Jegyzet

az Operációkutatás II című tantárgyhoz

Király Tamás és Papp Olga

Utolsó frissítés: 2015. február

Tartalomjegyzék

1. fejezet

Lineáris programozás

1.1. TU mátrixok: kerekítés és színezés

1.1.1. Emlékeztető

Egy mátrix **teljesen unimoduláris (TU)**, ha minden aldeterminánsa 0, 1, vagy -1 értékű. Néhány egyszerű tulajdonság:

- TU mátrix transzponáltja is TU,
- Ha A TU, akkor $[A, -A]$ és $[A, I]$ is TU,
- Páros gráf incidencia-mátrixa és irányított gráf incidencia-mátrixa TU.

1.1.1. Tétel. *Ha $A \in \mathbb{R}^{m\times n}$ TU mátrix és $b \in \mathbb{Z}^m$, akkor a $\max\{cx : Ax \le b\}$ feladatnak, ha megoldható, van egész optimális megoldása, hiszen minden erős bázismegoldás egészértékű.*

1.1.2. Tétel (Farkas Lemma TU mátrixokra)**.** *Ha $A \in \mathbb{R}^{m\times n}$ TU mátrix és $b \in \mathbb{Z}^m$, akkor a következő két állítás közül pontosan az egyik igaz:*

1. *Az $Ax \le b$ egyenlőtlenség-rendszernek van egész megoldása.*
2. *Az $yA = 0$, $y \ge 0$, $yb < 0$ rendszernek van $y \in \{0,1\}^m$-beli megoldása.*

Legyen $D = (V, E)$ egy irányított gráf, és A az incidencia-mátrixa. Tudjuk, hogy A TU mátrix. Legyen $b \in \mathbb{Z}^V$ tetszőleges. Az $Ax = b, x \ge 0$ rendszer másképp úgy írható, hogy $\varrho_x(v) - \delta_x(v) = b_v$ minden $v \in V$-re, és $x \ge 0$. A TU Farkas Lemma erre a feladatra a következőt adja:

1.1.3. Tétel. *Legyen $D = (V, E)$ egy irányított gráf, és $b \in \mathbb{Z}^V$ tetszőleges. Pontosan akkor van olyan x nemnegatív egész vektor az éleken, amire $\varrho_x(v) - \delta_x(v) = b_v$ minden $v \in V$-re, ha nincs olyan $U \subseteq V$ (esetleg üres) halmaz, amire $\varrho_D(U) = 0$ és $\sum_{u\in U} b_u > \sum_{v\notin U} b_v$.*

1.1.2. Kerekítés és egyenletes színezés

1.1.4. Tétel. *Legyen $A \in \mathbb{R}^{m\times n}$ TU mátrix és $z \in \mathbb{R}^n$. Ekkor z kerekíthető úgy (minden komponensét alsó vagy felső egészrészre változtatva), hogy Az minden komponense kevesebb mint 1-gyel változik.*

Bizonyítás. Tekintsük a következő egyenlőtlenség-rendszert:

$$\lfloor z\rfloor \le x \le \lceil z\rceil$$
$$\lfloor Az\rfloor \le Ax \le \lceil Az\rceil$$

Ennek az egyenlőtlenség-rendszernek a mátrixa TU. Tudjuk, hogy van megoldása (maga z), tehát van egész megoldása is, ami pont a tételnek megfelelő kerekítést ad. □

Ennek a tételnek például következménye, hogy tetszőleges $M \in \mathbb{R}^{k\times l}$ valós mátrix elemeit lehet úgy kerekíteni, hogy minden sorösszeg és oszlopösszeg kevesebb mint 1-gyel változzon. Legyen ugyanis $A \in \mathbb{Z}^n$ a $K_{k,l}$ teljes páros gráf incidencia-mátrixa (ahol $n = kl$), és legyen $z \in \mathbb{R}^n$ az M mátrix elemei által alkotott vektor (a mátrix elemeit természetes módon megfeleltethetjük a gráf éleinek). Mivel A TU mátrix, alkalmazhatjuk a fenti tételt, ami pont olyan kerekítést ad, ahol a sor- és oszlopösszegek kevesebb mint 1-gyel változnak.

1.1.5. Tétel. *Legyen $A \in \mathbb{R}^{m\times n}$ TU mátrix, $b \in \mathbb{Z}^m$, és k pozitív egész szám. Legyen $z \in \mathbb{Z}^n$ olyan egész vektor, amire $Az \le kb$. Ekkor z előáll $z = z^1 + z^2 + \cdots + z^k$ alakban, ahol $z^i \in \mathbb{Z}^n$ és $Az^i \le b$ $(i = 1, \ldots, k)$.*

Bizonyítás. A tételt k szerinti indukcióval bizonyítjuk. $k = 1$-re az állítás triviális. Tehát tegyük fel, hogy a tétel $(k-1)$-re igaz, és legyen $z \in \mathbb{Z}^n$ olyan egész vektor, amire $Az \le kb$. Az $\{x \in \mathbb{R}^n : \; Az - (k-1)b \le Ax \le b\}$ poliéder nemüres, hiszen z/k benne van. Mivel az egyenlőtlenség-rendszer mátrixa TU, létezik egész megoldás is; legyen ez z^k. Ekkor $Az^k \le b$, és $A(z - z^k) \le (k-1)b$, tehát a $z - z^k$ vektorra és $(k-1)$-re használva az indukciót kész vagyunk. □

A tétel segítségével megmutatjuk, hogy egy TU mátrix oszlopainak mindig van egyenletes színezése.

1.1.6. Tétel. *Legyen $A \in \mathbb{R}^{m\times n}$ TU mátrix, és k pozitív egész szám. Ekkor A oszlopait k csoportra lehet osztani úgy, hogy minden sorban az egyes csoportokban az elemek összege legfeljebb 1-gyel tér el egymástól.*

Bizonyítás. Legyen $d \in \mathbb{Z}^m$ az A oszlopainak összege, és legyen $b_1 = \lfloor d/k\rfloor$, $b_2 = \lceil d/k\rceil$. Ekkor az $\{x \in \mathbb{R}^n : \; kb_1 \le Ax \le kb_2, x \ge 0\}$ poliéderben benne van az $\mathbf{1}$ vektor (az azonosan 1 vektor). Az előző tétel szerint léteznek $z^1, \ldots, z^k$ nemnegatív egész vektorok, amikre $z^1 + \cdots + z^k = \mathbf{1}$, és $b_1 \le Az^i \le b_2$. Álljon az oszlopok i-edik csoportja azokból az oszlopokból, ahol z^i megfelelő komponense 1. Ez pont jó csoportokra osztást ad. □

Alkalmazzuk ezt a tételt egy páros gráf incidencia mátrixára!

1.1.7. Következmény. *Adott egy $G = (V, E)$ páros gráf. Tetszőleges k pozitív egész számra ki lehet színezni G éleit k színnel úgy, hogy minden v csúcsra minden színosztályból vagy $\lfloor d_G(v)/k \rfloor$ vagy $\lceil d_G(v)/k \rceil$ él illeszkedik.*

Ennek speciális esete két ismert tétel:

- Ha $k = \Delta$ (a maximális fokszám): Kőnig élszínezési tételét kapjuk, miszerint E felbontható Δ darab párosításra.
- Ha $k = \delta$ (a minimális fokszám): Gupta tételét kapjuk, ami szerint E felbontható δ részre úgy, hogy minden rész fedi az összes csúcsot.

1.2. Szimplex módszer

Tekintsük a következő primál feladatot:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \\ \max\ &cx, \end{aligned}$$

ahol $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^{1\times n}$, és a változók vektora $x \in \mathbb{Q}^n$. Ha az A mátrix rangja $r(A) < m$, akkor vagy már az $Ax = b$ egyenletrendszer sem oldható meg, vagy valamelyik egyenlet redundáns. Tehát feltehetjük, hogy $r(A) = m$. A duális feladat:

$$\begin{aligned} yA &\geq c \\ \min &\, yb, \end{aligned}$$

ahol $y \in \mathbb{Q}^{1\times m}$, azaz minden egyenlethez egy duál változó tartozik. Idézzük fel az ilyen alakú feladatokra vonatkozó dualitás tételeket.

1.2.1. Tétel (Gyenge dualitás tétel)**.** *Legyen x primál megengedett megoldás és y duál megengedett megoldás. Ekkor teljesül*

$$cx \leq yb.$$

Bizonyítás. $cx \underbrace{\leq}_{x\geq 0, yA\geq c} (yA)x = y(\underbrace{Ax}_{=b}) = yb.$ □

1.2.2. Tétel (Erős dualitás tétel)**.** *Ha a primál feladat megoldható, és az optimuma korlátos (azaz cx nem lehet tetszőlegesen nagy), akkor*

$$\max cx = \min yb.$$

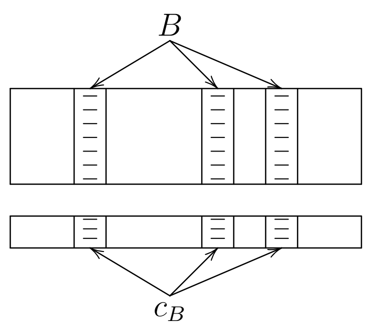
1.2.3. Tétel (Ekvivalens alak – komplementaritási feltétel)**.** *Ha x^* optimális primál megoldás, akkor $\exists\ y^*$ duál megoldás, amire $cx^* = y^*b$, azaz ha $x^*_j > 0$, akkor $(y^*A)_j = c_j$.*

Definíció (Bázis, bázismegoldás)**.** A primál feladat **bázismegoldása** egy olyan x megoldás, amire A-nak az $x_j > 0$-khoz tartozó oszlopai lineárisan függetlenek. Az A-nak egy $m \times m$-es nemszinguláris részmátrixát **bázis**nak nevezzük. Formálisan ebbe beleértjük, hogy a részmátrix oszlopainak egy sorrendje is adott.

Rögzített B bázis esetén egy $x \in \mathbb{R}^n$ vektort $x = (x_B, x_N)$ alakban írhatunk, ahol x_B-vel jelöljük a bázishoz tartozó koordinátákat, x_N-nel pedig a többit, azaz a nem-bázis koordinátákat. A B-hez tartozó primál vektor: $\bar{x} \in \mathbb{R}^n$: $\bar{x}_B = B^{-1}b$, $\bar{x}_N = 0$. Ha $B^{-1}b \geq 0$, akkor $\bar{x}$ primál megoldás, azaz a B bázis **primál megengedett**.

Megjegyzés. Ha x bázismegoldás, akkor az $x_j > 0$-khoz tartozó oszlopokat kiegészítve m db lineárisan független oszloppá bázist kapunk. Ehhez a B bázishoz pedig pont x lesz a hozzárendelt bázismegoldás, mivel tudjuk, hogy $Bx = b$-nek egyetlen megoldása van. Egy bázismegoldás viszont nem csak egy bázishoz lehet hozzárendelve: ha m-nél kevesebb helyen pozitív, akkor ezeket bárhogy kiegészíthetjük m lineárisan független oszloppá, így több, egymástól különböző bázist kaphatunk. Két bázist különbözőnek tekintünk akkor is, ha ugyanaz a bázismegoldás tartozik hozzájuk.

Definíció (Bázishoz tartozó duális vektor)**.** A B bázishoz tartozó duális vektor: $\bar{y} = c_B B^{-1}$, ahol c_B a c célfüggvény B bázishoz tartozó része.



Az $\bar{y}$ vektor nem feltétlenül megoldása a duál feladatnak.

Észrevétel. *Az $\bar{x}$ és $\bar{y}$ vektorok teljesítik a komplementaritási feltételeket.*

Nézzük $\bar{y}A - c$-t. Erről annyit tudunk, hogy a B-hez tartozó koordinátái nullák.

$$(\bar{y}A - c)_B = (\bar{y}B - c_B) = c_B B^{-1} B - c_B = 0$$

Ha $\bar{y}A - c \geq 0$, akkor $\bar{y}$ duál megoldás. Mivel a komplementaritási feltételek teljesülnek, $\bar{x}$ a primál feladatnak és $\bar{y}$ a duál feladatnak optimális megoldása.

Definíció. A B bázishoz tartozó **redukált költség**: $\bar{c} = \bar{y}A - c$. A bázis **duál megengedett** ha $\bar{c} \geq 0$, és **optimális** ha primál és duál megengedett.

Először a szimplex módszernek azt az egyszerűbb változatát tárgyaljuk, ahol kiindulásként rendelkezésre áll egy primál megengedett bázis, és a cél egy optimális bázis megtalálása.

1.2.1. A szimplex módszer tulajdonságai

- primál megengedett bázisokon lépked
- minden lépésben egy oszlopot cserélünk ki B-ben.
- a primál célfüggvényérték folyamatosan nő (azaz nem csökken)
- véges sok lépésben eljutunk egy optimális bázishoz.

Észrevétel. *A szimplex módszer során mindig teljesül, hogy* $c\bar{x} = \bar{y}b$.

Tegyük fel, hogy B primál megengedett bázis. Tartozik hozzá egy $\bar{x}$ bázismegoldás és egy $\bar{y}$ duál vektor. Figyeljük meg, hogy a $B^{-1}Ax = B^{-1}b$ egyenletrendszer ekvivalens az eredeti, $Ax = b$ egyenletrendszerrel. Vezessük be a következő jelöléseket:

$$\begin{aligned} \bar{A} &= B^{-1}A \\ \bar{b} &= B^{-1}b \\ \bar{c} &= \bar{y}A - c \qquad \text{– a bázishoz tartozó } \textbf{redukált költség}. \\ \bar{z} &= c\bar{x} = \bar{y}b \end{aligned}$$

A B bázis pontosan akkor primál megengedett, ha $\bar{b} \geq 0$, és pontosan akkor duál megengedett, ha $\bar{c} \geq 0$. Az itt bevezetett mátrixokat és vektorokat szokás egyetlen táblázatban ábrázolni, amit a B bázishoz tartozó szimplex táblának nevezünk:

$$\begin{array}{cc} \boxed{\qquad\qquad \bar{A} \qquad\qquad} & \boxed{\bar{b}} \\ \boxed{\qquad\qquad \bar{c} \qquad\qquad} & \boxed{\bar{z}} \end{array}$$

Megjegyzés. $\bar{A}$-ban B helyén egységmátrix van, $\bar{b}$ pedig $\bar{x}_B$ értékeit tartalmazza. Tehát ha pl. $\bar{x}_B = (\bar{x}_7, \bar{x}_3, \bar{x}_9)$, akkor $\bar{b}_1 = \bar{x}_7$, $\bar{b}_2 = \bar{x}_3$, $+\bar{b}_3 = \bar{x}_9$, és az $\bar{A}$ mátrixban így néz ki a megfelelő rész:

$$\begin{array}{r|c|c|c|c|c|c|c|} \multicolumn{1}{r}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{x_3} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{x_7} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{x_9} & \multicolumn{1}{c}{} \\ \cline{2-8} x_7 & & 0 & & 1 & & 0 & \\ x_3 & & 1 & & 0 & & 0 & \\ x_9 & & 0 & & 0 & & 1 & \\ \cline{2-8} \end{array}$$

A $\bar{c}$ vektort nem véletlenül nevezzük redukált költségnek. Írjuk át a következő alakra:

$$\bar{c} = \bar{y}A - c = c_B B^{-1} A - c = c_B \bar{A} - c$$

Jelölje N a bázisban nem szereplő indexek halmazát. Mi történik akkor, ha egy adott $p \in N$-re $\bar{x}_p$-t növeljük δ-val, és közben $\bar{x}_B$-t úgy változtatjuk, hogy $\bar{A}x = \bar{b}$ továbbra is teljesüljön?

		x_3		x_q		x_7		x_9	
x_7		0		–		1		0	
x_3		1		–		0		0	
x_9		0		–		0		1	

1. egyenletnél: $\bar{x}_7$-et változtatjuk
2. egyenletnél: $\bar{x}_3$-at változtatjuk
3. egyenletnél: $\bar{x}_9$-et változtatjuk

Jelölés. Az A mátrix i-edik sorát $a_{i.}$ vagy $A_{i.}$ jelöli, j-edik oszlopát pedig $a_{.j}$ vagy $A_{.j}$.

$$\begin{aligned} \bar{x}'_p &= \bar{x}_p + \delta \\ \bar{x}'_B &= \bar{x}_B - \delta \bar{a}_{.p} \\ c\bar{x}' &= c\bar{x} + \delta c_p - \delta c_B \bar{a}_{.p}. \end{aligned}$$

Tehát a célfüggvény-érték csökkenésének mértéke: $\delta(c_B\bar{A} - c)_p = \delta\bar{c}_p$. Azaz az x_p változó redukált költsége azt adja meg, hogy lokálisan mi a költsége a változó egységnyi növelésének. A szimplex módszer során olyan p-t választunk, amire $\bar{c}_p < 0$, így ez a csökkenés negatív, azaz a célfüggvény értéke nő (pontosabban nem csökken, mert majd látjuk, hogy $\delta = 0$ is előfordulhat), így minden lépésben az előzőnél jobb (azaz nem rosszabb) megoldást kapunk.

1.2.2. A szimplex módszer egy lépése

Feltesszük, hogy kiindulásként adott egy B primál megengedett bázis.

0. ha $\bar{c} \geq 0$, akkor kész vagyunk, hiszen a bázis optimális.
1. ha nem, válasszunk egy $p \in N$-t, amire $\bar{c}_p < 0$. Ezt többféleképpen megtehetjük:
   - Bland szabály: válasszuk a legkisebb ilyen p-t. Ez a választási módszer garantálja, hogy az algoritmusunk véges lesz (bizonyítás később).
   - válasszuk a legkisebb $\bar{c}_p$ értéket. Ez nem garantálja a végességet, de a gyakorlatban sokszor gyorsabb.

   Az így választott x_p kerül majd a bázisba.
2. Ha $\bar{a}_{.p} \leq 0$, akkor

1.2.4. Állítás. *Ilyenkor a célfüggvény nem korlátos.*

Bizonyítás.

$$\bar{x}'_p = \bar{x}_p + \delta$$

$$\bar{x}'_B = \bar{x}_B - \delta \bar{a}_{.p}$$

ami tetszőleges $\delta \geq 0$-ra megengedett megoldást ad, mivel $\bar{a}_{.p} \leq 0$. A célfüggvényérték tehát szigorúan nő ($-\delta \bar{c}_p$-vel), azaz tetszőlegesen nagy lehet. □

3. Ha $\bar{a}_{.p} \nleq 0$, akkor ki kell választani a bázisból kikerülő változót. Azt az r-et válasszuk, amire a

$$\frac{\bar{b}_r}{\bar{a}_{rp}} = \min_{i:\bar{a}_{ip}>0} \frac{\bar{b}_i}{\bar{a}_{ip}}$$

Ekkor a szimplex tábla r. sorához tartozó bázisváltozó kerül ki a bázisból.

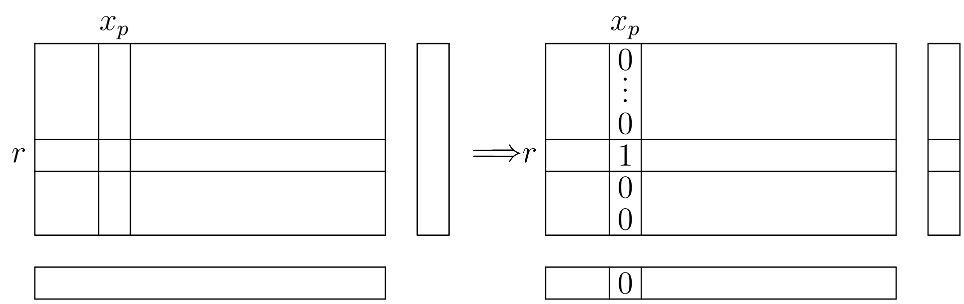
Ha több i is minimális, akkor alkalmazzuk a Bland szabályt: az a bázisváltozó kerül ki, amelyiknek az indexe a legkisebb.

4. Új szimplex tábla kiszámítása (pivotálás): a szimplex tábla r. sorának többszöröseit adjuk hozzá a többi sorhoz. ($\bar{b}_r$ is hozzátartozik a sorhoz, és a redukált költség sorát is módosítjuk.)

$$\bar{a}'_{rj} = \frac{\bar{a}_{rj}}{\bar{a}_{rp}} \qquad \bar{b}'_r = \frac{\bar{b}_r}{\bar{a}_{rp}}$$

$$i \neq r : \bar{a}'_{ij} = \bar{a}_{ij} - \bar{a}_{rj}\frac{\bar{a}_{ip}}{\bar{a}_{rp}} \qquad \bar{b}'_i = \bar{b}_i - \bar{b}_r\frac{\bar{a}_{ip}}{\bar{a}_{rp}}$$

$$\bar{c}'_j = \bar{c}_j - \bar{a}_{rj}\frac{\bar{c}_p}{\bar{a}_{rp}}$$



1.2.5. Tétel. *A Bland szabályt használva a szimplex módszer véges sok lépésben véget ér.*

Bizonyítás. Ha egy lépésnél változik $\bar{x}$, akkor $c\bar{x}$ szigorúan nő. Ezért csak abból lehetne probléma, hogy végtelen ciklusba kerülünk, miközben $\bar{x}$ nem változik. Tegyük fel indirekt, hogy van egy ilyen ciklus, aminek tehát az elején és a végén ugyanaz a bázis van.

Egy indexet mozgónak nevezünk, ha a hozzá tartozó változó a ciklus során ki- illetve bekerül a bázisba. A nem mozgó indexek tehát a ciklus során vagy végig a bázisban vannak, vagy végig a bázison kívül.

Legyen p a legnagyobb mozgó index, és legyen t_1 egy olyan lépés, amikor bekerül, és t_2 egy olyan lépés, ahol kikerül. Feltehetjük, hogy $t_1 < t_2$. Jelölés: a t_1 lépés előtt: $B, \bar{B}, \bar{c}, \bar{A}$; a t_2 lépés előtt: $B', \bar{B}', \bar{c}', \bar{A}'$.

Mivel p kerül be a t_1-edik lépésben, $\bar{c}_p < 0$ és $j < p$ esetén $\bar{c}_j \geq 0$.

Nézzük most a t_2-edik lépést: legyen r az x_p bázisváltozóhoz tartozó sor, és legyen q az az index, ami bekerül a bázisba. Ekkor $\bar{c}'_q < 0$, $\bar{a}'_{rq} > 0$, és $\bar{a}'_{iq} \leq 0$ az összes olyan i-re, ami mozgó bázisváltozóhoz tartozik. Az utóbbi azért igaz, mert ezekre az i-kre $\bar{b}'_i = 0$, és az ezekhez a sorokhoz tartozó változóknak p-nél kisebb az indexük.

A fent elmondottakból

$$0 < \bar{c}_q^{t_1} - \bar{c}_q^{t_2} = \bar{c}_B B^{-1} a_{.q} - \bar{c}_{B'} (B')^{-1} a_{.q} = (c_B B^{-1} B' - c_{B'}) \bar{a}'_{.q} = \bar{c}_{B'} \bar{a}'_{.q}.$$

De ha a jobb oldalon szereplő skalárszorzatot tagonként nézzük, a $\bar{c}_p \bar{a}'_{rq}$ tag szigorúan kisebb mint nulla, a többi mozgó indexhez tartozó tag legfeljebb 0, míg a nem mozgó indexekhez tartozó tagok értéke 0 (hiszen ha egy ilyen j index benne van B'-ben, akkor B-ben is benne van, tehát $\bar{c}_j = 0$).

Azt kaptuk, hogy $\bar{c}_{B'} \bar{a}'_{.q} < 0$, ellentmondás. □

1.2.3. Érzékenységvizsgálat

Legyen B optimális bázis. A gyakorlatban előforduló feladatoknál sokszor hasznos tudni, hogy a megoldásunk mennyire érzékeny a bemeneti adatok változásaira. Ebben a részben azt vizsgáljuk, hogy mennyire változtathatjuk meg a c-nek vagy b-nek egy adott koordinátáját, hogy B optimális maradjon.

Tudjuk, hogy B pontosan akkor optimális, ha $\bar{b} \geq 0$ (primál megengedett) és $\bar{c} \geq 0$ (duál megengedett).

