0]$ minden $\lambda \geq a_1$ esetén. Ekkor a fenti rekurzió segítségével egymás után ki tudjuk számítani az $f_1, f_2, \dots f_n$ értékeit minden 0-tól b-ig terjedő λ-ra.

Az összes megfelelő függvényérték kiszámítása után felmerül az a kérdés is, hogy hogyan találhatunk meg egy optimális megoldást (amelyhez ugye $f_r(\lambda)$ célfüggvényérték tartozik). Vagy megtarjuk az összes $f_r(\lambda)$ értéket, vagy csak egy $p_r(\lambda)$ indikátort, amelynek az értéke 0 legyen, ha $f_r(\lambda) = f_{r-1}(\lambda)$, és 1 különben.

Ha $p_n(b) = 0$, akkor $f_n(b) = f_{n-1}(b)$, és $x_n^* = 0$. A többi koordinátát pedig $f_{n-1}(b)$-gyel folytatva hasonlóképp határozhatjuk meg. Ha $p_n(b) = 1$, akkor $f_n(b) = c_n + f_{n-1}(b - a_n)$, és $x_n^* = 1$. A többi koordinátát pedig $f_{n-1}(b - a_n)$-nel folytatva hasonlóképp határozhatjuk meg.

Az algoritmus lépésszáma $O(nb)$, hiszen az $f_r(\lambda)$ értékek mindegyikének a kiszámításához csak konstans sok összeadásra, kivonásra és összehasonlításra van szükség. Az optimum meghatározása ezek után hasonló nagyságrendű lépés megtételével történik.

Gyakorlat. A fenti algoritmus segítségével oldjuk meg a következő feladatot!

$$z := \max 10x_1 + 7x_2 + 25x_3 + 24x_4$$

$$2x_1 + x_2 + 6x_3 + 5x_4 \leq 7$$
$$x \in \{0,1\}^4$$

4.2.2. Egészértékű hátizsákfeladat

Tekintsük az alábbi egészértékű hátizsákfeladatot.

$$z := \max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leq b$$
$$x \in \mathbb{Z}_+^n$$

ahol szokás szerint az a_j együtthatók és a b pozitív egész számok.

Az előző alfejezet gondolatmenetét követve definiáljuk az alábbi $P_r(\lambda)$ feladatokat, ahol a λ szám a $0, 1, \dots b$ értékeket veszi fel, az r paraméter pedig az $1, 2, \dots n$ értéket.

$$g_r(\lambda) := \max \sum_{j=1}^{r} c_j x_j$$

$$\sum_{j=1}^{r} a_j x_j \leq \lambda$$

$$x \in \mathbb{Z}_+^r$$

Hogyan kapunk rekurziós formulát? $P_r(\lambda)$ optimális x^* megoldásában az $x_r^* = t$ mellett $g_r(\lambda) = c_r t + g_{r-1}(\lambda - ta_r)$. Tehát

$$g_r(\lambda) = \max_{t=0,1,\dots\lfloor\lambda/a_r\rfloor} \{c_r t + g_{r-1}(\lambda - ta_r)\}.$$

Mivel $\lfloor \frac{\lambda}{a_r} \rfloor = b$ a legrosszabb esetben, ezért az algoritmus lépésszáma $O(nb^2)$.

Fontos megjegyezni, hogy a fenti elgondolás alkalmas felső korlátokkal rendelkező hátizsák-feladat megoldására is. Legyen tehát

$$z := \max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \le b$$

$$x_j \le d_j \quad j = 1, 2, \dots n$$

$$x \in \mathbb{Z}_+^n$$

ahol szokás szerint az a_j együtthatók, a d_j számok és a b pozitív egészek.

Ekkor a szokásos jelöléseinkkel az alábbi rekurzió egy $O(nb^2)$ lépésszámú algoritmust határoz meg.

$$g_r(\lambda) = \max_{t=0,1,\dots \min[d_r, \lfloor\lambda/a_r\rfloor]} \{c_r t + g_{r-1}(\lambda - ta_r)\}.$$

A továbbiakban azt vizsgáljuk meg, hogy tudunk-e gyorsabb algoritmust meghatározni a felső korlátokat nem tartalmazó esetben. A válasz igen, mégpedig a következő apró módosítással élve: a $P_n(\lambda)$ optimális x^* megoldását tekintve az alábbi megállapításokat tehetjük.

- Ha $x_r^* = 0$, akkor $g_r(\lambda) = g_{r-1}(\lambda)$.
- Ha $x_r^* \ge 1$, akkor $g_r(\lambda) = c_r + g_r(\lambda - a_r)$.

Ez egy $O(nb)$ lépésszámú algoritmust eredményez, ahol az optimális megoldások meghatározása hasonlóan az előzőekhez $p_r(\lambda)$ indikátorok segítségével történhet.

Gyakorlat. A fenti algoritmus segítségével oldjuk meg a következő feladatot!

$$z := \max 7x_1 + 9x_2 + 2x_3 + 15x_4$$

$$3x_1 + 4x_2 + x_3 + 7x_4 \le 10$$

$$x \in \mathbb{Z}_+^4.$$

Még egy rekurziót megadunk az egészértékű feladatra az alábbi részfeladatok segítségével.

$$h(\lambda) := \max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \le \lambda$$

$$x \in \mathbb{Z}_+^n$$

Ekkor ha valamely $x_j^* = 1$, akkor $h(\lambda) = c_j + h(\lambda - a_j)$. Így hát a következő rekurzióhoz jutunk:

$$h(\lambda) = \max[0, \max_{j: a_j \le \lambda} \{c_j + h(\lambda - a_j)\}].$$

Ez a formula szintén $O(nb)$ futási idejű algoritmushoz vezet. Vegyük észre, hogy $h = g_n$.

A teljesség kedvéért megjegyezzük, hogy a hátizsákfeladat a felső korlátokat nem tartalmazó esetben megfogalmazható aciklikus irányított gráfban való leghosszabb út kereséseként. Definiáljuk a $D = (V, A)$ irányított gráfot a következőképpen. $V := \{1, 2, \dots b\}$. Az irányított élek pedig legyenek a következők: $(\lambda, \lambda + 1)$ minden lehetséges λ-ra 0 hosszúsággal, és $(\lambda, \lambda + a_j)$ minden lehetséges λ-ra és j-re c_j hosszúsággal. Ekkor $h(\lambda)$ éppen a 0-ból a λ-ba vezető leghosszabb út hosszával egyenlő.

Nemnegatív mátrixú egészértékű feladat

Most a fentihez hasonló dinamikus programozási algoritmust adunk több egyenlőtlenség esetén. Legyen a feladatunk

$$\max\{cx : \ Ax \le b, x \ge 0, x \in \mathbb{Z}^n\},$$

ahol A, b, c nemnegatív és egész.

Adott $k \in \{1, \dots, n\}$ szám és $0 \le d \le b$ egész vektor esetén legyen

$$f_k(d) = \max\{\sum_{j=1}^{k} c_j x_j : \sum_{j=1}^{k} a_{ij} x_j \le d_i \ (i = 1, \dots, m), \ x \ge 0\}.$$

Definiáljuk ezen kívül $f_0(d)$-t 0-nak minden $0 \le d \le b$-re. A következő rekurzióval lehet $k \ge 1$-re és $0 \le d \le b$-re kiszámolni a függvényértékeket:

$$f_k(d) = \begin{cases} f_{k-1}(d) & \text{ha valamilyen } i\text{-re } a_{ik} > d_i, \\ \max\{f_{k-1}(d), f_k(d - a_{.k}) + c_k\} & \text{ha } a_{.k} \le d. \end{cases}$$

Az optimumértéket $f_n(b)$ adja meg. A lépésszám $O(n \prod_{i=1}^{m} (b_i + 1))$, ami sajnos a legtöbb feladatnál nagyon lassú algoritmust ad.

4.3. Fa optimális részfája

Most egy olyan dinamikus programozási megoldást ismertetünk, amelynek nincs közvetlen köze legrövidebb utak kereséséhez. Legyen adva egy $T = (V, E)$ fagráf (azaz összefüggő, körmentes irányítatlan gráf) és egy kijelölt $r \in V$ úgynevezett gyökérpontja. Továbbá adott a gráf minden $v \in V$ csúcsán egy $c_v \in \mathbb{R}$ súly. Határozzuk meg azt a maximális pozitív összsúlyú részfát, amely tartalmazza az r csúcsot.

A T egy tetszőleges $v \neq r$ csúcsára jelölje $p(v)$ a v-ből az r felé egyértelmű úton elindulva az első csúcsot ($p(v)$-t a v *szülőjének* nevezik). Jelölje $S(v)$ a v többi szomszédját, azaz $S(v) = \{w \in V : p(w) = v\}$ ($S(v)$ elemeit szokás v *gyerekeinek* nevezni). Jelölje továbbá $T(v)$ azt a v gyökerű részfát, amely az összes olyan csúcsot tartalmazza, amelyből az r-be vezető egyértelmű út átmegy v-n.

A T egy tetszőleges v csúcsára jelölje $H(v)$ a v-gyökerű $T(v)$ részfán a feladatunk optimális megoldását. Amennyiben nincs pozitív összsúlyú v-gyökerű részfa, akkor $H(v) := 0$. Minden más esetben az optimális részfa tartalmazza a v csúcsot. Ezen optimális részfa tartalmazhat további $T(w)$-részfákat, amelyek valamely $w \in S(v)$ gyökerűek. Ekkor könnyen látható, hogy ezeknek a részfáknak a súlya $H(w)$. Ezek alapján a következő rekurzió adódik:

$$H(v) = \max\{0, c_v + \sum_{w \in S(v)} H(w)\}.$$

A rekurziót a fa leveleinél, azaz az elsőfokú csúcsainál tudjuk elkezdeni a következőképp: $H(v) := \max[c_v, 0]$. Ezek után a $H(v)$ értékek *felfelé haladva* könnyen kiszámíthatók. Magának az optimális fának a meghatározása visszafelé haladva történhet azon fáknak a törlésével, amelyek v gyökerére $H(v) = 0$. Az algoritmus lépésszáma $O(|V|)$.

4.4. Maximális súlyú független csúcshalmaz fában

Egy gráfban egy csúcshalmaz *független* (vagy stabil), ha nem feszít élt. Egy tetszőleges gráfban megkeresni a legnagyobb független csúcshalmazt NP-nehéz feladat, de mint látni fogjuk, fában még a maximális súlyú független csúcshalmazt is meg tudjuk találni tetszőleges súlyozás esetén.

Legyen $T = (V, E)$ egy fagráf, és legyen adott a gráf minden $v \in V$ pontján egy $c_v \in \mathbb{R}_+$ nemnegatív súly. Feladatunk a maximális súlyú független csúcshalmaz megkeresése. Jelöljünk ki egy tetszőleges $r \in V$ gyökeret. Használjuk ugyanazokat a jelöléseket egy csúcs szülőjére, gyerekeire és részfájára, mint az előző fejezetben.

A T egy tetszőleges v csúcsára jelölje $a(v)$ a v-gyökerű $T(v)$ részfán a feladatunk optimális megoldását, $b(v)$ pedig jelölje az olyan független halmaz maximális súlyát $T(v)$-ben, ami nem tartalmazza v-t. Ha v levél, akkor mind $a(v)$, mind $b(v)$ könnyen kiszámolható: $a(v) = c_v,\ b(v) = 0$. Egyéb v csúcsokra pedig a következő rekurzió adódik:

$$a(v) = \max\{\sum_{w \in S(v)} a(w), c_v + \sum_{w \in S(v)} b(w)\},$$

$$b(v) = \sum_{w \in S(v)} a(w).$$

Ez alapján a levelektől felfele haladva az $a(v)$ és $b(v)$ értékek minden csúcsra kiszámolhatók. Az $a(r)$ érték pedig megadja a maximális súlyú független csúcshalmaz súlyát. Magának a csúcshalmaznak a meghatározása visszafelé haladva történhet.

4.5. Termelésütemezési feladat

Ebben az alfejezetben az 1 fejezetben bemutatott termelésütemezési feladat megoldására mutatunk be egy dinamikus programozási megközelítést.

A továbbiakban feltesszük, hogy $z_n = 0$, azaz az n-edik időperiódus végére már nem raktározhatunk semmit, azaz az összes megtermelt anyag mennyisége meg kell, hogy egyezzék az igények összegével.

4.4. állítás. *Létezik olyan optimális megoldása is a termelésütemezési feladatnak, amelyben $z_{t-1}x_t = 0$ minden t-re.*

A fenti állítás azt jelenti, hogy csak abban az időperiódusban termelünk, amikorra a raktár már kiürült. Ez az alábbival ekvivalens.

4.5. állítás. *Létezik olyan optimális megoldás is, amelyben ha $x_t > 0$, akkor $x_t = \sum_{i=t}^{t+k} d_i$ valamely $k > 0$ értékre.*

Amint az könnyen látható, a z vátozók kifejezhetők az x változók, és a d igények segítségével:

$$z_t = \sum_{i=1}^{t} x_i - \sum_{i=1}^{t} d_i.$$

Ezt a célfüggvénybe behelyettesítve az alábbit kapjuk:

$$\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t z_t + \sum_{t=1}^{n} f_t y_t =$$

$$\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t (\sum_{i=1}^{t} x_i - \sum_{i=1}^{t} d_i) + \sum_{t=1}^{n} f_t y_t =$$

$$\sum_{t=1}^{n} c_t x_t - \sum_{t=1}^{n} h_t (\sum_{i=1}^{t} d_i) + \sum_{t=1}^{n} f_t y_t,$$

ahol $c_t = p_t + \sum_{i=t}^{n} h_i$. A h és d értékeket tartalmazó kifejezés egy konstans érték, amelyre az optimum kiszámításánál elég csak végül beszámítani.

A 4.5 állítás szerint érvényes az alábbi rekurzió, ahol $H(t)$ az $t = n$ periódus mellett jelöli az optimális célfüggvényértéket (a konstanstagtól eltekintve).

$$H(k) = \min_{t=1,2,\ldots,k} H(t-1) + f_t + c_t(\sum_{i=t}^{k} d_i).$$

A $H(k)$ értékeket tehát egymás után meg tudjuk határozni 1-től kezdve n-ig. Visszafelé haladva az optimális x_t értékeket is meg tudjuk határozni, ezzel egy $O(n^2)$ algoritmust kapva.

10. feladat. *Fogalmazzuk meg a termelésütemezési feladatot irányított gráfbéli legrövidebb út feladatként.*

5. fejezet

Általános egészértékű programozási módszerek

5.1. Vágósíkos eljárások

Tekintsük a következő egészértékű programozási feladatot.

$$\max cx \tag{5.1}$$

$$Ax = b$$

$$x \geq 0,$$

$$x \text{ egész.} \tag{5.2}$$

Amennyiben nem kötjük ki (5.2)-ot, akkor a feladat *lineáris programozási relaxáltjáról* beszélünk.

> *Általános vágósíkos algoritmus*
>
> - Oldjuk meg a (5.1) lineáris programozási relaxáltját. Legyen x^* a kapott optimális megoldás.
> - Ha x^* egész, akkor megoldottuk a feladatot.
> - Ha nem egész, akkor adjunk a feltételrendszerhez egy olyan feltételt, amelyet az összes egész megoldás teljesít, de x^* nem, és iteráljuk az eljárást.

Ezen általános algoritmus egy olyan megvalósítását találta Gomory 1958-ban, amelyről bebizonyította, hogy véges sok lépésben megkeresi tetszőleges egészértékű programozási feladat optimumát. Annak ellenére, hogy a gyakorlatban az eljárás nem bizonyult hatékonynak, elméleti jelentősége igen nagy.

Tegyük fel, hogy a megoldott lineáris porgramozási relaxált egy optimális bázismegoldása x^*, és jelölje N a bázisban nem szereplő változók indexhalmazát. Könnyen látható, hogy ekkor

$$\sum_{i \in N} x_i \geq 1$$

érvényes vágás, azaz minden egész megoldás teljesíti. Viszont x^* nem. Tehát vágósíkos algoritmusunkat használhatnánk úgy, hogy mindig ilyen új fetételt veszünk a rendszerhez. A gond ezzel az, hogy nem tudjuk garantálni az algoritmus végességét. Gomory véges algoritmusához egy másfajta vágást kell tekintenünk.

A Gomory-féle érvényes vágáshoz a következőképpen jutunk. Legyen a lineáris programozási relaxált egy optimális bázismegoldását mutató egyenletrendszer a következő:

$$x_B + B^{-1}A_N x_N = B^{-1}b.$$

Ennek a felírásnak az együtthatóira használjuk a megszokott jelöléseket: $d_{ij} := (B^{-1}A_j)_i$ és $d_{i0} := (B^{-1}b)_i$. Legyen t a legkisebb olyan index, hogy az optimális bázismegoldásban a t-edik változó nem egész érték. Ekkor a hozzá tartozó egyenlet:

$$x_t + \sum_{j\in N} d_{tj}x_j = d_{t0}.$$

Ebből következik, hogy

$$x_t + \sum_{j\in N} \lfloor d_{tj}\rfloor x_j \leq x_t + \sum_{j\in N} d_{tj}x_j = d_{t0},$$

mivel $x_j \geq 0$ teljesül minden j-re. Ebből az következik az egészértékű x_j megoldásokra, hogy

$$x_t + \sum_{j\in N} \lfloor d_{tj}\rfloor x_j \leq \lfloor d_{t0}\rfloor. \tag{5.3}$$

Ezt az egyenlőtlenséget tehát a feladatunk minden egész megoldása teljesíti, de a lineáris programozási relaxált aktuális optimális megoldása nem. Az így kapott egyenlőtlenségeket *Gomory-vágásnak* nevezzük. Az (5.3) egyenlőtlenséget egy új (nem negatív) slack változóval kiegészítve a következő ekvivalens alakot kapjuk.

$$x_t + \sum_{j\in N} \lfloor d_{tj}\rfloor x_j + s = \lfloor d_{t0}\rfloor. \tag{5.4}$$

Figyeljük meg, hogy amennyiben egy megoldásban minden x_j változó egész, akkor s is egész. Az x_t változó eliminálható az eredeti szimplex tábla sor felhasználásával, ami a következő ekvivalens alakot eredményezi:

$$s - \sum_{j\in N} \{d_{tj}\}x_j = -\{d_{t0}\}$$

ahol $\{\alpha\}$ az α szám törtrésze.

Ha a vágásokat mindig a szimplex tábla legaljára tesszük, akkor az s egészértékűsége miatt a módszer során sosem fogunk s miatt vágást generálni.

Gomory mutatott egy olyan eljárást, ami véges sok lépésben vagy megadja az optimumot, vagy megmutatja, hogy nincs megengedett egész megoldás. Ennek belátásához először röviden ismertetjük a lexikografikus duál szimplex módszert.

5.1.1. Lexikografikus duál szimplex módszer

A lexikografikus duál szimplex módszernél a 0. sorban szerepel a célfüggvény aktuális alakja (ami nemnegatív, mert duál megengedett megoldásunk van), míg a 0. oszlopban szerepelnek a változók értékei (a nembázis változók is szerepelnek, az ő soraik az egységmátrix -1-szeresét alkotják).

A lexikografikus duál szimplex módszer fontos tulajdonsága, hogy a táblázat minden oszlopa (a 0-dik kivételével) mindig lexikografikusan pozitív. Hogy ez az elején teljesüljön, esetleg egy új egyenletet kell hozzávenni a rendszerhez az 1. sorba:

$$\sum_{j:d_{0j}=0} x_j + u = M,$$

ahol u egy új slack változó, M pedig egy kellően nagy szám.

Adott $v, w \in \mathbb{R}^n$; v *lexikografikusan nem-negatív* ha $v = 0$ vagy létezik olyan i index, melyre $v_i > 0$, és $v_j = 0$ minden $1 \le j < i$ indexre. v *lexikografikusan pozitív* ha $v \neq 0$ és lexikografikusan nem-negatív. v *lexikografikusan kisebb, mint* w, ha $w - v$ lexikografikusan pozitív.

Ezek után a pivotálás szabályai:

- A bázist elhagyó változó: a legkisebb indexű, amire $d_{i0} < 0$. Tegyük fel hogy ez az index r (ha nincs ilyen, akkor kész vagyunk: primál megengedett megoldásunk van).
- A bázisba bekerülő változó: Azok közül a $j > 0$ indexek közül, amelyekre $d_{rj} < 0$, válasszuk azt, amelyikre a $\left(-\frac{d_{0j}}{d_{rj}}, -\frac{d_{1j}}{d_{rj}}, -\frac{d_{2j}}{d_{rj}}, \ldots, -\frac{d_{mj}}{d_{rj}}\right)$ vektor lexikografikusan a legkisebb. Legyen ez az oszlop index k.
- Ha nincs olyan $j > 0$ index amire $d_{rj} < 0$, akkor a feladatnak nincs megoldása, hiszen x_r értéke csak negatív lehet.

5.1. lemma. *A lexikografikus duál szimplex módszerben a pivotálás során*

1. *a 0. oszlop lexikografikusan csökken,*
2. *a tábla többi oszlopa pedig lexikografikusan pozitív marad.*

Tehát véges sok lépésben elérünk egy optimális megoldást (ha van), hiszen a 0. oszlop lexikografikusan mindig legalább $(OPT_{LP}, 0, 0, \ldots, 0)$*, ahol* OPT_{LP} *az optimumérték.*

Bizonyítás. Először megmutatjuk, hogy a táblázat oszlopai (a 0.-tól eltekintve) lexikografikusan pozitívak maradnak, ugyanis jelölje D_j és D'_j a táblázat j. oszlopát a pivotálás előtt és után, valamint d_i és d'_i az i. sorát pivotálás előtt és után. Ekkor

$$d'_r = \frac{d_r}{d_{rk}}.$$

Tehát $d'_{rk} = 1$. A pivotálás során az r. oszlop többi elemét lenullázzuk:

$$d'_i = d_i - \frac{d_{ik}}{d_{rk}} d_r, \quad \forall i \neq r.$$

Ebből következik, hogy ha $d_{rj} < 0$ ($j \neq k$), akkora pivotálás után a j. oszlop

$$D'_j = D_j - \frac{D_k}{d_{rk}} d_{rj} = d_{rj} \left(\frac{D_j}{d_{rj}} - \frac{D_k}{d_{rk}} \right),$$

ami lexikografikusan pozitív a k oszlop-index választása miatt (feltesszük, hogy nincs két olyan oszlop, hogy az egyik a másik számsorosa). Ha viszont $d_{rj} \geq 0$, akkor

$$D'_j = D_j - \frac{D_k}{d_{rk}} d_{rj} = D_j + \frac{D_k}{|d_{rk}|} d_{rj},$$

ami lexikografikusan pozitív, mivel D_j és D_K is az.

A 0. oszlop változása a pivotlás után:

$$D'_0 = D_0 - \frac{D_k}{d_{rk}} d_{r0} = D_0 + \frac{D_k}{|d_{rk}|} d_{r0}.$$

Mivel D_k lexikografikusan pozitív, és $d_{r0} < 0$ az r. sor választása miatt, D'_0 lexikografikusan kisebb, mint D_0. □

A fenti lemma következménye, hogy soha nem térhet vissza egy bázis, és az eljárás véges sok lépésben megáll.

5.1.2. Gomory módszerének végessége

5.2. tétel. *Tegyük fel, hogy ismerük egy w számot, amire teljesül, hogy ammennyiben az egészértékű feladat megoldható, akkor az optimum értéke legalább w. Ekkor Gomory vágósíkos algoritmusa véges sok lépésben véget ér a következő végkifejletek valamelyikével:*

- *Találunk egy optimális megoldást*
- *Találunk egy bizonyítékot hogy nincs megoldás: valamilyen i-re $d_{i0} < 0$ és $d_{ij} \geq 0$ minden $j \geq 1$-re*
- *Nincs megoldás, mert a kapott LP relaxáció optimum-értéke kisebb mint w.*

Bizonyítás. Először belátjuk, hogy a 0. sorból csak véges sokszor generálhatunk vágást. Ilyenkor d_{00} nem egész, és a generált vágás:

$$s - \sum_{j \in N} \{d_{0j}\} x_j = -\{d_{00}\}.$$

A lexikografixus duál szimplex módszer első lépésében s hagyja el a bázist. Tegyük fel hogy $d_{n+1,p}$ a pivotelem; jelöljük d'_{ij}-vel a pivotálás utáni elemeket. Ekkor

$$d'_{00} = d_{00} - \frac{\{d_{00}\}}{\{d_{0p}\}} d_{0p}.$$

Tudjuk: $d_{0p} \geq 0$, $\{d_{0p}\} > 0$ (mert pivotelem), tehát $d_{0p} > 0$. Így $\frac{d_{0p}}{\{d_{0p}\}} \geq 1$, tehát

$$d'_{00} \leq d_{00} - \{d_{00}\} = \lfloor d_{00} \rfloor.$$

Tehát a 0. sorból csak véges sokszor generálhatunk vágást, különben a célfüggvény érték w alá csökkenne.

A fentiekből következik, hogy d_{00} egy idő után állandó. Mivel a 0. oszlop lexikografikusan nem nő, ezután d_{10} nem nőhet. Megmutatjuk, hogy csak véges sokszor csökkenhet. Ilyenkor az első sorból generálunk vágást, d_{10} nem egész, és a generált vágás:

$$s - \sum_{j \in N} \{d_{1j}\} x_j = -\{d_{10}\}.$$

A lexikografixus duál szimplex módszer első lépésében s hagyja el a bázist. Tegyük fel hogy $d_{n+1,p}$ a pivotelem; jelöljük d'_{ij}-vel a pivotálás utáni elemeket. Ekkor

$$d'_{10} = d_{10} - \frac{\{d_{10}\}}{\{d_{1p}\}} d_{1p}.$$

Tudjuk: $\{d_{1p}\} > 0$ (mert pivotelem), tehát $d_{1p} \neq 0$. Másrészt $d_{0p} = 0$, mert különben a pivot során d_{00} változna, de az már fixálódott. Tehát $d_{1p} > 0$ mert a p-edik oszlop lexikografikusan pozitív. Így

$$d'_{10} \leq d_{10} - \{d_{10}\} = \lfloor d_{10} \rfloor.$$

Tehát véges sok lépés után d_{10} vagy fixálódik, vagy negatívvá válik. Nézzük az utóbbi esetet: x_1 negatív az aktuális megoldásban. Ebből következik, hogy $d_{1j} \geq 0$ minden $j \in N$-re, mert különben az oszlop lexikografikusan pozitív mivolta miatt $d_{0j} > 0$, ami azt jelenti hogy pivotáláskor d_{00} csökkenne ellentétben a feltevésünkkel. Viszont ilyenkor annál az esetnél vagyunk, amikor bizonyítékunk van hogy a feladatnak nincs megoldása.

Marad az az eset, amikor d_{10} is fixálódik egy idő után. Hasonló gondolatmentettel belátható, hogy $d_{20}, d_{30}, \ldots,$ is fixálódik véges sok lépésben, azaz egy idő után az összes eredeti változó fixálódik. De mint láttuk, az utólag hozzávett slack változókból soha nem generálunk vágást, tehát az algoritmus véget ér. □

11. feladat. *Mutassuk meg, hogy a Gomory-vágások előállnak Gomory-Chvatal-vágásként, és fordítva. (Vigyázzunk arra, hogy más alakú feladatoknál képeztük ezen vágásokat. Előbb azonos alakra kell hoznunk a feladatokat.)*

5.2. Korlátozás és szétválasztás módszere

5.2.1. LP alapú korlátozás és szétválasztás

Tekintsük a következő alakú feladatot:

$$z_{IP} := \max cx \tag{5.5}$$

$$Ax \leq b \tag{5.6}$$

$$x \text{ egész,} \tag{5.7}$$

ahol A, b, c egészek. A *korlátozás és szétválasztás* módszerének alapgondolata az, hogy a feladatot szétválasztjuk részfeladatokra úgy, hogy az eredeti feladat megengedett megoldásainak halmaza a részfeladatok megengedett megoldás-halmazainak diszjunkt uniója legyen. Ekkor az eredeti feladat optimum-értéke megegyezik a részfeladatok optimum-értékeinek maximumával. Az egyes részfeladatokat további részfeladatokra lehet szétbontani, így a vizsgált feladatok egy fát alkotnak, aminek gyökerében az eredeti feladat található.

A részfeladatok relevanciájának vizsgálatához van szükség a *korlátozásra.* A korlátozás azt jelenti, hogy minden részfeladatnál felső (és esetleg alsó) korlátot számolunk az optimum értékére. Amennyiben egy részfeladatra vonatkozó felső korlát kisebb mint egy már ismert alsó korlát, akkor azzal a részfeladattal nem kell tovább foglalkozni, törölhetjük a fából.

LP alapú korlátozás és szétválasztásról akkor beszélünk, ha a felső korlátokat az LP relaxált optimuma adja, a szétválasztás pedig lineáris egyenlőtlenségek hozzáadásával történik. A továbbiakban leírjuk az LP alapú általános algoritmust.

Ha van egy S részfeladatuk, $u(S)$ jelöli az LP relaxált optimumértékét, ami felső korlátot ad az S részfeladat optimum-értékére. Az eredeti feladatunkat S_0 jelöli. Az algoritmus során kapott részfeladatokat egy (S_0 gyökerű) *fa* csúcsainak tekintjük, és fenntartunk egy *jelölt-listát*, amin a megoldandó részfeladatok vannak. Az algoritmus lépései:

1. *Inicializálás*: Az S_0 feladatot a jelölt-listára tesszük. Valamilyen heurisztikus módszerrel kiszámolunk egy L alsó korlátot.

2. A jelölt-listáról kiválasztunk egy S feladatot, és kiszámoljuk az $u(S)$ értéket.

   (a) Ha $u(S)$ nem létezik, mert az LP relaxált nem megoldható: töröljük S-et a fából

   (b) Ha $u(S) \leq L$: töröljük S-et a fából

   (c) Ha $u(S) > L$ és az LP relaxáltnak van egész optimális megoldása: megjegyezzük ezt a megoldást, és $L := u(S)$

   (d) Ha $u(S) > L$ és az LP relaxáltnak nincs egész optimális megoldása: legyen x^* egy optimális megoldás, ahol x_i^* nem egész. Válasszuk szét S-t S' és S'' részfeladatokra, ahol S'-t az $x_i \leq \lfloor x_i^* \rfloor$ feltétel, S''-t pedig az $x_i \geq \lceil x_i^* \rceil$ feltétel hozzávételével kapjuk S-ből. Adjuk S'-t és S''-t a jelöltlistához (és legyenek ők S gyerekei a fában).

3. Töröljük S-t a jelölt-listáról. Ha a lista nemüres, ugorjunk a 2. lépésre Ha a lista üres, akkor az utoljára megjegyzett megoldás, aminek értéke L, optimális.

Látható, hogy az algoritmus során több helyen választási lehetőségünk van.

- **Hogyan válasszuk ki a 2.(d) lépésben azt a változót, ami szerint szétválasztunk?** A szétválasztás annál jobb, minél jobban lecsökkennek a felső korlátok. Persze nem tudjuk minden lehetséges szétválasztásra kiszámolni, hogy milyen felső korlátokat adna, ez túl időigényes lenne. Két módszert említünk meg:

  – **Strong branching:** minden lehetséges szétválasztásnál a kapott részfeladatokon teszünk néhány lépést a duál szimplex módszerrel. Az így kapott felső korlátokat hasonlítjuk össze.

