

Operációkutatás I.

2018/2019-2.

Szegedi Tudományegyetem
Informatikai Intézet
Számítógépes Optimalizálás Tanszék

6. Előadás

Áttekintés

Kezdjük újra a klasszikus **erőforrás allokációs problémával** (katonák, vonatok...)

Az eredeti probléma LP formában:

$$\begin{array}{rcll}
 \max z = 3x_1 & + & 2x_2 & \\
 \hline
 x_1 & + & x_2 & \leq 80 \quad [\text{fafaragás}] \\
 2x_1 & + & x_2 & \leq 100 \quad [\text{festés}] \\
 x_1 & & & \leq 40 \quad [\text{kereslet}] \\
 & x_1, & x_2 & \geq 0
 \end{array}$$

A kezdő szótár:

$$\begin{array}{rcllcl}
 x_3 & = & 80 & - & x_1 & - & x_2 \\
 x_4 & = & 100 & - & 2x_1 & - & x_2 \\
 x_5 & = & 40 & - & x_1 & & \\
 \hline
 z & = & 0 & + & 3x_1 & + & 2x_2
 \end{array}$$

Áttekintés

Optimális szótár:

$$\begin{array}{rclclcl}
 x_1 & = & 20 & + & x_3 & - & x_4 \\
 x_2 & = & 60 & - & 2x_3 & + & x_4 \\
 x_5 & = & 20 & - & x_3 & + & x_4 \\
 \hline
 z & = & 180 & - & x_3 & - & x_4
 \end{array}$$

Vegyünk hozzá egy új termék gyártását: **játékaútót is gyártunk**, 0.5 óra fafaragás, 1 óra festés, 1\$-os ár, a keresletkorátunk is változik $\rightarrow x_6$ a gyártandó játékaútok száma

Az új feladat:

$$\begin{array}{rclclcl}
 \max z = & 3x_1 & + & 2x_2 & + & x_6 \\
 x_1 & + & x_2 & + & 0.5x_6 & \leq & 80 \\
 2x_1 & + & x_2 & + & x_6 & \leq & 100 \\
 x_1 & & & + & x_6 & \leq & 40 \\
 & & & & x_1, x_2, x_6 & \geq & 0
 \end{array}$$

Áttekintés

Az új feladat kezdő szótára:

$$\begin{array}{rclclclcl}
 x_3 & = & 80 & - & x_1 & - & x_2 & - & 0.5x_6 \\
 x_4 & = & 100 & - & 2x_1 & - & x_2 & - & x_6 \\
 x_5 & = & 40 & - & x_1 & & & - & x_6 \\
 \hline
 z & = & 0 & + & 3x_1 & + & 2x_2 & + & x_6
 \end{array}$$

Az új feladat optimális szótára?

Rendezzük át a szótárunkat:

$$\begin{array}{rclclclcl}
 x_3 + 0.5x_6 & = & 80 & - & x_1 & - & x_2 \\
 x_4 + x_6 & = & 100 & - & 2x_1 & - & x_2 \\
 x_5 + x_6 & = & 40 & - & x_1 & & \\
 \hline
 z - x_6 & = & 0 & + & 3x_1 & + & 2x_2
 \end{array}$$

Legyen $x'_3 = x_3 + 0.5x_6$, $x'_4 = x_4 + x_6$, $x'_5 = x_5 + x_6$, $z' = z - x_6$

Áttekintés

Az új feladat egy lehetséges módosított kezdő szótára:

$$\begin{array}{rclclcl}
 x'_3 & = & 80 & - & x_1 & - & x_2 \\
 x'_4 & = & 100 & - & 2x_1 & - & x_2 \\
 x'_5 & = & 40 & - & x_1 & & \\
 \hline
 z' & = & 0 & + & 3x_1 & + & 2x_2
 \end{array}$$