Nem-bázis változó súlyának változtatása: $p \in N$-re: $c'_p = c_p + \delta$ valamilyen valós δ-ra.

- $\bar{b}$ nem változik, ezért B primál megengedett marad
- $\bar{c}' = \bar{y}A - c' = \underbrace{c_B \bar{A}}_{\text{nem változik}} - \underbrace{c'}_{\text{csak ez változik}}$

Tehát $\bar{c}'_j = \bar{c}_j$ ha $j \neq p$ és $\bar{c}'_p = \bar{c}_p - \delta$, vagyis B akkor és csak akkor marad optimális bázis, ha $\delta \leq \bar{c}_p$.

Bázisváltozó súlyának változtatása: Az r sorhoz tartozó bázisváltozó súlyát növeljük δ-val.

- $\bar{b}$ nem változik, ezért B primál megengedett marad
- $\bar{c}' = c'_B\overline{A} - c'$. Ekkor $\bar{c}'_B \equiv 0$ (ez mindig igaz), $\bar{c}'_N = \bar{c}_N + (\delta\bar{a}_{r.})_N$. Azaz $j \in N$ esetén $\bar{c}'_j = \bar{c}_j + \delta\bar{a}_{rj}$. Ez mikor marad nemnegatív?
  - Ha $\bar{a}_{rj} = 0$, akkor mindig.
  - Ha $\bar{a}_{rj} > 0$, akkor szükséges, hogy $\delta \geq -\frac{\bar{c}_j}{\bar{a}_{rj}}$
  - Ha $\bar{a}_{rj} < 0$, akkor szükséges, hogy $\delta \leq -\frac{\bar{c}_j}{\bar{a}_{rj}}$

Tehát

$$\bar{c}' \geq 0 \Leftrightarrow \max\{-\frac{\bar{c}_j}{\bar{a}_{rj}} : j \in N, \bar{a}_{rj} > 0\} \leq \delta \leq \min\{-\frac{\bar{c}_j}{\bar{a}_{rj}} : j \in N, \bar{a}_{rj} < 0\}.$$

Látjuk, hogy az alsó korlát egy nempozitív szám, a felső korlát egy nemnegatív szám, de mindkettő lehet nulla is. Továbbá ha üreshalmazon maximalizálunk, akkor $-\infty$ az alsó korlát, és ha üreshalmazon minimalizálunk, akkor $+\infty$ a felső korlát.

Jobboldal változtatása: Legyen $b'_r = b_r + \delta$. Ekkor

- $\bar{c}$ nem változik, ezért B duál megengedett marad.
- $\bar{b}' = B^{-1}b'$, tehát $\bar{b}'_i = B^{-1}_{i.}b' = \bar{b}_i + \delta B^{-1}_{ir}$.

Hasonlóan az előző esethez:

$$\bar{b}' \geq 0 \Leftrightarrow \max\{-\frac{\bar{b}_i}{B^{-1}_{ir}} : \bar{b}_i > 0\} \leq \delta \leq \min\{-\frac{\bar{b}_i}{B^{-1}_{ir}} : \bar{b}_i < 0\}.$$

Ha a feladatunkat egyenlőtlenség-rendszerből kaptuk kiegészítő változók hozzávételével, akkor B^{-1} és $\bar{y}$ is könnyen kiolvasható a szimplex táblából:

$$A : \begin{array}{|c|c|} \hline \qquad\qquad & \mathcal{I} \\ \hline \end{array} \qquad \bar{A} : \begin{array}{|c|c|} \hline \qquad\qquad & B^{-1} \\ \hline \end{array}$$

$$c : \begin{array}{|c|c|} \hline \qquad\qquad & 0\ldots0 \\ \hline \end{array} \qquad \bar{c} : \begin{array}{|c|c|} \hline \qquad\qquad & \bar{y} \\ \hline \end{array}$$

Nézzük meg, hogyan változik a célfüggvényérték a fenti változtatás során, ha B optimális bázis marad:

$$\bar{z}' = \bar{y}b' = \bar{z} + \delta\bar{y}_r.$$

Definíció. A $\bar{y}$ vektort **árnyékár** vektornak is nevezzük, mivel $\bar{y}_r$ meghatározza, hogy – élve a termelési feladat példájával – milyen egységáron érdemes az r-edik alapanyagból vásárolni (feltéve, hogy a vásárolt mennyiség a fenti határokon belül marad).

1.2.4. Módosított szimplex módszer

A szimplex módszer számítógépes implementációjakor nem érdemes a teljes szimplex táblát nyilvántartani. Vegyük észre, hogy a bázisba belépő x_p változó kiválasztásához csak a $\bar{c}$ vektorra van szükség. Ha ez megvan, a kilépő változót a $\bar{b}$ vektor és az $\bar{a}_{.p}$ oszlop segítségével határozzuk meg. Ez összesen $2m+n$ adat az $(m+1)\times(n+1)$-es szimplex táblából! Kérdés, hogy ezeket ki tudjuk-e számolni anélkül, hogy az egész táblát nyilvántartanánk.

A válasz az, hogy igen, feltéve hogy ismerjük az aktuális bázis inverzét. Valóban, az ismert képletek alapján

$$\begin{aligned} \bar{c} &= c_B B^{-1} A - c \\ \bar{b} &= B^{-1} b \\ \bar{a}_{.p} &= B^{-1} a_{.p} \end{aligned}$$

A **módosított szimplex módszer** lényege, hogy a szimplex tábla fenntartása helyett mindig csak az aktuális bázis inverzét számoljuk ki, és ennek segítségével számoljuk a fenti mennyiségeket. Az előző részben láttuk, hogy ha a feladatunkat egyenlőtlenség-rendszerből kaptuk kiegészítő változók hozzávételével, akkor B^{-1} nem más, mint a szimplex táblának a kiegészítő változókhoz tartozó része.

$$A: \begin{array}{|c|c|}\hline \qquad\qquad & \quad\mathcal{I}\quad \\ \hline\end{array} \qquad\qquad \bar{A}: \begin{array}{|c|c|}\hline \qquad\qquad & \quad B^{-1}\quad \\ \hline\end{array}$$

Tehát elég a táblának ezt az $m\times m$-es részét fenntartani; cserébe viszont a $\bar{c}, \bar{b}, \bar{a}_{.p}$ vektorok kiszámolásához mátrix-szorzás kell. Ha n jóval nagyobb mint m, akkor ez jelentős időmegtakarítást eredményez.

Általános esetben a módosított szimplex módszer az úgynevezett LU-felbontás segítségével valósítható meg hatékonyan.

1.3. Duál szimplex módszer

Ha kezdetben nem ismerünk primál megengedett bázist, de duál megengedettet igen, akkor használhatjuk a duál szimplex módszert. Mint később látni fogjuk, ez a helyzet például akkor, amikor egy megoldott feladatnál kiderül, hogy újabb feltételeket kell hozzávenni.

1.3.1. A duál szimplex módszer tulajdonságai

- duál megengedett bázisokon lépked
- minden lépésben egy oszlopot cserélünk ki B-ben.

- ugyanazt a szimplex táblát használjuk, mint a primál szimplex módszernél
- $c\bar{x}$ folyamatosan csökken (azaz nem nő)
- véges sok lépésben eljutunk egy primál megengedett bázishoz.

A fő különbség a primál szimplex módszerhez képest, hogy először a bázisból kilépő változót határozzuk meg, és csak utána a belépőt.

1.3.2. A duál szimplex módszer egy lépése

0. Ha $\bar{b} \geq 0$, akkor kész vagyunk. Primál megengedett bázisunk van, azaz optimális bázist találtunk.

1. Ha nem, válasszunk egy r-et, amire $\bar{b}_r < 0$. Ezt többféleképpen megtehetjük:

   - Bland szabály: válasszuk azt az r-et, amihez a legkisebb indexű bázisváltozó tartozik. Ez a választási módszer garantálja, hogy az algoritmusunk véges lesz.
   - Válasszuk a legkisebb $\bar{b}_r$ értéket. Ez nem garantálja a végességet, de a gyakorlatban gyorsabb.

   Az így választott r-hez tartozó bázisváltozó lép ki a bázisból.

2. Ha $\bar{a}_{r.} \geq 0$, akkor

   1.3.1. Állítás. *A primál feladatnak nincs megoldása.*

   Bizonyítás.
   $$\underbrace{\bar{a}_{r.}x}_{\geq 0} = \underbrace{\bar{b}_r}_{<0}$$
   egy érvényes egyenlet lenne, ami nem lehetséges. □

3. Ha $\bar{a}_{r.} \not\geq 0$, akkor ki kell választani a bázisba bekerülő változót. Azt az x_p-t válasszuk, amire $\bar{a}_{rp} < 0$ és
   $$-\frac{\bar{c}_p}{\bar{a}_{rp}} = \min\{-\frac{\bar{c}_j}{\bar{a}_{rj}} : j \in N, \bar{a}_{rj} < 0\}$$

   Megjegyzés. Ha ez a minimum nulla, akkor degeneráció lép fel, tehát $\bar{c}$ nem változik a báziscsere során.

   Ha több p is minimális, akkor alkalmazzuk a Bland szabályt: válasszuk a minimális ilyen p-t. Az x_p változó kerül a bázisba.

   Miért pont így kell választani a bemenő változót? Arra van szükségünk, hogy $\bar{c} \geq 0$ maradjon. Báziscsere után: $\bar{c}'_j = \bar{c}_j - \bar{a}_{rj}\frac{\bar{c}_p}{\bar{a}_{rp}}$ nemnegatív marad, ha

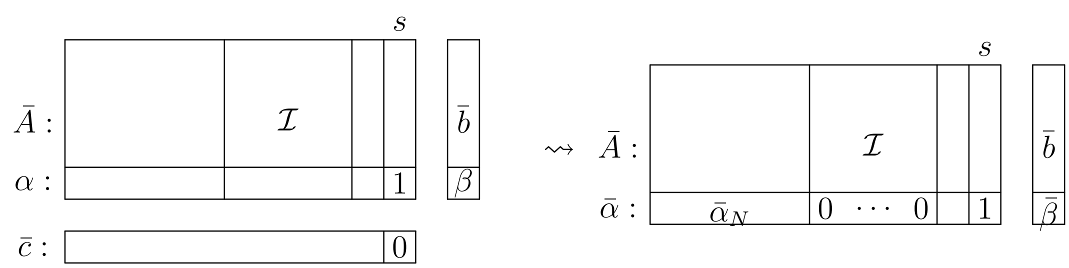
- $\bar{a}_{rj} \geq 0$, mivel $\bar{c}_j$-t ekkor növeljük
- $\bar{a}_{rj} < 0$, de $-\frac{\bar{c}_j}{\bar{a}_{rj}} \geq -\frac{\bar{c}_p}{\bar{a}_{rp}}$

4. A báziscsere ugyanúgy történik, mint a primál szimplex módszernél.

1.3.3. Alkalmazás: új feltétel hozzávétele

Tegyük fel, hogy már megoldottunk egy feladatot, és kiderül, hogy hozzá kell vennünk még a rendszerhez egy $\alpha x \leq \beta$ feltételt.

Vegyünk egy új kiegészítő változót: $s \geq 0$ úgy, hogy $\alpha x + s = \beta$. Írjuk át a feltételt ekvivalensen: $\bar{\alpha} x + s = \bar{\beta}$, ahol $\bar{\alpha}_B = 0$. Azaz:



Megjegyzés. $\bar{c}$-on nem kell változtatni, ugyanis a kiegészítő változóhoz 0 tartozik $\bar{c}$-ban.

A bázist kibővítjük s-sel, így az új feladatra egy duál megengedett bázist kapunk, és innentől fogva használhatjuk a duál szimplex módszert, mivel kaptunk egy kiindulási táblát.

1.3.4. Alkalmazás: primál megengedett bázis keresése

A duál szimplex módszert használhatjuk egy primál megengedett bázis megkeresésére is. Nézzük a következő primál feladatot és a hozzá tartozó duált:

$$
(P) \quad \begin{array}{r} Ax = b \\ x \geq 0 \\ \max 0x \end{array} \qquad\qquad (D) \quad \begin{array}{l} yA \geq 0 \\ \\ \min yb \end{array}
$$

A (P) feladatnak minden megengedett megoldása optimális, miközben a (D) feladatnak a $(0 \ldots 0)$ egy megengedett megoldása.

Ebben a feladatban $\bar{c} = \underbrace{c_B \bar{A}}_{=0} - \underbrace{c}_{=0} = 0$, tehát minden bázis duál-megengedett.

Induljunk ki tetszőleges bázisból, és használjuk a duál szimplex módszert. Ekkor vagy kapunk egy primál megengedett bázist, vagy kapunk egy bizonyítékot arra, hogy a feladat nem megoldható. (Ez a bizonyíték pont a Farkas lemmából következik.)

1.4. Kétfázisú szimplex módszer

A gyakorlatban, ha nem áll rendelkezésre kezdeti primál megengedett bázis, az úgynevezett kétfázisú szimplex módszert szoktuk használni. Ennek első fázisában primál megengedett bázist keresünk, míg a második fázisban ebből a bázisból kiindulva alkalmazzuk a primál szimplex módszert.

Nézzük az **első fázist**. Legyen a feladat: $\max\{cx : \; Ax = b, x \geq 0\}$. Feltehető, hogy $b \geq 0$, mert egyenleteket szorozhatunk (-1)-gyel. Vezessünk be minden sorhoz új mesterséges változókat: u_i $(i = 1 \ldots m)$.

$$\begin{aligned} Ax + Iu &= b \\ (x, u) &\geq 0 \end{aligned}$$

A fenti rendszer nem ekvivalens az eredetivel. Ahhoz, hogy az eredeti feladat megoldását kapjuk, olyan megoldást kell keresni, ahol $u = 0$. Ennek megtalálásához az első fázisban legyen a célfüggvény $\max - \sum_{i=1}^{m} u_i$.

- Ha ennek a feladatnak 0 az optimuma, akkor az eredeti feladat egy megoldását kaptuk.
- Ha ennek a feladatnak < 0 az optimuma, akkor az eredeti feladatnak nincs megoldása.

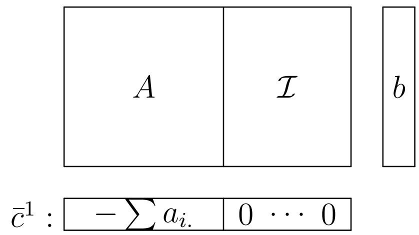
Legyen B a mesterséges változók oszlopaiból álló bázis. Ekkor B primál megengedett, tehát használható az első fázisban kiindulási bázisként.

$$\begin{array}{rl} A^1 : & \begin{array}{|c|c|} \hline \quad A \quad & \quad \mathcal{I} \quad \\ \hline \end{array} \quad \begin{array}{|c|} \hline b \\ \hline \end{array} \\[2ex] c^1 : & \begin{array}{|c|c|} \hline 0 \quad \cdots \quad 0 & -1 \cdots -1 \\ \hline \end{array} \end{array}$$

$$\begin{aligned} \bar{A}^1 &= A^1 \\ \bar{b}^1 &= b \\ \bar{c}^1 &= c_B^1 \bar{A}^1 - c^1 \end{aligned}$$

Tudjuk, hogy $\bar{c}_B^1 = 0$, és $\bar{c}_N^1 = -\sum_{i=1}^{m} a_{i.}$.

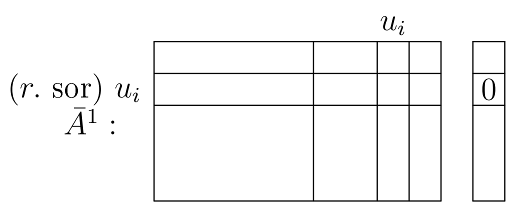
A kiindulási szimplex tábla tehát:



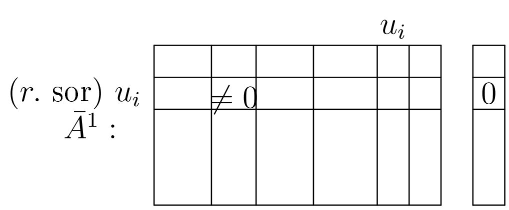
Erre a szimplex táblára kell alkalmazni a primál szimplex módszert.

- Ha az optimum < 0, akkor nincs megoldás.

- Ha az optimum $= 0$, de marad mesterséges változó a bázisban (azaz a feladatunk degenerált volt), akkor tegyük a következőt a mesterséges változók kiküszöbölése érdekében:

  Legyen a megmaradt változó a szimplex tábla r-edik sorában. Ez a sor nem lehet azonosan 0, mert A sorai lineárisan függetlenek. Válasszunk tehát egy tetszőleges nemnulla elemet egy nem-mesterséges változó oszlopában, és ott pivotáljunk:

  Mivel a jobboldalon 0 van, ezért a primál megoldás nem változik, és a célfüggvényérték sem változik.

- Ha az optimum $= 0$ és nincs mesterséges változó a bázisban (azaz az eredeti feladatra kaptunk egy megengedett bázist), akkor áttérhetünk a **második fázisra:** elhagyjuk a mesterséges változókat, és az eredeti c célfüggvényre alkalmazzuk a szimplex módszert.

1.5. Hálózati szimplex módszer

Adott egy $D = (V, E)$ irányított, gyengén összefüggő gráf, $c : E \to \mathbb{R}$ élsúlyokkal. Ezen túl adott egy $b : V \to \mathbb{Z}$ igényfüggvény, amelyik minden csúcsra előírja, hogy mennyi legyen a bemenő és a kimenő folyam különbsége. Feltesszük, hogy $\sum_{v \in V} b_v = 0$.

Legyen $x : E \to \mathbb{R}$ a változók vektora.

Jelölés (Bemenő folyam). $\varrho_x(v)$ jelöli a v csúcsba belépő éleken az x-ek összegét.

Jelölés (Kimenő folyam). $\delta_x(v)$ jelöli a v csúcsból kilépő éleken az x-ek összegét.

A következő alakú feladatot szeretnénk megoldani:

$$\begin{aligned} \varrho_x(v) - \delta_x(v) = b_v \qquad & \forall v \in V \\ x \geq 0 \\ \max \sum_{e \in E} c_e x_e \end{aligned}$$

Megjegyzés. Ha $\sum b_v = 0$ nem lenne igaz, akkor a feladatnak nem lenne megoldása.

Legyen v_0 egy kijelölt csúcs. Ha $\varrho_x(v) - \delta_x(v) = b_v$ minden $v \in V \setminus \{v_0\}$-ra teljesül, akkor v_0-ra is teljesül. Vezessük be tehát az eredeti egyenlőségrendszer helyett a következőt: $\varrho_x(v) - \delta_x(v) = b_v \ \forall v \in V \setminus \{v_0\}$. Így a sorok lineárisan függetlenek lesznek, ezáltal a feladatra alkalmazhatjuk a szimplex módszert: $Ax = b$, $x \geq 0$, ahol A sorai lineárisan függetlenek. Az A mátrixunk a következő lesz:

$$V = \{v_0, v_1, \dots, v_n\}, \ E = \{e_1, \dots, e_m\}$$

$$A \in \mathbb{Z}^{nxm}$$

$$a_{ij} = \begin{cases} -1 & \text{ha } v_i \text{ töve } e_j\text{-nek} \\ +1 & \text{ha } v_i \text{ feje } e_j\text{-nek} \\ \ \ 0 & \text{különben} \end{cases}$$

Megjegyzés. Az A mátrix minden oszlopában legfeljebb egy $+1$-es és legfeljebb egy -1-es található, ezért hálózati mátrix.

Jelölés. A továbbiakban kontextustól függően a következő ekvivalens jelöléseket fogjuk használni:

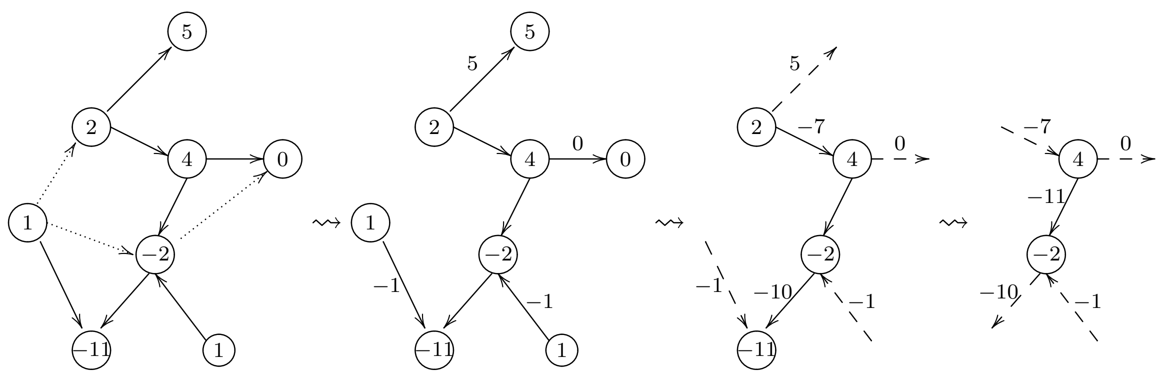
$$\begin{aligned} b_{v_i} &\sim b_i \\ c_{e_j} &\sim c_j \\ y_{v_i} &\sim y_i \\ x_{v_i} &\sim x_i \end{aligned}$$

A feladatunk tehát felírható $\max\{cx, Ax = b, x \geq 0\}$ alakban.

1.5.1. Állítás. *B bázis $\Leftrightarrow$ $\{e_j : j \in B\}$ feszítőfa (irányítás nélkül).*

Bizonyítás. $\Leftarrow$: Belátjuk, hogy $Bx = b$ egyértelműen megoldható. Keressünk a fán olyan folyamot, ami minden igényt kielégít, azaz élre adjunk olyan értéket, ahol $\varrho_x(v) - \delta_x(v) = b_v$. A fa leveleire egy-egy él illeszkedik. Ezekre egyértelműen meg tudjuk adni a változó értékét.

Ha a leveleket elhagyjuk, akkor újabb fát kapunk, amely fa leveleire illeszkedő élekre ugyancsak egyértelműen meghatározható a változó értéke, és így tovább.



$\Rightarrow$: Indirekt bebizonyítjuk, hogy ha a B-hez tartozó élek nem alkotnak feszítőfát, akkor B nem bázis. Adott tehát n darab él, ami nem alkot feszítőfát. Ekkor a részgráf tartalmaz kört. Legyen ez a kör C, és legyen

$$x_e = \begin{cases} +1 & \text{, ha } e \in C \text{ előre-él} \\ -1 & \text{, ha } e \in C \text{ hátra-él} \\ 0 & \text{, ha } e \notin C \end{cases}$$

Ekkor a $Bx = 0$ és $x \neq 0$, tehát B szinguláris, azaz B nem bázis. $\square$

Az alábbiakban ismertetett hálózati szimplex módszer tulajdonképpen egyszerűen a szimplex módszer alkalmazása a feladatunkra, de mátrixok helyett gráfelméleti fogalmakkal elmondva. Amint látni fogjuk, ennek előnye, hogy az algoritmus során nem kell szorzást és osztást végezni, csak összeadást és kivonást, ezért nem merülhetnek fel numerikus pontatlanságok.

Legyen $\bar{x}$ a $Bx = b$ egyértelmű megoldása és legyen $\bar{y}$ a következő: $\bar{y}_0 = 0$ lesz a v_0-hoz tartozó duális változó. Ha $uv \in B$, akkor legyen $\bar{y}_v - \bar{y}_u = c_{uv}$; ez egyértelműen meghatározza $\bar{y}$-t (v_0-ból kiindulva kiszámolható). A továbbiakban az egyszerűség kedvéért B-vel jelöljük az $\{e_j : j \in B\}$ feszítőfát is.

Az uv él redukált költsége: $\bar{c}_{uv} = \bar{y}_v - \bar{y}_u - c_{uv}$. A korábbiaknak megfelelően a B bázis primál megengedett, ha $\bar{x} \geq 0$, és duál megengedett, ha $\bar{c} \geq 0$.