  – **Hányados módszer:** Csak egy lépést teszünk a duál szimplex módszerrel, ez pedig bizonyos hányadosok kiszámolását jelenti. Ha az $x_i \leq \lfloor x_i^* \rfloor$ egyenlőtlenséget vesszük a rendszerhez, a célfüggvény csökkenése egy duál szimplex lépés után

  $$\min_{j \in N: d_{ij} > 0} \frac{d_{0j}}{d_{ij}} \{d_{i0}\},$$

  ahol N a nem-bázis változók halmaza.
  Ha az $x_i \geq \lceil x_i^* \rceil$ egyenlőtlenséget vesszük a rendszerhez, a célfüggvény csökkenése egy duál szimplex lépés után

  $$\min_{j \in N: d_{ij} < 0} \frac{d_{0j}}{d_{ij}} (\{d_{i0}\} - 1).$$

  Ezen értékek alapján választjuk meg, hogy melyik változó szerint választunk szét. Pl. választhatjuk azt, amelyikre a két érték minimuma maximális.

  Megjegyezzük, hogy a jó szétválasztás a fa gyökerének a közelében igazán fontos. Ha a feladatban vannak fontos és kevésbé fontos változók, akkor a fontosabbak szerint érdemes először szétválasztani. Egyébként szétválasztás nem csak változó szerint lehetséges, hanem más lineáris feltétel szerint is: $dx \leq \lfloor dx^* \rfloor$ illetve $dx \geq \lceil dx^* \rceil$ feltétel hozzávételével.

- **Hogyan válasszuk ki a jelölt-listáról a vizsgálandó feladatot?** Itt is több módszer lehetséges, a hasznosságuk a feladat típusától függ.

  – **Best first:** A legmagasabb felső korlátot adó részfeladatot választjuk. Ennek az az előnye, hogy az algoritmus során csak olyan részfeladatokat vizsgálunk meg, amiket mindenképpen meg kell vizsgálnunk. Hátránya, hogy egyrészt nem kapunk hamar megengedett megoldást, másrészt össze-vissza ugrálunk a fában, így nem tudjuk kihasználni az egymás után vizsgált feladatok hasonlóságát.

  – **Depth first:** Olyan mélyre megyünk a fában, amilyen mélyre csak tudunk. Itt az egymás után vizsgált részfeladatok hasonlóak és hamar megengedett megoldást kapunk, de az össz-lépésszám sokkal több lehet mint Best first-nél.

5.2.2. Korlátozás ügyesebben

A leírt algoritmusban a felső korlátokat egyszerűen az LP relaxáció optimális megoldása adta. Ezen többféle módon is lehet javítani. Néhány lehetőség a részletek ismertetése nélkül:

- **Preprocessing:** Az IP feladatunkat leíró lineáris rendszer sokszor jelentősen egyszerűsíthető (pl. redundáns feltételek elhagyása, változók számának csökkentése lineáris összefüggések segítségével, logikai implikációk hozzáadása). Ilyen átalakításokat minden részfeladatnál elvégezhetünk, mielőtt kiszámoljuk a felső korlátot.

- **Vágás és szétválasztás:** Vágás típusú algoritmusokat szubrutinként beépíthetünk a korlátozás és szétválasztás módszerébe, a következőképpen: amikor egy részfeladathoz jutunk, az őt leíró lineáris rendszeren futtatunk meghatározott ideig egy vágósíkos algoritmust. Az így kapott jobb lineáris rendszert használjuk a felső korlát kiszámítására és szétválasztás esetén az új részfeladatok definiálására.

- **Alsó korlátok számolása:** Sokszor az egyes részfeladatoknál a felső korlát kiszámolása mellett alsó korlátokat is érdemes számolni. Ezt valamilyen egyszerű heurisztikával tehetjük meg (mohó algoritmus, kerekítés, lokális keresés).

5.2.3. Alkalmazás a bináris hátizsák-feladatra

Nézzünk egy bináris hátizsák-feladatot:

$$\max\{\sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_j x_j \leq b,\ x \in \{0,1\}^n\}.$$

A feladatnak két tulajdonsága van, ami miatt a korlátozás és szétválasztás módszere egyszerűsödik:

- A hátizsák-feladat LP relaxációja nagyon egyszerűen megoldható: fajlagos érték szerinti sorrendben kell betenni a tárgyakat, és legfeljebb egy lesz, aminek csak törtrésze fér be, tehát az optimális megoldásban legfeljebb egy törtértékű változó lehet.

- Ha bizonyos változók értékét 0-ra vagy 1-re rögzítjük, továbbra is bináris hátizsákfeladatunk lesz: b-ből le kell vonni az 1-re rögzített változók súlyát.

A korlátozás és szétválasztás algoritmusra nézve ez a következőket jelenti:

- Nagyon gyorsan meghatározható az a változó, ami szerint szétválasztunk. Ráadásul az LP optimumból a részben bevett tárgy elhagyásával egészértékű megoldást kapunk, amivel esetleg lecserélhetjük az eddig talált legjobb megoldást és növelhetjük az alsó korlátot.

- A szétválasztás során keletkezett feladatok is hátizsákfeladatok, hiszen egy változót rögzítünk 0-ra vagy 1-re. Így az LP minden részfeladatnál könnyen megoldható lesz, sőt egyre könnyeb ahogy egyre kevesebb a rögzítetlen változó.

Bináris változónál történő szétválasztásnál

5.3. Lagrange-relaxáció

Bevezető.

5.3.1. Lagrange-duális feladat

Tekintsük a következő feladatot.

$$\begin{aligned} z_{IP} := \max cx \\ Ax \leq b \\ Dx \leq d \\ x \text{ egész,} \end{aligned} \tag{5.8}$$

ahol az A, D, b, d, c komponensei egészek. Vezessük be a következő jelölést:

$$X := \{x \in \mathbb{Z}^n : Dx \leq d\}.$$

A most következő eljárás motivációjaként tegyük fel, hogy az X halmazon könnyen ki tudjuk számítani tetszőleges lineáris célfüggvény optimumát (például minimális súlyú feszítőfa, vagy hátizsákfeladat), ámde az $Ax \leq b$ feltételek megléte mellett a feladat már nehézzé válik (például utazóügynök feladattá). Az $Ax \leq b$ feltétel elhagyásával is relaxáljuk a feladatot, de ne elégedjünk meg most ennyivel, valamelyest ügyeljünk a teljesülésére. Tegyük a következőt: legyen $\lambda \geq 0$ egy, a b dimenziójával megegyező dimenziójú vektor, és a rögzítése mellett tekintsük a következő, λ-paraméterű *Lagrange-feladatot*:

Lagrange–feladat

$$L(\lambda) := \max cx + \lambda(b - Ax) \tag{5.9}$$
$$x \in X.$$

Azt mondjuk, hogy az $Ax \leq b$ feltételeket relaxáltuk, és beépítettük a célfüggvénybe.

5.3. lemma. *Amennyiben a (5.8) feladatnak van optimális megoldása, akkor tetszőleges $\lambda \geq 0$ esetén $z_{IP} \leq L(\lambda)$.*

Bizonyítás. Trivi. □

Az előző lemma fényében érdemes megkeresnünk a legjobb felső korlátot, azaz tekintsük a következő, *Lagrange-duális* feladatot:

Lagrange-duális feladat

$$z_{LD} := \min L(\lambda) \tag{5.10}$$
$$\lambda \geq 0.$$

Amenyiben $X = \{x^1, \ldots x^m\}$ véges halmaz, akkor az $L : \mathbb{R}^m \to \mathbb{R}$ függvény a következő alakban is írható.

$$L(\lambda) = \max_{i=1,\ldots m} (cx^i + \lambda(b - Ax^i)). \tag{5.11}$$

Vegyük észre, hogy az L függvény lineáris függvények ún. felső burkolója, következésképp konvex és „szakaszonként lineáris”[1] Mindebből az is következik, hogy a nagyon sok feltétellel rendelkező lineáris program optimuma z_{LD}!

$$\min \beta \tag{5.12}$$
$$\beta \geq cx^i + \lambda(b - Ax^i) i = 1, \ldots m.$$

A 5.3 lemma értelmében $z_{IP} \leq z_{LD}$ teljesül, ámde példa adható arra, amikor a szigorú egyenlőtlenség áll. A következő fejezet témája a fölső korlát erőssége.

Könnyen látható, hogy amennyiben a relaxálandó feltételek között szerepel $ax \leq \gamma$ és $-ax \leq -\gamma$, akkor ekvivalens módon járunk el, ha az $ax = \gamma$ feltételt tetszőleges előjelű együtthatóval építjük a célfüggvénybe.

[1] Szögletes parabola, ahogyan a diáknyelv szellemeskedi.

5.3.2. A Lagrange-duális feladat erőssége

A következő tétel azt mutatja meg, milyen erős is a Lagrange-duális.

5.4. tétel (Geoffrion, 1974)**.** *A Lagrange-duális feladat z_{LD} optimuma megegyezik a következő lineáris program optimumával:*

$$\begin{aligned} \max cx& \\ Ax \le b& \\ x \in \operatorname{conv}(X).& \end{aligned} \tag{5.13}$$

Bizonyítás.

$$\begin{aligned} z_{LD} &= \min_{\lambda \ge 0} \max\{cx + \lambda(b - Ax) \mid x \in X\} \\ &= \min_{\lambda \ge 0} \max\{cx + \lambda(b - Ax) \mid x \in \operatorname{conv}(X)\} \end{aligned}$$

Ha $X = \emptyset$, akkor a belső maximum $-\infty$ minden $\lambda \ge 0$ esetén, és ekkor $z_{LD} = -\infty$. Különben legyenenek $x^k \in \mathbb{R}^n$, $k \in K$, és $r^j \in \mathbb{R}^n$, $j \in J$, a $\operatorname{conv}(X)$ poliéder csúcsai és extrém irányai. Ekkor a belső maximum

$$\max\{cx + \lambda(b - Ax) \mid x \in \operatorname{conv}(X)\} = \begin{cases} \infty & \text{ha } (c - \lambda A)r^j > 0 \text{ valamely } r^j\text{-re} \\ cx^k + \lambda(b - Ax^k) & \text{valamely } x^k \text{ csúcsra} \end{cases}$$

Ebből következik, hogy ha z_{LD} véges, akkor

$$\begin{aligned} z_{LD} = \min_{\lambda} & \left\{ \max_{k \in K} cx^k + \lambda(b - Ax^k) \right\} \\ & (c - \lambda A)r^j \le 0, \ \forall j \in J \\ & \lambda \ge 0 \end{aligned}$$

Ezt átírhatjuk egy lineáris programra

$$\begin{aligned} z_{LD} = \min_{\beta, \lambda} & \ \beta \\ & \beta \ge cx^k + \lambda(b - Ax^k), \forall k \in K \\ & (c - \lambda A)r^j \le 0, \ \forall j \in J \\ & \lambda \ge 0 \end{aligned}$$

A lineáris programozás dualitás tételét használva kapjuk, hogy

$$\begin{aligned} z_{LD} = \max \ & c\left(\sum_{k \in K} u_k x^k + \sum_{j \in J} v_j r^j\right) \\ & \sum_{k \in K} u_k = 1 \\ & A\left(\sum_{k \in K} u_k x^k + \sum_{j \in J} v_j r^j\right) \le b\left(\sum_{k \in K} u_k\right) \\ & u_k, v_j \ge 0, \forall k \in K, \forall j \in J. \end{aligned}$$

Ez utóbbi viszont éppen $\max\{cx \mid Ax \le b, x \in \operatorname{conv}(X)\}$. □

Az eredeti feladat lineáris programozási relaxáltjának nevezzük a következő feladatot.

$$z_{LP} := \max cx \tag{5.14}$$

$$Ax \leq b$$

$$Dx \leq d$$

$$x \in \mathbb{R}^n.$$

A 5.4 tétel következménye az alábbi.

5.5. következmény. *$z_{IP} = z_{LD}$ minden c költségfüggvényre pontosan akkor teljesül, ha*

$$\mathrm{conv}(X \cap \{x : Ax \leq b\}) = \mathrm{conv}(X) \cap \{x : Ax \leq b\},$$

$z_{LP} = z_{LD}$ minden c költségfüggvényre pontosan akkor teljesül, ha

$$\mathrm{conv}(X) = \{x : Dx \leq d\}.$$

5.3.3. A szubgradiens-módszer

Az alábbiakban ismertetjük a szubgradiens-módszert „szakaszonként lineáris" konvex függvény minimumának megkeresésre, aztán megmutatjuk, hogy ez hogyan alkalmazható a Lagrange-duális feladat optimumának megkeresésére.

A következő lemmát bizonyítás nélkül felhasználjuk.

5.6. lemma. *Az $f : \mathbb{R}^n \to \mathbb{R}$ függvény pontosan akkor konvex, ha tetszőleges $\lambda^* \in \mathbb{R}^n$ ponthoz létezik olyan $s \in \mathbb{R}^n$ vektor, melyre*

$$f(\lambda) \geq f(\lambda^*) + s(\lambda - \lambda^*)$$

teljesül minden $\lambda \in \mathbb{R}^n$ esetén.

A lemmában szereplő s vektort az f λ^*-beli *szubgradiensének* nevezzük. Az λ^*-beli szubgradiensek halmazát pedig $\partial f(\lambda^*)$ jelöli.

(Amennyiben f differenciálható λ^*-ban, könnyen látható, hogy $\partial f(\lambda^*) = \{\nabla f(\lambda^*)\}$.)

A továbbiakban tekintsük a következő $\mathbb{R}^n \to \mathbb{R}$ lineáris függvényeket:

$$\lambda \mapsto h_i + f_i\lambda,$$

ahol $h_i \in \mathbb{R}$, $f_i \in \mathbb{R}^n$, és $i = 1, 2, \ldots m$.

Ezen függvények felső burkolója:

$$Z(\lambda) := \max_{i=1,\ldots m} h_i + f_i\lambda.$$

Legyen $E(\lambda) := \{i : Z(\lambda) = h_i + f_i\lambda\}$. Világos, hogy Z konvex függvény. Z szubgradienseiről szól az alábbi.

5.7. lemma. *Minden $i \in E(\lambda^*)$ index esetén $f_i \in \partial Z(\lambda^*)$. Továbbá $\partial Z(\lambda^*) = \mathrm{conv}(f_i : i \in E(\lambda^*))$.*

Bizonyítás. Minden $i \in E(\lambda^*)$ index esetén $f_i \in \partial Z(\lambda^*)$, hiszen

$$Z(\lambda) >= h_i + f_i\lambda = h_i + f_i\lambda^* + f_i(\lambda - \lambda^*) = Z(\lambda^*) + f_i(\lambda - \lambda^*).$$

Másrészt be kell látnunk, hogy $\partial Z(\lambda^*) = \mathrm{conv}(f_i \ : \ i \in E(\lambda^*)$. Indirekt tegyük fel, hogy nem, azaz van olyan $s \in \partial Z(\lambda^*)$, ami nem írható fel az f_i, $i \in E(\lambda^*)$ vektorok konvex kombinációjaként. Ez azzal ekvivalens, hogy a következő lineáris egyenletrendszernek nincs megoldása

$$\sum_{i \in E(\lambda^*)} \mu_i \begin{pmatrix} f_i \\ 1 \end{pmatrix} = \begin{pmatrix} s \\ 1 \end{pmatrix},$$

$$\mu \geq 0.$$

A Farkas Lemma megfelelő alakját használva ebből következik, hogy van olyan v vektor és v_0 skalár, hogy

$$\begin{aligned} f_i v + v_0 &\leq 0, \quad i \in E(\lambda^*) \\ sv + v_0 &> 0. \end{aligned}$$

Igen ám, de ekkor $f_i v \leq -v_0 < sv$ minden $i \in E(\lambda^*)$ indexre. A (v, v_0)-t átskálázva (megszorova egy kicsi pozitív számmal) olyan $\lambda = \lambda^* + v$ vektort definiálhatunk, amelyre $Z(\lambda) = h_i + f_i\lambda$ teljesül legalább egy $i \in E(\lambda^*)$ indexre (itt használjuk ki, hogy $Z(\lambda)$ szakaszosan lineáris. Ekkor azonban

$$Z(\lambda) = h_i + f_i\lambda = h_i + f_i\lambda^* + f_i(\lambda - \lambda^*) = Z(\lambda^*) + f_i v < Z(\lambda^*) + sv,$$

ami ellentmond annak, hogy $s \in \partial Z(\lambda^*)$. □

Most megadjuk az általános szubgradiens algoritmust, amely az f függvény minimumát határozza meg.

Általános szubgradiens-algoritmus

- Válasszunk egy kezdő λ^1 pontot.
- Iteratív lépés: $\lambda^{i+1} := \lambda^i - \alpha_i s^i$, ahol $s^i \in \partial f(\lambda^i)$, α_i pedig egy pozitív lépéshossz.
- Leállási kritérium: $0 \in \partial f(\lambda^i)$, azaz minimumhelyhez jutottunk.

A fenti algoritmus alkalmazásaiban a leállási kritérium ritkán teljesül, általában elegendő idő elteltével fejeződik be az algoritmus. Az α_i lépéshosszt a konvergencia lehetőségének megadása miatt célszerű egyre csökkenteni. (Amennyiben $\sum_{i=1}^{\infty} \alpha_i$ divergens, α_i pedig a 0-hoz konvergál, megmutatható, hogy a fenti algoritmus konvergálni fog az f minimumhelyéhez. Ámde a tapasztalat szerint ekkor elég lassú a konvergencia.)

A továbbiakban megadjuk szubgradiens-módszer (5.11) minimalizálására szolgáló módosítását.

Szubgradiens-algoritmus

- Válasszunk egy kezdő λ^1 értéket.
- Iteratív lépés: $\lambda^{i+1} := \max(\lambda^i - \alpha_i(b - Ax(\lambda^i)), 0)$, ahol $x(\lambda^i)$ a (5.11) feladat optimális megoldása, α_i pedig egy pozitív lépéshossz.

Az algoritmus módosítására azért volt szükség, mert az L csak nemnegatív vektorokon van értelmezve. Ezért az iteratív lépésben a negatív koordinátákra való változtatást egyszerűen 0-ra való változtatással kerüljük el. Meg kell még mutatnunk, hogy valóban egy szubgradienssel ellentétes irányba lépünk.

5.8. állítás. $b - Ax(\lambda^i) \in \partial L(\lambda^i)$.

Bizonyítás. Tekintsük a (5.11) Lagrange függvényt. $\square$

5.4. Benders dekompozíció

A Benders dekompozíció (Benders, 1962) főleg olyan vegyes programozási feladatoknál használható jól, ahol kevés egészértékű változó van. A módszer bizonyos értelemben a Lagrange relaxáció „duálisa”: míg ott az egyelőtlenségeket bontottuk egy könnyebb és egy nehezebb részre, itt a változókkal tesszük ezt.

A következő feladatot szeretnénk megoldani:

$$\begin{aligned} \min cx + fz \\ Ax + Bz \geq b \\ x \in \mathbb{R}^n_+ \\ z \in \mathbb{Z}^d \end{aligned}$$

Rögzített $\overline{z} \in \mathbb{Z}^d$ vektorra tekinthetjük a következő $LP(\overline{z})$ alfeladatot:

$$\begin{aligned} \min cx + f\overline{z} \\ Ax \geq b - B\overline{z} \\ x \in \mathbb{R}^n_+ \end{aligned}$$

A duális $D(\overline{z})$ feladat:

$$\begin{aligned} \max y(b - B\overline{z}) + f\overline{z} \\ yA \leq c \\ y \in \mathbb{R}^m_+ \end{aligned}$$

Legyen $Q = \{y \in \mathbb{R}^m_+ : \ yA \leq c\}$ a megengedett duális megoldások halmaza; vegyük észre, hogy ez nem függ $\overline{z}$-tól. Feltesszük, hogy Q nemüres. Legyenek $q_1, \ldots, q_k$ a Q csúcsai, és $r_1, \ldots, r_k$ a Q extrém irányai (csúcs biztos van, extrém irány nem feltétlenül). Ha egy r_i extrém irányra $r_i(b - B\overline{z}) > 0$, akkor a $D(\overline{z})$ feladat célfüggvényértéke nem korlátos, tehát az $LP(\overline{z})$ alfeladat nem megoldható.

Ha viszont nem létezik ilyen r_i, akkor az $LP(\overline{z})$ alfeladat optimumértéke $\max_{i=1,\ldots,k} q_i(b - B\overline{z})$. Tehát a következő úgynevezett *Benders főfeladat* ekvivalens az eredeti feladatunkkal:

$$\min \alpha \tag{5.15}$$
$$fz + q_i(b - Bz) \leq \alpha \qquad (i = 1, \ldots, k) \tag{5.16}$$
$$r_i(b - Bz) \leq 0 \qquad (i = 1, \ldots, l) \tag{5.17}$$
$$z \in \mathbb{Z}^d \tag{5.18}$$

A (5.16) egyenlőtlenségeket optimalitási feltételeknek, míg a (5.17) egyenlőtlenségeket megengedettségi feltételeknek nevezzük. *Csökkentett főfeladatról* beszélünk, ha az optimalitási és megengedettségi feltételek egy részét kihagyjuk a rendszerből. A Benders módszer úgy működik, hogy kiindulunk egy csökkentett főfeladatból amit könnyen meg tudunk oldani, és utána egyesével veszünk hozzá optimalitási vagy megengedettségi feltételeket. Ez e következőképpen történik.

1. Megoldjuk a csökkentett főfeladatot. Optimális megoldás: $(\overline{z}, \overline{\alpha})$. Figyeljük meg, hogy $\overline{\alpha}$ alsó korlát az eredeti feladat optimumára.
2. Megoldjuk az $LP(\overline{z})$ és $D(\overline{z})$ feladatokat.
   - Ha $LP(\overline{z})$-nak az optimumértéke $\overline{\alpha} - f\overline{z}$, kész vagyunk.
   - Ha az optimumérték szigorúan nagyobb mint $\overline{\alpha} - f\overline{z}$, legyen q_i a $D(\overline{z})$ feladat optimális csúcsa. Ekkor a főfeladatban a $(\overline{z}, \overline{\alpha})$ megoldás nem teljesíti a q_i-hez tartozó optimalitási feltételt. Vagyük ezt hozzá a csökkentett főfeladathoz.
   - Ha $LP(\overline{z})$-nak nincs megoldása, akkor $D(\overline{z})$ nem korlátos, tehát van r_i, amire $r_i(b - B\overline{z}) > 0$. Ekkor a főfeladatban a $(\overline{z}, \overline{\alpha})$ megoldás nem teljesíti az r_i-hez tartozó megengedettségi feltételt. Vagyük ezt hozzá a csökkentett főfeladathoz.
3. Folytassuk az algoritmust az új csökkentett főfeladattal.

5.9. tétel. *Ha a kiinduló vegyes-egészértékű program optimuma véges, akkor az eljárás véges sok lépésben megtalálja azt. Mivel*

Megjegyzés: az algoritmus közelítő algoritmusként is jól használható, hiszen ha az $LP(\overline{z})$ alfeladat optimumértéke közel van $\overline{\alpha}$-hoz, akkor az optimumhoz is közel van.

Alkalmazás a Szolgáltató-elhelyezési feladatra

Tekintsük a Szolgáltató-elhelyezési feladatot (1.5. fejezet) abban az esetben, ha a telephelyek tetszőlegesen nagy kapacitásúak lehetnek:

$$\min \sum_{i\in M}\sum_{j\in N} c_{ij}x_{ij} + \sum_{j\in N} f_j z_j \tag{5.19}$$

feltételek

$$\sum_{j\in N} x_{ij} = 1, \quad i \in M \tag{5.20}$$

$$-x_{ij} + z_j \geq 0, \quad i \in M, j \in N \tag{5.21}$$

$$x \in \mathbb{R}_+^{mn}, \ z \in \{0,1\}^n \tag{5.22}$$

Rögzített $\bar{z} \in \{0,1\}^n$ mellett tekintsük az $LP(\bar{z})$ feladatot:

$$LP(\bar{z}) = \min \sum_{i\in M}\sum_{j\in N} c_{ij}x_{ij} + \sum_{j\in N} f_j \bar{z}_j \tag{5.23}$$

feltételek

$$\sum_{j\in N} x_{ij} = 1, \quad i \in M \tag{5.24}$$

$$-x_{ij} \geq -\bar{z}_j, \quad i \in M, j \in N \tag{5.25}$$

$$x \in \mathbb{R}_+^{mn} \tag{5.26}$$

Ez szétesik $|M|$ darab, egymástól független LP-re:

$$LP^i(\bar{z}) = \min \sum_{j \in N} c_{ij} x_{ij} \tag{5.27}$$

feltételek

$$\sum_{j \in N} x_{ij} = 1, \tag{5.28}$$

$$-x_{ij} \geq -\bar{z}_j, \quad j \in N \tag{5.29}$$

$$x_{ij} \geq 0, \quad j \in N \tag{5.30}$$

Adott $i \in M$ esetén $LP^i(\bar{z})$ duális feladata:

$$\max u_i - \sum_{j \in N} v_{ij} \bar{z}_j \tag{5.31}$$

$$u_i - v_{ij} \leq c_{ij}, \quad j \in N \tag{5.32}$$

$$v_{ij} \geq 0, \quad j \in N \tag{5.33}$$

A $Q_i = \{(u_i, v) \in \mathbb{R}^{n+1} \mid v \geq 0,\ u_i - v_{ij} \leq c_{ij},\ \forall j \in N\}$ poliédernek a csúcsai ($u_i^k = c_{ik}$, $v_{ij}^k = (c_{ik} - c_{ij})^+$, $j \in N$) alakúak, ahol $k \in N$. Ezen túl van egy végtelen iránya is: $(u_i = 1, v_{ij} = 1, j \in N)$.

A végtelen irány alapján az $LP^i(\bar{z})$ megoldható, a.cs.a ha

$$1 - \sum_{j \in N} \bar{z}_j \leq 0. \tag{5.34}$$

A csúcsok pedig meghatározzák az optimum értékét, tehát

$$LP^i(\bar{z}) = \max_{k \in N} (c_{ik} - \sum_{j \in N} (c_{ik} - c_{ij})^+ \bar{z}_j). \tag{5.35}$$

Mindezeket összerakva kapjuk a Szolgáltató elhelyezési feladat Benders alakját:

$$\min \sum_{i \in M} \alpha_i + \sum_{j \in N} f_j z_j \tag{5.36}$$

feltételek

$$c_{ik} - \sum_{j \in N} (c_{ik} - c_{ij})^+ z_j \leq \alpha_i, \quad k \in N,\ i \in M \tag{5.37}$$

$$\sum_{j \in N} z_j \geq 1, \tag{5.38}$$

$$z \in \{0, 1\}^n. \tag{5.39}$$

Vegyük észre, hogy összesen $nm + 1$ feltételünk van, ami polinomiális az input hosszában.

5.5. Oszlopgenerálás

5.5.1. Többtermékes folyamok

Jól tudjuk, hogy a maximális folyamproblémára jobbnál jobb kombinatorikus algoritmusok adhatók, amelyek ráadásul még egészmegoldásokat is adnak, ha az élkapacitások egészek. Mi a helyzet több termék közös hálózatban való áramoltatásakor? Sajnos ez a feladat a legegyszerűbb esetekben is nehezen kezelhető, ha az egészértékű folyamok keresése a célunk.

Többtermékes (tört) folyam probléma

Adott a $G = (V, E)$ irányított gráf, az (s_i, t_i) pontpárok $i = 1, 2, \dots k$, és a nemnegatív c_e élkapacitások minden $e \in E$ élen. Továbbá adottak a d_i igények is minden i-re.
Minden i-re keressünk x^i d_i értékű folyamot úgy, hogy minden e élre $\sum_{i=1}^k x_e^i \leq c_e$.

Az (s_i, t_i) párok természetesen különbözőek, és a hozzájuk tartozó folyamot nevezzük az i-edik terméknek.

A probléma természetesen megfogalmazható irányítatlan gráfokban is.

Többtermékes egészértékű folyam probléma

A fenti problémában olyan folyamokat keresünk, melyek minden élen egész értéket vesznek fel, azaz x_e^i egész minden i-re és e élre.

Az egészértékű feladat NP-teljes, már két termék esetén minden élen $c_e = 1$ mellett is. Természetesen a sokrétű gyakorlati alkalmazások miatt igen sokféle közelítő és heurisztikus algoritmust kidolgoztak.

Lagrange relaxáció

Nézzük a feladat költséges változatát, amikor minimális költségű többtermékes folyamot keresünk. Tekinthetjük a feladatnak azt a Lagrange-relaxációját, ahol a kapacitás-feltételeket relaxáljuk és építjük be a célfüggvénybe. Az így kapott feladatban nincsenek felső korlátok, tehát az nem más mint k darab, egymástól független minimális költségű folyam feladat. Ráadásul mivel nincsenek felső korlátok, ezek a minimális költségű folyam feladatok valójában minimális költségű út kereséssel megoldhatók.

A 5.3. Fejezetben leírtak szerint a Lagrange duális feladat megoldása kiszámolható (legalábbis közelíthető) szubgradiens módszerrel. A szubgradiens kiszámolása a mi esetünkben egyszerű: meg kell nézni, hogy az egyes éleken mennyivel lépjük túl a kapacitást, ha a legrövidebb utakon szállítjuk a termékeket.

A módszer kicsit finomítható: ugyan a $\sum_{i=1}^k x_e^i \leq c_e$ feltételeket relaxáljuk, de az $x_e^i \leq c_e$ feltételeket benne hagyhatjuk a rendszerben. Ekkor a Lagrange feladat k darab független minimális költségű folyam feladat lesz, tehát nehezebb kiszámolni, viszont a szubgradiens algoritmus gyorsabban közelít a Lagrange duális feladat optimumához.

Mi lesz a Lagrange duális feladat optimuma? Mivel a relaxált rendszer egy egész poliédert ír le, a 5.4. Tétel alapján ez az optimum megegyezik az eredeti feladat LP relaxáltjának optimumával, azaz az optimális tört többtermékes folyam költségével.

Ford-Fulkerson algoritmus

A továbbiakban a súlyozatlan törtváltozat megoldására ismertetjük Ford és Fulkerson 1958-ból származó klasszikus oszlopgenerálós algoritmusát.

A törtváltozat tehát azt kérdezi, hogy az alábbi egyenlőtlenség-rendszernek van-e megoldása.

$$\begin{gathered}
\delta_{x^i}(v) - \varrho_{x^i}(v) = 0 \quad \forall i = 1, 2, \dots k, v \in V - \{s_i, t_i\} \qquad (5.40) \\
\delta_{x^i}(s_i) - \varrho_{x^i}(s_i) = d_i \quad \forall i = 1, 2, \dots k \\
\sum_{i=1}^k x_e^i \leq c_e \quad \forall e \in E
\end{gathered}$$

$$x_e^i \geq 0 \quad \forall i = 1, 2, \ldots k, e \in E.$$

Ezek szerint a fenti lineáris programot kell egy alkalmas módszerrel megoldanunk.

Ha k, valamint a gráf nagy, akkor a fenti LP nagyon sok változót és feltételt tartalmazhat, és a megoldása lassú lehet a szokásás módszerekkel (primál vagy duál simplex algoritmus). Ezen probléma leküzdésére Ford és Fulkerson meglepő módon egy még nagyobb lineáris programot írt fel! (Annyit előre elárulunk, hogy szerencsére magát a feladatot majd nem kell felírni, csak azokat a részeit, amikre szükség lesz.)

