Jegyzet

az Operációkutatás (elemző, programozó matematikus) tárgyhoz

Fábián Csaba, Király Tamás, Papp Olga

2015. április

Tartalomjegyzék

1. A lineáris programozási feladat

1.1. Bevezetés

Egy olajfeldolgozó üzemben kétféle nyersolaj áll rendelkezésre: az A típusból 8 millió hordó, a B típusból 5 millió. Ezekből készítenek benzint és gázolajat. Az üzemben három technológiai eljárás közül lehet választani. Az első eljárás bemeneti-kimeneti arányait az jellemzi, hogy 3 hordó A-kőolajból és 5 hordó B-kőolajból 4 hordó benzint és 3 hordó gázolajat állít elő. A második eljárás 1 hordó A-ból és 1 hordó B-ből készít 1 hordó benzint és 1 hordó gázolajat, míg a harmadik eljárásnál ezek az értékek rendre 5, 3 és 3, 4. Tudván, hogy a benzin hordójáért 4 dollárt, a gázolaj hordójáért 3 dollárt kapunk, a meglévő nyersolaj készletet miképp osszuk fel a három eljárás között, ha célunk az össz-bevétel maximalizálása? (Egyszerűség kedvéért nem vesszük most tekintetbe az eljárások esetleg eltérő üzemi költségeit).

Jelölje x_i $(i = 1, 2, 3)$ azt, hogy az egyes eljárásokat milyen mértékben használjuk. x_1 tehát azt jelenti, hogy az első eljárással $3x_1$ A-olajat és $5x_1$ B-olajat dolgozunk fel, és ennek során $4x_1$ benzint és $3x_1$ gázolajat kapunk. Az x_i értékeknek természetesen nemnegatívnak kell lenniük. Az adatok alapján az A-olajból $3x_1 + x_2 + 5x_3$ hordót használunk, és így ez az összeg legfeljebb 8 millió. A B-olajra az $5x_1 + x_2 + 3x_3 \leq 5000000$ egyenlőtlenség adódik.

Az eljárásokkal benzinből összesen $4x_1 + x_2 + 3x_3$ hordó áll elő, melynek értéke $4(4x_1 + x_2 + 3x_3)$ dollár. Gázolajból $3x_1 + x_2 + 4x_3$ hordót nyerünk, melynek értéke $3(3x_1 + x_2 + 4x_3)$. Az összbevételünk tehát $25x_1 + 7x_2 + 24x_3$ dollár. Feladatunk maximalizálni a $25x_1 + 7x_2 + 24x_3$ célfüggvényt az $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$ valamint a $3x_1 + x_2 + 5x_3 \leq 8000000$ és $5x_1 + x_2 + 3x_3 \leq 5000000$ feltételek mellett. (Mivel x_i ebben a modellben a hordók számát jelöli, így ki kellene kötnünk, hogy minden x_i egész. A fenti feladatban azonban a hordók száma nagy, így gyakorlati szempontból nem számít, ha elengedjük az egészértékű megkötést. Jelezzük ugyanakkor, hogy számos gyakorlati problémában szükséges lehet a változókra tett egészértékűségi megkötés. Lineáris egyenlőtlenség-rendszerek egészértékű megoldhatóságával az *egészértékű programozás* foglalkozik.)

A lineáris algebra egyik kiinduló pontja a lineáris egyenletrendszerek vizsgálata volt. A Gauss-elimináció segítségével elvi és algoritmikus választ kaptunk arra a kérdésre, hogy egy lineáris egyenletrendszernek mikor van megoldása. A **lineáris programozás** lineáris egyenlőtlenség-rendszerekkel foglalkozik. Egy egyenlőtlenség lehet szigorú vagy egyenlőséget is megengedő, de a továbbiakban egyenlőtlenségen mindig ezen utóbbit értjük. A legelső kérdés az, hogy egy egyenlőtlenség-rendszernek mikor létezik megoldása, vagy másképp fogalmazva, egy egyenlőtlenség-rendszer megoldás-halmaza, melyet majd poliédernek nevezünk, mikor nemüres. Az erre vonatkozó eredmény (Farkas lemma) az egyenletrendszerekről szóló Fredholm tétel direkt általánosítása.

Egyenlőtlenség-rendszerekkel kapcsolatban azonban olyan új típusú kérdések is felvetődnek, amelyeknek nincs is értelmes speciális esetük egyenletrendszerekre vonatkozólag. Megkérdezhetjük, hogy valamely c vektorra a cx lineáris célfüggvény korlátos-e a P poliéderen (mondjuk) felülről. (Egy affin altéren egy lineáris célfüggvény vagy konstans vagy nem korlátos). Ha korlátos, úgy harmadik célunk meghatározni a cx függvény maximumát (vagy ha alulról korlátos, úgy minimumát) P-n. Persze most még azt a (később majd bizonyításra kerülő) tényt sem tudjuk, hogy a szóbanforgó maximum egyáltalán létezik-e: Weierstrass általános tétele szerint egy korlátos zárt halmazon folytonos függvény felveszi maximumát, így miután cx folytonos és egy poliéder bizonyosan zárt, P korlátossága esetén már most is bizonyosak lehetünk a maximum létezésében. Nemsokára ezt is és a nem-korlátos esetet is igazoljuk, Weierstrass nélkül.

1.2. Lineáris programozási feladat (LP) sztenderd alakja

Bár a lineáris programozás általános egyenlőtlenség-rendszerekkel foglakozik, a továbbiakban speciális, úgynevezett sztenderd alakú feladatot nézünk.

1.1. Definíció. A **sztenderd alakú feladatban** adott egy $A \in \mathbb{R}^{m \times n}$ mátrix ahol $m \leq n$, egy $c^T \in \mathbb{R}^n$ és egy $b \in \mathbb{R}^m$ vektor. Keresünk olyan $x \in \mathbb{R}^n$ vektort, ami kielégíti az

$$Ax = b$$

$$x \geq 0$$

feltételeket, és ezen belül cx maximális. Azt mondjuk, hogy A a feladat mátrixa, b a feladat jobboldala, c a lineáris célfüggvény, x pedig a változók vektora.

Ha A sorai lineárisan összefüggőek, akkor vagy van olyan egyenlet a rendszerben, ami következménye a többinek, tehát elhagyható, vagy $Ax = b$ nem megoldható. Ráadásul Gauss-eliminációval el tudjuk dönteni, hogy a két eset közül melyik áll fenn. Ezért a sztenderd alakba azt is beleértjük, hogy az A mátrix sorai lineárisan függetlenek, azaz A rangja m.

$$\begin{array}{cc} \begin{bmatrix} \quad & x^T & \quad \end{bmatrix} & \\ \begin{bmatrix} & & \\ \quad & A & \quad \\ & & \end{bmatrix} & \begin{bmatrix} \\ b \\ \\ \end{bmatrix} \\ \begin{bmatrix} \quad & c & \quad \end{bmatrix} & \end{array}$$

1.1. Megjegyzés. Egy $Ax \leq b$, $x \geq 0$, $\max cx$ alakú feladat könnyen átalakítható sztenderd alakúvá m darab új változó, úgynevezett **eltérésváltozó** bevezetésével. Legyen z az eltérésváltozók vektora, és tekintsük az $Ax + Iz = b$, $x \geq 0$, $\max cx$ feladatot. Ez már sztenderd alakú, és optimális megoldásai megegyeznek az eredeti feladat optimális megoldásaival.

1.2. Definíció (Megoldás)**.** Azt mondjuk, hogy $x \in \mathbb{R}^n$ vektor **megoldás**, ha $Ax = b$ a fenti LP feladatra.

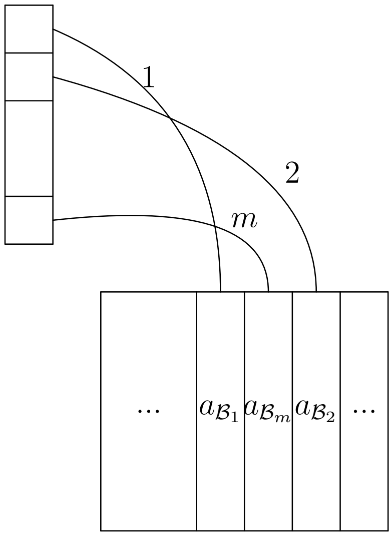
1.3. Definíció (Megengedett megoldás)**.** Azt mondjuk, hogy $x \in \mathbb{R}^n$ vektor **megengedett megoldása** a fenti LP feladatnak, ha $Ax = b$ és $x \geq 0$.

1.4. Definíció (Optimális megoldás)**.** Azt mondjuk, hogy $x \in \mathbb{R}^n$ vektor **optimális megoldás**, ha x megengedett megoldás, és tetszőleges x' megengedett megoldásra $cx' \leq cx$.

1.2. Megjegyzés. Biztos, hogy van megoldás, mert A rangja m. A oszlopai kifeszítik az $\mathbb{R}^m$ teret, így b előállítható ezen vektorok lineáris kombinációjaként. Megengedett megoldás viszont nem biztos hogy van. Ha van is megengedett megoldás, nem biztos hogy van optimális, mert lehet hogy cx akármilyen nagy lehet a megengedett megoldások halmazán.

1.5. Definíció (Bázis)**.** Bázisnak nevezünk egy olyan $\mathcal{B}$ leképezést a sorok halmazából az oszlopok halmazába, amelyre az $a_{\mathcal{B}_1}$, $a_{\mathcal{B}_2}$, ..., $a_{\mathcal{B}_m}$ oszlopvektorok lineárisan függetlenek.

1.3. Megjegyzés. A lineáris algebrától eltérően itt nem halmazt, hanem rendezett halmazt tekintünk bázisnak.



1.6. Definíció (Bázishoz tartozó bázismátrix)**.**

$$\begin{bmatrix} \\ a_{\mathcal{B}_1} & a_{\mathcal{B}_2} & \dots & & a_{\mathcal{B}_m} \\ \\ \end{bmatrix} =: B$$

1.4. Megjegyzés. A B mátrix invertálható, mert teljesrangú.

A továbbiakban tegyük fel, hogy adott a $\mathcal{B}$ bázis. Az ábrákon az ábrázolhatóság kedvéért A-ban a bázisoszlopok egymás után következnek, de valójában ez nincs mindig így.

$$A = \begin{bmatrix} \\ \dots & B & \dots & \\ \\ \end{bmatrix}$$

1.1. Jelölés. Jelöljük $x_{\mathcal{B}}$-vel az x vektor bázishoz tartozó részét.

$$(x_{\mathcal{B}})_i := x_{\mathcal{B}_i}$$

Jelöljük $c_{\mathcal{B}}$-vel a c vektor bázishoz tartozó részét.

$$(c_{\mathcal{B}})_i := c_{\mathcal{B}_i}$$

1.7. Definíció ($\mathcal{B}$ bázishoz tartozó bázismegoldás)**.** Ha az $\bar{x} \in \mathbb{R}^n$ vektorra teljesül, hogy a bázison kívüli komponensei 0-ák, a bázishoz tartozó komponensekre pedig teljesül, hogy $B\bar{x}_{\mathcal{B}} = b$, akkor $\bar{x}$-et $\mathcal{B}$ bázishoz tartozó bázismegoldásnak nevezzük.

1.5. Megjegyzés. Ilyen bázismegoldás létezik és egyértelmű: $\bar{x}_{\mathcal{B}} = B^{-1}b$. Ez a vektor a feladat megoldása, ugyanis $A\bar{x} = B\bar{x}_{\mathcal{B}} = b$. Viszont lehet, hogy nem megengedett megoldás.

1.6. Megjegyzés. Minden bázishoz egyetlen bázismegoldás tartozik, de előfordulhat, hogy két különböző bázishoz tartozó bázismegoldás ugyanaz:

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

A fenti példában két különböző bázishoz is (1,2 és 2,3 oszlopok) ugyanaz a bázismegoldás tartozik (az ábra tetején szereplő $\bar{x}$). Általában elmondható, hogy ha egy x megoldás nemnulla komponenseihez tartozó oszlopok lineárisan függetlenek, akkor x bázismegoldás, hiszen egy lineárisan független oszlophalmaz kiegészíthető bázissá.

1.8. Definíció. A $\mathcal{B}$ bázis **megengedett**, ha a hozzá tartozó bázismegoldás megengedett: $x_{\mathcal{B}} = B^{-1}b \geq 0$.

1.1. Tétel. *Ha egy sztenderd alakú feladadatnak van egy megengedett megoldása, akkor van megengedett bázismegoldása is. Ha cx felülről korlátos a megengedett megoldások halmazán, akkor az is igaz, hogy az optimális megoldások között van bázismegoldás.*

Bizonyítás. Vegyünk egy olyan $\bar{x}$ megengedett megoldást, aminek a lehető legkevesebb pozitív komponense van. Belátjuk, hogy $\bar{x}$ bázismegoldás, azaz a pozitív komponenseihez tartozó oszlopok lineárisan függetlenek. Indirekt tegyük fel, hogy nem, azaz van olyan nemnulla $z \in \mathbb{R}^n$ vektor, ami mindenhol nulla ahol $\bar{x}$ nulla, és $Az = 0$. A z vektort esetleg -1-gyel megszorozva feltehetjük, hogy z-nek van negatív komponense. Így létezik egy legnagyobb olyan $\epsilon > 0$, hogy $\bar{x} + \epsilon z$ megengedett megoldás. Mivel ϵ-t maximálisnak választottuk, $\bar{x}$-nek van olyan pozitív komponense, ami $\bar{x} + \epsilon z$-ben 0. Így $\bar{x} + \epsilon z$ egy megengedett megoldás kevesebb pozitív komponenssel, ellentmondás.

A tétel második felének bizonyításához feltesszük, hogy cx felülről korlátos a megengedett megoldások halmazán. Megmutatjuk, hogy minden x megengedett megoldáshoz van olyan $\bar{x}$ megengedett bázismegoldás, amire $c\bar{x} \geq cx$. Ebből már következik az állítás, hiszen megengedett bázismegoldásból csak véges sok lehet, így van köztük optimális.

Legyen tehát x egy megengedett megoldás, és legyen $\bar{x}$ olyan megengedett megoldás, amire $c\bar{x} \geq cx$, és ezen belül a lehető legkevesebb pozitív komponense van. Belátjuk, hogy $\bar{x}$ bázismegoldás, azaz a pozitív komponenseihez tartozó oszlopok lineárisan függetlenek. Indirekt tegyük fel, hogy nem, azaz van olyan $z \in \mathbb{R}^n$ nemnulla vektor, ami mindenhol nulla ahol $\bar{x}$ nulla, és $Az = 0$.

Ha $cz = 0$, akkor a z vektort esetleg -1-gyel megszorozva feltehetjük, hogy z-nek van negatív komponense. Ha $cz \neq 0$, akkor a z vektort esetleg -1-gyel megszorozva feltehetjük, hogy $cz > 0$. Mivel a célfüggvény felülről korlátos a megengedett megoldások halmazán, z-nek szükségképp van negatív komponense. Összefoglalva, mindkét esetben találtunk egy $z \in \mathbb{R}^n$ nemnulla vektort a következő tulajdonságokkal:

- z mindenhol nulla ahol $\bar{x}$ nulla
- $Az = 0$
- $cz \geq 0$
- z-nek van negatív komponense.

Utóbbi miatt létezik egy legnagyobb olyan $\epsilon > 0$, hogy $\bar{x} + \epsilon z$ megengedett megoldás. Mivel ϵ-t maximálisnak választottuk, $\bar{x}$-nek van olyan pozitív komponense, ami $\bar{x} + \epsilon z$-ben 0. Így $\bar{x} + \epsilon z$ egy megengedett megoldás kevesebb pozitív komponenssel, ráadásul $c(\bar{x} + \epsilon z) = c\bar{x} + \epsilon cz \geq c\bar{x}$, ellentmondás. □

Bázishoz tartozó kanonikus alak

Tegyük most fel, hogy adott egy megengedett $\mathcal{B}$ bázisunk. Megmutatjuk, hogy az feladatunknak van egy olyan ekvivalens alakja, ahol egyrészt a bázismátrix helyén az egységmátrix van, másrészt a célfüggvényben a bázisváltozóknak 0 az együtthatója. Az előbbi tulajdonság úgy érhető el, hogy az egyenletrendszert balról megszorozzuk B^{-1}-zel:

$$B^{-1}A = \begin{bmatrix} & & \\ \ldots & I & \ldots \\ & & \end{bmatrix}$$

A továbbiakban ezt a transzformált mátrixot fogjuk $\bar{A}$-val jelölni, a transzformált jobboldalt, azaz $B^{-1}b$-t pedig $\bar{b}$-vel. Egy $x \in \mathbb{R}^n$ vektor pontosan akkor megoldása $\bar{A}x = \bar{b}$-nek, ha megoldása $Ax = b$-nek.

A c célfüggvény helyett pedig tekintsük a

$$\bar{c} = c - c_{\mathcal{B}}\bar{A}$$

transzformált célfüggvényt. Ez olyan értelemben ekvivalens az eredetivel, hogy tetszőleges x megoldás esetén $cx - \bar{c}x = c_{\mathcal{B}}\bar{A}x = c_{\mathcal{B}}\bar{b}$, azaz a két célfüggvény-érték közti különbség nem függ x-től. Ez azt jelenti, hogy c-re nézve ugyanazok az optimális megoldások, mint $\bar{c}$-re nézve (bár maga az optimum-érték különbözik). Könnyű ellenőrizni, hogy $\bar{c}_{\mathcal{B}} = \mathbf{0}$.

1.9. Definíció. A feladat $\mathcal{B}$ **bázishoz tartozó kanonikus alakja** az $\bar{A}x = \bar{b}$, $x \geq 0$, $\max \bar{c}x$ feladat. Ennek a feladatnak ugyanazok a megengedett és optimális megoldásai, mint az eredeti feladatnak.

1.7. Megjegyzés. Az 1.1 Megjegyzésben eltérésváltozók bevezetésével kaptunk sztenderd alakú feladatot. Ez pont az eltérésváltozók által alkotott bázishoz tartozó kanonikus alak, hiszen az eltérésváltozók oszlopai egységmátrixot alkotnak, és a célfüggvényben minden eltérésváltozó együtthatója 0.

1.2. Tétel. *Ha $\mathcal{B}$ megengedett bázis, és a hozzá tartozó kanonikus alakban $\bar{c} \leq 0$, akkor $\bar{x}$ optimális megoldása a feladatnak.*

Bizonyítás. Figyeljük meg, hogy $\bar{c}\bar{x} = 0$, mert $\bar{c}$ a bázisban szereplő komponensekben 0, $\bar{x}$ pedig a bázisban nem szereplő komponensekben. Másrészt tetszőleges megengedett x' megoldásra $\bar{c}x' \leq 0$, hiszen $\bar{c} \leq 0$ és $x' \geq 0$. Ezek alapján $\bar{x}$ optimális megoldása a feladatnak. □

Az alábbiakban ismertetjük a szimplex módszert a sztenderd alakú LP feladat megoldására. A szimplex módszer egy megengedett bázisból indul ki, és minden lépésben megpróbál egy olyan megengedett bázisba átlépni, ahol nagyobb a bázismegoldás célfüggvényértéke. Akkor áll le, ha eljut egy bázishoz ahol $\bar{c} \leq 0$ (amikor is az előző tétel értelmében az aktuális bázismegoldás optimális), vagy ha talál egy félegyenest ami mentén akármilyen nagyra lehet növelni a célfüggvényt.

