Glevitzky Béla

# Operációkutatás I.

mobiDIÁK könyvtár

Glevitzky Béla

# Operációkutatás I.

mobiDIÁK könyvtár

SOROZATSZERKESZTŐ

Fazekas István

Glevitzky Béla

# Operációkutatás I.

**mobiDIÁK könyvtár**
Debreceni Egyetem Informatikai Intézet

Debreceni Egyetem
Informatikai Intézet
4010 Debrecen, Pf. 12
http://mobidiak.unideb.hu

# Tartalomjegyzék

# Bevezetés

Minden tudatos emberi tevékenység lényegében a következő részekből tevődik össze: tervezés (döntéselőkészítés), döntés, megvalósítás, ellenőrzés. A tervezés (döntéselőkészítés) során olyan tervvariánsokat dolgoznak ki, amelyek megkönnyítik a döntéshozatalt. Az élet különböző területein, különösen pedig a gazdasági tevékenységeknél olyan döntések meghozatalára törekednek, amelyek valamilyen szempontból optimálisak. (Optimálisnak nevezzük például azt a döntést, amely lehetővé teszi, hogy a kívánt célt, vagy célokat a legkisebb ráfordítással, vagy pedig a legnagyobb haszonnal érjük el.) Bonyolult problémák esetében optimális döntést azonban csak akkor tudnak megvalósítani, ha a tervvariánsok kidolgozásakor tudományos módszereket használnak.

Az operációkutatás az a tudomány, amely az optimális döntések előkészítésében matematikai módszereket használ fel. Kialakulását a II. világháborútól számíthatjuk, amikor harcászati jellegű problémák megoldására használták ezeket a módszereket. A világháborút követő időszakban aztán egyre inkább előtérbe került az operációkutatás gazdasági alkalmazása. Ma már az operációkutatást egyre inkább felhasználják mind a modern ipargazdaságtanban (készletgazdálkodási, sorbanállási problémák), mind pedig a konkrét vállalati gyakorlatban.

Meg kell jegyeznünk, hogy az operációkutatás csak a döntéselőkészítés eszköze, nem egyenlő magával a döntéssel, így az embert nem iktathatjuk ki a döntési folyamatból. A legtökéletesebb operációkutatási módszer sem elegendő egymagában valamely döntési probléma megoldására, hiszen a figyelembe vett tényezőkön kívül sok egyéb, általában nem számszerűsíthető tényező is hat a döntési folyamatra. Ha pedig már megtaláltuk a „legjobb döntési variánst”, annak gyakorlati megvalósítása során felléphetnek olyan problémák, amelyek miatt a várt eredmény nem realizálható.

Az operációkutatási módszerek egyik csoportjába tartoznak azok, amelyek széleskörűen alkalmazhatók különböző, egymástól lényegesen eltérő, de bizonyos követelményeknek eleget tevő döntési és ellenőrzési problématipusok matematikai közgazdasági, statisztikai leírására, modellezések elemzésére (matematikai programozás, hálózati folyamok, digitális szimuláció). A másik csoportot azok a módszerek alkotják, amelyek speciális problémákból fejlődtek ki, adott tipusú problémák vizsgálatára alkalmasak (sorbanállási és készletgazdálkodási problémák).

Az operációkutatást az oktatási félévek számának megfelelően – önkényesen – két részre bontjuk. A jegyzetek is ennek megfelelően készültek el. Az „Operációkutatás I" jegyzet alapvetően a matematikai programozással foglalkozik, míg az „Operációkutatás II" jegyzet a hálótervezést, a készletgazdálkodást és a sorbanállási problémákat tartalmazza.

# Matematikai programozás

## Bevezetés

Az operációkutatás egyik legfontosabb része a matematikai programozás, amelynek alapproblémája:

$$\text{az} \qquad f_i(\underline{x}) \le b_i, \qquad i = 1, 2, \ldots, m, \ \underline{x} \in \mathbb{R}^n$$

($\mathbb{R}^n$ az n-dimenziós euklideszi tér)

feltételek mellett meghatározandó egy $f(\underline{x})$ függvény maximuma vagy minimuma és legalább egy szélsőértékhelye. (Az $f_i(\underline{x})$ függvényeket feltételi függvényeknek, az $f(\underline{x})$ -et pedig célfüggvénynek nevezzük.) A feltételi függvényekre és a célfüggvényre bizonyos megszorításokat téve más-más matematikai programozási problémához jutunk.

Ha a feltételi függvények és a célfüggvény lineárisak, akkor lineáris programozási problémáról beszélünk. Amennyiben a feltételi függvények illetve a célfüggvény között vannak nemlineárisak is, a problémát nemlineáris programozási problémának nevezzük. Ha a feltételi függvények és a célfüggvény konvexitását (konkávitását) követeljük meg, akkor az ún. konvex programozási problémákat nyerjük. Ha pedig a feltételi függvények és a célfügvény szeparábilisek, azaz felírhatók egyváltozós függvények összegeként, akkor a dinamikus programozási problémakörhöz jutunk.

A matematikai programozási problémákat osztályozhatjuk aszerint is, hogy az $\underline{x} \in \mathbb{R}^n$ komponensei folytonosan változhatnak, vagy pedig vannak köztük olyanok is, amelyek csak legfeljebb megszámlálható végtelen sok értéket vehetnek fel. Ennek alapján beszélhetünk folytonos, vegyes és tiszta diszkrét programozási problémákról.

A matematikai programozási problémákat végül osztályozhatjuk aszerint is, hogy a változók, a paraméterek között vannak-e valószínűségi változók, vagy sem. A valószínűségi változókat tartalmazó problémákat sztochasztikus programozási problémáknak, a valószínűségi változókat nem tartalmazókat pedig determinisztikus problémáknak nevezzük.

Az Operációkutatás I. jegyzetben először a lineáris programozást tárgyaljuk. Ezt követően sort kerítünk néhány nemlineáris programozási probléma bemutatására. Részletesen tárgyaljuk a diszkrét programozást, majd röviden ismertetjük a dinamikus és a sztochasztikus problémákat. Végül bemutatunk néhány számítógépes programcsomagot is, amelyeket fel lehet használni matematikai programozási feladatok számítógépes megoldására.

# I. Lineáris programozás

## I.1. Bevezetés

Mint ahogyan azt már említettük, ha a matematikai programozási problémában a feltételi függvények és a célfüggvény lineárisak, akkor lineáris programozási problémáról beszélünk, amelynek matematikai modellje: az

$$a_{i1}x_1 + \ldots + a_{in}x_n \le b_i, \qquad i = 1, 2, \ldots, m$$

feltételek mellett meghatározandó a

$$c_1x_1 + \ldots + c_nx_n$$

függvény minimuma, vagy maximuma és legalább egy szélsőértékhelye.

A gyakorlatban előforduló problémák többsége olyan, hogy az $\underline{x} \in \mathbb{R}^n$ vektor komponensei csak nemnegatívok lehetnek. Ezt külön is kihangsúlyozva, mátrixos alakban a következőképpen fogalmazhatjuk meg a problémát:

$$\begin{gathered} A\underline{x} \le \underline{b} \\ \underline{x} \ge \underline{0} \\ \text{max vagy } \min\{\underline{c}^T\underline{x}\}, \end{gathered}$$

ahol az $A$ egy $m \times n$-es mátrix, az $\underline{x}$, $\underline{c}$ $n$-dimenziós, a $\underline{b}$ $m$-dimenziós vektor. (Az $A$ mátrix, a $\underline{c}$ és a $\underline{b}$ vektorok adottak, az $\underline{x}$ vektor ismeretlen.) A lineáris programozás fejezetben először két olyan gyakorlati problémát említünk meg, amelyek – bizonyos megszorítások mellett – lineáris programozási problémaként kezelhetők. Ezután bemutatjuk a különböző lineáris programozási modelleket, valamint megmutatjuk, hogy ekvivalens átalakításokkal ezek bármelyikét átfogalmazhatjuk egy másik lináris programozási modellé. A következő részben röviden tárgyaljuk a konvex poliéderek elméletét, amely a lineáris modellek megoldási algoritmusának, a szimplex módszernek az elméleti hátterét jelenti. (A részletes tárgyalásra a függelékben kerül sor.) Ezt követi a lineáris programozási problémák megoldási módszere, a szimplex módszer és a vele kapcsolatos problémák (a módszer beindítása, végtelen ciklus elkerülése) tárgyalása, majd a szimplex módszer néhány variánsát és ezek egyikének alkalmazásaként a szállítási problémát mutatjuk be. Végül sort kerítünk a dualitás és az érzékenységvizsgálat tárgyalására is.

## I.2. Lineáris programozási problémaként modellezhető gyakorlati problémák

*1. Optimális termékösszetétel probléma.* Adott egy gyár, amelynek rendelkezésére állnak bizonyos erőforrások. Jelöljük ezek adott időszakra vonatkozó menynyiségeit $b_1, b_2, \ldots, b_m$ -mel, ahol $m$ a különböző erőforrások száma. Az adott gyárban $n$ féle terméket gyárthatnak, amelyeknek egyelőre ismeretlen mennyiségeit jelöljük $x_1, x_2, \ldots, x_n$ -nel. Nyilvánvaló, hogy $x_i \geq 0$, $i = 1, 2, \ldots, n$ kell, hogy teljesüljön. Ismerjük, hogy a $j$-edik termék egységének előállításakor az $i$-edik erőforrásból mennyit kell felhasználni, ezeket a mennyiségeket $a_{ij}$ -vel fogjuk jelölni. Végül adottak az egyes termékek egységeinek értékesítésekor elérhető hasznok, ezeket $c_1, c_2, \ldots, c_n$ -nel fogjuk jelölni. Ezek után a problémát úgy fogalmazhatjuk meg, hogy meghatározandó egy olyan termékösszetétel, amely a rendelkezésre álló erőforrásokból nem használ fel többet, mint amennyi azokból rendelkezésre áll, s amely mellett a nyereség maximális. Ennek megfelelő matematikai modell – feltételezve, hogy a felhasznált erőforrások és a haszon az $x_i$-knek lineáris függvényei – mátrixos alakban:

$$
\begin{gathered}
A\underline{x} \leq \underline{b} \\
\underline{x} \geq \underline{0} \\
\max\{\underline{c}^T \underline{x}\}.
\end{gathered}
$$

(Megjegyezzük, hogy az említett linearitási feltevések gyakran nem bizonyulnak jogosnak, így általában a lineáris modell csak közelítőleg írja le a valós problémát. Gondoljunk például a haszonra (azaz a célfüggvényre), ami általában nem lineáris függvénye a megtermelt mennyiségnek!)

*2. Táplálkozási probléma.* Egy adott időszakban egy adott élőlénynek bizonyos tápanyagokból meghatározott mennyiségűt kell a szervezetébe juttatni élete fenntartásához (illetve fejlődéséhez). Ezeket a tápanyagokat „ételek” (tápanyagkeverékek) formájában kapja meg. A különböző ételfajták száma legyen $n$, az ezekből elkészítendő (egyelőre ismeretlen) mennyiségeket pedig $x_1, x_2, \ldots, x_n$ jelöljék. Jelölje továbbá $a_{ij}$ a $j$-edik étel egységében az $i$-edik tápanyag mennyiségét. Jelöljék $b_1, b_2, \ldots, b_m$ az adott időszakban a különböző tápanyagból minimálisan felhasználandó mennyiségeket. Jelöljék végül az egyes ételek egységeinek elkészítési költségét $c_1, c_2, \ldots, c_n$. A probléma ekkor a következő: meghatározandók úgy az $x_1, x_2, \ldots, x_n$ nemnegatív ételmennyiségek, hogy azok az adott időszakban minden tápanyagból összességében tartalmazzák legalább a minimálisan szükséges menynyiséget és emellett az ételek elkészítésének összköltsége minimális legyen. A linearitási megszorítások mellett a matematikai modell mátrixos alakban:

$$A\underline{x} \geq \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\min\{\underline{c}^T\underline{x}\}.$$

Megjegyzés: a vizsgált időszak tartama nincs megszabva, ugyanakkor a modell nem foglalkozik az ételek felhasználásának az adott időszakon belüli ütemezésével.

## I.3. Lineáris programozási modellek

Az előbbi két klasszikus gyakorlati problémának – linearitást feltételezve – két különböző lineáris modell felelt meg, amelyek mátrixos alakban

$$\begin{array}{ccc} A\underline{x} \leq \underline{b} & & A\underline{x} \geq \underline{b} \\ \underline{x} \geq \underline{0} & \text{illetve} & \underline{x} \geq \underline{0} \\ \max\{\underline{c}^T\underline{x}\} & & \min\{\underline{c}^T\underline{x}\} \end{array}$$

voltak. A lineáris programozási problémák legáltalánosabb matematikai modelljét a következőképpen fogalmazhatjuk meg: az

$$\begin{aligned} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i, &\qquad i \in I_1 \\ a_{j1}x_1 + \cdots + a_{jn}x_n = b_j, &\qquad j \in I_2 \\ a_{k1}x_1 + \cdots + a_{kn}x_n \geq b_k, &\qquad k \in I_3 \\ x_l \geq 0, \qquad l \in L \subseteq \{1, 2, \ldots, n\} \end{aligned}$$

feltételek mellett meghatározandó a

$$c_1x_1 + c_2c_2 + \cdots + c_nx_n$$

lineáris függvény valamelyik szélsőértéke, és legalább egy szélsőértékhelye. Nyilvánvalóan igaz, hogy $I_i \cap I_j = \emptyset$, ha $i \neq j$ és $i = 1, 2, 3; j = 1, 2, 3$.

Legyen $I = I_1 \cup I_2 \cup I_3$, amelynek számosságát jelölje $m$, vagyis a nemnegativitási feltételeken kívül $m$ lesz a korlátozó feltételek száma.

Így tehát az általános lineáris programozási modellben lehetnek $\geq$, $=$ és $\leq$ formában megfogalmazható feltételek. Az ezeknek eleget tevő $\underline{x} \in \mathbb{R}^n$ vektorok halmazának (a megoldáshalmaznak) a zártságát a korlátozó feltételek mindegyikében szereplő egyenlőség biztosítja. (A megoldáshalmaz ezen tulajdonságát a későbbiekben ki fogjuk használni.) Az általános modellben nem kötelezően kell teljesítenie a döntési változóknak (az $x_l$-eknek) a nemnegativitási feltételeket, azaz lehetnek a modellben nem korlátozott változók is.

A szimplex módszernek nevezett megoldási algoritmust az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \leq \underline{0}$$
$$\max\{\underline{c}^T \underline{x}\}$$

formájú feladat megoldására fogjuk megadni, ezért ha a kiindulási feladatunk nem ilyen formájú, akkor azt előbb – „bizonyos értelemben ekvivalens” módon – ilyen alakra kell hoznunk. Felmerül a kérdés, hogy melyek azok az átalakítások, amelyek „bizonyos értelemben ekvivalens” problémákhoz vezetnek. Ezek a következők:

- Az $\underline{a}_i^T \underline{x} = b_i$ egyenlőség helyett a vele ekvivalens

$$\underline{a}_i^T \underline{x} \leq b_i$$
$$\underline{a}_i^T \underline{x} \geq b_i$$

  egyenlőtlenségrendszer írható.
- Az $\underline{a}_i^T \underline{x} \leq b_i$ egyenlőtlenség helyett vagy az
  $-\underline{a}_i^T \underline{x} \geq -b_i$ egyenlőtlenség, vagy pedig az $\underline{a}_i^T \underline{x} + u_i = b_i, u_i \geq 0$ írható.
- Az $\underline{a}_i^T \underline{x} \geq b_i$ egyenlőtlenség helyett vagy az $-\underline{a}_i^T \underline{x} \leq -b_i$ egyenlőtlenség, vagy pedig az $\underline{a}_i^T \underline{x} - u_i = b_i, u_i \geq 0$ írható.
- Ha egy $x_k$-ra nincs nemnegativitási megkötés, akkor helyette két nemnegatív változó különbségét írhatjuk, tehát

$$x_k = x'_k - x''_k, \quad x'_k \geq 0, \quad x''_k \geq 0$$

- Amennyiben a célfüggvény minimumát kell meghatároznunk, akkor helyette vehetjük a célfüggvény negatívjának maximumát. Hasonlóan: a célfüggvény maximuma helyett a negatív célfüggvény minimumát vehetjük.

A „bizonyos értelemben vett ekvivalens”-ség azt jelenti, hogy az eredeti és az átalakított feladatok egyszerre lesznek megoldhatók, illetve egyszerre rendelkezhetnek optimális megoldással, az optimum értékek megegyeznek, ugyanakkor a két feladat megoldásvektora dimenzióban eltérhet egymástól.

Nézzük most meg példaként, hogy az általános lineáris programozási feladat hogyan fogalmazható át az

$$\begin{gathered} A\underline{x} = \underline{b} \\ \underline{x} \geq \underline{0} \\ \max\{\underline{c}^T \underline{x}\} \end{gathered}$$

formájú feladattá, és hogy ezen két feladat „bizonyos értelemben ekvivalens”-e.