1.5.1. Primál hálózati szimplex módszer lépései

Tegyük fel, hogy B primál megengedett bázis. Az alábbi lépéssorozatnál nem kell az A értékeivel műveleteket végezni, csak az $\bar{x}, \bar{y}, \bar{c}$ vektorokkal.

0. Ha $\bar{c} \geq 0$, akkor kész vagyunk (a bázisunk primál és duál megengedett).

1. Ha nem, akkor válasszunk egy olyan e_p élt, amire $\bar{c}_p < 0$. Az így választott e_p él kerül majd a bázisba.

2. Vegyük hozzá a B feszítőfához az e_p élt. Ekkor egy egyértelmű C kört kapunk, aminek e_p éle. Nevezzük a C-ben e_p-vel egyirányú éleket előreéleknek, a többi C-beli élt pedig hátraélnek.

   Ha C-ben nincsenek hátraélek, akkor tetszőleges $\delta > 0$-ra

   $$\bar{x}' = \begin{cases} \bar{x}_e + \delta, \text{ ha } e \in C \\ \bar{x}_e, \text{ különben} \end{cases}$$

   megengedett megoldás.

   1.5.2. Állítás. *Ebben az esetben a célfüggvény nem korlátos.*

   Bizonyítás. $\sum_{uv \in C} \bar{c}_{uv} = \sum_{uv \in C} (\bar{y}_v - \bar{y}_u - c_{uv}) = -\sum_{uv \in C} c_{uv}$, tehát

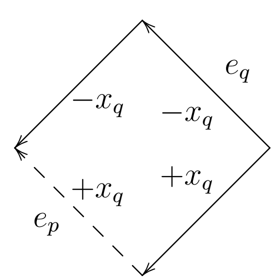
   $$c\bar{x}' = c\bar{x} + \delta \sum_{uv \in C} c_{uv} = c\bar{x} - \delta \sum_{uv \in C} \bar{c}_{uv} = c\bar{x} - \delta \bar{c}_p \xrightarrow[\delta \to \infty]{} +\infty$$

   □

3. Ha van C-ben hátraél, akkor legyen e_q az a hátraél, amire $\bar{x}_q$ minimális. Ez az él fog kikerülni a bázisból.

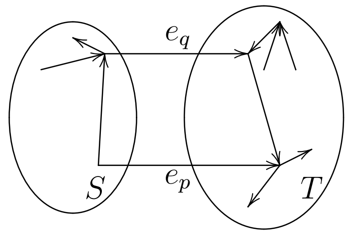
4. Vegyük hozzá a B bázishoz az e_p élt, és hagyjuk el a bázisból az e_q élt: $B' = B + \{p\} - \{q\}$.

   $$\bar{x}'_j = \begin{cases} \bar{x}_j + \bar{x}_q, \text{ ha } e_j \text{ előreél } C\text{-ben} \\ \bar{x}_j - \bar{x}_q, \text{ ha } e_j \text{ hátraél } C\text{-ben} \\ \bar{x}_j, \text{ ha } e_j \notin C \end{cases}$$



   Számoljuk ki az $\bar{y}'$-t:

Ha az e_p élt kihagyjuk a fából, akkor a fa két komponensre esik: S és T. Válasszuk S-t és T-t úgy, hogy e_p a T-be lépjen.



Ekkor

- Ha $v_o \in S$, akkor
$$\bar{y}'_v = \begin{cases} \bar{y}_v, \text{ ha } v \in S \\ \bar{y}_v - \bar{c}_p, \text{ ha } v \in T \end{cases}$$
- Ha $v_o \in T$, akkor
$$\bar{y}'_v = \begin{cases} \bar{y}_v + \bar{c}_p, \text{ ha } v \in S \\ \bar{y}_v, \text{ ha } v \in T \end{cases}$$

A fenti megkülönböztetés azért szükséges, hogy $y_0 = 0$ maradjon.

Megjegyzés. Ha b és c egészek, akkor $\bar{y}$, $\bar{x}$ és $\bar{c}$ végig egészek maradnak. Sőt, ha a költségek(súlyok) egészek, a duál végig egész lesz, és ha az igények egészek, a primál végig egész lesz. Tehát egész igények esetén a hálózati feladatnak mindig van egész optimális megoldása, ha egyáltalán megoldható. A ciklizálás elkerülése érdekében használhatjuk a Bland szabályt.

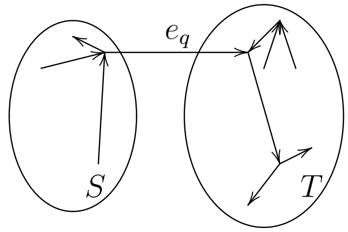
1.5.2. Duál hálózati szimplex módszer

Természetesen a hálózati szimplex módszernek is van duál változata. Tegyük fel, hogy kezdetben van egy B duál megengedett bázis (azaz $\bar{c} \geq 0$). Az általános lépés a következő:

0. Ha $\bar{x} \geq 0$, akkor kész vagyunk (a bázisunk primál és duál megengedett).

1. Ha nem, akkor legyen $q \in B$ olyan, hogy $\bar{x}_q < 0$. Ha több alternatívánk van, alkalmazzuk a Bland szabályt: válasszuk a legkisebb indexű ilyet. Az így választott változó kerül majd ki a bázisból.

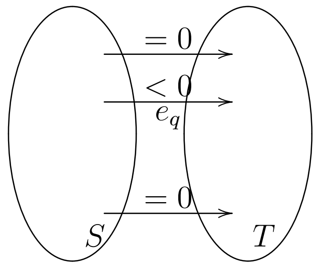
   Ha elhagyjuk az e_q élt, akkor a fa két részre esik: S és T. Válasszuk S-t és T-t úgy, hogy e_q a T-be lépjen.



2.

1.5.3. Állítás. *Ha nincs T-ből S-be vezető él, akkor a primál feladatnak nincs megoldása.*

Bizonyítás. Az $\bar{x}$ aktuális primál vektor a $\varrho_x(v) - \delta_x(v) = b_v$, $\forall v \in V$ feladat megoldása. Az e_q élen: $\bar{x} < 0$. A többi S-ből T-be vezető élen $\bar{x} = 0$, mivel ezek nincsenek a bázisban.



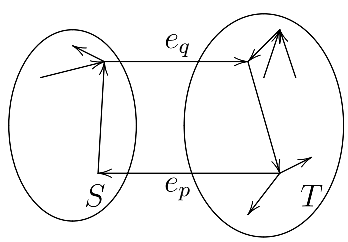
Emiatt S igénye nagyobb T igényénél, hiszen

$$\sum_{v\in T} b_v - \sum_{v\in S} b_v = \sum_{v\in T}(\varrho_{\bar{x}}(v) - \delta_{\bar{x}}(v)) - \sum_{v\in S}(\varrho_{\bar{x}}(v) - \delta_{\bar{x}}(v)) = 2\sum_{e:S\to T} \bar{x}_e < 0.$$

Nemnegatív folyammal $T \to S$ élek híján nem lehet az igényeket kielégíteni, tehát nincs megoldás. □

3. Ha van T-ből S-be él, akkor válasszuk ki azt az e_p $T \to S$ élt, amire $\bar{c}_p = \min\{\bar{c}_e$, ahol $e : T \to S$ él$\}$. Ez az él fog bekerülni a bázisba.

4. Hagyjuk el a B bázisból az e_q élt, és vegyük hozzá az e_p élt: $B' = B - \{q\} + \{p\}$.



Az $\bar{x}'$ kiszámolásához legyen C a $B \cup \{e_p\}$ egyetlen köre. Nevezzük az e_p-vel egyirányú éleket előreéleknek, a vele ellentétes irányú éleket pedig hátraéleknek. Ekkor

$$\bar{x}'_e = \begin{cases} \bar{x}_e - \bar{x}_q, \text{ ha } e \text{ előreél } C\text{-ben} \\ \bar{x}_e + \bar{x}_q, \text{ ha } e \text{ hátraél } C\text{-ben} \\ \bar{x}_e, \text{ ha } e \notin C \end{cases}$$

Számoljuk ki az $\bar{y}'$-t:

- Ha $v_o \in S$, akkor

$$\bar{y}'_v = \begin{cases} \bar{y}_v, \text{ ha } v \in S \\ \bar{y}_v + \bar{c}_p, \text{ ha } v \in T \end{cases}$$

- Ha $v_o \in T$, akkor

$$\bar{y}'_v = \begin{cases} \bar{y}_v - \bar{c}_p, \text{ ha } v \in S \\ \bar{y}_v, \text{ ha } v \in T \end{cases}$$

1.5.3. Kezdeti primál bázis keresése

Több módszert is ismertetünk:

1. A $c \equiv 0$ súlyfüggvényre alkalmazzuk a duál hálózati szimplex módszert. Ilyenkor tetszőleges B bázisra $\bar{y} \equiv 0$, $\bar{c} \equiv 0$. A gyakorlatban ez lassú módszer, és csak azért nem ciklizál, mert a Bland szabályt alkalmazzuk.

2. Legyen B tetszőleges bázis. Ha ez a bázis se nem primál-, se nem duál megengedett, akkor módosítsuk c-t úgy, hogy duál megengedett legyen:

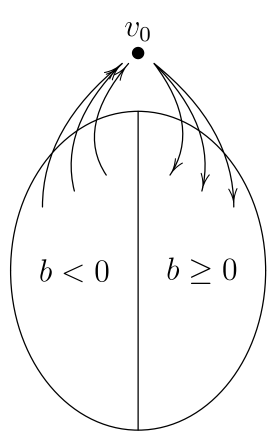
$$c'_{uv} = \begin{cases} c_{uv}, \text{ ha } \bar{c}_{uv} \geq 0 \\ \bar{y}_v - \bar{y}_u, \text{ ha } \bar{c}_{uv} < 0 \end{cases}$$

   Ezzel a c'-vel B duál megengedett lesz. Most alkalmazzuk a duál hálózati szimplex módszert, ezáltal B' optimális bázist kapunk c'-re, ami primál megengedett bázis az eredeti c célfüggvényre. Ezzel a B'-vel kezdve alkalmazzuk a primál szimplex módszert az eredeti c célfüggvényre.

3. **Kétfázisú hálózati szimplex módszer**

   Vegyük hozzá az eredeti gráfhoz a következő új éleket: $E' = E \cup \{v_0 v : b_v \geq 0\} \cup \{v v_0 : v_v < 0\}$. Legyen

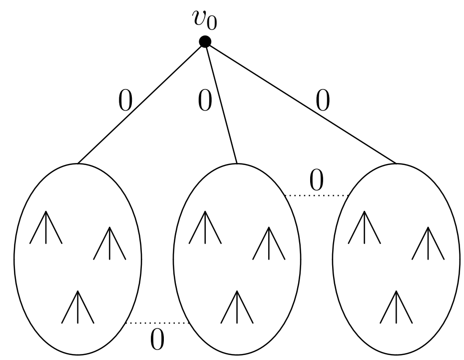
$$c'_e = \begin{cases} -1, \text{ ha } e \in E' \setminus E \\ 0, \text{ ha } e \in E. \end{cases}$$



   Az új élek primál megengedett bázist határoznak meg, mivel $\bar{x}_{v_0 v} = b_v$ és $\bar{x}_{v v_0} = -b_v$, tehát $\bar{x} \geq 0$. Alkalmazzuk a primál hálózati szimplex módszert. Ekkor

   - Ha az optimum negatív, akkor az eredeti feladatnak nincs megoldása.

- ha az optimum nulla, akkor hagyjuk el az $E' \setminus E$-beli éleket, és az így keletkezett részfákat egészítsük ki feszítőfává 0-ás éleket hozzávéve. Ez lehetséges, mert az eredeti gráf gyengén összefüggő. Így primál megengedett bázist kapunk az eredeti feladatra.



Második fázis: alkalmazzuk ezzel a kiinduló feszítőfával a hálózati szimplex módszert az eredeti célfüggvényre.

Megjegyzés. A hálózati szimplex módszer kis módosítással használható arra az általánosabb feladatra is, ahol minden élre adott egy alsó és egy felső korlát az él értékére. Ez már magában foglalja a minimális költségű maximális folyam és a minimális költségű áram feladatot. Fordítva is igaz, hogy a javító utas minimális költségű folyam algoritmus is használható a jelen fejezetben vizsgált feladat megoldására. A tapasztalatok azonban azt mutatják, hogy a gyakorlatban felmerülő súlyozott folyam feladatoknál sokszor a hálózati szimplex módszer különféle változatai közül kerül ki a leghatékonyabb algoritmus.

1.5.4. Erősen megengedett bázisok

A hálózati szimplex módszer esetében van a Bland szabálynál természetesebb és hatékonyabb pivotálási szabály, ami garantálja az algoritmus végességét. Ehhez azonban be kell vezetni az erősen megengedett bázis fogalmát.

Definíció. Egy primál megengedett B bázis **erősen megengedett**, ha a feszítő fa összes olyan uv élére, amire $\bar{x}_{uv} = 0$, v közelebb van a fában v_0-hoz mint u.

1.5.4. Állítás. *Ha a hálózati feladatnak van megoldása, de nincs erősen megengedett bázisa, akkor szétbontható két részfeladatra.*

Bizonyítás. Adott bázisnál nevezzünk egy uv élt **tiltottnak**, ha $\bar{x}_{uv} = 0$, és u közelebb van a fában v_0-hoz mint v. Vegyük azt a B primál megengedett bázist, ahol irányítatlan értelemben a legtöbb csúcs elérhető v_0-ból nem tiltott élen, és legyen U az elérhető pontok halmaza. Ha U-ba belépne D-nek egy e éle, akkor e-t hozzávéve a bázishoz, és egy U-ból kilépő, $B + e$ körén lévő tiltott élt kihagyva olyan bázist kapnánk, ahol U-nál nagyobb halmaz érhető el v_0-ból nem tiltott élen. Mivel ez nem lehet, U-ba nem lép be él D-ben, és az összes kilépő élre $\bar{x}_e = 0$. Ez viszont azt jelenti, hogy teszőleges x megengedett megoldásban $x_e = 0$ az összes U-ból kilépő élen, tehát a feladat szétbontható a $D[U]$ és a $D[V \setminus U]$ digráfokon értelmezett feladatra. $\square$

Tegyük fel tehát, hogy kiindulásként adott egy erősen megengedett bázis. A primál hálózati szimplex módszer lépését a következőképpen változtatjuk meg. Tegyük fel, hogy $e_p = uv$ lép be a bázisba, és C a keletkező kör. Legyen v_C a kör v_0-hoz legközelebbi pontja. Legyen $e_q = u'v'$ az a hátra-él, amin $\bar{x}_q$ minimális, és ezek közül az, ami v_C-től előre-irányba elindulva a legutolsó a körön.

1.5.5. Állítás. *A $B' = B + \{p\} - \{q\}$ bázis erősen megengedett.*

Bizonyítás. Két esetet különböztetünk meg.

1. eset: $\bar{x}_q = 0$. Ekkor, mivel B erősen megengedett, e_q a $v_C u$ szakaszon van, és ezen a szakaszon az utolsó olyan hátra-él, amire $\bar{x}_e = 0$. Ezért a báziscsere után sem keletkezik tiltott él, hiszen az csak az $u'u$ szakaszon lévő 0-ás hátra-élekből keletkezhetne.

2. eset: $\bar{x}_q > 0$. Ekkor a körön lévő 0-ás élek a báziscsere után már nem 0-ásak, tehát csak olyan e hátra-él válhat tiltottá, amire $\bar{x}_e = \bar{x}_q$. Az ilyen hátra-élek a $v_C v'$ szakaszon vannak. Ha e_q a $v_C u$ szakaszon van, akkor ezek az élek a báziscsere után is v_0 fele mutatnak. Ha pedig e_q a $v v_C$ szakaszon van, akkor a báziscsere során pontosan akkor fordulnak meg, ha előtte nem v_0 fele mutattak, tehát a báziscsere után v_0 fele mutatnak. $\square$

Most belátjuk, hogy ezzel a választással nem lehet ciklizálás.

1.5.6. Állítás. *Ha a fenti báziscsere során $\bar{x}_q = 0$, akkor $\sum_{v \in V} \bar{y}_v$ szigorúan csökken.*

Bizonyítás. Ebben az esetben e_q a $v_C u$ szakaszon van, tehát $\bar{y}_v$ az $u'u$ szakasz pontjaiban (és a belőlük induló részfákon) változik, mégpedig $\bar{c}_p$-vel nő, azaz szigorúan csökken. $\square$

Mivel $\sum_{v \in V} \bar{y}_v$ szigorúan csökken ha a célfüggvényérték nem nő, az algoritmus során nem térhetünk vissza ugyanahhoz a bázishoz, tehát nem lehet ciklizálás.

2. fejezet

Egészértékű lineáris programozás

2.1. Bevezetés

A következő alapfeladattal foglalkozunk:

$$\begin{aligned} &\max cx \\ &Ax \leq b \\ &x \in \mathbb{Z}^n \end{aligned}$$

A fenti feladat LP-relaxáltja:

$$\begin{aligned} &\max cx \\ &Ax \leq b \\ &x \in \mathbb{R}^n \end{aligned}$$

Speciális eset: bináris programozási feladat:

$$\max\{cx : Ax \leq b, x \in \{0,1\}^n\}$$

Példa: Bináris hátizsák feladat

Adott n db tárgy. A j-edik tárgy értéke legyen $c_j \geq 0$, súlya pedig $a_j > 0$. Adott továbbá egy hátizsák, melynek teherbíró képessége $b > 0$. A lehető legnagyobb összértékű tárgyat akarunk a hátizsákba pakolni úgy, hogy az még elbírja őket. Azaz:

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

A feladat NP-nehéz, tehát nem várható rá polinom idejű algoritmus. Az LP relaxáltja könnyen megoldható: az érték/súly arány szerint helyezzük csökkenő sorrendbe a tárgyakat. Sorra tegyük be őket a hátizsákba, amíg a hátizsák be nem telik (az utolsó betett tárgy lehet hogy csak részben fér be, de a relaxált feladatnál ez nem baj).

2.1.1. Állítás. *Ez a mohó algoritmus optimálisan megoldja az LP relaxáltat.*

Bizonyítás. Az algoritmus teljesen megtölti a hátizsákot. Mivel érték/súly arány szerinti csökkenő sorrendben vettük a tárgyakat, nyilvánvaló, hogy minden olyan tárgynak, ami (egészen vagy részben) kimaradt, legfeljebb annyi az érték/súly aránya, mint bárminek, ami (egészen vagy részben) bekerült. Ezért semmilyen bent lévő rész kicserélésésével nem járhatunk jobban, tehát a megoldás optimális. □

Az egészértékű programozási feladatnál kicsit általánosabb a **vegyes programozási feladat**, ahol nem feltétlenül az összes változónak kell egészértékűnek lenni:

$$\max\{cx + dz : \ Ax + Bz \leq b, x \in \mathbb{Z}^{n_1}, z \in \mathbb{R}^{n_2}\}.$$

Példa: Szolgáltató-elhelyezési feladat

Adott m db ügyfél, és n db lehetséges szolgáltatóhely. Minden ügyfélnek egységnyi igénye van, ezt megoszthatjuk több szolgáltatóhely között.

c_{ij}: i-edik ügyfél kiszolgálásának költsége a j-edik szolgáltatóhelyről.

f_j: j-edik szolgáltatóhely megnyitásának költsége

u_j: j-edik szolgáltatóhely kapacitása

Ez a feladat is NP-nehéz. Vegyes programozási feladatként úgy tudjuk felírni, hogy minden szolgáltatóhelyhez bevezetünk egy bináris változót (y_j), ami azt „dönti el”, hogy megnyitjuk-e a szolgáltatóhelyet:

$$\begin{aligned}
\min \sum_{j=1}^{n}(f_j y_j + \sum_{i=1}^{m} c_{ij} x_{ij}) & \\
0 \leq x_{ij} \leq y_j & \qquad i \in \{1,\ldots,m\},\ j \in \{1,\ldots,n\} \\
y_j \in \{0,1\} & \qquad j \in \{1,\ldots,n\} \\
\sum_{j=1}^{n} x_{ij} = 1 & \qquad i \in \{1,\ldots,m\} \\
\sum_{i=1}^{m} x_{ij} \leq u_j y_j & \qquad j \in \{1,\ldots,n\}.
\end{aligned}$$

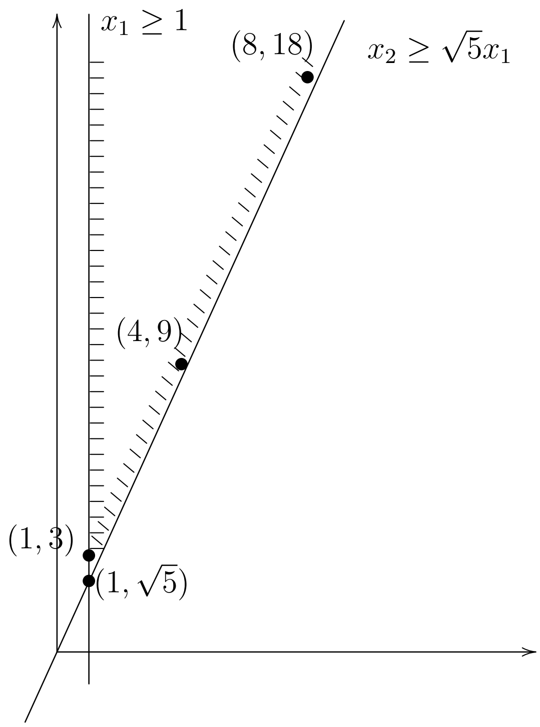
Észrevétel. *Ha rögzítjük, hogy melyik szolgáltatóhelyeket nyitjuk meg, akkor az optimális ügyfél-hozzárendelés feladatában a feltételi mátrix TU, tehát egész kapacitások esetén a szolgáltató-elhelyezési feladatnak van olyan optimális megoldása, ahol minden ügyfelet 1 szolgáltatóhoz rendelünk.*

Ha egy egészértékű programozási feladatot meg akarunk oldani, akkor tulajdonképpen egy poliéder egész pontjainak konvex burkán akarunk egy lineáris célfüggvényt maximalizálni.

Definíció. Legyen P poliéder, $P \subseteq \mathbb{R}^n$. Ekkor $P_I = \operatorname{conv}(P \cap \mathbb{Z}^n)$ a P egész pontjainak konvex burka.

Megjegyzés. Ha P leírásában irracionális együtthatók is szerepelhetnek, akkor az egész pontok konvex burka nem lesz mindig poliéder. Például 2-dimenzióban: Legyen $P = \{(x_1, x_2) : x_1 \geq 1, x_2 \geq \sqrt{5}x_1\}$. Ekkor az egész pontok konvex burkának végtelen sok csúcsa lesz. Például: $(1, 3)$, $(4, 9)$, $(8, 18)$, stb.

Az irracionális meredekségű egyeneshez tetszőlegesen közel tudunk egész pontot találni. $(4, 9)$ közelebb van, mint $(1, 3)$, de $(8, 18)$ már közelebb van, mint $(4, 9)$, és így tovább, egyre közelebb kerülünk, és közben mindig új csúcsokat definiálunk, ezáltal végtelen sok csúcsot kapunk. Márpedig egy poliédernek csak véges sok csúcsa lehet.



2.1.2. Tétel (Meyer). *Ha P racionális poliéder (azaz megadható racionális együtthatós lineáris egyenlőtlenségrendszerrel), akkor P_I poliéder.*

Bizonyítás. Ha P korlátos, akkor P_I véges sok pont konvex burka, tehát poliéder. Az tehát az érdekes eset, amikor P nem korlátos. Ekkor Motzkin tétel szerint $P = Q + C$, ahol Q egy racionális politop (korlátos poliéder), és C egy végesen generált racionális kúp. A C kúpot véges sok egész vektor is generálja, legyenek ezek $g_1, \ldots, g_k$. Tekintsük a következő ponthalmazt, ami politóp, hiszen k darab szakasz vektor-összege:

$$D = \left\{ \sum_{i=1}^{k} \lambda_i g_i : \; 0 \leq \lambda_i \leq 1 \; (i = 1, \ldots, k) \right\}.$$

Belátjuk, hogy $P_I = (Q+D)_I + C$. Ez elég a tétel bizonyításához, hiszen $Q+D$ korlátos, tehát $(Q+D)_I$ poliéder, és így $(Q+D)_I + C$ is.