Ebből az optimális szótár (mint előbb):

$$\begin{array}{rclclcl}
 x_1 & = & 20 & + & x'_3 & - & x'_4 \\
 x_2 & = & 60 & - & 2x'_3 & + & x'_4 \\
 x'_5 & = & 20 & - & x'_3 & + & x'_4 \\
 \hline
 z' & = & 180 & - & x'_3 & - & x'_4
 \end{array}$$

Áttekintés

Helyettesítsünk vissza:

$$\begin{array}{rclclcl}
 x_1 & = & 20 & + & (x_3 + 0.5x_6) & - & (x_4 + x_6) \\
 x_2 & = & 60 & - & 2(x_3 + 0.5x_6) & + & (x_4 + x_6) \\
 x_5 + x_6 & = & 20 & - & (x_3 + 0.5x_6) & + & (x_4 + x_6) \\
 \hline
 z' & = & 180 & - & (x_3 + 0.5x_6) & - & (x_4 + x_6)
 \end{array}$$

amiből

$$\begin{array}{rclclcl}
 x_1 & = & 20 & + & x_3 & - & x_4 & - & 0.5x_6 \\
 x_2 & = & 60 & - & 2x_3 & + & x_4 & & \\
 x_5 & = & 20 & - & x_3 & + & x_4 & - & 0.5x_6 \\
 \hline
 z & = & 180 & - & x_3 & - & x_4 & - & 1.5x_6
 \end{array}$$

Azaz lényegében nem (éri meg) fogunk gyártani egyetlen kisautót sem ilyen feltételek mellett ($x_6 = 0$ az optimális megoldásban).

Áttekintés

Vegyünk egy újabb feltételt: csomagoljuk be a játékokat → 250 egység **karton** áll rendelkezésre, a katonához 3, a vonathoz 4, a kisautóhoz 1 egység kell.

Az új feladat:

$$\begin{array}{rclclcl}
 \text{Max } z = & 3x_1 & + & 2x_2 & + & x_6 & \\
 & x_1 & + & x_2 & + & 0.5x_6 & \leq 80 \\
 & 2x_1 & + & x_2 & + & x_6 & \leq 100 \\
 & x_1 & & & + & x_6 & \leq 40 \\
 & 3x_1 & + & 4x_2 & + & x_6 & \leq 250 \\
 & & & & x_1, & x_2, x_6 & \geq 0
 \end{array}$$

Áttekintés

Az új feladat kezdőszótára ($\rightarrow x_7$ új mesterséges változó)

$$\begin{array}{rclclclcl}
 x_3 & = & 80 & - & x_1 & - & x_2 & - & 0.5x_6 \\
 x_4 & = & 100 & - & 2x_1 & - & x_2 & - & x_6 \\
 x_5 & = & 40 & - & x_1 & & & - & x_6 \\
 x_7 & = & 250 & - & 3x_1 & - & 4x_2 & - & x_6 \\
 \hline
 z & = & 0 & + & 3x_1 & + & 2x_2 & + & x_6
 \end{array}$$

Tegyük x_7 -et bázisváltóvá $\rightarrow$ fejezzük ki x_7 -et az eredeti szótár bázisa segítségével (ld. 6. dia):

$$x_7 = 250 - 3x_1 - 4x_2 - x_6 =$$

$$\begin{aligned}
 250 - 3(20 + x_3 - x_4 - 0.5x_6) - 4(60 - 2x_3 + x_4) - x_6 = \\
 -50 + 5x_3 - x_4 + 0.5x_6
 \end{aligned}$$

Áttekintés

Az új szótár:

$$\begin{array}{rclclclcl}
 x_1 & = & 20 & + & x_3 & - & x_4 & - & 0.5x_6 \\
 x_2 & = & 60 & - & 2x_3 & + & x_4 & & \\
 x_5 & = & 20 & - & x_3 & + & x_4 & - & 0.5x_6 \\
 x_7 & = & -50 & + & 5x_3 & - & x_4 & + & 0.5x_6 \\
 \hline
 z & = & 180 & - & x_3 & - & x_4 & - & 0.5x_6
 \end{array}$$

Vegyük észre, hogyha a kapott szótár lehetséges lenne, akkor a megoldás optimális.