Induljunk ki tehát az

$$\begin{gathered} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i, \qquad i \in I_1 \\ a_{j1}x_1 + \cdots + a_{jn}x_n = b_j, \qquad j \in I_2 \\ a_{k1}x_1 + \cdots + a_{kn}x_n \geq b_k, \qquad k \in I_3 \\ x_l \geq 0, \qquad l \in L \subseteq \{1, 2, \ldots, n\} \\ \min\{c_1x_1 + c_2c_2 + \cdots + c_nx_n\} \end{gathered}$$

feladatból. A $\leq$ feltételek mindegyikét nemnegatív $u_i$ változók hozzáadásával egészítsük ki egyenlőséggé:

$$\begin{gathered} a_{i1}x_1 + \cdots + a_{in}x_n + u_i = b_i, \qquad i \in I_1 \\ u_i \geq 0. \end{gathered}$$

Az egyenlőséggel megadott feltételeket nem kell megváltoztatnunk, tehát azokat változatlanul visszük át az új modellbe. A $\geq$ feltételek mindegyikének baloldalából le kell vonnunk egy-egy $u_k \geq 0$ változót, ezzel egyenlőséggé alakítva az ilyen feltételeket is:

$$\begin{gathered} a_{k1}x_1 + \cdots + a_{kn}x_n - u_k = b_k, \qquad k \in I_3 \\ u_i \geq 0. \end{gathered}$$

Ha $s \notin L$, akkor az $x_s$ változó helyett be kell vezetnünk kettő új nemnegatív változó különbségét:

$$\begin{gathered} x_s = x_s' - x_s'', \quad \text{ahol} \\ x_s' \geq 0 \quad \text{és} \quad x_s'' \geq 0. \end{gathered}$$

(Ezt a helyettesítést a feladat minden feltételében végre kell hajtani!) Végül a

$$\min\{c_1x_1 + c_2c_2 + \cdots + c_nx_n\}$$

helyett a

$$\max\{-c_1x_1 - c_2c_2 - \cdots - c_nx_n\}$$

írandó. Így az új feladat:

$$
\begin{aligned}
a_{i1}x_1 + \dots + a_{in}x_n + u_i &= b_i, \qquad & i \in I_1 \\
a_{j1}x_1 + \dots + a_{jn}x_n &= b_j, \qquad & j \in I_2 \\
a_{k1}x_1 + \dots + a_{kn}x_n - u_k &= b_k, \qquad & k \in I_3 \\
u_i \geq 0, i \in I_1, u_k \geq 0, k \in I_3, x_l &\geq 0, \qquad & l \in L \subseteq \{1, 2, \dots, n\}
\end{aligned}
$$

(ha $s \notin L$, akkor $x_s = x'_s - x''_s$, ahol $x'_s \geq 0, x''_s \geq 0$)

$$\max\{-c_1x_1 - c_2x_2 - \dots - c_nx_n\}.$$

Egységes jelölést használva ez a probléma tehát az

$$
\begin{gathered}
A\underline{x} = \underline{b} \\
\underline{x} \geq \underline{0} \\
\max\{\underline{c}^T \underline{x}\}
\end{gathered}
$$

formának felel meg.

Vizsgálandó még, hogy teljesül-e a „bizonyos értelemben ekvivalens"-ség. Ehhez be kell látnunk, hogy ha az egyik feladat lehetséges (azaz megoldáshalmaza nem üres), akkor a másik is lehetséges, továbbá ha az egyik feladatnak van optimális megoldása, akkor a másiknak is van, és ekkor az optimum értékek megegyeznek.

Ha az általános feladat lehetséges, azaz a feltételeknek eleget tevő $\underline{x} \in \mathbb{R}^n$ vektorok halmaza nem üres, akkor az

$$
\begin{aligned}
u_i &= b_i - a_{i1}x_1 - \dots - a_{in}x_n, \qquad & i \in I_1 \\
u_k &= a_{k1}x_1 + \dots + a_{kn}x_n - b_k, \qquad & k \in I_3 \\
\text{ha } s \notin L, &\quad \text{akkor } x'_s = \max\{x_s, 0\}, \quad & x''_s = \max\{-x_s, 0\}
\end{aligned}
$$

választással megkapjuk az átalakított feladatban szereplő $u, x'_s, x''_s$-változók értékét, amelyek nyilvánvalóan teljesítik a nemnegativitást is. Ha pedig az átalakított feladat lehetséges, akkor annak tekintve egy megoldásának az $\underline{x} \in \mathbb{R}^n$ részét (felhasználva $x_s = x'_s - x''_s$ egyenlőségeket is minden $s \notin L$ esetén), az eleget tesz az általános feladatnak.

Vizsgáljuk most az általános feladat egy $\underline{x}^* \in \mathbb{R}^n$ optimális megoldását. Minthogy az $u_i, i \in I_1$ és $u_k, k \in I_3$ változók nem szerepelnek a átalakított feladat célfüggvényében, könnyen belátható (indirekt úton), hogy az

$$
\begin{aligned}
\underline{x}^* \in \mathbb{R}^n, \quad u_i^* &= b_i - a_{i1}x_1^* - \dots - a_{in}x_n^*, \qquad & i \in I_1 \\
u_k^* &= a_{k1}x_1^* + \dots + a_{kn}x_n^* - b_k, \qquad & k \in I_3 \\
{x'_s}^* &= \max\{x_s^*, 0\}, \quad {x''_s}^* = \max\{-x_s^*, 0\}
\end{aligned}
$$

optimális megoldása az átalakított feladatnak az $\underline{x}^*$-hoz tartozó célfüggvény-értékkel.

(Ha ugyanis ez a megoldás nem optimális az átalakított feladatnál, akkor létezik egy olyan megoldása ennek a feladatnak, amelyik az előbbi megoldáshoz tartozó célfüggvényértéknél nagyobb értéket ad. Ekkor azonban ezen megoldás $\underline{x} \in \mathbb{R}^n$ része eleget tesz az eredeti általános feladatnak, s hozzá nagyobb függvényérték tartozik, mint ami az $\underline{x}^*$-hoz tartozott. Ez ellentmondás, hiszen $\underline{x}^*$ optimális megoldása volt az általános feladatnak.)

Az előbbiekhez hasonlóan belátható, hogy ha megtaláltuk az átalakított feladat egy optimális megoldását, akkor annak $\underline{x}$ része optimális megoldása lesz az általános feladatnak. (Természetesen figyelembe véve az $x_s = x'_s - x''_s, \forall s \notin L$, egyenlőségeket is!)

Mindezek alapján a továbbiakban az

$$
\begin{gathered}
A\underline{x} = \underline{b} \\
\underline{x} \geq \underline{0} \\
\max\{\underline{c}^T \underline{x}\}
\end{gathered}
$$

feladatot fogjuk vizsgálni. Ha a kiinduló feladatunk nem ilyen formájú, akkor a leírt átalakításokat használva a már említett értelemben ekvivalens ilyen formájú feladatot definiálhatunk.

## I.4. A szimplex módszer elméleti hátterének rövid áttekintése

Tekintsük az

$$
\begin{gathered}
A\underline{x} = \underline{b} \\
\underline{x} \geq \underline{0} \\
\max\{\underline{c}^T \underline{x}\}
\end{gathered}
$$

lineáris programozási feladatot. Az $A\underline{x} = \underline{b}$ lineáris egyenletrendszer helyett tekintve a vele ekvivalens

$$
\begin{gathered}
A\underline{x} \leq \underline{b} \\
-A\underline{x} \leq -\underline{b}
\end{gathered}
$$

egyenlőtlenségrendszert, belátható, hogy a feladat feltételrendszerének eleget tevő $\underline{x} \in \mathbb{R}^n$ vektorok összessége egy ún. konvex poliédert alkot. (Véges sok zárt féltér metszetét konvex poliédernek nevezzük. Egy zárt félteret algebrailag az $\underline{a}^T \underline{x} \leq b$ vagy az $\underline{a}^T \underline{x} \geq b$ alakban adhatunk meg, így az előbbi feladat feltételrendszere valóban egy konvex poliédert határoz meg.) Számosságát tekintve egy konvex poliéder lehet üres halmaz, egyelemű, illetve kontinuum számosságú. Az első esetben a feladat feltételrendszere ellentmondásos, a második esetben pedig az egyetlen megoldás egyben az optimális megoldás is. A harmadik esetben viszont kontinuum számosságú pontból kell kiválasztanunk azt az egyet, amelynél a célfüggvény felveszi a kívánt szélsőértéket. A lineáris programozási feladatok megoldása szempontjából óriási jelentősége van annak, hogy ebben az esetben is véges sok pont vizsgálata alapján dönthetünk arról, hogy a feladatunk megoldható-e vagy sem, s ha megoldható, akkor mi lesz az optimális értéket szolgáltató megoldás.

Lineáris programozási feladatokra grafikus megoldást az $n = 2$ esetben használhatunk, $n \geq 3$ esetén már csak algebrailag kezelhetők ezek a problémák. Mindezek miatt fontos kérdés az, hogy egy lineáris programozási feladat feltételrendszere által meghatározott konvex poliédert hogyan lehet algebrailag megadni. A választ Motzkin-Goldman tétele adja meg, amely szerint a

$$K = \{\underline{x} \mid \underline{x} \in \mathbb{R}^n, \quad A\underline{x} \leq \underline{b}\}$$

konvex poliéder előállítható $P^{\Delta} + Q^{\angle}$ alakban, ahol

$$\begin{aligned} P^{\Delta} &= \{\underline{x} \mid \underline{x} = \sum_{i=1}^{r} \lambda_i \underline{p}_i, \quad \lambda_i \geq 0, \quad i = 1, \ldots, r, \quad \sum_{i=1}^{r} \lambda_i = 1\} \\ Q^{\angle} &= \{\underline{x} \mid \underline{x} = \sum_{j=1}^{s} \mu_j \underline{q}_j, \quad \mu_j \geq 0, \quad j = 1, \ldots, s\} \\ &\underline{x} \in \mathbb{R}^n. \end{aligned}$$

Ezen előállításban a $Q^{\angle} = \{\underline{x} \mid A\underline{x} \leq \underline{0}\}$. Másrészt minden $P^{\Delta} + Q^{\angle}$ alakban előálló $K$ vektorhalmaz egy konvex poliéder az $n$ dimenziós térben, azaz $P^{\Delta} + Q^{\angle} = \{\underline{x} \mid A\underline{x} \leq \underline{b}\}$.

A $P^{\Delta}$ alakú halmazokat konvex politopoknak, a $Q^{\angle}$ alakúakat pedig konvex kúpoknak nevezzük. Így Motzkin-Goldman tétele azt állítja, hogy bármely konvex poliéder előállítható egy konvex politop és egy konvex kúp összegeként, illetve megfordítva: egy konvex politop és egy konvex kúp összegeként előálló halmaz konvex poliédert képez az $n$ dimenziós térben. Ezen tétel alapján az

$$\begin{gathered} A\underline{x} = \underline{b} \\ \underline{x} \geq \underline{0} \\ \max\{\underline{c}^T \underline{x}\} \end{gathered}$$

problémát átfogalmazhatjuk a

$$\max_{\underline{x}\in K}\left\{\sum_{i=1}^{r}\lambda_i\underline{c}^T\underline{p}_i+\sum_{j=1}^{s}\mu_j\underline{c}^T\underline{q}_j,\quad \lambda_i\geq 0,\quad i=1,\dots,r,\right.$$
$$\left.\sum_{i=1}^{r}\lambda_i=1,\quad \mu_j\geq 0,\quad j=1,\dots,s\right\}$$

problémává, ahol

$$K=\{\underline{x}|A\underline{x}\leq\underline{b},\quad \underline{x}\geq 0\}\text{ és}$$
$$\underline{x}=\sum_{i=1}^{r}\lambda_i\underline{p}_i+\sum_{j=1}^{s}\mu_j\underline{q}_j$$
$$\lambda_i\geq 0,\qquad \mu_j\geq 0,\quad j=1,2,\dots,s$$
$$i=1,2,\dots,r$$
$$\sum_{i=1}^{r}\lambda_i=1.$$

A $P^{\Delta}$ konvex politop egy korlátos, zárt halmaz, így ezen halmaz felett a célfüggvény korlátos, vagyis korlátos a

$$\left\{\sum_{i=1}^{r}\lambda_i\underline{c}^T\underline{i}_i\mid\lambda_i\geq 0,\quad i=1,\dots,r,\quad \sum_{i=1}^{r}\lambda_i=1\right\}$$

halmaz. Ugyanakkor $Q^{\angle}$ előállításában szereplő $\mu_j$ értékektől csak a nemnegativitást követeljük meg, ezért ha létezik olyan $j\in\{1,2,\dots,s\}$, hogy $\underline{c}^T\underline{q}_j>0$, akkor a $\mu_j>0$ értéket tetszőleges nagyra választva a célfüggvény értéke minden határon túl fog növekedni, azaz a problémánknak nem lesz optimális megoldása. Ahhoz tehát, hogy létezzen véges maximum érték, a $\underline{c}^T\underline{q}_j\leq 0$ kell, hogy teljesüljön minden $j\in\{1,2,\dots,s\}$-re. Ez esetben viszont akkor lehet egy $\underline{x}\in K$ optimális megoldás, ha $\mu_j\underline{c}^T\underline{q}_j=0$ teljesül minden $j\in\{1,2,\dots,s\}$ esetén. A célfüggvény maximumát ilyenkor tehát a $P^{\Delta}$ politop felett kell meghatároznunk. Minthogy azonban a $P^{\Delta}$ halmaz számossága általában kontinuum, ezen probléma megoldása még mindig komoly nehézséget jelent. A konvex poliéder extrémális pontjainak segítségével azonban ezt a problémát már visszavezethetjük véges sok pont közül a legjobb kiválasztásának problémájára.

Egy konvex poliéder (általánosabban: egy konvex halmaz) extrémális pontján (másképpen: csúcspontján) olyan pontot értünk, amelyik nem belső pontja egyetlen olyan valódi szakasznak sem, amely szakasz végpontjaival együtt a konvex poliéderhez tartozik.

A konvex poliéder egy ilyen pontjának algebrai előállítását adja meg a következő tétel: a

$$K = \{\underline{x} \mid \underline{x} \in \mathbb{R}^n, \quad A\underline{x} = \underline{b}, \quad \underline{x} \geq \underline{0}\}$$

konvex poliéder extrémális pontjait pontosan a megengedett bázismegoldások adják. (Ezen állításban a megengedettség az $\underline{x}$ nemnegativitását jelenti. Az $A\underline{x} = \underline{b}$ egyenletrendszer egy bázismegoldását pedig úgy nyerjük, hogy az $A$ mátrix oszlopvektorrendszeréből kiválasztunk egy $B$ bázist, majd megoldjuk a $B\underline{x}_B = \underline{b}$ redukált egyenletrendszert, és a kapott $\underline{x}_B$ megoldásvektort 0 komponensekkel kiegészítjük $n$ dimenziós vektorrá. Itt $\underline{x}_B$ az $\underline{x}$ vektor azon komponenseiből képezett vektort jelenti, amely komponensek a $B$ bázisba választott $A$ mátrixbeli oszlopvektorokhoz tartoznak.)

Ezen tétel alapján tehát az $A\underline{x} = \underline{b}, \quad \underline{x} \geq \underline{0}$ feltételek által meghatározott konvex poliéder extrémális pontjait a megengedett bázismegoldások és csakis azok alkotják. Minthogy az $A$ mátrix oszlopvektor rendszeréből csak véges sok $B$ bázis választható ki, ezért méginkább igaz, hogy az $A\underline{x} = \underline{b}$ lineáris egyenletrendszernek véges sok megengedett bázismegoldása van, azaz véges sok extrémális pontja van a $K = \{\underline{x} \mid A\underline{x} = \underline{b}, \quad \underline{x} \geq 0\}$ konvex poliédernek.

A problémánk megoldása szempontjából az extrémális pontok jelentőségét a következő tétel fogalmazza meg:

Tekintsük az $\mathbb{R}^n$-ben egy tetszőleges $K \neq \emptyset$ konvex poliédert és legyen $\underline{c}^T\underline{x}$ egy lineáris függvény. Ekkor a $\underline{c}^T\underline{x}$ vagy nem korlátos felülről a $K$ felett, vagy van a $K$-nak olyan extrémális pontja is, amelyben a $\underline{c}^T\underline{x}$ felveszi a maximumát.

Ezen tételt felhasználva a lineáris programozási problémánkat a következőképpen is megfogalmazhatjuk:

Meghatározandó az $A\underline{x} = \underline{b}$, $\underline{x} \geq \underline{0}$ feltételek által meghatározott konvex poliéder (amennyiben nem üres halmaz) azon extrémális pontja, amelyben a $\underline{c}^T\underline{x}$ lineáris függvény felveszi a poliéder felett a maximumát, vagy megmutatandó, hogy a feltételek által meghatározott konvex poliéder felett a $\underline{c}^T\underline{x}$ lineáris függvény felülről nem korlátos.

Így tehát a lineáris programozási feladatunk megoldását véges sok pont vizsgálata segítségével adhatjuk meg. Meg kell azonban jegyeznünk, hogy még ez is komoly feladat, mert egyrészt elő kell tudnunk állítani az extrémális pontokat, másrészt az extrémális pontok száma is lehet olyan nagy (persze véges), hogy a gyakorlati kivitelezés nem valósítható úgy meg, hogy minden extrémális pontot előállítunk, kiszámítjuk ezeknél a célfüggvény értékét, majd kiválasztjuk közülük az optimumot szolgáltatót.