Amennyiben létezik d_i értékű x^i $(s_i - t_i)$-folyam, akkor létezik egy $y^i \leq x^i$ folyam is, amely előáll $(s_i - t_i)$-utak nemnegatív lineáris kombinációjaként.

Jelölje n_i az $(s_i - t_i)$-utak számát. Ekkor a következő feladat ekvivalens (5.40)-tel.

$$\sum_{i=1}^{k} \sum_{j=1}^{n_i} z_{ij} \chi_e^{P_{ij}} \leq c_e \qquad \text{minden } e \in E, \tag{5.41}$$

$$\sum_{j=1}^{n_i} z_{ij} = d_i \qquad \forall\ i,$$

$$z_{ij} \geq 0 \qquad \forall\ i, j,$$

ahol χ^P a P út élekre vonatkozó karakterisztikus vektorát jelöli.

Most fogalmazzuk meg maximalizálási feladatként (5.41)-t.

Adjunk a gráfhoz k darab új pontot $s'_1, s'_2, \ldots s'_k$, és k darab új irányított élet: $s'_1 s_1, s'_2 s_2, \ldots s'_k s_k$ rendre a $d_1, d_2, \ldots d_k$ kapacitásokkal. Erre a gráfra felírva a (5.41) feltételeket a változók nemnegativitása mellett keressük

$$\sum_{i=1}^{k} \sum_{j=1}^{n_i} z_{ij}$$

maximumát. Amennyiben ez a maximum $\sum_{i=1}^{k} d_i$, akkor az eredeti feladatnak is van megoldása, különben pedig nincs.

Ez tehát ekvivalens az alábbi lineáris programmal, ahol $\mathcal{P}$ jelöli az összes (s_i, t_i)-út halmazát.

$$\max \sum_{P \in \mathcal{P}} x_P.$$

$$\sum_{P \in \mathcal{P}} x_P \chi_e^P \leq c_e \qquad \forall\ e \in E, \tag{5.42}$$

$$x_P \geq 0 \qquad \forall\ P \in \mathcal{P}, \tag{5.43}$$

Adjunk minden kapacitási megszorítást jelentő sorhoz egy fölöslegváltozót, jelöljük A-val az él-út illeszkedési mátrixot, és $w \in \mathbb{R}^{E \times \mathcal{P}}$ pedig legyen az a vektor, mely minden úthoz tartozó komponensében 1, az élekhez tartozó komponensében pedig 0.

$$\max wx$$

$$[A\ I]x = c \tag{5.44}$$

$$x \geq 0$$

Tegyük fel, hogy $\mathcal{P}' \subseteq \mathcal{P}$ és $E' \subseteq E$ olyan részhalmazok, melyekre $|\mathcal{P}'| + |E'| = |E|$, és a hozzájuk tartozó B-vel jelölt $[A\ I]$ oszlopok lineárisan függetlenek. Jelölje B a kiválasztott indexeket. Ekkor a B bázishoz tartozó bázis-megoldásnak nevezzük azt az x-et, amely a nem B-beli indexekben 0, a többiben pedig a $Bx_B = c$ egyértelmű megoldása.

Most megmutatjuk, hogy a szimplex algoritmus hogyan valósítható meg oly módon, hogy ne kelljen felírnunk magát a teljes A mátrixot, amely oszlopainak a száma rettenetesen nagy (sok-sok út darabszáma).

(5.44) egy azonnal adódó nemnegatív bázis-megoldását adja a $B = E$ eset.

Egy adott nemnegatív bázis-megoldás vajon optimális-e? Ezt a következőképpen ellenőrizhetjük.

Először oldjuk meg a B-hez tartozó program $\min\{yc \mid yB \geq w_B\}$ duálisát, majd ellenőrizzük, hogy ez az y megengedett megoldása-e (5.44)-nak, mert ha igen, akkor az x_B optimális. ha pedig nem, akkor átlépünk egy másik bázisra, amelyhez tartozó egyértelmű megoldás nemnegatív marad, és iteráljuk az eljárást.

Kétféle duális feltétel van, hogyan ellenőrizhetjük ezek teljesülését? A B-ben benne nem lévő élekhez tartozó I-beli oszlopokból származó feltétel egyszerűen ezen y_e változó nemnegativitását mondja ki. Ezt könnyű ellenőrizni, és ha találunk negatív értékű változót, akkor az legyen a bázisba bemenő oszlop indexe, és majd megmondjuk, nemsokára, hogy ezt a bevitelt hogyan valósítjuk meg.

A B-ben nem lévő utak rengetegen vannak. Egyesével nem tudjuk ellenőrizni, hogy egy adott P útra az $y\chi^P$ skaláris szorzat vajon 1-nél nagyobb-egyenlő-e. De szerencsére egyszerre ez megtehető! Ugyanis számítsuk ki minden i termékre a legrövidebb (s_i, t_i)-utat az y élsúlyozás mellett. Ha mindegyik nagyobb egynél, akkor minden feltétel teljesül, ha pedig nem mind, akkor kapunk olyan egy P útat, amelyhez tartozó duális feltétel sérül, és egy ilyen úthoz tartozó oszlopot vigyük be ekkor a bázisba. Lényeges elem, hogy ekkor már túl vagyunk az y nemnegativitásának vizsgálatán, tehát a legrövidebb utak nemnegatív élsúlyokkal határozandók meg, ami könyűszerrel végrehajtható.

Hogyan határozzuk meg azt az indexet, amelynek el kell hagynia a bázist? Ha a bejövő index az e élhez tertozik, akkor oldjuk meg a $Bx = 1_e$ egyenletet, ha pedig a bejövő index egy úthoz tartozó P, akkor pedig a $Bx = \chi^P$ egyenletet. A $Bz = c$ megoldása után a szimplex módszernél tanult hányados-módszer segítségével kiválaszthatunk egy olyan indexet, amelyhez tartozó oszlop B-ből való törlésével és a meghatározott oszlop hozzávételével a $Bx = c$ egyértelmű megoldása ismételten (automatikusan) nemnegatív lesz.

Ezzel be is fejeztük a Ford és Fulkeerson szerzőpáros oszlopgenerálós algoritmusának a leírását. Az oszlopgenerálás elnevezés onnan ered, hogy nem tekintjük az összes oszlopot, csak azokat, amelyekre szükségünk van. Ezen fenti algoritmus a szimplex-módszerhez hasonlóan ciklizálhat, de ennek előfordulására vajmi kevés az esély, illetve a szimplex-módszernél alkalmazott lexikografikus választási szabály alkalmazásával ez itt is elkerülhető (ahogyan azt Chvátal 1983-ban bizonyította).

12. feladat. *Mutassuk meg, hogy a többtermékes folyam feladatnak a d_i igények mellett akkor és csak akkor van megoldása, ha minden $l \in \mathbb{R}_+^E$ hosszfüggvényre teljesül a következő*

$$\sum_{i=1}^{k} d_i \cdot dist_l(s_i, t_i) \leq \sum_{e \in E} l(e) \cdot c(e),$$

ahol $dist_l(s_i, t_i)$ az s_i-ből a t_i-be vezető l szerinti legrövidebb irányított út hosszát jelöli. (Útmutatás: Használjuk a Farkas-lemmát!)

5.5.2. Egydimenziós szabási feladat

Ennél a feladatnál W hosszúságú papírtekercseink vannak, amikből adott hosszúságú darabokat kell vágni, minél kevesebb tekercset felhasználva. Tegyük fel, hogy w_i hosszú papírból

b_i darabra van szükség ($i = 1, \dots, m$). Nevezzük az m különböző hosszú papírt típusoknak. Hogy tudjuk ezt a feladatot egészértékű programozási feladatként felírni?

Egy tekercset sokféleképpen lehet felvágni darabokra (nevezzük a lehetséges felvágásokat szabásmintáknak). Figyeljük meg, hogy a lehetséges felvágások pont egy egészértékű hátizsák feladat megoldásai, ahol a tárgyak súlya w_i ($i = 1, \dots, m$), a hátizsák teherbírása pedig W. Kézenfekvő volna tehát a hátizsákfeladathoz hasonló felírást keresni. Azonban most teljesen mást csinálunk: az összes lehetséges szabásmintához változót rendelünk!

Tegyük fel hogy n lehetséges szabásminta van. Legyen A az az $m \times n$-es mátrix, aminek egy adott szabásmintához tartozó oszlopában az szerepel, hogy melyik típusból hányat ad. Ez az A mátrix nagyon nagy, úgyhogy az algoritmus során sosem fogjuk használni, de a feladatot ennek segítségével írjuk fel:

$$\begin{aligned} \min \sum_{j=1}^{n} x_j& \\ Ax &\geq b \\ x &\geq 0 \\ x &\text{ egész.} \end{aligned}$$

Ennek a feladatnak a lineáris relaxáltját fogjuk megoldani az oszlopgenerálás módszerével:

$$\begin{aligned} \min \sum_{j=1}^{n} x_j& \\ Ax &\geq b \\ x &\geq 0. \end{aligned}$$

Ha a b_i számok viszonylag nagyok, akkor a lineáris relaxált optimális megoldásából általában könnyen kapható egy jó egészértékű megoldás. Probléma viszont, hogy a feladatnak nagyon sok változója van. Ezért először kiválasztunk néhány szabásmintát, és megnézzük hogy csak ezeket használva mi a legjobb megoldás. Legyen A' a kiválasztott szabásmintákhoz tartozó oszlopokból álló részmátrix. Ekkor a feladatunk:

$$\begin{array}{rll} & \min & \sum_{j=1}^{n} x_j \\ \text{Primál:} & A'x & \geq b \\ & x & \geq 0 \end{array} \qquad \begin{array}{rll} & \max & \sum_{i=1}^{m} b_i y_i \\ \text{Duál:} & yA' & \leq \mathbf{1} \\ & y & \geq 0. \end{array}$$

Ezt szimplex módszerrel meg tudjuk oldani, és kapunk egy optimális primál megoldást (x') és egy optimális duál megoldást (y').

Persze x' nem feltétlenül optimális megoldása az eredeti feladatunknak. Annyit tudunk hogy ha y' megoldása az eredeti feladat duálisának, tehát $ya \leq 1$ minden a szabásmintára, akkor x' optimális. Látszólag nem jutottunk előrébb, hiszen hogy tudnánk ezt az összes lehetséges szabásmintára ellenőrizni? Másszóval: hogy tudnánk megoldani a

$$\max\{ya : \ a \text{ egy lehetséges szabásminta}\}$$

feladatot?

A válasz az, hogy ez pont egy hátizsák feladat, hiszen így fogalmazható át (vigyázzunk, y itt most konstans, nem változó):

$$\max\{\sum_{i=1}^{m} y_i z_i : \ \sum_{i=1}^{m} w_i z_i \leq W, \ z \geq 0, \ \text{egész.}\}$$

Hátizsákfeladat megoldására ismerünk módszereket, ezek valamelyikét használva kapunk egy optimális z vektort, ami megegyezik az A mátrix valamelyik a oszlopával, hiszen A-ban az összes lehetséges szabásminta szerepelt. Ha $ya \leq 1$, akkor kész vagyunk, az x' optimális megoldás. Ha $ya > 1$, akkor hozzávesszük az a oszlopot az A' mátrixot, és ezzel az új mátrixszal megismméltejük az eljárást. Az eredmény a "nagy" LP egy optimális megoldása, amiben az x_j változók nem feltétlenül egészértékűek.

A tapasztalat azt mutatja, hogy ez az eljárás hatékony: nem túl sok iteráció után megtalálja az optimális (tört) megoldását az LP-nek.

5.6. Graver-féle teszthalmaz

Ezt a módszert akkor használhatjuk, ha sok különböző feladatot akarunk megoldani, amiknek a mátrixa ugyanaz, csak a változók korlátjában és a jobboldalban térnek el. Definiáljunk egy feladatosztályt:

$$IP(b, c, u) = \min\{cx : Ax = b,\ 0 \leq x \leq u,\ x \in \mathbb{Z}^n\}.$$

Jelölje O_j az $\mathbb{R}^n$ j-edik ortánsát ($j = 1, \ldots, 2^n$). Legyen $C_j = O_j \cap \{x \in \mathbb{R}^n : Ax = 0\}$, ez egy csúcsos racionális kúp, tehát van egy H_j egyértelmű minimális Hilbert-bázisa.

5.10. definíció. A $H = \cup_j H_j$ halmazt a feladatcsalád Graver teszthalmazának nevezzük.

A teszthalmaz elnevezést az indokolja, hogy egy adott megoldásról a teszthalmaz segítségével el tudjuk dönteni, hogy optimális-e, és ha nem, tudunk egy jobb megoldást találni. Erről szól a következő tétel.

5.11. tétel. *Ha az $IP(b, c, u)$ feladatnak egy x megoldása nem optimális, akkor létezik $h \in H$, amire $x + h$ jobb megoldás.*

Bizonyítás. Legyen x^* egy optimális megoldása $IP(b, c, u)$-nak. Ekkor $A(x^* - x) = 0$, $x^* - x \in \mathbb{Z}^n$, és $c(x^* - x) < 0$. Legyen O_j az $x^* - x$ vektort tartalmazó ortáns. Mivel $x^* - x$ a C_j egész eleme, előáll H_j-beli vektorok nemnegatív egész kombinációjaként: $x^* - x = \sum_{i_1}^{k} \lambda_i h^i$, ahol minden λ_i pozitív egész.

Mivel $c(x^* - x) < 0$, valamelyik i-re $ch^i < 0$. Ekkor $h = h^i$ teljesíti a tétel feltételeit: egyrészt $c(x + h) < cx$, másrészt $x_j^* - x_j \geq 0$ esetén $0 \leq h_j \leq x_j^* - x_j$, $x_j^* - x_j \leq 0$ esetén pedig $0 \geq h_j \geq x_j^* - x_j$, tehát $0 \leq x + h \leq u$ következik abból, hogy $0 \leq x, x^* \leq u$. □

Természetesen H lehet exponenciális méretű, tehát ez a módszer nem mindig hatékony. Ha H nem túl nagy, akkor is nehéz kiszámolni, de ezt csak egyszer kell megtenni, és utána a feladatosztály tetszőleges feladatára használhatjuk.

6. fejezet

Az utazóügynök feladat heurisztikái

Az utazóügynök feladata a $G = (V, E)$ n pontú teljes gráfban a $c : E \to \mathbb{R}_+$ élhosszak esetén a legrövidebb olyan körnek a megkeresése, amely a gráf minden pontján áthalad. Ezt a kört a gráf legrövidebb Hamilton-körének vagy optimális (utazóügynök-) túrájának fogjuk nevezni. Természetesen megfogalmazható az a feladat is, amikor az u pontból v-be haladva más az él hossza, mint a v-ből az u-ba haladva. A továbbiakban mi csak a fenti, úgynevezett szimmetrikus utazóügynök feladattal foglalkozunk, ráadásul még feltesszük azt is, hogy az élhosszakra (más szóval élsúlyozásra vagy élköltségekre) teljesül az háromszög-egyenlőtlenség is, azaz tetszőleges u, v, z pontokra igaz az alábbi:

$$c(uv) + c(vz) \geq c(uz).$$

> *Utazóügynök feladat*
>
> A $G = (V, E)$ teljes gráfban a $c : E \to \mathbb{R}_+$ élhosszak esetén a legrövidebb olyan körnek a megkeresése, amely a gráf minden pontján áthalad. Feltesszük, hogy c teljesíti a háromszög-egyenlőtlenséget.

Most tekintsünk néhány eljárást, amelyek nagyon gyorsan megadnak valamilyen túrát. Azt, hogy mennyire van közel az optimumhoz egy ilyen ún. heurisztika által megadott túra, többféle mérőszámmal is jellemezhetjük. Legkézenfekvőbbnek látszik a matematikai viselkedés, azaz megadni, hogy tetszőleges gráfra a heurisztika által adott megoldás sohasem lesz rosszabb, mint az optimum K-szorosa. Ez a matematikailag garantált K érték azonban a legtöbb algoritmus esetén nagyon távol áll a gyakorlatban felmerülő problémákra adott megoldások optimummal való hányadosától! A következő áthidaló megoldás született. Reinelt összeállított hozzávetőleg 100 tipikus gyakorlati feladatot a TSPLIB [11] könyvtárba, és a különféle heurisztikákat ezen feladatokra adott megoldásaik alapján jellemezhetjük úgy, hogy a TSPLIB mérőszámuk azon számok átlaga, ahányszorosát a TSPLIB-feladatok optimumának adják. Ez a mérőszám adja meg legjobban, hogy egy konkrét feladatra várhatóan milyen jól működik egy heurisztika, vagy akár a saját magunk által kidolgozott algoritmus.

6.1. A legközelebbi szomszéd heurisztikája

Keressünk meg egy legrövidebb élet, aztán ennek az egyik végpontjából elindulva lépjünk tovább mindig egy legközelebbi pontba, ahol még nem jártunk. Végül egy n-hosszú utat kapunk, és kényszerűségből lépjünk a kiindulópontunkba. Ennek az algoritmusnak a TSPLIB mérőszáma 1.26. (Esetleg indíthatjuk tetszőleges pontból is az algoritmust.)

(Ezen módszer nem meglepő módon igen kacifántos bejárásokat ad annak ellenére, hogy lokálisan a legjobb továbblépést választja. Nagyon könnyen kihagy ugyanis pontokat, amiket a túra utolsó pontjainak meghatározásakor csak igen nagy költségekkel jár be.)

6.2. Túranövelő módszerek

Az alábbi heurisztikák mind úgy működnek, hogy kiválasztanak valamilyen szabály szerint egy kiindulási élet, aztán hozzávesznek egy pontot, három pontból álló túrát kapva így. A továbbiakban, ha van már egy kisebb túránk, akkor valamilyen szabály szerint kiválasztják a következő beszúrandó pontot (ez a szabály különbözteti meg ezen algoritmusokat), és beszúrják a túrába azon két pont közé, amellyel a túra hossza a lehető legkevesebbel nő, azaz megkeresi azon két, a meglévő túrában egymás után következő u, v pontot, amelyre a beszúrandó s ponttal a következő érték minimális:

$$c(us) + c(vs) - c(uv).$$

- **Legközelebbi pont beszúrása.** Egy legrövidebb élből indulunk ki. Egy meglévő túrától egy adott pont távolsága a túra pontjaitól vett távolságok minimuma. A legközelebbi pont beszúrása azt jelenti, hogy a túrához legközelebbi pontot választjuk ki, és azt szúrjuk be (a fenti általános szabály szerint, azaz a legkisebb hossznövekedéssel). Ezen algoritmus TSPLIB mérőszáma: 1.??.
- **Legtávolabbi pont beszúrása.** Egy leghosszabb élből indulunk ki, és mindig a túrától legtávolabbi pontot szúrjuk be. TSPLIB: 1.16.
- **Legolcsóbb beszúrás.** Egy legrövidebb élből indulunk ki, és megnézzük, hogy a túrában nem szereplő pontok beszúrása mekkora túrahossz-növekedést eredményez. Azt a pontot szúrjuk be, amelyre ez minimális. TSPLIB: 1.??.
- **Véletlen pont beszúrása.** Véletlenül választjuk ki a soron következő beszúrandó pontot.

Talán meglepőnek tűnik, hogy a legtávolabbi pont beszúrása működik a legjobban a földi körülmények között. Erre magyarázat talán, hogy a végső túra hozzávetőleges alakját már elég korán megtaláljuk, és aztán már csak kicsi változások történnek. (Persze ennyire azt is meg tudnánk indokolni, ha bármely másik beszúrási módszer működne jobban a többinél.)

Legyen T egy részleges túrája G-nek (azaz túra G egy részgráfjában), és $v \notin T$ csúcs. Jelölje $\delta(T, v)$ a minimális túra-hossz növekedést a v csúcs hozzávételével.

6.1. lemma. *Legyen T egy részleges túrája G-nek, $v \notin T$, $u \in T$. Ekkor $\delta(T, v) \leq 2c(uv)$.*

Bizonyítás. Ha v az u és w csúcsok közé kerülne T-ben, akkor a túra hossza $c(vu) + c(vw) - c(uw)$-val növekszik. Mivel $c(vw) - c(uw) \leq c(uv)$ a háromszög-egyenlőtlenség miatt, $c(vu)+c(vw)-c(uw) \leq 2c(uv)$. Mivel a $\delta(T, v)$ a minimális növekedés mértéke, ezért $\delta(T, v) \leq c(vu) + c(vw) - c(uw) \leq 2c(uv)$. □

6.2. tétel (Rosenkrantz, Stearns, Lewis, 1977)**.** *A legközelebbi pont beszúrási heurisztikája által az utazóügynök feladatra adott megoldás hossza az optimális megoldás legfeljebb* 2*-szerese.*

Bizonyítás. Prim feszítőfa algoritmusát (11.3 fejezet) használjuk. Megmutatjuk, hogy a Prim algoritmusa által adott F feszítőfa és a vizsgált heurisztika által szolgáltatott T túra hosszára teljesül, hogy $c(T) \leq 2c(F)$. Mivel az optimális túra hosszánál nem lehet hosszabb egy minimális feszítőfa hossza, ezzel a tételünket be is bizonyítjuk.

Belátjuk, hogy Prim algoritmusa és a heurisztika párhuzamos lefutása közben a $c(T') \leq 2c(F')$ összefüggés mindig fennáll az azonos pontszámú T' résztúra és F' fa hossza között. A kiindulási három pontú fára és körre nyilván igaz, ha pedig egy résztúrára teljesül, akkor vegyük észre, hogy a fa és a résztúra ponthalmaza megegyezik, az új pont hozzávételekor pedig a háromszög-egyenlőtlenség szerint legfeljebb kétszerannyival nő a túra hossza, mint a fáé. Ugyanis, az előző lemma szerint $\delta(T', v) \leq 2c(u, v)$, ahol v a beszúrandó pont, és $u \in T'$ az a közös pont a túrában, és a fában, ami v-hez a legközelebb van. □

6.3. tétel (Rosenkrantz, Stearns, Lewis, 1977)**.** *Tetszőleges beszúrási módszer által az utazóügynök feladatra adott megoldás hossza az optimális megoldás legfeljebb* $(\lceil \log_2 n \rceil + 1)$*-szerese.*

A tétel bizonyításához több állításon keresztül vezet az út.

6.4. lemma. *A* $G = (V, E)$ *minden csúcsához hozzárendelünk egy* $\ell_v \geq 0$ *számot, amely teljesíti a következ? két feltételt:*

(i) $c(uv) \geq \min\{\ell_u, \ell_v\}$,

(ii) $\ell_u \leq \frac{1}{2} OPT$.

Ekkor $\sum_{v \in V} l_v \leq \frac{1}{2}(\lceil \log_2 n \rceil + 1) OPT$.

Bizonyítás. A csúcsok átsorszámozásával feltehetjük, hogy $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_n$. Legyen $H_k = (V_k, E_k)$ részgráfja G-nek, ahol $V_k = \{1, \ldots, \min\{2k, n\}\}$, és E_k a V_k csúcsok által feszített összes él.

Legyen T egy túra H_k-ban, amely ugyanolyan sorrendben látogatja végig a H_k csúcsait, mint egy G-feletti optimális túra. Ekkor $c(T) \leq OPT$ a háromszög egyenlőtlenség miatt.

Továbbá léteznek olyan $\alpha_v \in \{0, 1, 2\}$, $v \in V_k$, számok, amelyekre $\sum_{v \in V_k} \alpha_v = |V_k|$, és $c(T) \geq \sum_{v \in V_k} \alpha_v \ell_v$. Ugyanis

$$c(T) = \sum_{uv \in T} c(uv) \geq \sum_{uv \in T} \min\{\ell_u, \ell_v\} = \sum_{v \in V_k} \alpha_v \ell_v,$$

ahol az első egyenlőtlenség az (i) feltételből következik, és a második peddig abból, hogy egy $v \in V_k$ csúcshoz a túrának 2 éle illeszkedik.

Mivel $\sum_{v \in V_k} \alpha_v = |V_k|$, ezért

$$\sum_{v \in V_k} \alpha_v \ell_v \geq \sum_{i=k+1}^{\min\{2k, n\}} 2\ell_i.$$

Legyen $k = 2^j$. Ekkor kapjuk, hogy

$$\sum_{j=0}^{\lceil \log_2 n \rceil - 1} OPT \geq \sum_{j=0}^{\lceil \log_2 n \rceil - 1} 2 \sum_{i=2^j+1}^{\min\{2^{(j+1)}, n\}} \ell_i \geq \sum_{i=2}^{n} 2\ell_i.$$

Mivel $\ell_1 \leq \frac{1}{2} OPT$ az (ii) feltétel miatt, kapjuk a

$$(\lceil \log_2 n \rceil + 1) OPT \geq \sum_{i=1}^{n} 2\ell_i,$$

és ezt akartuk bizonyítani. □

Most már jöhet a 6.3. Tétel bizonyítása:

Legyen $v_1, v_2, \ldots, v_n$ a csúcsok beszúrási sorrendje egy T túra építése során, (v_1, v_2) a kezdeti él. Jelölje T_i az első i csúcs beszúrásával nyert részleges túrát, azat T_1 egy hurokél, T_2 két párhuzamos él, stb. Legyen $\ell_{v_i} = \frac{1}{2}\delta(T_{i-1}, v_i)$, $i = 2, \ldots, n$, és $\ell_{v_1} = 0$. Megmutatjuk, hogy az ℓ_v súlyok teljesítik a 6.4. Lemma feltételeit.

Az ℓ_{v_1} nyilván teljesíti az (i) feltételt, mivel $c(uv_1) \geq 0 = \ell_{v_1}$. Másrészt, mivel $v_i \notin T_{i-1}$, és beszúráskor a v_i csúcsot a legkisebb hossznövekedés alapján illesztjük a T_{i-1} részleges túrába, ezért $\delta(T_{i-1}, v_i) \leq 2c(v_i u)$ minden $u \in T_i$, a Lemma 6.1 alapján. Tehát $\ell_{v_i} \leq c(v_i u)$, amiből következik, hogy $c(v_i u) \geq \ell_{v_i} \geq \min\{\ell_{v_i}, \ell_u\}$. Tehát az (i) teljesül minden $\{v_j, v_i\}$ élre, ahol $j < i$. Mivel az összes csúcs sorra kerül a túra építése során, ezért (i) teljesül.

Most nézzük az (ii) feltételt. $\ell_{v_1} = 0$ miatt v_1 teljesíti. v_2 esetében $\delta(T_1, v_2) = 2c(v_1, v_2) \leq OPT$, mivel az optimális túra tartalmazza a v_1 és v_2 csúcsokat valamilyen sorrendben, tehát a háromszög egyenlőtlenség miatt a v_1 -ből a v_2-be vezető darabja nem rövidebb, mint $c(v_1, v_2)$, és a v_2-ből a v_1-be vezető része nem rövidebb, mint $c(v_2, v_1)$. Mivel $c(v_1, v_2) = c(v_2, v_1)$, így $2c(v_1, v_2) \leq OPT$. Általában $\delta(T_{i-1}, v_i) = c(v_j v_i) + c(v_i v_k) - c(v_j v_k)$, valamely $v_j, v_k \in T$ csúcsokra (amelyek közé beszúrtuk v_i a túra építése során). Csakhogy az optimális túra hossza felső becslés a $c(v_k v_i) + c(v_i v_j)$ összegre, mivel mindhárom csúcsot az optimális túra végiglátogatja, és a háromszög egyenlőtlenségből következik a felső becslés. De ez azt jelenti, hogy $\delta(T_{i-1}, v_i) \leq OPT$, amiből következik, hogy $\ell_{v_i} \leq \frac{1}{2}OPT$. $\square$

6.3. Túrajavító módszerek

Ha már ismerünk egy Hamilton-kört, akkor abból könnyen kaphatunk újabbakat is, hátha rövidebbekhez juthatunk.

6.5. definíció. Egy T túrát *2-optimálisnak* nevezünk, ha tetszőleges uv, ab élpárjára teljesül, hogy $c(uv) + c(ab) \geq c(ua) + c(bv)$, ahol $T - \{uv, ab\} \cup \{ua, bv\}$ szintén túra.

Könnyen látható, hogy ha $T - \{uv, ab\} \cup \{ua, bv\}$ nem túra (azaz két diszjunkt kör), akkor $T - \{uv, ab\} \cup \{ub, av\}$ viszont túra.

A 2-opt heurisztika úgy működik, hogy mevizsgáljuk, hogy az adott T túra 2-optimális-e. Ha nem, akkor azonnal a rövidebb túrával folytatjuk az eljárást. Annak az eldöntése, hogy egy T túra 2-optimális-e, $O(n^2)$ időt igényel, ám a fenti algoritmus mégsem polinomiális, azaz nem feltétlenül áll le polinomiális idő alatt. TSPLIB: 1.06.

A fentiekhez hasonlóan meg tudunk fogalmazni k-optimális túrákat, és k-opt algoritmusokat is. De mivel k növekedtével a k él eltörlése után a különböző lehetőségek száma, amikor túrát kapunk, nagyon gyorsan nő, ezen algoritmusokat ritkán alkalmazzuk. Például csak már 2-optimális túrát vizsgálunk, 3-optimális-e. 3-opt TSPLIB: 1.04.

$$\begin{aligned} k &= 2 \to 1 \\ k &= 3 \to 4 \\ k &= 4 \to 20 \\ k &= 5 \to 148 \\ k &= 6 \to 1358 \\ k &= 10 \to 51290496 \end{aligned}$$

6.4. Christofides heurisztikája

Christofides heurisztikája 1976-ból származik, és mindmáig a legjobb elméleti korlátot adó algoritmus. Az már más kérdés, hogy gyakorlatban nem a várakozásokhoz mérten működik. TSPLIB mérőszáma: 1.14, és továbbfejlesztett változatáé is mindössze 1.09, amely eredmények könnyen túlszárnyalhatóak voltak az elemibb heurisztikákkal. Megjegyezzük, hogy ezek az eredmények is természetesen jobbak, mint az elméleti 1.5-ös felső korlát.

Az algoritmus a háromszög-egyenlőtlenséget kielégítő nemnegatív súlyok esetén működik. Első lépésként keressünk a gráfban egy minimális súlyú F feszítő fát, például Kruskal algoritmusával.

Legyen W azon csúcsok halmaza, amik F-ben páratlan fokúak. Keressük meg $G[W]$ (a W által feszített súlyozott részgráf) minimális súlyú teljes párosítását, jelöljük ezt M-mel.

Most nézzük az $F \cup M$ gráfot, ami a mindkettőben szereplő éleket két példányban tartalmazza. Ennek a gráfnak minden foka páros, tehát van Euler-séta. Ha minden csúcs foka 2, akkor egy túrát kaptunk, tehát leállunk. ha van legalább 4 fokú csúcs, akkor az Euler sétának erre illeszkedő egymás utáni két élét egy éllel helyettesítjük. Az így kapott gráfban továbbra is van Euler-séta. Ilyen műveleteket csinálunk egészen addig, amíg túrát nem kapunk.