2. A szimplex módszer

2.1. A szimplex módszer alaplépése

Tegyük fel, hogy a $\mathcal{B}$ megendedett bázisban vagyunk. Vegyünk egy tetszőleges j oszlopot, ami nem tartozik a bázishoz (ekkor $x_j = 0$, mert $j \notin \mathcal{B}$). Kezdjük el növelni x_j-t, és a báziskomponenseket igazítsuk úgy, hogy az $\bar{A}x = \bar{b}$ egyenlőség megmaradjon (ez ekvivalens azzal, hogy az $Ax = b$ egyenlőség megmarad). Meg fogjuk mutatni, hogy igazíthatunk, és hogy ez az igazítás egyértelmű lesz.

Kezdeti megoldás

$$0\bar{a}_j + \sum_{i=1}^{m} x_{\mathcal{B}_i}\bar{a}_{\mathcal{B}_i} = \bar{b}$$

Mivel $\bar{a}_{\mathcal{B}_i} = e_i$, a kezdeti megoldás $\bar{x}_{\mathcal{B}_i} = \bar{b}_i$. Növeljük x_j-t 0-ról ϑ-ra.

$$\vartheta\bar{a}_j + \sum_{i=1}^{m}(\bar{b}_i - \vartheta\bar{a}_{ij})e_i = \bar{b}$$

Láthatjuk, hogy az utánaigazítás lehetséges és egyértelmű. Ha x_j-t ϑ-ra növeljük, akkor $x_{\mathcal{B}_i}$-t a következőképp kell változtatni:

$$x'_{\mathcal{B}_i} := \bar{x}_{\mathcal{B}_i} - \vartheta\bar{a}_{ij} \ (= \bar{b}_i - \vartheta\bar{a}_{ij})$$

Javítunk ezzel a lépéssel a célfüggvényen?

Változó	Eredeti érték	Új érték	célfüggvény változás
j. változó	0	ϑ	$+\vartheta c_j$
megoldás i. báziskomponense	$\bar{b}_i$	$\bar{b}_i - \vartheta\bar{a}_{ij}$	$-\vartheta c_{\mathcal{B}_i}\bar{a}_{ij}$

Összegezve a változást: $\vartheta c_j - \sum_{i=1}^{m} \vartheta c_{\mathcal{B}_i}\bar{a}_{ij} = \vartheta c_j - \vartheta c_{\mathcal{B}}\bar{a}_j$
Tehát akkor érdemes az alaplépést elvégezni, ha $c_j - c_{\mathcal{B}}\bar{a}_j > 0$, azaz $\bar{c}_j > 0$.

2.1. Megjegyzés. Ezt könnyű végigvenni az egyes nembázisbeli oszlopokkal: ha az eredmény > 0, akkor javító oszlopvektort találtunk.

2.1. Definíció (Javító-oszlop)**.** Azt mondjuk, hogy a j-edik oszlopvektor javító oszlop, ha $\bar{c}_j > 0$, azaz $c_{\mathcal{B}}\bar{a}_j < c_j$.

Ha olyan állapotba érünk, hogy nincs javító oszlopvektor, akkor az eljárás leáll. A 1.2 Tétel szerint optimális megoldást találtunk.

A továbbiakban tegyük fel, hogy a p-edik oszlopvektor javítóoszlop. Meddig lehet növelni a ϑ-t?

Az alaplépésben $\vartheta\bar{a}_p + \sum_{i=1}^{m}(x_{\mathcal{B}_i} - \vartheta\bar{a}_{ip}) = \bar{b}$, ahol $x_{\mathcal{B}_i} = \bar{b}_i$ és $\bar{a}_{\mathcal{B}_i} = e_i$. A célfüggvény növekedés pedig: $\vartheta\bar{c}_p$. ϑ tehát addig növelhető, amíg $\bar{b}_i - \vartheta\bar{a}_{ip}$ sehol sem válik negatívvá, azaz a megoldás megengedett marad.

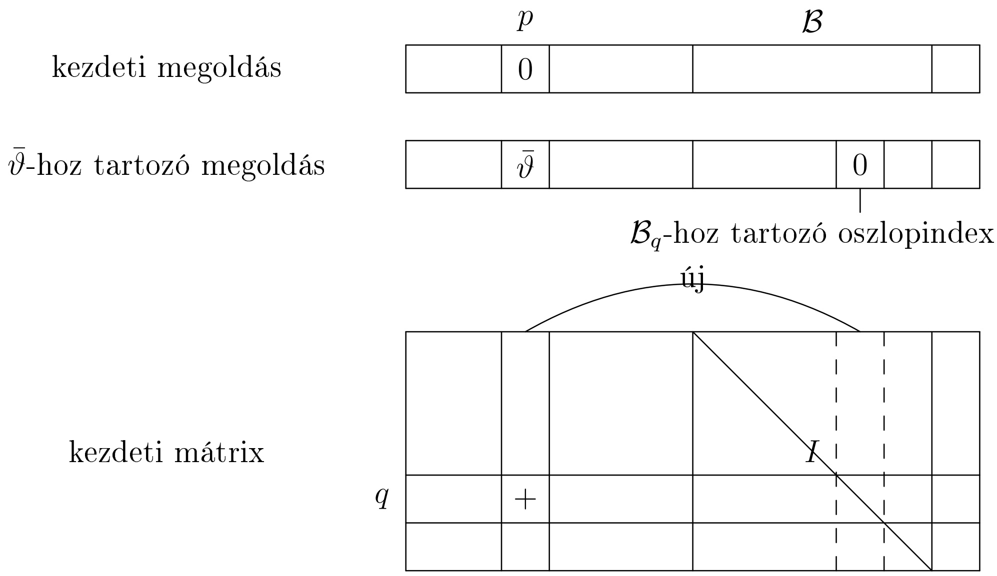
- Ha $\bar{a}_{ip} \leq 0$ valamely i. sorindexre, akkor a $(\bar{b}_i - \vartheta\bar{a}_{ip})$ nem fog csökkenni. Az ilyen i-k nem jelentenek korlátot.
- Ha $\bar{a}_{ip} > 0$, akkor $\vartheta \leq \frac{\bar{b}_i}{\bar{a}_{ip}}$

A legnagyobb megengedett ϑ így: $\vartheta = \min_{\bar{a}_{ip}>0} \frac{\bar{b}_i}{\bar{a}_{ip}}$

2.2. Megjegyzés. Ha $\bar{a}_{ip} \leq 0 \ \forall i = 1..m$, akkor ϑ minden határon túl növelhető. Ezért a célfüggvény is minden határon túl növelhető lesz. **Az eljárás leáll.**

A továbbiakban feltesszük, hogy az $\bar{a}_p$ oszlopvektornak vannak pozitív komponensei.
Legyen $1 \leq q \leq n$ olyan sorindex, amelyre $\frac{\bar{b}_q}{\bar{a}_{qp}} \leq \frac{\bar{b}_i}{\bar{a}_{ip}}$ minden olyan i sorindexre, ahol $\bar{a}_{ip} > 0$.
Válasszuk ϑ-t a lehető legnagyobbra: $\bar{\vartheta} = \frac{\bar{b}_q}{\bar{a}_{qp}}$.
Nézzük meg az alaplépés hatását az adott p, q-ra. Az x_q változó értéke lemegy 0-ra:

$$\bar{b}_q - \frac{\bar{b}_q}{\bar{a}_{qp}}\bar{a}_{qp} = 0$$



Az új bázis:

$$\mathcal{B}'_i = \begin{cases} \mathcal{B}_i & \text{ha } i \neq q \\ p & \text{ha } i = q \end{cases}$$

Ez valóban bázis.
Az alábbi oszlopok lineárisan függetlenek:

$$a_{\mathcal{B}'_1}, \ldots, a_{\mathcal{B}'_q}, \ldots, a_{\mathcal{B}'_m}$$

azaz

$$a_{\mathcal{B}_1}, \ldots, a_p, \ldots, a_{\mathcal{B}_m}$$

Bizonyítás. Legyenek $\lambda_1, \ldots, \lambda_q, \ldots \lambda_m$ olyan súlyok, hogy $\sum \lambda_i \bar{a}_{\mathcal{B}'_i} = 0$. Az $\bar{a}_{\mathcal{B}_1}, \ldots, \bar{a}_p, \ldots, \bar{a}_{\mathcal{B}_m}$ -ből csak a q. egységvektor hiányzik (e_q) – ehelyett vettük be $\bar{a}_p$-t. A q sorban a vektorok poziciói mind 0-k, kivéve $\bar{a}_{qp} > 0$. Ebből következik, hogy $\lambda_q = 0$. A többi vektor együtthatója is 0, hiszen azok eleve lineárisan függetlenek voltak. □

Az új $\mathcal{B}'$ bázishoz tartozó bázismegoldás megengedett. Transzformáljuk a feladatot (sorkivonogatással) úgy, hogy megkapjuk a $\mathcal{B}'$ bázishoz tartozó kanonikus alakot. A lépés hatására a célfüggvény nőtt $\bar{\vartheta}(\bar{c}_p$ értékkel.

Ismételjük az eljárást.

Hogyan állhat meg az eljárás?

- Nincs javítóvektor. **Stop. A bázismegoldás optimális.**
- A javítóoszlopban nincs > 0 komponens. **Stop. A célfüggvényérték nem korlátos.**

2.2. Definíció. A $\mathcal{B}$ bázis **optimális**, ha megengedett és teljesül a szimplex módszer megállási feltétele (azaz nincs javítóvektor):

$$c_{\mathcal{B}} \underbrace{B^{-1}A}_{\text{transzformált mátrix}} \geq c$$

A szimplex módszer megengedett bázisokon lépked. A bázisok száma véges: legfeljebb $\binom{n}{m}$. Tegyük fel, hogy minden lépésben $\bar{\vartheta} > 0$ adódik. Ekkor minden lépésben határozottan javul a célfüggvény érték. Ezért ilyenkor nem térhetünk vissza olyan bázisba, amiben már jártunk. A fenti feltétel mellett a szimplex módszer véges sok lépés után leáll.

Mikor lehet $\bar{\vartheta} = 0$? Akkor, ha $\bar{b}_i = 0$ valamely i-re. $\bar{b}_i = 0$ azt jelenti, hogy a b jobboldal-vektor benne van az $a_{\mathcal{B}_1}, \ldots, a_{\mathcal{B}_{i-1}}, a_{\mathcal{B}_{i+1}}, \ldots, a_{\mathcal{B}_m}$ bázisoszlopok által kifeszített altérben.

2.3. Megjegyzés. Ha a jobboldalt véletlenszerűen választom folytonos eloszlás szerint, akkor ennek az esetnek 0 a valószínűsége.

2.3. Definíció (Általános helyzetű jobboldal)**.** Ha a jobboldal egyetlen, $m-1$ oszlop által generált altérben sincs benne, akkor azt mondjuk, hogy a jobboldal általános helyzetű.

2.2. A szimplex módszer elméleti vs. gyakorlati megfontolásai

Elméletileg Van olyan feladat, amelyre ha elég *ügyetlenül* választjuk a báziscseréket - a szimplex módszer ciklizál.

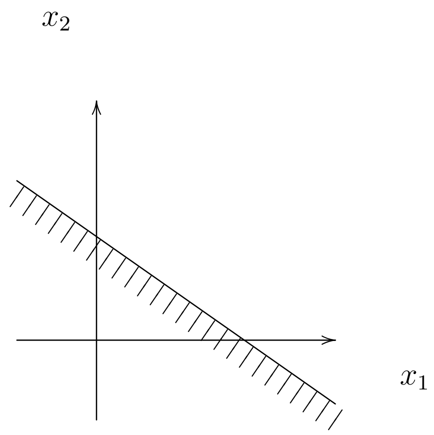
Tekintsük a kétdimenziós teret. Ebben a térben az

$$Ax \leq b$$

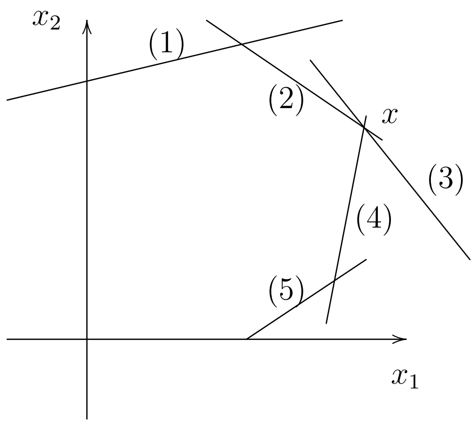
$$x \geq 0$$

$$\max cx$$

feladatban minden $a^i x \leq b_i$ feltételnek megfelel a következő ábra:



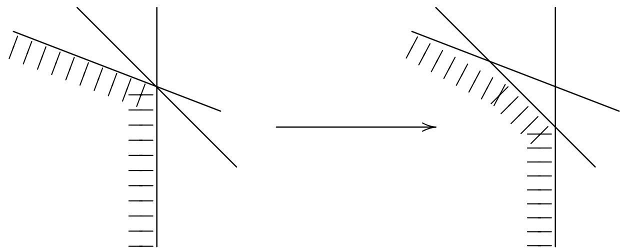
Előfordulhat **degenerált feladat** - ilyenkor egy adott csúcs (bázismegoldás) több bázishoz is tartozik(lásd az x csúcsot, amely a (2,3), (3,4) és (2,4) bázisokhoz egyaránt tartozik)



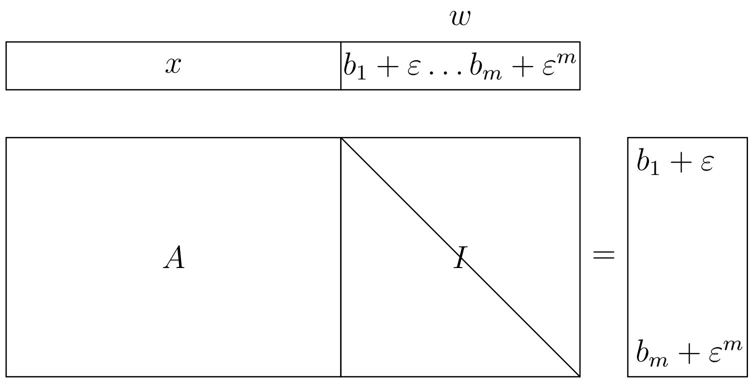
A szimplex a megengedett tartomány szomszédos csúcsain lépked. Ha a feladat degenerált, előfordulhat, hogy a fent említett x csúcshoz érve, a szimplex ciklizál - báziscsere van, de nem tud kilépni a csúcsból.

A ciklizálás elkerülésére két megoldást írunk le.

1. megoldás: perturbáció csúsztassuk el a redundáns feltételt:



Ez a jobboldal perturbálásának felel meg.



Könnyű megmutatni, hogy ha ε elég kicsi, akkor a degeneráció megszűnik.

Eredeti feladat	Perturbált feladat
$\mathcal{B}$ megengedett kezdőbázis	$\mathcal{B}$ megengedett kezdőbázis
nem tudjuk, hogy a jobboldal általános helyzetű-e	van olyan pozitiv ε, hogy a jobboldal általános helyzetű

2.1. Állítás. *Ha ε-t elég kicsire választjuk, akkor teljesül a következő állítás: Legyen $\mathcal{B}$ a perturbált feladatnak megengedett bázisa (azaz a hozzá tartozó bázismegoldás megengedett). Ekkor $\mathcal{B}$ az eredeti feladatban is megengedett bázis.*

Bizonyítás. A definicióból következik. □

Megmutattuk tehát, hogy lehet úgy perturbálni a feladatot, hogy azon végighaladva a szimplex módszerrel, és ezeket a lépéseket megfeleltetve az eredeti feladatnak, szabályos lépések sorozatát kapjuk. Megjegyezzük, hogy nem kell feltétlenül konkrét ε értéket kiszámolni, ehelyett nézhetjük azt, hogy milyen lépéseket tenne az algoritmus infinitezimális ε-ra. Az így kapott módszer a *lexikografikus szimplex módszer.*

2. megoldás: Bland-szabály A Bland-szabály alkalmazása azt jelenti, hogy a bázisba belépő és kilépő változók kiváláasztásánál több lehetőség esetén mindig a legkisebb indexű változót választjuk. Azaz egyrészt ha több változónak is negatív a redukált költsége, akkor ezek közül legkisebb indexűt választjuk belépőnek, másrészt ha a hányados-szabálynál több hányados is minimális, akkor közülük azt választjuk, ami a legkisebb indexű változóhoz tartozik.

2.1. 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: $\mathcal{B}, \bar{b}, \bar{c}, \bar{A}$; a t_2 lépés előtt: $\mathcal{B}', \bar{b}', \bar{c}', \bar{A}'$.

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

Nézzük most a t_2-edik lépést: legyen $p = \mathcal{B}'(r)$, 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

$$0 < \bar{c}_q^{t_2} - \bar{c}_q^{t_1} = \bar{c}_{\mathcal{B}} B^{-1} a_q - \bar{c}_{\mathcal{B}'} (B')^{-1} a_q = (c_{\mathcal{B}} B^{-1} B' - c_{\mathcal{B}'}) \bar{a}'_q = -\bar{c}_{\mathcal{B}'} \bar{a}'_q.$$

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 $\mathcal{B}'$-ben, akkor $\mathcal{B}$-ben is benne van, tehát $\bar{c}_j = 0$).

Azt kaptuk, hogy $-\bar{c}_{\mathcal{B}'} \bar{a}'_q < 0$, ellentmondás. $\square$

Elméletileg ha $x \in \mathbb{R}^n$, akkor tudok olyan LP feladatot szerkeszteni, amelynek 2^n megengedett bázismegoldása van. Például: $(0 \dots 0) \leq x \leq (1 \dots 1)$ - nek a megengedett tartománya az n-dimenziós kocka. A megengedett bázismegoldások száma pedig: 2^n. Megmutatták, hogy az n-dimenziós kockát lehet úgy torzítani (a csúcsok pozíciójának kis változtatásával), hogy alkalmas célfüggvény esetén a Bland szabály szerinti szimplex algoritmus az összes csúcson végigmenjen.

Nézzük a következő táblázatot:

n	10	20	30	40	50	60
pontok bejárásának ideje	$\frac{1}{1000}$ sec	1 sec	18 min	12 nap	35 év	36.000 év

Ha a számítógépek 1024-szer gyorsabbak, akkor is csak eggyel tolódik jobbra az alsó oszlop. Ez azt jelentené, hogy még a kis feladatokat sem tudnánk megoldani a szimplex módszerrel valós időben.

Gyakorlatban nem találkozunk olyan feladatokkal, amiken a szimplex módszer ilyen lassan futna. Ennek az az oka, hogy a fenti torzított n-dimenziós kockának nagyon speciális struktúrája van, ami gyakorlati feladatokból származó LP-kben nem jelenik meg. Általában nagyjából $2m$ lépésre van szüksége a szimplex módszernek, ha jól van implementálva.

2.3. Kétfázisú szimplex módszer

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.

Legyen A teljes sorrangú mátrix, b megfelelő dimenziós vektor.

$$Ax = b$$
$$x \geq 0$$

Az első fázisban nincs szükségünk célfüggvényre, mivel az csak megengedett bázismegoldást akarunk keresni.

Tegyük fel, hogy $b \geq 0$. (Ha b-nek negatív komponensei lennének, a megfelelő sort (-1)-el kell szorozni.)

A kibővített feladat: $(A, I)(x, u) = b$, $(x, u) \geq 0$. A feladat megengedett bázismegoldása: $(x, u) = (0, b)$.

Legyen a célfüggvény: $\min(\mathbf{0}, \mathbf{1})^T(x, u)$.