Egyik irány: $P_I \subseteq (Q+D)_I + C$. Mivel $(Q+D)_I + C$ konvex halmaz, elég belátni, hogy P minden egész p pontja benne van. Legyen tehát $p \in P \cap \mathbb{Z}^n$.

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

$$p = q + c = q + \sum_{i=1}^{k} \mu_i g_i = (q + \sum_{i=1}^{k} (\mu_i - \lfloor \mu_i \rfloor) g_i) + \sum_{i=1}^{k} \lfloor \mu_i \rfloor g_i =: (q+d) + c'.$$

Itt $d \in D$ és $c' \in C$ nyilvánvalóan teljesül, továbbá $q + d$ egész, mivel $q + d = p - c'$, azaz előáll két egész vektor különbségeként. Így p tényleg benne van $(Q+D)_I + C$-ben.

Másik irány: $P_I \supseteq (Q+D)_I + C$. Nyilván $Q + D \subseteq P$, így

$$(Q+D)_I + C \subseteq P_I + C = P_I + C_I.$$

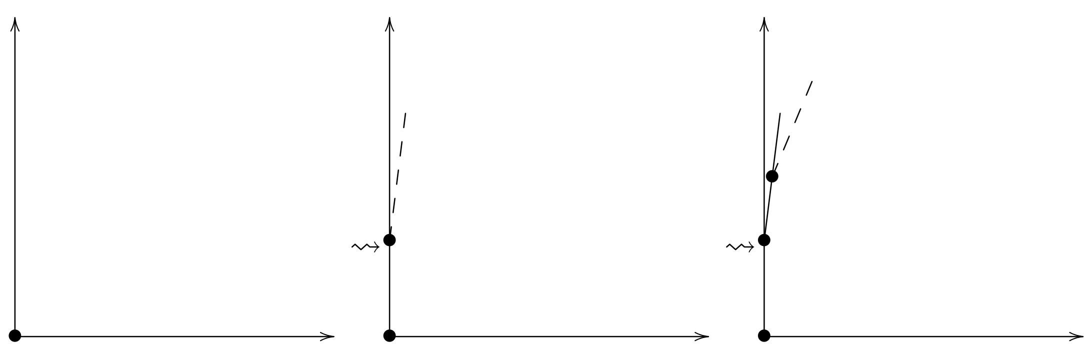
Tetszőleges X és Y halmazokra igaz, hogy $\mathrm{conv}(X) + \mathrm{conv}(Y) \subseteq \mathrm{conv}(X+Y)$. Ezt használva:

$$P_I + C_I \subseteq \mathrm{conv}((P \cap \mathbb{Z}^n) + (C \cap \mathbb{Z}^n)) \subseteq (P + C)_I = P_I.$$

□

Megjegyzés. Ugyan P_I is poliéder, de lehet, hogy sokkal bonyolultabb, mint P, sokkal több csúcsa van. Például induljunk ki a következő kúpból: $\{x_1 \geq 0, x_2 \geq 0\}$. Itt az origó benne van P_I-ben. Válasszunk az $x_2 = 0$ egyenesen tetszőleges egész v_1 pontot, majd „forgassuk el" az $x_1 \geq 0$ félsíkot a v_1 körül úgy, hogy az origó benne legyen, de az újonnan bekerült részben ne legyenek egész pontok, és a meredekség racionális maradjon. Keressünk egy újabb egész pontot az új határoló egyenesen, és folytassuk az eljárást.

Ezzel az eljárással tetszőleges N pozitív egész számra le tudunk gyártani olyan egy-csúcsú poliédert, ahol az egész pontok konvex burkának N csúcsa van, és mindezt már 2 dimenzióban is megtehetjük.



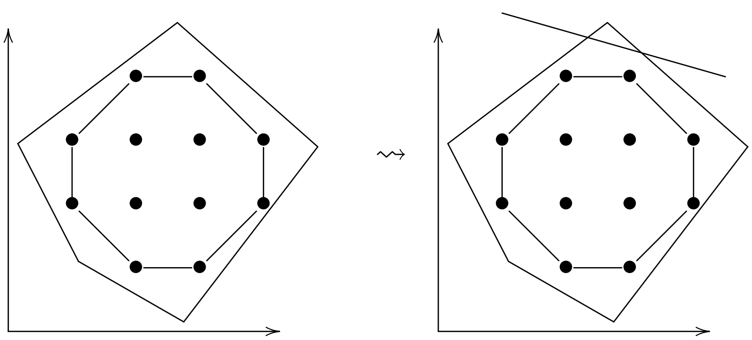
A továbbiakban olyan algoritmusokat ismertetünk, amikkel egészértékű programozási feladatokat lehet többé-kevésbé hatékonyan megoldani.

2.2. Vágósíkos eljárás

Célunk a $\max\{cx : Ax \leq b, x \in \mathbb{Z}^n\}$ optimalizálási feladat megoldása. Legyen $P = \{x \in \mathbb{R}^n : Ax \leq b\}$.

Általános vágósíkos algoritmus

1. Oldjuk meg a $\max\{cx : x \in P\}$ feladatot (például szimplex módszerrel)
2. Ha az x^* optimális megoldás egész, akkor kész vagyunk.
3. Ha x^* nem egész, akkor keressünk olyan $\alpha x \leq \beta$ egyenlőtlenséget, amire $\alpha x^* > \beta$, de $\alpha x \leq \beta$ $\forall x \in P \cap \mathbb{Z}^n$
4. Legyen $P := P \cap \{x : \alpha x \leq \beta\}$, és lépjünk az 1. pontra.



Megjegyzés. A fenti algoritmus ebben a formában nem feltétlenül véges, még két dimenzióban sem. A másik probléma, hogy hogyan találunk az algoritmus 3. pontjában egy megfelelő $\alpha x \leq \beta$ egyenlőtlenséget. Erre ad megoldást (kicsit más alakú feladatnál) Gomory módszere.

2.2.1. Gomory-vágás

Legyen a feladat $\max\{cx : Ax = b, x \geq 0, x \in \mathbb{Z}^n\}$. Tudjuk, hogy $\max\{cx : Ax = b, x \geq 0\}$ megoldható szimplex módszerrel. Legyen B az így kapott optimális bázis. Használjuk a korábbi jelölést: $\bar{x}$, $\bar{A}$, $\bar{b}$.

Tegyük fel, hogy $\bar{x}$ nem egész: $\bar{x}_q \notin \mathbb{Z}$ valamilyen $q \in B$ indexre. Tudjuk, hogy ha x_q az r-edik sorhoz tartozó bázisváltozó, akkor $\bar{x}_q = \bar{b}_r$. Az r-edik sor egyenlete az $\bar{A}x = \bar{b}$ egyenletrendszerben:

$$x_q + \sum_{j \in N} \bar{a}_{rj} x_j = \bar{b}_r \quad \text{– érvényes egyenlet minden megoldásra.}$$

Az egyenletből kaphatunk egy érvényes egyenlőtlenséget, kihasználva, hogy minden megoldás nemnegatív:

$$x_q + \sum_{j \in N} \lfloor \bar{a}_{rj} \rfloor x_j \le \bar{b}_r \quad \text{– érvényes egyenlőtlenség minden megoldásra.}$$

Egészértékű megoldás esetén a baloldal egész, tehát a jobboldalt is kerekíthetjük:

$$x_q + \sum_{j \in N} \lfloor \bar{a}_{rj} \rfloor x_j \le \lfloor \bar{b}_r \rfloor \quad \text{– érvényes egyenlőtlenség minden egész megoldásra.}$$

Észrevétel. *Az $\bar{x}$ megoldásra nem teljesül az egyenlőtlenség, mert $\bar{x}_q = \bar{b}_r > \lfloor \bar{b}_r \rfloor$ és $\bar{x}_j = 0$ $\forall j \in N$. Tehát kaptunk egy olyan egyenlőtlenséget, ami minden egész megoldásra érvényes, de az $\bar{x}$ megoldásra nem, így ezt használhatjuk a vágósíkos algoritmus 3. pontjában.*

Megjegyzés. Megfelelő megvalósítás esetén a Gomory vágásokkal véges algoritmust kapunk. Azonban az algoritmus még így is lehet nagyon lassú.

Egy lehetséges gyorsítás, ha olyan vágást próbálunk találni, ami a „lehető legjobb” olyan értelemben, hogy az egész pontok konvex burkának egy lapját határozza meg. A probléma az, hogy ezt a konvex burkot nem ismerjük, így a lapjait sem tudjuk meghatározni. Ez azonban nem zárja ki, hogy néhány lapját a feladat struktúrájából adódóan ismerjük, és azokkal próbálkozzunk.

Tehát ha ismerjük a konvex burok lapjainak egy részét, akkor amint az LP-re optimális megoldást találunk, megnézzük, hogy van-e a lapok között olyan, amit a talált optimum nem teljesít (ez persze nehéznek tűnik, ha exponenciálisan sok lapot ismerünk, de néha még akkor is megoldható!) Ha találunk ilyet, akkor ezt a lapot hozzávesszük a feladathoz, és ismételjük az eljárást.

Bináris hátizsák-feladat

Nézzünk egy bináris hátizsák-feladatot, ahol $b \ge a_1 \ge a_2 \ge \cdots \ge a_n$. Vegyünk egy olyan $\{j_1, \ldots, j_k\}$ indexhalmazt ($j_1 \le j_2 \le \cdots \le j_k$), amire a következő négy feltétel teljesül:

1. $\sum_{i=1}^{k} a_{j_i} > b$,
2. $\sum_{i=1}^{k-1} a_{j_i} \le b$,
3. Ha $j_1 \ne 1$, akkor $a_1 + \sum_{i=3}^{k} a_{j_i} \le b$,
4. Ha $j > j_1$ és $j \notin \{j_1, \ldots, j_k\}$, akkor $a_j + \sum_{i=2}^{k} a_{j_i} \le b$.

Feladat. *Mutassuk meg, hogy ha nem fér be az összes tárgy a hátizsákba, akkor mindig létezik ilyen indexhalmaz, és a $\sum_{j<j_1} x_j + \sum_{i=1}^{k} x_{j_i} \le k-1$ egyenlőtlenség az egész megoldások konvex burkának lapját határozza meg. (Tipp: próbáljunk n darab affin független megoldást mutatni, ami ezt az egyenlőtlenséget egyenlőséggel teljesíti!)*

Utazó ügynök feladat

Adott n város, és bármely kettő között a távolság. Az utazó ügynök szeretné a legrövidebb olyan körsétát megtalálni, ami minden várost pontosan egyszer érint. Megengedett megoldásai tehát az n pontú teljes gráf Hamilton körei, és ezek közül szeretné a legrövidebbet kiválasztani. Jelölje $c(uv)$ az u és v városok távolságát, és legyen V az összes város halmaza.

Első próbálkozásként tekintsük a következő bináris feladatot:

$$\begin{aligned} \min cx& \\ d_x(v) = 2& \quad \forall v \in V \\ x \in \{0,1\}^{\binom{n}{2}}& \end{aligned}$$

A Hamilton-körök ugyan megoldások, de sajnos több diszjunkt kör uniója is lehet megoldás, ha együtt az összes várost tartalmazzák. Ezért még hozzá kell venni az összefüggőséget biztosító egyenlőtlenségeket:

$$d_x(U) \geq 2 \quad \forall U \subsetneq V,\ U \neq \emptyset.$$

Ennek a megoldásai már kizárólag a Hamilton-körök. A feladat LP relaxáltja:

$$\min cx \tag{2.1}$$
$$d_x(v) = 2 \quad \forall v \in V \tag{2.2}$$
$$0 \leq x \leq 1 \tag{2.3}$$
$$d_x(U) \geq 2 \quad \forall U \subsetneq V,\ U \neq \emptyset. \tag{2.4}$$

Legyen P a (2.2), (2.3), (2.4) egyenlőtlenségeket kielégítő vektorok halmaza. Ekkor P_I a Hamilton körök karakterisztikus vektorainak a konvex burka.

Megjegyzés. A poliéder nem függ a súlyfüggvénytől, minden n-re egy poliéderünk van. A (2.2) egyenletek egy olyan hipersíkot határoznak meg, ami tartalmazza a P poliédert.

Ha a vágósíkos eljárást akarjuk alkalmazni, már az elején egy problémába ütközünk: a (2.4) típusú egyenlőtlenségekből exponenciálisan sok van, ezért a szimplex módszerben exponenciális méretű mátrixokkal kellene dolgozni, ami túl lassú. Ezért már az LP relaxáció megoldásához is egy vágósíkos eljárást használunk, a következőképpen:

Keressük meg az optimális olyan vektort, ami a (2.2), (2.3) feltételeket teljesíti. Legyen ez x^*.

Feladat. *Mutassuk meg, hogy $n-1$ maximális folyam kereséssel eldönthető, hogy x^* teljesíti-e az összes (2.4) típusú egyenlőtleséget, és ha nem, található egy olyan, amit nem teljesít.*

A feladat alapján el tudjuk dönteni, hogy x^* optimális megoldása-e az LP relaxációnak, és ha nem, találunk egy egyenlőtlenséget, amit nem teljesít. Ezt hozzávéve az eddigiekhez újra megoldjuk az LP-t, és ezt addig folytatjuk, amíg a (2.4) feltételek nem teljesülnek.

Megjegyzés. Ha a végén kapott megoldás nem egész, akkor alkalmazhatunk Gomory vágást, vagy korlátozás és szétválasztást (lásd később).

2.3. Dinamikus programozási algoritmusok

Dinamikus programozást olyan feldatok megoldásánál használhatunk, ahol a megoldás előállítható egyszerűbb részfeladatok megoldása segítségével. A dinamikus programozás lényege, hogy egy bizonyos részfeladatot csak egyszer oldunk meg, és a megoldást tároljuk, így ezt a megoldást a nagyobb részfeladatok megoldásához már további számolás nélkül használhatjuk.

2.3.1. Bináris hátizsákfeladat

Legyen $k \in \{1, \ldots, n\}$, $d \in \{0, \ldots, b\}$. Definiáljuk a következő függvényt:

$$f_k(d) = \max\{\sum_{j=1}^{k} c_j x_j : \sum_{j=1}^{k} a_j x_j \leq d,\ x \in \{0,1\}^k\}.$$

Látjuk, hogy $f_n(b)$ lesz az eredeti bináris hátizsákfeladat optimumértéke.

Legyen $f_0(d) = 0$ minden $d \in \{0, \ldots, b\}$ értékre. Írjunk fel rekurzív képletet $f_k(d)$ értékére, ha $k \geq 1$:

$$f_k(d) = \begin{cases} f_{k-1}(d), \text{ ha } a_k > d \\ \max\{f_{k-1}(d), f_{k-1}(d-a_k) + c_k\}, \text{ ha } a_k \leq d \end{cases}$$

Algoritmus

```
for  k = 1...n
     for  d = 0...b
     do { számoljuk ki fk(d) -t a fenti képlettel}
```

Ez a dinamikus programozási algoritmus $\mathcal{O}(nb)$ lépésben kiszámolja az optimumértéket. Megjegyzendő, hogy ez nem polinomiális futási idő, hiszen az input mérete $\mathcal{O}(n \log b)$.

Az optimális megoldást a következőképpen kapjuk:

$$x_n := \begin{cases} 0, \text{ ha } f_n(b) = f_{n-1}(b) & (1) \\ 1, \text{ ha } f_n(b) = f_{n-1}(b-a_n) + c_n & (2) \end{cases}$$

$x_{n-1}, \ldots, x_1$-et pedig esetszétválasztással kapjuk:

- Ha az (1) eset következik be, akkor

$$x_{n-1} := \begin{cases} 0, \text{ ha } f_{n-1}(b) = f_{n-2}(b) \\ 1, \text{ ha } f_{n-1}(b) = f_{n-2}(b - a_{n-1}) + c_{n-1} \end{cases}$$

- Ha a (2) eset következik be, akkor

$$x_{n-1} := \begin{cases} 0, \text{ ha } f_{n-1}(b - a_n) = f_{n-2}(b - a_n) \\ 1, \text{ ha } f_{n-1}(b - a_n) = f_{n-2}(b - a_n - a_{n-1}) + c_{n-1} \end{cases}$$

Így folytatjuk, amíg x_1-et el nem érjük.

2.3.2. Nemnegatív mátrixú egészértékű feladat

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

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

ahol A, b, c nemnegatív és egész.

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

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

Definiáljuk ezen kívül $f_0(d)$-t 0-nak minden $0 \leq d \leq b$-re. A következő rekurzióval lehet $k \geq 1$-re és $0 \leq d \leq b$-re kiszámolni a függvényértékeket (fontos eltérés a bináris hátizsákfeladathoz képest, hogy a maximumban $f_k(d - a_{.k})$ szerepel):

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

Az $f_k(d)$ értékeket k szerint növekvő sorrendben, azon belül d szerint lexikografikusan növekvő sorrendben számoljuk ki. 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, de ha b kicsi, akkor használható.

2.4. Korlátozás és szétválasztás

Tekintsük a következő alakú feladatot:

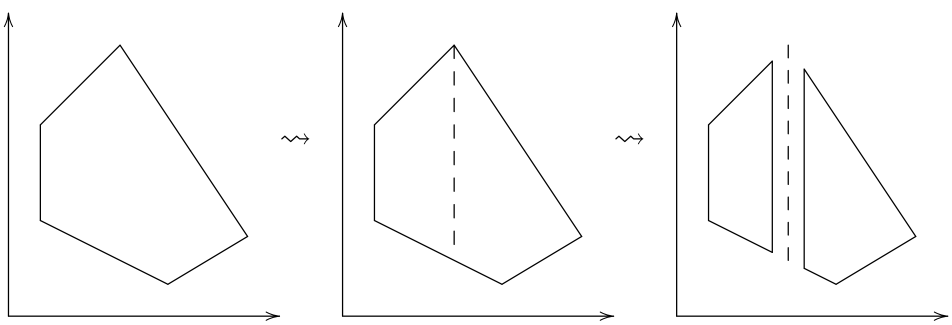
$$\max cx \tag{2.5}$$
$$Ax \le b \tag{2.6}$$
$$x \text{ egész}, \tag{2.7}$$

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.

Az algoritmus működése:



A részfeladatokra osztás során kihagyunk részeket a feladat teréből, de csak olyanokat, amikben nincsenek egész pontok. A továbbiakban a legegyszerűbb LP alapú általános algoritmust írjuk le. Legyen $P = \{x \in \mathbb{R}^n : Ax \le b\}$. A részfeladataink mindig $\max\{cx : x \in P' \cap \mathbb{Z}^n\}$ alakúak lesznek, ahol $P' \subseteq P$ poliéder. Jelölések:

- $u(P') = \max\{cx : x \in P'\}$. Ez felső korlát a részfeladat optimumértékére.
- $x^*(P')$ a $\max\{cx : x \in P'\}$ feladat egy optimális bázismegoldása. Ez kiszámolható szimplex módszerrel.
- x^*: az eddig talált legjobb egész megoldás
- L: az eddig talált legjobb egész megoldás célfüggvényértéke, illetve $-\infty$ ha még nem találtunk egész megoldást.

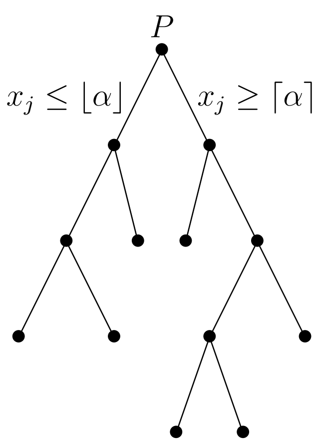
- $\mathcal{F}$: A feldolgozandó poliéderek listája.

Az algoritmus inicializálása: $\mathcal{F} = \{P\}$, és $L = -\infty$. Egy általános lépés a következő:

1. Ha $\mathcal{F}$ üres: kész vagyunk. Különben az $\mathcal{F}$ listáról kiválasztunk egy P' feladatot, és kiszámoljuk az $u(P')$ értéket.
2. Ha $u(P')$ nem létezik, mert az LP relaxált nem megoldható: töröljük P'-t $\mathcal{F}$-ből.
3. Ha $u(P') \leq L$: töröljük P'-t $\mathcal{F}$-ből.
4. Ha $u(P') > L$ és $x^*(P')$ egész: $x^* := x^*(P')$, és $L := u(P')$. Töröljük P'-t $\mathcal{F}$-ből.
5. Ha $u(P') > L$ és $x^*(P')$ nem egész: válasszunk egy olyan j indexet, amire $x^*(P')$ j-edik komponense egy nem egész α szám. Legyen $P'_1 = P' \cap \{x \in \mathbb{R}^n : \; x_j \leq \lfloor \alpha \rfloor\}$, és $P'_2 = P' \cap \{x \in \mathbb{R}^n : \; x_j \geq \lceil \alpha \rceil\}$. Töröljük P'-t $\mathcal{F}$-ből, és adjuk hozzá P'_1-et és P'_2-t $\mathcal{F}$-hez.

2.4.1. Állítás. *Ha P korlátos, akkor az algoritmus véges sok lépésben talál egy optimális megoldást.*

Bizonyítás. Mivel P korlátos, létezik olyan N szám, hogy tetszőleges $x \in P$-re és $j \in \{1, \ldots, n\}$-re $|x_j| \leq N$. Az algoritmus futása alapján felépíthetünk egy gyökeres bináris fát, amelynek minden csúcsa egy feldolgozott poliédernek felel meg (a gyökér P-nek) és az egyes elágazások a szétválasztásoknak.



Tekintsünk a fában egy tetszőleges utat a gyökérből egy levélbe! Egy adott x_j változó szerint ezen az úton legfeljebb $2N$ szétválasztás lehet, mert minden szétválasztásnál egy 1 hosszú nyílt intervallum kimarad az x_j lehetséges értékei közül. Így minden ilyen út legfeljebb $2nN$ hosszú, amiből következik, hogy az algoritmus véges sok lépésben véget ér.

Az optimális megoldás P-ben van, és ha az algoritmus egy adott szétválasztási lépésénél P'-ben ott van, akkor vagy P'_1-ben vagy P'_2-ben is ott van. Tehát a fenti fa valamelyik leveléhez tartozó poliéderben van az optimális megoldás. Mivel egy levélben nem választunk szét, ezért ott meg is találjuk. Tehát az algoritmus mindenképpen megtalálja az optimális megoldást. $\square$

Korlátozás és szétválasztás algoritmusa a bináris hátizsák feladatra

Ha bináris hátizsákfeladatra futtatjuk a fenti algoritmust, az algoritmusban szereplő poliéderek mindig a következő alakúak lesznek:

$$P(J_0, J_1) = \{0 \leq x \leq 1 : \sum_{j=1}^{n} a_j x_j \leq b,\ x_j = 0 \text{ ha } j \in J_0,\ x_j = 1 \text{ ha } j \in J_1\}.$$

A $\max\{cx : x \in P(J_0, J_1)\}$ feladat nagyon egyszerűen megoldható a mohó algoritmussal:

- Rakjuk a nem $(J_0 \cup J_1)$-beli változókat $\frac{c_j}{a_j}$ szerint csökkenő sorrendbe.
- Az elején $d = b - \sum_{j \in J_1} a_j$ - ennyi még belefér a hátizsákba.
- Ha j_1 az első index a fenti sorrendben, akkor legyen $x_{j_1} = \min\{1, \frac{d}{a_{j_1}}\}$, és legyen $d := d - a_{j_1} x_{j_1}$. Majd vegyük a következőt a sorrend szerint, és ugyanígy folytassuk.

Megjegyzés. Ha egy változó értéke nem 1, hanem $\frac{d}{a_j}$, akkor az utána következő összes többi változó értéke 0 lesz, hiszen a hátizsák megtelt. Tehát legfeljebb egyetlen nem egész érték lesz. Ezért mindig egyértelmű, melyik változó szerint kell szétválasztani.

Az algoritmust érdemes kiegészíteni azzal, hogy ha egy adott lépésben az x_j változó szerint választunk szét, akkor az optimális tört megoldásban x_j-t 0-ra módosítva egy megengedett egész megoldást kapunk. Ha ez jobb, mint az eddig talált legjobb x^* megoldás, akkor lecserélhetjük x^*-ot erre.