6.6. tétel (Christofides, 1976)**.** *Az algoritmus által az utazóügynök feladatra adott megoldás hossza az optimális megoldás legfeljebb* $\frac{3}{2}$*-szerese.*

Bizonyítás. Legyen T_{opt} egy optimális túra. Tudjuk, hogy $c(F) \leq c(T_{opt})$. Tekintsük a $G[W]$ gráfnak azt a T_W túráját, amit a W pontjainak T_{opt}-beli sorrendje ad. A háromszög-egyenlőtlenség teljesülése miatt $c(T_W) \leq c(T_{opt})$. Mivel T_W előáll mint W két teljes párosításának uniója, M definíciójának értelmében $2c(M) \leq c(T_{opt})$, azaz $c(T \cup M) \leq \frac{3}{2}c(T_{opt})$. Megint a háromszög-egyenlőtlenség miatt a végső túra súlya is legfeljebb $c(T \cup M)$, így bebizonyítottuk a tételt. □

6.5. Lin és Kernighan heurisztikája

Ebben a fejezetben az általában legjobban működő utazóügynök bejárást adó heurisztikák alapsémájával ismerkedünk meg. Az elgondolás Lintől és Kernighantől származik 1973-ból.

6.7. definíció. A $v_1, e_1, v_2, e_2, \ldots, v_n, e_n, v_{n+1}$ n pontú és n élű gráfot δ-útnak nevezzük, ha $v_{n+1} = v_i$ valamely $1 \leq i \leq n$ értékre, és minden más pont különböző.

Ha P egy δ-út, akkor jelölje $T(P)$ a P-ből az $e_n = v_n v_{n+1}$ él eltörlése és az $e'_n = v_n v_1$ él hozzávételével kapott túrát. A Lin-Kernighan-algoritmus, amint látni fogjuk, δ-utak sorozata segítségével keres egyre rövidebb bejárásokat.

Lin-Kernighan-alapalgoritmus

Adott egy T túra G-ben.

1. lépés: (Élkeresés) A G minden v pontjára és rá illeszkedő T-beli $e = uv$ élre hajtsuk végre a *2–5. lépést.*

2. lépés: (Élkeresés inicializálása) $u_0 := u$. P^0 legyen az a δ-út, melyet T-ből az $u_0 v$ él törlésével, és egy olyan $u_0 w_0$ hozzáadásával kapunk, melyre $c(u_0 w_0) \leq c(u_0 v)$. Ha nincs ilyen w_0 pont, akkor az Élkeresésnek vége. $i := 0$.

3. lépés: (Túrateszt) Ha $c(T(P^i))$ kisebb, mint az eddig talált legjobb túra hossza, akkor $T(P^i)$-t megjegyezzük.

4. lépés: (Következő δ-út) Legyen u_{i+1} a w_i másik szomszédja a deltaút körén. Ha $w_i u_{i+1}$ T-nek nem éle, akkor ugorjunk az *5. lépésre.* Különben keressünk olyan w_{i+1} pontot, amelyre $u_{i+1}w_{i+1}$ nem éle T-nek, és $c(P^i - w_i u_{i+1} + u_{i+1}w_{i+1}) \leq c(T)$. Ha nincs ilyen pont, akkor ugorjunk az *5. lépésre.* Különben legyen $P^{i+1} := P^i - w_i u_{i+1} + u_{i+1}w_{i+1}$, $i := i+1$, és ugorjunk a *3. lépésre.*

5. lépés: Ha találtunk jobb túrát, akkor legyen az az új T, és ugorjunk az *1. lépésre.* Különben folytassuk az Élkeresést.

Fontos megjegyezni, hogy egy Élkeresés során a T éleit csak törölhetjük a δ-utakból, a T-ben nem szereplő éleket pedig csak hozzávehetjük. A fenti algoritmusnak az is egy lényeges eleme, hogy amikor egy rövidebb túrára lelünk, nem iteráljuk rögtön az algoritmust, hanem előbb befejezzük az élkeresést, remélve, hogy még jobb túrákat találunk a már felépített struktúra segítségével. (Bár ennek jelentőségét sokan kétlik).

Ezen alapalgoritmusnak is van két finomítása, amelyek közül az egyik az inicializálás körültekintőbb végrehajtását végzi el, a másik pedig a δ-utak választásakor jár el figyelmesebben, de ezen részleteket itt nem tárgyaljuk.

Láncolt Lin-Kernighan-algoritmus

Természetesen, ha időnk engedi, célszerű a Lin-Kernighan-alapalgoritmust több kiindulási túrából is lefuttatni, és a legjobb kapott túrát kiválasztani. A tapasztalat szerint a különféle kiindulási túrák egy jó választása az, amikor a Lin-Kernighan-alapalgoritmus által adott legjobb túrán végrehajtunk egy véletlenszerű 4-cserét, és azzal iteráljuk az algoritmust. Magyarázatként az szolgálhat, hogy a 4-cserék annyira megváltoztatják egy adott túra szerkezetét, olyannyira eltávolodunk az addig tekintett túráktól, hogy az esetlegesen nagyon rossz (hosszú) kiindulási túra ellenére is reménykedhetünk majd jobb lokális optimumban.

A Lin-Kernighan-alapalgoritmus TSPLIB mérőszáma 1.02, a láncolt változaté 1.01. Persze a láncolt változat nem véges, leállási kritériumként megfogalmazhatjuk, hogy ha már egy idő elteltével nem tapasztalunk javulást, akkor véget vetünk az algoritmus működésének.

6.6. Feladatok

13. feladat. *Mutassuk meg, hogy Christofides heurisztikájának működésére adott $\frac{3}{2}$-es felső korlát nem javítható.*

14. feladat. *Mutassuk meg, hogy amennyiben a Lin-Kernighan-algoritmus során minden lehetséges élet hozzávehetünk a δ-utakhoz, akkor az algoritmus lefutása után kapott T túra 3-optimális, azaz T-n a 3-opt algoritmus nem változtat.*

7. fejezet

Az utazóügynök feladat alsó korlátjai

Amennyiben nem tudjuk előre feladatunk optimális megoldását, nem tudjuk azt sem, hogy a kapott heurisztikus megoldásunk mennyire jó, azaz mennyire van közel az optimális megoldáshoz. Mivel a TSPLIB feladataira már ismertek javarészt az optimális megoldások, arra nagyon alkalmasak, hogy heurisztikus algoritmusunk várható viselkedését előre megjósoljuk az átlagos működés tekintetében, de egy új feladat megoldásánál újabb teendőink vannak. Ezen teendők az optimális megoldására vonatkozó alsó korlátok keresését jelentik, ezen alsó korlátokkal tudjuk megmondani, hogy a heurisztikus megoldásaink milyen jók, és persze minél jobb alsó korlátot tudunk adni, annál pontosabban értékelhetjük a már meglévő megoldásainkat.

Ebben a fejezetben megismerkedünk néhány, az utazóügynök feladatra vonatkozó alsó korlát számítási módszerrel. Ezen módszerek vezettek el később az olyan általános módszerekhez, mint a Lagrange-feladat, a korlátozás és vágás módszere (branch and cut).

7.1. Held és Karp korlátja

Azt már korábban is használtuk, hogy az utazóügynök feladat megoldásának egy alsó korlátját adja a minimális feszítőfa hossza. Most ennél egy lehelletnyit kifinomultabb alsó korlátot adunk, amiből azonban egy igen használható módszer születik.

Adott tehát a $G = (V, E)$ teljes gráf a $c : E \to \mathbb{R}_+$ élsúlyozással. Legyen $v_1 \in V$ egy rögzített pont. Legyen A a v_1-re illeszkedő két legolcsóbb él költségének az összege, B pedig a v_1 elhagyása utáni gráf minimális feszítőfájának a súlya. Ekkor $A+B$ az utazóügynök feladat egy alsó korlátja.

> *1-fa korlát*
>
> A $G = (V, E)$ teljes gráfban a $c : E \to \mathbb{R}_+$ élhosszak esetén a $v_1 \in V$ ponthoz tartozó 1-fa korlát az $A + B$ összeg, ahol $A := \min\{c_e + c_f : e, f \in \delta(v_1), e \neq f\}$, B pedig a $G - v_1$ gráf minimális költségű feszítőfájának a költsége.

Ezzel a korláttal egyáltalán nem lehetünk elégedettek, hiszen elképzelhető, hogy az 1-fa korlátot adó élhalmaz alakja nagyon távol esik az utazóügynök túráktól, például nagyon nagy fokú pontok lehetnek a feszítőfában, és ennek következtében nagyon sok elsőfokú pont is létezhet. Ezen viszont valamelyest tudunk segíteni a következő módon. Válasszunk ki egy, a fában nagyfokú v pontot, és minden, a G-ben rá illeszkedő él súlyát növeljük meg M-mel. Mi történik az új súlyozásban az 1-fa korláttal? Jól eltalált M érték esetén v kisebb fokú lesz a fában, azaz az 1-fa korlát értéke kevesebbel nő, mint $d_F(v) \cdot M$, de azért $2M$-nél többel nőhet. Ámde minden túra hossza pontosan $2M$-mel nőtt, azaz jobb alsó korláthoz juthatunk.

Ezt az elgondolást egyszerre fogjuk alkalmazni az összes pontra. Legyen $y : V \setminus \{v_1\} \to \mathbb{R}$ a pontokon egy súlyfüggvény. Legyen C a következő új élsúlyozással kapott 1-fa korlát: minden

$e = uv \in E$ élre $\bar{c}_e := c_e - y_u - y_v$, ahol $y_{v_1} = 0$, rögzített.

7.1. állítás (Held és Karp korlátja)**.** *$C + 2\sum_{v \in V \setminus \{v_1\}} y_v$ az eredeti utazóügynök feladat egy alsó korlátja.*

Az állítás bizonyítása abból adódik, hogy az új súlyozás mellett minden túra hossza $2\sum_{v \in V \setminus \{v_1\}} y_v$ értékkel csökken.

Held–Karp-korlát

A $G = (V, E)$ teljes gráfban a a $c : E \to \mathbb{R}_+$ élhosszak esetén adott a $v_1 \in V$ pont, és az $y : V \setminus \{v_1\} \to \mathbb{R}$ súlyfüggvény. A $C + 2\sum_{v \in V \setminus \{v_1\}} y_v$ összeget az y súlyozáshoz tartozó Held–Karp-korlátnak nevezzük, ahol C a $\bar{c}_e := c_e - y_u - y_v$ élsúlyozású gráfban az 1-fa korlát.

A korlát nagy előnye abban rejlik, hogy újabb és újabb pontsúlyozásokat tekintve jelentősen javítható. Held és Karp 1971-ben az alábbi módszert javasolta: jelölje $d_F(v)$ a v pont fokát az F 1-fa korlátot adó élhalmazban, és minden v ponton változtassuk meg az y_v pontsúlyozást a következőképpen:

$$y_v^{\text{új}} := y_v + t(2 - d_F(v)),$$

ahol t egy pozitív szám, az úgynevezett lépéshossz.

Ez azt jelenti, hogy az elsőfokú pontokra illeszkedő élek súlya csökkenni fog (lehetőséget adva ezzel az új súlyozás mellett az 1-fa korlátbeli foka növekedésének), a harmad- vagy magasabbfokú pontokra illeszkedő élek súlya pedig nőni fog (a fokszám csökkenését előmozdítandó). Persze mindezeket az összes pontra egyszerre tekintjük.

A tapasztalat azt mutatja, hogy a lépéshossz ügyes megválasztása esetén a kapott Held–Karp-korlátok az optimális (azaz a lehető legjobb) Held–Karp-korláthoz konvergálnak, bár az nem igaz, hogy szigorúan monoton nőne a sorozatuk. (Persze az nem világos, hogy miért is létezik legjobb Held–Karp-korlát, ezt csak a következő fejezetben bizonyítjuk.)

7.2. Lineáris programozási alsó korlát

Az alapgráf minden élének feleltessünk meg egy változót, és tekintsük az alábbi egyenlőtlenségrendszert.

$$x(\delta(v)) = 2 \quad \forall v \in V \tag{7.1}$$

$$x_e \in \{0, 1\} \quad \forall e \in E. \tag{7.2}$$

Ennek megoldásai a gráfot fedő diszjunkt körök karakterisztikus vektorai. A következő feltételek hozzávétele után azonban már csak a Hamilton-körök lesznek a megoldások.

$$x(\delta(S)) \geq 2 \quad \forall S \subset V, S \neq \emptyset. \tag{7.3}$$

Ezek fényében az alábbi feladatot *az utazóügynök feladat lineáris programozási relaxáltjának* nevezzük.

$$\min \sum (c_e x_e : e \in E) \tag{7.4}$$

$$x(\delta(v)) = 2 \quad \forall v \in V \tag{7.5}$$

$$x(\delta(S)) \geq 2 \quad \forall S \subset V, S \neq \emptyset \tag{7.6}$$

$$0 \leq x_e \leq 1 \quad \forall e \in E. \tag{7.7}$$

Az nem igaz, hogy tetszőleges célfüggvényre ezen lineáris programnak létezne egész megoldása. A feladat optimumának meghatározása tehát csak egy alsó korlátot ad az optimális utazóügynök bejárás hosszára. Ezen alsó korlátot meghatározó módszert aztán beépíthetjük korlátozás és szétválasztás vagy korlátozás és vágás módszerbe, ahogy azt látni is fogjuk.

Most belátjuk, hogy a (7.6) feltételeket, amelyeket *résztúra feltételeknek* hívunk, az alábbiakkal helyettesítve ekvivalens feladathoz jutunk.

$$x(\gamma(S)) \leq |S| - 1 \quad \forall S \subset V, S \neq \emptyset. \tag{7.8}$$

Ugyanis igaz a következő:

$$2 + 2x(\gamma(S)) \leq x(\delta(S)) + 2x(\gamma(S)) = \sum_{v \in S} x(\delta(v)) = 2|S|,$$

ahol az első egyenlőtlenség a (7.6)-ból, míg at utolsó egyenlet a (7.5) egyenletből következik.

A következő tételt fogjuk belátni, amelynek egyik jelentősége, hogy a Held és Karp féle algoritmus a nagyon sok feltétellel rendelkező lineáris programozási relaxált megoldására egy heurisztikus eljárást szolgáltat.

7.2. tétel. *A lineáris porgramozási alsó korlát megegyezik az optimális Held-Karp-korláttal.*

Két bizonyítást is adunk.

Bizonyítás I. Jelöljünk ki egy rögzített v_1 pontot. Vegyük észre, hogy az alábbi feladat a (7.4) LP-relaxálttal ekvivalens.

$$\min \sum (c_e x_e : e \in E) \tag{7.9}$$

$$x(\delta(v)) = 2 \quad \forall v \in V \tag{7.10}$$

$$x(\gamma(V - v_1)) = |V| - 2 \tag{7.11}$$

$$x(\gamma(S)) \leq |S| - 1 \quad \forall S \subset V - v_1, S \neq \emptyset \tag{7.12}$$

$$0 \leq x_e \leq 1 \quad \forall e \in E. \tag{7.13}$$

Ugyanis, a (7.11) feltétel ekvivalens az $x(\delta(v_1)) = 2$ feltétellel, hiszen $x(\delta(v_1)) = x(\delta(V \setminus \{v_1\}))$, és emiatt

$$x(\delta(v_1)) + 2x(\gamma(V - v_1)) = \sum_{v \in V \setminus v_1} x(\delta(v)) = 2|V - v_1|.$$

Továbbá, elegendő csak azokat az S halmazokat tekinteni a (7.6) korlátokban, amelyek nem tartalmazzák a v_1 csúcsot, mivel a feltételek tetszőleges S, és $V \setminus S$ halmazra ekvivalensek.

Ha elhagyjuk a v_1-re illeszkedő élekhez tartozó változókat és a (7.10) feltételeket, akkor a maradék rendszernek a 11.10 tétel (11.2 fejezet) alapján a $V - v_1$ alaphalmazon vett minimális súlyú feszítőfa lesz az optimális megoldása.

Az alábbinak pedig a minimális 1-fa!

$$\min\{cx : x \text{ teljesíti a következőket: } (7.11),\ (7.12),\ (7.13), x(\delta(v_1)) = 2\} \tag{7.14}$$

A pontokra írt súlyokhoz a (7.9) feladat duálisának az elemzésével jutunk. Könnyen látható, hogy a (7.13) feltételben az $x_e \leq 1$ következik a megelőző feltételekből, így azt a duális elkészítésében nem is szerepeltetjük. A duálisban tehát az 1-fa feladat feltételein kívül szerepel minden $v \in V - v_1$ ponthoz egy y_v vátozó. A duális egy optimális megoldásában jelöljük ezen változók értékét y_v^*-gal. Amennyiben a szóbanforgó változók értékét a megadott értékekre rögzítjük, akkor a többi változó az alábbi lineáris program optimális megoldását adja.

$$\min \sum_{uv\in E} (c_{uv} - y_u^* - y_v^*)x_{uv}$$

$$x \text{ teljesíti a következőket: (7.12), (7.11), (7.13)}$$

$$x(\delta(v_1)) = 2.$$

A korábbiak szerint ez a célfüggvényben szereplő élsúlyokkal egy minimális súlyú 1-fa feladat, azaz a (7.9) feladat optimális megoldása megegyezik a pontokra írt y_v^* súlyokkal vett 1-fa feladat megoldásával. (Ahol a v_1 pont súlya 0.)

A másik irány bizonyításához vegyük észre, hogy tetszőleges pontsúlyozással a (7.9) feladat duálisának egy megengedett megoldását kapjuk, amelynek a célfüggvényértéke megegyezik a súlyozáshoz tartozó Held-Karp-korláttal. □

Bizonyítás II. Hasonlóan az első bizonyításhoz előállítjuk a (7.9)-(7.13) lineris programot, majd képezzük a Lagrange duál feladatot az $x(\delta(v)) = 2$, $v \in V \setminus \{v_1\}$ feltételek dualizálásával. Ekkor éppen a Held-Karp korlátot kapjuk. □

7.3. Oszlopgenerálós megoldás

Held és Karp heurisztikája, amint azt az előző fejezetben láttuk, egyben a lineáris programozási alsó korlátra is számol egy heurisztikus becslést. Milyen problémákkal szembesülünk, ha közvetlenül szeretnénk megoldani a lineáris programozási relaxáltat, azaz az alábbi feladatot?

$$\min \sum (c_e x_e : e \in E) \tag{7.15}$$

$$x(\delta(v)) = 2 \quad \forall v \in V \tag{7.16}$$

$$x(\delta(S)) \geq 2 \quad \forall S \subset V, S \neq \emptyset \tag{7.17}$$

$$0 \leq x_e \leq 1 \quad \forall e \in E. \tag{7.18}$$

Az első jelentős akadályt az jelenti, hogy rengeteg feltételt tartalmaz a feladat: a (7.17) feltételek száma exponenciális az alapgráf méretéhez képest – nevezetesen majdnem $2^{|V|}$ – még akkor is, ha észrevesszük, hogy fölösleges minden részhalmazra és a komplementerére is felírni ugyanazt a lineáris egyenlőtlenséget, élhetünk az $|S| \leq |V|/2$ megszorítással – de még így is majdnem $2^{|V|-1}$ darab feltételünk marad.

A továbbiakban ismertetjük a Dantzig, Fulkerson, Johnson szerzőhármas fenti nehézség legyőzésére irányuló megközelítését, amely 1954-ből származik.

Oldjuk meg először a kevés feltételt tartalmazó (7.15), (7.16), (7.18) feladatot. Ha – óriási véletlen folytán – a kapott megoldás egy Hamilton-kör karakterisztikus vektora, akkor meg is oldottuk a TSP feladatunkat. Ha nem, akkor keressünk olyan (7.17)-beli feltételeket, amelyeket megsért a kapott optimális megoldás, és vegyük a feladathoz ezeket is. Ha nem találunk ilyen feltételeket, akkor megoldottuk a relaxált feladatot. Következő lépésként oldjuk meg a kapott kibővített lineáris programot, és ismételgessük az eljárást, amíg meg nem állunk (esetleg nem győzünk már várakozni, esetleg a számítógépünk memóriája lett tele, esetleg a hócipőnk).

Ezen elgondolás megvalósításához két újabb nehézséget kell legyőzni: hogyan ellenőrizzük, hogy egy megoldás teljesíti-e a (7.17) résztúra-feltételeket, és hogyan oldjuk meg a folytonosan előálló lineáris programokat.

Az első nehézséget egy egyszerű folyamproblémával tudjuk kezelni. Nevezetesen az x^* élkapacitásokkal megkeressük a $G = (V, E)$ gráf minimális súlyú vágását, és amennyiben ez

2-nél nagyobb, akkor minden feltételt teljesít x^*, ha pedig kisebb, akkor olyan vágásokat is kapunk, amelyekre nem teljesül (7.17) feltétele.

A minimális értékű vágás megtalálására a folyamos módszernél jobb megoldást nyújt Nagamochi és Ibaraki 1992-ből származó algoritmusa, amit a 11.4. Fejezetben ismertetünk.

Kis gráfok esetén a második probléma nem merül fel, viszont nagy problémák esetén olyan nagy számú változóval szembesülünk, amit már nem tudunk kezelni. A Négy Bill Könyvének sokat emlegetett chip-bejárási példáján például 687 368 darab változót tartalmazó feladathoz jutunk. Jobban járunk, ha ezt a feladatot nem akarjuk közvetlenül semelyik forgalomban lévő szimplex-módszerre épülő lineáris programozási megoldóprogrammal megoldani.

Amit tehetünk, a következő. A résztúra-vágásokhoz hasonlóan ne kezdjünk rögtön az összes változóval dolgozni, számolni, hanem csak egy részükkel, és majd fokozatosan azokat a változókat vegyük a feladathoz, amelyekre szükség lesz.

Tegyük fel, hogy választottunk egy $E' \subseteq E$ élhalmazt, amelyre az alábbi $G' = (V, E')$-re vonatkozó LP-feladatnak van megoldása.

$$\min \sum (c_e x_e : e \in E') \tag{7.19}$$
$$x(\delta_{E'}(v)) = 2 \quad \forall v \in V$$
$$x(\delta_{E'}(S)) \geq 2 \quad \forall S \subset V, S \neq \emptyset$$
$$0 \leq x_e \leq 1 \quad \forall e \in E'.$$

(E'-re jó választás például néhány (10 darab?) túra, amelyet a Láncolt Lin-Kernighan algoritmus ad.)

(7.19) egy optimális x' megoldása kiterjeszthető a (7.15) feladat egy megengedett x^* megoldásává az $x^*_e := 0$, $e \in E - E'$ definiálásával. Igaz-e, hogy x^* a (7.15) optimális megoldása? Általában persze nem.

x^* optimalitásának ellenőrzéséhez tekintsük a (7.19) feladat lineáris programozási duálisát:

$$\max \sum (2y_v : v \in V) + \sum (2Y_S : S \subset V, S \neq \emptyset) \tag{7.20}$$

$$y_u + y_v + \sum (Y_S : uv \in \delta(S), S \subset V, S \neq \emptyset) \leq c_{uv} \quad \forall\ uv \in E', \tag{7.21}$$

$$Y_S \geq 0 \quad \forall\ S \subset V, S \neq \emptyset. \tag{7.22}$$

Legyen y', Y' ennek egy optimális megoldása. Ha ez megengedett megoldása a (7.15) duális megoldásának is, azaz annak a feladatnak, amelyet a (7.21)-ben E' helyett E-vel kapunk, akkor bizony x^* az eredeti (7.15) feladatnak, azaz a lineáris programozási relaxáltnak is optimális megoldása. Ha nem ez a helyzet, akkor vegyük hozzá E'-höz azokat az $e \in E - E'$ éleket, amelyekre (7.21) nem teljesül. Oldjuk meg a kibővített élhalmazzal adódó feladatot, és ismételgessük az eljárást.

Az ilyen algoritmusokat oszlopgenerálós algoritmusoknak nevezzük. Ezt az oszlopgenerálós algoritmust a résztúra-feltételek fokozatos hozzáadásával kombinálva esetleg meg tudjuk oldani a lineáris programozási relaxáltat, esetleg nem.

7.4. Vágósíkok

Legyen Q a túrává kiegészíthető élhalmazok konvex burka. Mivel a teljes gráfot tekintjük, ez egy n dimenziós poliéder. Az utazó ügynök feladat megoldásában sokat segít, ha ennek a poliédernek minél több lapját ismerjük. Mindenesetre néhány lapot könnyen kaphatunk az eddig tekintett egyenlőtlenségekből:

- $x_e \geq 0$ lap, mert a $\{0, \chi_f \ (f \neq e)\}$ vektorhalmaz affin független, és egyenlőséggel teljesítik.

- $x_e \leq 1$ lap, mert a $\{\chi_e, \chi_{\{e,f\}} \ (f \neq e)\}$ vektorhalmaz affin független, és egyenlőséggel teljesítik.

- $x(\delta(v)) \leq 2$ lap, de ezt már kicsit nehezebb belátni. Vegyük az éleknek egy olyan sorrendjét, hogy $\delta(v) = \{e_1, \ldots, e_{n-1}\}$, és $\{e_1, e_2, e_m\}$ kör. Ekkor a $\chi_{\{e_1,e_j\}}$ $(2 \leq j \leq n-1)$, $\chi_{\{e_2,e_3\}}$, $\chi_{\{e_1,e_2,e_j\}}$ $(n \leq j \leq m-1)$, és $\chi_{\{e_1,e_3,e_m\}}$ vektorok affin függetlenek, egyenlőséggel teljesítik az egyenlőtlenséget, és mind kiegészíthetők túrává.

Bizonyítás nélkül megemlítjük, hogy $2 \leq |U| \leq n-2$ esetén $x(E[U]) \leq |U| - 1$ is lapja Q-nak. Ha $n \leq 5$, akkor a fentieken kívül nincs is más lap, de $n \geq 6$ esetén van. Az alábbiakban megmutatjuk, hogyan lehet további, bonyolultabb lap-osztályokat kapni.

7.4.1. Fésű-egyenlőtlenségek

A $G = (V, E)$ gráf pontjainak néhány részhalmazát *fésűnek* nevezzük, ha teljesítik a következőket. A $H \subset V$ kitüntetett halmazt a fésű nyelének nevezzük, a $T_1, T_2, \ldots T_{2k+1}$ páronként diszjunkt, nemüres halmazokat pedig a fésű fogainak, ha $k \geq 1$ valamely egész szám. Továbbá megköveteljük, hogy $H \cap T_i \neq \emptyset, T_i - H \neq \emptyset$ teljesüljön.

A következő tétel Chvátaltól (1973) és Grötscheltől és Padbergtől származik (1979).

7.3. tétel. *Legyen C egy fésű a H nyéllel és a T_i fogakkal, $i = 1, 2, \ldots 2k+1$. Ekkor tetszőleges Hamilton-kör karakterisztikus vektora teljesíti a következő egyenlőtlenséget.*

$$x(\gamma(H)) + \sum_{i=1}^{2k+1} (x(\gamma(T_i)) \leq |H| + \sum_{i=1}^{2k+1} (|T_i| - 1) - (k+1) \tag{7.23}$$

Bizonyítás. Tegyük fel, hogy $x \in \mathbb{R}^E$ egy tetszőleges Hamilton-kör karakterisztikus vektora. Ekkor x teljesíti (7.5),(7.6),(7.7)-et, továbbá

$$2x(\gamma(H)) + 2\sum_{i=1}^{2k+1} (x(\gamma(T_i)) =$$

$$= \sum_{v \in H} x(\delta(v)) - x(\delta(H)) + \sum_{i=1}^{2k+1} (x(\gamma(T_i \cap H)) +$$

$$+ \sum_{i=1}^{2k+1} (x(\gamma(T_i - H)) + \sum_{i=1}^{2k+1} (x(\delta(T_i, H)) + \sum_{i=1}^{2k+1} (x(\gamma(T_i)) \leq$$

$$\leq \sum_{v \in H} x(\delta(v)) - x(\delta(H)) + \sum_{i=1}^{2k+1} (x(\gamma(T_i \cap H)) + \sum_{i=1}^{2k+1} (x(\gamma(T_i - H)) + x(\delta(H)) + \sum_{i=1}^{2k+1} (x(\gamma(T_i)) \leq$$

$$\leq \sum_{v \in H} x(\delta(v)) + \sum_{i=1}^{2k+1} (|T_i \cap H| - 1) + \sum_{i=1}^{2k+1} (|T_i - H| - 1) + \sum_{i=1}^{2k+1} (|T_i| - 1) =$$

$$= 2|H| + 2\sum_{i=1}^{2k+1} (|T_i| - 1) - (2k+1),$$

ahol a második egyenlőtlenségnél (7.8)-t használtuk, a rákövetkező egyenlőségnél pedig (7.5)-t. Mivel a levezetés elején egy páros szám szerepel, ezért a levezetés végén szereplő nála nem kisebb páratlan számnál eggyel kisebb páros szám sem kisebb nála. Kettővel való osztás után adódik (7.23). □

Az előző tétel az jelenti, hogy a (7.4) poliéderre vonatkozóan a (7.23) egyenlőtlenség egy érvényes vágás. (Nevezetesen egy elsőfajú Gomory-Chvátal-vágás.) Ezeket az egyenlőtlenségeket *fésű-egyenlőtlenségnek* nevezik.

Pillanatnyilag nem ismert polinomiális algoritmus arra vonatkozóan, hogy egy adott x teljesíti-e az összes fésű-egyenlőtlenséget.

7.4.2. Virág-egyenlőtlenségek

Ebben a fejezetben megismerkedünk a fésű-egyenlőtlenségek egy részcsaládjával, amelyek esetében el tudjuk dönteni ezt a kérdést.

Ha a fésűnek minden foga kételemű halmaz, akkor a (7.23) egyenlőtlenséget *virág-egyenlőtlenségnek* nevezzük. Legyen $H \subseteq V$ egy csúcshalmaz, $M \subseteq \delta(H)$ páratlans sok H-ból kilépő független él. Ekkor a H-hoz és M-hez tartozó virág-egyenlőtlenség a következő:

$$x(\gamma(H)) + x(M) \leq |H| + \frac{|M|-1}{2}. \tag{7.24}$$

Most leírjuk Padberg és Rao (1982) módszerét, amely az utazóügynök feladat esetén eldönti egy x vektorról, hogy teljesíti-e az összes virág-egyenlőtlenséget, és ha nem, akkor meg is ad egyet. Adott tehát egy $x \in \mathbb{R}^E_+$, amely teljesíti a (7.16) egyenlőtlenségeket. Ekkor (7.24) a következő alakba is írható:

$$x(\delta(H) - M) - x(M) \geq 1 - |M|. \tag{7.25}$$

Ugyanis (7.16) szerint a bal oldal a következőképp alakítható:

$$x(\gamma(H)) + x(M) =$$
$$= \frac{1}{2}\left(\sum_{v \in H} x(\delta(v)) - x(\delta(H))\right) + x(M) =$$
$$= |H| - \frac{1}{2}x(\delta(H) - M) + \frac{1}{2}x(M).$$

(7.25)-t az alábbi alakba írva azt kell meghatározni, hogy van-e olyan H halmaz, amelyből kilépő éleken a súlyok összege – ahol egy párosításon, azaz az M élhalmazon x_e helyett $1-x_e$ súllyal számolva – kisebb mint 1.

$$x(\delta(H) - M) + (|M| - x(M)) \geq 1. \tag{7.26}$$

Ezt a következő segédgráfban a minimális T-vágás meghatározásával (l. 11.6 fejezet) el tudjuk dönteni.

Legyen G' a következő gráf. G-ből töröljük azokat az éleket, amelyekre $x_e = 0$. A maradék élek mindegyikét osszuk fel két új ponttal: az $e = uv$ élet v'_e-vel és v''_e-vel. Az uv'_e és a vv''_e élek súlya *maradjon* x_e, a $v'_e v''_e$ él súlya pedig legyen $1 - x_e$. A T álljon az új pontok halmazából (T elemszáma nyilván páros.)