2.4. Megjegyzés. A kibővített feladatban minden megengedett megoldáshoz nemnegatív célfüggvényérték tartozik, mivel $(\mathbf{0}, \mathbf{1}) \geq 0$ és $(x, u) \geq 0$.

Ha van megengedett kezdőmegoldásunk, alkalmazhatjuk a szimplex módszert. Mivel a célfüggvény korlátos (nem javítható minden határon túl) – a szimplex módszer optimális megoldást fog találni.

Két eset fordulhat elő:

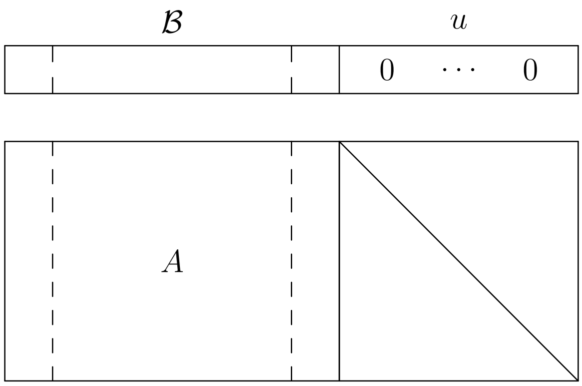
- az optimális célfüggvényérték > 0
- az optimális célfüggyvényérték $= 0$

Tegyük fel, hogy az első eset áll fenn. Ekkor:

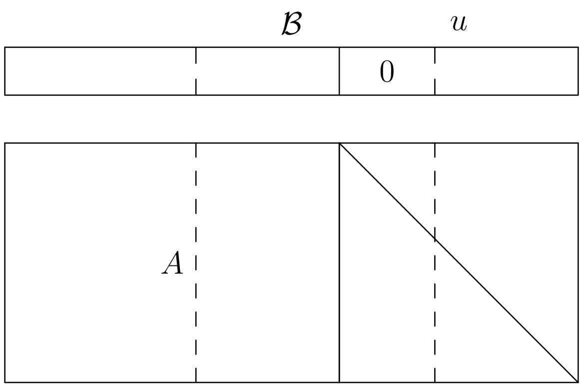
2.2. Állítás. *Az eredeti feladatnak nincs megengedett megoldása.*

Bizonyítás. **(indirekt)** Legyen $\bar{x}$ megengedett megoldása az eredeti feladatnak. Ekkor $(\bar{x}, \mathbf{0})$ a kibővített feladatnak olyan megengedett megoldása, amelynek 0 a célfüggvényértéke. Ez nyilvánvalóan ellentmondás, mivel az optimális célfüggvényérték pozitív. $\square$

Tegyük fel, hogy a második eset áll fenn. Ekkor a bázisunk kétféleképpen nézhet ki:



vagy



Jelölje (x^*, u^*) az optimális megoldást. Mivel $(\mathbf{0},\mathbf{1})^T(x^*,u^*) = \mathbf{0}^T x^* + \mathbf{1}^T u^* = 0$, így x^* megengedett megoldása az eredeti feladatnak. Ha a $\mathcal{B}$ bázis tartalmaz mesterséges változókat, akkor ezeket kihagyhatjuk a bázisból és bevehetünk helyettük eredeti változókat, hiszen az A mátrix teljes sorrangú, így oszlopvektorok tetszőleges lineárisan független halmaza kiegészíthető bázissá. A bázismegoldás ettől a cserétől nem változik, továbbra is x^* marad.

Az így kapott megengedett bázisból indítjuk a második fázist, immár az eredeti célfüggvénnyel.

Implementációs megfontolások Nézzük a következő textilipari feladatot:

$$\left[\begin{array}{cccccc} FonA & FonB & SzovA & SzovB & AllFon & AllSzov \\ \hline 3 & 2 & 10 & 4 & 1 & 0 \\ 0 & 0 & 2 & \frac{1}{2} & 0 & 1 \end{array}\right] x = \left[\begin{array}{c} b \\ 18 \\ 3 \end{array}\right]$$

Jelöljük a kezdeti mátrixot A_0-val, a kezdeti jobboldalt pedig b_0-val.
Végezzük el a következő transzformációkat:

- Kerüljön $SzovA$ a bázisba. A transzformált mátrixot jelöljük A_1-el, az új jobboldalt pedig b_1-el.

$$\begin{bmatrix} 3 & 2 & 0 & \frac{3}{2} & 1 & -5 \\ 0 & 0 & 1 & \frac{1}{4} & 0 & \frac{1}{2} \end{bmatrix} x = \begin{bmatrix} 3 \\ \frac{3}{2} \end{bmatrix}$$

  Ez a transzformáció megfelel annak, hogy az A_0 mátrixot balról az E_1 mátrixszal beszorozzuk:

$$E_1 = \begin{bmatrix} 1 & -5 \\ 0 & \frac{1}{2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & -5 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 3 & 2 & 10 & 4 & 1 & 0 \\ 0 & 0 & 2 & \frac{1}{2} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 0 & \frac{3}{2} & 1 & -5 \\ 0 & 0 & 1 & \frac{1}{4} & 0 & \frac{1}{2} \end{bmatrix}$$

$$A_1 = E_1 A_0$$
$$b_1 = E_1 b_0$$

- Vigyük ezután $SzovB$-t a bázisba. Az új mátrix legyen A_2, az új jobboldal legyen b_2, a transzformációs mátrix pedig E_2:

$$\begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{1}{6} & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 & 0 & \frac{3}{2} & 1 & -5 \\ 0 & 0 & 1 & \frac{1}{4} & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 2 & \frac{4}{3} & 0 & 1 & \frac{2}{3} & -\frac{10}{3} \\ -\frac{1}{2} & -\frac{1}{3} & 1 & 0 & -\frac{1}{6} & \frac{4}{3} \end{bmatrix}$$

$$A_2 = E_2 A_1$$
$$b_2 = E_2 b_1$$

A ritka mátrix nemnulla elemekkel való feltöltése illetve a numerikus hibák felhalmozódása ellen az implementációs trükk az eredeti mátrix és az E-mátrixok tárolása lesz. Ily módon egy adott oszlop transzformáltját úgy kapjuk meg, hogy az eredeti oszlopot megszorozzuk a transzformációs mátrixszal.

$$A_2 = E_2 E_1 A_0$$
$$b_2 = E_2 E_1 b_0$$

A kérdés az, hogy meg lehet úgy oldani a szimplex módszer implementálását, hogy csak ezeket az adatokat tároljuk? Ha igen, az így kapott megoldás optimális lesz?

Javítóvektor keresése Hasonlítsuk össze $c_{\mathcal{B}}A_2$-t c-vel. Ha teljesül a $\geq$ reláció, akkor optimális megoldásunk van. Különben keressünk olyan indexet, ahol a komponensek között van $<$ reláció.

Hogyan számítható $c_{\mathcal{B}}A_2$?

$$c_{\mathcal{B}}A_2 = c_{\mathcal{B}}(E_2E_1A_0) = (c_{\mathcal{B}}E_2E_1)A_0$$

A javítóvektor kereséséhez elég az $A_0, E_2, E_1, \ldots$ mátrixok ismerete.

Általános eset Tegyük fel, hogy a j-edik oszlopvektor javító. Ekkor szükségünk lesz az eredeti j oszlopvektor transzformált alakjára.

$$\begin{array}{cccc}
 & & \left[\begin{array}{c|c|c} \quad & \qquad B_K \qquad & \quad \end{array}\right] & A_0 \\
E_1 & \left[\qquad\right] & & \\
 & \vdots & & \\
E_K & \left[\qquad\right] & \left[\begin{array}{c|c|c} \quad & \qquad I \qquad & \quad \end{array}\right] & A_K \\
 & & \left[\begin{array}{c|c|c} \quad & \qquad c^T_{\mathcal{B}_K} \qquad & \quad \end{array}\right] &
\end{array}$$

$$\underbrace{E_K \ldots E_1}_{B_K^{-1}} B_K = I$$

A szimplex módszer így sokkal tömörebben tárolható – lényegében az E transzformációs mátrixnak is elég, ha csak egy-egy oszlopát tároljuk.

Hasonlítsuk tehát össze $c_{\mathcal{B}_K}(\underbrace{E_KE_{K-1}\ldots E_1}_{B_K^{-1}})A_0$-t és a c vektorokat. Ehhez számítsuk ki: $c_{\mathcal{B}_K}B_K^{-1}$-et. Ha találunk javítóvektort, akkor azt transzformálni kell az új bázisba, azaz szorozzuk balról a $B_K^{-1} = E_KE_{K-1}\ldots E_1$ mátrixszal.

3. Dualitás

Tekintsünk egy sztenderd alakú LP feladatot:

$$Ax = b$$

$$x \geq 0$$

$$\max cx,$$

ahol $A \in \mathbb{Q}^{m\times n}$ teljes sorrangú mátrix. Próbáljunk meg az egyenletek kombinálásával felső korlátot adni az optimum értékére! Legyen $y = (y_1, \ldots, y_m)$ egy m-dimenziós sorvektor. Ha x megoldása az $Ax = b$ egyenletrendszernek, akkor megoldása az $(yA)x = yb$ egyenletnek is. Tegyük fel, hogy találunk egy olyan y vektort, amire $yA \geq c$, azaz yA minden komponense legalább akkora, mint c megfelelő komponense. Ekkor nemnegatív x esetén $cx \leq (yA)x$. Tehát ha x megoldása az LP feladatnak, akkor $cx \leq (yA)x = yb$. Vagyis yb felső korlát az LP feladat optimum-értékére!

Ha a legjobb ilyen felső korlátot szeretnénk megtalálni, akkor egy olyan $y = (y_1, \ldots, y_m)$ sorvektort kell keresnünk, amire $yA \geq c$, és yb a lehető legkisebb. Ez a következő, úgynevezett **duális** lineáris programra vezet:

$$\begin{aligned} yA &\geq c \\ y^T &\in \mathbb{R}^m \\ \min yb&. \end{aligned}$$

Vegyük észre, hogy a duális feladat változóinak a száma megegyezik az eredeti feladat egyenleteinek a számával, és viszont. Amikor a dualitás keretében beszélünk ezekről a feladatokról, akkor az eredeti feladatot **primál** feladatnak, a duálisát pedig **duál** feladatnak szokás nevezni.

Tehát a primál feladat a következő:

$$(P) \quad \begin{array}{l} Ax = b \\ x \geq 0 \\ \max cx \end{array}$$

A (P) feladatnak megfelelő duál feladat pedig:

$$(D) \quad \begin{array}{l} yA \geq c \\ \min yb \end{array}$$

Az alábbi tétel formálisan kimondja azt, amit a bevezetőben már láttunk: a duális feladat egy megengedett megoldása felső korlátot ad a primál feladat célfüggvény-értékére.

3.1. Tétel (Gyenge dualitási tétel). *Legyen x^* a (P) feladatnak, y^* pedig a (D) feladatnak megengedett megoldása. Ekkor $cx^* \leq yb^*$.*

Bizonyítás. Az y^* (D)-megengedettsége miatt: $yA \geq c$, azaz $yA - c \geq 0$. Az x^* (P)-megengedettsége miatt pedig $x^* \geq 0$. Így $(y^*A - c)x^* \geq 0$, azaz $Ax^* = b$ miatt $y^*b \geq cx^*$, és kész a bizonyítás. □

Tegyük fel, hogy a (P) feladatnak van egy optimális $\mathcal{B}$ bázisa.

3.1. Definíció (Árnyékár vektor). Az $\bar{y} = c_{\mathcal{B}}B^{-1}$ vektort a $\mathcal{B}$ optimális bázishoz tartozó árnyékár-vektornak nevezzük.

3.2. Lemma. *Ha $\mathcal{B}$ optimális bázis, akkor a hozzá tartozó $\bar{y}$ árnyékár-vektor optimális megoldása a (D) duális feladatnak, és $\bar{y}b$ megegyezik a primál feladat optimumértékével.*

Bizonyítás. Egyrészt $\bar{y}A = c_{\mathcal{B}}B^{-1}A \geq c$ a megállási feltétel miatt, tehát $\bar{y}$ megengedett megoldása a (D) feladatnak. Másrészt $\bar{y}b = c_{\mathcal{B}}B^{-1}b = c\bar{x}$, így a gyenge dualitás tétel miatt $\bar{y}$ optimális megoldása (D)-nek. □

3.3. Tétel (Erős dualitási tétel). *Tegyük fel, hogy mind a (P) mind a (D) feladatnak van megengedett megoldása. Ekkor mindkettőnek van optimális megoldása is, és az optimális célfüggvényértékek megegyeznek.*

Bizonyítás. Oldjuk meg a (P) feladatot szimplex módszerrel.

- Ha nem állna rendelkezésünkre kezdő megengedett bázismegoldás, akkor az ismertetett első fázissal tudunk ilyet keresni. Ez az eljárás is szimplex módszeren alapszik. Amint láttuk, a szimplex módszerrel a ki- és belépő vektorokat meg lehet úgy választani (Bland szabály), hogy az eljárás véges sok lépésben leálljon. Mivel feltettük, hogy a (P) feladatnak van megengedett megoldása, ezért az első fázis megengedett kezdő bázismegoldással fog leállni.

- A megengedett kezdő bázismegoldásból kiindulva, szimplex módszerrel oldjuk meg a (P) feladatot. A Bland szabály használatával az eljárás véges sok lépésben véget ér.

A szimplex módszer általában kétféleképpen érhet véget:

1. kiderül, hogy a célfüggvény minden határon túl növelhető,

2. optimális bázist kapunk.

Az első eset nem fordulhat elő, mert a (D) feladatnak a feltevésünk szerint van megengedett megoldása, amihez tartozó duál célfüggvényérték felső korlátja a (P) feladat megengedett megoldásaihoz tartozó célfüggvényértékeknek a gyenge dualitási tétel miatt.

Tehát a szimplex módszer a második eset szerint egy $\mathcal{B}^*$ optimális bázissal áll le. Jelölje x^* a $\mathcal{B}^*$ bázishoz tartozó (optimális) bázismegoldást, y^* pedig a hozzá tartozó árnyékár-vektort. a 3.2. Lemma szerint $cx^* = y^*b$, és y^* optimális megoldáas a (D) feladatnak. □

3.1. Állítás. *Tegyük fel, hogy a (P) feladatnak van megengedett megoldása, de a (D) feladatnak nincs. Ekkor a (P) feladat célfüggvényértéke minden határon túl javítható.*

Bizonyítás. A bizonyítás menete az erős dualitási tétel bizonyításához hasonló. A (P) feladatra alkalmazzuk a kétfázisú szimplex módszert, ami a Bland szabállyal véges sok lépésben leáll. Általában két eset lehetséges:

1. kiderül, hogy nem korlátos a célfüggvény,

2. optimális bázist találunk.

Most a második eset nem fordulhat elő, mert az optimális bázishoz tartozó árnyékár vektor a (D) feladatban megengedett megoldás volna, ami a feltételezés szerint nem létezik. Marad az első eset. □

3.2. Állítás. *Tegyük fel, hogy a (D) feladatnak van megengedett megoldása, de a (P) feladatnak nincs. Ekkor a (D) feladat célfüggvényértéke minden határon túl csökkenthető.*

Bizonyítás. Mivel a (P) feladatnak nincs megengedett megoldása, a szimplex módszer első fázisa 0-nál kisebb célfüggvényértékkel ér véget. Az első fázis a következő feladatot oldja meg:

$$(P') \quad \begin{array}{l} Ax + Iu = b \\ x \geq 0 \\ u \geq 0 \\ \max - \sum_{i=1}^{m} u_i \end{array}$$

A (P) feladatnak megfelelő duál feladat pedig:

$$(D') \quad \begin{array}{l} yA \geq 0 \\ y \geq -\mathbf{1} \\ \min yb \end{array}$$

Mivel a szimplex módszer első fázisa 0-nál kisebb célfüggvényértékkel ér véget, az erős dualitás tétel szerint létezik (D')-nek olyan $\bar{y}$ megengedett megoldása, amire $\bar{y}b < 0$. Legyen y^* a (D) feladat egy megengedett megoldása. Ekkor tetszőleges $\lambda > 0$ számra $y^* + \lambda\bar{y}$ is megengedett megoldása (D)-nek, és a célfüggvényértéke $y^*b + \lambda\bar{y}b$, ami λ növelésével minden határon túl csökkenthető. □

Az erős dualitás-tételből és a két fenti állításból már következik az alábbi tétel.

3.4. Tétel (Alternatíva-tétel). *A (P)-(D) feladatokra a következő esetek lehetségesek:*

1. *Mindkettőnek van optimális megoldása, és az optimális célfüggvényértékek megegyeznek.*
2. *A (P) célfüggvényértéke minden határon túl javítható, a (D) feladatnak nem megoldható.*
3. *A (D) célfüggvényértéke minden határon túl javítható, a (P) feladatnak nem megoldható.*
4. *Sem a (P), sem a (D) feladatnak nincs megengedett megoldása.*

Optimalitási feltételek

3.5. Tétel. *Ha x^* optimális megoldása (P)-nek és y^* optimális megoldása (D)-nek, akkor $x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$. Ennek a fordítottja is igaz: ha x^* és y^* megengedett megoldása (P)-nek illetve (D)-nek, és $x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$, akkor x^* és y^* optimális megoldása (P)-nek illetve (D)-nek.*

Bizonyítás. Az erős dualitás-tételből következik, hogy ha x^* optimális megoldása (P)-nek és y^* optimális megoldása (D)-nek, akkor $cx^* = y^*b$. Másrészt a megoldások megengedettségét használva a gyenge dualitás-tétel bizonyításához hasonlóan láthatjuk, hogy $cx^* \leq (y^*A)x^* = y^*(Ax^*) = y^*b$. A $cx^* = y^*b$ egyenlőségből tehát az következik, hogy $cx^* = (y^*A)x^*$. Ez pontosan akkor teljesül, ha $x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$.

A tétel második felének a bizonyításához azt kell megfigyelni, hogy ha $x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$, akkor $cx^* = (y^*A)x^*$, és így $cx^* = (y^*A)x^* = y^*(Ax^*) = y^*b$. Ebből következik, hogy x^* is és y^* is optimális. □

A fenti tételben szereplő feltételeket ($x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$) **optimalitási feltételek**nek nevezzük. Ha ismerünk egy y^* optimális duális megoldást, akkor az optimalitási feltételek segítségével könnyebben tudjuk (P)-nek megtalálni egy optimális megoldását: bizonyos x_j változókról tudjuk, hogy csak 0 lehet az értékük, tehát ezeket kihagyhatjuk a feladatból. Az így kapott feladatnak elég egy megengedett megoldását találni, hiszen az teljesíti az optimalitási feltételeket, így a tétel értelmében optimális.

Más alakok

Legyen most a feladatunk a következő formájú:

$$(P) \quad \begin{array}{l} Ax \leq b \\ x \geq 0 \\ \max cx \end{array}$$

Mi lesz ennek a feladatnak a duálisa? Ha a duális feladat szemléletes jelentését nézzük, azaz hogy minél jobb felső korlátot akarunk kapni a célfüggvényre az egyenlőtlenségek kombinálásával, akkor láthatjuk, hogy most csak nemnegatív együtthatókkal kombinálhatjuk az egyenlőtlenségeket. Negatív szorzók esetén ugyanis megfordulna az egyenlőtlenség iránya, tehát nem kapnánk felső korlátot. A duális feladatban így az y változókra bekerül a nemnegativitási feltétel.