A szimplex módszer lesz az a módszer, amely az előbbi eredményeket is felhasználja, s amelynek segítségével a lineáris programozási problémánkat meg fogjuk tudni oldani.

## I.5. A szimplex módszer

A szimplex módszer lényege – geometriailag – a következőképpen fogalmazható meg: kiindulunk a feladat feltételrendszere által meghatározott konvex poliéder egyik extrémális pontjából, majd kiszámítjuk az ebben a pontban adódó célfüggvényértéket. Ezt követően eldöntjük, hogy ez a megoldás optimális-e, vagy hogy van-e egyáltalán véges optimum érték. Amennyiben egyikre sem lehet választ adni, áttérünk egy olyan szomszédos extrémális pontra, amelyhez tartozó célfüggvényérték nem kisebb, mint amekkora a korábbi volt. Ezt az eljárást mindaddig folytatjuk, amíg vagy meg nem kapjuk az optimális megoldást, vagy ki nem mutatjuk, hogy nincs véges optimum érték.

Ezzel az algoritmussal kapcsolatosan azonnal felmerülhetnek a következő kérdések:

- Hogyan határozzunk meg egy kiinduló extrémális pontot?
- Mi a biztosíték arra, hogy véges sok lépésben véget ér az algoritmus?

Ezekre és még néhány más kérdésre is az algoritmus ismertetése után adjuk meg a választ.

Legyen adott az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max\{\underline{c}^T \underline{x}\}$$

lineáris programozási feladat, ahol $A$ egy $m \times n$-es adott mátrixot, a $\underline{b}$ egy adott $m$-dimenziós vektort, a $\underline{c}$ egy adott $n$-dimenziós vektort és $\underline{x}$ egy $n$-dimenziós ismeretlen vektort jelölnek. Az $A$ mátrix oszlopaira bevezetve az

$$\underline{a}_i = \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix} \qquad i = 1, 2, \ldots, n$$

jelölést, az $A\underline{x} = \underline{b}$ egyenletrendszert az

$$x_1\underline{a}_1 + x_2\underline{a}_2 + \ldots + x_n\underline{a}_n = \underline{b}$$

alakban is felírhatjuk. Az $m$ és $n$ számokkal kapcsolatban – ami ezek egymáshoz viszonyított nagyságát illeti – nincs semmilyen kikötésünk, de megjegyezzük, hogy a gyakorlatban előforduló problémákban $n > m$ esettel találkozunk legtöbbször.

Az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_n$ vektorok által kifeszített altérnek legyen $r$ a dimenziója. Ha az

$$x_1\underline{a}_1 + x_2\underline{a}_2 + \ldots + x_n\underline{a}_n = \underline{b}$$

egyenletrendszernek van megoldása, akkor $\underline{b}$ eleme az előbb említett altérnek. Korábban már megállapítottuk, hogy az $A\underline{x} = \underline{b}$, $\underline{x} \geq \underline{0}$ feltételek által meghatározott konvex poliéder extrémális pontjait az $A\underline{x} = \underline{b}$ megengedett bázismegoldásai alkotják. Azt is tudjuk, hogy egy $K$ konvex poliéder felett a $\underline{c}^T\underline{x}$ lineáris függvény vagy nem korlátos felülről, vagy van olyan extrémális pontja a $K$-nak, amelyben a $\underline{c}^T\underline{x}$ függvény felveszi a maximumát.

Legyen $B = \{\underline{a}_i \mid i \in I_B\}$ egy megengedett bázisa az $A$ mátrix oszlopvektor redszerének. Az $I_B \subseteq \{1, 2, \ldots, n\}$ a $B$ bázist alkotó vektorok indexeinek halmaza. Jelöljük $\underline{x}_B$-vel a $B$-hez tartozó, ún. báziskomponensekből készített vektort, amely a $B\underline{x}_B = \underline{b}$ egyenletrendszernek egyértelmű megoldása. Ez az $\underline{x}_B$ megoldás megadja a $\underline{b}$ vektornak a $B$ bázisba tartozó vektorokkal való előállítását. Állítsuk elő az $\underline{a}_k$, $k \in \{1, 2, \ldots, n\}$ vektorokat is a bázisba tartozó vektorokkal:

$$\underline{a}_k = \sum_{i \in I_B} d_{ik}\underline{a}_i, \qquad k = 1, 2, \ldots, n.$$

Vezessük be a

$$z_k = \sum_{i \in I_B} d_{ik}c_i, \qquad k = 1, 2, \ldots, n$$

mennyiségeket is, majd számítsuk ki a $z_k - c_k$, $k = 1, 2, \ldots, n$ értékeket. Adatainkat a következő táblázatba foglalhatjuk, amit szokás szimplex táblázatnak is nevezni:

A megengedett bázist itt a $B = \{\underline{a}_{i_1}, \underline{a}_{i_2}, \ldots, \underline{a}_{i_r}\}$ jelöli, amelybe tehát $r$ darab vektor tartozik. Így $I_B = \{i_1, i_2, \ldots, i_r\}$. Az $f$ az $\underline{x}_B$ bázismegoldáshoz tartozó célfüggvényérték, tehát

$$f = \underline{c}_B^T\underline{x}_B = c_{i_1}x_{i_1} + \ldots + c_{i_r}x_{i_r}.$$

A szimplex táblázatunk második sora és második oszlopa csak szimbólumokat tartalmaz, a többi helyen a megfelelő számszerű adat szerepel.

Egy szimplex tábla az alábbi három, egymást kizáró tipus valamelyikébe tartozik:

a) $z_k - c_k \geq 0$ teljesül minden $k \in \{1, 2, \ldots, n\}$ esetén.
b) Létezik olyan $k$, hogy $z_k - c_k < 0$ és egy ilyen $k$ esetén $d_{ik} \leq 0$ teljesül minden $i \in I_B$-re.
c) Létezik olyan $k$, hogy $z_k - c_k < 0$, és minden ilyen $k$ esetén létezik olyan $i \in I_B$, hogy $d_{ik} > 0$.

Az a) esettel kapcsolatban bebizonyítjuk a következő tételt.

**Tétel.** *Ha a szimplex táblázat az a) esetnek megfelelő, azaz $z_k - c_k \geq 0$ teljesül minden $k \in \{1, 2, \ldots, n\}$ esetén, akkor a táblázatban lévő $\underline{x}_B$ bázismegoldás optimális megoldása a feladatnak, az optimum értéke pedig az $f$ érték lesz.*

*Bizonyítás.* Megmutatjuk, hogy tetszőleges $\underline{y}$ megengedett megoldás esetén $\underline{c}^T \underline{y} \leq \underline{c}_B^T \underline{x}_B$ teljesül. Minthogy $z_k - c_k \geq 0$, ezért $z_k \geq c_k$ teljesül minden $k \in \{1, 2, \ldots, n\}$ esetén. Az $\underline{y} \geq 0$ miatt igaz továbbá, hogy $y_k \geq 0, k \in \{1, 2, \ldots, n\}$.

Így

$$\begin{aligned} \sum_{k=1}^{n} c_k y_k \leq \sum_{k=1}^{n} z_k y_k = \sum_{k=1}^{n} \left( \sum_{i \in I_B} d_{ik} c_i \right) y_k = \\ = \sum_{i \in I_B} c_i \sum_{k=1}^{n} d_{ik} y_k. \end{aligned} \qquad (*)$$

Továbbá a $\sum_{i\in I_B} \underline{a}_i x_i = \underline{b}$ lineáris egyenletrendszernek egyértelmű megoldása az $\underline{x}_B$ vektor, ezért

$$\underline{b} = \sum_{k=1}^{n} \underline{a}_k y_k = \sum_{k=1}^{n} y_k \left( \sum_{i\in I_B} d_{ik} \underline{a}_i \right) = \sum_{i\in I_B} \underline{a}_i \left( \sum_{k=1}^{n} d_{ik} y_k \right) = \sum_{i\in I_B} \underline{a}_i x_i,$$

ahol az említett egyértelmű megoldhatóság miatt $\sum_{k=1}^{n} d_{ik} y_k = x_i$. Ez utóbbit felhasználva a $(*)$ egyenlőtlenségnél kapjuk, hogy

$$\sum_{k=1}^{n} c_k y_k \le \sum_{i\in I_B} c_i x_i,$$

ami éppen a bizonyítandó egyenlőtlenség.

A b) és a c) esetek tárgyalásához végezzük el a következő átalakításokat:

$$\begin{aligned} \underline{b} &= \sum_{i\in I_B} x_i \underline{a}_i = \sum_{i\in I_B} x_i \underline{a}_i - \beta \underline{a}_k + \beta \underline{a}_k = \\ &= \sum_{i\in I_B} x_i \underline{a}_i - \beta \sum_{i\in I_B} d_{ik} \underline{a}_i + \beta \underline{a}_k = \sum_{i\in I_B} (x_i - \beta d_{ik}) \underline{a}_i + \beta \underline{a}_k \ , \end{aligned} \qquad (**)$$

ahol $\beta \ge 0$ és a $k \notin I_B$ olyan index, amelyre $z_k - c_k < 0$.

Ha az $x_i - \beta d i_k \ge 0$ minden $i \in I_B$ esetén teljesül valamely $\beta > 0$ mellett, akkor az $A\underline{x} = \underline{b}$, $\underline{x} \ge \underline{0}$ feltételrendszernek megkaptuk egy újabb megoldását, amelyhez a

$$\begin{aligned} \sum_{i\in I_B} c_i (x_i - \beta d_{ik}) + c_k \beta &= \sum_{i\in I_B} c_i x_i - \beta \sum_{i\in I_B} c_i d_{ik} + c_k \beta = \\ &= \sum_{i\in I_B} c_i x_i - \beta z_k + c_k \beta = \sum_{i\in I_B} c_i x_i - \beta (z_k - c_k) \end{aligned}$$

célfüggvényérték tartozik. A kapott eredményből látszik, hogy az új célfüggvényérték a régitől a

$$-\beta(z_k - c_k)$$

-val tér el. Így ha a szimplex táblázatunk a b) esetnek megfelelő, akkor bármilyen nagy $\beta > 0$ érték mellett a $(**)$-ban szereplő $\underline{x}$ vektor megoldása lesz a problémánknak (azaz eleget tesz az $A\underline{x} = \underline{b}$, $\underline{x} \ge \underline{0}$ feltételeknek). Ezen $\underline{x}$-hez tartozó célfüggvényérték a $\beta$ növelésével minden határon túl növelhető lesz, így ebben az esetben a problémának nem létezik véges optimuma. Ezzel tehát bebizonyítottuk a következő tételt: ha a $B$ bázishoz tartozó szimplex tábla a b) tipushoz tartozik, akkor az $A\underline{x} = \underline{b}$, $\underline{x} \ge \underline{0}$ feltételek mellett a $\underline{c}^T \underline{x}$ célfüggvény felülről nem korlátos, azaz a lineáris programozási feladatunknak nincs véges optimum értéke.

Más a helyzet azonban a c) esetben, ugyanis minden olyan $k$-ra, amelynél $z_k - c_k < 0$, létezik olyan $i \in I_B$, hogy $d_{ik} > 0$, így a $\beta$ értékét nem növelhetjük tetszőlegesen. Ez esetben is egy adott $\beta > 0$ mellett $-\beta(z_k - c_k)$-val változik a célfüggvény értéke. El kell azonban érnünk azt is, hogy az új megoldás is teljesítse a megengedettséget, vagyis minden olyan $i \in I_B$ esetén, amikor $d_{ik} > 0$, az $x_i - \beta d_{ik} \geq 0$ kell, hogy teljesüljön. Minthogy az új megoldáshoz tartozó célfüggvényérték a $\beta > 0$-nak szigorúan monoton növekvő függvénye, ezért arra törekszünk, hogy a $\beta$-t a lehető legnagyobb értékűnek válasszuk. Az előbbiek alapján ez a

$$\beta = \frac{x_j}{d_{jk}} = \min_{\substack{i\,:\,d_{ik} > 0 \\ i \in I_B}} \left\{ \frac{x_i}{d_{ik}} \right\}$$

lesz.

(Itt $j \in I_B$ azt az indexet jelöli, ahol az előbbi minimumérték adódik.) Ha tehát a $\beta$ értékét az előbbi minimumfeltételnek megfelelően választjuk, akkor az $\underline{a}_j$ vektor együtthatója, azaz az új $x_j$ értéke 0 lesz, míg az eddig 0 együtthatóval rendelkező $\underline{a}_k$ vektornak $\beta \geq 0$ lesz az együtthatója, azaz az új $x_k$ értéke $\beta$ lesz. Ez valójában tehát azt jelenti, hogy a c) esetben a $B = B_0$ bázisról áttérünk egy olyan $B_1$ bázisra, amely megengedett, s amely a $B_0$-tól csak egy vektorban különbözik és amelyhez tartozó bázismegoldás célfüggvényértéke nem kisebb, mint a korábbi (induló) bázismegoldáshoz tartozó célfüggvényérték.

A $\beta = 0$ csak abban az esetben fordulhat elő, ha a $B_0$ bázishoz tartozó $\underline{x}_{B_0}$ bázismegoldásnak van 0 komponense, azaz ha degenerált esetről van szó.

Ha tehát a szimplex táblánk a c) esetnek megfelelő, akkor kiválasztunk egy olyan $k \in \{1, 2, \ldots, n\}$ indexet, amelyhez $z_k - c_k < 0$ tartozik, majd a $\beta$ értékét a minimum feltételeknek megfelelően választva végrehajtjuk a bázistranszformációt. Az új $B_1$ bázis a régi $B = B_0$ bázistól csak egy vektorban tér el, azaz elemi bázistranszformációt hajtottunk végre. A $B_1$ bázist az $\underline{a}_i, i \in I_B, i \neq j$ és az $\underline{a}_k$ vektorok alkotják.

Az, hogy ezen $B_1$ vektorrendszerhez tartozó megoldás megengedett, azaz teljesíti a nemnegativitást, s hogy ezen új megoldáshoz tartozó célfüggvényérték nem kisebb, mint a korábbi $f$ célfüggvényérték, nyilvánvaló. Nem nyilvánvaló azonban, hogy az $\underline{a}_j \longleftrightarrow \underline{a}_k$ csere után újból bázishoz jutunk, így ez utóbbit még be kell látnunk. A $B_1$-hez tartozó vektorok lineáris függetlenségét a következő tétel biztosítja. $\square$

**Tétel.** *Ha az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_n$ vektorok lineárisan függetlenek, továbbá ha az $\underline{a} = \sum_{i=1}^{n} \alpha_i \underline{a}_i$, $\underline{a} \neq \underline{a}_i$ $(i = 1, 2, \ldots, n)$ teljesül, akkor az*

$$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_{j-1}, \underline{a}, \underline{a}_{j+1}, \ldots, \underline{a}_n$$

*vektorrendszer lineáris függetlenségének szükséges és elégséges feltétele az $\alpha_j \neq 0$.*

Minthogy a c) esetben a $d_{jk} > 0$, ez biztosítja, hogy a $B_1$-et alkotó vektorok is lineárisan függetlenek, tehát, hogy bázist alkotnak.

Az előbbi eredményeinket összefoglalva:

A szimplex módszer alkalmazásakor kiindultunk egy $B_0$ megengedett bázisból. Ehhez elkészítettük a szimplex táblázatot, majd azt kiértékeltük. Ha a szimplex táblánk az a) esetnek megfelelő volt, akkor az eljárás be is fejeződött. A táblázat tartalmazza az optimális megoldást és az optimum értéket is. Ha a b) eset adódott, akkor is vége lett az eljárásnak, ebben az esetben azonban nem létezett a feladatunknak véges optimum értéke. Végül a c) esetnek megfelelő szimplex táblázat esetén még sem azt nem tudjuk, hogy már optimális esetnél vagyunk-e, sem pedig azt, hogy egyáltalán a feladatnak van-e véges optimum értéke. Ez esetben áttértünk egy újabb $B_1$ bázisra, amelyhez újból elkészítjük a szimplex táblázatot, majd kezdjük elölről a kiértékelést. Ahányszor a c) esetnek megfelelő a szimplex táblánk, annyiszor hajtjuk végre a bázistranszformációt. Így a $B_0, B_1, B_2, \ldots$ megengedett bázisok sorozatához jutunk, amelyben az egymást követők csak egy vektorban különböznek egymástól, és amelyekhez tartozó $\underline{x}_{B_0}, \underline{x}_{B_1}, \ldots$ bázismegoldásokon a célfüggvényértékek monoton növekvő sorozatot alkotnak. Amennyiben

$$\underline{c}_{B_0}^T \underline{x}_{B_0} < \underline{c}_{B_1}^T \underline{x}_{B_1} < \cdots$$

teljesül, akkor az $A\underline{x} = \underline{b}$, $\underline{x} \geq \underline{0}$ feltételrendszer által meghatározott $K$ konvex poliédernek mindig olyan csúcsára (extrémális pontjára) megyünk, ahol korábban még nem voltunk. Minthogy pedig a konvex poliédernek véges sok csúcsa van, ilyen esetben véges sok lépés után vagy a)-nál, vagy pedig b)-nél áll le az algoritmus.