Tegyük fel, hogy az $x \in \mathbb{R}^E$ megsérti a H, M párhoz tarozó virág-egyenlőtlenséget. Álljon S a G' H-nak megfelelő pontjaiból, továbbá azon *új* pontokból, amelyek a G H-ban feszített éleire kerültek, továbbá a H-ból kilépő M-beli élekre került új pontok közül a H-beli végponthoz közelebbiekből. Ekkor S nyilván T-vágás (azaz $S \cap T$ páratlan elemszámú), és az S-ből kilépő G'-beli élek összsúlya

$$x(\delta(H) - M) + (|M| - x(M)), \tag{7.27}$$

ami tehát kisebb, mint 1, azaz a minmimális T-vágás értéke is kisebb 1-nél.

Megfordítva, ha a G'-beli minimális T-vágás értéke kisebb mint 1, akkor létezik H, M pár G-ben, amelyre vonatkozóan x megsérti a virág-egyenlőtlenséget. Tekintsünk ugyanis egy S minimális T-vágást, amelynek az értéke tehát kisebb, mint 1. Mivel egy G-beli élből csinált 3 hosszú út két szomszédos élének összsúlya 1, nem léphetnek ki egyszerre S-ből; másrészt a 3 hosszú út két szélső éle sem léphet ki, mert akkor a két középső csúcs átrakásával egy kisebb T-vágást kapnánk. Tehát minden ilyen 3 hosszú útnak legfeljebb 1 éle lép ki S-ből. Álljon M azokból a G-beli élekből, amikhez tartozó 3 hosszú út középső éle lép ki S-ből.

Ekkor M komponensei legfeljebb kétélűek (azt akarjuk elérni, hogy egyélűek legyenek, hiszen ekkor kapunk virágegyenlőtlenséget). Egy kétélű komponens *középső* pontjainak az S-be való be-, illetve kirakásával olyan minimális T-vágást kapunk, amelyhez tartozó M élhalmaz elemszáma csökken. Eszerint legfeljebb $|V|$ lépésben olyan minimális T-vágáshoz jutunk, amelyhez tartozó H, M pár olyan virág-egyenlőtlenséget ad, amelyet x megsért. (Ekkor M nem lehet egyelemű, l. 15 feladatot.)

Ennél fogva egy adott x-ről a G'-ben a minimális T-vágás megkeresésével el tudjuk dönteni, hogy teljesíti-e a virág-egyenlőtlenségeket, ha pedig nem, akkor a minimális T-vágás segítségével meg is tudunk adni egy S, M párt, amelyre nem teljesül, és ezen vágásnak az egyenletrendszerhez való hozzávételével az eljárást folytatva az utazóügynök feladat egyre jobb alsó korlátjához jutunk.

7.5. Feladatok

15. feladat. *A lineáris programozási relaxált feladat tetszőleges megoldására $k = 0$ esetén a fésű-egyenlőtlenségek automatikusan teljesülnek. Ugyanez igaz $|A| = 1$ esetében a virág-egyenlőtlenségekre.*

7.6. Korlátozás és szétválasztás módszere

Az előző fejezetek vágásaival nagyon jó alsó korlátokhoz juthatunk, amelyek segítségével aztán meggyőzhetjük magunkat vagy a megrendelőnket, hogy a Láncolt Lin–Kernighan Algoritmus segítségével megkeresett túrák valóban nagyon megközelítik az optimumot. De mit tegyünk, ha még mindig nem vagyunk elégedettek, és az optimumhoz még közelebbi, netán optimális megoldást keresünk? Akkor próbálkozhatunk a széles körben alkalmazott korlátozás és szétválasztás módszerével javítani az alsó korláton, illetve esetleg még jobb túrák megkeresésében is.

Tegyük fel, hogy adott a szokásos módon a $G = (V, E)$ teljes gráf a c élsúlyozással. Jelölje $\mathcal{T}$ a G összes túrájának a halmazát. Az utazóügynök feladat egy alsó korlátja a K szám, ha minden $T \in \mathcal{T}$ túrára $c(T) \geq K$ teljesül. A lineáris programozási alsó korlát, és aztán a vágósíkok keresésével rendelkezünk egy eljárással, amely alsó korlátot számol. Amennyiben a $\mathcal{T}$ egy $\mathcal{T}_0 \cup \mathcal{T}_1$ partíciójára tudunk mondani egy-egy alsó korlátot, azaz meg tudunk határozni K_0 és K_1 alsó korlátokat, melyekre $c(T) \geq K_0$ minden $T \in \mathcal{T}_0$ esetén, és $c(T) \geq K_1$ minden $T \in \mathcal{T}_1$ esetén, akkor a $\min\{K_0, K_1\}$ érték alsó korlátja az összes $\mathcal{T}$-beli túra hosszának. Esetleg jó eséllyel így nagyobb korlátot kapunk, mint a K volt.

Ha a továbbiakban tovább bontjuk a halmazainkat, és alkalmazzuk rájuk alsó korlát számolási eljárásunkat, akkor az egyes részek alsó korlátjai közül a minimálisat választva remélhetjük, hogy egyre nagyobb alsó korláthoz jutunk, ugyanis elképzelhető, hogy alsó korlát számolási eljárásunk egyre ügyesebben dolgozik a megszorított halmazokon.

Ha továbbra is a lineáris programozási alsó korlátot tekintjük (majd különféle vágásokkal továbbjavított változatát), akkor a fenti eljárásnak a következőképp képzelhető el egy megvalósítása.

Kiindulásunk a P probléma, amely tehát a összes túra $\mathcal{T}$ halmazában kérdezi a legrövidebbet. Kézenfekvő úgy kettévágnunk $\mathcal{T}$-t, hogy egy e él kiválasztása után $\mathcal{T}_0$ az e élet nem tartalmazó túrák halmaza legyen, $\mathcal{T}_1$ pedig az e-t tartalmazó túrák halmaza. Legyen P_0 és P_1 a megadott halmazokban a legrövidebb túrák hosszának a meghatározása. Ekkor a P-re vonatkozó alsó korlátot számoló lineáris programot egyszerűen ki kell egészíteni az első esetben az $x_e = 0$, a második esetben az $x_e = 1$ feltétellel, amelyek megoldása után valóban a $\mathcal{T}_0$-ra, illetve a $\mathcal{T}_1$-re vonatkozó alsó korlátokat kapunk.

Ezután válasszunk ki egy még nem vizsgált problémát (például P_0), és aztán egy meghatározandó f él szerint bontsuk tovább a $\mathcal{T}_0$ halmazt, előállítva így a P_{00} és P_{01} problémákat. A problámáknak az ily módon való kettéosztogatása ábrázolható egy bináris fával, amelynek pontjaiban a problémák szerepelnek.

Hogyan korlátozzunk? Azaz hogyan számoljuk az alsó korlátokat. Erre már megadtuk a választ: a lineáris programozási relaxálttal, illetve további vágósíkokkal. (Kiegészítés: mivel a vágósíkok megkeresésében konkrét lineáris programok konkrét optimális megoldásai segítenek, ezért célszerű az előállított vágások külön helyen történő összegyűjtése, mivel azok minden más problémára is alkalmazhatóak (lásd 7.3. tétel), és aztán tényleges alkalmazása.

Melyik levélben csücsülő problémát válasszuk ki kettévágásra érdemesnek? Többféle megközelítés létezik. Egyik kézenfekvő hozzáállás az, amikor mindig a legkisebb alsó korlátot adó problémát választjuk ki, hiszen ekkor fog eséllyel tovább nőni a pillanatnyilag legjobb alsó korlát, azaz a levelek alsó korlátainak minimuma. (Mivel így a fa növekedése nagyon kiegyensúlyozott lehet, igen sok probléma tárolására kerülhet sor, ami komoly méretű tárhelyet igényelhet.) A korlátozás és szétválasztás módszerének általános feladatokra való alkalmazásának vizsgálatakor más egyéb megközelítéseket is megvizsgálunk (5.2. fejezet).

Hogyan vágjuk szét a feladatokat? Azt már láttuk, hogy egy e él kiválasztása a legkézenfekvőbb (és valóban leghasznosabb) módja ennek, de mi alapján válasszuk ki az e élet? Mivel a legnagyobb változásokban vagyunk érdekeltek, ezért megfelelő választásnak tűnik a kiválasztott probléma alsó korlátját adó lineáris program megoldásában az 1/2-hez legközelebb értékű változóhoz tartozó él. Ha több él is „ilyen", akkor ezek közül a célfüggvényben a legnagyobb együtthatóval rendelkező esetében remélhetjük az alsó korlátunk legmarkánsabb javulását.

Megjegyezzük, hogy a fentiek végrehajtása során az is megeshet, hogy valamelyik probléma lineáris programozási relaxáltjának megoldása közben egyszer csak egy túra karakterisztikus vektorát kapjuk, aminek a hossza talán még az eljárás kezdetekor kezünkben lévő túra hosszánál még jobb. Ez örömteli eset.

8. fejezet

Egészértékű programozás fix dimenzióban

Ebben a fejezetben megmutatjuk, hogy fix dimenzióban az egészértékű programozási feladat polinom időben megoldható. Bináris feladatnál ez persze könnyű, hiszen d dimenzió esetén csak végig kell nézni a 2^d lehetőséget. De általános esetben korántsem ilyen egyszerű a dolgunk: ha M egy felső korlát a változókra, akkor M^d lehetőséget kellene végignézni, de az input lehet hogy csak $O(\log M)$ hosszú.

A módszer, amit ismertetni fogunk, nagy vonalakban úgy írható le, hogy az LP relaxált legalább α célfüggvény-értékű megoldásai vagy egy „sovány” poliédert alkotnak, és akkor a poliédert alacsonyabb dimenziós szeletekre vághatjuk, vagy „kövér” poliédert alkotnak, és könnyen találunk egész megoldást. A poliéderek testalkatának precíz definiálásához azonban szükség van a rácsok elméletére, úgyhogy először ezzel foglalkozunk.

8.1. Rácsok

Definíció. Adott egy $B = [b_1, \ldots, b_d] \in \mathbb{R}^{n\times d}$ mátrix, aminek oszlopai lineárisan függetlenek. Az mondjuk, hogy

$$\mathcal{L} = \mathcal{L}(B) = \{B\lambda : \ \lambda \in \mathbb{Z}^d\}$$

a B által generált rács, és B az $\mathcal{L}$ rács egy bázisa.

8.1. tétel. *Egy $\mathcal{L} \subseteq \mathbb{R}^n$ részhalmaz pontosan akkor rács, ha $\mathbb{R}^n$-nek diszkrét részcsoportja, azaz olyan részcsoport, ami tetszőleges korlátos halmazból csak véges sok elemet tartalmaz.*

Bizonyítás. Először lássuk be, hogy ha $B \in \mathbb{R}^{n\times d}$ d rangú mátrix, akkor $\mathcal{L} = \mathcal{L}(B)$ diszkrét részcsoport (az nyilvánvaló, hogy részcsoportja $\mathbb{R}^n$-nek). Egészítsük ki B-t új oszlopokkal nemszinguláris mátrixszá, legyen ez Q. Jelölje α a Q^{-1} mátrix elemeinek legnagyobb abszolút értékét. Adott $z = Q\lambda \in \mathcal{L}$-re $\lambda = Q^{-1}z$, amiből a $|\lambda_i| \leq \alpha n\|z\|$ becslés adódik ($i = 1, \ldots, d$). Tehát ha egy korlátos halmaz rácspontjait nézzük, akkor az előállításukban szereplő λ_i együtthatók abszolút értékei egy közös korlát alatt vannak, tehát véges sok λ jöhet szóba, azaz véges sok rácspont van a halmazban.

A másik irányhoz tekitsük $\mathbb{R}^n$ egy diszkrét $\mathcal{L}$ részcsoportját. Legyen $g_1, \ldots, g_d \in \mathcal{L}$ az $\mathcal{L}$-t tartalmazó legszűkebb altér egy bázisa, és legyen

$$X = \{x \in \mathbb{R}^n : x = \sum_{i=1}^{d} \lambda_i g_i, \ 0 \leq \lambda_i < 1 (i = 1, \ldots, d)\}.$$

Az X halmaz korlátos, tehát az $X \cap \mathcal{L}$ halmaz véges. Válasszuk meg $g_1, \dots, g_d$-t úgy, hogy az egész együtthatós kombinációikként előálló $\mathcal{L}$-beli pontok halmaza tartalmazásra nézve maximális.

8.2. állítás. $X \cap \mathcal{L} = \{0\}$.

Bizonyítás. Indirekt tegyük fel, hogy $X \cap \mathcal{L}$ tartalmaz nemnulla vektort, és válasszuk azt az $x^* = \sum_{i=1}^d \lambda_i g_i \in X \cap \mathcal{L}$ nemnulla vektort, aminek az előállításában λ lexikografikusan minimális (ilyen van, mert a diszkrétség miatt $X \cap \mathcal{L}$ véges). Legyen λ_i a legkisebb indexű pozitív együttható. Vegyük azt a k számot, amire $(k-1)\lambda_i < 1$, de $k\lambda_i \geq 1$, és tekintsük a

$$z = (k\lambda_i - 1)g_i + \sum_{j=i+1}^{d} (k\lambda_j - \lfloor k\lambda_j \rfloor) g_j$$

rácsvektort. Ennek az együtthatóvektora lexikografikusan kisebb λ-nál $k\lambda_i - 1 < \lambda_i$ miatt, tehát z csak a nullvektor lehet. Ez azt jelenti, hogy

$$g_i = kx^* - \sum_{j=i+1}^{d} \lfloor k\lambda_j \rfloor) g_j,$$

azaz ha g_i-t lecseréljük x^*-ra a bázisban, akkor szigorúan nagyobb halmaz áll elő egész együtthatós kombinációként, ellentmondásban $g_1, \dots, g_d$ választásával. $\square$

Ha $X \cap \mathcal{L} = \{0\}$, akkor $\mathcal{L} = \mathcal{L}(g_1, \dots, g_d)$, tehát $\mathcal{L}$ rács. $\square$

8.3. tétel. *Legyen $B \in \mathbb{R}^{n \times d}$ egy $\mathcal{L}$ rács bázisa.*
a) Ha $U \in \mathbb{Z}^{d \times d}$ unimoduláris, akkor $B' = BU$ is bázisa $\mathcal{L}$-nek.
b) Ha B és B' bázisa $\mathcal{L}$-nek, akkor létezik U unimoduláris mátrix amire $B' = BU$.
c) Ha B és B' bázisa $\mathcal{L}$-nek, és $n = d$, akkor $|\det(B)| = |\det(B')|$.

Bizonyítás. Legyen B egy $\mathcal{L}$ rács bázisa, és legyen B' az $\mathcal{L}$-t tartalmazó legszűkebb altér egy bázisa. Ekkor van egy egyértelmű U invertálható mátrix, amire $B = B'U$. Világos, hogy B' pontosan akkor bázisa $\mathcal{L}$-nek, ha mind U, mind U^{-1} egész mátrix, hiszen pontosan ekkor lesznek egész együtthatók mind B oszlopainak előállításában a B' bázisban, mind B' oszlopainak előállításában a B bázisban. Ez pedig azzal ekvivalens, hogy U unimoduláris. A determinánsra vonatkozó állítás következik abból, hogy $|\det(U)| = 1$. $\square$

A tétel értelmében definiálhatjuk a $\det(\mathcal{L}) = |\det(B)|$ (B bázisa $\mathcal{L}$-nek) értéket, ami megegyezik a rács-parallelepipedon térfogatával. Ez a mennyiség szerepel Minkowski híres tételében, ami centrálisan szimmetrikus konvex halmazokban lévő rácspontokról szól. Mielőtt kimondanánk és bizonyítanánk Minkowski tételét, lássunk be egy Blichfeldt nevéhez köthető lemmát, amiből már könnyen következik az előbbi.

8.4. lemma (Blichfeldt)**.** *Legyen $\mathcal{L} \subseteq \mathbb{R}^n$ egy n-dimenziós rács, és $T \subseteq \mathbb{R}^n$ egy mérhető halmaz, amire* $\operatorname{vol}(T) > \det(\mathcal{L})$. *Ekkor léteznek $x, y \in T$ különböző pontok, hogy $x - y \in \mathcal{L}$.*

Bizonyítás. Legyen P a rács-parallelepipedon, és $v \in \mathcal{L}$-re legyen $T_v = (T - v) \cap P$. Mivel $\sum_{v \in \mathcal{L}} \operatorname{vol}(T_v) = \operatorname{vol}(T) > \det(\mathcal{L}) = \operatorname{vol}(P)$, a T_v halmazok nem lehetnek mind diszjunktak, tehát van $z \in T_{v_1} \cap T_{v_2}$. Ekkor $z + v_1 \in T$, $z + v_2 \in T$, és a különbségük $v_1 - v_2$, ami $\mathcal{L}$-ben van. $\square$

8.5. tétel (Minkowski)**.** *Legyen $\mathcal{L} \subseteq \mathbb{R}^n$ egy n-dimenziós rács, és $S \subseteq \mathbb{R}^n$ egy konvex, 0-ra szimmetrikus halmaz, amire* $\operatorname{vol}(S) > 2^n \det(\mathcal{L})$. *Ekkor S-ben van nem-0 rácspont.*

Bizonyítás. Legyen $T = \frac{1}{2}S$. Ez teljesíti a 8.4. lemma feltételeit, tehát létezik $x, y \in S$, hogy $\frac{1}{2}(x-y) \in \mathcal{L}$. A szimmetria miatt $-y \in S$, a konvexitás miatt pedig $\frac{1}{2}(x-y) \in S$. □

Megjegyzés. A Minkowski tételben, ha S kompakt, akkor elég feltenni, hogy $\mathrm{vol}(S) \geq 2^n \det(\mathcal{L})$. Valóban, a tétel igaz $(1+\epsilon)S$-re akármilyen kis ϵ-ra, tehát ha S nem tartalmazna rácspontot, akkor bármilyen közel lenne hozzá rácspont. De ez lehetetlen, hiszen a rács diszkrét, így S kompaktsága miatt egy környezete csak véges sok rácspontot tartalmazhat.

A tétel következménye, hogy tetszőleges normára a legrövidebb rácsvektor normája becsülhető a rácsdeterminánssal, hiszen bármilyen normára tekintett egységgömb 0-ra szimmetrikus és konvex. Ha a norma szerinti egységgömb térfogata $2^n\alpha$, akkor van legfeljebb $(\det(\mathcal{L})/\alpha)^{1/n}$ normájú rácsvektor.

8.1.1. Kitérő: ILP feladat alternatív formulációja

Tudjuk, hogy ha $A \in \mathbb{Z}^{m\times n}$ egy m rangú mátrix, akkor létezik U unimoduláris mátrix, hogy $AU = [B, 0]$, ahol B egy reguláris alsó-háromszögmátrix. Ezt nevezzük az A mátrix egész normál formájának.

8.6. tétel. *Legyen $A \in \mathbb{Z}^{m\times n}$ egy m rangú mátrix és $AU = [B, 0]$ az egész normál formája. Ekkor*
a) $\{x \in \mathbb{Z}^n : Ax = b\}$ pontosan akkor nemüres, ha $B^{-1}b \in \mathbb{Z}^m$.
b) Ha $\{x \in \mathbb{Z}^n : Ax = b\}$ nemüres, akkor minden eleme felírható $x = U_1B^{-1}b + U_2z$ ($z \in \mathbb{Z}^{n-m}$) alakban, ahol $U = [U_1, U_2]$.
c) $\mathcal{L} = \{x \in \mathbb{Z}^n : Ax = 0\}$ rács, és U_2 oszlopvektorai bázist alkotnak.

Ennek a tételnek a segítségével egy $\max\{cx : Ax = b, x \in \mathbb{Z}^n_+\}$ feladatot más lineáris rendszerekkel is leírhatunk. Feltesszük hogy A m rangú egész mátrix, $b \in \mathbb{Z}^m$ és $c \in \mathbb{Z}^n$. Legyen $\mathcal{F} = \{x \in \mathbb{Z}^n_+ : Ax = b\}$, és legyen az A mátrix egész normál formája $AU = [B, 0]$. Ha $B^{-1}b$ nem egész, akkor nincs megoldása a feladatnak, tehát tegyük fel hogy egész. Ekkor $\exists x_0 \in \mathbb{Z}^n : Ax_0 = b$, és így

$$x \in \mathcal{F} \Leftrightarrow \exists y \in \mathbb{Z}^n : -x_0 \leq y, Ay = 0, x = x_0 + y.$$

Legyen $\mathcal{L} = \{y \in \mathbb{Z}^n : Ay = 0\}$. Ha D ennek a rácsnak egy bázisa, akkor a feladatunk felírható

$$\max\{cDz : Dz \geq -x_0, z \in \mathbb{Z}^{n-m}\}$$

alakban. Különböző bázisok (amiket különböző unimoduláris mátrixokkal kaphatunk) különböző leírásokat adnak.

8.1.2. Redukált bázisok

Adott $b_1, \dots, b_d \in \mathbb{Q}^n$ lineárisan független vektorhalmazra jelölje $b_1^*, \dots, b_d^*$ a Gram-Schmidt ortogonalizációt, azaz

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*,$$

ahol

$$\mu_{ij} = \frac{b_i b_j^*}{\|b_j^*\|^2}.$$

Ennek megvannak a következő jó tulajdonságai:

- $\langle b_1, \ldots, b_i \rangle = \langle b_1^*, \ldots, b_i^* \rangle$ $(i = 1, \ldots, d)$ (ahol $\langle x_1, \ldots, x_k \rangle$ az $x_1, \ldots, x_k$ vektorok által generált altért jelöli),
- $b_1^*, \ldots, b_d^*$ páronként merőlegesek,
- $\|b_j^*\| \leq \|b_j\|$ $(j = 1, \ldots, d)$
- Ha $d = n$, akkor $\det(\mathcal{L}(b_1, \ldots, b_n)) = \prod_{j=1}^n \|b_j^*\|$.

Ha egy $\mathcal{L} = \mathcal{L}(b_1, \ldots, b_n)$ rácsnak $b_1^*, \ldots, b_n^*$ is bázisa, akkor könnyen látható, hogy egy $x = \sum_{j=1}^n \lambda_j b_j^*$ vektorhoz a legközelebbi rácspont $\sum_{j=1}^n \lfloor \lambda_j \rceil b_j^*$, ahol $\lfloor a \rceil$ az a-hoz legközelebbi egész számot jelöli.

Általános esetben már a legrövidebb rácsvektor probléma is NP-nehéz, de ha találunk egy nagyjából merőleges bázist, akkor viszonylag gyorsan meg tudjuk találni. A $\prod_{i=1}^n \|b_i\| / \det(\mathcal{L})$ értéket a bázis **merőlegességi defektusának** nevezzük.

8.7. állítás. *Ha a bázis merőlegességi defektusa legfeljebb c, és $v = \sum_{j=1}^n \lambda_j b_j$ a legrövidebb rácsvektor, akkor $|\lambda_j| \leq c$ minden j-re.*

Bizonyítás. Indirekt bizonyítunk; az indexek felcserélésével feltehető, hogy $|\lambda_n| > c$. Ha v-t a b_j^* vektorok lineáris kombinációjaként írjuk fel, akkor az n-edik együttható ugyanúgy λ_n, így $\|v\|^2 \geq \lambda_n^2 \|b_n^*\|^2 > c^2 \|b_n^*\|^2 \geq \|b_n\|^2$, ahol az utolsó egyenlőtlenség azért igaz, mert egyrészt $(\prod_{i=1}^n \|b_i\|)/(\prod_{i=1}^n \|b_i^*\|) \leq c$, másrészt $\|b_j^*\| \leq \|b_j\|$ minden j-re. Tehát azt kaptuk, hogy $\|v\| > \|b_n\|$, ellentmondásban v minimalitásával. □

Az állítás következményeként $(2c+1)^n$ vektor végignézésével megkereshetjük a legrövidebb rácsvektort. Ez csak kis c értékekre jó algoritmus, általában nem tudunk olyan bázist találni, aminek ilyen kicsi lenne a merőlegességi defektusa. Viszont tudunk olyan bázis-tulajdonságot definiálni, ami egyrészt nem túl nagy merőlegességi defiektust implikál, másrészt polinom időben konstruálható ilyen tulajdonságú bázis. Ez lesz a *redukált bázis* fogalma. Megjegyzendő, hogy másféle redukált bázis definíció is használatos az irodalomban, ami többé-kevésbé hasonló tulajdonságokkal rendelkezik.

Definíció. Legyen $\mathcal{L} = \mathcal{L}(b_1, \ldots, b_n)$, és legyen $b_1^*, \ldots, b_n^*$ a Gram-Schmidt ortogonalizáció. A $b_1, \ldots, b_n$ bázis *redukált*, ha

(i) $|\mu_{ij}| \leq \frac{1}{2}$ minden $1 \leq j < i \leq n$-re,

(ii) $\|b_{i+1}^*\|^2 \geq \frac{1}{2}\|b_i^*\|^2$ $(i = 1, \ldots, n-1)$.

8.8. tétel. *Redukált bázis esetén igaz, hogy*

a) $\|b_1\| \leq 2^{(n-1)/4} \det(\mathcal{L})^{1/n}$,

b) $\prod_{i=1}^n \|b_i\| \leq 2^{n(n-1)/4} \det(\mathcal{L})$.

Bizonyítás. a) A (ii) tulajdonságból következik, hogy

$$\|b_j^*\|^2 \geq \left(\frac{1}{2}\right)^{j-1} \|b_1^*\|^2 = \left(\frac{1}{2}\right)^{j-1} \|b_1\|^2 \quad (j = 1, \ldots, n).$$

Ebből

$$\det(\mathcal{L})^2 = \prod_{j=1}^n \|b_j^*\|^2 \geq \left(\frac{1}{2}\right)^{n(n-1)/2} \|b_1\|^{2n}.$$

b) A redukált bázis (i) és (ii) tulajdonságait használva

$$\|b_i\|^2 = \|b_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2 \|b_j^*\|^2 \le \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1} \|b_j^*\|^2$$
$$\le \|b_i^*\|^2 + \frac{1}{4}\sum_{j=1}^{i-1} 2^{i-j}\|b_i^*\|^2 = \|b_i^*\|^2(1+\frac{1}{4}(2^i-2)) \le 2^{i-1}\|b_i^*\|^2$$

tehát

$$\prod_{i=1}^{n} \|b_i\|^2 \le 2^{n(n-1)/2} \prod_{i=1}^{n} \|b_i^*\|^2 = 2^{n(n-1)/2} \det(\mathcal{L})^2.$$

□

8.2. LLL algoritmus: redukált bázis keresése

Legyen $b_1, \ldots, b_n$ egy rács bázisa. Először egy olyan algoritmust adunk, ami kiszámol egy új bázist, aminek az ortogonalizációja ugyanaz, és $\mu_{ij} \le \frac{1}{2}$ minden $1 \le j < i \le n$-re.

Együttható-csökkentő algoritmus

For $i = 2, \ldots, n$
    For $j = i-1, \ldots, 1$
        Ha $|\mu_{ij}| > \frac{1}{2}$: $b_i := b_i - \lfloor \mu_{ij} \rceil b_j$;
        Számoljuk újra a Gram-Schmidt ortogonalizáció együtthatóit.
Return $b_1, \ldots, b_n$

8.9. állítás. *A kapott bázisnál* $|\mu_{ij}| \le \frac{1}{2}$ $(1 \le j < i \le n)$.

Bizonyítás. Tegyük fel hogy $k \in \{2, \ldots, n\}$ és $l \in \{1, \ldots, k-1\}$ indexeknél tartunk, és $\mu_{kl} > \frac{1}{2}$, de $\mu_{kj} \le \frac{1}{2}$ $(j = l+1, \ldots, k-1)$.

Legyen $q_1, \ldots, q_n$ az új bázis, és μ'_{ij} az új együtthatók. Ekkor

- $i < k$ esetén $q_i = b_i$, tehát $q_i^* = b_i^*$ és $j < i < k$-ra $\mu'_{ij} = \mu_{ij}$.
- $i = k$, $j = l+1, \ldots, k-1$-re $\mu'_{kj} = \mu_{kj}$, mert b_l merőleges b_j^*-ra, és így $b_k - \lfloor \mu_{kl} \rceil b_l$-nek ugyanaz a vetülete mint b_k-nak.
- $i = k$, $j = l$-re:
$$\mu'_{kl} = \frac{q_k q_l^*}{\|q_l^*\|^2} = \frac{(b_k - \lfloor \mu_{kl} \rceil b_l) b_l^*}{\|b_l^*\|^2} = \mu_{kl} - \lfloor \mu_{kl} \rceil$$
és ezért $|\mu'_{kl}| \le \frac{1}{2}$.

A fentiekből látszik, hogy μ_{kl}-t az algoritmus további részében már nem módosítjuk. □

8.10. állítás. *Az algoritmus során a Gram-Schmidt ortogonalizáció nem változik.*

Bizonyítás. Már láttuk, hogy $i < k$ esetén $q_i^* = b_i^*$, $i > k$-ra pedig $q_i = b_i$, úgyhogy elég $i = k$-ra bizonyítani.

$$q_k^* = q_k - \sum_{j=1}^{k-1} \mu'_{kj} q_j^* = b_k - \lfloor \mu_{kl} \rceil b_l - \sum_{j=1}^{k-1} \mu'_{kj} b_j^*,$$

tehát q_k^* és b_k^* különbsége $\langle b_1^*, \dots, b_{k-1}^* \rangle$-beli, de mindketten merőlegesek erre az altérre, tehát egyenlőek. $\square$

Az együttható-csökkentő algoritmussal téhát bármilyen bázist módosíthatunk a Gram-Schmidt ortogonalizáció változása nélkül úgy, hogy a redukált bázis definíciójában szereplő (i) tulajdonság teljesüljön. Ahhoz, hogy a (ii) tulajdonságot is elérjük, valójában a következő, erősebb tulajdonságot fogjuk elérni:

(ii') $\|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2 \geq \frac{3}{4}\|b_i^*\|^2$ minden $i \in \{1, \dots, n-1\}$-re.

8.11. állítás. *Ha* (i) *és (ii') teljesül, akkor* (ii) *is.*

Bizonyítás. A két tulajdonságot használva

$$\frac{3}{4}\|b_i^*\|^2 \leq \|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2 = \|b_{i+1}^*\|^2 + \|\mu_{i+1,i} b_i^*\|^2 \leq \|b_{i+1}^*\|^2 + \frac{1}{4}\|b_i^*\|^2,$$

tehát $\|b_{i+1}^*\|^2 \geq \frac{1}{2}\|b_i^*\|^2$. $\square$

Megjegyzés. A (ii') feltételben szereplő $b_{i+1}^* + \mu_{i+1,i} b_i^*$ nem más, mint a Gram-Schmidt ortogonalizáció i-edik vektora akkor, ha b_i és b_{i+1} sorrendjét felcseréljük a bázisban.

Az úgynevezett LLL-algoritmus (Lenstra-Lenstra-Lovász algoritmus) az együttható-csökkentést kombinálja azzal, hogy a b_i és b_{i+1} sorrendjét felcseréljük, ha i-re nem teljesül a (ii') tulajdonság.