Viszont ez nem egy lehetséges kezdőszótár, mivel a bázismegoldásban $x_7 < 0 \rightarrow$ Mehet a **kétfázisú szimplex módszer**.

Geometriai alapfogalmak

- A lineáris programozás szoros kapcsolatban áll a **konvex geometriával**
- Az eddig **tárgyalt fogalmaknak**, mint a bázismegoldás, lineáris feltétel, lehetséges megoldások halmaza, stb., **mind megfeleltethető egy-egy geometriai objektum**

R^n : n -dimenziós **lineáris tér** a valós számok felett – Elemei az n elemű valós **vektorok**

E^n : n -dimenziós **euklideszi tér**, olyan lineáris tér, amelyben értelmezett egy **belső szorzat** és egy **távolság** függvény a következő módon

- $\langle x, y \rangle = x^T y = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$, $\|x\| = \sqrt{\langle x, x \rangle}$ norma
- $d(x, y) = \|x - y\|_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$

Geometriai alapfogalmak

Pont: egy $x \in E^n$ vektor

Lehetséges megoldások $\leftrightarrow$ Pontok az E^n térben

$x_1, x_2 \in E^n$ **különböző pontokat összekötő szakasz:**

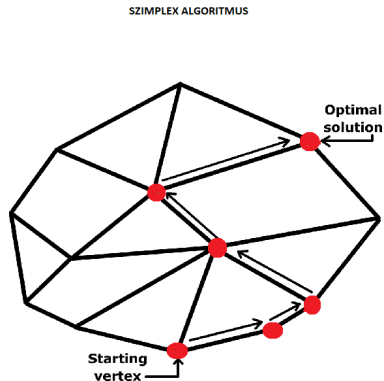
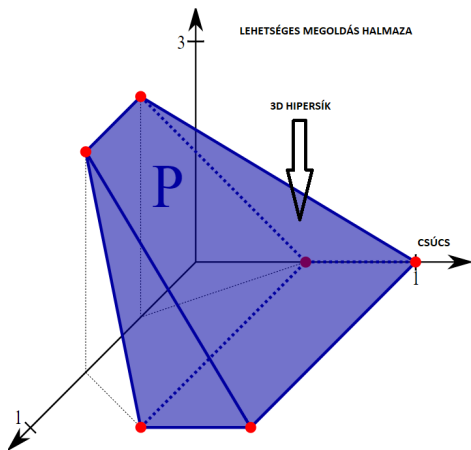
$$\{x : x \in E^n, x = \lambda x_1 + (1 - \lambda)x_2\},$$

ahol $\lambda \in [0,1]$ tetszőleges

x_1, x_2 végpontú **szakasz felezőpontja:** $\frac{1}{2}x_1 + \frac{1}{2}x_2$ pont

Ponthalmaz **csúcspontja:** olyan pont, amely nem áll elő egyetlen ponthalmazbeli szakasz felezőpontjaként sem

Geometriai alapfogalmak: példa



Geometriai alapfogalmak

n -dimenziós sík:

$$\{ x : x \in E^n, a_1x_1 + a_2x_2 + \dots + a_nx_n = b \},$$

ahol $a_1, a_2, \dots, a_n, b \in \mathbb{R}$ rögzített számok

n -dimenziós zárt féltér:

$$\{ x : x \in E^n, a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b \},$$

ahol $a_1, a_2, \dots, a_n, b \in \mathbb{R}$ rögzített számok

Lineáris feltételek $\leftrightarrow$ Zárt félterek ($\leq$) és síkok ($=$)

Lehetséges megoldások halmaza $\leftrightarrow$ Zárt félterek (és síkok) metszete