Formálisan úgy is meghatározhatjuk a feladat duálisát, hogy először átírjuk a feladatot sztenderd alakba, és annak a duálisát vesszük. A sztenderd alak:

$$(P') \quad \begin{array}{l} Ax + Iu = b \\ x \geq 0 \\ u \geq 0 \\ \max cx \end{array}$$

Írjuk fel (P') feladathoz tartozó duál feladatot:

$$(D') \quad \begin{array}{l} yA \geq c \\ yI \geq 0 \\ \min yb \end{array} \quad \equiv \quad (D) \quad \begin{array}{l} yA \geq c \\ y \geq 0 \\ \min yb \end{array}$$

A gyenge és erős dualitási-tétel, valamint az alternatíva-tétel öröklődik erre a (P)-(D) feladatpárra is. Vegyük észre, hogy ha a (D) feladatnak felírjuk a duálisát (transzponálva és az egyenőtlenségeket és a célfüggvényt -1-gyel megszorozva, hogy jó alakú legyen), akkor pont a primál feladatot kapjuk vissza. A duál feladat duálisa tehát a primál feladat.

Az optimalitási feltételek a következők lesznek erre a primál és duál feladatra.

3.6. Tétel. *Ha x^* optimális megoldása (P)-nek és y^* optimális megoldása (D)-nek, akkor $x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$, és $y_i^* = 0$ minden olyan i-re, amire $(Ax^*)_i < b_i$. Ennek a fordítottja is igaz: ha x^* és y^* megengedett megoldása (P)-nek illetve (D)-nek, és $x_j^* = 0$ minden olyan j-re, amire $(y^*A)_j > c_j$ valamint $y_i^* = 0$ minden olyan i-re amire $(Ax^*)_i < b_i$, akkor x^* és y^* optimális megoldása (P)-nek illetve (D)-nek.*

Bizonyítás. A bizonyítás ugyanúgy megy, mint a 3.5. Tétel esetén, csak most a $cx^* \leq (y^*A)x^* = y^*(Ax^*) \leq y^*b$ egyenlőtlenség-sort használjuk. A $cx^* = y^*b$ egyenlőség pontosan akkor teljesül, ha $cx^* = (y^*A)x^*$ és $y^*(Ax^*) = y^*b$. Ezek pont a tételben szereplő optimalitási feltételeknek felelnek meg. $\square$

Hogyan kezeljük az olyan feladatokat, amelyeknél nincs előjel-megkötés a változókra? Nézzük például az alábbi feladatot:

(I) $Ax \leq b$, $\max cx$

$$\begin{array}{c} \left[\qquad x^T \qquad\right] \\ \left[\begin{array}{c} \\ \qquad A \qquad \\ \\ \end{array}\right] \leq \left[\begin{array}{c} \\ b \\ \\ \end{array}\right] \\ \left[\qquad c \qquad\right] \end{array}$$

Alakítsuk át a feladatot úgy, hogy a változókra nemnegativitási feltételünk legyen. Legyen $x = x^+ - x^-$, $x^+, x^- \geq 0$. Ekkor a feladat az alábbi alakban írható fel:

$$\begin{array}{c} \left[\qquad x^+ \qquad \Big| \qquad x^- \qquad\right] \\ \left[\begin{array}{c|c} & \\ \qquad A \qquad & \qquad -A \qquad \\ & \\ \end{array}\right] \leq \left[\begin{array}{c} \\ b \\ \\ \end{array}\right] \\ \left[\qquad c \qquad \Big| \qquad -c \qquad\right] \end{array}$$

(II) $(A, -A)(x^+, x^-) \leq b$
$(x^+, x^-) \geq 0$
$\max(c, -c)(x^+, x^-)$
Mennyire ekvivalens ez a feladat az előzővel?

3.3. Állítás. *Ha (x^+, x^-) a (II) feladat megengedett megoldása, akkor $x^+ - x^-$ az (I) feladat megengedett megoldása és a megfelelő célfüggvény értékek azonosak.*

Bizonyítás. $(c, -c)(x^+, x^-) = cx^+ - cx^- = c(x^+ - x^-)$ □

3.4. Állítás. *Ha x az (I) feladat megengedett megoldása, akkor $x^+ = [x]_+$, $x^- = [x]_-$ választással (x^+, x^-) a (II) feladat megengedett megoldása.*

Írjuk fel a (II) feladat duálisát:

$$\begin{array}{c} \\ \begin{bmatrix} y \end{bmatrix} \end{array} \begin{array}{c} \begin{bmatrix} A & \big| & -A \end{bmatrix} \le \begin{bmatrix} b \end{bmatrix} \\ \begin{bmatrix} c & | & -c \end{bmatrix} \end{array}$$

$$\begin{array}{l} y(A, -A) \ge (c, -c) \\ y \ge 0 \\ \min yb \end{array} \quad \rightsquigarrow \quad \begin{array}{l} \underbrace{yA \ge c \text{ és } y(-A) \ge (-c)}_{yA = c} \\ y \ge 0 \\ \min yb \end{array}$$

Láthattuk tehát, hogy ha a primál feladatban egyenlőség van, akkor a duál feladatban nincs nemnegativitási feltétel a változóra és fordítva, ha a primál feladatban van nemnegativitási feltétel, akkor a duál feladatban egyenlőtlenség van.

Most nézzük meg, hogy lehet (viszonylag) automatikusan felírni egy feladat duálisát:

$$\begin{array}{cccc} & & 0 \le & - \\ & & [\; x_1^T & | \; x_2^T \;] \\ - & y_1^T & A_{11} & A_{12} & = & b_1 \\ 0 \le & y_2^T & A_{21} & A_{22} & \le & b_2 \\ & & \ge & = \\ & & [\; c_1 & | \; c_2 \;] \end{array}$$

$$(P) \quad \begin{array}{l} A_{11}x_1 + A_{12}x_2 = b_1 \\ A_{21}x_1 + A_{22}x_2 \le b_2 \\ x_1 \ge 0 \\ x_2 - \\ \max c_1x_1 + c_2x_2 \end{array} \qquad (D) \quad \begin{array}{l} y_1A_{11} + y_2A_{21} \ge c_1 \\ y_1A_{12} + y_2A_{22} = c_2 \\ y_1 - \\ y_2 \ge 0 \\ \min y_1b_1 + y_2b_2 \end{array}$$

4. Farkas lemma

4.1. Tétel (Farkas Lemma). *Az alábbi egyenlőtlenségrendszerek közül pontosan az egyik megoldható:*

$$(I) \quad \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \quad (II) \quad \begin{array}{l} yA \geq 0 \\ yb < 0 \end{array}$$

Bizonyítás. Könnyen látható, hogy egyszerre nem oldhatók meg:

$0 \leq \underbrace{(yA)}_{\geq 0} \underbrace{x}_{\geq 0} = y(Ax) = yb < 0$

Írjuk fel az (I) feladatot a következő ekvivalens formában, és írjuk fel a hozzá tartozó duál feladatot:

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

A fenti (D) feladatnak az $y = (0, \ldots, 0)$ vektor mindig megengedett megoldása. Ezért a 3.4. Tételben felsorolt négy eset közül kettő következhet csak be:

- Ha a (P) feladatnak van megengedett megoldása, akkor van optimális megoldása is, és az optimális célfüggvényérték 0, mivel minden primál célfüggvényérték 0. Ekkor a (D) feladat célfüggvényértéke nem csökkenthető 0 alá, azaz $yb \geq 0$ minden megengedett megoldásra (azaz minden olyan y-ra, amire $yA \geq 0$). Tehát a (II) feladat nem megoldható, miközben az (I) feladat megoldható.

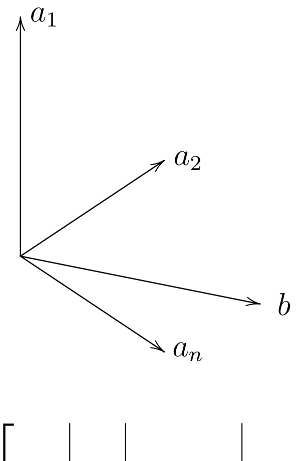
- Ha a (P) feladatnak nincs megengedett megoldása, azaz (I) nem megoldható, akkor a dualitási tételből következik, hogy a duál célfüggvényérték minden határon túl csökkenthető (nevezetesen 0 alá is). Tehát a (II) feladat megoldható, miközben az (I) feladat nem megoldható.

□

Szemléltetés – szeparációs tétel Jelölje $a_1, a_2, \ldots, a_n$ az A mátrix oszlopait.

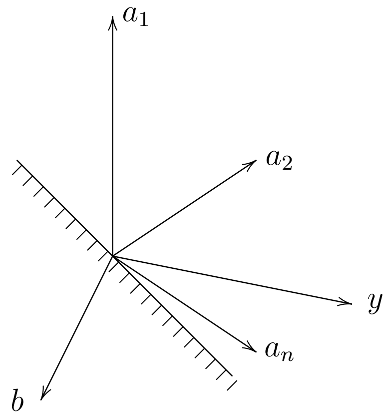
$$\left[\begin{array}{c|c|c|c} a_1 & a_2 & \quad \ldots \quad & a_n \end{array} \right] x = \left[\begin{array}{c} b \end{array} \right]$$

Az, hogy (I) megoldható, azt jelenti, hogy b előállítható az a_i-kből nemnegatív súlyokkal, azaz $b \in cone(a_1, \ldots, a_n)$.



$$\left[\begin{array}{c} y \end{array} \right] \left[\begin{array}{c|c|c|c} a_1 & a_2 & \quad \ldots \quad & a_n \end{array} \right] \left[\begin{array}{c} b \end{array} \right]$$

Az, hogy (II) megoldható, azt jelenti, hogy van olyan y^* vektor, ami minden a_i vektorral hegyes- vagy derékszöget zár be, b-vel viszont tompaszöget. Azaz az y^*-ra merőleges hipersíknak az összes a_i az y^* felé eső oldalán van (esetleg magán a hipersíkon), a b viszont szigorúan a hipersík másik oldalán. A Farkas-lemma tehát ekvivalens azzal az állítással, hogy egy generált kúp (véges sok adott vektor nemnegatív kombinációiként előálló pontok halmaza) és egy kúpon kívüli pont mindig elválasztható egy hipersíkkal.



A Farkas lemmának is felírható olyan alakja, ahol a primál feladatban egyenlőtlenségek szerepelnek.

4.2. Tétel (Farkas Lemma, egyenlőtlenséges alak)**.** *Az alábbi egyenlőtlenségrendszerek közül pontosan az egyik megoldható:*

$$(I)\quad \begin{array}{l} Ax \le b \\ x \ge 0 \end{array} \qquad (II)\quad \begin{array}{l} yA \ge 0 \\ y \ge 0 \\ yb < 0 \end{array}$$

Bizonyítás. Ha átírjuk (I)-et sztenderd alakra, akkor pont az eredeti Farkas Lemmát kapjuk. □

5. Teljesen unimoduláris mátrixok

5.1. Definíció. Egy mátrix **teljesen unimoduláris** (röviden TU), ha minden négyzetes részmátrixának determinánsa 0, 1, vagy -1.

5.1. Állítás. *Ha az A mátrix TU, akkor a következő mátrixok is TU mátrixok:*

- *A transzponáltja*
- (A, e_i)*, ahol e_i az i-edik egységvektor, azaz az i-edik koordinátája 1, a többi 0*
- *Ha A egy oszlopát megszorozzuk -1-gyel*
- (A, a)*, ahol az a vektor az A mátrix valamelyik oszlopa.*

Bizonyítás. Mindegyik esetben igaz az, hogy az új mátrix minden nemszinguláris részmátrixának a determinánsa megegyezik az A mátrix egy nemszinguláris részmátrixának determinánsával, vagy annak -1-szeresével. □

5.1. Tétel. *Ha egy A mátrix minden oszlopában legfeljebb egy $+1$-es és legfeljebb egy -1-es található, akkor A teljesen unimoduláris.*

Bizonyítás. A mátrix mérete szerinti indukcióval bizonyítunk. 1×1-es mátrixra természetesen igaz az állítás. Ha az A mátrix nem négyzetes, akkor minden négyzetes részmátrixa valódi részmátrix, és ezért indukció szerint A is TU. Tegyük fel, hogy az A mátrix négyzetes; mivel A minden valódi részmátrixáról indukció szerint tudjuk, hogy determinánsa 0, 1, vagy -1, elég A determinánsát ellenőrizni. Ha A-nak van olyan oszlopa, amiben egyetlen nemnulla elem van, akkor a kifejtési tétel miatt A determinánsának abszolút értéke megegyezik egy valódi részmátrix determinánsának abszolút értékével, így indukció szerint 0, 1, vagy -1, tehát A TU.

Tegyük most fel, hogy az A mátrix négyzetes, és minden oszlopában 1 db 1 és 1 db -1 van. Ekkor A sorainak összege a 0-vektor, tehát A sorai lineárisan összefüggőek, következésképp determinánsa 0. □

5.2. Következmény. *Irányított gráf* $0, \pm 1$*-es incidencia-mátrixa TU.*

Bizonyítás. Minden oszlopban legfeljebb egy $+1$-es és legfeljebb egy -1-es van. □

5.3. Következmény. *Páros gráf incidencia-mátrixa TU.*

Bizonyítás. Ha az egyik csúcsosztályhoz tartozó sorokat megszorozzuk -1-gyel, akkor egy irányított gráf incidencia-mátrixát kapjuk. □

5.4. Tétel. *Ha az* A *mátrix TU, akkor az* $\{Ax = b,\ x \geq 0\}$ *valamint az* $\{Ax \leq b, x \geq 0\}$ *rendszerek bázismegoldásai egész* b *esetén egész vektorok.*

Bizonyítás. Az első rendszer esetében egy B bázis az A mátrix nemszinguláris $m \times m$-es részmátrixa. A Cramer szabály értelmében $Bx = b$ egyértelmű megoldása egész, hiszen a Cramer szabályban a számláló egész, a nevező pedig 1 vagy -1. A második rendszer esetében egy B bázis az (A, I) mátrix egy nemszinguláris $m \times m$-es részmátrixa, amiből az I-beli részt kifejtve következik, hogy B determinánsának abszolút értéke megegyezik A egy aldeterminánsának abszolút értékével. Mivel A teljesen unimoduláris, ez 1, tehát a Cramer szabályban itt is 1 vagy -1 szerepel a nevezőben. □

5.5. 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* $Ax \leq b$, $x \geq 0$ *egyenlőtlenség-rendszernek van egész megoldása.*
2. *Az* $yA \geq 0$, $y \geq 0$, $yb < 0$ *rendszernek van* $y \in \{0, 1\}^m$*-beli megoldása.*

Bizonyítás. A Farkas Lemma szerint a két egyenlőtlenség-rendszer közül az egyiknek van (nem feltétlenül egész) megoldása. Ha az elsőnek van, akkor az 5.4. Tétel szerint egész megoldása is van. Ha a második megoldható, akkor olyan megoldása is van, ahol minden érték 0 és 1 közötti, hiszen a jobboldalon minden sorban 0 áll. Az 5.4. Tételt alkalmazva kapjuk, hogy van $\{0, 1\}^m$-beli megoldás. □

5.6. Következmény. *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* $\sum_{u \in U} b_u > \sum_{v \notin U} b_v$*.*

6. Egészértékű lineáris programozás – bevezető

Egészértékű lineáris programozási feladatnak (IP – integer programming) a következő alakú feladatot nevezzük:

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

A fenti feladat **LP-relaxáltját** úgy kapjuk, hogy elhagyjuk az egészértékűségi feltételt:

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

Előfordulhat, hogy két IP feladatnak ugyanaz a megoldás-halmaza, de az LP-relaxációjuknak nem. Például a $2x \le 3$ és $x \le 1$ egyenlőtlenségek egész megoldásai ugyanazok, de a $3/2$ csak az elsőnek megoldása.

Az IP feladatok részosztályát alkotják a bináris programozási feladatok, ahol a változók értéke csak 0 vagy 1 lehet:

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

Példa: Bináris hátizsák feladat

Adott n db tárgy. A j-edik tárgy értéke legyen $c_j \ge 0$, 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:

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

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).

6.1. 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 kicserélésével 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:

$$\max\{cx + dz :\ A_1 x + A_2 z \le b, x \in \mathbb{Z}^{n_1}, z \in \mathbb{R}^{n_2}\},$$

ahol A_1 $m \times n_1$-es, A_2 pedig $m \times n_2$-es mátrix.

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:

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

6.1. 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.

A példában használt elvet általánosan is kimondhatjuk.

6.1. Tétel. *Tegyük fel, hogy egészértékű programozási feladatunk*

$$\max\{cx + dz :\ A_1x + A_2z \le b, x \in \mathbb{Z}^{n_1}, z \in \mathbb{Z}^{n_2}\}$$

alakban modellezhető, ahol A_2 TU mátrix. Ekkor a feladat optimumértéke megegyezik a

$$\max\{cx + dz :\ A_1x + A_2z \le b, x \in \mathbb{Z}^{n_1}, z \in \mathbb{R}^{n_2}\} \tag{1}$$

vegyes programozási feladat optimumértékével.

Bizonyítás. Elég megmutatni, hogy az (1) feladatnak van egészértékű optimális megoldása. Vegyünk egy tetszőleges (x^*, z^*) optimális megoldást. Ha az x^*-ot rögzítjük, akkor z^* optimális megoldása a

$$\max\{dz :\ A_2z \le b - A_1x^*, z \in \mathbb{R}^{n_2}\}$$

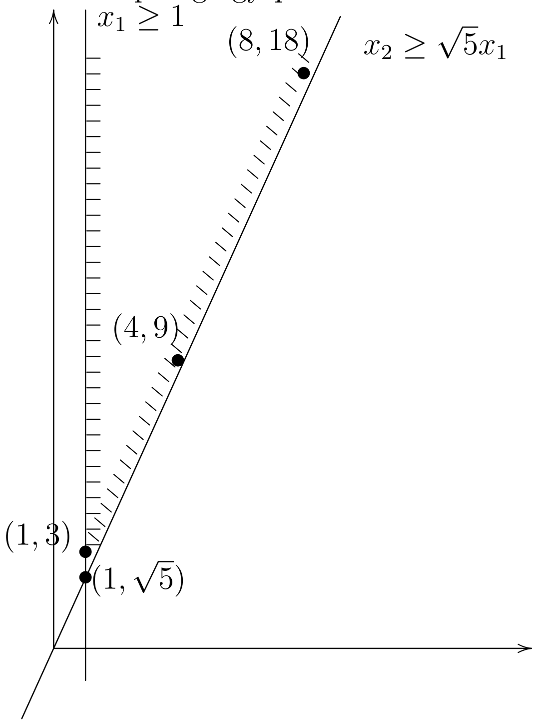
feladatnak. De mivel A_2 TU mátrix, az 5.4. Tétel értelmében ennek a feladatnak van egészértékű optimális megoldása is; legyen ez z'. Ekkor az (x^*, z') egészértékű optimális megoldása az (1) feladatnak. □

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

6.1. 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.

6.1. Megjegyzés. Ha P leírásában irracionális együtthatók is szerepelhetnek, akkor az egész pontok konvex burka nem lesz mindig poliéder. Például 2-dimenzióban: Legyen $P = \{(x_1, x_2) : x_1 \ge 1, x_2 \ge \sqrt{5}x_1\}$. 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.



7. Érvényes vágások

Ha az egészértékű programozási feladatunk LP relaxációjának optimális megoldása nem egészértékű, akkor valamilyen módszerrel tovább kell lépnünk, hogy megtaláljuk a legjobb egészértékű megoldást. Ennek egyik módja, hogy keresünk egy olyan lineáris egyenlőtlenséget, amit az összes egész megoldás teljesít, de az LP relaxáció optimális megoldása(i) nem. Az ilyen egyenlőtlenség neve **érvényes vágás**. Ha az egyenlőtlenséget új sorként hozzávesszük a rendszerhez, akkor az egész megoldások nem változnak, de az LP relaxáció optimuma közelebb kerül az IP feladat optimumához.