Degenerált esetben, azaz ha az $\underline{x}_{B_j}$-nek van 0 komponense is, előfordulhat az is, hogy a szimplex algoritmus során olyan bázishoz is eljutunk, amely már korábban is szerepelt. Ezt a jelenséget ciklizálásnak nevezzük, ilyenkor ugyanis az algoritmus végtelen ciklusba kerül. Ha azonban az ún. lexikografikus szimplex módszert alkalmazzuk, akkor ez az eset nem fordulhat elő, azaz degenerált esetben is biztosítva lesz az algoritmus véges sok lépésben történő befejeződése.

*I.5.1. Transzformációs formulák.* Most azt vizsgáljuk, hogy a c) esetben az egyik bázisról a másikra való áttérés során hogyan nyerhetjük a régi szimplex táblából az újat. Ezen áttérést leíró formulákat transzformációs formuláknak nevezzük.

Legyen

$B_0 = \{\underline{a}_{i_1}, \underline{a}_{i_2}, \dots, \underline{a}_{i_r}\}$

$I_{B_0} = \{i_1, i_2, \dots, i_r\}$

$B_1 = \{\underline{a}_i \mid \underline{a}_i \in B_0,\ i \neq j,\ i \in I_{B_0},\ j \in I_{B_0}\} \cup \{\underline{a}_k\}$,

ahol a $k \notin I_{B_0}$ olyan index, amelyre

$z_k - c_k < 0$ teljesül; a $j$ index pedig olyan, hogy $\dfrac{x_j}{d_{jk}} = \min\limits_{\substack{i\,:\,i \in I_{B_0}\\ d_{ik} > 0}} \left\{\dfrac{x_i}{d_{ik}}\right\}$.

Tegyük fel, hogy az $\underline{a}_p, p \in \{1, 2, \dots, n\}$ vektornak ismert a $B_0$ bázisbeli előállítása, és meghatározandó a $B_1$-beli előállítás az előbbi információkat felhasználva. Legyen

$$\underline{a}_p = \sum_{i \in I_{B_0}} d_{i_p}^{(0)} \underline{a}_i \qquad \forall p \in \{1, 2, \dots, n\}$$

és

$$\underline{a}_p = \sum_{i \in I_{B_1}} d_{i_p}^{(1)} \underline{a}_i \qquad \forall p \in \{1, 2, \dots, n\}.$$

Minthogy a $B_1$ bázis a $B_0$-tól csak az $\underline{a}_k$ vektorban különbözik, amelyet az $\underline{a}_j$ helyébe hozhatunk be, ezért végezzük el a következő átalakításokat:

$$\underline{a}_p - \gamma \underline{a}_k = \sum_{i \in I_{B_0}} d_{ip}^{(0)} \underline{a}_i - \gamma \sum_{i \in I_{B_0}} d_{ik}^{(0)} \underline{a}_i = \sum_{i \in I_{B_0}} \left(d_{ip}^{(0)} - \gamma d_{ik}^{(0)}\right) \underline{a}_i.$$

Felhasználva a $d_{jk}^{(0)} > 0$ relációt, a $\gamma$ számot úgy határozzuk meg, hogy az $\underline{a}_p = \sum_{i \in I_{B_0}} \left(d_{ip}^{(0)} - \gamma d_{ik}^{(0)}\right) \underline{a}_i + \gamma \underline{a}_k$ előállításban az $\underline{a}_j$ együtthatója 0 legyen. Így a $d_{jp}^{(0)} - \gamma d_{jk}^{(0)} = 0$, azaz $\gamma = \frac{d_{jp^{(0)}}}{d_{jk}^{(0)}}$ kell, hogy teljesüljön. Ekkor

$$\underline{a}_p = \sum_{\substack{i \in I_{B_0}\\ i \neq j}} \left(d_{ip}^{(0)} - \frac{d_{jp}^{(0)}}{d_{jk}^{(0)}} d_{ik}^{(0)}\right) \underline{a}_i + \frac{d_{jp}^{(0)}}{d_{jk}^{(0)}} \underline{a}_k,$$

vagyis megkaptuk az $\underline{a}_p$ vektor $B_1$-beli előállítását. Azt kaptuk tehát, hogy

$$d_{ip}^{(1)} = d_{ip}^{(0)} - \frac{d_{jp}^{(0)}}{d_{jk}^{(0)}} d_{ik}^{(0)}, \text{ minden } i \in I_{B_0},\ i \neq j \text{ esetén}$$

és

$$d_{kp}^{(1)} = \frac{d_{jp}^{(0)}}{d_{jk}^{(0)}}.$$

Jelöljük ezek után

$$\underline{\delta}_i^{(0)}\text{-lal illetve } \underline{\delta}_i^{(1)}\text{-gyel}$$

a $B_0$-hoz illetve a $B_1$-hez tartozó szimplex tábla $\underline{a}_i$ $(i \in I_{B_0} \cap I_{B_1})$ vektorához tartozó sorát, beleértve első komponensként az $x_i$ értéket is.

A $\underline{b} = \underline{a}_0$ jelölést, és az

$$x_i^{(1)} = x_i^{(0)} - \frac{x_j^{(0)}}{d_{jk}^{(0)}} d_{ik}^{(0)} \qquad \forall i \in I_{B_0}, i \neq j$$

$$x_k^{(1)} = \frac{x_j^{(0)}}{d_{jk}^{(0)}}$$

összefüggéseket felhasználva kapjuk, hogy

$$\underline{\delta}_i^{(1)} = \underline{\delta}_i^{(0)} - \frac{d_{ik}^{(0)}}{d_{jk}^{(0)}} \underline{\delta}_j^{(0)}, \qquad \forall i \in I_{B_0},\ i \neq j$$

$$\underline{\delta}_k^{(1)} = \frac{1}{d_{jk}^{(0)}} \underline{\delta}_j^{(0)}.$$

Amennyiben a $d_{jk}^{(0)} > 0$ számot generáló elemnek nevezzük és végrehajtjuk az elemi bázistranszformációt kicserélve az $\underline{a}_j$ vektort az $\underline{a}_k$ vektorral, akkor az előbbi transzformációs formulákat „szövegesen" a következőképpen fogalmazhatjuk meg.

Az új szimplex tábla generáló elemének új sorát (az $\underline{a}_k$ sorát) megkapjuk, ha a régi megfelelő sort (az $\underline{a}_j$ sorát) végigosztjuk a generáló elemmel.

Az új szimplex táblánk minden más sorát (mint ahogy azt mindjárt látni fogjuk, beleértve az utolsó sort is) megkapjuk, ha a régi megfelelő sorból kivonjuk a generáló elem új sorának annyiszorosát, amennyi a régi szimplex táblázatban a generáló elem oszlopában és a megfelelő sorban van (azaz a $d_{ik}^{(0)}$-szorosát).

Hiányzik még a szimplex tábla utolsó sorának a transzformációja. Jelöljük $\underline{\delta}^{(0)}$-lal a $B_0$-hoz, $\underline{\delta}^{(1)}$-gyel pedig a $B_1$-hez tartozó szimplex tábla utolsó sorát. Már tudjuk, hogy az

$$f^{(1)} = f^{(0)} - \beta\left(z_k^{(0)} - c_k\right) = f^{(0)} - \frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}} d_{j_0}^{(0)}$$

egyenlőség teljesül. Most vizsgáljuk meg, hogy hogyan transzformálódik a $\underline{\delta}^{(0)}$ vektor többi komponense: minthogy

$$z_p^{(1)} = \sum_{i \in I_{B_1}} d_{ip}^{(1)} c_i = \sum_{\substack{i \in I_{B_0} \\ i \neq j}} \left( d_{ip}^{(0)} - \frac{d_{jp}^{(0)}}{d_{jk}^{(0)}} d_{ik}^{(0)} \right) c_i + \frac{d_{jp}^{(0)}}{d_{jk}^{(0)}} c_k =$$

$$= z_p^{(0)} - \frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}} d_{jp}^{(0)}, \qquad \forall p \in \{1, \ldots, n\}\text{-re},$$

ezért

$$z_p^{(1)} - c_p = z_p^{(0)} - c_p - \frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}} d_{jp}^{(0)}.$$

Az $f^{(0)}$ transzformálódását is figyelembe véve a

$$\underline{\delta}^{(1)} = \underline{\delta}^{(0)} - \frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}} \underline{\delta}_j^{(0)}$$

adódik.

Ha az előbbi transzformációs formulákat felhasználjuk, akkor a $B_0$ bázishoz tartozó szimplex tábla ismeretében könnyen meg lehet határozni a $B_1$ bázishoz tartozó szimplex táblát.

*I.5.2. Kiindulási megengedett bázismegoldás meghatározása.* A szimplex módszer elindításához egy kiindulási $B_0$ megengedett bázisra és a hozzá tartozó $\underline{x}_{B_0}$ megengedett bázismegoldásra van szükségünk.

Előfordulhat, hogy a feladattal együtt rendelkezésünkre áll egy ilyen bázis és a hozzá tartozó bázismegoldás, ilyenkor nincs akadálya a szimplex módszer alkalmazásának.

Adódhat olyan eset is, amikor csak egy megengedett megoldás adott a feladattal együtt. Ekkor ebből a megengedett megoldásból elindulva kellene meghatároznunk egy megengedett bázist és bázismegoldást. Most megmutatjuk, hogy mindezt hogyan lehet a gyakorlatban megvalósítani.

Legyen tehát az $\underline{x} \neq \underline{0}$ megengedett megoldása az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max\{\underline{c}^T \underline{x}\}$$

feladatnak, ahol $\underline{b} \neq \underline{0}$. (A $\underline{b} = \underline{0}$ esetben az $\underline{x} = \underline{0}$ mindig megengedett megoldása, sőt degenerált megengedett bázismegoldása a feladatnak)

Minthogy $\underline{x} \neq \underline{0}$, ezért legalább egy komponense pozitív. Az egyszerűség kedvéért legyenek az $x_1, x_2, \ldots, x_s$ az előbbi $\underline{x}$ pozitív komponensei. Ekkor az

$$x_1\underline{a}_1 + x_2\underline{a}_2 + \ldots + x_s\underline{a}_s = \underline{b}$$

egyenlőség teljesül. Ha az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_s$ vektorok lineárisan függetlenek, akkor az $\underline{x}$ egyben megengedett bázismegoldás is. Ha $s < r$, akkor ez az $\underline{x}$ egy degenerált bázismegoldás lesz. Ilyenkor az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_s$ vektorrendszert $r - s$ darab, az $A$ oszlopvektor rendszeréből származó vektorral ki kell egészíteni bázissá, ezt követően elindíthatjuk a szimplex módszert. Az $s = r$ esetben nincs szükségünk semmilyen egyéb lépésre, hiszen az adott $\underline{x}$ egyben megengedett bázismegoldás is. Ekkor az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_s$ képezi a megengedett bázist. Ha viszont az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_s$ vektorok lineárisan függőek, akkor léteznek olyan $v_1, \ldots, v_s$ nem mind 0 számok, hogy

$$v_1\underline{a}_1 + \ldots + v_s\underline{s} = \underline{0}.$$

Szükség esetén $(-1)$-gyel való szorzással elérhetjük, hogy a nem nulla $v_i$ számok között legyen pozitív. Ezen egyenlőségből fejezzük ki azt az $\underline{a}_j$ vektort, amelynek $j$ indexe olyan, hogy

$$\frac{x_i}{v_i} \geq \frac{x_j}{v_j}, \qquad v_j > 0$$

teljesül minden $v_i > 0$ esetén:

$$\underline{a}_j = \sum_{\substack{i=1\\ i \neq j}}^{s} \left(-\frac{v_i}{v_j}\underline{a}_i\right).$$

Ekkor

$$\underline{b} = \sum_{i=1}^{s} x_i\underline{a}_i = \sum_{\substack{i=1\\ i \neq j}}^{s} x_i\underline{a}_i + \sum_{\substack{i=1\\ i \neq j}}^{s} \left(-\frac{v_i}{v_j}x_j\underline{a}_i\right) = \sum_{\substack{i=1\\ i \neq j}}^{s} \left(x_i - \frac{v_i}{v_j}x_j\right)\underline{a}_i.$$

A konstrukció következménye, hogy

$$x_i - \frac{v_i}{v_j} x_j \geq 0, \qquad i \in \{1, 2, \ldots, s\} \setminus \{j\},$$

tehát származtattunk egy újabb megengedett megoldást, amelynek viszont legalább eggyel kevesebb nem nulla komponense van, mint amennyi a korábbinak volt. Ezt az eljárást addig folytatjuk, míg végül a $\underline{b}$ vektort lineárisan független vektorok nemnegatív együtthatókkal vett lineáris kombinációjaként elő nem állítjuk. Az eljárás végén kapott megengedett megoldás egyben megengedett bázismegoldás is lesz, ami a nem nulla komponenseihez tartozó lineárisan független $A$-beli oszlopvektorrendszer alkotta bázishoz tartozik. (Degenerált esetben ezt a vektorrendszert még ki kell egészíteni bázissá.) Ezek szerint tehát ha a feladattal együtt van egy megengedett megoldásunk is, akkor abból kiindulva meg tudunk határozni egy megengedett bázismegoldást is, amivel aztán elindíthatjuk a szimplex algoritmust.

Leggyakrabban azonban nem ismeretes a megadott feladatnak egyetlen megoldása sem. Minden olyan esetben, amikor a feladattal kapcsolatban semmilyen használható plusz információval sem rendelkezünk, az ún. kétfázisú szimplex módszert alkalmazhatjuk, amellyel leküzdhetjük a szimplex módszer beindításakor fellépő nehézségeket. A kétfázisú szimplex módszer első fázisa szolgál a kiinduló megengedett bázis meghatározására, a második fázisban pedig magát a kiindulási kiindulási feladatunkat adjuk meg.

Tekintsük tehát az

$$\begin{gathered} A\underline{x} = \underline{b} \\ \underline{x} \geq \underline{0} \\ \max\{\underline{c}^T \underline{x}\} \end{gathered}$$

feladatot, amelynél teljesüljön a $\underline{b} \geq \underline{0}$ nemnegativitási feltétel is. Ez utóbbi valójában nem jelent megszorítást, hiszen azon egyenletek mindkét oldalát megszorozhatjuk $(-1)$-gyel, amelyeknél a $\underline{b}$ komponense negatív. Ezen feladatnál annak ellenére, hogy a feltételrendszer egyenletrendszer, minden egyenlet baloldalához adjunk hozzá egy-egy nemnegatív változót. Jelöljék az új változókat $y_1, y_2, \ldots, y_m$. Ezekkel az ún. mesterséges változókkal a kiindulási feladatunk feltételrendszerét az

$$\begin{aligned} &(A, E)\begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} = \underline{b} \qquad (\underline{b} \geq \underline{0}) \\ &\underline{x} \geq \underline{0}, \quad \underline{y} \geq \underline{0} \qquad (E \text{ egy } m \times m\text{-es egységmátrix}) \end{aligned}$$

formájúvá alakíthatjuk. Minthogy az $(A, E)$ mátrix rangja $E$ miatt mindig $m$, és minthogy $\underline{b} \geq \underline{0}$, ezen feltételrendszer esetében azonnal meghatározhatunk egy kiinduló megengedett bázist és bázismegoldást. Legyen ugyanis $B_0 = E$ a kiinduló bázis, $\underline{x} = \underline{0}$, $\underline{y} = \underline{b}$ pedig a hozzá tartozó bázismegoldás, ami nyilvánvalóan teljesíti a nemnegativitást is. Ezek után az első fázisban olyan feladatot kell megfogalmaznunk, amelynek megoldásával valóban megkapjuk az eredeti problémánk egy megengedett bázisát, bázismegoldását (amennyiben megoldható az eredeti feladat).

Az első fázisban arra kell törekednünk, hogy lehetőség szerint minden $y_i$ mesterséges változó értéke 0 legyen, hiszen akkor a megoldás $\underline{x}$ része eleget tesz majd az eredeti feladat feltételrendszerének. Ezt elérhetjük, ha első fázisbeli feladatnak az

$$\begin{aligned} &(A, E)\begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} = \underline{b} \qquad (\underline{b} \geq \underline{0}) \\ &\underline{x} \geq \underline{0}, \quad \underline{y} \geq \underline{0} \\ &\max\left\{ -\sum_{i=1}^{m} y_i \right\} \end{aligned}$$

feladatot tekintjük. Ha a kiindulási feladatunknak létezik megengedett megoldása, akkor a $-\sum_{i=1}^{m} y_i$ célfüggvény optimum értéke 0. Amennyiben az első fázisbeli feladat optimum értéke kisebb, mint 0, akkor ez azt jelenti, hogy az eredeti problémánk feltételrendszere ellentmondásos, így azt tovább már nem is kell vizsgálnunk.