Konvex poliéderek

Konvex ponthalmaz: olyan ponthalmaz, amely tartalmazza bármely két pontját összekötő szakasz pontjait is

Zárt ponthalmaz: olyan ponthalmaz, amely tartalmazza a pontjaiból képezhető tetszőleges konvergens sorozat határértékét is

Korlátos ponthalmaz: olyan ponthalmaz, amelynek minden x pontjára teljesül, hogy $d(0, x) \leq K$, ahol K egy rögzített valós szám (inkább használatos: komplementere nyílt)

Poliéder: zárt, véges sok csúcsponttal rendelkező ponthalmaz

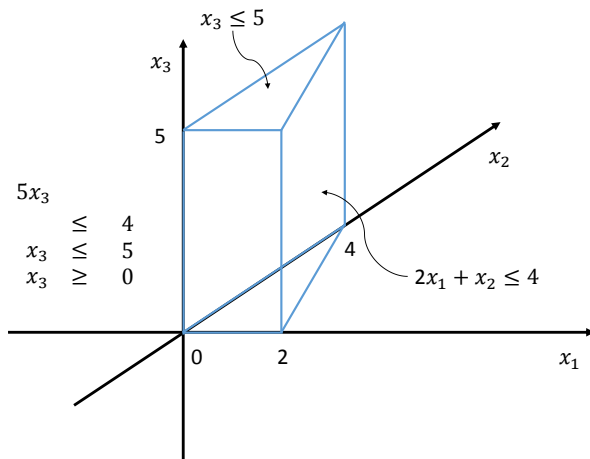
A lehetséges megoldások halmaza egy konvex poliéder

Politóp: korlátos poliéder

Konvex poliéderek

Példa.

$$\begin{array}{rcll}
 \max & 3x_1 & + & 2x_2 & + & 5x_3 \\
 & 2x_1 & + & x_2 & & & \leq & 4 \\
 & & & & & x_3 & \leq & 5 \\
 & x_1 & , & x_2 & , & x_3 & \geq & 0
 \end{array}$$



LP és Konvex poliéderek

- Feltételek (egyenletek) $\leftrightarrow$ Poliéder lapok
- Egy **lineáris célfüggvény a poliéder valamelyik csúcsában veszi fel a maximumát**
 - Belső pontban van olyan irány amerre elmozdulva nő a célfüggvény
 - Élre eső pont esetében mindkét csúcs felé mozdulva nem csökkenhet a célfüggvény
- Egy $n + m$ elemű bázismegoldásban n változó értékét (nembázis változók) 0-ra állítjuk; ez meghatározza a többi értékét is
- Az n dimenziós térben n hipersík metszete meghatároz egy pontot
- **Bázismegoldások $\leftrightarrow$ Poliéder csúcsok**

Szimplex algoritmus és geometria

$$\begin{array}{rcll}
 \max & 3x_1 & + & 2x_2 & + & 5x_3 & & \\
 & 2x_1 & + & x_2 & & & \leq & 4 \\
 & & & & & x_3 & \leq & 5 \\
 & x_1 & , & x_2 & , & x_3 & \geq & 0
 \end{array}$$

A példán végrehajtva a szimplex algoritmust a klasszikus pivot stratégiával

$$[0 \ 0 \ 0 \ 4 \ 5] \rightarrow [0 \ 0 \ 5 \ 4 \ 0] \rightarrow [2 \ 0 \ 5 \ 0 \ 0] \rightarrow [0 \ 4 \ 5 \ 0 \ 0]$$

bázismegoldások adódnak (*hf: ellenőrizzük*)