Bázis redukciós algoritmus

Input: $b_1, \dots, b_n \in \mathbb{Z}^n$ bázis
Output: $\mathcal{L}(b_1, \dots, b_n)$ redukált bázisa

- Kiszámítjuk a Gram-Schmidt ortogonalizációt
- Alkalmazzuk az együttható-csökkentő algoritmust
- Ha van olyan i, hogy $\|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2 < \frac{3}{4}\|b_i^*\|^2$, akkor a legkisebb ilyen i-re b_i-t és b_{i+1}-et felcseréljük; vissza az első lépésre.
- A végén kapott bázis redukált bázis.

Az algoritmus helyességéhez csak azt kell belátni, hogy polinom időben végetér. Ezt nem bizonyítjuk precízen, csak a bizonyítás lényegét ismertetjük. Ez a lényeg abban áll, hogy az

$$F(b_1, \dots, b_n) = \prod_{i=1}^{n} \prod_{j=1}^{i} \|b_j^*\|^2$$

mennyiségre a következők teljesülnek az algoritmus során:

- $F(b_1, \dots, b_n)$ Mindig egész szám. Ez abból látszik, hogy $\prod_{j=1}^{i} \|b_j^*\|^2$ felírható egy egész mátrix deteminánsaként:

  $$\prod_{j=1}^{i} \|b_j^*\|^2 = \det((B_i^*)^T B_i^*) = \det((B_i^* R_i)^T B_i^* R_i) = \det(B_i^T B_i),$$

  ahol B_i a b_j $(j = 1, \dots, i)$ oszlopvektorokból álló mátrix, B_i^* a b_j^* $(j = 1, \dots, i)$ oszlopvektorokból álló mátrix, és R_i az a felső háromszög-mátrix, ami a főátlóban 1-eket, felette pedig a megfelelő μ értékeket tartalmazza.

- Az együttható-csökkentésnél $F(b_1, \ldots, b_n)$ nem változik.

- b_i és b_{i+1} felcserélésekor legalább $\frac{3}{4}$-ére csökken.

A harmadik állításhoz tekintsük a felcserélési lépést, legyen $b_1, \ldots, b_n$ az eredeti bázis, és $q_1, \ldots, q_n$ az új bázis. Ekkor $i < k$ és $i > k+1$ esetén $q_i^* = b_i^*$. Azt is tudjuk, hogy $\prod_{j=1}^n \|q_j^*\| = \prod_{j=1}^n \|b_j^*\|$ hisz ugyanahhoz a rácshoz tartoznak; tehát

$$\|q_k^*\|\|q_{k+1}^*\| = \|b_k^*\|\|b_{k+1}^*\|.$$

Másrészt

$$q_k^* = b_{k+1} - \sum_{j=1}^{k-1} \mu'_{kj} b_j^* = b_{k+1} - \sum_{j=1}^{k-1} \mu_{k+1,j} b_j^* = b_{k+1}^* + \mu_{k+1,k} b_k^*,$$

tehát

$$\|q_k^*\|^2 = \|b_{k+1}^* + \mu_{k+1,k} b_k^*\|^2 < \frac{3}{4}\|b_k^*\|^2.$$

8.2.1. Legközelebbi vektor közelítése

A legközelebbi vektor problémánál az a feladat, hogy egy rácsban találjuk meg a legközelebbi elemet egy adott p ponthoz. Ez NP-nehéz, úgyhogy egy közelítő algoritmust ismertetünk, ami legfeljebb $\sqrt{2^n - 1}$-szer olyan messze lévő rácsvektort ad, mint az optimális.

Mivel egy rácshoz tudunk polinom időben redukált bázist találni, feltesszük, hogy a rács eleve egy redukált bázissal van megadva.

Legközelebbi vektor közelítő algoritmus

Input: $b_1, \ldots, b_n \in \mathbb{Z}^n$ redukált bázis, $p \in \mathbb{Q}^n$
Output: $v \in \mathcal{L}$, amire

$$\|v - p\|^2 \leq (2^n - 1) \min\{\|q - p\|^2 : q \in \mathcal{L}\},$$
$$v - p = \sum_{i=1}^n \lambda_i b_i^* \text{ ahol } |\lambda_i| \leq \frac{1}{2}.$$

- Legyen $p_n = p$.

- For $i = n, \ldots, 1$

  1) σ_{ij} együtthatók kiszámolása, amikre $p_i = \sum_{j=1}^i \sigma_{ij} b_j^*$. (Ez értelmes, mert p_i mindig benne lesz a $\langle b_1^*, \ldots, b_i^* \rangle$ altérben)
  2) $\lambda_i = \lfloor \sigma_{ii} \rceil - \sigma_{ii}$
  3) $p_{i-1} = p_i - \lfloor \sigma_{ii} \rceil b_i + \lambda_i b_i^*$ (így $p_{i-1} \in \langle b_1^*, \ldots, b_{i-1}^* \rangle$)

- Return $v = \sum_{i=1}^n \lfloor \sigma_{ii} \rceil b_i = p + \sum_{i=1}^n \lambda_i b_i^*$.

Az algoritmus végén $p_0 = 0$, tehát tényleg $v = p + \sum_{i=1}^n \lambda_i b_i^*$, és $|\lambda_i| \leq \frac{1}{2}$ minden i-re. Később fontos lesz, hogy itt még nem használtuk, hogy a bázis redukált.

8.12. állítás. $\|v - p\|^2 \leq (2^n - 1) \min\{\|q - p\|^2 : q \in \mathcal{L}\}$.

Bizonyítás. Legyen q a p-hez legközelebbi rácspont, legyen $q - p = \sum_{i=1}^n \sigma_i b_i^*$, valamint legyen $v - q = \sum_{i=1}^n \alpha_i b_i$ ahol $\alpha_i \in \mathbb{Z}$ (hiszen $v - q$ rácsvektor).

Ha k a legnagyobb index, amire $\sigma_k \neq \lambda_k$, akkor $i > k$-ra $\alpha_i = 0$, és $\lambda_k - \sigma_k = \alpha_k \in \mathbb{Z}$. Így $|\sigma_k| \geq |\alpha_k| - |\lambda_k| \geq \frac{1}{2}$. Ebből a következő becsléseket kapjuk:

$$\|q - p\|^2 \geq \frac{1}{4}\|b_k^*\|^2 + \sum_{i=k+1}^{n} \lambda_i^2 \|b_i^*\|^2,$$

$$\|v - p\|^2 \leq \frac{1}{4}\sum_{i=1}^{k} \|b_i^*\|^2 + \sum_{i=k+1}^{n} \lambda_i^2 \|b_i^*\|^2$$

Innen az állítás kijön azzal, hogy a redukált bázisnál $1 \leq i \leq k$ esetén $\|b_i^*\|^2 \leq 2^{k-i}\|b_k^*\|^2$. □

8.3. Polinomiális algoritmus fix dimenziós egészértékű programozásra

A fix dimenziós egészértékű programozási algoritmus két összetevőből áll. Az egyik a legközelebbi vektor közelítő algoritmus, ahol ugyan a közelítés faktora a dimenzióban exponenciális, azonban ez fix dimenzióban nem okoz gondot. A másik összetevő egy adott korlátos poliéderre egy őt tartalmazó "nem túl nagy" ellipszoid keresése. Ezt nem ismertetjük részletesen, csak kimondjuk a tételt az algoritmus által adott ellipszoid egy tulajdonságáról. A bizonyítás történhet ellipszoid módszerrel, vagy belsőpontos algoritmussal.

Egy n-dimenziós ellipszoid egy $D \in \mathbb{R}^{n\times n}$ nemszinguláris mátrix és egy $d \in \mathbb{R}^n$ vektor segítségével adható meg. Az $E(D, d)$ elliposzidot úgy kapjuk, hogy az egységgömbre alkalmazzuk a $Dx + d$ affin transzformációt, azaz

$$E(D, d) = \left\{x \in \mathbb{R}^n : (D^{-1}(x - d))^T D^{-1}(x - d) \leq 1\right\}.$$

Az ellipszoid térfogata $|\det(D)|$-vel arányos. Az alábbi tételt bizonyítás nélkül használjuk.

8.13. tétel. *Legyen $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ egy korlátos teljes dimenziós poliéder, ahol A egész mátrix és b egész vektor. Ekkor polinom időben megtalálható egy olyan E ellipszoid, amire $\frac{1}{n+1}E \subseteq P \subseteq E$.* □

A továbbiakban feltesszük, hogy a $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ poliéder korlátos és teljes dimenziós. A 8.13. Tételt felhasználva a megengedettségi feladatra adunk algoritmust, azaz annak eldöntésére, hogy van-e P-nek egész pontja. Megjegyezzük, hogy ennek segítségével az optimalizálás is megoldható, bináris kereséssel az optimális célfüggvény-értékre.

Az algoritmus fő része a dimenzió csökkentése: A poliéderben vagy találunk egy egész pontot, vagy megadunk konstans darab alacsonyabb dimenziós poliédert, amik közül pontosan akkor tartalmaz valamelyik egész pontot, ha az eredeti poliéder tartalmazott.

Dimenzió-csökkentő algoritmus

Input: $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$, $P = \{x \in \mathbb{R}^n : Ax \le b\}$ teljes dimenziós és korlátos
Output: $z \in P \cap \mathbb{Z}^n$, vagy legfeljebb $f(n)$ darab $(n-1)$ dimenziós poliéder: P-nek pontosan akkor van egész pontja ha ezek valamelyikének van.

- Kiszámolunk egy $E(D,d)$ ellipszoidot, amire $\frac{1}{n+1}E \subseteq P \subseteq E$.
- Legyen $P' = \{x \in \mathbb{R}^n : ADx \le b\}$, és legyen $p = D^{-1}d$.
- Legyen $\mathcal{L} = \mathcal{L}(D_1^{-1}, \dots, D_n^{-1})$, ahol D_j^{-1} a D^{-1} mátrix j-edik oszlopa. Számoljuk ki a bázis-redukciós algoritmussal $\mathcal{L}$ egy redukált bázisát: $b_1, \dots, b_n$. Számozzuk át a b_i vektorokat hossz szerint növekvő sorrendbe: $q_1, \dots, q_n$. Számoljuk ki a Gram-Schmidt ortogonalizációt.
- A legközelebbi vektor közelítő algoritmussal számoljunk ki egy p-hez közelítően legközelebbi rácsvektort: $v \in \mathcal{L}$. Ha $v \in P'$, kész vagyunk, hisz $Dv \in P \cap \mathbb{Z}^n$.
- Ha $v \notin P'$: Legyen $c = q_n^*/\|q_n^*\|^2$.
- Minden $k \in K = [c^Tp - \|c\|, c^Tp + \|c\|] \cap \mathbb{Z}$ egészre legyen
$$P_k = \{z \in \mathbb{R}^{n-1} : ADQ \cdot (z,k) \le b\},$$
ahol Q a $q_1, \dots, q_n$ oszlopok által alkotott mátrix.
- Return P_k $(k \in K)$.

8.14. állítás. *Tetszőleges $x \in P'$-re $c^Tx \in [c^Tp - \|c\|, cp + \|c\|]$. Ha $v \notin P'$, akkor* $\max\{c^Tx : x \in P'\} - \min\{c^Tx : x \in P'\} < \sqrt{n}(n+1)2^{\frac{n(n-1)}{4}}$.

Bizonyítás. Jegyezzük meg, hogy mikor megváltoztatjuk a redukált bázis sorrendjét, akkor megszűnik redukált bázis lenni.

Legyen $v - p = \sum_{i=1}^n \lambda_i q_i^*$ (itt $|\lambda_i| \le 1/2$, mivel ez akkor is igaz a legközelebbi vektor közelítő algoritmusnál, ha nem redukált bázisunk van). Jelölje $B(p,r)$ a p körüli r sugarú gömböt. Ekkor
$$B(p, \frac{1}{n+1}) \subseteq P' \subseteq B(p,1).$$
Mivel $v \notin P'$, így $\|v-p\| > \frac{1}{n+1}$, tehát
$$\frac{1}{n+1} < \|v-p\| \le \sqrt{\frac{1}{4}\sum_{i=1}^n \|q_i^*\|^2} \le \frac{1}{2}\sqrt{n}\|q_n\|,$$
azaz $1/\|q_n\| < \frac{1}{2}(n+1)\sqrt{n}$. A redukált bázisokról bizonyítottak alapján
$$\prod_{i=1}^n \|q_i\| = \prod_{i=1}^n \|b_i\| \le 2^{\frac{n(n-1)}{4}} \det(\mathcal{L}) = 2^{\frac{n(n-1)}{4}} \prod_{i=1}^n \|q_i^*\| \le 2^{\frac{n(n-1)}{4}} \|q_n^*\| \prod_{i=1}^{n-1} \|q_i\|,$$
tehát
$$2^{-\frac{n(n-1)}{4}} \|q_n\| \le \|q_n^*\|.$$

Most már készen állunk az állítás bizonyítására:

$$\begin{aligned}
\max\{c^T x : x \in P'\} &- \min\{c^T x : x \in P'\} \\
&\le \max\{c^T x : x \in B(p,1)\} - \min\{c^T x : x \in B(p,1)\} \\
&\le c^T \left(p + \frac{c}{\|c\|}\right) - c^T \left(p - \frac{c}{\|c\|}\right) = 2\|c\| \\
&= \frac{2}{\|q_n^*\|} \le \frac{2 \cdot 2^{\frac{n(n-1)}{4}}}{\|q_n\|} < (n+1)\sqrt{n} \cdot 2^{\frac{n(n-1)}{4}}.
\end{aligned}$$

□

8.15. állítás. *P-nek pontosan akkor van egész pontja ha a P_k ($k \in K$) poliéderek valamelyikének van.*

Bizonyítás. Világos, hogy minden $x \in P' \cap \mathcal{L}$-re $x = Q(z,k)$ valamilyen $z \in \mathbb{Z}^{n-1}$-re és $k \in \mathbb{Z}$-re. Mivel $\|cq_n\| = 1$ és $i < n$ esetén $cq_i = 0$, $cx = k \in \mathbb{Z}$. Az előző állítás alapján $cx \in [cp - \|c\|, cp + \|c\|]$, tehát $cx = k \in K$. □

8.16. tétel. *Fix n-re van polinomiális algoritmus, ami eldönti, hogy $P \cap \mathbb{Z}^n$ üres-e, és ha nem, talál egy ilyen vektort.*

Bizonyítás. Hogy rekurzívan használhassuk a Dimenzió-csökkentő algoritmust, azt kell megmutatnunk, hogy a $P_k = \{z \in \mathbb{R}^{n-1} : ADQ(z,k) \le b\}$ poliéderek is teljesítik a feltételeket. Ehhez az kell, hogy ADQ egész mátrix legyen. Tudjuk, hogy D^{-1} és Q ugyanannak a $\mathcal{L}$ rácsnak a bázisa, így létezik U unimoduláris mátrix, hogy $D^{-1}U = Q$. Így $ADQ = ADD^{-1}U = AU \in \mathbb{Z}^{m \times n}$. □

Megjegyzés. Kannan megmutatta, hogy olyan dimenzió-csökkentő algoritmus is adható, ahol csak $O(n^{\frac{5}{2}})$ darab alacsonyabb dimenziós poliéderre bontjuk a feladatot. Barvinok bebizonyította, hogy fix dimenzióban a következő feladat is polinomiálisan megoldható: adott P poliéderre számoljuk meg, hogy hány egész pontja van.

8.4. Racionális számok egyidejű közelítése kis nevezőjű törtekkel

8.17. tétel. *Legyenek $\alpha_1, \ldots, \alpha_n$ valós számok, és $0 < \varepsilon < 1$. Ekkor léteznek $c_1, \ldots, c_n$ és d egészek, amikre*

$$1 \le d \le \varepsilon^{-n}$$

és

$$|\alpha_i d - c_i| \le \varepsilon \quad (i = 1, \ldots n).$$

Bizonyítás. Legyen $a = (-\alpha_1, \ldots, -\alpha_n, \varepsilon^{n+1})$, és legyen $\mathcal{L} = \mathcal{L}(e_1, \ldots, e_n, a)$. Ekkor $\det(\mathcal{L}) = \varepsilon^{n+1}$.

A Minkowski konvex test tétel alapján létezik v rácsvektor, amire $\|v\|_\infty \le \varepsilon$. A v rácsvektor felírható $v = c_1e_1 + c_2e_2 + \cdots + c_ne_n + da$ alakban, ahol c_i ($i = 1, \ldots, n$) és d egészek, és $d \ne 0$ mert $\varepsilon < 1$. Feltehető, hogy $d > 0$, mert különben v helyett nézhetjük $-v$-t.

Ekkor egyrészt $v_i = c_i - d\alpha_i$, tehát $|c_i - d\alpha_i| \le \varepsilon$ ($i = 1, \ldots, n$), másrészt $v_{n+1} = \varepsilon^{n+1}d$, tehát $d \le \varepsilon^{-n}$. □

8.18. tétel. *Legyenek* $\alpha_1, \dots, \alpha_n$ *racionális számok, és* $0 < \varepsilon < 1$. *Ekkor van polinomiális algoritmus, ami kiszámol* $c_1, \dots, c_n$ *és* d *egészeket, amikre*

$$1 \leq d \leq 2^{\frac{n(n+1)}{4}} \varepsilon^{-n}$$

és

$$|\alpha_i d - c_i| \leq \varepsilon \quad (i = 1, \dots n).$$

Bizonyítás. Legyen $a = (-\alpha_1, \dots, -\alpha_n, 2^{-\frac{n(n+1)}{4}} \varepsilon^{n+1})$, és legyen $\mathcal{L} = \mathcal{L}(e_1, \dots, e_n, a)$. Ekkor $\det(\mathcal{L}) = 2^{-\frac{n(n+1)}{4}} \varepsilon^{n+1}$.

Számoljuk ki az $\mathcal{L}$ rács $b_1, \dots, b_{n+1}$ redukált bázisát, és legyen $v = b_1$. Ekkor $\|v\| \leq 2^{\frac{n}{4}} \det(\mathcal{L})^{\frac{1}{n+1}} = \varepsilon$. A v rácsvektor felírható $v = c_1 e_1 + c_2 e_2 + \dots + c_n e_n + da$ alakban, ahol c_i $(i = 1, \dots, n)$ és d egészek, és $d \neq 0$ mert $\varepsilon < 1$. Feltehető, hogy $d > 0$, mert különben v helyett nézhetjük $-v$-t.

Ekkor egyrészt $v_i = c_i - d\alpha_i$, tehát $|c_i - d\alpha_i| \leq \varepsilon$ $(i = 1, \dots, n)$, másrészt $v_{n+1} = 2^{-\frac{n(n+1)}{4}} \varepsilon^{n+1} d$, tehát $d \leq 2^{\frac{n(n+1)}{4}} \varepsilon^{-n}$. □

9. fejezet

Közelítő algoritmusok és iterált kerekítés

Legyen $\alpha \geq 1$ valós szám. Egy minimalizálási feladatra egy algoritmus α-*közelítő*, ha polinomiális futási idejű, és tetszőleges inputra az output értéke legfeljebb α-szorosa az optimálisnak. Ennek persze csak nemnegatív célfüggvény esetén van értelme.

9.1. 2-közelítő algoritmus minimális súlyú lefogó csúcshalmaz keresésére

9.1.1. Legkisebb lefogó csúcshalmaz

Adott $G = (V, E)$ irányítatlan gráfnak egy $X \subseteq V$ halmaz *lefogó csúcshalmaza*, ha minden élnek legalább egyik végpontját tartalmazza.

> *Legkisebb lefogó csúcshalmaz probléma*
>
> Adott egy $G = (V, E)$ irányítatlan gráf. Keressünk olyan lefogó csúcshalmazt, aminek az elemszáma minimális.

Az alábbi tétel mutatja, hogy ez a feladat igen nehéz (a bizonyítás meghaladja a jegyzet kereteit).

9.1. tétel. *Ha $\alpha < \frac{7}{6}$-ra van α-közelítő polinomiális algoritmus a legkisebb lefogó csúcshalmaz feladatra, akkor $P = NP$.*

A feladatra nem ismert $\alpha < 2$ esetén α-közelítő polinomiális futási idejű algoritmus. Viszont 2-közelítő algoritmust könnyű adni. Kezdetben legyen $U = \emptyset$ a lefogó csúcshalmaz.

1. Válasszunk egy tetszőleges uv élt a gráfban.
2. Legyen $U := U \cup \{u\} \cup \{v\}$
3. A G gráfból hagyjuk ki ezt a két csúcsot, és az összes rájuk illeszkedő élt.
4. Ha nem marad él, akkor kész vagyunk. U a lefogó csúcshalmaz. Különben lépjünk az 1. pontra.

Lássuk be, hogy ez az algoritmus 2-közelítő! Legyen F a kiválasztott élek halmaza: F diszjunkt élekből áll. Ekkor $|U_{opt}| \geq |F|$, mivel az optimális lefogó csúcshalmaz lefogja F éleit. Másrészt: $|U| = 2|F|$, tehát $|U_{opt}| \geq \frac{1}{2}|U|$.

A következő részben megmutatjuk, hogy az algoritmust továbbfejlesztve a minimális költségű lefogó csúcshalmaz feladatra is 2-közelítő algoritmust kapunk.

9.1.2. Minimális költségű lefogó csúcshalmaz

A fenti feladatot általánosítsuk úgy, hogy nemnegatív súlyokat adunk a csúcsoknak:

> *Minimális költségű lefogó csúcshalmaz probléma*
>
> Adott egy $G = (V, E)$ irányítatlan gráf és egy $c \in \mathbb{R}_+^V$ költségfüggvény. Keressünk olyan U lefogó csúcshalmazt, amire $\sum_{u \in U} c_u$ minimális.

Írjuk fel a feladatot egészértékű programozási feladatként:

$$\begin{aligned} x &\in \{0,1\}^V \\ x_u + x_v &\geq 1 \quad \forall uv \in E \\ \min\ & cx \end{aligned}$$

A feladat LP-relaxáltja:

$$\begin{aligned} x &\in \mathbb{R}_+^V \\ x_u + x_v &\geq 1 \quad \forall uv \in E \\ \min\ & cx \end{aligned}$$

Nézzük az LP-relaxált duálisát:

$$\begin{aligned} y &\in \mathbb{R}^E \\ y &\geq 0 \\ \sum_{u:uv \in E} y_{uv} &\leq c_v \quad \forall v \in V \\ \max\ & \sum_{e \in E} y_e \end{aligned}$$

A 2-közelítő algoritmusunk egy úgynevezett primál-duál algoritmus lesz: a futás során végig lesz egy csúcshalmazunk (ami a végén lefogóvá válik), és az LP-relaxáltnak egy duális megoldása (ami a végén bizonyítja, hogy 2-közelítést kaptunk).

> *2-közelítő primál-duál algoritmus*
>
> Minden lépésben lesz egy $U \subseteq V$ csúcshalmaz és lesz egy y duális megoldás. Kezdetben legyen $U = \emptyset$, $y \equiv 0$, $E' = E$.
>
> 1. Válasszunk tetszőleges $uv \in E'$ élt.
> 2. Emeljük meg y_{uv}-t annyira, hogy u-nál vagy v-nél a duál egyenlőtlenség teljesüljön egyenlőséggel – előfordulhat, hogy az egyenlőség mindkettőre teljesül.
> 3. U-hoz vegyük hozzá u és v közül azt, amelyiknél egyenlőség van – ha mindkettőnél egyenlőség van, akkor mindkettőt vegyük hozzá. Töröljük E'-ből a lefogott éleket.
> 4. Ha $E' = \emptyset$, akkor kész vagyunk. U a lefogó csúcshalmaz. Különben lépjünk az 1. pontra.

Figyeljük meg, hogy y_{uv} emelése csak az u-hoz és v-hez tartozó duális egyenlőtlenségeknél számít, a többit nem módosítja.

9.2. állítás. *A fenti algoritmus 2-közelítő.*

Bizonyítás. Legyen $\pi \in \mathbb{R}_+^V$ segédváltozónk. Kezdetben legyen $\pi \equiv 0$. Amikor y_{uv}-t megemeljük δ-val, akkor $\pi(u)$-t és $\pi(v)$-t is megemeljük δ-val.

A végén $\sum_{v \in V} \pi(v) = 2 \sum_{e \in E} y_e$. Másrészt: amikor v bekerül U-ba, akkor $\pi(v) = c(v)$, és ez később sem változik. Tehát $\sum_{u \in U} c_u \le \sum_{v \in V} \pi(v)$, amiből

$$\sum_{u \in U} c_u \le 2 \sum_{e \in E} y_e \underbrace{\le}_{\text{gyenge dualitás}} \le 2OPT_{LP} \le 2OPT_{IP}$$

□

Igazából kicsit többet bizonyítottunk az állításnál: az LP-relaxált optimumának legfeljebb kétszerese a kapott megoldás értéke.

9.2. Goemans–Williamson-algoritmus

Ebben a fejezetben Goemans és Williamson 1995-ös algoritmusát ismertetjük, ami különféle minimális súlyú erdő problémákra ad 2-közelítő algoritmust. Speciális esetként adódik majd a Steiner-fa illetve az euklideszi párosítási probléma (bár utóbbira van polinomiális algoritmus is).

9.2.1. A feladat

Legyen V egy alaphalmaz. Nevezzünk egy $\mathcal{F}$ halmazrendszert szépnek, ha teljesíti a következő tulajdonságokat:

$$\emptyset \notin \mathcal{F}, \tag{9.1}$$

$$X \in \mathcal{F} \Rightarrow V - X \in \mathcal{F}, \tag{9.2}$$

$$X \cap Y = \emptyset, X \cup Y \in \mathcal{F} \Rightarrow \{X, Y\} \cap \mathcal{F} \neq \emptyset. \tag{9.3}$$

Legyen $G = (V, E)$ a teljes gráf V-n, és legyen $c : E \to \mathbb{R}_+$ egy költségfüggvény ami teljesíti a háromszög-egyenlőtlenséget. A feladat egy minimális súlyú élhalmazt keresni, ami minden $\mathcal{F}$-beli halmazra tartalmaz egy belépő élt. Könnyen látható, hogy az optimális megoldás egy erdő lesz, hiszen minden körből törölhetünk egy tetszőleges élt.

A (9.3) tulajdonság miatt fogalmazhatunk így is: olyan minimális súlyú erdőt keresünk, aminek nincs $\mathcal{F}$-beli komponense.

9.2.2. Alkalmazások

Irányítatlan hálózat tervezése

Ennél a feladatnál egy gráfban adottak $s_1, s_2, \ldots, s_k$ valamint $t_1, t_2, \ldots, t_k$ csúcsok, és egy költségfüggvény az éleken. Minimális költségű olyan részgráfot akarunk találni, ahol s_i és t_i ugyanabban a komponensben van minden i-re. Könnyen látható, hogy a háromszög-egyenlőtlenség teljesülését feltehetjük: az uv költsége legyen az u és v közötti utak minimális költsége.

Hogy fér ez bele a fenti feladatba? Álljon $\mathcal{F}$ azokból az X halmazokból, amikre létezik i, hogy $|\{s_i, t_i\} \cap X| = 1$. Ez a halmazrendszer szép lesz, mert ha két diszjunkt halmaz uniója s_i és t_i közül pontosan egyet tartalmaz, akkor a két halmaz közül valamelyik szintén rendelkezik ezzel a tulajdonsággal.

Ha $s_1 = s_2 = \cdots = s_k$ akkor a minimális költségű Steiner-fa feladatot kapjuk.

Az euklideszi párosítás probléma

Adott az euklideszi síkon pontok egy páros elemszámú véges V halmaza. Keressük meg a pontokon azt a teljes párosítást, amely mentén az éleket egyenes szakaszokkal behúzva a párosítás összhossza minimális. (Könnyen látható, hogy az optimális megoldás esetén nem metszik egymást ezek a szakaszok, továbbá az is világos, hogy mivel véges sok teljes párosítás van, ezért létezik minimum.)

Álljon az $\mathcal{F}$ halmazrendszer a páratlan elemszámú halmazokból. Ez szép, hiszen két páros halmaz diszjunkt uniója nem lehet páratlan. Erre a halmazrendszerre az optimális megoldás nem feltétlenül egy párosítás, hanem egy erdő aminek minden komponense páros elemszámú. Azonban megmutatjuk, hogy ha a költségekre teljesül a háromszög-egyenlőtlenség, akkor van egy legfeljebb ekkora költségű teljes párosítás is:

- Ha van csúcs amire legalább két levél illeszkedik, akkor azt a két csúcsot levágjuk és egy éllel összekötjük. A háromszög-egyenlőtlenség miatt a költség nem nőtt, de a komponensek számát növeltük.

- Ha nincs ilyen csúcs de nem párosításunk van, akkor van egy másodfokú csúcs amire levél illeszkedik. Töröljük a csúcsra illeszkedő másik élt. Így továbbra is minden komponens páros lesz, de nő a komponensek száma.

9.2.3. Az algoritmus

A feladatunk lineáris programozási relaxáltja a következő.

$$\min \sum_{e\in E} c_e x_e \tag{9.4}$$

$$x(\delta(Z)) \geq 1 \quad \forall Z \in \mathcal{F}, \tag{9.5}$$

$$0 \leq x_e \quad \forall e \in E. \tag{9.6}$$

A lineáris programozási duális a következő.

$$\max \sum_{Z\in\mathcal{F}} y_Z \tag{9.7}$$

$$\sum_{Z\in\mathcal{F}: uv\in\delta(Z)} y_Z \leq c_{uv} \quad \forall\ uv \in E, \tag{9.8}$$

$$y_Z \geq 0 \quad \forall\ Z \in \mathcal{F}. \tag{9.9}$$

A Goemans–Williamson-algoritmus megkonstruál egy *megengedett erdőt*, azaz egy olyan erdőt, amelynek nincs $\mathcal{F}$-beli komponense. Az algoritmus fenntart egy F erdőt, amely csak a leálláskor lesz megengedett. Továbbá fenntartja a (9.8)-(9.9) egy y duális megoldását is. Tetszőleges $e = uv \in E$ élre vezessük be a következő jelölést.

$$\bar{c}_e := c_e - \sum_{Z\in\mathcal{F}: uv\in\delta(Z)} y_Z.$$

Jelölje $\mathcal{C}_F$ az F erdő komponenseinek halmazát. Ennek egy $C \in \mathcal{C}_F$ elemére legyen $q(C)$ értéke 1, ha $C \in \mathcal{F}$, és legyen 0 egyébként. Az algoritmus elindulásakor $F = \emptyset$, $\mathcal{C}_F = \{\{v\} : v \in V\}$, továbbá $y \equiv 0$.

Goemans–Williamson-algoritmus

1. lépés. Ha F-nek nincs $\mathcal{F}$-beli komponense, akkor az F-be kerüléssel fordított sorrendben töröljük F-ből az összes olyan élt, amelyek törlése után sem keletkezik $\mathcal{F}$-beli komponens (ha egy élt töröltünk, előlről kezdjük nézni a még F-ben lévő éleket). Return F.

2. lépés. Legyen $e = uv$ az az él, melyre $u \in C_i \in \mathcal{C}_F$, $v \in C_j \in \mathcal{C}_F$, $C_i \neq C_j$ és a következő érték minimális:

$$\varepsilon = \bar{c}_e / (q(C_i) + q(C_j)).$$

   (Vegyük észre, hogy C_i és C_j legalább egyike $\mathcal{F}$-beli.)

3. lépés. Minden $C \in \mathcal{C}_F \cap \mathcal{F}$ halmazra y_C-t növeljük meg ε-nal.