Ha az első fázis végén az optimum értéke 0, akkor még két eset fordulhat elő. Az egyik eset az, amikor az első fázis végén adódó optimális bázisban csak $A$ mátrix-beli oszlopvektorok vannak. Ilyenkor megtaláltuk a kiindulási feladatunk egy megengedett bázisát és bázismegoldását, és ezt követően folytathatjuk az algoritmusunkat a második fázissal, azaz az eredeti problémánk megoldásával. Ha viszont az első fázis végén adódó optimális bázisban még szerepelnek ún. mesterséges vektorok (azaz a mesterséges változókhoz tartozó egységvektorok) is, akkor az optimum érték 0 volta miatt az azokhoz tartozó báziskomponensek csak 0 értékűek lehetnek. (Ha ugyanis csak egyetlen $y_i$ értéke is nagyobb lenne 0-nál, akkor a $-\sum_{i=1}^{m} y_i < 0$ teljesülne.) Ez esetben próbáljuk meg kicserélni az optimális bázisban szereplő mesterséges vektorokat az $A$ mátrix nem bázisbeli oszlopvektoraival úgy, hogy a cserék után is bázist kapjunk. (Az ilyen cserék során egy 0 együtthatójú bázisbeli vektort cserélünk ki egy 0 együtthatójú vektorral, ezért maga a megoldás nem változik, csak a bázis.) Ezt a cserét addig folytatjuk, amíg minden kicserélhető mesterséges vektort ki nem cseréltünk. Ha ezek után a mesterséges vektorok a bázisból elfogytak, akkor megkaptuk az eredeti problémánk egy kiinduló megengedett bázisát, bázismegoldását. Ha viszont maradtak kicserélhetetlen mesterséges vektorok is az utolsó bázisban, akkor számunkra ez azt jelenti, hogy az $A$ mátrix rangja $m$-nél kisebb. Minthogy azonban a feladatunk feltételrendszere nem ellentmondásos (az első fázis végén 0 adódott optimum értéknek, így az optimális megoldásban szereplő $\underline{x}$ eleget tesz az eredeti feladat feltételrendszerének), ezért az ilyen esetekben is meghatározhatjuk az eredeti problémának egy kiinduló megengedett bázisát, bázismegoldását. Tekintsük ugyanis a cserék utáni optimális bázist, bázismegoldást. Az ebben a bázisban szereplő $A$ mátrix-beli oszlopvektorok megengedett bázisát alkotják az eredeti problémának, az optimális megoldás $\underline{x}$ része pedig az ahhoz tartozó bázismegoldás lesz. Így tehát ebben az esetben is folytathatjuk a feladatmegoldást a második fázissal.

*I.5.3. Ciklizálás, lexikografikus szimplex módszer.* A szimplex módszer alkalmazásakor előfordulhat, hogy visszatérünk egy olyan bázishoz, amelyet az algoritmus során már korábban is használtunk. Ha ilyen helyzet előáll, akkor az algoritmus végtelen ciklusba kerül, s így sohasem ér véget a feladat megoldása. Mindez csak degenerált esetben fordulhat elő, akkor sem kötelezően. Degenerált esetben lesznek olyan bázisbeli vektorok, amelyekhez tartozó báziskomponensek értéke 0. Ha egy ilyen vektort kicserélünk a bázistranszformáció során, akkor az új bázishoz tartozó bázismegoldás megegyezik a régivel, mindösszesen csak a bázis változott meg. Többször egymás után alkalmazva ilyen cserét előfordulhat, hogy visszajutunk ugyanahhoz a bázishoz. Nemdegenerált esetben a transzformáció utáni célfüggvényérték mindig nagyobb, mint a transzformáció előtti, így ugyanahhoz a bázishoz sohasem juthatunk viszsza.

A ciklizálás a gyakorlatban a feladatok nagy mérete miatt ritkán fordul elő. Ugyanakkor pontosan a feladat nagy mérete miatt nem célszerű olyan módszert alkalmazni, amelyik nem biztosítja a végtelen ciklusba kerülés kizárását. Ezt a problémát az ún. lexikografikus szimplex módszer használatával oldhatjuk meg.

A szimplex módszer alkalmazásakor a transzformációs lépésnél szabadon választhattuk ki a bázisba beviendő vektort azok közül, amelyek $z_k - c_k < 0$ értékhez tartoznak. Amikor viszont már eldöntöttük, hogy melyik $\underline{a}_k$ vektort visszük be a bázisba, akkor a $\min_{i:d_{ik}>0} \left\{ \frac{x_i}{d_{ik}} \right\}$ feltétellel határozhattuk meg az elhagyandó vektor indexét. Ha ez a minimum feltétel egyetlen $i$ esetén adódik, akkor a bázisból kikerülő vektort egyértelműen határoztuk meg, ilyenkor csak végre kell hajtani a transzformációt. Degenerált esetben viszont előfordulhat, hogy a minimum érték több $i$ esetén adódik. Ilyenkor az elhagyható vektorok közül szabadon dönthetünk, hogy melyik kerüljön ki a bázisból. Ez az a lépés, amikor olyan elvet állapíthatunk meg az elhagyandó vektor indexének meghatározására, amely aztán végtelen ciklust eredményez. A lexikografikus szimplex módszer alkalmazásával érhető el, hogy a bázisból elhagyandó vektor indexét mindig egyértelműen tudjuk meghatározni, így a ciklizálás elkerülhetővé válik.

A módszer ismertetéséhez szükségünk lesz a következő definíciókra.

**Definíció.** *Egy $\underline{x} \neq \underline{0}$ vektort lexikografikusan pozitívnak mondjuk, ha az első nem zéró komponense pozitív. Jelölése: $\underline{x} \succ \underline{0}$.*

**Definíció.** *Az $\underline{x}_1$ vektort $\underline{x}_2$-nél lexikografikus értelemben nagyobbnak nevezzük, ha $\underline{x}_1 - \underline{x}_2 \succ \underline{0}$, amit $\underline{x}_1 \succ \underline{x}_2$-vel is jelölhetünk.*

**Definíció.** *A $B_0$ bázishoz tartozó szimplex táblát lexikografikusan pozitívnak mondjuk, ha a bázisbeli vektorokhoz tartozó sorai, mint vektorok, lexikografikusan pozitívak, azaz $\underline{\delta}_{i_j}^{(0)} \succ \underline{0}$ minden $i_j \in I_{B_0}$ esetén.*

Lexikografikusan pozitív például egy $B_0$ nemdegenerált bázishoz tartozó szimplex táblázat. Természetesen degenerált bázis esetén is elérhetjük – az $A$ mátrix oszlopvektorainak átrendezésével –, hogy lexikografikusan pozitív legyen a szimplex táblázatunk. (Például az $A$ mátrixban előre hozzuk a bázisbeli vektorokat, így ha egy adott sor még ha 0-val kezdődik is, az első nem zéró komponense 1 lesz, vagyis ez a sorvektor is lexikografikusan pozitív lesz.)

Jelöljük most a $B_0$ bázishoz tartozó szimplex táblát $D_0$-lal, amely legyen lexikografikusan pozitív, és legyen olyan $k$ index, hogy

$$\begin{aligned} z_k^{(0)} - c_k < 0, &\quad \text{valamint} \\ d_{ik}^{(0)} > 0 &\quad \text{valamely } i \in I_{B_0} \text{ esetén.} \end{aligned}$$

Tekintsük az $\frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(0)}$ sorvektorokat azon $i$-kre, amelyeknél $d_{ik}^{(0)} > 0$. Minthogy a $D_0$ sorvektorai lineárisan függetlenek, ezért az $\frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(0)}$ sorvektorok mind különbözőek lesznek. Ebből viszont az következik, hogy ezen vektorok között pontosan egy olyan vektor van, amely lexikografikus értelemben minimális. Legyen $j$ az az index, amelyre a lexikografikus minimum megvalósul. Ekkor az $\underline{a}_j$ vektort kell elhagynunk a $B_0$ bázisból és helyébe kell az $\underline{a}_k$ vektort behoznunk a $B_1$ bázisba. Érvényes a következő tétel:

ha a $B_0$ bázishoz tartozó $D_0$ szimplex tábla lexikografikusan pozitív és a $j$ indexet az előbbiek szerint választjuk meg, akkor a $B_1$ bázishoz tartozó $D_1$ szimplex tábla is lexikografikusan pozitív lesz.

A transzformációs formulák alapján ugyanis érvényesek a következő összefüggések:

$$\begin{aligned} \underline{\delta}_i^{(1)} &= \underline{\delta}_i^{(0)} - \frac{d_{ik}^{(0)}}{d_{jk}^{(0)}}\underline{\delta}_j^{(0)} \qquad \forall i \in I_{B_0},\ i \neq j\text{-re} \\ \underline{\delta}_k^{(1)} &= \frac{1}{d_{jk}^{(0)}}\underline{\delta}_j^{(0)}. \end{aligned}$$

Tudjuk továbbá, hogy minden $\underline{\delta}_i^{(0)} \succ \underline{0}$, ahol $i \in I_{B_0}$. A $\underline{\delta}_k^{(1)}$ nyilvánvalóan lexikografikusan pozitív, hiszen a $\underline{\delta}_j^{(0)}$ lexikografikusan pozitív volt, és a $d_{jk}^{(0)} > 0$ is teljesül. Ha $d_{ik}^{(0)} \leq 0$, akkor – hasonló meggondolással – a $\underline{\delta}_i^{(1)}$

vektor is lexikografikusan pozitív. Be kell még bizonyítanunk, hogy olyan $i \in I_{B_0}$ esetén, amelyhez $d_{ik}^{(0)} > 0$ tartozik, a $\underline{\delta}_i^{(1)} \succ \underline{0}$ teljesül, azaz a $\underline{\delta}_i^{(1)}$ vektor lexikografikusan pozitív lesz.

Legyen a $j$ index olyan, hogy a lexikografikus értelemben vett minimum az $\frac{1}{d_{jk}^{(0)}}\underline{\delta}_j^{(0)}$ vektorra teljesüljön az előbb említett

$$\frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(0)}, \qquad d_{ik}^{(0)} > 0$$

vektorok között. Emiatt az

$$\frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(1)} = \frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(0)} - \frac{1}{d_{jk}^{(0)}}\underline{\delta}_j^{(0)} \qquad \forall d_{ik}^{(0)} > 0$$

vektorok lexikografikusan pozitívak. Minthogy egy vektor akkor és csak akkor lexikografikusan pozitív, ha pozitív konstansszorosa is az, ezért

$$d_{ik}^{(0)}\left(\frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(1)}\right) = \underline{\delta}_i^{(1)} \succ \underline{0}, \qquad \forall d_{ik}^{(0)} > 0\text{-ra.}$$

Beláttuk tehát, hogy ha az elhagyandó vektor indexét a lexikografikus értelemben vett minimumnak megfelelően választjuk, akkor a transzformációs formulákat alkalmazva lexikografikusan pozitív szimplex táblázatból lexikografikusan pozitív szimplex táblázathoz jutunk. Be kell azonban még azt is látnunk, hogy ezzel a választással valóban elkerülhető lesz a ciklizálás.

**Tétel.** *A lexikografikus szimplex módszert alkalmazva eljárásunk véges sok lépésben véget ér a szimplex módszernél említett a), vagy b) eset valamelyikénél. Ezen túlmenően a lexikografikus szimplex módszernél egyetlen bázis sem térhet vissza, és az utolsó sorok lexikografikus értelemben szigorúan monoton növekvő sorozatot alkotnak, azaz*

$$\underline{\delta}^{(0)} \prec \underline{\delta}^{(1)} \prec \underline{\delta}^{(2)} \prec \cdots$$

*Bizonyítás.* Tulajdonképpen csak a $\underline{\delta}^{(0)} \prec \underline{\delta}^{(1)} \prec \underline{\delta}^{(2)} \prec \cdots$ relációt kell bizonyítanunk, mert ebből következik, hogy egyetlen felhasznált bázis sem térhet vissza, amiből pedig az következik, hogy véges sok lépésben vagy az a), vagy a b) esethez kell eljutnunk. Az előbbi relációsorozatból pedig elegendő belátnunk a $\underline{\delta}^{(0)} \prec \underline{\delta}^{(1)}$ relációt.

A transzformációs formula alapján

$$\underline{\delta}^{(1)} = \underline{\delta}^{(0)} - \frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}}\underline{\delta}_j^{(0)},$$

amiből

$$\underline{\delta}^{(1)} - \underline{\delta}^{(0)} = -\frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}}\underline{\delta}_j^{(0)} \succ \underline{0}$$

adódik. □

A lexikografikus szimplex módszer csak az elhagyandó vektor indexének meghatározásában tér el az alap szimlex módszertől, ebben a lépésben is csak akkor, ha degenerált esettel van dolgunk. Nemdegenerált esetben a $D_0$ minden sorának már az első komponense pozitív $(x_i > 0,\ i \in I_{B_0})$, így az $\frac{1}{d_{ik}^{(0)}}\underline{\delta}_i^{(0)},\ i \in I_{B_0},\ d_{ik}^{(0)} > 0$, lexikografikus értelemben vett minimuma ugyanott adódik, mint ahol a $\min_{i:d_{ik}>0}\left\{\frac{x_i}{d_{ik}}\right\}$. Ez különben elvárható volt egy "jó algoritmustól", hiszen egy nagyméretű feladatról nehéz lenne előre megállapítani, hogy az a degenerált esetnek megfelelő-e, ezért minden ilyen feladatra ezt a módszert alkalmazzuk tudva azt, hogy ha a feladat nem degenerált, akkor ez az algoritmus semmiben sem különbözik az alap szimplex módszertől. (Nem kell tehát például attól félnünk, hogy csak azért, mert nem az alapalgoritmust használtuk, nemdegenerált esetben megnövekszik például a program futási ideje.)

*I.5.4. Az optimalitás feltétele.* A szimplex módszer tárgyalásakor az a) esetben bebizonyítottuk, hogy ha a $B_0$ bázishoz tartozó szimplex táblázatban $z_k - c_k \geq 0,\ k = 1, 2, \ldots, n$, akkor a $B_0$ optimális bázis. Ez tehát azt jelenti, hogy a $z_k - c_k \geq 0,\ k = 1, 2, \ldots, n$ feltétel az optimalitásnak elegendő feltétele. Nyilvánvalóan felmerül az a kérdés, hogy ez a feltétel mikor lesz szükséges feltétel is, illetve, hogy véges sok lépésben elérhetjük-e, hogy $z_k - c_k \geq 0$ legyen minden $k \in \{1, \ldots, n\}$ esetén. Erre a választ a következő tételekben adhatjuk meg.

Feltesszük, hogy az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max\{\underline{c}^T\underline{x}\}$$

feladat lehetséges, azaz van megoldása az $A\underline{x} = \underline{b}, \underline{x} \geq \underline{0}$ rendszernek, ellenkező esetben ugyanis nincs mit vizsgálnunk.

**Tétel.** *Ha az*

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max\{\underline{c}^T\underline{x}\}$$

*lineáris programozási feladatnak van véges optimuma, akkor mindig van olyan bázismegoldás, amely optimális és van olyan $B$ bázis, hogy a*

$$z_k - c_k \geq 0, \qquad k = 1, 2, \ldots, n$$

*egyenlőtlenségek is teljesülnek. (Egy ilyen $B$ bázist optimális bázisnak, a hozzá tartozó bázismegoldást optimális megoldásnak nevezünk.)*

*Bizonyítás.* Minthogy van megengedett megoldás, van megengedett bázismegoldás is. Ezen bázishoz tartozó szimplex táblát lexikografikusan pozitívvá téve és a lexikografikus szimplex módszert alkalmazva véges sok lépés után a szimplex módszernél ismertetett a) esethez jutunk, azaz minden $k$-ra a $z_k - c_k \geq 0$ fog teljesülni. (A b) esetet a véges optimum létezésének feltételezésével kizártuk.)

A $z_k - c_k \geq 0,\ k = 1, 2, \ldots, n$ azonban nem minden optimális bázis esetén teljesül, mint ahogy ezt a következő példából is láthatjuk, Legyen a feladat az

$$x_1 + x_2 = 0$$
$$x_1 \geq 0, x_2 \geq 0$$
$$\max\{x_1 + 2x_2\}$$

lineáris programozási feladat. A megengedett megoldások halmaza most a $\underline{0}$ vektorból áll. Ennél a feladatnál az $\underline{a}_1 = (1)$, illetve az $\underline{a}_2 = (1)$ vektorok külön-külön megengedett bázisok. Az első bázis esetén

$$z_1 - c_1 = 0$$
$$z_2 - c_2 = -1 < 0$$

a másodiknál pedig

$$z_1 - c_1 = 1 > 0$$
$$z_2 - c_2 = 0.$$

□

A következő tétel azt állítja, hogy ilyen csak degenerált bázis esetén fordulhat elő, vagyis degenerált bázis esetén a $z_k - c_k \geq 0,\ k = 1, 2, \dots, n$ feltétel csak elégséges, de nem szükséges feltétele az optimalitásnak.