2.5. Közelítő algoritmusok

2.5.1. Minimális lefogó csúcshalmaz

Adott egy $G = (V, E)$ irányítatlan gráf. Csúcsok egy $U \subseteq V$ halmaza **lefogó csúcshalmaz**, ha minden élnek legalább egyik végpontját tartalmazza. A minimális méretű lefogó csúcshalmaz megkeresése NP-nehéz feladat. Most két algoritmust ismertetünk lefogó csúcshalmaz keresésére, amik nagyon gyorsak, de nem feltétlenül szolgáltatnak optimális megoldást.

1. Algoritmus (mohó)

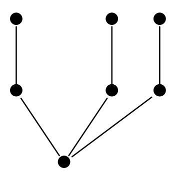
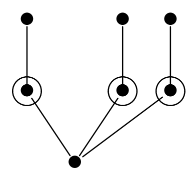
Kezdetben legyen $U = \emptyset$.

1. Válasszunk maximális fokú csúcsot a gráfban. Legyen ez v.

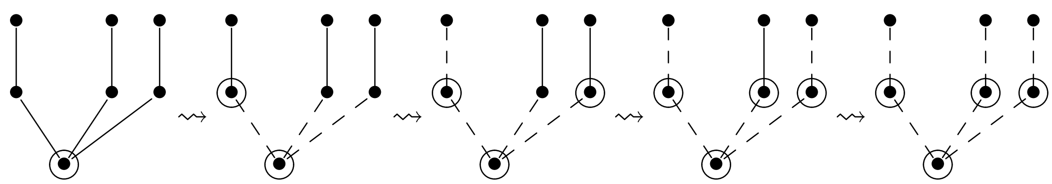
2. Legyen $U := U \cup \{v\}$, és a G gráfból hagyjuk ki a v csúcsot és az összes rá illeszkedő élt.

3. Ha nem marad él, akkor kész vagyunk, U a lefogó csúcshalmaz. Különben lépjünk az 1. pontra.

Megjegyzés. A fenti algoritmus nem feltétlenül talál optimális megoldást. Például:

 optimális megoldása: 

Az algoritmusunk egy lehetséges futása viszont a következő:



2. Algoritmus (párosító)

Kezdetben legyen $U = \emptyset$.

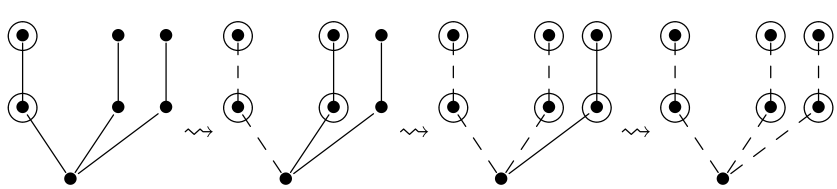
1. Válasszunk egy tetszőleges uv élt a gráfban.

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.

Megjegyzés. A párosító algoritmus nem tűnik túl jó megoldásnak, mivel feleslegesen két csúcsot is bevesz lépésenként a lefogó csúcshalmazba. Az előbbi példára rosszabban teljesít, mint a mohó algoritmus:



Elsőre nem is könnyű olyan példát találni, ahol a párosító algoritmus jobban teljesít, mint a mohó. Azonban megmutatjuk, hogy bizonyos szempontból a párosító számít a jobb algoritmusnak. Ehhez bevezetjük az α-közelítés fogalmát:

Definíció. Egy minimalizálási feladatra egy algoritmus α-**közelítő**, ha tetszőleges inputra az output értéke legfeljebb α-szorosa az optimálisnak.

2.5.1. Állítás. *A párosító algoritmus 2-közelítő.*

Bizonyítás. Legyen F az algoritmus során kiválasztott élek halmaza. F független élekből áll, tehát egy párosítás. Ekkor $|U_{opt}| \geq |F|$, mivel az optimális lefogó csúcshalmaz lefogja F éleit. Másrészt: $|U| = 2|F|$. Tehát $|U_{opt}| \geq \frac{1}{2}|U|$. □

2.5.2. Állítás. *A mohó algoritmus nem 2-közelítő.*

Bizonyítás. Definiálunk egy $G = (A, B; E)$ páros gráfot. Az A osztályban 60 csúcs van, mind 5 fokú. A B osztályban 12 db 5 fokú, 15 db 4 fokú, 20 db 3 fokú, 30 db 2 fokú, és 60 db 1 fokú. Minden $i \in \{1, \dots, 5\}$-re a B-beli i fokúak szomszédságai A egy partícióját adják (hogy pontosan melyik partícióját, az mindegy).

A mohó algoritmus lefuthat úgy, hogy a teljes B halmazt választja ki lefogó csúcshalmaznak. Másrészt A is lefogó csúcshalmaz, tehát mivel $|B| > 2|A|$, ez nem 2-közelítő algoritmus. (Hasonló konstrukcióval az is belátható, hogy a mohó algoritmus semmilyen α-ra nem lesz α-közelítő.) □

Megjegyzés. Érdekes módon nem ismert $\alpha < 2$-re α-közelítő polinomiális futási idejű algoritmus erre a feladatra, tehát ilyen értelemben a párosító algoritmus a „legjobb ismert”. Persze az algoritmus nyilvánvalóan javítható, például ha a végén a felesleges csúcsokat töröljük, de ettől nem kapunk garantáltan 2-nél jobb közelítést.

2.5.3. Tétel (Dinur és Safra, 2005). *Ha $P \neq NP$, akkor semmilyen $\alpha < 1.3607$-re nincs α-közelítő polinomiális algoritmus a minimális lefogó csúcshalmaz feladatra. (Bizonyítás nélkül)*

2.5.2. Minimális költségű lefogó csúcshalmaz

Legyen $G = (V, E)$ irányítatlan gráf, és $c \in \mathbb{R}_+^V$ költségfüggvény. A minimális költségű lefogó csúcshalmaz feladatban olyan U lefogó csúcshalmazt keresünk, amire $\sum_{u \in U} c_u$ minimális.

A következőkben erre a feladatra adunk egy 2-közelítő algoritmust, ami a párosító algoritmus általánosításának tekinthető. Írjuk fel a feladatot egészértékű programozási feladatként:

$$\begin{aligned} \min cx& \\ x_u + x_v \geq 1& \qquad \forall uv \in E \\ x \in \{0,1\}^V& \end{aligned}$$

A feladat LP-relaxáltja:

$$\begin{aligned} \min cx & \\ x_u + x_v \geq 1 & \qquad \forall uv \in E \\ x \in \mathbb{R}_+^V & \end{aligned}$$

Megjegyzés. Azért nem vettük hozzá az $x \leq 1$ feltételt, mert felesleges: az 1-nél nagyobb értékek lecsökkenthetők 1-re a feltételek megsértése nélkül, és a költség csak csökkenhet.

Nézzük az LP-relaxált duálisát:

$$\begin{aligned} \max \sum_{e \in E} y_e & \\ \sum_{u: uv \in E} y_{uv} \leq c_v & \qquad \forall v \in V \\ y \in \mathbb{R}_+^E & \end{aligned}$$

Primál-duál algoritmus

Minden lépésben lesz egy $U \subseteq V$ csúcshalmazunk, 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. Töröljük E'-ből a lefogott éleket.
4. Ha $E' = \emptyset$, akkor kész vagyunk, U a lefogó csúcshalmaz. Különben lépjünk az 1. pontra.

Megjegyzés. Az y_{uv} emelése csak az u-hoz és v-hez tartozó duál egyenlőtlenségek teljesülését befolyásolja, a többit nem. Ezért y végig duál megengedett megoldás marad. Könnyen látható, hogy ha minden csúcs költsége 1, akkor pont a párosító algoritmust kapjuk.

2.5.4. Állítás. *A fenti primál-duál algoritmus 2-közelítő.*

Bizonyítás. Vezessünk be $\pi \in \mathbb{R}_+^V$ segédváltozókat. Kezdetben legyen $\pi \equiv 0$. Az algoritmus során, amikor y_{uv}-t megemeljük δ-val, növeljük meg $\pi(u)$-t és $\pi(v)$-t is δ-val.

Az algoritmus befejezésekor $\sum_{v\in V}\pi(v) = 2\sum_{e\in E} y_e$. Másrészt, amikor v bekerül U-ba, akkor $\pi(v) = c(v)$, és ez később sem változik. Tehát $\sum_{u\in U} c_u \leq \sum_{v\in V}\pi(v)$, amiből

$$\sum_{u\in U} c_u \leq 2\sum_{e\in E} y_e \underbrace{\leq}_{\text{gyenge dualitás}} 2OPT_{LP} \leq 2OPT_{IP}$$

$\square$

Megjegyzés. Valójában többet bizonyítottunk, mint amit a tétel állít: a kapott megoldás költsége legfeljebb kétszerese az LP-relaxált optimum-értékének.

3. fejezet

Játékelmélet

3.1. Bevezető és alapfogalmak

A játékelméleten belül számos különböző megközelítés létezik a „játék” definíciójára, aszerint hogy mit feltételezünk a játékosok tudásáról és viselkedéséről. Ezen az előadáson az úgynevezett stratégiai alakú játékok elméletébe nyújtunk bepillantást. Ezekben a játékokban véges sok játékos egymástól függetlenül választ stratégiát a rendelkezésre álló stratégiák halmazából. Ismerik a többiek lehetőségeit, de nem ismerik a többiek döntését. A játékosok hasznát az összes játékos döntése együtt határozza meg.

A továbbiakban példákon keresztül mutatjuk be az alapfogalmakat.

3.1.1. Fogoly-dilemma

Egy bűnténynek két gyanúsítottja van, de a rendőrség nem tudja eldönteni, hogy melyikük a tettes, nekik kell vallani. Ha az egyik gyanúsított a másikra vall, akkor az előbbi 1 évre kerül börtönbe, a másik viszont 4 évre. Ha egyikük sem vall, akkor mindkettőjüket 2 év börtön várja, ha mindkettejük vallanak, akkor mindkettőjük 3 év börtönbüntetést kap.

A feladatban a két játékos haszna a különféle stratégiák esetén az úgynevezett **hasznossági mátrix** segítségével írható fel. Jelen esetben a hasznosságok negatívak:

1. \ 2.	vall	nem vall
vall	-3 -3	-4 -1
nem vall	-1 -4	-2 -2

Nézzük meg, mi lenne a jó stratégia a játékosok számára? Az első játékosnak mindenképpen jobb, ha vall, bármit is választ a második játékos. Ugyanez mondható el a második játékosról is. Ezt nevezzük **(szigorúan) domináns stratégiának**.

Ha mindkét játékos önző, akkor mindketten vallani fognak, ami az ő szempontjukból a legjobbnak tűnik. De a domináns stratégia rosszabb mindkettejüknek, mint ha egyikük sem vall.

3.1.2. Szennyezési játék

Ebben a játékban n ország mindegyike döntést hoz, hogy keveset vagy sokat költ környezetvédelemre. Ha egy ország sokat költ, az 3 egységgel megnöveli az ország költségét, viszont ha keveset költ, az a szennyezés miatt mindenkit érint – 1-gyel növeli minden ország költségét. Az i. ország költségét felírhatjuk tehát így:

$$i.\ \text{ország költsége} = \begin{cases} 3 + \text{keveset költő országok száma} & \text{ha sokat költ} \\ \text{keveset költő országok száma} & \text{ha keveset költ} \end{cases}$$

Van domináns stratégia?

Tegyük fel, hogy az ország, aminek a stratégiáját el akarjuk dönteni tudja, hogy mit választott a többi ország. Ekkor ha az ország sokat költ, akkor 3-mal növelné a költségét, ha viszont keveset költene, akkor csak 1-gyel.

A domináns stratégia tehát az, hogy mindenki keveset költ környezetvédelemre. Ha háromnál több ország vesz részt a játékban, akkor a domináns stratégia rosszabb, mint ha minden ország sokat költene.

Jelölés. Legyen n a játékosok száma, és legyen S_i az i-edik játékos stratégiáinak halmaza (ezek lehetnek végtelen halmazok is – ha végesek, akkor véges játékról beszélünk). Egy $s = (s_1, s_2, \ldots, s_n)$ vektort, ahol $s_i \in S_i$ $(i = 1, \ldots, n)$, **stratégia-vektornak** nevezünk. Ha $s \in S$, akkor legyen s_{-i} az i-edik játékos kivételével a többiek stratégiája.

Legyen $S = S_1 \times S_2 \times \cdots \times S_n$ a stratégiavektorok halmaza. Az i-edik játékos hasznossági függvénye $u_i : S \to \mathbb{R}$.

Definíció. Az $s \in S$ stratégiavektor **domináns**, ha minden $s' \in S$ és $i \in \{1, 2, \ldots, n\}$ esetén $u_i(s_i, s'_{-i}) \geq u_i(s')$, ahol (s_i, s'_{-i}) jelöli azt, hogy az i-edik játékos az s_i stratégiát választotta, mindenki más pedig az s' szerinti stratégiát.

Az $s \in S$ stratégiavektor **szigorúan domináns**, ha minden $i \in \{1, 2, \ldots, n\}$ és $s' \in S$: $s'_i \neq s_i$ esetén $u_i(s_i, s'_{-i}) > u_i(s')$.

Megjegyzés. Domináns stratégiavektor akár több is lehet egy játéknál, de szigorúan domináns legfeljebb egy lehet. A fogoly-dilemma és a szennyezési játék esetében van szigorúan domináns stratégia.

3.1.3. Vickrey árverés

Az n játékos egy adott tárgyra licitálhat. Az i-edik játékosnak v_i-t ér a tárgy. A licitálás abból áll, hogy mindenki mond egymástól függetlenül egy nemnegatív valós összeget. Formálisan, $S_i = \mathbb{R}_+$ minden i-re. A legnagyobb összeget licitáló kapja meg a tárgyat (döntetlen esetén a kisebb indexű játékos), de csak a második legnagyobb összeget kell kifizetnie.

3.1.1. Állítás. *Az $s_i = v_i$ $(i = 1, \ldots, n)$ domináns stratégiavektor, azaz a Vickrey árverésen mindenkinek érdemes a valódi értéket licitálnia.*

Bizonyítás. Azt kell belátnunk, hogy minden $s' \in S$-re és minden i-re $u_i(s_i, s'_{-i}) \geq u_i(s')$. Két esetet különböztetünk meg.

- Ha $\max\{s'_j : j \neq i\} \geq s_i$, akkor $u_i(s_i, s'_{-i}) = 0$ és $u_i(s') \leq 0$.
- Ha $\max\{s'_j : j \neq i\} < s_i = v_i$, akkor $u_i(s_i, s'_{-i}) = v_i - \max\{s'_j : j \neq i\} > 0$ és $u_i(s') \in \{0, v_i - \max\{s'_j : j \neq i\}\}$.

Tehát mindkét esetben teljesül a dominancia feltétele. □

3.1.4. Közös ló játék

Van n játékosunk, akik ugyanazt a közös erőforrást szeretnék használni. Mindegyik játékos eldöntheti, hogy mennyire szeretné kihasználni az erőforrást, de az erőforrás annál kevésbé hatékony, minél többet használják.

A játékosoknak végtelen sok stratégiájuk van: $S_i = [0, 1]$, $i \in \{1, \ldots, n\}$.

$$u_i(s) = \begin{cases} 0, \text{ ha } \sum_{j=1}^{n} s_j \geq 1 \\ s_i(1 - \sum_{j=1}^{n} s_j) \text{ egyébként} \end{cases}$$

Hogyan válasszunk stratégiát az i-edik játékosnak, ha a többi játékos stratégiája rögzített? Tegyük fel, hogy $\sum_{i \neq j} s_j = t \leq 1$. Ekkor $u_i(s) = s_i(1 - t - s_i)$. Ennek a maximuma: $\frac{1-t}{2}$

Megjegyzés. A fenti feladatban nincs domináns stratégia, mivel tetszőleges $s_i \in S_i$ stratégiához tudunk olyan s' stratégiavektort mondani, amire $u_i(s_i, s'_{-i}) < u_i(s')$.

Beszélhetünk viszont egyensúlyi stratégiavektorról: ilyenkor egyik játékosnak sem érdemes változtatnia a jelenlegi stratégiáján. Ha van olyan s, ahol $s_i = \frac{1 - \sum_{i \neq j} s_j}{2}$ minden i-re, akkor ez egyensúlyi helyzet, mert senki sem akarna változtatni a stratégiáján.

Felírható tehát a következő egyenlőtlenség-rendszer: $s_i + \sum_{j=1}^{n} s_j = 1$ $\forall i$. Összeadva őket azt kapjuk, hogy $(n+1)\sum_{j=1}^{n} s_j = n$, tehát $\sum_{j=1}^{n} s_j = \frac{n}{n+1}$. Visszahelyettesítve, az egyensúly $s_i = \frac{1}{n+1}$, $\forall i = (1, \ldots, n)$.

Van azonban olyan stratégiavektor, aminél minden játékosnak sokkal több a haszna, mint az egyensúlynál. Az egyensúlyi stratégiavektor esetében egy játékos hasznossága

$$u_i(s) = \frac{1}{n+1}(1 - \frac{n}{n+1}) = \frac{1}{(n+1)^2}.$$

Legyen $s' = (\frac{1}{2n}, \frac{1}{2n}, \dots, \frac{1}{2n})$. Ennek hasznossága:

$$u_i(s') = \frac{1}{2n}(1 - \frac{1}{2}) = \frac{1}{4n}.$$

Definíció. Az $s \in S$ stratégiavektor **tiszta Nash-egyensúly**, ha ha minden $i \in \{1, 2, \dots, n\}$ és $s'_i \in S_i$ esetén $u_i(s) \geq u_i(s'_i, s_{-i})$.

3.1.2. Állítás. *Minden domináns stratégiavektor Nash-egyensúly. Ha van szigorúan domináns stratégiavektor, akkor nincs más Nash-egyensúly.*

Bizonyítás. A definícióból látszik, hogy minden domináns stratégiavektor Nash-egyensúly. Legyen s szigorúan domináns stratégia. Tegyük fel, hogy s' Nash-egyensúly. Ekkor ha $s_k \neq s'_k$, akkor $u_k(s_k, s'_{-k}) > u_k(s')$, mert s szigorúan domináns, de ugyanakkor s' Nash-egyensúly, emiatt $u_k(s_k, s'_{-k}) \leq u_k(s')$. Ez ellentmondás, így csak az lehetséges, hogy $s = s'$. □

3.1.5. Fej vagy írás játék

Két játékos a következőképpen játszik: az első játékosnál van egy érme, azt elrejti fejjel vagy írással. A második játékosnak ki kell találnia, hogy az első játékos fejet vagy írást választott. Ha a második játékos eltalálja, akkor 1 pontot kap, és az első játékos 1 pontot veszít, ha nem találja el, akkor fordítva – az első játékos kap 1 pontot, a második pedig veszít egyet. A hasznossági mátrix:

1. \ 2.	fej	írás
fej	1 -1	-1 1
írás	-1 1	1 -1

Definíció (0 összegű játék)**.** Tetszőleges $s \in S$-re $\sum_{i=1}^{n} u_i(s) = 0$. Ilyen például a fej vagy írás játék, de nem ilyen a fogolydilemma vagy a közös ló játék.

Nincs tiszta Nash-egyensúly, mert

- Ha $s_1 = s_2$, akkor az első játékos jobban jár, ha változtat
- Ha $s_1 \neq s_2$, akkor a második játékos jár jobban, ha változtat.

Azonban ha az első játékos $50-50\%$ valószínűséggel választja a fej- vagy írást, a második játékos pedig $50-50\%$ valószínűséggel tippel fej- vagy írásra, akkor teljesül az, hogy egyiküknek sem érdemes változtatnia ezen a stratégiáján. Az ilyen, véletlent is használó stratégiát kevert stratégiának nevezzük.

Definíció. Az i-edik játékosnak egy **kevert stratégiája** egy m_i valószínűségi eloszlás az S_i halmazon. Az $m=(m_1,\dots,m_n)$ egy **kevert stratégiavektor**, ha m_i kevert stratégiája az i-edik játékosnak ($i=1,\dots,n$). Az m_i-ket egymástól függetleneknek tekintve m-re úgy tekinthetünk, mint egy valószínűségi eloszlásra S-en.

Egy kevert stratégia-vektorhoz tartozó hasznosságok: $u_i(m)$ az $u_i(s)$ várható értéke az m valószínűségi eloszlás szerint.

Definíció. Az m kevert stratégiavektor **kevert Nash-egyensúly**, ha minden $i\in\{1,2,\dots,n\}$ és $s_i'\in S_i$ esetén $u_i(m)\geq u_i(s_i',m_{-i})$.

A játékelmélet egyik alaptétele John Nash 1951-es tétele a kevert Nash-egyensúly létezéséről. Ezt csak két játékos esetére fogjuk később bebizonyítani.

3.1.3. Tétel (Nash, 1951)**.** *Véges sok játékos és véges stratégia-halmazok esetén mindig van kevert Nash-egyensúly.*

Nézzünk példát olyan játékra, ahol nincs kevert Nash-egyensúly:

Árazási játék

Van két eladó, akik ugyanazt a terméket árulják, de meghatározhatják a termék árát: $s_1\in[0,1]$, $s_2\in[0,1]$. Továbbá van három vevő: A, B, C, akik a következőképpen döntenek:

- A csak az első eladótól hajlandó vásárolni
- B az olcsóbb árú terméket vásárolja meg (ha a két eladónál ugyanaz az ár, akkor az elsőtől vásárol)
- C csak a második eladótól hajlandó vásárolni.

Feladat. *Mustassuk meg, hogy az árazási játékban nincs kevert Nash-egyensúly.*

A továbbiakban csak olyan játékokkal foglalkozunk, ahol minden játékosnak véges sok stratégiája van. Ekkor a kevert stratégia definíciója is egyszerűsíthető.

Definíció. Az i-edik játékos kevert stratégiája egy olyan $m_i: S_i\to[0,1]$ függvény, amire $\sum_{x_i\in S_i} m_i(x_i)=1$.

Jelölés. Legyen az i-edik játékos kevert stratégiáinak halmaza: M_i, és jelöljük M-mel a kevert stratégiavektorok halmazát. Ezeket eloszlásként is tekintjük, azaz $s \in S$-re és $m \in M$-re legyen $m(s) = \prod_{i=1}^{n} m_i(s_i)$.

3.1.4. Állítás. *$\sum_{s\in S} m(s) = 1$, azaz $m \in M$ tényleg eloszlást ad S-en.*

Bizonyítás. Játékosok számára vett indukcióval:

$$\sum_{s\in S} m(s) = \sum_{s_n\in S_n} \sum_{s_{-n}\in S_{-n}} m(s_n, s_{-n}) = \sum_{s_n\in S_n} m_n(s_n) \sum_{s_{-n}\in S_{-n}} m_{-n}(s_{-n})$$

$$\underbrace{=}_{\text{indukció}} \sum_{s_n\in S_n} m_n(s_n) \underbrace{=}_{\text{definició}} 1.$$

□

Definíció. Az i-edik játékos **várható haszna** $m \in M$ kevert stratégia esetén:

$$u_i(m) := \sum_{s\in S} m(s)u_i(s) = \sum_{s_i\in S_i} m(s_i)u_i(s_i, m_{-i}).$$

Legyen $m \in M$. Az m_i kevert stratégia az i-edik játékos **legjobb válasza** m_{-i}-re, ha tetszőleges $m_i' \in M_i$ esetén $u_i(m) \geq u_i(m_i', m_{-i})$.