4. lépés. Az e élet vegyük F-hez, $\mathcal{C}_F$-ből töröljük C_i-t és C_j-t, és vegyük hozzá $C_i \cup C_j$-t. Ugorjunk az 1. lépésre.

Az algoritmus leállásakor tehát kapunk egy F^* feszítő erdőt és egy y^* duális megoldást. A 2. lépésben történő minimálizálás miatt y^* a (9.8) megengedett megoldása, azaz a redukált $\bar{c}_e$ súlyok mind nemnegatívak. Következésképpen a $\sum_{Z \in \mathcal{F}} y_Z$ érték az optimális megengedett erdő költségének egy alsó korlátját adja.

9.3. lemma. $c(F^*) \leq 2 \sum_{Z \in \mathcal{F}} y_Z$.

Bizonyítás. Jelölje $\mathcal{C}^k$ és ε^k a k-adik iterációban szereplő $\mathcal{C}_F$ partíciót és ε-t. Az F^* egy $e = uv$ ($u \in C_i \in \mathcal{C}^k$, $v \in C_j \in \mathcal{C}^k$) élére definiáljuk a c_e^k értéket 0-nak, ha $i = j$, és $\varepsilon^k \cdot (q(C_i) + q(C_j))$-nak különben.

Ekkor

$$c_e = c_e^1 + c_e^2 + \ldots + c_e^t,$$

ha t iteráció alatt fut le az algoritmus. Vegyük észre, hogy ha e-t az F-hez az s-edik lépésben vesszük hozzá, akkor $c_e^k = 0$ minden $k > s$ értékre. Most megmutatjuk, hogy

$$\sum_{e \in F^*} c_e^k \leq 2\varepsilon^k |\mathcal{C}^k \cap \mathcal{F}|,$$

amiből a lemma azonnal következik, mivel

$$\sum_{Z \in \mathcal{F}} y_Z = \sum_{k=1}^{t} \varepsilon^k \cdot |\mathcal{C}^k \cap \mathcal{F}|.$$

Legyen $k \in \{1, 2, \ldots, t\}$, és tekintsük az F^*-ból a $C \in \mathcal{C}^k$ halmazok egy-egy pontra való összehúzásával keletkező F' erdőt (ez azért lesz erdő, mert egy $C \in \mathcal{C}^k$ halmazon belül kiválasztott élek összefüggő gráfot alkotnak). Jelölje V_1 az $\mathcal{F}$-beli halmazok összehúzásaival keletkezett pontok halmazát, V_2 pedig a többi pontot.

9.4. állítás. $\sum_{v \in V_1} d_{F'}(v) \leq 2|V_1|$.

Bizonyítás. Mivel F^*-ból fordított sorrendben elhagytuk a felesleges éleket, ezért F' levelei mind V_1-ben vannak, azaz V_2 pontjai legalább másodfokúak: $\sum_{v \in V_1} d_{F'}(v) = \sum_{v \in V} d_{F'}(v) - \sum_{v \in V_2} d_{F'}(v) \leq 2|V| - 2|V_2| = 2|V_1|$. □

Figyelembe véve, hogy $\sum_{e \in F^*} c_e^k = \varepsilon^k \sum_{v \in V_1} d_{F'}(v)$, a lemmát bebizonyítottuk. □

9.3. Jain iteratív kerekítő algoritmusa

A 9.2. fejezetben szerepelt egy irányítatlan hálózattervezési feladat, amire a Goemans-Williamson algoritmus 2-közelítést adott. Most ennek a feladatnak egy általánosítására írjuk le Jain 2001-ből származó 2-közelítő algoritmusát. Az algoritmus fő ötlete, hogy az LP relaxációnak egy lépésben csak bizonyos komponenseit kerekíti egészre, a többi komponensre újra megoldja az immár kevesebb változós LP feladatot, és ezt iterálja. Innen ered az *iteratív kerekítés* elnevezés.

9.3.1. Az általános irányítatlan hálózattervezési feladat

A feladatban adott egy $G = (V, E \cup E_0)$ irányítatlan gráf, egy $c : E \to \mathbb{R}_+$ nemnegatív költségfüggvény, és minden $u, v \in V$ csúcspárra egy $r(u, v) \in \mathbb{Z}_+$ élösszefüggőségi igény. Erről feltesszük, hogy szimmetrikus, azaz $r(u, v) = r(v, u)$, és $r(v, v) = 0$.

A cél minimális költségű olyan $F \subseteq E$ élhalmazt találni, amire $F \cup E_0$-ban tetszőleges $u, v \in V$ csúcspárra van u és v közt $r(u, v)$ éldiszjunkt út. Természetesen lehet, hogy a feladatnak nincs megoldása, de ezt ellenőrizni tudjuk a G gráf Gomory-Hu fájának kiszámolásával.

Számos jól ismert feladat megfogalmazható speciális esetként:

- Minimális költségű feszítő fa: $r(u, v) = 1$ ha $u \neq v$.
- Minimális költségű k-folyam s-ből t-be szimmetrikus irányított gráfban: $r(s, t) = k$, a többi csúcspárra pedig $r(u, v) = 0$.
- Minimális költségű Steiner fa: $r(u, v) = 1$ ha u és v különböző terminálok, egyébként $r(u, v) = 0$.
- Minimális költségű, T-ben k-élösszefüggő részgráf (ahol $T \subseteq V$): $r(u, v) = k$ ha $u, v \in T$ és $u \neq v$, egyébként $r(u, v) = 0$.
- Hamilton-kör létezése a $G = (V, E)$ gráfban: $r(u, v){=}2$ ha $u \neq v$, $c(e) = 1$ minden $e \in E$-re. Pontosan akkor létezik Hamilton-kör, ha az optimális F-re $c(F) = |V|$.

Ezekből látszik, hogy a feladat NP-teljes, és jelenleg nem is ismert rá 2-nél jobb közelítő algoritmus. Az egészértékű programozási feladatként való felíráshoz vezessük be a következő halmazfüggvényt:

$$R(U) = \max_{u \in U, v \notin U} r(u, v) \qquad (U \subseteq V).$$

Esetszétválasztással ellenőrizhető, hogy ez a halmazfüggvény *ferdén szupermoduláris*, azaz bármely $X \subseteq V$ és $Y \subseteq V$-re az alábbi két egyenlőtlenség legalább egyike teljesül:

$$R(X) + R(Y) \leq R(X \cap Y) + R(X \cup Y),$$
$$R(X) + R(Y) \leq R(X \setminus Y) + R(Y \setminus X).$$

Sőt, az is könnyen látható, hogy az $R(U) - d_{E_0}(U)$ halmazfüggvény is ferdén szupermoduláris, hiszen a $-d_{E_0}(U)$ halmazfüggvényre mindkét fenti egyenlőtlenség teljesül minden X, Y párra.

Az általános irányítatlan hálózattervezési feladat IP felírása:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & \text{(IP)} \\ x_e \in \{0, 1\} & \qquad \forall e \in E & \\ d_x(U) \geq R(U) - d_{E_0}(U) & \qquad \forall U \subseteq V. & \end{aligned}$$

A feladat LP relaxáltja:

$$\min \sum_{e\in E} c_e x_e \tag{LP}$$
$$0 \le x_e \le 1 \qquad \forall e \in E$$
$$d_x(U) \ge R(U) - d_{E_0}(U) \qquad \forall U \subseteq V.$$

Megjegyzendő, hogy ennek az LP relaxáltnak már a minimális költségű feszítő fa feladatnál sem lesz feltétlenül egész optimális megoldása, például C_5-re, ellentétben a 11.2. fejezetben ismertetett relaxációval. Ráadásul exponenciális sok feltétel van, tehát az (LP) feladatot sem könnyű megoldani. Azonban szeparálni tudunk (minden u, v csúcspárra megkeressük a minimális súlyú őket elválasztó vágást), tehát az 3.10. fejezetben leirtak szerint optimalizálni is tudunk polinom időben. A gyakorlatban elég jó algoritmusok vannak az (LP) feladat megoldására.

Jain módszerének alapja a következő mély eredmény.

9.5. tétel (Kamal Jain, 2001)**.** *Ha az $x \equiv 0$ nem megoldása (LP)-nek, akkor minden bázismegoldásnak létezik legalább $\frac{1}{2}$ értékű komponense.*

Bizonyítás. (Ravi, Singh, Nagarajan, 2007). Indirekt tegyük fel, hogy van egy olyan x^* bázismegoldás, amire $x^*_e < \frac{1}{2}$ minden $e \in E$-re. Legyen $E^* = \{e \in E : x^*_e > 0\}$. Nevezzük az $\{U_1, U_2, \dots, U_k\}$ halmazrendszert (ahol $U_i \subseteq V$) *függetlennek*, ha a $\delta_{E^*}(U_1), \dots, \delta_{E^*}(U_k)$ élhalmazok karakterisztikus vektorai lineárisan függetlenek. Mivel x^* bázismegoldás és $x^*_e < 1$ minden $e \in E$ élre, tudjuk, hogy $k = |E^*|$-ra létezik olyan $\{U_1, U_2, \dots, U_k\}$ független halmazrendszer, amire

$$d_{x^*}(U_i) = R(U_i) - d_{E_0}(U_i) \qquad (i = 1, \dots, k).$$

9.6. állítás. *Ez az $\{U_1, U_2, \dots, U_k\}$ független halmazrendszer választható laminárisnak.*

Bizonyítás. Egy U halamazt nevezzünk *szorosnak*, ha $d_{x^*}(U) = R(U) - d_{E_0}(U)$. Legyen $\{U_1, \dots, U_l\}$ egy maximális elemszámú független lamináris halmazrendszer, ami szoros halmazokból áll. Ha $l < k$, akkor van olyan X szoros halmaz, hogy még az $\{U_1, \dots, U_l, X\}$ halmazrendszer is független. Válasszunk olyan X-et, ami a lehető legkevesebb U_i-t metszi (itt a metszés azt jelenti, hogy az $X \cap U_i, X \setminus U_i, U_i \setminus X$ halmazok egyike sem üres).

Mivel maximális elemszámú lamináris rendszert válaszottunk, X legalább egy U_j halmazt metsz. Az $R(U) - d_{E_0}(U)$ halmazfüggvény ferde szupermodularitása miatt a következő két lehetőség legalább egyike teljesül.

1. Az $X \cap U_j$ és $X \cup U_j$ halmazok szorosak, és $\delta_{E^*}(X) + \delta_{E^*}(U_j) = \delta_{E^*}(X \cap U_j) + \delta_{E^*}(X \cup U_j)$ (ahol az összeadást mint a karakterisztikus vektorok összeadását értjük).

2. Az $X \setminus U_j$ és $U_j \setminus X$ halmazok pontosak, és $\delta_{E^*}(X) + \delta_{E^*}(U_j) = \delta_{E^*}(X \setminus U_j) + \delta_{E^*}(U_j \setminus X)$.

Az első esetben a laminaritás miatt $X \cap U_j$ is és $X \cup U_j$ is kevesebb U_i halmazt metsz mint X. Viszont a lineáris függetlenség tulajdonságai miatt vagy $\{U_1, \dots, U_l, X \cap U_j\}$, vagy $\{U_1, \dots, U_l, X \cup U_j\}$ független, ellentmondásban X váltasztásával.

A második esetben $X \setminus U_j$ is és $U_j \setminus X$ is kevesebb U_i halmazt metsz mint X. Viszont a lineáris függetlenség tulajdonságai miatt vagy $\{U_1, \dots, U_l, X \setminus U_j\}$, vagy $\{U_1, \dots, U_l, U_j \setminus X\}$ független, ellentmondásban X váltasztásával. □

Legyen $\{U_1, U_2, \dots, U_k\}$ az állításban szereplő lamináris halmazrendszer. Ellentmondásra fogunk jutni azzal, hogy $0 < x^*_e < 1/2$ minden $e \in E^*$-ra. Ehhez a Jain által bevezetett „zsetonos” módszer egy változatát használjuk.

Minden E^*-beli él kap egységnyi zsetont. Ebből első lépésben az $uv \in E^*$ él megtart $1 - 2x^*_{uv}$-t, és továbbad az u és v csúcsoknak x^*_{uv}-t.

A második lépésben a zsetonok az U_i halmazokhoz kerülnek át, a következő szabály szerint:

- Egy $uv \in E^*$ él zsetonja a legkisebb olyan U_i halmazhoz kerül, ami u-t és v-t is tartalmazza.
- Egy $v \in V$ csúcs zsetonja a legkisebb olyan U_i halmazhoz kerül, ami v-t tartalmazza.

Nézzük meg, hány zsetont kap egy adott U_j halmaz. Ehhez definiáljunk három diszjunkt élhalmazt E^*-ban:

- E_1 áll azokból az $uv \in E^*$ élekből, amikre u zsetonját U_j kapta és $v \notin U_j$,
- E_2 áll azokból az $uv \in E^*$ élekből, amikre u és uv zsetonját U_j kapta, de v zsetonját nem,
- E_3 áll azokból az $uv \in E^*$ élekből, amikre uv zsetonját U_j kapta, de u és v zsetonját nem.

Az U_j halmaz legalább $x^*(E_1)+|E_2|-x^*(E_2)+|E_3|-2x^*(E_3)$ zsetont gyűjtött be. Ez pozitív, mert $E_1 = E_2 = E_3 = \emptyset$ esetén

$$\delta_{E^*}(U_j) = \sum_{i=1}^{l} \delta_{E^*}(U_{j_i}),$$

ahol $U_{j_1}, \ldots, U_{j_l}$ az U_j-n belüli maximális halmazok a lamináris rendszerben. Ez viszont azt jelentené, hogy a rendszer nem független. Másrészt

$$d_{x^*}(U_j) - \sum_{i=1}^{l} d_{x^*}(U_{j_i}) = x^*(E_1) - x^*(E_2) - 2x^*(E_3).$$

Itt a baloldal egész szám az U_i halmazok szorossága miatt, tehát a jobboldal is egész. Következésképp $x^*(E_1) + |E_2| - x^*(E_2) + |E_3| - 2x^*(E_3)$ is egész, tehát legalább 1.

A fenti okoskodás szerint minden U_i halmaz legalább egységnyi zsetont kap. Azonban van $e \in E^*$ él, aminek zsetonját egyik halmaz sem kapja meg: vegyünk egy maximális halmazt a lamináris rendszerben, és legyen e egy ebből kilépő él (ilyen van, különben a halmaz nem szerepelhetne független rendszerben). Az e élhez tartozó $1 - 2x^*_e > 0$ zseton egyik halmazhoz sem kerül.

Mivel $|E^*|$ zseton van, minden halmaz legalább 1-et kap, és van amit semelyik sem kap meg, $k < |E^*|$, ellentmondásban a halmazrendszer választásával. □

A fenti tétel ismeretében Jain algoritmusa meglehetősen egyszerű és természetes. Adott tehát egy $G = (V, E + E_0)$ irányítatlan gráf, egy $c : E \to \mathbb{R}_+$ nemnegatív költségfüggvény, és minden $u, v \in V$ csúcspárra egy $r(u, v) \in \mathbb{Z}_+$ élösszefüggőségi igény.

Jain iteratív kerekítő algoritmusa

Kezdetben legyen $F = \emptyset$.

1. lépés. Ha $R(U) \le |E_0 \cap \delta(U)|$ minden $U \subseteq V$-re: kész vagyunk, `return` F.
2. lépés. Oldjuk meg az (LP) feladatot, és legyen x^* optimális bázismegoldás.
3. lépés. Legyen $H \subseteq E$ azoknak az e éleknek a halmaza, amikre $x^*_e \ge 1/2$. A tétel szerint $|H| \ge 1$.
4. lépés. Legyen $F := F \cup H$, $E_0 := E_0 \cup H$, $E := E \setminus H$. Ugorjunk az 1. lépésre.

9.7. tétel. *Jain iteratív kerekítő algoritmusa 2-közelítő algoritmus az általánosított irányítatlan hálózattervezési feladatra, sőt az LP-relaxált optimumának is legfeljebb kétszeresét adja.*

Bizonyítás. |E| szerinti indukcióval bizonyítunk. Az állítás nyilván igaz, ha $R(U) \leq |E_0 \cap \delta(U)|$ minden $U \subseteq V$-re, hiszen ekkor $F = \emptyset$ minimális költségű.

Tegyük fel, hogy nem ennél az esetnél vagyunk, és tekintsük az algoritmus első fázisát. Jelöljük E'-vel illetve E_0'-vel a negyedik lépés utáni élhalmazokat, és legyen (LP') a rájuk vonatkozó LP feladat (amit a második fázisban old meg az algoritmus). Figyeljük meg, hogy ha az x^* vektort megszorítjuk az E' élhalmazra akkor az (LP') feladat megengedett megoldását kapjuk, hiszen minden U-ra az $d_x(U) \geq R(U) - d_{E_0}(U)$ egyenlőtlenség jobboldala legalább annyival csökken, mint a baloldala. Legyen $F' = F \setminus H$. Az indukciós feltevés szerint

$$c(F') \leq 2 \sum_{e \in E'} c_e x_e^*.$$

Ebből, és abból hogy $x^*(e) \geq 1/2$ minden $e \in H$-ra, azt kapjuk, hogy

$$c(F) = c(F') + c(H) \leq 2 \sum_{e \in E'} c_e x_e^* + 2 \sum_{e \in H} c_e x_e^* = 2 \sum_{e \in E} c_e x_e^* = 2\mathrm{OPT}_{LP} \leq 2\mathrm{OPT}_{IP}.$$

□

9.4. Fokszámkorlátos feszítő fák

Ebben a fejezetben Lau és Singh iteratív relaxációs algoritmusát mutatjuk be a fokszámkorlátos minimális költségű feszítőfa feladat közelítő megoldására. Itt a „közelítő” nem a célfüggvény-értékre vonatkozik, hanem arra, hogy a megoldás csak közelítőleg teljesíti a fokszámkorlátokat. Ennél többet nem várhatunk, hiszen például NP-teljes eldönteni, hogy van-e olyan feszítőfa, amiben minden fok legfeljebb 2.

Adott egy $G = (V, E)$ összefüggő egyszerű irányítatlan gráf, egy $c \in \mathbb{R}_+^E$ költségfüggvény, és egy $W \subseteq V$ csúcshalmaz, amin felső koráltok adottak a fokszámra: $b_w \in \mathbb{Z}_+$ $(w \in W)$.

Tekintsük a következő IP felírást:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & \text{(IP)} \\ x_e \in \{0,1\} & \qquad \forall e \in E \\ x(E[U]) \leq |U| - 1 & \qquad \forall U \subseteq V \\ x(E) = |V| - 1 & \\ d_x(w) \leq b_w & \qquad \forall w \in W. \end{aligned}$$

A feladat LP relaxáltja:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & \text{(LP)} \\ x_e \geq 0 & \qquad \forall e \in E \\ x(E[U]) \leq |U| - 1 & \qquad \forall U \subseteq V \\ x(E) = |V| - 1 & \\ d_x(w) \leq b_w & \qquad \forall w \in W. \end{aligned}$$

Legyen x^* az (LP) rendszer egy bázismegoldása, és $E^* = \{e \in E : x_e^* > 0\}\}$. Nevezzünk egy $U \subseteq V$ halmazt, illetve $w \in W$ csúcsot pontosnak, ha $x^*(E[U]) = |U| - 1$, illetve $d_{x^*}(w) = b_w$.

Az előző fejezethez hasonlóan belátjuk, hogy pontos halmazoknak és csúcsoknak létezik egy „szép” független rendszere. Mivel x^* bázismegoldás, léteznek $U_1, \dots, U_k$ pontos halmazok és $w_1, \dots, w_\ell$ pontos csúcsok úgy, hogy $k + \ell = |E^*|$, és a

$$\{\chi^{E^*[U_i]} : i \in [k]\} \cup \{\chi^{\delta_{E^*}(w_i)} : i \in [\ell]\}$$

vektorhalmaz lineárisan független. Válasszuk ezeket úgy, hogy k a lehető legnagyobb.

9.8. állítás. *Az $U_1, \dots, U_k$ halmazok választhatók laminárisnak.*

Bizonyítás. A bizonyítás hasonló a 9.6. állítás bizonyításához. Azt használjuk, hogy ha X és Y pontosak, akkor $X \cap Y$ és $X \cup Y$ is pontos, és E^*-ban nincs $X \setminus Y$ és $Y \setminus X$ között él, azaz $\chi^{E^*[X]} + \chi^{E^*[Y]} = \chi^{E^*[X \cap Y]} + \chi^{E^*[X \cup Y]}$. □

A továbbiakban feltesszük, hogy $U_1, \dots, U_k$ lamináris. Először az $\ell = 0$ esettel foglalkozunk.

9.9. állítás. *Ha $\ell = 0$, akkor x^* egy feszítő fa karakterisztikus vektora.*

Bizonyítás. Az $U_1, \dots, U_k$ halmazrendszerben nincsenek egyelemű halmazok, hiszen azokhoz a 0-vektor tartozna. Egy ilyen lamináris rendszernek legfeljebb $|V| - 1$ eleme van, másrészt $x^*(E) = |V| - 1$ és $x^*_e \le 1$ minden élre, tehát $|E^*| = |V| - 1 = k$, és $x^*_e = 1$ minden $e \in E^*$ élre. □

A következő állításnak az a meglepő következménye, hogy ha $\ell > 0$, akkor van olyan W-beli csúcs, amire olyan kevés él illeszkedik, hogy a fokszámkorlátot nem is tudjuk 1-nél többel megsérteni, ha csak E^* éleit használjuk. Legyen $W^* = \{w_1, \dots, w_\ell\}$.

9.10. állítás. *Ha $\ell > 0$, akkor van olyan $w \in W^*$ csúcs, amire $d_{E^*}(w) \le b_w + 1$.*

Bizonyítás. Az előző fejezethez hasonlóan itt is zsetonos érvelést használunk egy indirekt bizonyításban. Tegyük fel, hogy $\ell > 0$ és minden $w \in W^*$ csúcsra $d_{E^*}(w) \ge B_w + 2$. Minden $e \in E^*$ él kap egy zsetont, amiből x^*_e-nyit annak a legkisebb U_i halmaznak ad, ami mindkét végpontját tartalmazza, valamint $(1 - x^*_e)/2$-t ad minden W^*-beli végpontjának.

Először belátjuk, hogy minden U_i halmaz legalább 1 zsetont kap. Legyenek $U_{i_1}, \dots, U_{i_t}$ a tartalmazásra nézve maximális halmazok U_i-ben. A lineáris függetlenség miatt $E^*[U_i] \supsetneq \cup_{j=1}^t E^*[U_{i_j}]$, így $x^*(E^*[U_i]) > \sum_{j=1}^t x^*(E[U_{i_j}])$. A bal és jobb oldal is egész a pontosság miatt, így a különbség legalább 1, és U_i legalább ennyi zsetont kap.

Lássuk most be, hogy minden $w \in W^*$ csúcs is kap legalább 1 zsetont. A w által kapott zsetonok száma

$$\sum_{e \in \delta_{E^*}(w)} \frac{1 - x^*_e}{2} = \frac{d_{E^*(w) - b_w}}{2} \ge 1,$$

hiszen az indirekt feltevés szerint $d_{E^*}(w) \ge b_w + 2$.

Hogy ellentmondást kapjunk, belátjuk, hogy van él ami nem osztotta ki az összes zsetonját. A feltételekből következik, hogy E^* összefüggő, különben valamelyik komponensen sérülne a feltétel. Így ha egyik U_i sem egyezik meg V-vel, akkor a nem feszített élen marad zseton. Feltehetjük tehát, hogy $V = U_1$.

Ha $v \notin W^*$, akkor a v-re illeszkedő e éleken $x^*_e = 1$, különben maradna náluk zseton. Azt is feltehetjük, hogy az ilyen $e = uv$ élre a χ^e egységvektort generálják a $\chi^{U_1}, \dots, \chi^{U_k}$ vektorok, hiszen különben az $\{u, v\}$ pontos halmazzal eggyel nagyobb független halmazrendszert kapnánk, ellentmondva k maximális választásának. Felírhatjuk a következő azonosságot:

$$2\chi^{E^*[U_1]} = 2\chi^{E^*} = \sum_{w \in W^*} \chi^{\delta_{E^*}(w)} + \sum_{v \in V \setminus W^*} \chi^{\delta_{E^*}(v)} = \sum_{w \in W^*} \chi^{\delta_{E^*}(w)} + \sum_{v \in V \setminus W^*} \sum_{e \in \delta_{E^*}(v)} \chi^e,$$

azaz $\sum_{w \in W^*} \chi^{\delta_{E^*}(w)}$-t generálják a $\chi^{U_1}, \dots, \chi^{U_k}$ vektorok, ellentmondás. □

A két állítás alapján a következő egyszerű algoritmus talál egy olyan $F \subseteq E$ feszítő fát, aminek költsége legfeljebb $\mathrm{OPT}_{\mathrm{LP}}$, és $d_F(w) \le b_w + 1$ minden $w \in W$-re:

Lau és Singh iteratív relaxációs algoritmusa

Kezdetben legyen $F = \emptyset$.

1. lépés. Oldjuk meg az (LP) feladatot, és legyen x^* optimális bázismegoldás. Legyen $E^* = \{e \in E : x^*_e > 0\}$.
2. lépés. Töröljük E-ből $E \setminus E^*$ éleit. Ha van $w \in W$, amire $d_{E^*}(w) \le b_w + 1$, akkor töröljük w-t W-ből, és ugorjunk az 1. lépésre.
3. lépés. Ha nincs ilyen csúcs, akkor az állítások szerint E^* feszítő fa. Return $F := E^*$.

Az algoritmus során igazából nem kell minden lépésben optimális bázismegoldást találnunk, elég az első lépésben. A további lépésekben elég olyan bázismegoldást találni, aminek költsége nem nagyobb mint az előző lépés bázismegoldásának költsége (ami ebben a lépésben is megengedett megoldás, csak nem feltétlenül bázismegoldás a fokszámfeltételek törlése miatt).

9.5. Diszkrepancia

Tekintsünk egy véges V alaphalmazon egy $\mathcal{E}$ halmazrendszert, és legyen $n = |V|$, $m = |\mathcal{E}|$. A diszkrepancia feladatban V elemeit akarjuk két színnel színezni úgy, hogy az $\mathcal{E}$-beli halmazok minél egyenletesebbek legyenek, azaz a legnagyobb eltérés $\mathcal{E}$-beli halmazon a két színszám között a lehető legkisebb legyen. Formálisan,

$$\mathrm{disc}(\mathcal{E}) = \min_{x:V\to\{-1,1\}} \max_{U\in\mathcal{E}} |x(U)|.$$

Véletlen színezéssel azt kapjuk, hogy tetszőleges $\mathcal{E}$ diszkrepanciája $O((n \log m)^{1/2}$. Spencer és Gluskin egymástól függetlenül bizonyították a következő erősebb állítást:

9.11. tétel. *Ha $m \ge n$, akkor* $\mathrm{disc}(\mathcal{E}) = O((n \log(m/n))^{1/2})$. *Ha $m \le n$, akkor* $\mathrm{disc}(\mathcal{E}) = O(m^{1/2})$.

Spencer és Gluskin bizonyításai nem algoritmikusak, azaz nem számolnak ki polinom időben egy kis diszkrepanciájú színezést. Bansal volt az első, aki megmutatta, szemidefinit programozás segítségével, hogy a korlátot elérő színezés polinom időben kiszámolható. Ezután Lovett és Meka egy egyszerűbb, lineáris programozáson alapuló iteratív kerekítéses algoritmust adott. Újabb áttörés volt Rothvoss eredménye, aki megmutatta, hogy Gluskin tétele egyetlen lépésben is megkapható algoritmikusan egy nagyon egyszerű véletlen módszerrel.

Az alábbiakban Gluskin és Lovett-Meka eredményeit ismertetjük részletesen, Rothvoss algoritmusát csak vázoljuk. Mindegyik bizonyítás egy részleges színezési algoritmus iteratív alkalmazásán alapul: először egy részleges színezést kapunk, ami a csúcsok konstans részét színezi, és kellően kicsi a diszkrepanciája, majd a kiszínezett csúcsokat törölve rekurzívan meghívjuk a részleges színezési algoritmust a maradékra.

Gluskin bizonyítása

Gluskin bizonyítása a részleges színezés létezésére hasonló a Minkowski-tétel bizonyításához, de bonyolultabb mértéket használ. Egy $x \in \mathbb{R}^n$ pontra legyen $\gamma_n(x)$ az n-dimenziós sztenderd normális eloszlás sűrűségfüggvénye, azaz

$$\gamma_n(x) = (2\pi)^{-n/2} \exp(-\|x\|_2^2/2).$$

Ez egy mértéket definiál $\mathbb{R}^n$-en, amit Γ_n jelöl.

9.12. tétel (Gluskin)**.** *Kellően kis $\delta > 0$ konstansra igaz, hogy ha $S \subseteq \mathbb{R}^n$ egy konvex, 0-ra szimmetrikus mérhető halmaz, amire $\Gamma_n(S) \geq 2^{-\delta n}$, akkor van $z \in S \cap \{-1, 0, 1\}^n$, aminek legalább δn nemnulla komponense van.*

Bizonyítás. Nézzük meg, hogy a szimmetrikus $y, -y$ pontok $x \in \mathbb{R}^n$-nel való eltolása hogy változtatja meg a sürűségfüggvény szerinti értékük összegét:

$$\gamma_n(x+y) + \gamma_n(x-y) \geq 2(\gamma_n(x+y)\gamma_n(x-y))^{1/2} \geq 2\exp(-\|x\|_2^2/2)\gamma_n(y).$$

Jelölje S_x az S eltoltját x-szel. A fenti egyenlőtlenség alapján $\Gamma_n(S_x) \geq \exp(-\|x\|^2/2)\Gamma_n(S)$.

Tekintsük az $\{S_x : x \in \{-1, 1\}^n$ halmazokat. Ezek mértékenek összege legalább $2^n \exp(-n/2)\Gamma_n(S) = 2^{(1/2-\delta)n}$. Így létezik olyan $y \in \mathbb{R}^n$ pont, ami legalább $2^{(1/2-\delta)n}$ halmazban benne van ezek közül. Ha δ elég kicsi, akkor létezik $x, x' \in \{-1, 1\}^n$, amik legalább δn koordinátában különböznek, és $y \in S_x \cap S_{x'}$. Ekkor $y = v + x = v' + x'$ valamilyen $v, v' \in S$-re. Legyen $z = (x - x')/2 = (v' - v)/2$; ez $S \cap \{-1, 0, 1\}^n$-ben van a konvexitás és szimmetria miatt, és legalább δn nemnulla komponense van. □

Gluskin tételének felhasználásával a következőképpen kaphatunk algoritmust a 9.11. Tétel bizonyítására. Feltesszük, hogy $m \geq n$ (az $m \leq n$ eset könnyen visszavezethető az $m = n$ esetre). Az algoritmus fázisokból áll, minden fázisban az elemek egy részét színezzük ki -1-re vagy 1-re. Legyen V_k a k-adik fázis elején kiszínezetlen csúcsok halmaza ($V_1 = V$), és $n_k = |V_k|$. A k-adik fázisban az a célunk, hogy legalább δn_k elemet kiszínezzünk úgy, hogy a diszkrepancia V_k-n legfeljebb $\Delta_k = (n_k \log(2m/n_k))^{1/2}$. Ha ezt meg tudjuk csinálni, akkor $n_k \leq (1-\delta)^{k-1}n$, tehát $O(\log n)$ fázis van, és a teljes diszkrepancia legfeljebb $\Delta_1 + \Delta_2 + \cdots = O((n\log(m/n))^{1/2})$.