Az alábbiakban két módszert mutatunk érvényes vágások keresésére.

7.1. Gomory vágások

Tekintsünk egy

$$\begin{aligned} \max cx \\ Ax = b \\ x \geq 0 \\ x \in \mathbb{Z}^n \end{aligned}$$

alakú egészértékű programozási feladatot, ahol A egész mátrix és b egész vektor. Ha olyan szerencsénk van, hogy az A mátrix TU, akkor az 5.4 Tétel értelmében az LP relaxáció optimális bázismegoldása egészértékű. Ez azt jelenti, hogy ilyen esetben az IP feladatot meg tudjuk oldani a szimplex módszerrel!

Ha azonban A nem TU, akkor könnyen lehet, hogy az optimális $\mathcal{B}$ bázishoz tartozó bázismegoldás nem egészértékű. Gomory megmutatta, hogy ebben az esetben a $\mathcal{B}$ bázishoz tartozó kanonikus alak segítségével könnyen konstruálható egy érvényes vágás, ami a $\mathcal{B}$-hez tartozó bázismegoldást „levágja”, azaz a bázismegoldás nem teljesíti az egyenlőtlenséget.

Ahogy korábban, jelölje a feladat $\mathcal{B}$ bázishoz tartozó kanonikus alakját $\bar{A}x = \bar{b}$, $x \geq 0$, $\max \bar{c}x$. Tegyük fel, hogy $\bar{b}_i$ nem egész, ami azt jelenti, hogy a bázismegoldás $\mathcal{B}(i)$-edik koordinátája nem egész. Vezessük be a $q = \mathcal{B}(i)$ jelölést, és jelölje N a bázison kívüli változók indexhalmazát.

A bázishoz tartozó kanonikus alak i-edik sora a következőképpen néz ki:

$$x_q + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{b}_i.$$

Ezt az egyenletet tehát a feladat *összes megoldása* teljesíti. Mivel a megengedett megoldások nemnegatívak, az összes *megengedett megoldás* teljesíti a következő egyenlőtlenséget:

$$x_q + \sum_{j \in N} \lfloor \bar{a}_{ij} \rfloor x_j \leq \bar{b}_i.$$

Következésképp az összes *egészértékű megengedett megoldás* teljesíti a következőt:

$$x_q + \sum_{j \in N} \lfloor \bar{a}_{ij} \rfloor x_j \leq \lfloor \bar{b}_i \rfloor. \tag{2}$$

Vegyük észre azonban, hogy a $\mathcal{B}$-hez tartozó $\bar{x}$ bázismegoldás nem teljesíti az (2) egyenlőtlenséget! Ugyanis $\bar{x}_j = 0$ minden $j \in N$-re, és $\bar{x}_q = \bar{b}_i > \lfloor \bar{b}_i \rfloor$, hiszen $\bar{b}_i$ nem egészértékű. Így az (2) egyenlőtlenség egy érvényes vágás, ami levágja a $\mathcal{B}$-hez tartozó bázismegoldást. Ezt nevezzük a $\mathcal{B}$ bázishoz és i indexhez tartozó **Gomory vágás**nak.

Nézzük meg hogy az (2) egyenlőtlenséget hogyan tudjuk hozzávenni a rendszerhez úgy, hogy egyből egy bázishoz tartozó kanonikus alakot kapjunk. Mivel egyenlőtlenséget tekintünk, be kell vezetni egy új eltérés-változót, legyen ez x_{n+1}. Az új rendszer bázisát úgy kapjuk, hogy a $\mathcal{B}$ bázist kiegészítjük a $\mathcal{B}(n+1) = n+1$ definícióval, azaz az új sorhoz tartozó bázisváltozó x_{n+1}. Az új sort tehát olyan alakban kell felírnunk, hogy x_{n+1} együtthatója 1, a többi bázisváltozó együtthatója pedig 0 legyen. Ehhez figyeljük meg, hogy ha az

$$x_q + \sum_{j \in N} \lfloor \bar{a}_{ij} \rfloor x_j + x_{n+1} = \lfloor \bar{b}_i \rfloor$$

egyenletből kivonjuk a rendszerben szereplő

$$x_q + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{b}_i$$

egyenletet, akkor az

$$x_{n+1} - \sum_{j \in N} \{\bar{a}_{ij}\} x_j = -\{\bar{b}_i\} \tag{3}$$

egyenletet kapjuk, ahol $\{\alpha\}$ az α szám törtrészét jelöli. A (3) egyenlet megfelelő alakú, tehát a rendszerhez hozzávéve kanonikus alakú feladatot kapunk, és a célfüggvényen sem kell módosítani, hiszen az x_{n+1} eltérés-változó együtthatója 0 a célfüggvényben.

7.2. Klikk vágások

Bináris feladatoknál a változókra gyakran úgy tekintünk, hogy igaz-hamis értékűek, ahol 1 felel meg az igaznak és 0 a hamisnak. A logikából ismerős jelöléssel az $1 - x_i$ jelölhető $\neg x_i$-ként, és így $\neg x_i$ pontosan akkor igaz, ha x_i hamis. A **literálok** halmazába x_i és $\neg x_i$ tartozik az összes bináris x_i változóra.

Egy bináris feladat egyenlőtlenségei sokszor tartalmaznak olyan információt, hogy két literál értéke nem lehet egyszerre igaz. Például az $x_1 - x_2 \geq 0$ egyenlőtlenség azt jelenti, hogy $\neg x_1$

és x_2 nem lehet egyszerre igaz. Bonyolultabb egyenlőtlenségeknek is vannak ilyen implikációi, például

$$3x_1 + 3x_2 - 4x_3 + 5x_4 \leq 6$$

miatt következő literál-párok nem lehetnek egyszerre igazak: $(x_1, \neg x_3)$, $(x_2, \neg x_3)$, (x_1, x_4), (x_2, x_4), $(\neg x_3, x_4)$.

Egy bináris feladatból elkészíthetjük azt a gráfot, aminek a csúcsai a literálok, és az élei azok a literál-párok, amik valamelyik egyenlőtlenség miatt nem lehetnek egyszerre igazak. Tegyük fel, hogy K egy tartalmazásra maximális klikk ebben a gráfban. Vegyük észre, hogy tetszőleges megoldásban a K-beli literálok közül legfeljebb egy lehet igaz! Ezért érvényes lesz a következő, K-hoz tartozó **klikk-vágás**:

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

Ha találunk olyan klikk-vágást, ami levágja az LP-relaxáció optimális bázismegoldását, akkor azt érdemes hozzávennünk a rendszerhez.

8. Korlátozás és szétválasztás

Tekintsük a következő alakú feladatot:

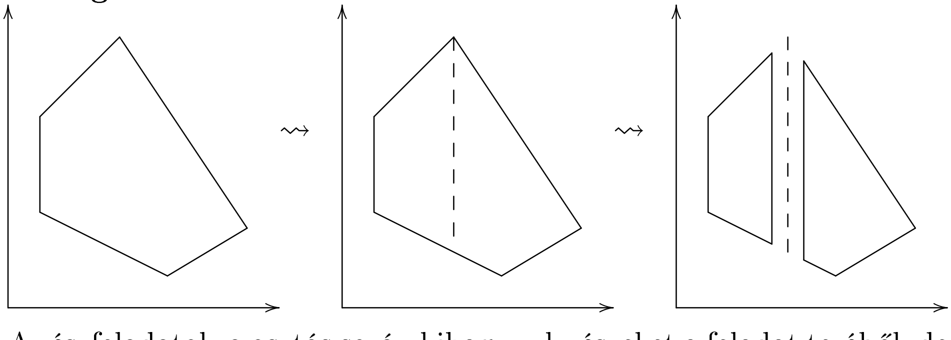
$$\max cx \tag{4}$$
$$Ax \leq b \tag{5}$$
$$x \text{ egész}, \tag{6}$$

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 $P = \{x \in \mathbb{R}^n : \ Ax \leq b\}$. A részfeladataink mindig $\max\{cx : \ x \in P' \cap \mathbb{Z}^n\}$ alakúak lesznek, ahol $P' \subseteq P$ poliéder. Jelölések:

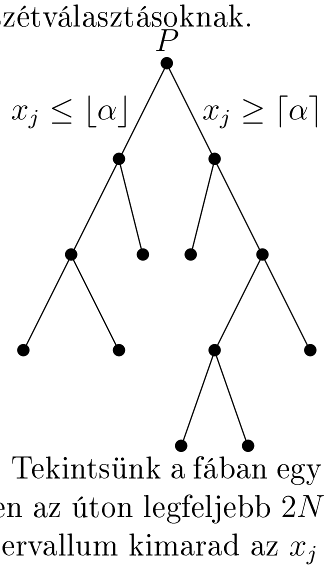
- $u(P') = \max\{cx : x \in P'\}$. Ez felső korlát a részfeladat optimumértékére.
- $x^*(P')$ a $\max\{cx : x \in P'\}$ feladat egy optimális bázismegoldása. Ez kiszámolható szimplex módszerrel.
- x^*: az eddig talált legjobb egész megoldás
- L: az eddig talált legjobb egész megoldás célfüggvényértéke,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

8.1. 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. □

Korlátozás és szétválasztás algoritmus a bináris hátizsák-feladatra

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

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

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.

8.1. 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.

9. 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.

9.1. Bináris hátizsákfeladat

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

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

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 $k \ge 1$:

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

Algoritmus

```
for k = 1...n
    for d = 0...b
    do { számoljuk ki f_k(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.

9.2. 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.

9.3. Maximális súlyú nem-átmetsző párosítás

Legyen $G = (S, T; E)$ páros gráf, ahol $S = \{s_1, \ldots, s_k\}$ és $T = \{t_1, \ldots, t_l\}$. Adott még egy $c : E \to \mathbb{R}$ súlyfüggvény. Az $s_i t_j \in E$ és $s_{i'} t_{j'} \in E$ élek **átmetszők**, ha vagy $i \leq i'$ és $j \geq j'$, vagy $i \geq i'$ és $j \leq j'$. Egy élhalmaz **nem-átmetsző**, ha nem tartalmaz átmetsző élpárt. Vegyük észre, hogy ha E egy részhalmaza nem-átmetsző, akkor automatikusan párosítás, azaz független élekből áll (fordítva nem igaz: egy párosítás tartalmazhat átmetsző élpárt).

Célunk egy maximális súlyú nem-átmetsző párosítás megtalálása. Megmutatjuk, hogy erre a feladatra hatékony dinamikus programozási algoritmus adható. A 10. fejezetben majd látjuk, hogy a maximális súlyú párosítás feladatra csak ennél jóval bonyolultabb algoritmust ismerünk, theát a nem-átmetsző feltétel kikötése könnyíti a feladatot.

Adott $i \leq k$ és $j \leq l$ esetén jelölje $G(i,j)$ a G gráf megszorításását az $\{s_1, \dots, s_i\} \cup \{t_1, \dots, t_j\}$ csúcshalmazra. Legyen

$$f(i,j) = \max\{c(M) : \ M \text{ nem-átmetsző párosítása } G(i,j)\text{-nek}\}.$$

Mivel $G(i,1)$ és $G(1,j)$ minden párosítása egyetlen élből áll, a következő értékeket könnyen megkapjuk:

$$f(i,1) = \max\{0, c_{s_h t_1} : s_h t_1 \in E, \ h \leq i\},$$

$$f(1,j) = \max\{0, c_{s_1 t_h} : s_1 t_h \in E, \ h \leq j\}.$$

Tegyük fel, hogy $2 \leq i \leq k$ és $2 \leq j \leq l$. Figyeljük meg, hogy ha $G(i,j)$-nek egy nem-átmetsző párosítás nem tartalmazza az $s_i t_j$ élt, akkor s_i és t_j közül csak az egyikre illeszkedhet éle. Ezek alapján a következő rekurzió érvényes:

$$f(i,j) = \max\{f(i-1,j-1) + c_{s_i t_j}, f(i,j-1), f(i-1,j)\}.$$

Az $f(i,j)$ értékeket tehát kiszámolhatjuk például $i+j$ szerint növekvő sorrendben, és minden egyes érték kiszámolásához csak három, már kiszámolt értéket kell összehasonlítani. A lépésszám így $O(n^2)$.

9.4. Utak aciklikus digráfban

Ebben a fejezetben megmutatjuk, hogy aciklikus, azaz irányított kört nem tartalmazó irányított gráfokban maximális és minimális súlyú utat is lehet dinamikus programozással keresni. Ehhez szükségünk van a csúcsok topologikus sorrendjének a fogalmára.

9.4.1. Topologikus sorrend

A mélységi keresés (DFS) egy alkalmazása aciklikus digráfban a pontok ún. topologikus sorrendjének meghatározására szolgál. A digráf csúcsainak egy sorrendjéről akkor mondjuk, hogy **topologikus**, ha minden él előre mutat, azaz a töve a sorrendben megelőzi hegyét (=fejét). Egy digráfot akkor neveztünk aciklikusnak, ha nem tartalmaz egyirányú kört. Azt, hogy egy digráf nem aciklikus, egy konkrét egyirányú körének bemutatásával tanúsíthatjuk. Milyen gyorsan ellenőrizhető igazolvány adható a digráf aciklikusságának tanúsítására? Erre ad választ a következő egyszerű, de hasznos tétel.

9.1. Tétel. *Egy $D = (V, A)$ digráf akkor és csak akkor aciklikus, ha pontjainak létezik topologikus sorrendje, azaz egy olyan $v_1, v_2, \dots, v_n$ sorrend, amelyben minden él korábbi pontból későbbibe vezet.*

Bizonyítás. Egy egyirányú kör pontjainak nem létezhet topologikus sorrendje, hiszen a kör minden pontjának pozitív a befoka. Emiatt egy egyirányú kört tartalmazó digráfnak se létezhet topologikus sorrendje, vagyis a feltétel szükséges.

Az elegendőséghez figyeljük meg, hogy egy aciklikus digráfnak létezik forráspontja, vagyis olyan pontja, amibe nem lép be él. Ha ugyanis minden pontba lép be él, akkor egy pontból a belépő él mentén visszafelé indulva a fordított sétát mindig tudnánk folytatni és előbb-utóbb egy kört kapnánk. Válasszunk ki egy v_1 forráspontot és legyen ez a sorrend első pontja. A $D - v_1$ digráf is aciklikus, ennek is létezik egy v_2 forráspontja. Ezt az eljárást folytatva megkapjuk a keresett $v_1, v_2, \dots, v_n$ topologikus sorrendet. $\square$

(Megjegyezzük, hogy megszámlálhatóan végtelen sok pontból álló aciklikus digráf pontjainak nem feltétlenül van topologikus sorrendje. Ezt példázza az a digráf, amelynek csúcsai a racionális számok és az u, v racionális számokra uv akkor él, ha $u < v$.)

A bizonyításból adódó algoritmus egyetlen forráspontot $O(m)$ lépésszámban tud megtalálni, így a topologikus sorrend megkeresésének összlépésszáma $O(mn)$. Mélységi keresés okos alkalmazásával a teljes topologikus sorrendet $O(m)$ lépésben meg lehet találni. Ennek érdekében feltehetjük, hogy a digráfnak van olyan s pontja, ahonnan minden más pont elérhető. Valóban, mert ha nem ez a helyzet, akkor adjunk a digráfhoz egy új s pontot, és vezessünk s-ből minden eredeti pontba élt. Így aciklikus digráfot kapunk, amely pontjainak topologikus sorrendjéből az újonnan hozzáadott s-t kihagyva megkapjuk az eredeti digráf pontjainak egy topologikus sorrendjét.

Indítsunk s-ből mélységi keresést, és jegyezzük fel a csúcsoknak az elhagyási sorrendjét (egy csúcsot akkor „hagyunk el", amikor már az összes belőle kilépő élt megnéztük, és visszalépünk a szülőjébe). Ha menet közben egy csúcsból megy él egy ősébe a fában, akkor megállhatunk, hiszen találtunk egy irányított kört, tehát a digráf nem aciklikus. Ha a mélységi keresés során nem találtunk kört, akkor a digráf aciklikus, és az elhagyási sorrend fordítottja topologikus sorrend. Ennek az az oka, hogy amikor egy csúcsot elhagyunk, akkor a belőle kilépő élek mind már korábban elhagyott csúcsokba mennek, hiszen az őseibe nem mehetnek élek.

9.4.2. Legolcsóbb utak aciklikus digráfban

Tegyük fel, hogy a D digráf aciklikus és az s pontból mindegyik másik pontba vezet egyirányú út, másszóval s-ből minden más pont elérhető. Legyen a $c : A \to \mathbb{R}$ súlyozás (vagy költségfüggvény) tetszőleges, tehát c-nek lehetnek pozitív és negatív komponensei is. Ez azért jó, mert a c negálása révén nem csupán a minimális költségű, hanem a maximális költségű (súlyú) út problémát is meg tudjuk oldani (tetszőleges irányított gráf esetén ez a feladat **NP**-teljes).

Aciklikus digráfban a következő egyszerű direkt eljárással fel lehet építeni a legolcsóbb utak fenyőjét. Az előző alfejezetben láttuk, miképp lehet egy topologikus sorrendet lineáris időben meghatározni. Tegyük fel, hogy az $s = v_1, v_2, \ldots, v_n$ topologikus sorrend első $j-1$ pontja által feszített részgráfban már meghatároztuk a legolcsóbb utak egy F_{j-1} fenyőjét a $\mu_c(v_i)$ költségekkel egyetemben ($1 \leq i \leq j-1$). Tekintsük a sorrendben következő v_j pontot. Miután v_j-be csak j-nél kisebb indexű pontból vezet él, a legolcsóbb sv_j-út költségét a

$$\mu_c(v_j) = \min\{\mu_c(v_i) + c(v_iv_j) : v_iv_j \in A\} \tag{7}$$

formula adja. Továbbá, ha v_iv_j jelöli azt az élt (pontosabban az egyik olyan élt), amelyen a minimum felvétetik, akkor $F_j := F_{j-1} + v_iv_j$ a legolcsóbb utak fenyője az első j csúcson. (ha több minimalizáló él van, mindegy, hogy melyikkel növeljük a fenyőt.) Ezt a rekurziót $j = 1, \ldots, n$-re követve a végül kapott F_n feszítő fenyő egy legolcsóbb utak fenyője lesz. Miután a $\mu_c(v_j)$ értékek illetve a fenyőbe kerülő v_iv_j élek kiszámításához a digráf minden élét egyszer kell csak tekintetbe venni, az algoritmus lépésszáma $O(m)$, felhasználva, hogy a topologikus sorrendet is $O(m)$ lépésben lehetett megkapni.

A (7) formulából adódik, hogy minden $v_iv_j \in A$ élre

$$\mu_c(v_j) \leq \mu_c(v_i) + c(v_iv_j),$$

míg ha v_iv_j az F_n fenyő éle, akkor itt egyenlőség teljesül:

$$\mu_c(v_j) = \mu_c(v_i) + c(v_iv_j). \tag{8}$$

Az algoritmus következményeként kapjuk az alábbi tételt.

9.2. Tétel. *$D = (V, A)$ aciklikus digráfban, amelyben minden pont elérhető s-ből, tetszőleges költségfüggvényre létezik legolcsóbb utak fenyője.* •

A legolcsóbb út költségére vonatkozik az alábbi min-max tétel.

9.3. Tétel. *Legyen $c : A \to \mathbb{R}$ a $D = (V, A)$ aciklikus irányított gráf élhalmazán egy tetszőleges költségfüggvény, és tegyük fel, hogy létezik st-út. Az s-ből t-be vezető utak költségének $\mu_c(t)$ minimuma egyenlő a*

$$\max\{\pi(t) - \pi(s) : \pi : V \to \mathbb{R},\ \pi(v) - \pi(u) \leq c(uv),\ uv \in A\}$$

értékkel. Ha c egészértékű, az optimális π is választható egészértékűnek.

Bizonyítás. Tetszőleges π-re és $P = \{s = v_0, v_1, \ldots, v_k = t\}$ st-útra

$$c(P) = \sum_i c(v_i v_{i+1}) \geq \sum_i [\pi(v_{i+1} - \pi(v_i)] = \pi(t) - \pi(s), \tag{9}$$

vagyis a tételben a max $\leq$ min irány következik. (Ehhez a egyenlőtlenséghez az aciklikusságra nincs is szükség).

A fordított irányú egyenlőtlenséget elég bebizonyítani abban a speciális esetben, amikor minden csúcs elérhető s-ből. Ha esetleg nem ez a helyzet, akkor egy új s' pontot a digráfhoz veszünk egy 0 költségű $s's$ éllel valamint minden $v \in V$-re egy M költségű $s'v$-éllel, ahol M kellően nagy. Az így nyert digráfot és költségfüggvényt jelölje D' és c'. Ekkor D'-ben minden $v \in V$ pont elérhető s'-ből és $\mu(v) = \mu'(v)$, ahol $\mu'(v)$ a legolcsóbb $s'v$-út c'-költsége D'-ben. Legyen most π' egy olyan függvény V'-n, amelyre $\pi'(v) - \pi'(u) \leq c'(uv)$ minden $uv \in A'$ élre és $\pi'(t) - \pi'(s') = \mu'(t)$. Ekkor a $\pi := \pi'|V$ függvényre (ami tehát π' megszorítása V-re) $\pi(v) - \pi(u) \leq c(uv)$ minden $uv \in A$ élre. Továbbá, $\pi(s) - \pi'(s') = \pi'(s) - \pi'(s') \leq c'(s's) = 0$ miatt $\pi(s) \leq \pi'(s')$ és ezért $\mu(t) \geq \pi(t) - \pi(s) = \pi'(t) - \pi(s) \geq \pi'(t) - \pi'(s') = \mu'(t) = \mu(t)$, amiből $\mu(t) = \pi(t) - \pi(s)$. Tehát, ha D'-re igaz a tétel, akkor D-re is, és emiatt feltehetjük, hogy D-ben minden csúcs elérhető s-ből.

Tekintsük a fenti algoritmus által szolgáltatott P utat, azaz az F_n fenyőben lévő egyértelmű st-utat. Minden $v \in V$-re legyen $\pi(v) := \mu_c(v)$ a legolcsóbb sv-út költsége. Ekkor egyrészt minden uv élre egy legolcsóbb su utat az uv éllel kiegészítve (az aciklikusság miatt) egy sv-utat kapunk, és ezért $\pi(v) \leq \pi(u) + c(uv)$, azaz $\pi(v) - \pi(u) \leq c(uv)$, másrészt (8) miatt P minden uv élére $\pi(v) - \pi(u) = c(uv)$, vagyis (9)-ben egyenlőség áll, és így $c(P) = \pi(t) - \pi(s)$. □

Alkalmazásokban előfordul, hogy aciklikus digráfban minimális helyett maximális súlyú st-utat kell keresni. A fenti eljárást ekkor a $-c$ költségfüggvényre kell alkalmazni, de az eredeti c nyelvén közvetlenül is megfogalmazhatjuk, a következőképpen.

Tegyük fel, hogy a topologikus sorrend első $j - 1$ pontja által feszített részgráfban már meghatároztuk a legsúlyosabb utak egy F_{j-1} fenyőjét a legsúlyosabb sv_i-út $\tau_c(v_i)$-vel jelölt súlyával egyetemben ($1 \leq i \leq j - 1$). A sorrendben következő v_j pontra legyen $\tau_c(v_j) := \max\{\tau_c(v_i) + c(v_i v_j) : v_i v_j \in A\}$ és legyen $F_j := F_{j-1} + v_i v_j$, ahol a $v_i v_j$ egy olyan él, amelyen a maximum felvétetik.

Fogalmazzuk meg a 9.3 tétel ellenpárját erre az esetre. A keveredés elkerülése érdekében π helyett τ-t használunk, és az utána következő alkalmazás érdekében felcseréljük a két oldalt.

9.4. Tétel. *Legyen $c : A \to \mathbb{R}$ a $D = (V, A)$ aciklikus irányított gráf élhalmazán egy tetszőleges súlyfüggvény, és tegyük fel, hogy az s pontból minden pontba vezet egyirányú út. Ekkor*

$$\min\{\tau(t) - \tau(s) : \tau : V \to \mathbb{R},\ \tau(v) - \tau(u) \geq c(uv), uv \in A\} \\ = \max\{c(P) : P \text{ út } s\text{-ből } t\text{-be}\}.$$ •

9.4.3. Egy projekt-ütemezési feladat: a PERT módszer

Alkalmazásokban előfordul, hogy nem elsősorban az optimális útra vagyunk kíváncsiak, hanem inkább az optimális τ függvényre. Tekintsük a következő ütemezési feladatot. Egy projekt különféle elemi feladatok elvégzéséből áll, melyeknek előre adott a végrehajtási ideje. (Például házépítésnél az alapok kiásása, betonozás, első szint felhúzása, második szint felhúzása, tető építés, belső vakolás, vízcső szerelés, villanyvezetékek, festés, ablakok, stb). Tudjuk továbbá, hogy technológiai előírások miatt bizonyos részfeladatok megelőznek másokat (az alapozás a fürdőszoba csempézés előtt van), azaz a részfeladatok halmazán adott egy részbenrendezés. Kérdés, hogy mi az a legrövidebb idő, amely alatt a teljes projekt elvégezhető azon kikötés mellett, hogy az egyes részfeladatok kezdési időpontját úgy megadni, hogy minden feladat kezdésére a részbenrendezésben őt megelőzők már elkészüljenek.

A megoldáshoz készítsünk el egy D irányított gráfot a következőképpen. Mindegyik z részfeladatot reprezentáljuk egy $u_z v_z$ irányított éllel, amelynek súlya legyen a z végrehajtási ideje. Amennyiben a z feladat technológiailag megelőzi az y feladatot, úgy vegyünk be D-be egy $v_z u_y$ élt 0 súllyal. Végül adjunk a digráfhoz egy s forráspontot, amelyből minden u_x pontba vezessünk 0 súlyú élt, és adjunk egy t nyelőpontot, amelybe minden v_x pontból vezessünk 0 súlyú élt. Feladatunk olyan $\tau(v)$ időpontok kijelölésével ekvivalens, amelyekre $\tau(s) = 0$, $\tau(t)$ minimális és minden uv élre a $\tau(v) - \tau(u)$ időpont-különbség legalább akkora, mint az él súlya.

A 9.4 tétel pontosan erre a kérdésre adott választ: *a projekt végrehajtásának minimális össz-ideje (vagyis $\tau(t) - \tau(s)$ minimuma) egyenlő a legsúlyosabb st-út súlyával.* Egy ilyen utat szoktak néha **kritikus út**nak nevezni, míg az előzőekben leírt minimális út problémának az itteni feladatra adaptált változatát kritikus út módszernek (angolul **PERT**: project evaluation and review technique).

A 9.4 tétel egy szemléletes interpretációja révén a főnökünket rögtön meg tudjuk arról győzni, hogy az általunk javasolt optimális ütemezés (amit tehát kezdési időpontokat meghatározó τ függvény ír le) valóban az elvileg legkorábbi befejezést garantálja, hiszen már a kritikus P úton lévő elemi munkák elvégzéséhez is annyi idő kell, mint a P út össz-súlya (vagyis a P úton lévő elemei munkák össz-ideje), márpedig mi a teljes projektet is ennyi idő alatt le tudjuk bonyolítani, és így az ütemezés szükségképpen optimális.

9.5. Legolcsóbb utak nem-negatív költségekre: Dijkstra algoritmusa

Tegyük most fel, hogy D tetszőleges, de a c költségfüggvény nem-negatív. Dijkstra algoritmusa a feladat megoldására két ötleten múlik. Az aciklikus esethez hasonlóan itt is s-ből induló legolcsóbb utaknak egy fenyőjét építjük fel élek egyenkénti hozzávételével. Lényeges különbség azonban, hogy a pontoknak a fenyőbe kerülési sorrendjét nem lehet előre megmondani (mint ahogy az aciklikus esetben a topologikus sorrenddel meg lehetett), mert az csak menetközben derül ki, a következő lemma szerint.

9.5. Lemma. *Legyen T egy legolcsóbb utak s-fenyője a $V(T)$ ponthalmazon. Tegyük fel, hogy az*

$$m_T := \min\{\mu_c(u) + c(uv) : uv \text{ kilép } V(T)\text{-ből}\} \tag{10}$$

minimum valamely $a = u_a v_a$ élen vétetik fel. Ekkor $T' := T + a$ is legolcsóbb utak s-fenyője a $V(T) + v_a$ ponthalmazon.

Bizonyítás. A T-re vonatkozó feltevés miatt csak az s-ből v_a-ba vezető T'-beli P' útról kell belátnunk, hogy D-ben legolcsóbb. A jelölések folytán $c(P') = m_T$. Legyen P tetszőleges sv_a-út D-ben. Legyen ennek (s felől indulva) az első $V(T)$-ből kilépő éle $e = u_e v_e$, míg az s-től u_e-ig tartó részútja P''. A c nem-negativitása valamint m_T és $\mu_c(u_e)$ jelentése miatt $c(P') = m_T \le \mu_c(u_e) + c(e) \le c(P'') + c(e) \le c(P)$. □

A 9.5 lemma kézenfekvő és hatékony megoldást kínál a legolcsóbb utak fenyőjének kiszámítására. Kiindulva az egyetlen s pontból álló fenyőből, élek egymás utáni hozzávételével, $n-1$ fázisban, felépítjük a V-t feszítő legolcsóbb utak fenyőjét. Ehhez csak az kell, hogy egy közbenső, már kiszámított T fenyőhöz meg tudjuk határozni a hozzáveendő élt. A 9.5 lemma alapján ez a (10) minimum kiszámításával megtehető. Kérdés, hogy hány lépésben. A naív megközelítés szerint e minimum meghatározásához számba kell venni az összes $V(T)$-ből kilépő élt. Ezek számára m-nél jobb felső korlátot nem lehet biztosítani, és ezért ez a megközelítés összességében egy $O(mn)$ lépésszámú algoritmust eredményez.

Dijkstra algoritmusának második ötlete az, hogy fenntartunk és menetközben alkalmasan módosítunk bizonyos adatokat, amelyek segítségével a (10) minimum $O(m)$ lépés helyett már $O(n)$ lépésben kiszámítható.

Tegyük fel, hogy egy közbenső fázisban a T fenyőn, valamint a T pontjaira már kiszámolt $\mu_c(v)$ értékeken kívül rendelkezésre állnak a következő adatok. Minden $v \in V - V(T)$ fenyőn kívüli pontra a $\mu_T(v)$ címke tartalma legyen az s-ből v-be vezető, csak $V(T)$ pontjait használó sv-utak költségének minimuma, míg az $e_T(v)$ címke tartalma egy ilyen sv-út utolsó éle (amely tehát kilép T-ből és a hegye v). Ezen kívül a v_T címke tartalma az a $z \in V - V(T)$ pont, ahol a $\mu_T(v)$ értékek minimuma ($v \in V - V(T)$) felvétetik, míg m_T a minimum értéke.

Ezen címkék segítségével rögtön meg tudjuk mondani, hogy melyik élt kell T-hez venni. Nevezetesen, tekintjük a v_T-ben lévő z pontot és az $e_T(z)$-ben lévő uz élt és ezt adjuk T-hez. A lemma miatt $\mu_c(z) = \mu_T(z)$.