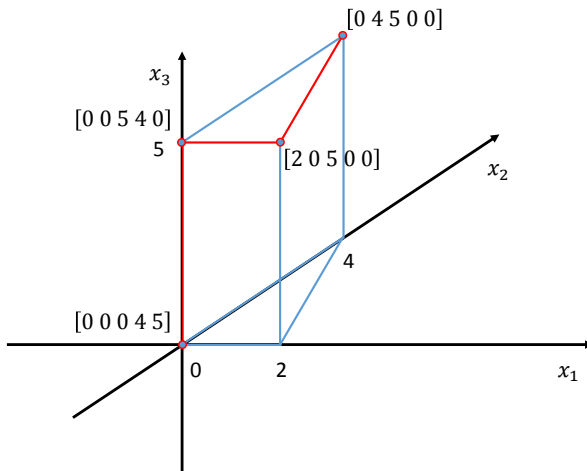
A **bázismegoldások** rendre a

$$[0 \ 0 \ 0] \rightarrow [0 \ 0 \ 5] \rightarrow [2 \ 0 \ 5] \rightarrow [0 \ 4 \ 5]$$

poliéder csúcsoknak felelnek meg

Egy pivot lépés az aktuális bázismegoldásból induló valamelyik él mentén történő ellépés egy „szomszédos” bázismegoldásba (azaz csúcsba)

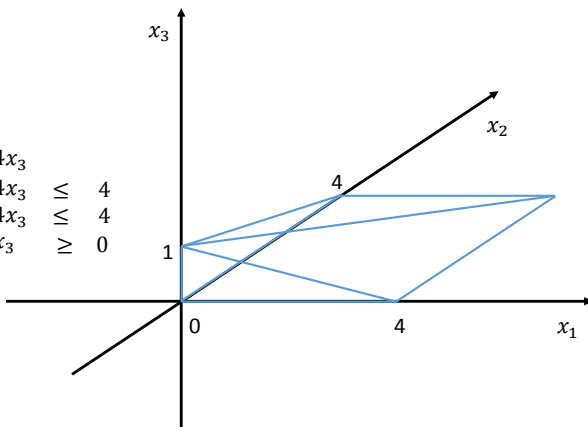
Szimplex algoritmus



Ciklizáció és geometria

Tekintsük a következő feladatot:

$$\begin{array}{llllll}
 \max & x_1 & + & x_2 & + & 4x_3 \\
 & x_1 & & & + & 4x_3 & \leq & 4 \\
 & & & x_2 & + & 4x_3 & \leq & 4 \\
 & x_1 & , & x_2 & , & x_3 & \geq & 0
 \end{array}$$



Ciklizáció

A klasszikus pivot stratégiával a

$$\begin{bmatrix} 0 & 0 & 0 & 4 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 4 & 0 & 0 & 0 \end{bmatrix}$$

bázismegoldások adódnak

A bázismegoldások rendre a

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 4 & 0 \end{bmatrix}$$

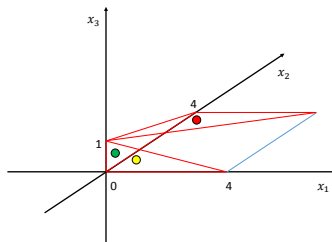
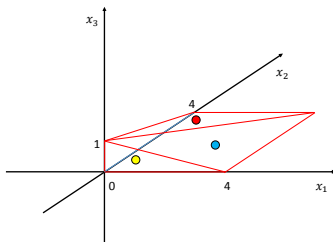
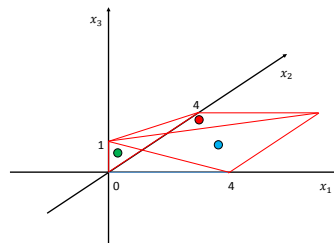
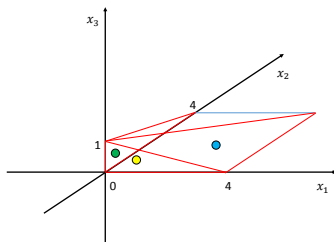
poliéder csúcsoknak felelnek meg