**Tétel.** *Ha $B$ megengedett és nemdegenerált bázis, akkor annak, hogy $B$ optimális bázis legyen, szükséges és elégséges feltétele a $z_k - c_k \geq 0,\ k = 1, 2, \dots, n$ teljesülése.*

*Bizonyítás.* Minthogy a szimplex módszer tárgyalásánál már bebizonyítottuk, hogy a $z_k - c_k \geq 0,\ k = 1, 2, \dots, n$ elégséges az optimalizáláshoz, ezért már csak a szükségességet kell bebizonyítanunk. Ehhez megmutatjuk, hogy az $\underline{x}$ nem optimális, ha a $z_k - c_k \geq 0,\ k = 1, 2, \dots, n$ nem teljesül. Legyen $k$ olyan index, hogy a $z_k - c_k < 0$ teljesül. Ekkor vagy van olyan $i \in I_B$, hogy $d_{ik} > 0$, vagy nincs. Az utóbbi esetben a célfüggvény nem korlátos felülről, az előbbi esetben pedig $\beta = \min_{i:d_{ik}>0} \left\{ \frac{x_i}{d_{ik}} \right\} > 0$, tehát a célfüggvény értéke megnövelhető az $\underline{a}_k$ vektor bevonásával, vagyis ekkor a $B$ nem optimális bázis, az $\underline{x}_B$ nem optimális megoldás. A lexikografikus szimplex módszert alkalmazva véges sok lépésben az algoritmus – ha feltételezzük, hogy van véges optimum – csak olyan $B$ bázissal fejeződhet be, amelyhez $z_k - c_k \geq 0,\ k = 1, 2, \dots, n$ tartozik. □

## I.6. A szimplex módszer variánsai

A bemutatott szimplex módszert az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max\{\underline{c}^T \underline{x}\}$$

formájú lineáris programozási feladat megoldására használhattuk. Ugyancsak ilyen formájú feladattal kapcsolatban tárgyaltuk a lexikografikus szimplex módszert is, amelyet a ciklizálás elkerülésére dolgoztak ki. Ezt már felfoghatjuk úgy, mint a szimplex módszer egyik olyan variánsát, amely a bázisból elhagyandó vektor indexének megállapításában tér el a közönséges szimplex módszertől.

Ha a feladatunk nem az előbbi formában van megadva, akkor a már tanult ekvivalens átalakításokkal előbb ilyen alakra hozzuk, majd alkalmazzuk a tanult módszert. Mint említettük, az ilyen alakra való hozás dimenzióban eltérő méretű feladatokra vezethet, s bár azt meg tudjuk oldani az alap szimplex módszerrel, a megoldás lehet lényegesen hosszabb időt igénylő is, mint amennyit az az eljárás igényel, amelyhez nem szükséges az átalakítás.

Egyszerű eset az, amikor az alapproblémánk

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\min\{\underline{c}^T \underline{x}\}$$

formájú. Az eddig bemutatott szimplex módszerrel ezt a feladatot úgy oldhattuk meg, hogy tekintettük helyette a vele ekvivalens

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max\{-\underline{c}^T \underline{x}\}$$

feladatot, amelyre aztán alkalmaztuk a szimplex módszert. Most megvizsgáljuk hogyan módosul a szimplex módszer, ha azt közvetlenül a minimum feladat megoldására akarjuk használni.

A korábbiaknak megfelelően, ugyanolyan jelöléseket használva készítsük el a kiinduló szimplex táblát. A következő tétel kimondja hogyan kell közvetlenül minimum feladatra használnunk a szimplex módszert.

**Tétel.**

a) *Ha minden* $k \in \{1, 2, \dots, n\}$ *esetén a* $z_k - c_k \leq 0$ *teljesül, akkor a* $B$ *bázishoz tartozó* $\underline{x}_B$ *bázismegoldás optimális megoldása a minimum problémának.*

b) *Ha létezik olyan* $k \in \{1, 2, \dots, n\}$, *hogy* $z_k - c_k > 0$, *és egy ilyen* $k$ *esetén minden* $i \in I_B$*-re* $d_{ik} \leq 0$, *akkor a célfüggvény nem korlátos alulról.*

c) *Ha létezik olyan* $k \in \{1, 2, \dots, n\}$, *hogy* $z_k - c_k > 0$, *és minden ilyen* $k$*-ra létezik olyan* $i \in I_B$, *hogy* $d_{ik} > 0$, *akkor az* $\underline{a}_k$ *vektort bevonva az új bázisba azon* $\underline{a}_j$ *vektor helyébe, amelynek az indexére az*

$$\frac{x_j}{d_{jk}} = \min_{\substack{i \,:\, i \in I_B \\ d_{ik} > 0}} \left\{ \frac{x_i}{d_{ik}} \right\}$$

*teljesül, akkor a célfüggvény értéke nem növekszik, és az új bázis is megengedett lesz.*

Ezen tétel bizonyítása alapvetően ugyanúgy történik, mint a maximum feladat esetében, ezért azt az olvasóra bízzuk.

Az alap és a mostani, a minimum feladat megoldására szolgáló szimplex algoritmusok ún. direkt algoritmusok. Ezeknél a kiindulási $B_0$ bázishoz tartozó szimplex táblából direkt transzformáció útján jutunk el a $B_1$ bázishoz tartozó szimplex táblához. Az explicit inverz algoritmus során – amelyet most bemutatunk – minden egyes transzformációs lépésben explicite meghatározzuk a bázis inverzét, és ennek felhasználásával származtatjuk a szimplex táblát a

$$\underline{d}_k^{(0)} = B_0^{-1}\underline{a}_k \qquad k \in \{0, 1, 2, \dots, n\}$$

$$\underline{a}_0 = \underline{b} \qquad \underline{d}_0^{(0)} = \underline{x}_{B_0}$$

$$z_k^{(0)} = \underline{c}_{B_0}^T \underline{d}_k^{(0)} = \underline{c}_{B_0}^T B_0^{-1} \underline{a}_k$$

formulák alapján. Tekintsük az

$$A\underline{x} = \underline{b}$$

$$\underline{x} \geq \underline{0}$$

$$\max\{\underline{c}^T \underline{x}\}$$

lineáris programozási feladatot, ahol $A$ $m \times n$-es $(m \leq n)$ mátrixról az általánosság megszorítása nélkül feltesszük, hogy a rangja megegyezik a sorai számával, azaz $m$-mel. Legyen $B_0$ egy kiinduló megengedett bázis. Ezen módszer során előbb meghatározzuk a $B_0$ bázis mátrixának inverzét, a $B_0^{-1}$-et, majd pedig kiszámítjuk a $\underline{c}_{B_0}^T B_0^{-1}$ vektort. Ez utóbbi felhasználásával minden $k \notin I_{B_0}$ esetén kiszámítjuk a $z_k^{(0)} - c_k$ értéket:

$$z_k^{(0)} - c_k = \left(\underline{c}_{B_0}^T B_0^{-1}\right) \underline{a}_k - c_k, \qquad k \notin I_{B_0}.$$

Ezek után megvizsgáljuk a $z_k^{(0)} - c_k$ értékek előjelét, majd döntünk afelől, hogy kell-e transzformációt végrehajtanunk, vagy pedig már abba kell hagyni az eljárást vagy amiatt, mert nincs megoldása a feladatnak, vagy mert már optimális megoldásnál vagyunk.

Tegyük fel, hogy a $k$ index esetén $z_k^{(0)} - c_k < 0$ és tegyük fel, hogy az $\underline{a}_k$ vektor bevonható a $B_1$ bázisba. Ahhoz, hogy megállapítsuk melyik vektort kell elhagynunk a $B_0$ bázisból, meg kell határoznunk a $\underline{d}_k^{(0)} = B_0^{-1}\underline{a}_k$ vektort, majd pedig az $\underline{x}_{B_0} = B_0^{-1}\underline{b}$, és ezen $\underline{d}_k^{(0)}$ vektor olyan komponensei hányadosának a minimumát, ahol $d_{ik}^{(0)} > 0$, $i \in I_{B_0}$. Ha döntöttünk afelől, hogy melyik $A$-beli vektort hagyjuk el a $B_0$ bázisból, akkor az eljárást elölről kezdjük, azaz meghatározzuk a $B_1$ bázis inverzét, stb.

Ezen módszernél az egyszerűsítés a direkt algoritmushoz képest abban áll, hogy a teljes szimplex tábla helyett annak csak egy részével dolgozunk, minthogy csak a bázisba bevonandó $\underline{a}_k$ vektor $k$ indexéhez számítjuk ki a $\underline{d}_k^{(0)}$ vektort. Így persze előfordulhat, hogy több transzformációt feleslegesen hajtunk végre, mert bár nincs véges optimum, de ezt – minthogy nem minden $z_k^{(0)} - c_k < 0$ értékhez számítjuk ki a $\underline{d}_k^{(0)}$ vektort – nem állapítottuk meg időben. (Természetesen eljárhatunk úgy is, hogy egy adott transzformációs lépés során minden $z_k^{(0)} - c_k < 0$ esetén kiszámítjuk a megfelelő $\underline{d}_k^{(0)}$ vektort, és csak akkor hajtjuk végre az újabb transzformációt, ha ezen $\underline{d}_k^{(0)}$ vektorok mindegyikének van pozitív komponense. Azonban ekkor is csak egy $\underline{d}_k^{(0)}$ vektort tüntetünk fel a táblában.) Ezt az algoritmust szokás módosított explicit inverz algoritmusnak nevezni, amelyhez a következő ún. módosított szimplex táblát kell elkészíteni:

| | $C_{B_0}^T B_0^{-1}$ | $f$ | $z_k^{(0)} - c_k$ |
|---|---|---|---|
| $x_{i_1}$ | | $x_{i_1}$ | $d_{i_1k}^{(0)}$ |
| $x_{i_2}$ | | $x_{i_2}$ | $d_{i_2k}^{(0)}$ |
| $\vdots$ | $B_0^{-1}$ | $\vdots$ | $\vdots$ |
| $x_{i_m}$ | | $x_{i_m}$ | $d_{i_mk}^{(0)}$ |

E táblában a bal oldalon álló oszlop szimbolikusan, az utolsó előtti oszlop pedig számszerűleg tartalmazza a bázisváltozókat.

Ez esetben is felmerül az a kérdés, hogy a $B_0$ bázishoz tartozó módosított szimplex tábla felhasználásával nyerhetjük-e a $B_1$ bázishoz tartozó módosított szimplex táblát? A válasz igen, ehhez azonban a $B_0$ mátrix helyett a

$$\hat{B}_0 = \begin{pmatrix} 1 & -\underline{c}_{B_0}^T \\ \underline{0} & B_0 \end{pmatrix}$$

$(m+1) \times (m+1)$-es mátrixot kell tekintenünk, és ennek kell képeznünk az inverzét:

$$\hat{B}_0^{-1} = \begin{pmatrix} 1 & -\underline{c}_{B_0}^T B_0^{-1} \\ \underline{0} & B_0^{-1} \end{pmatrix}.$$

Ez utóbbi mátrix felhasználásával a módosított szimplex tábla utolsó két oszlopa a következőképpen származtatható:

$$\hat{B}_0^{-1} \begin{pmatrix} -c_k \\ \underline{a}_k \end{pmatrix} = \begin{pmatrix} 1 & -\underline{c}_{B_0}^T B_0^{-1} \\ \underline{0} & B_0^{-1} \end{pmatrix} \begin{pmatrix} -c_k \\ \underline{a}_k \end{pmatrix} = \begin{pmatrix} -c_k + \underline{c}_{B_0}^T B_0^{-1} \underline{a}_k \\ B_0^{-1} \underline{a}_k \end{pmatrix} =$$

$$= \begin{pmatrix} z_k^{(0)} - c_k \\ \underline{d}_k \end{pmatrix}$$

$$\hat{B}_0^{-1} \begin{pmatrix} 0 \\ \underline{b} \end{pmatrix} = \begin{pmatrix} 1 & -\underline{c}_{B_0}^T B_0^{-1} \\ \underline{0} & B_0^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ \underline{b} \end{pmatrix} = \begin{pmatrix} \underline{c}_{B_0}^T B_0^{-1} \underline{b} \\ B_0^{-1} \underline{b} \end{pmatrix} = \begin{pmatrix} f \\ \underline{x}_{B_0} \end{pmatrix}$$

Mielőtt azonban ezen mennyiségeket kiszámolnánk, meg kell határozni a $\hat{B}_0^{-1}$ első sorának a skaláris szorzatait az $A$ mátrix (nem bázisbeli) oszlopvektoraival, mert az első lépés a $z_k^{(0)} - c_k$ különbségek meghatározása a bázisba bevonandó vektor meghatározása érdekében.

Ha a $B_0$ bázisról áttérünk a $B_1$ bázisra, akkor a $B_1^{-1}$ helyett célszerű a $\hat{B}_0^{-1}$ felhasználásával a $\hat{B}_1^{-1}$-et meghatározni. Ha a bázisba az $\underline{a}_k$ vektort visszük be, és $\underline{a}_j$-t hagyjuk el, akkor

$$\hat{B}_1^{-1} = \tilde{F} \cdot \hat{B}_0^{-1},$$

ahol

$$\tilde{F} = (\underline{e}_1, \ldots, \underline{e}_j, \underline{g}, \underline{e}_{j+2}, \ldots, \underline{e}_{m+1})$$

$$\underline{g}^T = \left( -\frac{z_k^{(0)} - c_k}{d_{jk}^{(0)}}, -\frac{d_{1k}^{(0)}}{d_{jk}^{(0)}}, \ldots, -\frac{d_{j-1,k}^{(0)}}{d_{jk}^{(0)}}, \frac{1}{d_{jk}^{(0)}}, -\frac{d_{j+1,k}^{(0)}}{d_{jk}^{(0)}}, \ldots, -\frac{d_{mk}^{(0)}}{d_{jk}^{(0)}} \right).$$

A felsorolt variánsokon kívül számtalan változat létezik nevesítve (pl. duál szimplex módszer, primál-duál szimplex módszer, stb.) illetve nevesítetlenül. Ezek között szép számmal találhatók olyan algoritmusok is, amelyek konkrét lineáris programozási probléma megoldására szolgálnak. A számítógépek megjelenésekor a drága gépidő miatt igyekeztek betartani azt az akkori „elvet", hogy inkább a matematikus, a programozó dolgozzon többet és hozzon létre olyan algoritmust, amely a lehető legjobban „illeszkedik" az adott feladathoz, azaz abból a lehető legfőbb információt hasznosítja, minthogy a számítógép egy perccel is többet dolgozzon. Ma már a számítógépek nagy száma miatt ez nem elsődleges cél, viszont a számítógéppel feldolgozható feladatok méretének növekedése ma is arra ösztönzi a matematikusokat, a programozókat, hogy egyre hatékonyabban működő variánsokat dolgozzanak ki, hogy a meglehetősen nagy méretű feladatok is elfogadható időn belül megoldhatók lehessenek.

A következő pontban a szállítási problémát tárgyaljuk, amelynek megoldási algoritmusa, az ún. „hurokszerkesztéses szimplex módszer" is egy szimplex módszer variánsnak tekinthető.

## I.7. Szállítási probléma

Legyen adott $m$ telephely, amelyeken bizonyos fajta, tetszés szerint osztható termékből $a_1, a_2, \ldots, a_m$ mennyiséget tárolnak. Adott továbbá $n$ felvevőhely, amelyek $b_1, b_2, \ldots, b_n$ mennyiséget igényelnek ebből a termékből. Egységnyi terméknek az $i$-edik telephelyről a $j$-edik felvevőhelyre való szállítási költsége $c_{ij}$-vel legyen jelölve. Jelölje továbbá $x_{ij}$ az $i$-edik telephelyről a $j$-edik felvevőhelyre szállítandó – egyelőre ismeretlen – mennyiséget. $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$.

Feltesszük, hogy

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j,$$

azaz, hogy a tárolt áru összmennyissége megegyezik az igényelt áru összmennyiségével. Ez nem jelenti az általánosság megszorítását, hiszen vagy fiktív telephely, vagy fiktív felvevőhely beiktatásával mindig elérhető az előbbi egyenlőség. Minthogy pedig ezen fiktív telephely, vagy felvevőhely esetén a szállítási költségeket 0-nak tekintjük, az átalakított feladat az eredetivel ekvivalens is lesz. Ha pl. az összigény nagyobb, mint az össztárolt mennyiség $(\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j)$, akkor bevezetünk egy $(m+1)$-edik fiktív telephelyet, amelyen $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$ mennyiséget „tárolnak". Erről a telephelyről minden felvevő helyre 0 lesz a szállítás költsége. Ha pedig a $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ teljesül, akkor felveszünk egy $(n+1)$-edik fiktív fogyasztót, ahová 0 lesz a szállítás költsége minden egyes telephelyről, és amelynek $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$ lesz az „igénye".

Ezek után a feladatot úgy fogalmazhatjuk meg, hogy olyan szállítást kell megvalósítanunk, amelynek során minden telephelyről minden árut elszállítanak, az egyes felvevőhelyek igényeit kielégítik, és ezt mind úgy teszik, hogy az össz-szállítási költség minimális.