A keletkező T' fenyőhöz tartozó címkék módosításához figyeljük meg, hogy egy $v \in V - T'$ pontra a legolcsóbb olyan sv-út, amely csak T'-beli pontot használ vagy használja a z pontot vagy nem, attól függően, hogy a $\mu_{T'}(v) := \min\{\mu_T(v),\ \mu_c(z) + c(zv)\}$ minimum a második vagy az első tagon vétetik-e fel. Ennek eldöntése tehát konstans lépésben megtehető. Miután n pont van összesen, megállapíthatjuk, hogy a T fenyő egy újabb éllel való növelésekor minden $\mu_{T'}(v)$ címke $O(n)$ lépésben meghatározható. Hasonlóan egyszerű megfontolás nyomán a $v_{T'}$ és az $m_{T'}$ címkék is $O(n)$ lépésben meghatározhatók. Miután $n-1$ fenyő növelést hajtunk végre, a Dijkstra algoritmus teljes lépésszáma $O(n^2)$.

10. Párosítások páros gráfban

10.1. Maximális elemszámú párosítások: a javító utak módszere

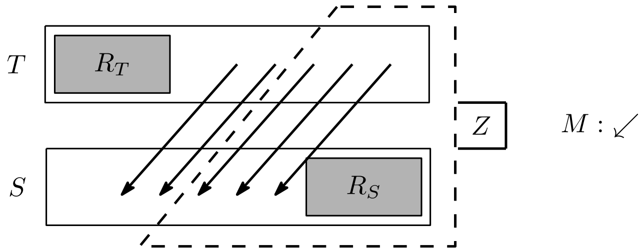
Vizsgálatainkat kezdjük a súlyozatlan eset áttekintésével. A kiindulási eredmény Kőnig Dénes tétele.

10.1. Tétel (Kőnig). *Egy $G = (S, T; E)$ páros gráfban a maximális párosítás $\nu = \nu(G)$ elemszáma egyenlő az éleket lefogó pontok minimális $\tau = \tau(G)$ elemszámával.*

Bizonyítás. Egy ν elemű párosítás lefogásához kell legalább ν csúcs, így az összes élhez is kell ennyi, ezért $\nu \le \tau$.

A nemtriviális $\nu \ge \tau$ irány igazolásához konstruálunk egy $M \subseteq E$ párosítást és egy $L \subseteq S \cup T$ lefogást, melyekre $|M| = |L|$. Az eljárás a G egy tetszőleges M párosításából indul ki, ami kezdetben az üres halmaz is lehet. Az általános lépésben vagy találunk egy nagyobb párosítást, és ekkor a nagyobb párosításra vonatkozóan iteráljuk az eljárást, vagy pedig egy $|M|$-mel megegyező elemszámú lefogást, amikor is az algoritmus véget ér.

Irányítsuk meg M éleit T-től S felé, míg az összes többi élt S-től T felé. Jelölje R_S illetve R_T az S-ben illetve a T-ben az M által fedetlen pontok halmazát. Jelölje Z az R_S pontjaiból az így kapott D_M irányított gráfban irányított úton elérhető pontok halmazát (amit például szélességi kereséssel találhatunk meg).



1. ábra. A Kőnig tétel bizonyításának illusztrálása

Két eset lehetséges. Amennyiben R_T-nek esik pontja Z-be, akkor megkaptunk egy olyan R_S-t és R_T-t összekötő P utat, amely M-ben alternál. Most M és P szimmetrikus differenciája egy M-nél eggyel több élből álló M' párosítás. (Technikailag az eljárást könnyű végrehajtani: a megtalált út éleinek irányítását egyszerűen megfordítjuk. Az így nyert átirányított gráf éppen $D_{M'}$, vagyis az a digráf, amit G-ből kapunk az M' éleinek T-ből S felé, és a többi élnek S-től T felé történő irányításával.)

A másik esetben R_T diszjunkt Z-től. Z definíciója folytán Z-ből nem lép ki irányított él. Érvényes továbbá, hogy Z-be nem lép be megirányított $uv \in M$ párosítás él, hiszen v csak u-n keresztül érhető el, így v csak akkor lehetett irányított úton elérhető R_S-ből, ha u is az volt. Tehát az M minden eleme vagy teljesen Z-ben fekszik vagy teljesen kívüle.

Következik, hogy az $L := (T \cap Z) \cup (S - Z)$ halmaz egyrészt lefogja a G összes élét, másrészt minden M-beli élnek pontosan az egyik végpontját tartalmazza, tehát $|M| = |L|$, amivel a Kőnig tétel bizonyítása teljes. $\square$

A fenti bizonyítás egyúttal hatékony eljárást is jelent a szóbanforgó optimumok meghatározására. Az algoritmust **Kőnig** (alternáló utas) **algoritmus**ának nevezzük. A lépésszám megbecsléséhez figyeljük meg, hogy legfeljebb $n/2$ alkalommal kell utat keresnünk. Miután egyetlen út megkeresése az élszámmal arányos időben történhet, az összlépésszám nem nagyobb, mint $O(nm)$ (ahol n a gráf pontszáma, míg m az élszáma).

10.2. Tétel (Hall)**.** *$G = (S, T; E)$ páros gráfban akkor és csak akkor létezik S-t fedő párosítás, ha teljesül a Hall-féle feltétel:*

$$|\Gamma(X)| \geq |X| \text{ minden } X \subseteq S \text{ részhalmazra},$$

ahol $\Gamma(X)$ azon T-beli pontok halmaza, amiknek legalább 1 X-beli szomszédja van.

Bizonyítás. A Kőnig algoritmus futásának végén kapott elérhető pontok Z halmaza az S beli fedetlen pontok R_S halmazából és bizonyos párosítás élek 2 végpontjából áll. A $H := Z \cap S$ halmazra $\Gamma(H) = Z \cap T$, így ha $R_S \neq \emptyset$, akkor $|\Gamma(H)| < |H|$. $\square$

10.2. Maximális súlyú teljes párosítások: a Magyar módszer

Tételezzük fel, hogy a $G = (S, T; E)$ páros gráf élein adott egy c súlyfüggvény. Tegyük fel, hogy G-nek létezik teljes párosítása, és vizsgáljuk meg a maximális súlyú teljes párosítás megkeresésének problémáját. Az első ezzel kapcsolatos kérdés az, hogy milyen hatékonyan ellenőrizhető igazolványt (tanúsítványt) tudunk elképzelni egy kiválasztott teljes párosítás súlyának maximalitására, annak mintájára, ahogy egy megadott M párosítás maximális elemszámára igazolvány az éleknek egy $|M|$ elemszámú lefogása. Ennek általánosításaként nevezzünk egy csúcsokon értelmezett $\pi : V \to \mathbb{R}$ függvényt **lefogó vektor**nak, ha minden $uv \in E$ élre

$\pi(u)+\pi(v) \geq c(uv)$. (Figyeljük meg, hogy ha c azonosan 1, akkor egy $(0,1)$-értékű lefogó vektor épp az élhalmaz egy lefogásának incidencia vektora). Egy élt **pontos**nak fogunk nevezni (π-re nézve), ha itt egyenlőség áll. A π lefogó vektor$\pi(V)$ **összértékén** a $\sum[\pi(v) : v \in V]$ összeget értjük, ahol $V = S \cup T$. A főtétel Egerváry Jenőtől származik 1931-ből.

10.3. Tétel (Egerváry). *A $G = (S, T; E)$ teljes párosítással rendelkező páros gráfban a $c : E \to \mathbb{R}_+$ súlyfüggvényre vonatkozó maximális súlyú teljes párosítás ν_c súlya egyenlő a lefogó vektorok minimális τ_c összértékével. Amennyiben c egészértékű, úgy az optimális lefogó vektor is választható annak. Amennyiben G teljes páros gráf és c nemnegatív, úgy az optimális lefogó vektor választható nemnegatívnak is.*

Bizonyítás. Megjegyezzük, hogy a tétel implicit azt is tartalmazza, hogy a szóbanforgó minimum létezik. Mivel a maximumot a teljes párosítások véges halmazán tekintjük, így annak létezése nem kérdéses.

A harmadik rész igazolásához azt mutatjuk meg, hogy tetszőleges π lefogó vektor nemnegatívvá alakítható az összérték megváltoztatása nélkül. Legyen a π legkisebb értéke $-K$ (ahol $K > 0$) és legyen mondjuk az S-ben $-K$ értékű pont. Mivel G teljes páros gráf, c nemnegatív és π lefogó vektor, így minden $v \in T$ pontra $\pi(v) \geq K$. Az S elemein a π értékeket egységesen K-val növelve, T elemein pedig K-val csökkentve olyan nemnegatív lefogó vektort kapunk, melynek összértéke $|S| = |T|$ miatt szintén $\pi(V)$.

A tétel min = max részének igazolásához először azt látjuk be, hogy max $\leq$ min. Tekintsük ehhez a gráf egy tetszőleges $M := \{u_1v_1, u_2v_2, \ldots, u_nv_n\}$ teljes párosítását valamint a c-nek egy π lefogó vektorát. Ekkor $c(M) := \sum c(u_iv_i) \leq \sum[\pi(u_i) + \pi(v_i) : i = 1, \ldots, n] = \pi(V)$, amiből $\nu_c \leq \pi_c$ következik. Itt egyenlőség pontosan akkor áll, ha M minden élen pontos.

A fordított irányú egyenlőtlenség bizonyításához tehát kell találnunk egy alkalmas π lefogó vektort és egy olyan teljes párosítást, amely pontos élekből áll. Más szóval olyan π-t kell keresnünk, hogy a π-re vonatkozó pontos élekből álló részgráf tartalmazzon teljes párosítást.

Erre szolgál a H. Kuhn által bevezetett Magyar módszer. Tetszőleges π lefogó vektorral indulunk, amely egészértékű, ha c az. Az általános lépésben tekintjük a pontos élek által alkotott $G_\pi = (S, T; E_\pi)$ részgráfot és ezen futtatjuk a Kőnig tétel fentebb leírt bizonyításának algoritmusát. Kiindulunk tehát a G_π-nek egy tetszőleges M párosításából. Megirányítjuk az M-beli éleket T-től S felé, míg az összes többi G_π-beli élt S-től T felé. Jelölje R_S illetve R_T az S-ben illetve a T-ben az M által fedetlen pontok halmazát. Jelölje Z az R_S pontjaiból az így kapott irányított gráfban irányított úton elérhető pontok halmazát.

Amennyiben R_T-nek esik pontja Z-be, úgy megkaptunk egy olyan R_S-t és R_T-t összekötő P utat, amely M-ben alternál. Az M és P szimmetrikus differenciája egy M-nél eggyel több élből álló M' párosítást alkot. Az M'-vel folytatva iteráljuk az eljárást.

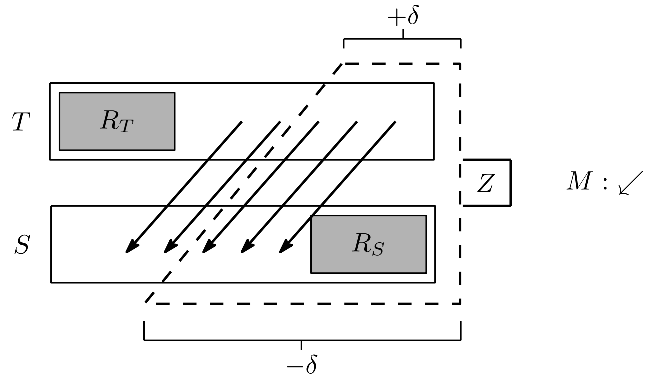
Nézzük most a másik lehetőséget, amikor R_T diszjunkt Z-től. Z definíciója folytán Z-ből nem vezet ki irányított él és Z-be nem lép be megirányított $uv \in M$ párosítás él. Legyen $H := Z \cap S$. Mivel G-nek van teljes párosítása, biztosan van olyan e éle G-nek, amely H és $T - \Gamma_{G_\pi}(H)$ között vezet, ahol $\Gamma_{G_\pi}(H)$ jelöli a H szomszédainak halmazát a G_π-ben. Ilyen él nem lehet pontos, így a

$$\delta := \min\{\pi(u) + \pi(v) - c(uv) : uv \in E, u \in H, v \in T - \Gamma_{G_\pi}(H)\}$$

érték pozitív. Módosítsuk π-t a következőképp:

$$\pi'(v) = \begin{cases} \pi(v) - \delta, & \text{ha } v \in H, \\ \pi(v) + \delta, & \text{ha } v \in \Gamma_{G_\pi}(H), \\ \pi(v) & \text{különben.} \end{cases}$$

A δ választása miatt az így módosított π' továbbra is lefogó vektor, amely egészértékű, ha c és π az volt. A π'-re vonatkozó pontos élek $G_{\pi'}$ gráfja és G_π ugyanazon éleket feszíti



2. ábra. A π változtatása a Magyar módszer egy lépésében

Z-ben, továbbá $G_{\pi'}$-nek van legalább egy éle (ahol a δ-t definiáló minimum felvétetett) H és $T - \Gamma_{G_\pi}(H)$ között, ezért $G_{\pi'}$-ben az R_S-ből elérhető pontok halmaza szigorúan bővebb, mint G_π-ben. (Figyelem: az NEM igaz, hogy $G_{\pi'}$-ben biztosan több pontos él van, mint G_π-ben.)

Emiatt a $G_{\pi'}$-höz és M-hez rendelt irányított gráfban az R_S-ből elérhető pontok halmaza szigorúan bővebb. Így egy fázis (ami során tehát a pontos élek gráfjában a maximális párosítás elemszáma nem nő) legfeljebb $|S|$ útkereső eljárás alkalmazása után véget ér. Ezzel a min-max tétel bizonyítását befejeztük. A tétel második állításához figyeljük meg, hogy ha c egészértékű, akkor a fenti eljárás során π egészértékűsége végig megőrződik. $\square$

Mivel egy útkeresés $O(|E|)$ lépésben végrehajtható és legfeljebb $|S|$ fázis van, az algoritmus teljes futásideje $O(|E||S|^2)$.

10.3. Maximális súlyú párosítások

Megmutatjuk, hogy a maximális súlyú (nem feltétlenül teljes) párosítás meghatározásának problémája egyszerű fogással visszavezethető a maximális súlyú teljes párosítására.

10.4. Tétel. *Egy $G' = (S', T'; E')$ páros gráfban nemnegatív c súlyfüggvény esetén a párosítások maximális ν'_c súlya egyenlő a nemnegatív (!) lefogó vektorok minimális τ'_c összértékével. Amennyiben c egészértékű, az optimális π'_c is választható egészértékűnek.*

Bizonyítás. A $\nu'_c \leq \tau'_c$ egyenlőtlenség nyilvánvaló, így csak a fordított iránnyal foglalkozunk. Új pontok esetleges hozzávételével elérhetjük, hogy a páros gráf két osztálya egyforma méretű legyen. Egészítsük ki a gráfot 0 súlyú élek bevételével egy G teljes páros gráffá. A súlyfüggvény ezen kiterjesztését továbbra is jelölhetjük c-vel. A 10.3 tétel szerint G-nek létezik egy M teljes párosítása és c-nek egy π nemnegatív lefogó vektora, melyekre $c(M) = \pi(V)$. Mivel az új élek súlya 0, így az új élek kihagyásával M-ből keletkező G'-beli M' párosítás súlya változatlanul $c(M)$. Továbbá, mivel M minden éle pontos, ezért egy 0 súlyú uv élének végpontjaira $\pi(u) = \pi(v) = 0$. Emiatt π értéke az új pontokon 0, hiszen új pontból csak 0 súlyú él megy ki.

Ha tehát π-t megszorítjuk az eredeti $V' = S' \cup T'$ ponthalmazra, akkor a keletkező π'-re $\pi'(V') = \pi(V)$ és $\pi'(V') = c(M')$. $\square$

11. Folyamok

Jelöljön $D = (V, E)$ egy irányított gráfot. Valamely $x : E \to \mathbb{Q}$ függvényre és $S \subseteq V$ részhalmazra legyen $\varrho_x(S) := \sum[x(uv) : uv \in E, uv \text{ belép } S\text{-be}]$ és legyen $\delta_x(S) := \varrho_x(V - S)$.

Adott egy $D = (V, E)$ irányított gráf, továbbá D-nek egy kijelölt s forráspontja (source) és egy t nyelőpontja (sink). A továbbiakban, amikor folyamokról lesz szó, végig feltesszük, hogy s-be nem lép be él és t-ből nem lép ki él. **Folyam**on egy olyan $x : E \to \mathbb{Q}_+$ nem-negatív függvényt értünk, amely minden, s-től és t-től különböző pontra teljesíti a megmaradási szabályt, azaz $\varrho_x(v) = \delta_x(v)$ fennáll minden $v \in V - \{s, t\}$ csúcsra. Ha adott egy $g : E \to \mathbb{Q}_+$ kapacitásfüggvény is, akkor az x folyam megengedett, ha $x \leq g$ feltétel is teljesül, g-**megengedett** (röviden, **megengedett**) **folyam**ról beszélünk.

Egy s-et tartalmazó, de t-t nem tartalmazó X halmazt nevezzünk $s\bar{t}$-halmaznak. Tetszőleges S $s\bar{t}$-halmaz és x folyam esetén

$$\delta_x(s) = \delta_x(s) - \varrho_x(s) = \sum[\delta_x(v) - \varrho_x(v) : v \in S] = \delta_x(S) - \varrho_x(S).$$

Vagyis minden S $s\bar{t}$-halmazra a $\delta_x(S) - \varrho_x(S)$ értékkel definiált **nettó kiáramlás** független az S választásától. Ezt a közös, $\delta_x(s)$-sel (és $\varrho_x(t) = \delta_x(V - t)$-vel) egyenlő értéket nevezzük az x folyam **nagyság**ának. Az $x(uv)$ szám a folyam **értéke** az $uv \in E$ élen.

A fő kérdés, hogy miként lehet meghatározni egy maximális nagyságú (egészértékű) folyamot, illetve adott költségfüggvény esetén hogyan lehet kiszámítani egy előre adott k értékre a k nagyságú folyamok közül a minimális költségűt. (Valamely $c : E \to \mathbb{Q}$ költségfüggvényre nézve a $cx := \sum[c(e)x(e) : e \in E]$ skalárszorzatot nevezzük az x folyam **költség**ének.)

11.1. Tétel (Maximális Folyam Minimális Vágás tétel)**.** *A megengedett st-folyamok maximális nagysága egyenlő a $\delta_g(S)$ értékek minimumával, ahol a minimum az összes $s\bar{t}$-halmazra megy. Ha g egészértékű, úgy a maximum egészértékű folyamon is felvétetik.*

A tételt úgy bizonyítjuk, hogy megadunk egy algoritmust, ami megtalál egy maximális folyamot és egy minimális vágást.

Legyen x megengedett folyam. Ekkor tetszőleges S $s\bar{t}$-halmaz esetén az x folyam nagyságára érvényes az alábbi becslés. $\delta_x(s) = \delta_x(s) - \varrho_x(s) = \sum[\delta_x(v) - \varrho_x(v) : v \in S] = \delta_x(S) - \varrho_x(S) \leq \delta_g(S)$. Ebből adódik a Maximális Folyam Minimális Vágás tételben a max $\leq$ min egyenlőtlenség.

Az is megállapítható, hogy egy x folyam bizonyosan maximális nagyságú, amennyiben létezik egy olyan S $s\bar{t}$-halmaz, amelyre teljesülnek az alábbi **optimalitási feltételek**.

(a) $x(e) = g(e)$ minden e élre, amely kilép S-ből, és
(b) $x(e) = 0$ minden e élre, amely belép S-be.

Jelen célunk egy ilyen x folyam és S halmaz algoritmikus megkeresése.

11.1. Ford és Fulkerson algoritmusa

A Fordtól és Fulkersontól származó algoritmus tetszőleges x megengedett st-folyamból indul ki (például $x \equiv 0$), és azt iteratívan javítja. Készítsünk el egy $D_x = (V, E_x)$ segédgráfot a következőképp. Egy uv él E_x-hez tartozik, ha vagy (i) $uv \in E$ és $x(uv) < g(uv)$, és ekkor ezen élet **előre-él**nek hívjuk D_x-ben, vagy (ii) $vu \in E$ és $x(vu) > 0$, és ekkor uv neve **hátra-él**. Jelölje S az s-ből D_x-ben irányított úton elérhető pontok halmazát.

1. eset *$t \notin S$, azaz t nem érhető el s-ből.* Mivel D_x semelyik éle sem lép ki S-ből, ezért D-ben minden S-ből kilépő él telített (azaz $x(uv) = g(uv)$) és minden S-be belépő uv élen $x(uv) = 0$. Vagyis az (a) és (b) optimalitási feltételek teljesülnek és az algoritmus véget ér: az adott x folyam nagysága egyenlő $\delta_g(S)$-sel.

2. eset *$t \in S$, azaz t elérhető s-ből.* Legyen P tetszőleges s-ből t-be vezető irányított út D_x-ben.

Legyen $\Delta_1 := \min\{g(uv) - x(uv) : uv$ előre-éle P-nek$\}$ és $\Delta_2 = \min\{x(vu) : uv$ hátra-éle P-nek$\}$. Legyen $\Delta = \min\{\Delta_1, \Delta_2\}$. Ekkor Δ pozitív. Nevezzük P egy élét **kritikusnak**, ha Δ ezen az élen éretik el.

Módosítsuk x-et a következőképp. Ha uv előre-éle P-nek, úgy a D uv élén növeljük $x(uv)$-t Δ-val. Ha uv hátra-éle P-nek, úgy a D vu élén csökkentsük $x(vu)$-t Δ-val. Könnyen látható, hogy a módosított x' megengedett folyam lesz, amelynek nagysága Δ-val nagyobb, mint x-é. Következésképp, ha g egészértékű, akkor a 2. eset csak véges sokszor fordulhat elő, vagyis véges sok növelés után az 1. eset következik be, amikor is az algoritmus véget ér. Tehát egész kapacitások esetén az MFMC tétel bizonyítást nyert.

Amennyiben g racionális, a nevezők legkisebb közös többszörösével a kapacitásokat végigszorozva visszajutunk az egész kapacitású esethez. •

11.1. Megjegyzés. Ha g irracionális, akkor a fenti eljárás nem biztosan ér véget véges sok lépésben. Másik hátrány, hogy még egész kapacitások esetén is az iterációk száma arányos lehet az előforduló legnagyobb kapacitás nagyságával. Így az algoritmus bonyolultsága az input méretének exponenciális függvényével arányos, azaz nem polinomiális.

Mindjárt belátjuk azonban, hogy ha a 2. esetben P-t legrövidebb st-útnak választjuk, akkor az algoritmus polinomiális.

11.2. Tétel (Edmonds, Karp és Dinitz)**.** *Ha a Ford–Fulkerson algoritmusban mindig a legrövidebb növelő utat használjuk, akkor az eljárás tetszőleges g kapacitásfüggvény esetén legfeljebb $O(|V||A|)$ növelés után véget ér.*

Bizonyítás. Jelölje $\sigma_x(v)$ a v távolságát D_x-ben s-től. (Ha egyáltalán nincs s-ből v-be út, akkor $\sigma_x(v) := \infty$). Legyen P egy legrövidebb út D_x-ben s-ből t-be. Ekkor P mindegyik uv élére, $\sigma_x(v) = \sigma_x(u) + 1$.

11.1. Állítás. *Amikor P mentén végrehajtunk egy növelést, a $\sigma_x(v)$ érték semmilyen v-re sem csökken.*