Degeneráció lépett fel

Degeneráció esetén egy n -dimenziós csúcspont legalább $n + 1$ síkra esik

Degenerált iterációs lépéskor a „csúcsban maradunk”

Ciklizáció



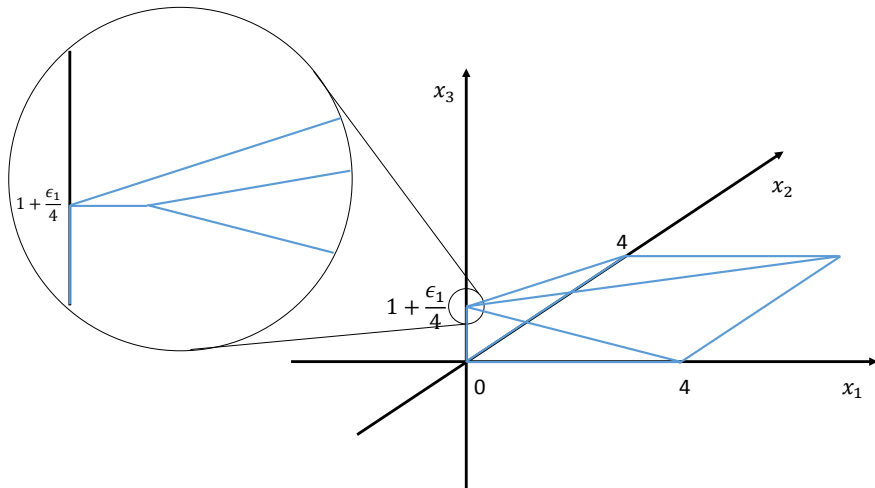
Ciklizáció

- Ciklizáció esetén egy csúcspontban ragadunk
- Csak a **csúcspont leírására használt síkokat cserélgetjük**
- A **perturbáció módszere** szétválasztja a degenerációt okozó síkokat egymástól

$$\begin{array}{rclcl}
 x_1 & & + & 4x_3 & \leq & 4 + \epsilon_1 \\
 & & & x_2 & + & 4x_3 & \leq & 4 + \epsilon_2 \\
 x_1 & , & x_2 & , & x_3 & \geq & 0 \\
 \hline
 \max & x_1 & + & x_2 & + & 4x_3
 \end{array}$$

- A degeneráció megszüntetésével elkerüljük a ciklizációt

Ciklizáció



Többcélú programozás

Tekintsük újra a gyártási alapeladatunkat (katonák, vonatok,...):

- eladási árak: 2 illetve 3 \$
- fafaragás és festés úgy mint eddig
- + katona 3, vonat 4 egység **csomagolópapír**

Tegyük fel, hogy a cég **non-profit**á válik és **néhány új célt** állít maga elé:

- **Cél #1:** **Legfeljebb 40 katonát** gyártanak (csak ennyi adható el 3 \$-ért)
- **Cél #2:** **Legalább 140\$ profit** szerzése, hogy fizetni tudják a dolgozókat
- **Cél #3:** **Legfeljebb 130 egység csomagolópapír** használata (jelenleg ennyi van csak / fontos a környezetvédelem)

Többcélú programozás

Minden cél egyedileg elérhető, de **a három egyszerre nem biztos, hogy teljesíthető!**

Vezessünk be ún. **büntetést**, arányosan azzal, hogy **mennyire térünk el az egyes céloktól** egy-egy megoldás esetén:

- **Büntetés #1:** 40-nél több katonát nem adunk el; a büntetés 3\$ (hiányzó profit a legyártott, de már eladhatatlan katonánként)
- **Büntetés #2:** Ha nincs legalább 140\$ profit, a dolgozókat hitelből kell fedezni (20% kamat)
- **Büntetés #3:** Ha több csomagolópapír kell, akkor 0.80\$-ért tudjuk beszerezni egységét