Példa. Nézzük a fej vagy írás játékot. Hasznossági mátrixa:

1. \ 2.	fej	írás
fej	1 -1	-1 1
írás	-1 1	1 -1

Legyen a kevert stratégia a következő:

$$\begin{array}{ll} m_1(\text{fej}) = \frac{1}{3} & m_1(\text{írás}) = \frac{2}{3} \\ m_2(\text{fej}) = \frac{3}{4} & m_2(\text{írás}) = \frac{1}{4} \end{array}$$

$(m_1, m_2) \in M$ kevert stratégia-vektor. Összesen négy stratégia-vektorunk van:

$$S = \{(\text{fej}, \text{fej}), (\text{fej}, \text{írás}), (\text{írás}, \text{fej}), (\text{írás}, \text{írás})\}.$$

Ezek valószínűségei:

$$m(\text{fej}, \text{fej}) = \frac{1}{3} \cdot \frac{3}{4} = \frac{1}{4}$$
$$m(\text{fej}, \text{írás}) = \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{12}$$
$$m(\text{írás}, \text{fej}) = \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}$$
$$m(\text{írás}, \text{írás}) = \frac{2}{3} \cdot \frac{1}{4} = \frac{1}{6}$$

Ha ezeket összeadjuk, az összegük: 1. Most nézzük az első játékos várható hozamát:

$$u_1(m) = \frac{1}{4} \cdot (-1) + \frac{1}{12} \cdot 1 + \frac{1}{2} \cdot 1 + \frac{1}{6} \cdot (-1) = \frac{1}{6}.$$

A második játékos hozamát is hasonlóan kiszámíthatnánk, de figyeljük meg, hogy ebben a játékban minden játékos haszna ugyanaz mint a másik játékos vesztesége, azaz 0−összegű játékról beszélünk, ezért $u_2(m) = -u_1(m) = -\frac{1}{6}$.

Figyeljük meg, hogy a második játékos aktuális stratégiája nem a legjobb válasz az első játékos stratégiájára. Helyette ha mindig írást mond, akkor $\frac{2}{3}$ valószínűséggel eltalálja, hogy mit választott az első játékos.

Definíció (Legjobb tiszta válasz)**.** Legyen $m_{-i} \in M_{-i}$ kevert stratégia az i-n kívül. Ekkor $s_i \in S_i$ az i-edik játékos **legjobb tiszta válasza** m_{-i}-re, ha tetszőleges $m'_i \in M_i$ kevert stratégia esetén $u_i(s_i, m_{-i}) \geq u_i(m'_i, m_{-i})$.

3.1.5. Állítás. *Legyen $m_{-i} \in M_{-i}$. Ekkor $m_i \in M_i$ a legjobb válasz m_{-i}-re $\Leftrightarrow$ minden $s_i \in S_i$-re, amire $m_i(s_i) > 0$, s_i legjobb tiszta válasz m_{-i}-re.*

Bizonyítás. $\Leftarrow$: Legyen α az i-edik játékos legjobb válaszának haszna. Ekkor $u_i(m) = \sum_{s_i \in S_i} m_i(s_i) u(s_i, m_{-i}) = \alpha$, mivel ha s_i a legjobb tiszta válasz, akkor $u(s_i, m_{-i}) = \alpha$.

$\Rightarrow$: $u_i(m) = \sum_{s_i \in S_i} m_i(s_i) u(s_i, m_{-i}) \leq \alpha$, és csak akkor lesz egyenlő α-val, ha $m_i(s_i) > 0$ esetén $u(s_i, m_{-i}) = \alpha$. □

3.1.6. Következmény. *$m \in M$ kevert Nash-egyensúly $\Leftrightarrow$ tetszőleges i-re és tetszőleges $s_i \in S_i$ tiszta stratégiára igaz, hogy $m_i(s_i) > 0$ esetén s_i legjobb tiszta válasz m_{-i}-re.*

3.1.6. Kő-papír-olló játék

Nézzük a hasznossági-mátrixot:

<table>
<tr><td>1. \ 2.</td><td>kő</td><td>papír</td><td>olló</td></tr>
<tr><td>kő</td><td>0
0</td><td>1
-1</td><td>-1
1</td></tr>
<tr><td>papír</td><td>-1
1</td><td>0
0</td><td>1
-1</td></tr>
<tr><td>olló</td><td>1
-1</td><td>-1
1</td><td>0
0</td></tr>
</table>

Definíció (Szimmetrikus játék)**.** Egy játék szimmetrikus, ha $S_1 = S_2 = \cdots = S_n$, és tetszőleges π permutációra és $s_1 \in S_1, \ldots, s_n \in S_n$-re $u_{\pi(i)}(s_1, \ldots, s_n) = u_i(s_{\pi(1)}, \ldots, s_{\pi(n)})$. Például a kő-papír-olló szimmetrikus, de a fej-vagy-írás játék nem szimmetrikus.

Keressünk Nash-egyensúlyt a kő-papír-olló játékban:

Tegyük fel, hogy $m = (m_1, m_2)$ kevert Nash-egyensúly. Mivel a játék szimmetrikus, ezért ugyanazok a hasznosságok, ha felcseréljük a játékosokat. Tegyük fel továbbá, hogy m_1 nem azonosan $\frac{1}{3}$, és m_1(kő) a legkisebb valószínűség. Ekkor a második játékos számára a papír nem legjobb tiszta válasz és a következmény miatt: m_2(papír) $= 0$. Emiatt az első játékos számára az olló nem legjobb tiszta válasz, és a következmény miatt: m_1(olló) $= 0$. Emiatt a második játékos számára a kő nem legjobb tiszta válasz, és a következmény miatt: m_2(kő) $= 0$. Emiatt az első játékos számára a papír nem legjobb tiszta válasz, és a következmény miatt: m_1(papír) $= 0$.

Azt kaptuk tehát, hogy az első játékosra m_1(kő) a legkisebb, de m_1(papír) $= 0$ és m_1(olló) $= 0$, ami ellentmondás. Tehát a Nash-egyensúly: $m_1 \equiv \frac{1}{3}$, $m_2 \equiv \frac{1}{3}$. Ez tényleg Nash-egyensúly, mivel ha az első játékos $\frac{1}{3}$ valószínűséggel választ, akkor a második játékosnak minden stratégia legjobb válasz.

3.2. Nash-egyensúly 0-összegű kétszemélyes játékokra

Definíció. Az $m_i \in M_i$ kevert stratégia **legjobb nyilvános stratégia** az i-edik játékos számára, ha

$$\min_{m_{-i} \in M_{-i}} u_i(m_i, m_{-i}) = \max_{m'_i \in M_i} \min_{m_{-i} \in M_{-i}} u_i(m'_i, m_{-i}).$$

0 összegű játék esetén elég az első játékos hasznossági mátrixát megadni, mert ebből adódik a második játékos haszna. A tobábbiakban $A \in \mathbb{R}^{k \times l}$ jelöli az első játékos hasznossági mátrixát.

Az első játékos égy m_1 kevert stratégiája egy $(p_1, p_2, \ldots, p_k)$ vektorral reprezentálható. A pA vektor j-edik eleme nem más, mint $u_1(m_1, j) = -u_2(m_1, j)$.

Ezért a második játékos legjobb tiszta válasza $p-re$: a pA egy legkisebb elemének indexe. Tehát az első játékos legjobb nyilvános stratégiája: olyan p , amire pA legkisebb eleme a lehető legnagyobb.

A fenti feladat egy LP feladat:

$$(P) \qquad \begin{array}{l} \max w \\ p \geq 0 \\ \sum_{i=1}^{k} p_i = 1 \\ pA \geq w\mathbf{1} \end{array}$$

Az LP feladat optimális megoldása megadja a legjobb nyilvános stratégiát. Nézzük a (P) feladat duálisát:

$$(D) \qquad \begin{array}{l} \min z \\ Aq \leq \mathbf{1}z \\ q \geq 0 \\ \mathbf{1}q = 1 \end{array}$$

3.2.1. Tétel (Neumann). *Legyen (p^*, w^*) a (P) feladat, (q^*, z^*) a (D) feladat optimuma. Ekkor*

i) $w^* = z^*$

ii) *A* $(p^*, q^*) \in M$ *kevert stratégiavektor Nash-egyensúly.*

Bizonyítás. Az első állítás következik a dualitás-tételből. A *ii)* bizonyításához tekintsük a következő állítást:

3.2.2. Állítás. q^* *legjobb válasz* p^**-ra.*

Bizonyítás. A komplementaritási feltételek miatt ahol $q^* > 0$, ott $p^*A - w^* = 0$, azaz w^* a p^*A vektor legkisebb eleme. Tehát q^* legjobb tiszta válaszok konvex kombinációja. □

Hasonlót állíthatunk p^*-ra: ahol $p^* > 0$, ott $Aq^* = z$, azaz z^* az Aq^* vektor legnagyobb eleme. Tehát p^* is legjobb válasz q^*-ra. Azt kaptuk tehát, hogy (p^*, q^*) kevert Nash-egyensúly. □

Most nézzük meg mit tudunk mondani szimmetrikus 0-összegű játékokról:

Definíció (Szimmetrikus stratégiavektor). Szimmetrikus játéknál egy kevert stratégiavektort szimmetrikusnak nevezünk, ha minden játékoshoz ugyanaz a kevert stratégia tartozik. Egy kevert stratégia szimmetrikus Nash-egyensúly, ha a belőle kapott szimmetrikus stratégiavektor Nash-egyensúly.

Szimmetrikus 0-összegű játék esetén a fent szereplő (P) és (D) feladat ugyanaz, így optimális megoldásaik is ugyanazok. Ebből következik, hogy szimmetrikus 0-összegű játéknál mindig van szimmetrikus Nash-egyensúly. A következő részben megmutatjuk, hogy ez nem 0-összegű játékokra is igaz.

3.3. Lemke-Howson algoritmus kétszemélyes szimmetrikus játékra

Először megmutatjuk, hogy ha szimmetrikus játékban tudunk szimmetrikus Nash-egyensúlyt találni, akkor ennek segítségével tetszőleges játékban megtalálható egy Nash-egyensúly.

3.3.1. Tétel. *Tekintsünk egy olyan játékot, ahol az első játékos hasznossági mátrixa A, a második játékos hasznossági mátrixa B, és A és B összes eleme pozitív (ez feltehető, mert A és B minden elemét ugyanannyival megnövelve ekvivalens játékot kapunk). Legyen C a következő mátrix:*

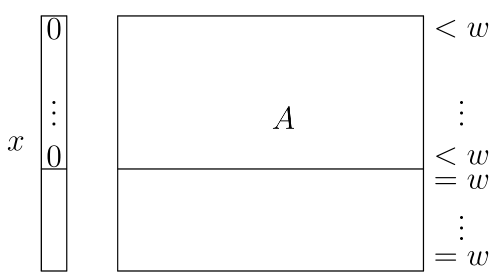
$$C = \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix} \qquad \begin{array}{c|cc} & x & y \\ \hline x & 0 & A \\ y & B^T & 0 \end{array}$$

Ha a C mátrixú játéknak (x, y) szimmetrikus Nash-egyensúlya, akkor $m_1 = \frac{x}{\sum x_i}$, $m_2 = \frac{y}{\sum y_i}$ Nash-egyensúly az eredeti játékban.

Bizonyítás. Mivel A és B minden eleme pozitív, sem x, sem y nem lehet azonosan 0, hiszen akkor (x, y) nem lenne legjobb válasz önmagára. Tehát $m_1 = \frac{x}{\sum x_i}$ és $m_2 = \frac{y}{\sum y_i}$ értelmes. Ha $x_i > 0$, akkor s_i legjobb tiszta válasz (x, y)-ra a szimmetrikus játékban, ami pont azt jelenti, hogy s_i legjobb tiszta válasz m_2-re az eredeti játékban. Fordítva is, ha $y_j > 0$, akkor s_j legjobb tiszta válasz (x, y)-ra a szimmetrikus játékban, és így s_j legjobb tiszta válasz m_1-re az eredeti játékban. Tehát a 3.1.6. Következmény alapján (m_1, m_2) Nash-egyensúly. □

A következőkben ismertetjük a Lemke-Howson algoritmust, ami szimmetrikus kétszemélyes játékban talál egy szimmetrikus Nash-egyensúlyt ($m_1 = m_2$). Ez egyben bizonyítja Nash tételét a kétjátékosos esetre.

Tekintsük az első játékos A hasznossági mátrixát, és legyen x egy kevert stratégia. Jelölje w az Ax maximális elemét. Ekkor lesznek olyan koordináták, amikre $Ax < w$, és lesznek olyanok, amikre az Ax eléri a maximumát.



A 3.1.6. Következmény alapján az x vektor pontosan akkor szimmetrikus Nash-egyensúly, ha $(Ax)_i < w$ esetén $x_i = 0$. Vizsgáljuk most a következő poliédert: $P = \{z : Az \leq 1, z \geq 0\}$. Feltehető, hogy $a_{ij} > 0$ minden i, j-re, mert ha A minden elemhez ugyanazt az értéket hozzáadhatjuk, akkor a várható hasznot konstanssal toljuk csak el, a Nash-egyensúly nem változik. Így feltehető, hogy a P poliéder korlátos.

Jelölés. Jelöljük a P csúcsainak halmazát V-vel. Vezessük be továbbá minden $z \in V$ csúcsra a következő indexhalmazokat:

$$N_z = \{i : z_i = 0\},$$
$$K_z = \{i : (Az)_i = 1\}.$$

Tudjuk, hogy $(|N_z| + |K_z|) \geq n$, mert egy csúcsnál legalább n egyenlőtlenség egyenlőséggel teljesül.

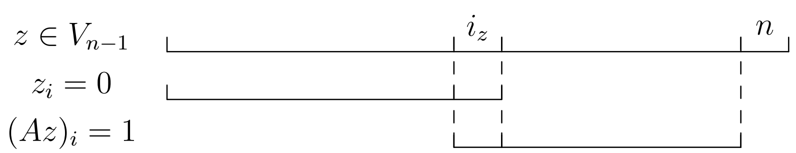
3.3.2. Lemma. *Ha $|N_z \cup K_z| = n$ és $z \neq \mathbf{0}$, akkor $x = \frac{z}{\sum z_j}$ egy szimmetrikus Nash-egyensúly.*

Bizonyítás. Ha $x_i > 0$, akkor $(Ax)_i = \max_{1 \leq j \leq n}(Ax)_j$, tehát x legjobb tiszta válaszok konvex kombinációja. $\square$

A továbbiakban az egyszerűség kedvéért feltesszük, hogy P nem degenerált, azaz minden csúcsában pontosan n db oldal találkozik ($|N_z| + |K_z| = n$). (Bizonyítható, hogy egy degenerált poliéderből nem degeneráltat kaphatunk az A mátrix kis perturbálásával, és elég kis perturbálás esetén az új játék Nash-egyensúlyának az eredeti játékban is Nash-egyensúly felel meg.)

Észrevétel. *Ha P nem degenerált, akkor tetszőleges z csúcs a poliéder pontosan n élének végpontja, és mindegyik él más-más $n-1$ feltételt teljesít egyenlőséggel a z egyenlőséggel teljesülő feltételei közül.*

Jelölés. Jelölje V_{n-1} az olyan $z \in V$-ket, amelyekre $N_z \cup K_z = \{1, 2, \ldots, n-1\}$, V_n pedig azokat, amelyekre $|N_z \cup K_z| = n$. Ha $z \in V_{n-1}$, akkor jelölje i_z azt az indexet, ami N_z-ben és K_z-ben is szerepel (tudjuk, hogy pontosan egy ilyen $i_z \in \{1, 2, \ldots, n-1\}$ index van).



Nevezzük a $z \in V_{n-1}$ csúcs i_z-szomszédainak a z-nek azt a két szomszédját, amit az i_z-hez tartozó egy-egy feltétel elhagyásával (pontosabban: az egyenlőséggel való megszorítás elhagyásával) kapunk.

Észrevétel. *A z csúcs i_z-szomszédai $(V_{n-1} \cup V_n)$-ben vannak.*

3.3.3. Állítás. *Ha $z \in V_{n-1}$ és az egyik i_z-szomszédja $z' \in V_{n-1}$, akkor z'-nek a z csúcs $i_{z'}$-szomszédja.*

Bizonyítás. Legyen F az az egyértelmű feltétel, amit z' egyenlőséggel teljesít, de z nem. Ekkor F vagy $(Az)_k \leq 1$ alakú, vagy $z_k \geq 0$ alakú, de mindkét esetben $k = i_{z'}$, mert a többi index nem szerepelhet $N_{z'}$-ben is és $K_{z'}$-ben is. Tehát z'-nek $i_{z'}$-szomszédját kapjuk, ha az F feltételt nem követeljük meg egyenlőséggel. Ez pedig pont a z csúcsot adja. $\square$

Vegyük észre, hogy $\mathbf{0} \in V_n$, de ez sajnos nem ad Nash-egyensúlyt. Az origónak viszont van egy $V_{n-1} \cup V_n$-beli z^1 szomszédja, amit a $z_n = 0$ feltétel elhagyásával kapunk. Ha ez V_n-beli, akkor találtunk egy szimmetrikus Nash-egyensúlyt a 3.3.2. Lemma szerint. Ha nem, akkor lépjünk tovább a z^1 másik i_{z^1}-szomszédjára. Ezt ismételjük, egészen addig amíg egy V_n-beli csúcsban nem érünk.

Figyeljük meg, hogy a lépések során nem érhetünk körbe: ha egy V_{n-1}-beli z csúcsba érnénk vissza, akkor annak a 3.3.3. Állítás alapján három különböző i_z-szomszédja lenne, ami lehetetlen. A **0** csúcsba sem érhetünk vissza, hiszen neki egyetlen szomszédja van V_{n-1}-ben. Az algoritmus tehát a következő:

0. Legyen z^1 az origónak a $z_n = 0$ feltétel elhagyásával kapott szomszédja, és legyen $j = 1$.
1. Ha $z^j \in V_n$, akkor kész vagyunk.
2. Ha nem, hagyjuk el azt az i_{z^j}-hez tartozó feltételt, amelyik nem az utolsó lépésnél jött be, és legyen az így kapott i_{z^j}-szomszéd z^{j+1}.
3. Legyen $j := j + 1$, és lépjünk az 1. lépésre.

3.3.4. Következmény (Nash)**.** *Véges stratégiahalmazú 2 személyes játékban mindig van Nash-egyensúly.*

Megjegyzés. A Lemke-Howson algoritmus véges, de lehet exponenciális futási idejű az A mátrix méretében, hiszen a poliédernek lehet exponenciálisan sok csúcsa.

4. fejezet

Konvex optimalizálás

4.1. Konvex halmazok

4.1.1. Alaptulajdonságok

Definíció. Egy $C \subseteq \mathbb{R}^n$ halmaz **konvex**, ha tetszőleges $x, y \in C$ és $0 \leq \lambda \leq 1$-re $\lambda x + (1-\lambda)y \in C$. Azaz: ha egy szakasz két végpontja benne van C-ben, akkor a teljes szakasz benne van.

Az $x_1, \ldots, x_k \in \mathbb{R}^n$ vektorok egy **konvex kombinációja** egy olyan vektor, ami előáll $\sum_{i=1}^k \lambda_i x_i$ alakban, ahol $0 \leq \lambda_i \leq 1$ $(i = 1, \ldots, k)$ és $\sum_{i=1}^k \lambda_i = 1$.

Egy $X \subseteq \mathbb{R}^n$ halmaz **konvex burka** az összes olyan vektorból áll, ami X véges sok elemének konvex kombinációja. Jelölése: $\mathrm{conv}(X)$.

4.1.1. Állítás (Konvex halmaz ekvivalens definiciója)**.** *C konvex $\Leftrightarrow \mathrm{conv}(C) = C$.*

Bizonyítás. Nyilvánvaló, hogy ha $\mathrm{conv}(C) = C$, akkor C konvex. A másik irány bizonyításához legyen $x = \sum_{i=1}^k \lambda_i x_i$. Indukciót használink k-ra, $k = 2$-re definíció szerint $x \in C$.

Tegyük fel, hogy $k-1$-re igaz az állítás. Feltehető, hogy $\lambda_i > 0$ $(i = 1, \ldots, k)$. Legyen $\lambda := \sum_{i=1}^{k-1} \lambda_i$, $x' := \sum_{i=1}^{k-1} \frac{\lambda_i}{\lambda} x_i$. Az x' konvex kombinációja $k-1$ C-beli pontnak, így indukció szerint $x' \in C$. Ekkor $\sum_{i=1}^k \lambda_i x_i = \lambda x' + \underbrace{\lambda_k}_{=1-\lambda} x_k = \lambda \underbrace{x'}_{\in C} + (1-\lambda) \underbrace{x_k}_{\in C} \in C$. □

4.1.2. Állítás. *Konvex halmazok metszete konvex, akár végtelen soké is.*

Bizonyítás. Legyen $C = \cap C_i$, $x, y \in C$. Ekkor $x, y \in C_i$ minden i-re. Tehát $0 \leq \lambda \leq 1$-re $\lambda x + (1-\lambda)y \in C_i$ $\forall i \Rightarrow \in C$. □

Definíció. Az $x_1, \ldots, x_k \in \mathbb{R}^n$ vektorok egy **affin kombinációja** egy olyan vektor, ami előáll $\sum_{i=1}^k \lambda_i x_i$ alakban, ahol $\sum_{i=1}^k \lambda_i = 1$. Egy $X \subseteq \mathbb{R}^n$ halmaz **affin burka** az összes olyan vektorból áll, ami X véges sok elemének affin kombinációja. Ez egy affin altér, jelölése: $\mathrm{aff}(X)$. Az affin burok lineáris eltoltját $\mathrm{lin}(X)$ jelöli. Az X halmaz **dimenziója**: $\dim(X) := \dim(\mathrm{lin}(X))$.

4.1.3. Állítás. *Konvex C-re* $\mathrm{aff}(C) = \{\mu x + (1-\mu)y : x, y \in C,\ \mu \in \mathbb{R}\}$.

Bizonyítás. Legyen $x_1, \dots, x_k \in C$, $z = \sum_{i=1}^k \mu_i x_i$, ahol $\sum \mu_i = 1$. Be kell látnunk, hogy z előáll $\mu x + (1-\mu)y$ alakban, ahol $x, y \in C$. Ha $\mu_i \geq 0$ $(i = 1, \dots, k)$, akkor $z \in C$ és kész vagyunk.

Legyen $\mu = \sum_{i:\mu_i>0} \mu_i$, $x = \sum_{i:\mu_i>0} \frac{\mu_i}{\mu} x_i$. Ekkor x konvex kombináció, tehát $x \in C$. Legyen $y = \sum_{i:\mu_i<0} \frac{\mu_i}{1-\mu} x_i \in C$. Ekkor $z = \mu x + (1-\mu)y$. □

Minden poliéder konvex halmaz, hiszen konvex halmazok (félterek) metszete. A következőkben azt vizsgáljuk, hogy poliéderek oldalainak mi a megfelelője általános konvex halmazok esetében.

Definíció. Egy $E \subseteq C$ halmaz **extremális halmaza** a C konvex halmaznak, ha E konvex, és teljesül, hogy ha $x, y \in C$-re és $0 < \lambda < 1$-re $\lambda x + (1-\lambda)y \in E$, akkor $x \in E$ és $y \in E$. Azaz: ha egy C-beli szakasz egy belső pontja benne van E-ben, akkor az egész szakasz benne van.

4.1.4. Állítás. *Ha* $\max\{wx : x \in C\}$ *megoldható, akkor az optimális megoldások halmaza extremális halmaz.*

Bizonyítás. Legyen S az optimális megoldások halmaza, α pedig az optimumérték. Először belátjuk, hogy S konvex: ha $x, y \in S$, akkor $wx = wy = \alpha$, tehát $w(\lambda x + (1-\lambda)y) = \alpha$.

Legyen most $z \in S$, $z = \lambda x + (1-\lambda)y$, ahol $0 < \lambda < 1$, és $x, y \in C$. Be kell látnunk, hogy $x, y \in S$.

$$\alpha = wz = \lambda \underbrace{wx}_{\leq \alpha} + (1-\lambda) \underbrace{wy}_{\leq \alpha} \leq \lambda\alpha + (1-\lambda)\alpha = \alpha.$$

Egyenlőség csak akkor lehet, ha $wx = \alpha$ és $wy = \alpha$, azaz $x, y \in S$. □

Példa. Henger extremális halmazai:

0 dimenziós extremális halmaz elemei (extremális pontok): a két körlap határpontjai;

1 dimenziós extremális halmazok: a henger alkotói;

2 dimenziós extremális halmazok: a két körlap;

3 dimenziós extremális halmazok: az egész henger.