Bizonyítás. Nézzük meg milyen hatással van a növelés a D_x segédgráfra. Mivel a folyamot D-nek csak olyan élein változtattuk, melyek P éleinek felelnek meg, D_x csupán P éleinél változhat. Éspedig, D'_x lehetséges új élei P élei megfordítva, ugyanakkor P kritikus élei (ahol Δ felvétetik) eltűnnek D_x-ből. A v pont s-től való távolsága csak akkor csökkenhetne, ha olyan uw éleket adnánk a segédgráfhoz, melyekre $\sigma_x(w) > \sigma_x(u) + 1$, amiből az állítás következik. □

A növelések sorozatát fázisokra bontjuk. Egy fázis során $\sigma_x(t)$ ugyanaz marad. A lemma szerint legfeljebb $|V| - 1$ fázis lehetséges.

11.2. Állítás. *Egy fázison belül legfeljebb $|A|$ növelésre kerülhet sor.*

Bizonyítás. Jelölje $\sigma_i(v)$ a v pont távolságát s-től az i fázis kezdetén az aktuális segédgráfban. Nevezzünk egy uv élt **i-szorosnak**, ha $\sigma_i(v) = \sigma_i(u) + 1$. Az i-dik fázis során csupán i-szoros éleket használunk. Tudjuk, hogy egy növelés legalább egy i-szoros élt eltüntet az aktuális segédgráfból és nem hoz be új i-szoros élt. Mivel a segédgráfnak legfeljebb $|A|$ darab i-szoros éle van, a lemma következik. □

Mindezeket összetéve kapjuk, hogy legfeljebb $|V||A|$ növelésre van szükségünk, így az algoritmus futási ideje $O(|V||A|^2)$, hiszen egyetlen növelés $O(|A|)$ lépést igényel. □

12. 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.

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

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

A következő alakú hálózati feladatot szeretnénk megoldani:

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

12.1. 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:

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

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

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

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

12.1. Definíció. Egy mátrix **teljesen unimoduláris** (röviden TU), ha minden négyzetes részmátrixának determinánsa 0, 1, vagy -1.

12.1. Állítás. *Ha az A mátrix TU, akkor a következő mátrixok is TU mátrixok:*

- *A transzponáltja*
- (A, e_i)*, ahol* e_i *az i-edik egységvektor, azaz az i-edik koordinátája 1, a többi 0*
- *Ha A egy oszlopát megszorozzuk* -1*-gyel*
- (A, a)*, ahol az a vektor az A mátrix valamelyik oszlopa.*

Bizonyítás. Mindegyik esetben igaz az, hogy az új mátrix minden nemszinguláris részmátrixának a determinánsa megegyezik az A mátrix egy nemszinguláris részmátrixának determinánsával, vagy annak -1-szeresével. □

12.1. Tétel. *Ha egy A mátrix minden oszlopában legfeljebb egy* $+1$*-es és legfeljebb egy* -1*-es található, akkor A teljesen unimoduláris.*

Bizonyítás. A mátrix mérete szerinti indukcióval bizonyítunk. 1×1-es mátrixra természeresen igaz az állítás. Ha az A mátrix nem négyzetes, akkor minden négyzetes részmátrixa valódi részmátrix, és ezért indukció szerint A is TU. Tegyük fel, hogy az A mátrix négyzetes; mivel A minden valódi részmátrixáról indukció szerint tudjuk, hogy determinánsa 0, 1, vagy -1, elég A determinánsát ellenőrizni. Ha A-nak van olyan oszlopa, amiben egyetlen nemnulla elem van, akkor a kifejtési tétel miatt A determinánsának abszolút értéke megegyezik egy valódi részmátrix determinánsának abszolút értékével, így indukció szerint 0, 1, vagy -1, tehát A TU.

Tegyük most fel, hogy az A mátrix négyzetes, és minden oszlopában 1 db 1 és 1 db -1 van. Ekkor A sorainak összege a 0-vektor, tehát A sorai lineárisan összefüggőek, következésképp determinánsa 0. □

12.2. Tétel. *Ha az A mátrix TU, akkor az $\{Ax = b,\ x \geq 0\}$ valamint az $\{Ax \leq b, x \geq 0\}$ rendszerek bázismegoldásai egész b esetén egész vektorok.*

Bizonyítás. Az első rendszer esetében egy B bázis az A mátrix nemszinguláris $m \times m$-es részmátrixa. A Cramer szabály értelmében $Bx = b$ egyértelmű megoldása egész, hiszen a Cramer szabályban a számláló egész, a nevező pedig 1 vagy -1. A második rendszer esetében egy B bázis az (A, I) mátrix egy nemszinguláris $m \times m$-es részmátrixa, amiből az I-beli részt kifejtve következik, hogy B determinánsának abszolút értéke megegyezik A egy aldeterminánsának abszolút értékével. Mivel A teljesen unimoduláris, ez 1, tehát a Cramer szabályban itt is 1 vagy -1 szerepel a nevezőben. □

A fenti két tételből következik, hogy a hálózati feladatnak minden bázismegoldása egész, tehát a szimplex módszer egész megoldásokon lépked. A következőkben ennél többet is belátunk: a szimplex módszer során egyáltalán nem kell szorzásokat és osztásokat végezni, csak összeadásokat és kivonásokat.

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

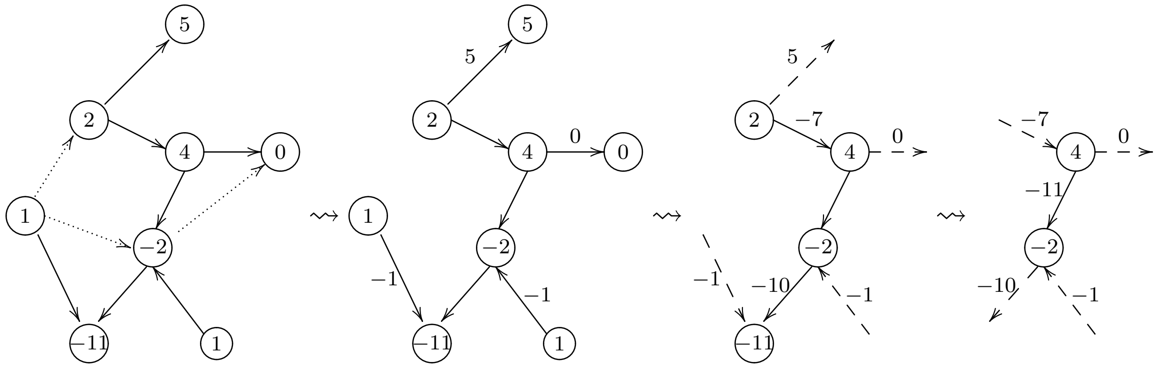
$$\begin{array}{ccc} b_{v_i} & \sim & b_i \\ c_{e_j} & \sim & c_j \\ y_{v_i} & \sim & y_i \\ x_{v_i} & \sim & x_i \end{array}$$

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

12.2. 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 minden 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. □

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 $\bar{c} \geq 0$.

12.1. 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.

   12.3. 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

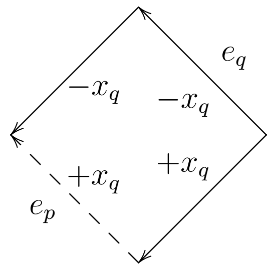
   $$c\bar{x}' = c\bar{x} + \delta \sum_{uv \in C} c_{uv} = c\bar{x} - \delta \sum_{uv \in C} \bar{c}_{uv} = c\bar{x} - \delta \bar{c}_p \xrightarrow[\delta \to \infty]{} +\infty$$

   □

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.

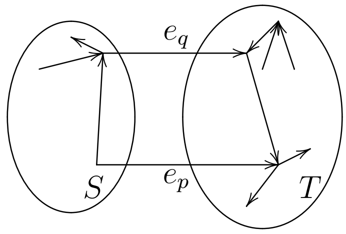
4. Vegyük hozzá a B bázishoz az e_p élt, és hagyjuk el a bázisból az e_q élt: $B' = B + \{p\} - \{q\}$.

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



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

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

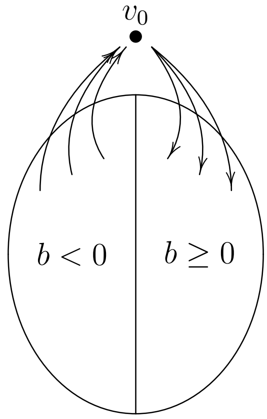
A fenti megkülönböztetés azért szükséges, hogy $y_0 = 0$ maradjon.

12.3. 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.

12.2. Kétfázisú hálózati szimplex módszer

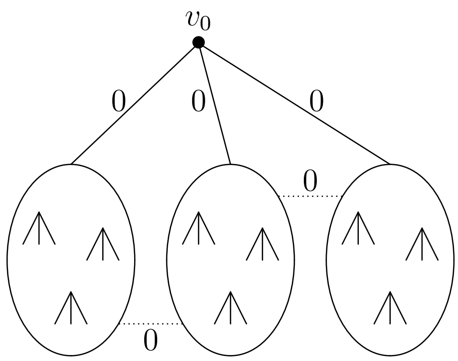
Vegyük hozzá az eredeti gráfhoz a következő új éleket: $E' = E \cup \{v_0 v : b_v \geq 0\} \cup \{v v_0 : v_v < 0\}$. Legyen

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



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.

12.4. Megjegyzés. A hálózati szimplex módszer kis módosítással használható arra az általánosabb feladatra is, ahol minden élre adott egy kapacitás-korlát. Ehhez minden élet egy 3 hosszú úttal kell helyettesíteni, aminek a középső éle fordított irányú. Az első új csúcsra b_v legyen a kapacitás, a második új csúcsra pedig a kapacitás -1-szerese.

12.3. 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.

12.2. 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.

12.4. 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. □

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.

12.5. 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. □

Most belátjuk, hogy ezzel a választással nem lehet ciklizálás.

12.6. 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. □

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.

13. Hálózati folyamok alkalmazásai

13.1. Szállítási feladat

Egy cégnek van m raktára és n üzlete; egyetlen termék szállítását kell megszerveznünk a raktárakból az üzletekbe. Az i-edik raktárban p_i teherautónyi áru van, míg a j-edik üzletben q_j teherautónyi árura van szükség. Egy teherautó útja az i-edik raktárból a j-edik üzletbe c_{ij}-be kerül. Határozzuk meg a minimális összköltségű szállítási beosztást.

A feladatot hálózati folyamként modellezzük; egészen pontosan minimális költségű maximális folyam feladatként. Definiálunk egy $D = (V, E)$ irányított gráfot, ahol V-ben $m + n + 2$ csúcs van: s, t, u_i ($i \in [m]$), v_j ($j \in [n]$). s-ből minden u_i-be megy egy él p_i kapacitással és 0 költséggel. u_i és v_j között $\min\{p_i, q_j\}$ kapacitású és c_{ij} költségű él megy. Végül minden v_j csúcsból t-be egy q_j kapacitású és 0 költségű élt húzunk.

Ha s-ből t-be a maximális folyam nagysága szigorúan kisebb, mint $\sum_{j=1}^{n} q_j$, akkor nincs jó szállítás, hiszen nem tudjuk az összes üzlet szükségletét kielégíteni. Ha a maximális folyam nagysága $\sum_{j=1}^{n} q_j$ (ennél nagyobb nyilván nem lehet, hiszen a t-be belépő élek összkapacitása ennyi), akkor az egészértékű maximális folyamok és a megengedett szállítási beosztások között bijekció van, hiszen egy folyam $u_i v_j$ éleken felvett értékei pont ugyanazokat a feltételeket teljesítik, mint amiket egy megengedett szállítási beosztásnak kell. Ráadásul ez a bijekció megtartja a költséget, így a minimális költségű maximális folyamok megfelelnek a minimális költségű szállítási beosztásoknak.

13.2. Megszakításos ütemezés párhuzamos gépeken

Adott m darab munka, és k darab gép. Az i-edik munkát p_i időbe kerül elvégezni, legkorábban az s_i időpontban kell elkezdeni, és legkésőbb a t_i időpontban kell befejezni.

Mindegyik munka bármelyik gépen végezhető, de egy gépen egyszerre csak egy. Az is megengedett, hogy egy munka egy részét elvégezzük az egyik gépen, aztán később egy másik gépen folytatjuk. Viszont egy munkát nem végezhetünk egyszerre több gépen.

A kérdés az, hogy el tudjuk-e végezni az összes munkát a megadott időintervallumokban? Lássuk, hogy lehet ezt a feladatot hálózati folyam feladatként megoldani.

Vegyük az összes munka kezdési és befejezési időpontjait, azaz az összes s_i és t_i számot, és rendezzük őket nagyság szerint sorba. Legyenek a sorba rendezett (különböző) időpontok $\tau_1 < \tau_2 < \cdots < \tau_n$ és legyen $I_j = [\tau_j, \tau_{j+1}]$ a τ_j-től τ_{j+1}-ig tartó időintervallum ($j = 1, \ldots, n-1$).

A feladathoz konstruálunk egy $D = (V, E)$ irányított gráfot. Minden munkához tartozik egy u_i csúcs ($i = 1, \ldots, m$), és minden I_j időintervallumhoz tartozik egy v_j csúcs ($j = 1, \ldots, n-1$). Ezen kívül lesz még egy w csúcs.

$u_i v_j$ akkor éle az irányított gráfnak, ha $s_i \le \tau_j$ és $t_i \ge \tau_{j+1}$, azaz az i-edik munkát végezhetjük az I_j időintervallumban. Ilyenkor az $u_i v_j$ él kapacitása $g(u_i v_j) = \tau_{j+1} - \tau_j$. Ezen kívül minden $1 \le j \le n-1$-re $w v_j$ is él, $(\tau_{j+1} - \tau_j)k$ kapacitással.

Az u_i csúcs igénye $b(u_i) = -p_i$, a v_j csúcs igénye pedig $b(v_j) = (\tau_{j+1} - \tau_j)k$. Tekintsük az így kapott hálózati feladatot:

$$
\begin{aligned}
& x \in \mathbb{R}^E \\
0 \le x(e) &\le g(e) && \forall e \in E \\
\varrho_x(v) - \delta_x(v) &= b(v) && \forall v \in V.
\end{aligned}
$$

13.1. Állítás. *Az ütemezési feladatnak akkor és csak akkor van megoldása, ha a fenti hálózati feladat megoldható.*

Bizonyítás. Ha az ütemezési feladatnak van egy megoldása, legyen $x(u_i v_j)$ annyi, amennyit az i-edik munkán dolgozunk az I_j időintervallumban, és $x(w v_j)$ legyen annyi, amennyit az I_j

időintervallumban a gépek összesen állnak. Könnyű ellenőrizni, hogy ez az x kielégíti a hálózati feladat feltételeit.

A fordított irányhoz tegyük fel, hogy a hálózati feladatnak van egy x megoldása; ebből kellene legyártani az ütemezési feladat egy megoldását. Mindenesetre $x(u_iv_j)$ megadja, hogy az i-edik munkából mennyit végzünk az I_j időintervallumban; és a hálózati feladat feltételei azt is garantálják, hogy legfeljebb $(\tau_{j+1} - \tau_j)k$ idejű munka jut erre az intervallumba. Az a kérdés, hogy ezeket a munkákat hogyan osszuk el a gépek között.

A beosztást mohó módon csinálhatjuk meg: először az első gépre rakjuk sorra a munkákat (index szerint növekvő sorrendben), amikor ez betelik ($(\tau_{j+1} - \tau_j)$-nyit rátettünk), elkezdünk a második gépre pakolni, stb. Tehát ha pl. az ötödik munkánál telt be az első gép, és már nem tudtuk a teljes $x(u_5v_j)$-nyit rápakolni, akkor a maradékot a második gép elejére pakoljuk.

Ily módon garantált, hogy egy munkát nem végzünk egyszerre több gépen, hiszen a fenti példánál maradva az ötödik munkát az első gépen az I_j időintervallum végén végezzük, míg a második gépen az I_j időintervallum elején, és $x(u_5v_j) \leq \tau_{j+1} - \tau_j$ miatt ezek nem fedhetik át egymást.

Az így kapott beosztás tehát az ütemezési feladat egy jó megoldása. □

13.3. Utaztatási feladat

Egy hajótársaság n város érintésével tervez hajóutat. Az i-edik városból a j-edik városba d_{ij} ember szeretne utazni, a jegy ára pedig c_{ij}. A hajón k darab férőhely van. A hajótársaság meg akarja határozni minden $1 \leq i < j \leq n$-re, hogy mennyi i-ből j-be szóló jegyet adjon el, hogy a jegybevétele maximális legyen. Természetesen legfeljebb d_{ij} darab i-ből j-be szóló jegy adható el, és semelyik útszakaszon nem utazhat a hajón k-nál több ember.

A megfelelő hálózati folyam feladathoz definiálunk egy $D = (V, E)$ irányított gráfot, ahol $V = \{v_1, \dots, v_n\}$. A csúcsok igénye: $b(v_1) = -k$, $b(v_j) = 0$ $(j = 2, \dots, n-1)$, $b(v_n) = k$. Kétfajta él van: egyrészt minden $1 \leq j \leq n-1$-re egy v_jv_{j+1} él, aminek súlya $w(v_jv_{j+1}) = 0$ és kapacitása $g(v_jv_{j+1}) = \infty$; másrészt minden $1 \leq i < j \leq n$-re egy v_iv_j él, aminek súlya $w(v_iv_j) = c_{ij}$ és kapacitása $g(v_iv_j) = d_{ij}$. A következő hálózati folyam feladatot oldjuk meg:

$$\begin{aligned} &\max wx \\ &x \in \mathbb{R}^E \\ 0 \leq x(e) &\leq g(e) && \forall e \in E \\ \varrho_x(v) - \delta_x(v) &= b(v) && \forall v \in V. \end{aligned}$$

13.2. Állítás. *Az utaztatási feladat minden megoldásának megfelel a fenti hálózati feladat egy ugyanolyan súlyú egész megoldása, és fordítva.*

Bizonyítás. Ha adott az utaztatási feladat egy megoldása, tekinthetjük a hálózati feladat azon megoldását, ahol az első típusú v_jv_{j+1} éleken $x(v_jv_{j+1})$ egyenlő a j-edik városból a $(j+1)$-edik városba tartó útszakaszon üresen maradt helyek számával, a második típusú v_iv_j éleken pedig $x(v_iv_j)$ egyenlő az eladott i-edik városból j-edik városba szóló jegyek számával. Könnyen ellenőrizhető, hogy x a hálózati folyam feladat megengedett megoldása, és wx pont a jegybevétellel egyenlő.

A másik irányhoz tekintsük a hálózati feladat egy x megoldását. Ez egy k nagyságú egészértékű folyam v_1-ből v_n-be, tehát felbontható k darab v_1-ből v_n-be menő útra; legyenek ezek $P_1, \dots, P_k$. Ebből megkaphatjuk az utaztatási feladat egy megoldását, ha úgy tekintjük, hogy "helyre szóló" jegyeket adunk el: a hajón az l-edik helyre azokra az ij útszakaszokra adunk el jegyet, amiknek megfelelő második típusú v_iv_j él szerepel a P_l úton. Ha a P_l úton szerepel egy első típusú v_jv_{j+1} él, az azt jelenti, hogy a j-edik és $(j+1)$-edik város közt az l-edik hely üresen marad.

Ennél az utaztatásnál a jegybevétel a wx értékkel lesz egyenlő, hiszen pont azokat a jegyeket adjuk el, amiknek megfelelő élek szerepelnek valamelyik P_l úton. □

13.4. Raktár-bérlési feladat

Egy n periódusból álló időszakban kell raktározási kapacitást bérelnünk az igényeknek megfelelően. A j-edik periódusban d_j tárolókapacitásra van szükség. Tárolókapacitást azonban nem csak külön-külön az egyes periódusokra bérelhetünk, hanem hosszabb időszakokra is, és a költség függhet az időszak hosszától. Azaz minden $1 \leq i \leq j \leq n$ egész számpárra adott egy c_{ij} érték, ami egységnyi tárolókapacitás bérlésének költsége az $i, i+1, \dots, j$ periódusokra. A cél úgy bérelni tárolókapacitásokat, hogy minden periódusban elég raktár álljon rendelkezésre, és az összköltség minimális legyen.

A feladat hálózati folyammal történő megoldásához konstruálunk egy $D = (V, E)$ irányított gráfot, ahol $V = \{v_{1,2}, \dots, v_{n+1}\}$. E-ben kétféle él van: egyrészt $v_j v_{j+1}$ élek ($j = 1, \dots, n$), amiknek költsége $w(v_j v_{j+1}) = 0$, másrészt $v_{j+1} v_i$ élek ($1 \leq i \leq j \leq n$), amiknek költsége $w(v_{j+1} v_i) = c_{ij}$.

A csúcsok igényei: $b(v_1) = d_1$, $b(v_j) = d_j - d_{j-1}$ ($j = 2, \dots, n$), $b(v_{n+1}) = -d_n$. Így az igények összege 0. A következő hálózati feladatot oldjuk meg:

$$\begin{aligned} \min wx& \\ x \in \mathbb{R}^E& \\ x(e) \geq 0& \qquad\qquad \forall e \in E \\ \varrho_x(v) - \delta_x(v) = b(v)& \qquad\qquad \forall v \in V. \end{aligned}$$

13.3. Állítás. *A raktár-bérlési feladat minden megoldásának megfelel a hálózati feladat egy ugyanolyan költségű megoldása, és fordítva.*

Bizonyítás. Először tekinsük a bérlési feladat egy megoldását, és ezt alakítsuk át a hálózati feladat megoldásává. Legyen $x(v_{j+1} v_i)$ annyi, ahány egységnyit bérlünk az $i, i+1, \dots, j$ periódusokra vonatkozó tárolókapacitásból. Egy $v_j v_{j+1}$ élre pedig legyen $x(v_j v_{j+1})$ annyi, amennyivel több tárolókapacitásunk van a j-edik periódusban d_j-nél.

Ekkor wx pont a tárolókapacitás bérlésének a költsége (hiszen a $v_j v_{j+1}$ élek költsége 0), és ellenőrizhető, hogy x pontosan teljesíti a hálózati feladat igényeit.

A fordított irányhoz tekintsük a hálózati feladat egy x megoldását. Ennek megfeleltethetjük azt a raktár-bérlést, ahol az $i, i+1, \dots, j$ periódusokra $x(v_{j+1} v_i)$-nyi raktárat bérlünk. Azt kell belátni, hogy ez tényleg jó, tehát, a j-edik peridusban rendelkezésre áll d_j egységnyi tárolókapacitás. Nézzük ezt adott j-re. Az igényeket úgy határoztuk meg, hogy

$$\sum_{i=1}^{j} b(v_i) = d_j \quad \text{és} \quad \sum_{i=j+1}^{n+1} b(v_i) = -d_j,$$

Tehát a $V_j = \{v_1 \dots, v_j\}$ csúcshalmaz össz-igénye d_j, ami miatt a V_j-be belépő éleken az x összege legalább d_j. Ez pedig pont azt jelenti, hogy a j-edik peridusban rendelkezésre áll d_j-nyi tárolókapacitás. □