Többcélú programozás

A kombinált **cél**, hogy **minimalizáljuk** az összes felmerülő **büntetést**.
Ez a következő feladathoz vezet:

$$\begin{array}{rclcl}
 x_1 + x_2 & \leq & 80 & & \\
 2x_1 + x_2 & \leq & 100 & & \\
 x_1 & \leq & 40 & [\text{Cél \#1}] & \\
 3x_1 + 2x_2 & \geq & 140 & [\text{Cél \#2}] & \\
 3x_1 + 4x_2 & \leq & 130 & [\text{Cél \#3}] & \\
 & \xrightarrow[\text{slack}]{} & & &
 \end{array}
 \qquad
 \begin{array}{rclcl}
 x_1 + x_2 + x_3 & = & 80 & & \\
 2x_1 + x_2 & + & x_4 & = & 100 \\
 x_1 & + & x_5 & = & 40 \\
 3x_1 + 2x_2 & - & x_6 & = & 140 \\
 3x_1 + 4x_2 & + & x_7 & = & 130
 \end{array}$$

A mesterséges változókra:

- x_3, x_4 nemnegatívok – **szigorú feltételek**, mindig fenn kell, hogy álljanak
- x_5, x_6, x_7 nem korlátozottak – **gyenge feltételek**: megszeghetjük, de akkor büntetést kell fizetni
- pl. ha x_5 pozitív $\Rightarrow$ a büntetés 0; ha x_5 negatív $\Rightarrow$ a büntetés $-3x_5$
- pl. ha x_6 pozitív $\Rightarrow$ a büntetés 0, mert 140\$ felett vagyunk a profitban, ...

Többcélú programozás

A kérdés, **hogyan formalizáljuk** ezt lineáris programként?

⇒ Fejezzük ki x_5, x_6, x_7 **két nemnegatív változó különbségeként** ⇒ pozitív és negatív részei a változóknak

- $x_5 = x_5^+ - x_5^-$
- $x_6 = x_6^+ - x_6^-$
- $x_7 = x_7^+ - x_7^-$

ahol

$$x^+ = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \text{ és } x^- = \begin{cases} x, & x \leq 0 \\ 0, & x > 0 \end{cases}$$

a szám pozitív, illetve negatív része, $x^+, x^- \geq 0$

⇒ Új célfüggvény, ami kifejezi az összes büntetés összegét, mely x_5, x_6, x_7 előjelei függvényében adódik

Többcélú programozás

A feladat így:

$$\begin{array}{llllll}
 x_5 = x_5^+ - x_5^- & \min & 3x_5^- & + 1.2x_6^- & + 0.8x_7^- & \\
 x_6 = x_6^+ - x_6^- & & x_1 + x_2 + x_3 & & & = 80 \\
 x_7 = x_7^+ - x_7^- & & 2x_1 + x_2 & + x_4 & & = 100 \\
 & & x_1 & + x_5^+ - x_5^- & & = 40 \\
 x_5^+, x_5^-, x_6^+, x_6^-, x_7^+, x_7^- \geq 0 & & 3x_1 + 2x_2 & - x_6^+ + x_6^- & & = 140 \\
 & & 3x_1 + 4x_2 & & + x_7^+ - x_7^- & = 130
 \end{array}$$

Ez megoldható a standard módon (szimplex algoritmus), illetve megoldható a duálisa (ld. következő előadás; szorgalmi feladat), és abból megkonstruálható a megoldása ennek (ld. dualitás tétel).

Egy dinamikus probléma

- SailCo vitorláshajókat gyártó cég a következő évét tervezi.
- **Cél:** Hány hajót gyártson az egyes **negyedévekben** ($t = 1, 2, 3, 4$)?
- **Kereslet:** 40, 60, 75, 25 a négy negyedévre.
- Az első negyedév kezdetén **10 hajó van raktáron**.
- Minden negyedév kezdetén meg kell határozni, hány hajót gyártsanak abban a periódusban.
- **Normál munkaidőben 40 hajót** tudnak gyártani **400\$/hajó** költségen egy negyedévben.
- **Túlóra esetén 450\$/hajó** költséggel kell számolni. (így persze tudnak 40-nél többet gyártani)
- **Tárolási költség (raktár) 20\$/hajó** minden negyedévben.