A szállítási problémát matematikailag a következőképpen fogalmazhatjuk meg: Legyen adott egy

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

$m \times n$-es mátrix, a költségmátrix. Legyenek továbbá adva az

$$a_1 \geq 0, \ldots, a_m \geq 0 \qquad \text{(tárolt)}$$

illetve

$$b_1 \geq 0, b_2 \geq 0, \ldots, b_n \geq 0 \qquad \text{(igényelt)}$$

mennyiségek, amelyekre a $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$ teljesül. Meghatározandók az olyan $x_{ij}$ mennyiségek, amelyek eleget tesznek a

$$\begin{aligned} \sum_{j=1}^{n} x_{ij} &= a_i, \qquad i = 1, \ldots, m \\ \sum_{i=1}^{m} x_{ij} &= b_j, \qquad j = 1, \ldots, n \\ x_{ij} &\geq 0, \qquad i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, n \end{aligned}$$

feltételeknek, s amelyekkel a

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

költségfüggvény felveszi a minimumát. Ezt a feladatot mátrixos alakban az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\min\{\underline{c}^T \underline{x}\}$$

formában írhatjuk fel, ahol az $A$ egy $(m+n) \times (m \cdot n)$-es mátrix, amelynek minden oszlopában csak 2-2 elem különbözik 0-tól, és azok is 1-gyel egyenlőek. (Az $A$ mátrix $\underline{a}_{ij}$-vel jelölt oszlopvektora olyan $m+n$ dimenziós vektor, amelynek $i$-edik és $m+j$-edik komponense 1, a többi 0).

Itt

$$\underline{b} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \qquad \underline{x} = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \\ x_{21} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{m1} \\ \vdots \\ x_{mn} \end{pmatrix}, \qquad \underline{c} = \begin{pmatrix} c_{11} \\ \vdots \\ c_{1n} \\ c_{21} \\ \vdots \\ c_{2n} \\ \vdots \\ c_{m1} \\ \vdots \\ c_{mn} \end{pmatrix}.$$

A szállítási probléma egy minimum lineáris programozási feladat, amit az előző pontban ismertetett minimum feladat megoldására vonatkozó szimplex módszer variánssal meg is oldhatnánk. Esetünkben azonban ez a módszer nem lenne hatékony, hiszen nem túl nagy $m$ és $n$ esetén is az $A$ mátrix mérete meglehetősen nagy, ugyanakkor sok 0 elemet is tartalmaz, amit az említett módszerrel nem tudunk kihasználni. Olyan szimplex módszer variánst („hurokszerkesztéses szimplex módszer") fogunk vizsgálni, amely nem igényli az $A$ mátrix oszlopvektorainak tárolását.

Először megmutatjuk, hogy bármely szállítási problémának van véges optimuma és optimális megoldása. Ez abból következik, hogy a szállítási probléma feltételeinek eleget tevő $\underline{x} \geq \underline{0}$ vektorok korlátos, zárt halmazt határoznak meg, amely sohasem üres. (Igaz ugyanis, hogy $0 \leq x_{ij} \leq \min\{a_i, b_j\}$, ahol $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$. Mindez a megoldáshalmaz korlátosságát jelenti. Az egyenlőtlenség megengedése biztosítja a megoldáshalmaz zártságát. Végül pedig az

$$x_{ij} = \frac{a_i b_j}{\sum_{k=1}^{m} a_k} \geq 0, \qquad i = 1, \ldots, m, j = 1, \ldots, n$$

mindig megoldása a szállítási problémának. Korlátos, zárt halmaz felett pedig a $\underline{c}^T \underline{x}$, amelyik folytonos, mindig felveszi a minimumát.)

Megoldási módszerünkhöz, a „hurokszerkesztéses szimplex módszer"-hez szükségünk lesz a cellagráfokkal kapcsolatos néhány definícióra és tételre, így a megoldási algoritmus ismertetése előtt ezeket adjuk meg.

A szállítási probléma megoldása során általában két $m \times n$-es táblázatot szokás kitölteni. Az egyikbe, az ún. szállítási táblába a $c_{ij}$ szállítási költségeket írják be, peremértékként pedig feltüntetik az $a_i$ és $b_j$ értékeket $(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ a következőképpen:

| | | | | |
|---|---|---|---|---|
| $c_{11}$ | $c_{12}$ | $\cdots$ | $c_{1n}$ | $a_1$ |
| $c_{21}$ | $c_{22}$ | $\cdots$ | $c_{2n}$ | $a_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $c_{m1}$ | $c_{m2}$ | $\cdots$ | $c_{mn}$ | $a_m$ |
| $b_1$ | $b_2$ | $\cdots$ | $b_n$ | |

A szállítási probléma akkor adott, ha adottak a szállítási tábla elemei a peremértékekkel együtt.

A másik $m \times n$-es táblázatban a vizsgált megoldásvektor komponensei kerülnek be; az $(i, j)$ pozicióba az $x_{ij}$ érték. $(i = 1, \ldots, m; j = 1, \ldots, n.)$

(Bizonyos esetekben a szállítási tábla minden egyes pozícióját két részre osztják, és az egyikben a szállítási költségeket, a másikban pedig a szállított mennyiségeket tüntetik fel. Ilyenkor természetesen egyetlen táblázatot használnak.)

Ha az $A$ mátrix oszlopvektorait rendre az

$$\underline{a}_{11}, \ldots, \underline{a}_{1n}, \underline{a}_{21}, \ldots, \underline{a}_{2n}, \ldots, \underline{a}_{m1}, \ldots, \underline{a}_{mn}$$

jelölik, akkor a szállítási tábla $(i, j)$ cellája és az $\underline{a}_{ij}$ vektor között egy kölcsönösen egyértelmű megfeleltetést létesíthetünk és a cellák geometriai elhelyezkedése, valamint a megfelelő oszlopvektorok lineáris függetlensége (függősége) között párhuzamot húzhatunk.

Tekintsünk most egy $m \times n$-es üres táblázatot, amelynek pozícióit szokás celláknak is nevezni. Az $m \cdot n$ számú cellából válasszunk ki tetszőleges számút és pozíciójút, majd kössük azokat össze az összes lehetséges vízszintes, illetve függőleges vonallal (átlósan nem). Az így kapott konfigurációt <u>cellagráfnak</u> nevezzük. A cellagráf ezen definíciójából következik, hogy az a cellarendszerrel egyértelműen van meghatározva. A cellagráfban az egy sorban, vagy egy oszlopban álló cellapárokat összekötő szakaszokat a <u>cellagráf éleinek</u>, a cellákat, amelyek az élek végpontjai, pedig a <u>cellagráf csúcspontjainak</u> nevezzük.

A cellagráfban az egymáshoz csatlakozó élek egy rendszerét <u>útnak</u> nevezzük, ha minden él minden végpontjából legfeljebb két él indul ki, és a szomszédos élek merőlegesek egymásra. Az út végein lévő csúcsokat az út végpontjainak nevezzük. Azt az utat, amely minden sorból és minden oszlopból legfeljebb két cellán megy át, <u>egyszerű útnak</u> nevezzük. Azt az utat, amelynél a végpontok egybeesnek, <u>huroknak</u> nevezzük, amely lehet egyszerű, illetve nem egyszerű attól függően, hogy az út egyszerű volt-e vagy sem.

Azt a cellagráfot, amely a szállítási tábla minden sorából és minden oszlopából tartalmaz cellát, és amelynek bármely két cellája összeköthető úttal, <u>összefüggőnek</u> nevezzük. Egy összefüggő és hurokmentes cellagráfot <u>fának</u> nevezünk.

A „hurokszerkesztéses szimplex módszer" megértése szempontjából fontosak a következő cellagráfokkal kapcsolatos tételek, amelyeket bizonyítás nélkül közlünk.

- „A cellagráf fa" illetve „A cellagráf $m + n - 1$ cellát tartalmaz és hurokmentes" állítások ekvivalensek.
- Ha tekintünk egy fa cellagráfot és azon kívül egy cellát, akkor az a cellagráf, amely tartalmazza az említett fát és cellát, egyben tartalmaz egyetlen olyan egyszerű hurkot is, amely áthalad az említett cellán.
- Ha a cellák egy sorozatában a szomszédosokat összekötő élek hurkot alkotnak, akkor az egyes celláknak megfelelő $A$-beli vektorok váltakozó előjelű összege $\underline{0}$.

- Egy $\{\underline{a}_{ij}\}$ vektorrendszer akkor és csak akkor lineárisan független, ha a vektoroknak megfelelő cellákhoz tartozó cellagráf hurokmentes.
- Egy $\{\underline{a}_{ij}\}$ vektorrendszer akkor és csakis akkor alkot bázist, ha a megfelelő cellagráf fa.

Ezek után tekintsük a szállítási problémát. Belátjuk, hogy ezen probléma $A$ mátrixának $m+n-1$ a rangja. Már tudjuk, hogy az $A$ mátrix $(m+n)\times(m\cdot n)$ tipusú, amelynél $m+n<m\cdot n$. Ha összeadjuk ezen $A$ mátrix első $m$, majd utolsó $n$ sorát, akkor két, csupa 1-es komponensekkel bíró sorvektorhoz jutunk. Ha ezt a két sorvektort kivonjuk egymásból, akkor tehát zérusvektort kapunk, ami azt bizonyítja, hogy az $A$ sorvektorai függő rendszert alkotnak, azaz rang$A<m+n$. Azt pedig, hogy az $A$ rangja legalább $m+n-1$ úgy láthatjuk be, hogy megmutatjuk: $A$-nak van nem 0 determinánsú $(m+n-1)\times(m+n-1)$-es méretű részmátrixa. Egyik ilyet úgy nyerhetjük, hogy elhagyjuk az $A$ mátrix utolsó sorát, majd tekintjük az első $n$, aztán a $(2n)$., a $(3n).,\ldots,(m\cdot n)$. oszlopát.

Mivel tehát az $A$ rangja $m+n-1$, az oszlopvektor rendszeréből kiválasztott bármely bázisba emiatt $m+n-1$ vektor tartozik.

Minthogy a szállítási probléma egy (minimum) lineáris programozási feladat, ezért a szimplex módszert fogjuk alkalmazni a megoldására. Nem fogjuk azonban elkészíteni a szokásos szimplex táblázatot, hiszen annak – $A$ mérete miatt – meglehetősen nagy lenne a mérete. Helyette két $m\times n$-es táblázatot fogunk mindösszesen használni: a szállítási költségeket tartalmazó szállítási és a megoldásvektort tartalmazó táblát. Ki fogjuk használni az $A$ mátrix speciális voltát is, így a korábbiakhoz viszonyítva sokkal egyszerűbben nyerhetjük a szimplex módszerhez szükséges információkat.

Mint minden szimplex módszernél, itt is szükségünk lesz egy kiinduló megengedett bázisra, bázismegoldásra. Ilyen megoldást előállíthatunk például az „észak-nyugati sarok" módszerrel. Léteznek más módszerek is, amelyek bizonyos esetekben „jobb" kiinduló megoldást szolgáltathatnak, mint az észak-nyugati sarok módszer, most azonban – az egyszerűség miatt – ez utóbbit mutatjuk be.

Tekintsük egy, az $a_i$ és $b_j$ $(i=1,2,\ldots,m; j=1,2,\ldots,n)$ peremértékekkel ellátott, egyébként üres $m\times n$ -es táblát, amibe majd a megoldásvektor komponenseit fogjuk beírni. Válasszuk ki a tábla bal felső sarkának megfelelő cellát, majd írjuk be oda az $a_1$ és $b_1$ peremértékek közül a kisebbiket. Ezt követően nullákkal töltsük fel a tábla első sorát, vagy első oszlopát attól függően, hogy az $a_1$ vagy $b_1$ volt a kisebb. (Ha $a_1=b_1$, akkor szabadon választhatunk, hogy az első sort, vagy első oszlopot töltjük fel nullákkal, azonban ezek közül csak az egyiket kell. Ilyen helyzet csak degenerált esetben fordulhat elő.) Végül a $\min\{a_1, b_1\}$ értéket vonjuk ki $a_1$-ből és $b_1$-ből is. A még üresen maradt táblarésszel és az előbbi módon módosított peremértékekkel hasonlóan járjunk el addig, amíg az $m \times n$-es tábla minden eleme értékkel nem bír. Belátható, hogy ekkor az $m \cdot n$ cellából pontosan $m+n-1$ cellát fogunk „kiválasztani”, a többibe pedig 0-t fogunk írni. Az ennek megfelelő cellagráf így $m + n - 1$ cellát tartalmaz és hurokmentes, azaz fa. A kiválasztott celláknak megfelelő $A$ mátrix-beli oszlopvektorok tehát egy bázist alkotnak, amely megengedett is. A bázishoz tartozó megoldást az eljárás során kitöltött $m \times n$-es tábla tartalmazza.

Jelölje ezek után $B$ az előbbi módszerrel meghatározott kiindulási megengedett bázisunkat. Az $A$ mátrix $\underline{a}_{ij}$ oszlopvektorait állítsuk elő a $B$ bázissal:

$$\underline{a}_{ij} = \sum_{(\alpha,\beta)\in I_B} d_{(\alpha,\beta),(i,j)}\underline{a}_{\alpha\beta}$$

Ha $(i, j) \in I_B$, akkor ez az előállítás triviális. Ha viszont $(i, j) \notin I_B$, akkor készítsünk el egy olyan egyszerű hurkot, amely az $(i, j)$ cellán kívül csak báziscellákon halad át. (A korábban kimondott egyik tétel értelmében ilyen létezik.) Ugyancsak egy másik kimondott tétel értelmében igaz a következő egyenlőség:

$$\underline{a}_{i_1j_1} - \underline{a}_{i_1j_2} + \underline{a}_{i_2j_2} - \underline{a}_{i_2j_3} \pm \cdots - \underline{a}_{i_{k-1}j_k} = \underline{0},$$

ahol $i_1 = i,\ j_1 = j_k = j$ és ahol az $(i_1, j_2), (i_2, j_2), \ldots, (i_{k-1}, j_k)$ báziscellák. Ebből viszont adódik az

$$\underline{a}_{ij} = \underline{a}_{i_1j_2} - \underline{a}_{i_2j_2} + \underline{a}_{i_2j_3} \mp \ldots + \underline{a}_{i_{k-1}j_k}$$

előállítás, ahol $i_1 = i$ és $j_k = j$. Ez viszont azt jelenti, hogy az $\underline{a}_{ij}$ $B$-bázisbeli előállításában szereplő $d_{(\alpha,\beta),(i,j)}$ értékek csak 0, 1 vagy -1 lehetnek. Felhasználva az $\underline{a}_{ij}$ előbbi előállítását, a szimplex táblázatban szereplő $Z_{ij}$ értékekre a következőt kapjuk:

$$Z_{ij} = C_{i_1j_2} - C_{i_2j_2} + C_{i_2j_3} \mp \cdots + C_{i_{k-1}j_k}, \quad i = i_1, \quad j = j_k.$$

Így

$$Z_{ij} - C_{ij} = C_{i_1j_2} - C_{i_2j_2} + C_{i_2j_3} \mp \cdots + C_{i_{k-1}j_k} - C_{ij}.$$

Minthogy minden szállítási problémának van véges optimuma, ezért csak a következő két eset lehetséges:

a) Minden $i$-re és $j$-re $Z_{ij} - C_{ij} \le 0$ $(i = 1, \ldots, m; j = 1, 2, \ldots, n)$. Ebben az esetben megoldásunk már optimális.

b) Van olyan $(i,j)$, hogy a $Z_{ij} - C_{ij} > 0$ teljesül. Ekkor $B = B_0$ bázisról áttérünk egy $B_1$ bázisra, amely $B_0$-tól csak egy vektorban különbözik. Azt az $\underline{a}_{ij}$ vektort bisszük be a $B_1$ bázisba, amelynek $(i,j)$ indexére igaz, hogy $Z_{ij} - C_{ij} > 0$. A $B_0$ bázisból azt a $(\gamma, \delta)$ indexű vektort hagyjuk el, amelyre

$$\min_{\substack{(\alpha,\beta) \in I_{B_0} \\ d_{(\alpha,\beta),(i,j)>0}}} \frac{x_{\alpha\beta}}{d_{(\alpha,\beta),(i,j)}} = \frac{x_{\gamma\delta}}{d_{(\gamma,\delta),(i,j)}}$$

teljesül, ahol $(\gamma, \delta) \in I_{B_0}$, az $(i,j)$ pedig olyan, hogy $Z_{ij} - C_{ij} > 0$. Minthogy azonban a $d_{(\alpha,\beta),(i,j)} > 0$ csak úgy teljesülhet, hogy ha $d_{(\alpha,\beta),(i,j)} = 1$, az előbbi minimum meghatározására a

$$\min_{(\alpha,\beta)\in I_{B_0}:\ d_{(\alpha,\beta),(i,j)=1}} x_{\alpha\beta}$$

meghatározására redukálódik. Így tehát az $(i,j)$ cellához tartozó hurokban körbemenve, az $(i,j)$ cellától páratlan lépésszámra lévő $(\alpha, \beta)$ cellához tartozó $x_{\alpha\beta}$ értékek közül kell meghatározni a minimálisat.

Legyen $(\gamma, \delta)$ az a cella, amelyre az

$$x_{\gamma\delta} = \min_{d_{(\gamma,\delta),(i,j)}} x_{\alpha\beta}$$

teljesül. Ekkor az $\underline{a}_{\gamma\delta}$ vektort hagyjuk el a bázisból és behozzuk a helyébe az $\underline{a}_{ij}$ vektort. Ha $y_{ij}$-vel jelöljük a $B_1$ bázishoz tartozó bázismegoldás komponenseit $(i = 1, \ldots, m; j = 1, \ldots, n)$, akkor ezekre a következő adódik:

$y_{ij} = x_{\gamma\delta}$

$y_{\alpha\beta} = x_{\alpha\beta}$ ha $(i,j) \neq (\alpha, \beta) \notin I_{B_0}$, vagy ha $(\alpha, \beta) \in I_{B_0}$, de az $(\alpha, \beta)$ cella az $(i,j)$ cellához tartozó hurokban nem szerepel (a $d_{(\alpha,\beta),(i,j)} = 0$ miatt).