A k-adik fázisban a részleges színezést úgy kapjuk, hogy Gluskin tételét alkalmazzuk az

$$S_k = \{x \in [-1, 1]^{V_k} : |x(U \cap V_k)| \leq \Delta_k \ \forall U \in \mathcal{E}\}$$

poliéderre. Bizonyítás nélkül felhasználjuk, hogy létezik $\delta > 0$ konstans, hogy $\Gamma_{n_k}(S_k) \geq 2^{-\delta n_k}$. A 9.12 tételből következik, hogy van $z_k \in S_k \cap \{-1, 0, 1\}^{V_k}$, aminek legalább δn_k nemnulla komponense van. Ez megad egy jó részleges színezést V_k-n.

Lovett és Meka algoritmusa

A 9.11. Tétel fenti bizonyítása csak akkor ad polinomiális algoritmust, ha S_k-nak polinom időben meg tudjuk találni egy olyan $z_k \in S_k \cap \{-1, 0, 1\}^{V_k}$ pontját, aminek legalább δn_k nemnulla komponense van. Lovett és Meka algoritmusa ehelyett egy olyan $z_k \in S_k$ pontot talál z_{k-1}-ből kiindulva, aminek legalább δn_k komponense ± 1. Ez szintén elég, hiszen minden lépésben $(1-\delta)$-szorosára csökken a még nem rögzített koordináták száma.

Az algoritmus elején legyen $x_0 = z_{k-1}$, $X_0 = \{j \in [n] : |x_0(j) < 1\}$, $\mathcal{E}_0 = \{U \in \mathcal{E} : |x_0(U)| < \Delta_k\}$, és

$$H_0 = \{y \in \mathbb{R}_n : y^T e_j = 0 \text{ ha } j \notin X_0,\ y(U) = 0 \text{ ha } U \notin \mathcal{E}_0\}.$$

Legyen $\epsilon > 0$ egy kellően kicsi konstans. Az algoritmus a következő általános lépést ismétli $t = 1, 2 \ldots, \ell$-re, ahol $\ell \equiv \epsilon^{-2}$.

1. Legyen y_t egy normális eloszlás szerinti véletlen vektor H_{t-1}-ben. Legyen $x_t = x_{t-1}+\epsilon y_t$.
2. Legyen $X_t = \{j \in [n] : |x_t(j) < 1\}$, $\mathcal{E}_t = \{U \in \mathcal{E} : |x_t(U)| < \Delta_k\}$, és

$$H_t = \{y \in \mathbb{R}_n : y^T e_j = 0 \text{ ha } j \notin X_t,\ y(U) = 0 \text{ ha } U \notin \mathcal{E}_t\}.$$

Az algoritmus során a feltételek legfeljebb ϵ-nal sérülnek. Be lehet látni, hogy ha a konstansokat megfelelően választjuk, akkor legfeljebb $1/8$ annak a valószínűsége, hogy $|\mathcal{E}_0 \setminus \mathcal{E}_\ell| > n_k/8$. Mostantól feltesszük, hogy $|\mathcal{E}_0 \setminus \mathcal{E}_\ell| \leq n_k/8$. Ha $|X_0 \setminus X_\ell| > n_k/2$, akkor kész vagyunk, tehát tegyük fel, hogy $|X_0 \setminus X_\ell| \leq n_k/2$. Így H_ℓ dimenziója legalább $n_k - n_k/2 - n_k/8 = 3n_k/8$. Be lehet látni, hogy ekkor $E(\|x_t - x_{t-1}\|_2^2) = \epsilon^2 E(\|y_t\|_2^2) \geq (3n_k/8)\epsilon^2$ minden $t \in [\ell]$ esetén, és ebből $\ell \equiv \epsilon^{-2}$ alapján azt kapjuk a normális eloszlás tulajdonságai alapján, hogy $E(\|x_\ell\|_2^2 - \|x_0\|_2^2) = \Omega(n_k)$. De minden j-re $x_\ell(j) - x_0(j) \in [-1, 1]$, tehát legalább $\Omega(n_k)$ változó várhatóan eléri a ± 1-et.

Rothvoss algoritmusa

Térjünk vissza Gluskin tételéhez, miszerinint ha $S \subseteq \mathbb{R}^n$ egy konvex, 0-ra szimmetrikus mérhető halmaz, amire $\Gamma_n(S) \geq 2^{-\delta n}$ egy elég kis $\delta > 0$ konstansra, akkor van $z \in S \cap \{-1, 0, 1\}^n$, aminek legalább δn nemnulla komponense van. Rothvoss bebizonyította, hogy a következő nagyon egyszerű algoritmus nagy valószínűséggel talál egy $z \in S \cap [-1, 1]^n$ pontot, aminek legalább δn ± 1-es komponense van:

9.13. tétel. *Legyen $v \in \mathbb{R}^n$ egy normális eloszlás (azaz a γ_n súrúségfüggvény) szerint véletlenül választott vektor, és legyen z az $S \cap [-1, 1]^n$-nek a v-hez legközelebbi pontja. Ekkor z-nek nagy valószínűséggel legalább δn ± 1-es komponense van.*

A bizonyítás azon az egyszerű tényen alapul, hogy ha az $S \cap [-1, 1]^n$-nek a v-hez legközelebbi z pontja a $-1 \leq z_j \leq 1$ feltételek valamelyikét nem teljesíti egyenlőséggel, akkor azt a feltételt elhagyva is z lesz a legközelebbi pont. Ha indirekt feltesszük, hogy z sok feltételt nem teljesít egyenlőséggel, akkor ez alapján felső korlát adható z várható távolságára v-től. Másrészt alsó korlát adható a távolság várható értékére az eredeti $S \cap [-1, 1]^n$ konvex halmazt haszánálva, és ez a két korlát ellentmond ellentmond egymásnak.

10. fejezet

Általános heurisztikus algoritmusok

10.1. Lokális keresés

10.2. Tabukeresés

10.3. Szimulált lehűlés

10.4. Genetikus algoritmusok

11. fejezet

Függelék

11.1. Lineáris programozási emlékeztető

Ebben a fejezetben összefoglaljuk azokat a lineáris programozási ismereteket, amelyek a jegyzetben felhasználásra kerülnek (megesik, hogy külön hivatkozás nélkül is).

11.1. definíció. A $C \subseteq \mathbb{R}^n$ halmaz

- **konvex kúp**, ha tetszőleges $x, y \in C, \lambda, \mu \geq 0$ esetén $\lambda x + \mu y \in C$.
- **poliéderkúp**, ha létezik olyan A mátrix, melyre $C = \{x \in \mathbb{R}^n : Ax \leq 0\}$.
- **végesen generált kúp**, ha léteznek az $x_1, x_2, \ldots, x_k$ $\mathbb{R}^n$-beli vektorok, amelyekre $C = \{x \in \mathbb{R}^n : x = \sum_{i=1}^n \lambda_i x_i, \lambda_i \geq 0\}$.

Szavakban azt mondhatjuk, hogy a poliéderkúpok véges sok origón átmenő féltér metszeteként állnak elő, a végesen generált kúpok pedig véges sok vektor összes nemnegatív együtthatós lineáris kombinációjaként előálló halmazok.

11.2. tétel (Weyl, 1935). *A $C \subseteq \mathbb{R}^n$ halmaz pontosan akkor poliéderkúp, ha végesen generált kúp.*

Az A mátrix elemei pontosan akkor választhatók racionális számoknak, ha az x_i vektorok is választhatók racionálisnak.

11.3. definíció. A $P \subseteq \mathbb{R}^n$ halmaz

- **poliéder**, ha létezik olyan A mátrix és megfelelő méretű b vektor, melyekre $P = \{x \in \mathbb{R}^n : Ax \leq b\}$.
- **politóp**, ha léteznek az $x_1, x_2, \ldots, x_k$ $\mathbb{R}^n$-beli vektorok, amelyekre $P = \{x \in \mathbb{R}^n : x = \sum_{i=1}^n \lambda_i x_i,\ \lambda_i \geq 0,\ \sum_{i=1}^n \lambda_i = 1\}$.

Szavakban azt mondhatjuk, hogy a poliéder véges sok féltér metszeteként előálló ponthalmaz (speciálisan az egész tér definíció szerint nulla darab féltér metszeteként áll elő, tehát poliéder), míg a politóp véges sok pont konvex burka.

11.4. tétel (Motzkin, 1936). *A $P \subseteq \mathbb{R}^n$ halmaz pontosan akkor poliéder, ha előáll $Q + C$ alakban, ahol Q egy politóp, C pedig egy végesen generált kúp.*

C a P poliéder **karakterisztikus kúpja**. A tétel egyszerű következménye, hogy a politópok éppen a korlátos poliéderek (ennek a nehéz iránya az, hogy a korlátos poliéderek véges sok pont konvex burkaként is megkaphatók, bár a szemléletünk szerint ez nagyon is elvárható).

A következő tétel (sőt, lemma, a szó nemes értelmében) az előző két eredmény bizonyításakor is alapvető fontosságú.

11.5. tétel (Farkas-lemma (első alak)). *Az $Ax \leq b$ poliéder pontosan akkor üres, ha létezik y megoldása az alábbinak*

$$yA = 0, y \geq 0, yb < 0.$$

11.6. tétel (Farkas-lemma (második alak)). *Az $Ax = b, x \geq 0$ poliéder pontosan akkor üres, ha létezik y megoldása az alábbinak*

$$yA \geq 0,\ yb < 0.$$

A Farkas-lemma most következő erősítését a szimplex-módszer bizonyításából kaphatjuk.

11.7. tétel. *Ha adott az A mátrix és a b vektor, akkor az alábbiak közül pontosan az egyik áll fenn. Az $[A, b]$ mátrix rangját jelölje t.*

- *Kiválasztható az A oszlopai közül néhány lineárisan független, amelyek nemnegatív lineáris kombinációjaként megkapható b.*
- *Létezik egy $\{x \in \mathbb{R}^n : cx = 0\}$ hipersík, amely A-nak $t-1$ darab lineárisan független oszlopát tartalmazza, és $cb < 0$, $cA \geq 0$.*

11.8. tétel (kiegészítő eltérések tétele). *Legyen x^* és y^* olyan, amelyekre $Ax^* \leq b, x^* \geq 0$, $y^*A \leq c, y^* \geq 0$, és*

$$x_j^* > 0 \implies (y^*Ax)_j = c_j,$$

$$y_i^* > 0 \implies (Ax^*)_i = b_i.$$

Ekkor x^* a primál $\max\{cx : Ax \leq b, x \geq 0\}$, y^* a duális $\min\{by : yA \geq c, y \geq 0\}$ feladatok optimális megoldása.

11.9. tétel (dualitástétel). *Amennyiben a következő egyenlőtlenségeknek van megoldása, akkor teljesül az alábbi egyenlőség:*

$$\max\{cx : Ax \leq b\} = \min\{by : yA = c, y \geq 0\}. \tag{11.1}$$

11.1.1. A lexikografikus duál-szimplex módszer

11.2. A feszítőfák poliédere

Adott az összefüggő $G = (V, E)$ irányítatlan gráf, és az élein a $c : E \to \mathbb{R}_+$ nemnegatív súlyfüggvény. Közismert, hogy a Kruskaltól származó mohó algoritmus megadja a G egy minimális súlyú feszítőfáját. Emlékeztetőül az algoritmus a következő. Rendezzük súly szerint növekvő sorrendbe az éleket: $e_1, e_2, \ldots, e_m$, majd haladjunk végig ebben a sorrendben, és egy élet válasszunk ki akkor, ha a már korábban kiválasztott élekhez hozzávéve nem keletkezik kör. Könnyű belátni, hogy így valóban minimális súlyú feszítőfát kapunk, bár ennek ténye következi fog a bizonyítandó tételünkből is.

Vegyük észre, hogy a célfüggvényről elfeletkezve az alábbi feladat egész, azaz 0-1-megoldásai a feszítőfák karakterisztikus vektorai.

$$\min \sum (c_e x_e : e \in E) \tag{11.2}$$

$$x(E) = |V| - 1 \tag{11.3}$$

$$x(\gamma(S)) \leq |S| - 1 \quad \forall S \subset V, S \neq \emptyset \tag{11.4}$$

$$0 \leq x_e \quad \forall e \in E. \tag{11.5}$$

11.10. tétel. *A fenti lineáris programnak tetszőleges célfüggvény esetén létezik egészértékű optimális megoldása.*

Bizonyítás. Jelölje $\kappa(A)$ a (V, A) gráf komponenseinek a számát tetszőleges A élhalmaz esetén. Vegyük észre, hogy az alábbi lineáris program ekvivalens a fentivel.

$$\min \sum (c_e x_e : e \in E) \tag{11.6}$$

$$x(E) = |V| - 1 \tag{11.7}$$

$$x(A) \leq |V| - \kappa(A) \quad \forall A \subset E \tag{11.8}$$

$$0 \leq x_e \quad \forall e \in E. \tag{11.9}$$

Most megmutatjuk, hogy a Kruskal algoritmusa által adott feszítőfa x^* karakterisztikus vektora optimális megoldása (11.6)-nek (amiből az is következik, hogy valóban minimális súlyú feszítőfát szolgáltat). Ezt úgy látjuk be, hogy a Kruskal algoritmus lefutása közben megadjuk a duális feladat egy olyan y^* megoldását, amely x^*-gal teljesíti a kiegészítő eltéréseket (11.8. tétel), azaz mindkettő optimális megoldás. A primál feladatban az egyszerűség kedvéért cseréljük le a célfüggvényt az ekvivalens $\max \sum (-c_e x_e : e \in E)$ alakra.

A duális feladat a következő:

$$\min \sum (((|V| - \kappa(A)) y_A : A \subseteq E) \tag{11.10}$$

$$\sum_{e \in A} y_A \geq -c_e \quad \forall e \in E \tag{11.11}$$

$$0 \leq y_A \quad \forall A \subset E. \tag{11.12}$$

Az y_E változónak nem kell nemnegatívnak lennie. Legyen $e_1, e_2, \ldots, e_m$ az élek azon sorrendje, amely szerint Kruskal algoritmusa végignézi őket. Jelölje R_i ezen sorrend első i élének a halmazát. Most megadjuk y^* definícióját.

$$y^*_A := \begin{cases} c_{e_{i+1}} - c_{e_i}, \text{ ha } A = R_i, i < m, \\ -c_{e_m}, \text{ ha } A = R_m, \\ 0 \text{ különben.} \end{cases}$$

Mivel az élek a súlyok szerinti növekvő sorrendben vannak, $y^*_A \geq 0$ minden $A \subset E$ esetén. Most megmutatjuk, hogy (11.11) is teljesül. Legyen $e = e_i$, ekkor

$$\sum_{A \ni e} y^*_A = \sum_{j=i}^{m} y^*_{R_j} = \sum_{j=i}^{m-1} (c_{e_{j+1}} - c_{e_j}) - c_{e_m} = -c_{e_i} = -c_e.$$

Tehát y^* valóban megoldása a duális feladatnak, amely ráadásul minden (11.11) feltételt egyenlőséggel teljesít.

Most belátjuk, hogy amennyiben $y^*_A > 0$, akkor az A-hoz tartozó (11.8) feltétel egyenlőséggel teljesül. Ekkor $A = R_i$ valamely i értékre. i-re vonatkozó indukcióval az állítás triviális, hiszen amikor Kruskal algoritmusa kiválaszt egy élt (azaz $x^*_e = 1$), akkor mivel kör nem keletkezhet, két meglévő komponenst köt össze.

Teljesülnek tehát a kiegészítő eltérések tételének a feltételei, azaz beláttuk, amit akartunk. □

11.3. Prim feszítőfa algoritmusa

Az alábbiakban röviden ismertetjük Prim minimális súlyú feszítőfát megkereső algoritmusát, amely valamelyest különbözik Kruskal algoritmusától.

A feladat tehát a következő: Adott az összefüggő $G = (V, E)$ irányítatlan gráf, és az élein a $c : E \to \mathbb{R}_+$ nemnegatív súlyfüggvény. Keressünk meg egy minimális súlyú feszítőfát.

Prim algoritmusa a következő: válasszuk ki a G gráf egy minimális súlyú élét, majd ebből kiindulva a már meglévő F fánkhoz mindig vegyünk hozzá egy belőle kiinduló minimális súlyú élt, amíg feszítőfához nem jutunk.

11.11. tétel. *Prim algoritmusa minimális súlyú feszítőfát ad.*

Bizonyítás. A bizonyítást az Olvasóra bízzuk. □

11.4. Nagamochi és Ibaraki algoritmusa

Legyen $G = (V, E)$ egy gráf, és $c : E \to \mathbb{Z}_+$ egy kapacitásfüggvény az éleken. A csúcsoknak egy $v_1, v_2, \ldots, v_n$ sorrendjét *legális sorrendnek* nevezzük, ha

$$c(\delta(V_{i-1}) \cap \delta(v_i)) \geq c(\delta(V_{i-1}) \cap \delta(v_j))$$

minden $2 \leq i < j \leq n$-re, ahol $V_i = \{v_1, \ldots, v_i\}$. Egy legális sorrendet könnyű kiszámolni a pontok egyenkénti hozzávételével.

Egy u és v csúcsokat elválasztó vágást (u, v)-*vágásnak* nevezünk. Jelölje $\lambda(G; u, v)$ az (u, v)-vágások minimális c-értékét.

11.12. tétel (Nagamochi, Ibaraki, 1992)**.** *Ha $v_1, v_2, \ldots, v_n$ legális sorrend, akkor $\delta(v_n)$ egy minimális értékű (v_n, v_{n-1})-vágás G-ben.*

Bizonyítás.

Azt kell belátni, hogy $c(\delta(v_n)) \leq \lambda(G; v_{n-1}, v_n)$. Indukciót használunk a pontok és élek számának az összegére (ha ez 2, az állítás triviális). Először tegyük fel, hogy $v_{n-1}v_n \in E$. Legyen G' az $e = v_{n-1}v_n$ él törlésével keletkező gráf. Könnyű látni, hogy a sorrendünk legális marad, tehát indukció szerint

$$c(\delta(v_n)) = c(\delta'(v_n)) + c(e) = \lambda(G'; v_{n-1}, v_n) + c(e) = \lambda(G; v_{n-1}, v_n).$$

Tehát feltehetjük, hogy $v_{n-1}v_n \notin E$. Könnyen látható, hogy

$$\lambda(G; v_{n-1}, v_n) \geq \min\{\lambda(G; v_{n-2}, v_n), \lambda(G; v_{n-2}, v_{n-1})\}.$$

Tehát azt kell belátnunk, hogy $c(\delta(v_n)) \leq \lambda(G; v_{n-2}, v_n)$ és $c(\delta(v_n)) \leq \lambda(G; v_{n-2}, v_{n-1})$.

Az előbbi bizonyításához legyen G' a v_{n-1} törlésével keletkező gráf. A $v_1, \ldots, v_{n-2}, v_n$ sorrend legális marad, és az indukciós feltétel miatt

$$c(\delta(v_n)) = c(\delta'(v_n)) = \lambda(G'; v_{n-2}, v_n) = \lambda(G; v_{n-2}, v_n).$$

A másik egyenlőtlenség bizonyításához legyen G' a v_n törlésével keletkező gráf. A $v_1, \ldots, v_{n-1}$ sorrend legális, és az indukciós feltétel miatt most

$$c(\delta(v_n)) \leq c(\delta(v_{n-1})) = c(\delta'(v_{n-1})) = \lambda(G'; v_{n-2}, v_{n-1}) = \lambda(G; v_{n-2}, v_{n-1}).$$

□

A fenti tétel értelmében a következő algoritmus talál egy minimális értékű A vágást G-ben:

0. Legyen kezdetben M értéke ∞, és legyen A definiálatlan.
1. Ha G-nek 1 pontja van: vége
2. Keressünk egy $v_1, v_2, \ldots, v_n$ legális sorrendet.
3. Ha $c(\delta(v_n)) < M$, akkor legyen $M := c(\delta(v_n))$, és $A := \delta(v_n)$.
4. Húzzuk össze G-ben a v_{n-1} és v_n pontokat, és lépjünk 1-re.

11.5. Az egészértékű programozási feladat NP-teljes

Láttuk, hogy a Hamilton-kör létezése felírható egészértékű programozási feladatként. Bizonyítja-e ez, hogy az egészértékű programozási feladat NP-teljes? Nem, mivel az általunk mutatott egyenlőtlenség-rendszer nagyon sok feltételt tartalmazott a gráf méretéhez képest. (Megadható olyan felírás is, melyben csak polinomiálisan sok feltétel szerepel a gráf méretében. Hogyan?)

11.13. tétel. *NP-teljes.*

Bizonyítás. A bizonyítást az Olvasóra bízzuk. (Vezessük vissza 3-SAT-ot!) □

11.6. Minimális T-vágások

Ebben a fejezetben ismertetünk két, minimális T-vágás meghatározására szolgáló algoritmust. Az első algoritmus hatékonyabb, de a Gomory-Hu-fa tulajdonságaira épül; ezeket itt csak kimondjuk, nem bizonyítjuk. A második algoritmus egyszerűbb, csak minimális vágás keresést használ.

11.6.1. Minimális T-vágás Gomory-Hu fával

Adott egy $G = (V, E)$ gráf és egy $c : E \to \mathbb{R}_+$ súlyozás. Az ehhez tartozó Gomory-Hu fa egy V csúcshalmazú F fa (nem feltétlenül részgráfja G-nek!), ami a következő tulajdonsággal rendelkezik: tetszőleges u, v csúcspár esetén van olyan $f \in F$ él az egyértelmű $u - v$ úton F-ben, hogy az $F - f$ két komponense által meghatározott vágás G-ben egy minimális súlyú $u - v$ vágás. Gomory és Hu tétele azt mondja ki, hogy mindig létezik ilyen tulajdonságú fa, és $O(n^4)$ időben megtalálható. Ennek segítségével Padberg és Rao adott algoritmust minimális súlyú T-vágás keresésére.

11.14. tétel (Padberg,Rao)**.** *Tetszőleges T páros elemszámú csúcshalmaz esetén az F Gomory-Hu fának van olyan f éle, hogy az $F - f$ két komponense által meghatározott vágás G-ben egy minimális súlyú T-vágás.*

Bizonyítás. Legyen $G' = (V, E \cup F)$, és definiáljunk a $c' : E \cup F \to \mathbb{R}_+$ súlyozást úgy, hogy E-n megegyezik c-vel, F-en pedig 0. Rögzítsük a T csúcshalmazt. Világos, hogy G-ben és G'-ben ugyanazok a halmazok határoznak meg T-vágást, és a súlyuk is ugyanaz. Legyen $(X, V \setminus X)$ egy minimális súlyú T-vágás. Legyen $J \subseteq F$ azoknak az $f \in F$ éleknek a halmaza, amikre $F - f$ két komponense T-vágást határoz meg. Könnyű ellenőrizni, hogy J egy T-kötés G'-ben, azaz pontosan T pontjaiban páratlan a fokszáma. Mivel G'-ben egy T-kötésnek és egy T-vágásnak mindig páratlan sok közös éle van, létezik $uv \in \delta(X) \cap J$. Legyen $(Y, V \setminus Y)$ az $F - uv$ két komponense. Ekkor egyrészt $(Y, V \setminus Y)$ T-vágás, másrészt $c(\delta(Y)) \leq c(\delta(X))$, hiszen a Gomory-Hu fa tulajdonságai miatt $(Y, V \setminus Y)$ minimális súlyú $u - v$ vágás, és $(X, V \setminus X)$ is egy $u - v$ vágás. Azt kaptuk tehát, hogy $(Y, V \setminus Y)$ minimális súlyú T-vágás. □

A minimális súlyú T-vágás tehát megkereshető úgy, hogy végignézzük az F Gomory-Hu fa éleit és azt az $f \in F$ élt választjuk, amelyikre $F - f$ két komponense T-vágást határoz meg, és ezen belül a legkisebb súlyút.

11.6.2. Rekurzív minimális T-vágás keresés

Egyszerűsége miatt érdemes áttekintenünk az alábbi, Grötschel, Lovász és Schrijver szerzőhármastól származó algoritmust.

Legyen adott a $G = (V, E)$ gráf az élein a $c : E \to \mathbb{R}_+$ súlyozással. Adott továbbá a páros elemszámú $T \subseteq V$ részhalmaz. A $Q \subseteq V$ halmazt T-vágásnak nevezzük, ha $|T \cap Q|$ páratlan. Minimális T-vágásnak nevezzük a Q T-vágást, ha a $c(\delta(Q))$ értéke, azaz a Q-ból kilépő élek súlyainak az összege minimális.

Keressük meg a minimális értékű T-vágást! Jól tudjuk, hogy maximális folyamot hogyan kereshetünk meg hatékonyan egy gráfban, azt is tudjuk, hogy egyúttal egy minimális vágást is hogyan határozhatunk meg.

Az algoritmus először megkeres egy olyan minimális S vágást, amelyre $S \cap T \neq \emptyset$ és $(V - S) \cap T \neq \emptyset$. Ezt megtehetjük a rögzített $s \in T$ és minden $t \in T - s$ mellett a minimális s, t-vágások kiszámításával. Ez $|T| - 1$ darab maximális folyam algoritmussal megkapható.

Amennyiben S már T-vágás, megkaptuk a minimális T-vágást. Ha nem ez a helyzet, akkor a következő lemma segítségével lépünk tovább.

11.15. lemma. *Létezik egy Q minimális T-vágás, melyre $Q \subseteq S$ vagy $Q \subseteq V - S$ teljesül.*

Bizonyítás. Legyen $U \subseteq V$ egy minimális T-vágás, és tegyük fel hogy $U \cap S$, $U \cap (V - S)$, $(V - U) \cap S$, és $(V - U) \cap (V - S)$ mind nemüresek. Feltehetjük, hogy $U \cap S \cap T$ páratlan elemszámú, ekkor $(V - U) \cap S \cap T$ is páratlan elemszámú. Feltehetjük azt is, hogy $(S \cup U) \cap T \neq \emptyset$ és $(V - (S \cup U)) \cap T \neq \emptyset$. Belátjuk, hogy ekkor $S \cap U$ is minimális T-vágás, amiből a lemma állítása következik. S választása miatt

$$c(\delta(S)) \leq c(\delta(S \cup U)).$$

A fokszám-függvény pedik szubmoduláris (l. 17. feladat)

$$c(\delta(S)) + c(\delta(U)) \geq c(\delta(S \cup U)) + c(\delta(S \cap U)),$$

az utóbbit kivonva az előbbiből a következő adódik:

$$c(\delta(S \cap U)) \leq c(\delta(U)),$$

azaz $S \cap U$ valóban minimális T-vágás. □

Eszerint a feladat redukálható két kisebb gráfra. Származzon G_1 a $V - S$ egy pontra való összehúzásával, és az élsúlyok meghagyásával (a keletkező hurokélek nem játszanak szerepet). $T_1 := S \cap T$. Hasonlóképpen G_2 legyen az S halmaz egy pontra való összehúzásával kapott gráf, az élsúlyokat hagyjuk meg. T_2 pedig legyen $(V - S) \cap T$.

A G-beli minimális T-vágás tehát a G_1-beli minimális T_1-vágás és a G_2-beli minimális T_2-vágás közül a kisebb.

A kisebb gráfokban ugyanezzel az eljárással keressük meg a minimális T-vágásokat a megfelelő T halmazra vonatkozóan.

16. feladat. *Mutassuk meg, hogy a fenti algoritmus $O(n^5)$ futási idejű, ha van egy $O(n^3)$ futási idejű folyam-algoritmusunk (például az előreemelő előfolyam-algoritmus ilyen). (Segítség: Alkalmazzunk indukciót a $|T| = |T_1| + |T_2|$ megfigyelésre építve.)*

Ezzel tehát befejeztük az algoritmus leírását. Ezen algoritmus kevésbé hatékony, mint a Gomory-Hu-fára épülő eljárás, mindössze egyszerűsége miatt szerepeltettük itt.

17. feladat. *Legyen adott a $G = (V, E)$ gráf, és az élein a $c : E \to \mathbb{R}_+$ súlyozás. Bizonyítsuk be, hogy a fokszámfüggvény szubmoduláris (élsúlyozással is), azaz minden $X, Y \subseteq V$ ponthalmazra a következő teljesül:*

$$c(\delta(X)) + c(\delta(Y)) \geq c(\delta(X \cup Y)) + c(\delta(X \cap Y)).$$

12. fejezet

Jelölések

Ebben a fejezetben összefoglaljuk a jegyzetben használt legfontosabb jelöléseket.

1.

2. fejezet

   - cx jelöli a $c^T x$ szorzatot két azonos számú komponensből álló a és c vektor esetén, ahot T a transzponálás jele, azaz cx egy szám (egykoponensű vektor:-)
   - P jelöli valamely A mátrix és b vektor mellett az $Ax \leq b$ egyenletrendszer (valós) megoldásainak a halmazát, azaz egy poliédert.
   - P_I jelöli P poliéder egész pontjainak konvex burkát.
   - $\lfloor \lambda \rfloor$: az a legnagyobb egész, amely a λ valós számnál nem nagyobb, azaz a λ alsó egészrésze.
   - Megfelelő méretű B mátris és y vektor esetén yB jelöli az $y^T B$ mátrixszorzat eredményéül kapott vektort.
   - $\mathbb{1}$ jelöli a csupa 1 komponensű vektort.
   - $\text{conv}(X)$: az X ponthalmaz konvex burka

3.

4.

5.

6.

7. fejezet

   - $c_e := c(e)$, ahol $x : E \to \mathbb{R}$ egy függvény. További példa x_e.
   - $\delta(v)$: a v pontra illeszkedő élek *halmaza.*
   - $\delta(S)$: az S ponthalmazból kilépő élek *halmaza.*
   - $\gamma(S)$: az S ponthalmazbeli pontokat összekötő (azaz feszített) élek *halmaza.*
   - $x(A) := \sum_{e \in A} x(e)$, ahol $x : E \to \mathbb{R}$ függvény, $A \subseteq E$ tetszőleges részhalmaz. Példák: $x(\delta(S)), x(\gamma(S))$.

8.

9.

10.

11.

Irodalomjegyzék

[1] Cook, Cunningham, Pulleybank, Schrijver: *Combinatorial Optimization*, Wiley, 1998.

[2] A. Schrijver: *Theory of Linear and Integer Programming*, Wiley, 1986.

[3] A. Schrijver: *Combinatorial Optimization*, Springer, 2003.

[4] G. Nemhauser and L. Wolsey: *Integer and Combinatorial Optimization*, Wiley, 1988.

[5] Vizvári Béla: *Egészértékű programozás*, ELTE jegyzet, 1992.

[6] L. Wolsey: *Integer Programming*, Wiley, 1998.

[7] D. Bertsimas, J.N. Tsitsiklis: *Introduction to Linear Optimization*, Athena Scientific, 1997.

[8] Frank András: *Operációkutatás II. éves matematikus hallgatóknak*, előadásjegyzet, 2003.

[9] Frank András: *Poliéderes kombinatorika*, előadásjegyzet, 2003.

[10] B. Korte, J. Vygen: *Combinatorial Optimization*, Springer, 2000.

[11] G. Reinelt: *TSPLIB – A traveling salesman problem library*, ORSA Journal of Computing **6**, (1991) 376-384.