Tehát még kompakt konvex halmazokra sem teljesül az a tulajdonság, ami poliéderek oldalaira igaz, hogy i dimenziós extrém halmaz mindig része egy $i+1$ dimenziósnak. Azonban az oldalak néhány fontos tulajdonsága általánosítható extremális halmazokra.

4.1.5. Állítás. *Ha E extremális halmaza C-nek, akkor $E = \text{aff}(E) \cap C$.*

Bizonyítás. Legyen $z \in \text{aff}(E) \cap C$. Ekkor $z = \mu x + (1-\mu)y$, ahol $x, y \in E$. Ha $0 \leq \mu \leq 1$, akkor az E konvexitása miatt $z \in E$ és kész vagyunk. Ha $\mu > 1$, akkor $x = \frac{1}{\mu}\underbrace{z}_{\in C} + \frac{\mu-1}{\mu}\underbrace{y}_{\in E}$. Mivel $y \in E$ és $E \subseteq C$, ezért $y \in C$, továbbá mivel $x \in E$, és E extremális halmaz, ezért $z \in E$. □

Megjegyzés. Ha E_1, E_2 extremális C-ben és $E_1 \subsetneqq E_2$, akkor E_1 extremális halmaza E_2-nek.

4.1.6. Állítás. *Ha E_1, E_2 extremális halmazai C-nek és $E_1 \subsetneqq E_2$, akkor* $\dim(E_1) < \dim(E_2)$.

Bizonyítás. Mivel E_1 extremális halmaza E_2-nek, ezért $E_1 = \text{aff}(E_1) \cap E_2$, tehát $\text{aff}(E_1) \subsetneqq \text{aff}(E_2)$, emiatt $\dim(E_1) < \dim(E_2)$. □

4.1.7. Állítás. *Ha E_1 extremális halmaza C-nek, és E_2 extremális halmaza E_1-nek, akkor E_2 extremális halmaza C-nek.*

Bizonyítás. Legyen $x, y \in C$, $0 < \lambda < 1$, $z = \lambda x + (1-\lambda)y \in E_2$. Ekkor $z \in E_1$, mert $E_2 \subseteq E_1$. E_1 extremális C-ben, emiatt $x, y \in E_1$. Továbbá $z \in E_2$, E_2 extremális E_1-ben, emiatt $x, y \in E_2$. □

4.1.8. Tétel (Minkowski, bizonyítás nélkül)**.** *Ha C korlátos és zárt konvex halmaz, akkor C az extrém pontjainak konvex burka.*

Definíció. A $C \subseteq \mathbb{R}^n$ halmaz **konvex kúp**, ha konvex és tetszőleges $x \in C$ és $\lambda \geq 0$ esetében $\lambda x \in C$. Egy konvex kúp **csúcsos**, ha a 0 extremális pontja.

4.1.9. Állítás. *C konvex kúp csúcsos $\Leftrightarrow$ nem tartalmaz egydimenziós lineáris alteret.*

Bizonyítás. $\Rightarrow$: Ha tartalmaz $\{\mu x : \mu \in \mathbb{R}\}$ egyenest, akkor $x \in C$, $-x \in C$, tehát 0 nem extremális pontja.

$\Leftarrow$: Ha 0 nem extremális, akkor létezik $x, y \in C$ és $0 < \lambda < 1$, amire $\lambda x + (1-\lambda)y = 0$, tehát az x-et és y-t összekötő egyenes egydimenziós lineáris altér C-ben. □

Definíció (Megengedett irányok kúpja)**.** Legyen C konvex halmaz, $z \in C$. Ekkor a megengedett irányok kúpja z-ben

$$K(C, z) = \{x \in \mathbb{R}^n : \exists \varepsilon > 0, z + \delta x \in C \ \forall 0 \leq \delta \leq \varepsilon\}.$$

4.1.10. Állítás. *$K(C, z)$ konvex kúp.*

Bizonyítás. Legyen $x, y \in K(C, z)$. Az x-hez tartozó ε-t jelöljük ε_x-el, az y-hoz tartozót pedig ε_y-al. Legyen $\varepsilon = \min\{\varepsilon_x, \varepsilon_y\}$. Legyen továbbá $\delta \leq \varepsilon$. Ekkor

$$z + \delta(\lambda x + (1-\lambda)y) = \lambda(\underbrace{z + \delta x}_{\in C}) + (1-\lambda)(\underbrace{z + \delta y}_{\in C}) \Rightarrow z + \delta(\lambda x + (1-\lambda)y) \in C.$$

□

4.1.11. Állítás. *$K(C, z)$ csúcsos $\Leftrightarrow$ z extremális pontja C-nek.*

Bizonyítás. Ha z nem extremális pont, akkor belső pontja egy $[x, y]$ szakasznak, és $(x - z) \in K(C, z)$, $(y - z) \in K(C, z)$, tehát tartalmaz 0-n átmenő egyenest. Ha viszont $K(C, z)$ tartalmaz 0-n átmenő, $\{\mu s : \mu \in \mathbb{R}\}$ egyenest, akkor $\exists \delta_1 : z + \delta_1 s \in C$, $\exists \delta_2 :$ $z - \delta_2 s \in C$, tehát z nem extremális. □

4.1.2. Konvex halmazok szeparációja

4.1.12. Tétel. *Ha C zárt konvex halmaz, és $z \notin C$, akkor van egy egyértelmű z-hez legközelebbi pont C-ben.*

Bizonyítás. Legyen $\mu = \inf\{\|x - z\| : \ x \in C\}$. Ez véges, és létezik $x^k \in C$ sorozat, amire $\|x^k - z\| \to \mu$. Ennek van konvergens részsorozata, ami egy $x \in C$ ponthoz tart, tehát $\|x - z\| = \mu$.

Ha $y \in C$-re $\|y - z\| = \mu$,akkor $\mu^2 \leq \|\frac{x+y}{2} - z\|^2 = \mu^2 - \frac{1}{4}\|y - x\|^2$, tehát $y = x$. □

Definíció (Vetítés konvex zárt halmazra)**.** A $z \notin C$ ponthoz legközelebbi pontot C-ben a z pont C-re való vetítésének nevezzük, jelölése $\pi_C(z)$.

Észrevétel. *Tetszőleges $x \in C$-re $(x - \pi_C(z))^T(z - \pi_C(z)) \leq 0$.*

4.1.13. Tétel (Konvex szeparációs tétel)**.** *Legyen C zárt konvex halmaz, és $z \notin C$. Ekkor létezik $s \in \mathbb{R}^n$ és $\epsilon > 0$: $s^T x \leq s^T z - \epsilon$ tetszőleges $x \in C$-re.*

Bizonyítás. Legyen $s = z - \pi_C(z)$, és legyen $\epsilon = \|s\|^2$. Figyeljük meg, hogy ha $x \in C$, akkor $(x - \pi_C(z))^T(z - \pi_C(z)) \leq 0$, hiszen különben az $[x, \pi_C(z)]$ szakaszon lenne olyan pont, ami közelebb van z-hez, mint $\pi_C(z)$. Így

$$s^T x \leq s^T \pi_C(z) = s^T z + s^T(\pi_C(z) - z) = s^T z - \epsilon.$$

□

Feladat. *Bizonyítsuk be a konvex szeparációs tétel segítségével a Farkas Lemmát!*

4.1.14. Tétel. *Legyen C konvex halmaz, és $z \notin C$. Ekkor létezik $s \in \mathbb{R}^n$ nemnulla vektor, amire $s^T x \leq s^T z$ tetszőleges $x \in C$-re.*

Bizonyítás. Létezik $z^k \to z$ sorozat, amire $z^k \notin \mathrm{cl}(C)$ semmilyen k-ra. Az előző tétel alapján vannak s^k nemnulla vektorok, amikre $(s^k)^T x \leq (s^k)^T z^k$ tetszőleges $x \in C$-re. Feltehetjük, hogy $\|s^k\| = 1$ minden k ra. Legyen s az s^k sorozat tetszőleges torlódási pontja; errre $s^T x \leq s^T z$ tetszőleges $x \in C$-re. □

4.1.15. Tétel. *Legyenek C_1 és C_2 diszjunkt konvex halmazok. Létezik $s \in \mathbb{R}^n$ nemnulla vektor, amire $s^T x \leq s^T y$ tetszőleges $x \in C_1$-re és $y \in C_2$-re.*

Bizonyítás. Legyen $C = C_1 - C_2$; ez konvex halmaz, és $0 \notin C$, tehát az előző tétel értelmében létezik $s \in \mathbb{R}^n$ nemnulla vektor: $s^T x \leq 0$ tetszőleges $x \in C$-re. Így $s^T x \leq s^T y$ tetszőleges $x \in C_1$-re és $y \in C_2$-re. □

4.2. Konvex függvények

Definíció (Konvex függvény)**.** Legyen $C \subseteq \mathbb{R}^n$ konvex halmaz. Az $f : C \to \mathbb{R}$ függvény konvex, ha tetszőleges $x, y \in C$ és $0 \leq \lambda \leq 1$ esetén $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$.

Szemléletesen ezt úgy is mondhatjuk, hogy teszőleges $x, y \in C$ esetén az $[x, y]$ szakaszon a függvény grafikonja a két végpontot összekötő szakasz alatt megy.

4.2.1. Tétel (Jensen egyenlőtlenség)**.** *Legyen C konvex halmaz, $f : C \to \mathbb{R}$ konvex függvény, továbbá $x_1, \ldots, x_k \in C$, $\lambda_i \geq 0$, $i = 1, \ldots, k$, $\sum_{i=1}^k \lambda_i = 1$. Ekkor $f(\sum_{i=1}^k \lambda_i x_i) \leq \sum_{i=1}^k \lambda_i f(x_i)$.*

Bizonyítás. Indukció k-ra; $k = 2$-re igaz az állítás a konvexitás definíciója szerint. Tegyük fel, hogy $k > 2$. Legyen $\lambda = \sum_{i=1}^{k-1} \lambda_i$, $x = \sum_{i=1}^{k-1} \frac{\lambda_i}{\lambda} x_i$. Mivel x C-beli pontok konvex kombinációja, ezért $x \in C$. Ekkor

$$f(\sum_{i=1}^k \lambda_i x_i) = f(\lambda x + \lambda_k x_k) = f(\lambda x + (1-\lambda) x_k) \underbrace{\leq}_{f \text{ konvex}} \lambda f(x) + (1-\lambda) f(x_k)$$

$$= \lambda f(\sum_{i=1}^{k-1} \frac{\lambda_i}{\lambda} x_i) + \lambda_k f(x_k) \underbrace{\leq}_{\text{indukció}} \sum_{i=1}^{k-1} \lambda_i f(x_i) + \lambda_k f(x_k) = \sum_{i=1}^{k} \lambda_i f(x_i).$$

□

Definíció (Szinthalmaz)**.** $D_\alpha(f) = \{x \in C : f(x) \leq \alpha\}$ az f függvény α-hoz tartozó szinthalmaza.

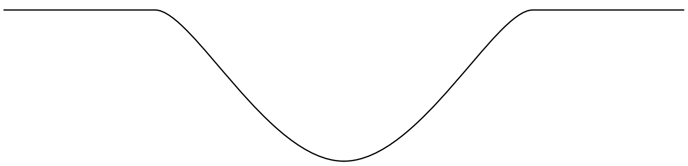
4.2.2. Lemma. *Ha f konvex, akkor D_α konvex tetszőleges α-ra.*

Bizonyítás. Legyen $x, y \in D_\alpha$, $0 \leq \alpha \leq 1$. Ekkor

$$f(\lambda x + (1-\lambda)y) \underbrace{\leq}_{f\ konvex} \lambda f(\underbrace{x}_{\in D_\alpha}) + (1-\alpha) f(\underbrace{y}_{\in D_\alpha}) \leq \lambda\alpha + (1-\lambda)\alpha = \alpha$$

. □

Megjegyzés. A lemma fordítottja nem igaz: egy függvény minden szinthalmaza konvex $\not\Rightarrow$ a függvény konvex. Az olyan függvényt, aminek minden szinthalmaza konvex, kvázikonvexnek nevezzük. Példa:



4.2.3. Lemma. *Legyen C konvex halmaz, és $f : C \to \mathbb{R}$. Az f függvény konvex $\Leftrightarrow$ tetszőleges $x, y \in C$-re a $\phi(\lambda) := f(x + \lambda(y - x))$ függvény konvex a $[0, 1]$ intervallumon.*

Bizonyítás. $\Leftarrow$: $x, y \in C$, $0 \leq \lambda \leq 1$. Ekkor $f(\lambda x + (1-\lambda)y) = \phi(1-\lambda) = \phi(0 \cdot \lambda + 1 \cdot (1-\lambda)) \underbrace{\leq}_{\phi\ konvex} \lambda \underbrace{\phi(0)}_{=f(x)} + (1-\lambda) \underbrace{\phi(1)}_{=f(y)} = \lambda f(x) + (1-\lambda) f(y)$.

$\Rightarrow$: $0 \leq \nu \leq 1$, $\lambda_1, \lambda_2 \in [0, 1]$. Ekkor

$$\begin{aligned}
\phi(\nu\lambda_1 + (1-\nu)\lambda_2) &= f(x + (\nu\lambda_1 + (1-\nu)\lambda_2)(y - x)) \\
&= f(\nu(x + \lambda_1(y - x)) + (1-\nu)(x + \lambda_2(y - x))) \\
&\underbrace{\leq}_{f \text{ konvex}} \nu f(x + \lambda_1(y - x)) + (1-\nu) f(x + \lambda_2(y - x)) \\
&= \nu\phi(\lambda_1) + (1-\nu)\phi(\lambda_2).
\end{aligned}$$

□

Analízis tanulmányainkból tudjuk, hogy egy nyílt I intervallumon értelmezett, folytonosan differenciálható $\phi : I \to \mathbb{R}$ függvény pontosan akkor konvex, ha $\phi'(x)$ monoton növő. Ennek az állításnak egy többdimenziós változatát bizonyítjuk alább.

Definíció. Ha f folytonosan differenciálható a C konvex nyílt halmazon, akkor f **gradiense** x-ben: $\nabla f(x) = (\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x))$.

A gradiens segítségével kiszámolhatjuk az iránymenti deriváltat tetszőleges s irányban:

$$\lim_{\lambda \to +0} \frac{f(x + \lambda s) - f(x)}{\lambda} = \nabla f(x)^T s.$$

4.2.4. Tétel. *Legyen C konvex, nyílt halmaz, $f : C \to \mathbb{R}$ folytonosan differenciálható függvény. Ekkor f konvex $\Leftrightarrow$ tetszőleges $x, y \in C$-re $\nabla f(x)^T(y - x) \leq f(y) - f(x) \leq \nabla f(y)^T(y - x)$.*

Bizonyítás. $\Rightarrow$: $f(x + (1-\lambda)(y - x)) = f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$. Ekkor

$$\frac{f(x + (1-\lambda)(y - x)) - f(x)}{1 - \lambda} \leq f(y) - f(x).$$

Ha $\lambda \to 1$, akkor a baloldal tart az iránymenti deriválthoz az x pontban, azaz $\nabla f(x)^T(y-x)$-hez. A másik egyenlőtlenséget ugyanígy bizonyíthatjuk x-et és y-t felcserélve.

$\Leftarrow$: Tetszőleges $x, y \in C$-re $\nabla f(x)^T(y-x) \leq \nabla f(y)^T(y-x)$. Ekkor tetszőleges $0 \leq \lambda_1 < \lambda_2 \leq 1$ számokra: $\nabla f(\lambda_2 x + (1-\lambda_2)y)^T(x-y) \leq \nabla f(\lambda_1 x + (1-\lambda_1)y)^T(x-y)$ Ezek szerint $\phi'(\lambda)$ monoton növő, tehát ϕ konvex, és így f konvex. $\square$

Definíció (Hesse mátrix). Legyen C konvex és nyílt halmaz, f kétszer folytonosan differenciálható függvény a C halmazon. f Hesse mátrixa: $(\nabla^2 f(x))_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x)$.

4.2.5. Tétel. *Ha f kétszer folytonosan differenciálható, akkor konvex $\Leftrightarrow$ $\nabla^2 f(x)$ pozitív szemidefinit minden $x \in C$-re.*

Bizonyítás. $\Rightarrow$: $x \in C$, $y \in C$, $s = y - x$. Ekkor $s^T \nabla^2 f(x) s = \phi''(0)$. A konvexitásból következik, hogy $\phi''(0) \geq 0$, hiszen ϕ' monoton növő. A C halmaz nyílt, így tartalmaz x körüli gömböt, ezért $s^T \nabla^2 f(x) s \geq 0$ $\forall s \in \mathbb{R}^n$, tehát $\nabla^2 f(x)$ pozitiv szemidefinit.

$\Leftarrow$: $0 \leq s \nabla^2 f(x+\lambda s)s = \phi''(\lambda)$. Így $\phi'' \geq 0$ $\forall \lambda \in [0,1] \Rightarrow \phi'$ monoton növő $\Rightarrow$ f konvex. $\square$

4.3. Feltétel nélküli optimalizálás

Definíció. Az $f : C \to \mathbb{R}$ függvénynek $\bar{x} \in C$ **lokális minimuma**, ha $\exists$ $\varepsilon > 0$ amire $f(x) \geq f(\bar{x})$ $\forall x \in B(\bar{x}, \varepsilon) \cap C$, ahol $B(\bar{x}, \varepsilon)$ az $\bar{x}$ körüli ε sugarú gömb.

$\bar{x} \in C$ **globális minimum**, ha $f(x) \geq f(\bar{x})$ $\forall x \in C$.

4.3.1. Tétel. *Ha f konvex és $\bar{x}$ lokális minimuma f-nek, akkor globális minimuma is.*

Bizonyítás. $f(x) < f(\bar{x}) \Rightarrow f(\lambda x + (1-\lambda)\bar{x}) \leq \lambda f(x) + (1-\lambda) f(\bar{x}) < f(\bar{x})$ tetszőleges $0 < \lambda \leq 1$ esetén. Tehát $\bar{x}$ tetszőleges környezetében van kisebb értékű pont. $\square$

4.3.2. Tétel. *Legyen C nyílt, konvex halmaz, és f konvex, folytonosan differenciálható függvény C-n. Ekkor $\bar{x}$ globális minimum $\Leftrightarrow$ $\nabla f(\bar{x}) = 0$.*

Bizonyítás. $\Leftarrow$: legyen $x \in C$ tetszőleges. Ekkor

$$0 = \nabla f(\bar{x})^T(x - \bar{x}) \underbrace{\leq}_{\text{4.2.4. tétel}} f(x) - f(\bar{x}) \Rightarrow f(x) \geq f(\bar{x}).$$

$\Rightarrow$: tetszőleges $s \in \mathbb{R}^n$-re: $f(\bar{x} + \lambda s) \geq f(\bar{x})$ ha λ elég kicsi ($\bar{x} + \lambda s \in C$). Ekkor $\frac{f(\bar{x}+\lambda s) - f(\bar{x})}{\lambda} \geq 0$ tart az iránymenti deriválthoz: $\nabla f(\bar{x})^T s$-hez ha $\lambda \to 0$. Tehát $\nabla f(\bar{x})^T s \geq 0$ $\forall s \in \mathbb{R}^n \Rightarrow \nabla f(\bar{x}) = 0$. $\square$

Definíció. A C konvex halmaz **relatív belseje**

$$\mathrm{rint}(C) = \{x \in C : \text{ tetszőleges } y \in C\text{-re } \exists x' \in C \text{ és } 0 < \lambda < 1\text{: } x = \lambda x' + (1-\lambda)y\}.$$

A C halmaz **relatív nyílt**, ha $C = \mathrm{rint}(C)$.

4.3.3. Tétel. *Legyen C relatív nyílt, konvex halmaz, és f konvex, folytonosan differenciálható függvény C-n. Ekkor $\bar{x}$ globális minimum $\Leftrightarrow \nabla f(\bar{x})^T s = 0\ \forall s \in \operatorname{lin}(C)$.*

Bizonyítás. $\Leftarrow$: legyen $x \in C$ tetszőleges. Ekkor

$$0 = \nabla f(\bar{x})^T \underbrace{(x - \bar{x})}_{\in \operatorname{lin}(C)} \underbrace{\leq}_{\text{4.2.4. tétel}} f(x) - f(\bar{x}) \Rightarrow f(x) \geq f(\bar{x}).$$

$\Rightarrow$: tetszőleges $s \in \operatorname{lin}(C)$-re: ha λ elég kicsi, akkor $\bar{x} + \lambda s \in C$, tehát $f(\bar{x} + \lambda s) \geq f(\bar{x})$. Ekkor $\frac{f(\bar{x}+\lambda s) - f(\bar{x})}{\lambda} \geq 0$ tart az iránymenti deriválthoz, $\nabla f(\bar{x}) s$-hez ha $\lambda \to 0$. Tehát $\nabla f(\bar{x}) s \geq 0\ \forall s \in \operatorname{lin}(C)$. □

4.4. Feltételes optimalizálás

A konvex optimalizálási (KO) feladat a következő:

$$\begin{aligned} \min f(x) & \\ x \in C & \\ g_i(x) \leq 0 & \qquad (i = 1, \ldots, m), \end{aligned}$$

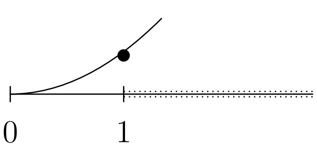
ahol $C \subseteq \mathbb{R}^n$ konvex, és f, g_i $(i = 1, \ldots, m)$ konvexek C-n és folytonosan differenciálhatóak C egy nyílt környezetében.

Definíció. A megengedett megoldások halmaza $C_g = \{x \in C : g_i(x) \leq 0\ (i = 1, \ldots, m)\}$.

A megengedett megoldások halmaza konvex, mivel konvex függvény szinthalmazai konvexek, és konvex halmazok metszete is konvex.

Megjegyzés. Általában C_g nem relatív nyílt (ha az, akkor az optimalitás szükséges és elégséges feltétele, hogy a gradiens merőleges a lineáris altérre). **Például** $n = m = 1$,

$$\begin{aligned} & f(x) = x^2 \\ & g_1(x) = 1 - x \\ & C = \mathbb{R} \end{aligned}$$



Definíció. Az x pontban a **megengedett irányok kúpja**

$$K(x) := \{s \in \mathbb{R}^n : \exists \varepsilon > 0 : \text{ tetszőleges } 0 \leq \lambda \leq \varepsilon\text{-ra } x + \lambda s \in C_g\}.$$

$K(x)$ tulajdonságai:

i) $K(x)$ konvex kúp;

ii) $K(x)$ csúcsos $\Leftrightarrow x$ extremális pontja C_g-nek;

iii) $K(x) \subseteq \mathrm{lin}(C_g)$;

iv) $K(x) = \mathrm{lin}(C_g) \Leftrightarrow x \in \mathrm{rint}(C_g)$

4.4.1. Tétel. *$\bar{x}$ optimális megoldása a (KO) feladatnak* $\Leftrightarrow \nabla f(\bar{x})^T s \geq 0 \ \forall s \in K(\bar{x})$

Bizonyítás. $\Leftarrow$: Legyen $\bar{x} \neq x \in C_g$, $s = x - \bar{x}$. Ekkor $s \in K(\bar{x})$. Tudjuk, hogy

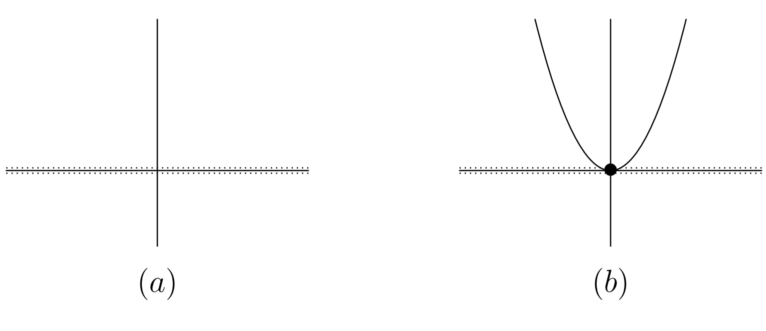
$$f(x) - f(\bar{x}) \underbrace{\geq}_{f\ konvex} \nabla f(\bar{x})^T s \underbrace{\geq 0}_{s \in K(\bar{x})} \Rightarrow f(x) \geq f(\bar{x}).$$

$\Rightarrow$: Legyen $s \in K(\bar{x})$. Ekkor definició szerint $\exists \varepsilon > 0$: $x_\lambda := \bar{x} + \lambda s \in C_g \ \forall 0 < \lambda \leq \varepsilon$. $f(x_\lambda) \geq f(\bar{x}) \Rightarrow \frac{f(x_\lambda) - f(\bar{x})}{\lambda} \geq 0$. Ha $\lambda \to 0$, akkor $0 \leq \frac{f(\bar{x} + \lambda s) - f(\bar{x})}{\lambda} \to \nabla f(\bar{x})^T s$. □

4.4.1. A Karush-Kuhn-Tucker tétel

Feltételes optimalizálás esetén is hasznos lenne a globális optimumot a gradiensekkel jellemezni, azaz a $\nabla f(x)$, $\nabla g_i(x)$ $(i = 1, \ldots, m)$ vektorok ismeretében eldönteni, hogy x globális optimum-e. Sajnos van arra példa, hogy két feladatban adott x pontra a $\nabla f(x)$, $\nabla g_i(x)$ $(i = 1, \ldots, m)$ értékek megegyeznek, de az egyikben x globális optimum, a másikban pedig nem.

$$\begin{array}{rl} & C = \mathbb{R}^2 \\ & f(x_1, x_2) = x_1 \\ & g_1(x_1, x_2) = x_2 \\ (a) & g_2(x_1, x_2) = -x_2 \\ (b) & g_2(x_1, x_2) = x_1^2 - x_2 \end{array}$$



(a) (b)

A fenti példában az (a) feladatnál és a (b) feladatnál a gradiensek 0-ban ugyanazok, de míg (a)-nál $-\infty$ az infimum, addig (b)-nél $C_g = \{0\}$, tehát az optimum 0.