Minimalizáljuk a gyártási és tárolási költséget egy optimális ütemezéssel!

Egy dinamikus probléma – megoldás

Minden negyedévre meg kell határozni, hogy hány hajót gyártsanak normál munkaidőben és túlórában.

Döntési változók:

- x_t : hány hajót gyártsanak t negyedévben **rendes munkaidőben** (400\$/hajó), $t = 1, 2, 3, 4$
- y_t : hány hajót gyártsanak t negyedévben **túlórában** (450\$/hajó), $t = 1, 2, 3, 4$

Tárolás:

- i_t : a t negyedév végén **tárolás alatt** lévő hajók száma

A teljes költség:

- Teljes költség = rendes munkaidőben gyártott hajók költsége + túlórában gyártott hajók költsége + tárolási költség
$$= 400(x_1 + x_2 + x_3 + x_4) + 450(y_1 + y_2 + y_3 + y_4) + 20(i_1 + i_2 + i_3 + i_4)$$

Egy dinamikus probléma – megoldás

Észrevételek:

- t negyedév végén a tárolt hajók száma = tárolt hajók száma $(t - 1)$ -ben + legyártott hajók száma t -ben – kereslet t -ben
- ... azaz, ha d_t a kereslet t -ben, akkor

$$i_t = i_{t-1} + (x_t + y_t) - d_t \text{ ahol } i_0 = 10$$

- Az elérhető (gyártott és tárolt) hajók száma és a kereslet között:

$$i_{t-1} + (x_t + y_t) \geq d_t \text{ azaz } i_t = i_{t-1} + (x_t + y_t) - d_t \geq 0$$

Egy dinamikus probléma – megoldás

A lineáris program:

$$\min z = 400(x_1 + x_2 + x_3 + x_4) + 450(y_1 + y_2 + y_3 + y_4) + 20(i_1 + i_2 + i_3 + i_4)$$

$$x_j \leq 40 \quad (j = 1, 2, 3, 4)$$

$$i_1 = 10 + (x_1 + y_1) - 40$$

$$i_2 = i_1 + (x_2 + y_2) - 60$$

$$i_3 = i_2 + (x_3 + y_3) - 75$$

$$i_4 = i_3 + (x_4 + y_4) - 25$$

$$i_t \geq 0, y_t \geq 0 \text{ és } x_t \geq 0 \quad (t = 1, 2, 3, 4)$$

Az optimális megoldás:

$$z = 78,450\$$$

$$x_1 = x_2 = x_3 = 40, x_4 = 25$$

$$y_1 = 0, y_2 = 10, y_3 = 35, y_4 = 0$$

$$i_1 = 10, i_2 = i_3 = i_4 = 0.$$

Egy dinamikus probléma – megoldás

Néhány megjegyzés

- ① Érdekes megfigyelni, hogy az LP formalizmus enged túlórában gyártani akkor is, ha a rendes munkaidejű termelés nem éri el a 40-et
→ érdemes lehet negyedévről negyedévre menően implementálni („*rolling planning horizon*”)
- ② A költség sokszor lineáris függvénye a legyártott mennyiségnek
- ③ A jövőbeli keresletet nem biztos, hogy ilyen pontosan ismerjük (ld. érzékenységvizsgálat)
- ④ „Büntetést” fizethetünk, ha nem készülünk el időre egy-egy hajóval (ld. többcélú programozás)
- ⑤ A termelés növelése extra költségekkel is járhat (pl. új munkások betanítása)
- ⑥ ... azaz a **valóságban általában nem egyszerű modellezni!**