Definíció. A (KO) feladatnak x **Slater pontja**, ha

i) $x \in \text{rint}(C)$,

ii) $g_i(x) \leq 0$ $(i = 1, \ldots, m)$,

iii) $g_i(x) < 0$ ha g_i nemlineáris.

A (KO) feladat **Slater reguláris**, ha van Slater pont.

Az előző példánál (a) esetben tetszőleges $(x_1, 0)$ pont Slater pont, tehát a feladat Slater reguláris, míg a (b) esetben nincs Slater pont, mert 0 az egyetlen megengedett pont, ami nem teljesíti a $iii)$ tulajdonságot: $g_2(0) = 0$, de g_2 nemlineáris.

4.4.2. Tétel (Karush-Kuhn-Tucker)**.** *Legyen C relatív nyílt és (KO) Slater reguláris. Ekkor $\bar{x} \in C_g$ optimális megoldása a (KO) feladatnak $\Leftrightarrow$ $\exists \mu_i \geq 0 \quad (i = 1, \ldots, m)$: $\mu_i = 0$ ha $g_i(\bar{x}) < 0$, és*

$$\nabla f(\bar{x}) + \sum_{i=1}^{m} \mu_i \nabla g_i(\bar{x}) \text{ merőleges } \text{lin}(C)\text{-re.}$$

Bizonyítás. $\Leftarrow$: Legyen $I = \{i \in \{1, \ldots, m\} : g_i(\bar{x}) = 0\}$. Vegyünk egy tetszőleges $s \in K(\bar{x})$ vektort, ami persze $\text{lin}(C)$-ben van. Ekkor tudjuk, hogy

$$0 = s^T(\nabla f(\bar{x}) + \sum_{i=1}^{m} \mu_i \nabla g_i(\bar{x})) = s^T \nabla f(\bar{x}) + \sum_{i=1}^{m} \mu_i s^T \nabla g_i(\bar{x}).$$

De $\sum_{i=1}^{m} \mu_i s^T \nabla g_i(\bar{x}) \leq 0$ mert $\sum_{i=1}^{m} \mu_i s^T \nabla g_i(\bar{x}) = \sum_{i \in I} \mu_i s^T \nabla g_i(\bar{x})$, és $i \in I$ esetén $s^T \nabla g_i(\bar{x}) \leq 0$, mert s megengedett irány és $g_i(\bar{x}) = 0$. Tehát $0 \leq s^T \nabla f(\bar{x}) \Rightarrow \bar{x}$ optimális megoldás.

$\Rightarrow$: Indirekt tegyük fel, hogy $\nexists \mu_i \geq 0$, $i \in I$ amire $\nabla f(\bar{x}) + \sum_{i \in I} \mu_i \nabla g_i(\bar{x})$ merőleges $\text{lin}(C)$-re. A Farkas lemma szerint ekkor $\exists v \in \text{lin}(C)$:

$$\begin{aligned} v^T \nabla g_i(\bar{x}) &\leq 0 \qquad (i \in I), \\ v^T \nabla f(\bar{x}) &< 0. \end{aligned}$$

Legyen x^s Slater pont. Ekkor $g_i(x^s) \leq 0$ $\forall i$, így

- ha $i \in I$: $\underbrace{g_i(x^s)}_{\leq 0} - \underbrace{g_i(\bar{x})}_{=0} \leq 0 \underbrace{\Rightarrow}_{g_i \text{ konvex}} \nabla g_i(\bar{x})(x^s - \bar{x}) \leq 0$,
- ha $i \in I$ és g_i nemlineáris: $g_i(x^s) < 0 \Rightarrow \nabla g_i(\bar{x})(x^s - \bar{x}) < 0$.

Legyen $s = v + \delta(x^s - \bar{x})$, $\delta > 0$. Ekkor tudjuk:

- $s \in \text{lin}(C)$,
- $i \in I$ esetén $\nabla g_i(\bar{x}) s^T \leq 0$,
- ha $i \in I$ és g_i nemlineáris: $\nabla g_i(\bar{x}) s^T < 0$,
- elég kicsi δ-ra $\nabla f(\bar{x}) s^T < 0$, mert $\nabla f(\bar{x}) v^T < 0$.

Válasszuk δ-t ilyen kicsinek. Az így kapott s segítségével találhatunk egy $\bar{x}$-nál kisebb értékű megengedett pontot. Elég kis $\varepsilon > 0$-ra a következők mind teljesülnek:

- $\bar{x} + \varepsilon s \in C$ (mert C relatív nyílt),
- $f(\bar{x} + \varepsilon s) < f(\bar{x})$,
- $i \in I$, g_i nemlineáris: $g_i(\bar{x} + \varepsilon s) < g_i(\bar{x}) = 0$,
- $i \in I$, g_i lineáris: $g_i(\bar{x} + \varepsilon s) \leq g_i(\bar{x}) = 0$,
- ha $i \notin I$: $g_i(\bar{x} + \varepsilon s) < 0$ mert $g_i(\bar{x}) < 0$ és g_i folytonos függvény.

Tehát $\bar{x} + \varepsilon s \in C_g$ és $f(\bar{x} + \varepsilon s) < f(\bar{x})$, azaz $\bar{x}$ nem optimális. □

4.4.2. Lagrange duális

A következőkben a lineáris programozás dualitás-tételének konvex általánosítását fogjuk bebizonyítani a Karush-Kuhn-Tucker tétel segítségével. Ehhez először definiáljuk a Lagrange függvényt, amit úgy is tekinthetünk, hogy a $g_i(x) \leq 0$ feltételek megsértését valamilyen szorzóval (lineárisan) büntetjük. A linearitás miatt viszont jutalmazni kell, ha a feltételek bőven teljesülnek.

Definíció (Lagrange függvény)**.** $x \in C$-re és $\mu \in \mathbb{R}^m_+$-re

$$L(x, \mu) := f(x) + \sum_{i=1}^{m} \mu_i g_i(x)$$

A Lagrange függvény tulajdonságai:

i) Ha $x \in C_g$ akkor $L(x, \mu) \leq f(x)$;

ii) Ha $x \in C \setminus C_g$ akkor $\exists \mu$: $L(x, \mu) > f(x)$;

Adott μ-re legyen $L(\mu) := \inf\{L(x, \mu) : x \in C\}$. Az első tulajdonságból következik, hogy $L(\mu) \leq \inf_{x \in C_g} f(x)$. Az $L(\mu)$ érték kiszámolása egy feltétel nélküli konvex optimalizálási feladat.

Definíció. A $\sup_{\mu\in\mathbb{R}^m_+} L(\mu)$ feladatot **Lagrange duális feladatnak** nevezzük.

Feladat. *Mutassuk meg, hogy a Lagrange duális feladat is egy konvex optimalizálási feladat.*

4.4.3. Tétel (Gyenge dualitás)**.** *Ha $L(\bar\mu) = f(\bar x)$, $\bar\mu \in \mathbb{R}^m_+$, $\bar x \in C_g$, akkor $\bar x$ optimális megoldása a (KO) feladatnak, és $\bar\mu$ optimális megoldása a Lagrange duális feladatnak.*

Bizonyítás. $L(\mu) \le f(\bar x) = L(\bar\mu) \le f(x)$ tetszőleges $\mu \in \mathbb{R}^m_+$-ra és $x \in C_g$-re. □

4.4.4. Tétel (Erős dualitás)**.** *Ha C relatív nyílt, a (KO) feladat Slater reguláris és $\bar x$ optimális megoldás, akkor $\exists\bar\mu \in \mathbb{R}^m_+$, amire $L(\bar\mu) = f(\bar x)$.*

Bizonyítás. Ha $\bar x$ optimális, akkor a Karush-Kuhn-Tucker tétel szerint $\exists\bar\mu \in \mathbb{R}^m_+$, amire

(1) $\nabla f(\bar x) + \sum_{i=1}^m \bar\mu_i \nabla g_i(\bar x)$ merőleges $\mathrm{lin}(C)$-re;

(2) ha $g_i(\bar x) < 0$, akkor $\bar\mu_i = 0$.

Először belátjuk, hogy $\bar x$ optimális megoldása a $\min_{x\in C} L(x, \bar\mu)$ feladatnak, azaz $L(\bar\mu) = L(\bar x, \bar\mu)$.

A $\min_{x\in C} L(x, \bar\mu)$ feladat konvex függvény minimalizálása relatív nyílt halmazon. Tudjuk, hogy $\bar x$ optimális $\Leftrightarrow \nabla_{x=\bar x} L(x, \bar\mu)$ merőleges $\mathrm{lin}(C)$-re.

$$\nabla_{x=\bar x} L(x, \bar\mu) = \nabla_{x=\bar x}(f(x) + \sum_{i=1}^m \bar\mu_i g_i(x)) = \nabla f(\bar x) + \sum_{i=1}^m \bar\mu_i \nabla g_i(\bar x)$$

ami merőleges $\mathrm{lin}(C)$-re (1) szerint.

Ezután belátjuk, hogy $L(\bar x, \bar\mu) = f(\bar x)$. Definíció szerint $L(\bar x, \bar\mu) = f(\bar x) + \sum_{i=1}^m \bar\mu_i g_i(\bar x)$. De $\sum_{i=1}^m \bar\mu_i g_i(\bar x) = 0$, mert

- (2) szerint: ha $g_i(\bar x) < 0$, akkor $\bar\mu_i = 0$;
- $\bar x \in F \Rightarrow g_i(\bar x) \le 0 \ \forall i \in \{1, \dots, m\}$.

Tehát beláttuk, hogy $L(\bar\mu) = L(\bar x, \bar\mu) = f(\bar x)$. □

Megmutatjuk, hogy lineáris programozási feladat esetén ez a tétel tényleg az erős LP dualitásnak felel meg. Tekintsük az $R := \{x : Qx \le b\}$ nemüres poliédert, ahol $Q \in \mathbb{R}^{m\times n}$, és legyen $c \in \mathbb{R}^n$ a maximalizálandó célfüggvény. A mostani jelöléseinkkel $f(x) = -c^T x$ és $g_i(x) = q_{i.}x - b_i$, tehát

$$L(\mu) = \inf_{x\in\mathbb{R}^n} -c^T x + \mu^T(Qx - b) = \begin{cases} -\infty \text{ ha } \mu^T Q \ne c^T, \\ -\mu^T b \text{ ha } \mu^T Q = c^T. \end{cases}$$

A feladat Slater-reguláris, mert R nemüres és minden g_i lineáris. Ha $\bar x$ optimális megoldás, akkor a fenti 4.4.4 tétel szerint létezik $\bar\mu \in \mathbb{R}^m_+$, amire $L(\bar\mu) = f(\bar x)$. Ez pont azt jelenti, hogy $\bar\mu^T Q = c^T$, tehát $\bar\mu$ megoldása a duális feladatnak, és $\bar\mu^T b = c^T x$.

4.5. Megoldási módszerek

Konvex programozási feladatokat általában nem lehet a szimplex módszerhez hasonló módon megoldani, mert az optimum nem feltétlenül éretik el extremális pontban, és ha igen, akkor is lehet végtelen sok extremális pont. Ennek ellenére a konvexitás sokat segít a megoldhatóságban, és fontos speciális eseteket (például a szemidefinit feladatokat) meg lehet oldani polinom időben. Az alábbiakban néhány egyszerű általános módszert ismertetünk, amik konvergálnak az optimumhoz.

4.5.1. Megengedett csökkenési irány keresése

Definíció. Az s vektor **megengedett csökkenési irány** x^0-ban, ha $\exists\varepsilon > 0$, hogy tetszőleges $0 < \lambda \leq \varepsilon$-ra $f(x^0 + \lambda s) < f(x^0)$ és $x^0 + \lambda s \in F$.

Megengedett csökkenési irány x^0-ban a következő LP feladat megoldásával kereshető:

$$
\begin{aligned}
\max u & \\
\nabla f(x^0)^T s + u \leq 0 & \\
\nabla g_i(x^0)^T s + u \leq 0 & \qquad \text{ha } g_i(x^0) = 0 \text{ és } g_i \text{ nem lineáris,} \\
\nabla g_i(x^0)^T s \leq 0 & \qquad \text{ha } g_i(x^0) = 0 \text{ és } g_i \text{ lineáris,} \\
u \geq 0 & \\
-1 \leq s_j \leq 1 & \qquad (j = 1, \ldots, n), \\
s \in \operatorname{lin}(C) &
\end{aligned}
$$

4.5.1. Állítás. *Ha $\exists(s,u)$ megoldás, ahol $u > 0$, akkor s megengedett csökkenési irány.*

Bizonyítás. $\nabla f(x^0)^T s < 0$, tehát s csökkenési irány. Másrészt $\nabla g_i(x^0)^T s \leq 0$ ha $g_i(x^0) = 0$, és $\nabla g_i(x^0)^T s < 0$ ha g_i nemlineáris, tehát s megengedett irány. $\square$

4.5.2. Állítás. *Ha a feladat Slater reguláris, akkor ha x^0 nem optimális, akkor $\exists(s,u)$ megoldás, amire $u > 0$.*

Bizonyítás. Ha x^0 nem optimális, akkor $\exists v \in \operatorname{lin}(C)$:

$$
\begin{aligned}
v^T \nabla g_i(\bar{x}) &\leq 0 \qquad (i = 1, \ldots, m), \\
v^T \nabla f(\bar{x}) &< 0.
\end{aligned}
$$

Legyen x^s Slater pont. Ekkor $g_i(x^s) \leq 0 \ \forall i$, így

- ha $g_i(\bar{x}) = 0$: $\underbrace{g_i(x^s)}_{\leq 0} - \underbrace{g_i(\bar{x})}_{=0} \leq 0 \underbrace{\Rightarrow}_{g_i \text{ konvex}} \nabla g_i(\bar{x})(x^s - \bar{x}) \leq 0.$

- ha $g_i(\bar{x}) = 0$ és g_i nemlineáris: $g_i(x^s) < 0 \Rightarrow \nabla g_i(\bar{x})(x^s - \bar{x}) < 0$.

Legyen $s = v + \delta(x^s - \bar{x})$, $\delta > 0$. Ekkor tudjuk:

- $s \in \mathrm{lin}(C)$;
- ha $g_i(\bar{x}) = 0$, akkor $\nabla g_i(\bar{x}) s^T \leq 0$;
- ha $g_i(\bar{x}) = 0$ és g_i nemlineáris, akkor $\nabla g_i(\bar{x}) s^T < 0$;
- elég kicsi δ-ra $\nabla f(\bar{x}) s^T < 0$, mert $\nabla f(\bar{x}) v^T < 0$.

Válasszuk δ-t elég kicsinek az utolsó feltételnek megfelelően. Szorozzuk meg s-t egy skalárral úgy, hogy $-1 \leq s_j \leq 1$ $\forall j$. Ehhez az s-hez választhatunk megfelelő $u > 0$-t. □

4.5.2. Gradiens módszer

Legyen $f : \mathbb{R}^n \to \mathbb{R}$ konvex, folytonosan differenciálható függvény. Ennek szeretnénk a minimumát megtalálni. Tegyük fel, hogy $x^0 \in \mathbb{R}^n$ pontban vagyunk.

4.5.3. Állítás. *x^0-ban az iránymenti derivált a $-\nabla f(x^0)$ irányban a legkisebb (azonos normájú vektorok közül).*

Bizonyítás. s irányban az iránymenti derivált: $\nabla f(x^0)^T s$.

$$-\nabla f(x^0)^T \nabla f(x^0) = \min_{||s|| = ||\nabla f(x^0)||} \nabla f(x^0)^T s$$

Tehát $-\nabla f(x^0)$ a legmeredekebb csökkenési irány. □

Algoritmus

$x^0 \in \mathbb{R}^n$, $k = 0$.

1. Kiszámoljuk $s = -\nabla f(x^k)$-t. Ha $s = 0$, akkor kész vagyunk.
2. Legyen $\lambda^k = \mathrm{argmin}_{\lambda \in \mathbb{R}_+} f(x^k + \lambda s)$.
3. Legyen $x^{k+1} = x^k + \lambda^k s$.
4. $k := k + 1$, és tovább az 1. lépésre.

4.5.4. Állítás. $f(x^{k+1}) < f(x^k)$

Bizonyítás. s irányban az iránymenti derivált: $-\nabla f(x^k)^T \nabla f(x^k) < 0$. □

4.5.5. Tétel. *Ha $D = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ kompakt, akkor az $x^0, x^1, x^2, \ldots$ tetszőleges $\bar{x}$ torlódási pontja optimális megoldása a feladatnak.*

Bizonyítás. Vegyünk egy $x^{j_1}, x^{j_2}, \dots \to \bar{x}$ konvergens részsorozatot. Mivel f folytonos, ezért $\lim_{i\to\infty} f(x^{j_i}) = f(\bar{x})$ és $\lim_{i\to\infty} \nabla f(x^{j_i}) = \nabla f(\bar{x})$. Legyen $s = -\nabla f(\bar{x})$. Ekkor $\nabla f(\bar{x})^T s \leq 0$ és $= 0 \Leftrightarrow s = 0$.

Másrészt: tetszőleges $\lambda \geq 0$-ra és i-re $f(x^{j_{i+1}}) \leq f(x^{j_i+1}) \leq f(x^{j_i} - \lambda \nabla f(x^{j_i}))$. Az $i \to \infty$ határátmenet mutatja, hogy $f(\bar{x}) \leq f(\bar{x}+\lambda s)$ tetszőleges $\lambda \geq 0$-ra. Tehát $\frac{f(\bar{x}+\lambda s)-f(\bar{x})}{\lambda} \geq 0$. Ha $\lambda \to 0$, akkor azt kapjuk, hogy $\nabla f(\bar{x})^T s \geq 0 \Rightarrow s = 0 \Rightarrow \nabla f(\bar{x}) = 0 \Rightarrow \bar{x}$ optimális. $\square$

4.5.3. Aranymetszés módszer

A fenti módszerben a $\lambda^k = \operatorname{argmin}_{\lambda\in\mathbb{R}_+} f(x^k + \lambda s)$ kiszámolásához meg kell oldanunk egy egydimenziós konvex függvény minimalizálási feladatot. Persze ezt is csak közelítőleg tudjuk megoldani. Egy lehetséges megoldási módszer az Aranymetszés módszer, ami nem csak differenciálható függvényekre alkalmazható.

Legyen f konvex függvény $\mathbb{R}$-en. Az algoritmus egyre csökkenő méretű intervallumokat ad, amik garantáltan tartalmaznak optimális megoldást. Minden lépésben adott lesz $\alpha_i < \beta_i < \gamma_i < \delta_i$, amikre $f(\alpha_i) \geq f(\beta_i)$, $f(\gamma_i) \leq f(\delta_i)$, és

$$\frac{\gamma_i - \alpha_i}{\delta_i - \alpha_i} = \frac{\delta_i - \beta_i}{\delta_i - \alpha_i} = q := \frac{\sqrt{5}-1}{2}.$$

Kiindulásként $i = 0$-ra találunk ilyeneket. A konvexitásból következik, hogy van optimális megoldás az $[\alpha_i, \delta_i]$ intervallumban.

Az aranymetszés szabálya miatt az is teljesül, hogy

$$\frac{\beta_i - \alpha_i}{\gamma_i - \alpha_i} = \frac{\delta_i - \gamma_i}{\delta_i - \beta_i} = q.$$

Az általános lépésben összehasonlítjuk az $f(\beta_i)$ és $f(\gamma_i)$ értékeket. Két eset van:

- Ha $f(\beta_i) < f(\gamma_i)$: legyen

$$\begin{aligned}
\alpha_{i+1} &= \alpha_i \\
\beta_{i+1} &= q\alpha_i + (1-q)\gamma_i \\
\gamma_{i+1} &= \beta_i \\
\delta_{i+1} &= \gamma_i
\end{aligned}$$

- Ha $f(\beta_i) \geq f(\gamma_i)$: legyen

$$\begin{aligned}
\alpha_{i+1} &= \beta_i \\
\beta_{i+1} &= \gamma_i \\
\gamma_{i+1} &= q\delta_i + (1-q)\beta_i \\
\delta_{i+1} &= \delta_i
\end{aligned}$$

Könnyen ellenőrizhető, hogy az új pontok is teljesítik a feltételeket, az $[\alpha_k, \delta_k]$ intervallum hossza q-szorosára csökken, és csak egy új függvényértéket kell kiszámolni.

4.5.4. Newton módszer

Tegyük fel, hogy $f : \mathbb{R}^n \to \mathbb{R}$ kétszer folytonosan differenciálható konvex függvény.

Legyen f másodrendű közelítése:

$$q_{x^0}(x) = f(x^0) + \nabla f(x^0)^T(x - x^0) + \frac{1}{2}(x - x^0)^T \nabla^2 f(x^0)(x - x^0).$$

Ekkor $\nabla^2 q_{x^0}(x) = \nabla^2 f(x^0)$ pozitiv szemidefinit, mert f konvex, tehát $q_{x^0}(x)$ konvex. Minimalizáljuk $q_{x^0}(x)$-et:

$$\bar{x} \text{ optimális} \Leftrightarrow \underbrace{\nabla q_{x^0}(\bar{x})}_{=\nabla f(x^0)+\nabla^2 f(x^0)(\bar{x}-x^0)} = 0$$

Tegyük fel, hogy $\nabla^2 f(x^0)$ invertálható:

$$\bar{x} \text{ optimális} \Leftrightarrow \bar{x} = x^0 - (\nabla^2 f(x^0))^{-1} \nabla f(x^0).$$

Algoritmus. Tegyük fel, hogy f szigorúan konvex (tehát $\nabla^2 f(x)$ pozitiv definit). $x^0 \in \mathbb{R}^n$, $k = 0$.

1. Kiszámoljuk $\nabla f(x^k)$-t és $\nabla^2 f(x^k)$-t. Ha $\nabla f(x^k) = 0$, akkor kész vagyunk.
2. $x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k)$.
3. $k := k + 1$ és tovább az 1. lépésre.

Megjegyzés. Az egydimenziós esetben $x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$.

Megjegyzés. Ha $\nabla^2 f(x^k)$ nem invertálható, akkor vegyük helyette $\nabla^2 f(x^k) + \alpha I$-t valamilyen megfelelően választott α-ra.