$y_{\alpha\beta} = x_{\alpha\beta} - x_{\gamma\delta}$ ha $(\alpha, \beta) \in I_{B_0}$ és az $(\alpha, \beta)$ cella az $(i,j)$ cellához tartozó hurokban az $(i,j)$ cellától páratlan lépésszámra van ($d_{(\alpha,\beta),(i,j)} = 1$ miatt).

$y_{\alpha\beta} = x_{\alpha\beta} + x_{\gamma\delta}$ ha $(\alpha, \beta) \in I_{B_0}$ és az $(\alpha, \beta)$ cella az $(i,j)$ cellához tartozó hurokban az $(i,j)$ cellától páros lépésszámra van ($d_{(\alpha,\beta),(i,j)} = -1$ miatt).

Ezzel megkaptuk a $B_1$ megengedett bázishoz tartozó bázismegoldást. Az eljárást mindaddig folytatjuk, amíg (a ciklizálást elkerülve véges sok lépés után) az optimális megoldáshoz el nem jutunk.

Minden lépésnél tehát meg kell határoznunk a $Z_{ij} - C_{ij}$, $(i,j) \notin I_B$ mennyiségeket, hiszen ezek előjeléből következtethetünk arra, hogy már optimális megoldásnál vagyunk-e, vagy pedig transzformációt kell végrehajtani. Minden egyes $Z_{ij} - C_{ij}$ mennyiség meghatározása hurokszerkesztéssel jár. Ha valamelyik $Z_{ij} - C_{ij} > 0$-nak adódik, akkor a hurokszerkesztést abbahagyhatjuk és a már említett módon meghatározzuk az új bázist és a hozzátartozó bázismegoldást.

## I.8. Dualitás

A szimplex módszer elméleti hátterének rövid áttekintésekor már megemlítettük, hogy minden $K = \{\underline{x} \mid A\underline{x} \leq \underline{b}\}$ konvex poliéder előállítható $P^{\Delta} + Q^{\angle}$ alakban, ahol

$$P^{\Delta} = \{\underline{x} \mid \underline{x} = \sum_{i=1}^{r} \lambda_i \underline{p}_i, \quad \lambda_i \geq 0, \quad i = 1, \ldots, r, \quad \sum_{i=1}^{r} \lambda_i = 1\}$$

$$Q^{\angle} = \{\underline{x} \mid \underline{x} = \sum_{j=1}^{s} \mu_j \underline{q}_j, \quad \mu_j \geq 0, \quad j = 1, \ldots, s\}$$

továbbá $Q^{\angle} = \{\underline{x} \mid A\underline{x} \leq \underline{0}\}$.

Vizsgáltuk azt is, hogy egy ilyen előállítás esetén mikor létezhet a $\max\limits_{K}\{\underline{c}^T\underline{x}\}$ problémának optimum értéke. Arra a megállapításra jutottunk, hogy a véges optimum létezésének szükséges és elégséges feltétele, hogy a $\underline{c}^T\underline{q}_j \leq 0$ teljesüljön minden $j \in \{1, 2, \ldots, s\}$ esetén. Ez viszont azt jelenti, hogy $c \in Q^{\angle}$, ami miatt pedig léteznie kell olyan $\underline{y} \geq \underline{0}$ $m$-dimenziós vektornak, amelyre $A^T\underline{y} = \underline{c}$ teljesül. Így tehát megállapíthatjuk, hogy a

$$\max_{K=\{\underline{x} \mid A\underline{x} \leq \underline{b}\}} \{\underline{c}^T\underline{x}\}$$

problémának pontosan akkor létezik véges optimuma, ha az

$$\begin{aligned} A^T\underline{y} &= \underline{c} \\ \underline{y} &\geq \underline{0} \end{aligned}$$

problémának van megoldása. Ez azt jelenti, hogy az

$$\begin{array}{ccc} A\underline{x} \leq \underline{b} & & A^T\underline{y} = \underline{c} \\ \max\{\underline{c}^T\underline{x}\} & \text{és az} & \underline{x} \geq \underline{0} \\ & & \min\{\underline{b}^T\underline{y}\} \end{array}$$

feladatok között szoros összefüggés van. Az ilyen feladatpárt duális párnak fogjuk nevezni, a két feladat közötti kapcsolatot pedig a dualitási tételben fogjuk megfogalmazni.

Most megadjuk a duál feladat pontos definícióját: az

$$\begin{aligned} A\underline{x} &\leq \underline{b} \\ \underline{x} &\geq \underline{0} \\ \max\{\underline{c}^T\underline{x}\} \end{aligned}$$

standard maximum feladat duálján az

$$\begin{gathered} A^T \underline{y} \geq \underline{c} \\ \underline{y} \geq \underline{0} \\ \min\{\underline{b}^T \underline{y}\} \end{gathered}$$

standard minimum feladatot értjük. Szokás az elsőt primál, a másodikat duál feladatnak, a kettőt együtt pedig duális párnak nevezni.

Definíciónk értelmében tehát egy standard maximum feladat duálja egy standard minimum feladat. Ha a kiinduló (primál) feladat nem az említett standard maximum formájú, akkor ekvivalens átalakítással előbb ilyen alakra kell hoznunk és csak azután képezhetjük a duálját, amit aztán ha lehet, még egyszerűbb alakra is hozhatunk. Ezek miatt bármilyen feladat lehet primál feladat, hiszen azt ekvivalens módon mindig standard maximum formára hozhatjuk.

A továbbiakban primál feladatnak mindig a kiindulási feladatot fogjuk tekinteni függetlenül attól, hogy az milyen formájú. Megjegyezzük viszont azt, hogy a standard maximum formát szokás primál, a standard minimum formát pedig duál formának nevezni.

A duál feladat definícióját felhasználva már könnyen bizonyíthatjuk, hogy az

$$\begin{array}{ccc} A\underline{x} \leq \underline{b} \text{ -nak} & & A^T \underline{y} \geq \underline{c} \\ \underline{x} \geq \underline{0} & \text{az} & \min\{\underline{b}^T \underline{y}\}, \\ \max\{\underline{c}^T \underline{x}\} & & \end{array}$$

valamint az

$$\begin{array}{cccc} A\underline{x} \leq \underline{b} \text{ -nak} & & A^T \underline{y} = \underline{c} & \\ \max\{\underline{c}^T \underline{x}\} & \text{az} & \underline{y} \geq \underline{0} & \text{a} \\ & & \min\{\underline{b}^T \underline{y}\} & \end{array}$$

duálja. A primál-duál megfeleltetésnek van egy még általánosabb változata is:

$$\begin{array}{ccc} A_{11}\underline{x}_1 + A_{12}\underline{x}_2 \leq \underline{b}_1 & & A_{11}^T \underline{y}_1 + A_{12}^T \underline{y}_2 \geq \underline{c}_1 \\ A_{21}\underline{x}_1 + A_{22}\underline{x}_2 = \underline{b}_2 & \text{duálja} & A_{21}^T \underline{y}_1 + A_{22}^T \underline{y}_2 = \underline{c}_2 \\ \underline{x}_1 \geq \underline{0} & & \underline{y}_1 \geq \underline{0} \\ \max\{\underline{c}_1^T \underline{x}_1 + \underline{c}_2^T \underline{x}_2\} & & \max\{\underline{b}_1^T \underline{y}_1 + \underline{b}_2^T \underline{y}_2\}, \end{array}$$

ahol

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \underline{b} = \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 \end{pmatrix}, \quad \underline{c} = \begin{pmatrix} \underline{c}_1 \\ \underline{c}_2 \end{pmatrix}, \quad \underline{x} = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix}, \quad \underline{y} = \begin{pmatrix} \underline{y}_1 \\ \underline{y}_2 \end{pmatrix}.$$

Ezekben a feladatokban bizonyos változók nincsenek nemnegativitási feltétellel korlátozva és egyes feltételek egyenlőség formájában adottak. Minthogy az ilyen primál-duál feladatpárt ekvivalens átalakításokkal visszavezethetjük a standard maximum illetve minimum feladatpárra, ezért elegendő csak ez utóbbit vizsgálnunk.

Belátható, hogy a duál duálja megegyezik a primál feladattal, ezért a duális pár bármelyik tagja tekinthető primálnak, a másik pedig duálnak. Az irodalomban a standard maximum feladatot szokás primál feladatnak nevezni.

Most egy lemmát mondunk ki, amelyre a dualitási tétel bizonyításánál lesz majd szükségünk.

**Lemma.** *Valamely lehetséges duális feladatpárban a maximum feladat tetszőleges megoldásához tartozó célfüggvényértéke nem nagyobb, mint a minimum feladat tetszőleges megoldásához tartozó célfüggvény értéke.*

*(Egy duális feladatpárt lehetségesnek nevezünk, ha mindkét tagja lehetséges, azaz mind a primál, mind pedig a duál feladatnak létezik megoldása.)*

*Bizonyítás.* A bizonyítást standard feladatpárra adjuk meg. Legyenek $\underline{x}$ és $\underline{y}$ a primál illetve a duál feladat tetszőleges megoldásai, azaz teljesüljenek az

$$\begin{aligned} A\underline{x} &\le \underline{b} & \qquad A^T\underline{y} &\ge \underline{c} \\ \underline{x} &\ge \underline{0} & \qquad \underline{y} &\ge \underline{0} \end{aligned}$$

egyenlőtlenségek. Az $A\underline{x} \le \underline{b}$ egyenlőtlenséget szorozzuk meg balról $\underline{y}^T$-tal, az $A^T\underline{y} \ge \underline{c}$ egyenlőtlenséget pedig $\underline{x}^T$-tal:

$$\underline{y}^T A\underline{x} \le \underline{y}^T\underline{b} \qquad \underline{x}^T A^T\underline{y} \ge \underline{x}^T\underline{c}$$

Minthogy $\underline{y}^T A\underline{x} = (\underline{y}^T A\underline{x})^T = \underline{x}^T A^T\underline{y}$, ezért igaz az

$$\underline{x}^T\underline{c} \;\le\; \underline{x}^T A^T\underline{y} \;=\; \underline{y}^T A\underline{x} \;\le\; \underline{y}^T\underline{b}$$

egyenlőtlenség, ami éppen az állításunk volt. □

Ezek után kimondjuk és bizonyítjuk a lineáris programozás alaptételét, a dualitási tételt.

**Tétel.** *(Dualitási tétel):*

a) *Ha a duális pár egyik tagjának van optimális megoldása, akkor a másiknak is van, és az optimum értékek egybeesnek.*

b) *Ha a duális pár egyik tagja lehetséges, a másik pedig nem, akkor a lehetséges feladatnak nincs véges optimuma.*
c) *Ha egy duális pár mindkét tagja lehetséges, akkor mind a kettőnek van optimális megoldása és az optimum értékek egybeesnek.*

*Bizonyítás.*
a) Minthogy az

$$\begin{array}{ccc} A\underline{x} \le \underline{b} & & A^T\underline{y} \ge \underline{c} \\ \underline{x} \ge \underline{0} & \text{és az} & \underline{y} \ge \underline{0} \\ \max\{\underline{c}^T\underline{x}\} & & \min\{\underline{b}^T\underline{y}\} \end{array}$$

feladatok közül bármelyik lehet a primál feladat, ezért elegendő bebizonyítani azt, hogy ha az

$$\begin{gathered} A\underline{x} \le \underline{b} \\ \underline{x} \ge \underline{0} \\ \max\{\underline{c}^T\underline{x}\} \end{gathered}$$

primál feladatnak van megengedett megoldása és véges optimuma, akkor az

$$\begin{gathered} A^T\underline{y} \ge \underline{c} \\ \underline{y} \ge \underline{0} \\ \min\{\underline{b}^T\underline{y}\} \end{gathered}$$

duál feladatnak is van és az optimum értékek egyenlők.

Tekintsük tehát a primál feladatot és a vele ekvivalens

$$\begin{gathered} (A, E)\begin{pmatrix} \underline{x} \\ \underline{u} \end{pmatrix} = \underline{b} \\ \underline{x} \ge \underline{0}, \quad \underline{u} \ge \underline{0} \\ \max\{\underline{c}^T\underline{x} + \underline{0}^T\underline{u}\} \end{gathered}$$

feladatot. Minthogy feltevésünk szerint a primál feladatnak van megengedett megoldása és véges optimuma, ezért ennek a feladatnak is van optimális megoldása. Így létezik olyan $B$ bázis, hogy az ahhoz tartozó szimplex táblázatban kiszámított $Z_k - C_k$ különbségek mind nemnegatívak. Minthogy az $(A, E)$ mátrix rangja $m$, a $B$ bázis négyzetes. Azt, hogy a $Z_k - C_k$ különbségek nemnegatívak, a következőképpen írhatjuk fel:

$$\begin{array}{ll} \underline{c}_B^T B^{-1}\underline{a}_k \ge c_k & \quad k = 1, 2, \dots, n \\ \underline{c}_B^T B^{-1}\underline{e}_k \ge 0 & \quad k = 1, 2, \dots, m. \end{array}$$

Definiáljuk most az $\underline{y}_0$ vektort az $\underline{y}_0 = \underline{c}_B^T B^{-1}$ egyenlőséggel. Az előbbi egyenlőtlenségek ekkor az

$$\underline{y}_0^T A \geq \underline{c}^T$$
$$\underline{y}_0^T E \geq \underline{0}^T$$

alakban is felírhatók, ezért azt kapjuk, hogy az $\underline{y}_0$ vektor megengedett megoldása a duál feladatnak.

Az előbbi lemma értelmében $\underline{c}^T \underline{x} \leq \underline{b}^T \underline{y}$ teljesül tetszőleges $\underline{x}, \underline{y}$ primál illetve duál megengedett megoldásokra. Most megmutatjuk, hogy van olyan primál illetve duál megengedett megoldás, amelyre az egyenlőség teljesül. Legyen $\underline{x}_0$ az átalakított primál feladat optimális megoldása és legyen $\underline{y}_0^T = \underline{c}_B^T \cdot B^{-1}$, ahol $B$ az $\underline{x}_0$-hoz tartozó optimális bázis. Minthogy $\underline{x}_{0_B} = B^{-1}\underline{b}$, ezért

$$\underline{c}^T \underline{x}_0 = \underline{c}_B^T \underline{x}_{0_B} = \underline{c}_B^T B^{-1} \underline{b} = \underline{y}_0^T \underline{b}.$$

Ezzel a dualitási tétel a) részét bebizonyítottuk.

b) Ha a lehetséges feladatnak volna véges optimuma, akkor a duális pár másik tagja is lehetséges lenne az előbbiek alapján, ami pedig ellentmondás.

c) Ha a duális pár mindkét tagja lehetséges, akkor a lemma értelmében a célfüggvényeik korlátosak, vagyis létezik mind a kettőnek véges optimuma. Az a)-nál megadott bizonyítás alapján beláthatjuk, hogy az optimum értékek egybeesnek. □

A primál és a duál feladat szoros kapcsolata alapján várható, hogy valamilyen összefüggés teljesül a primál és a duál optimális megoldások halmazai között. A következőkben erre a kapcsolatra utaló tételeket ismertetünk.

Tekintsük az

$$\begin{array}{cc} A\underline{x} \leq \underline{b} & A^T \underline{y} \geq \underline{c} \\ \underline{x} \geq \underline{0} & \underline{y} \geq \underline{0} \\ \max\{\underline{c}^T \underline{x}\} & \min\{\underline{b}^T \underline{y}\} \end{array}$$

primál-duál feladatpárt. Jelöljük $P$-vel a primál, $D$-vel pedig a duál lehetséges megoldások halmazát. Hasonlóan $P^0$ illetve $D^0$ jelöljék a megfelelő optimális megoldások halmazait.

Valamely $\underline{x}$ primál lehetséges megoldás esetén az $\underline{u} = \underline{b} - A\underline{x} \geq \underline{0}$ vektort primál slack vektornak nevezzük. Hasonlóan definiálhatjuk a duál slack vektort is: a $\underline{v}^T = \underline{y}^T A - \underline{c}^T \geq \underline{0}^T$ vektort duál slack vektornak nevezzük.

A slack vektorok lehetséges halmazait jelöljék $U$ (primál), illetve $V$, tehát

$$U = \{\underline{u} \mid \underline{u} = \underline{b} - A\underline{x},\ \underline{x} \in P\}$$
$$V = \{\underline{v} \mid \underline{v} = A^T\underline{y} - \underline{c},\ \underline{y} \in D\}$$

A $P$ és a $V$ halmazok kapcsolatára vonatkozik a következő tétel, amelyet bizonyítás nélkül közlünk.

**Tétel.** *Ha a primál-duál feladatpár lehetséges, akkor az* $x_i,\ i \in \{1, 2, \ldots, n\}$ *változó akkor és csakis akkor korlátos a* $P$*-ben, ha a megfelelő* $v_i = \underline{a}_i^T\underline{y} - c_i$ *duál slack változó korlátlan a* $V$ *halmazon.*

Ezen tétel következményeként megállapíthatjuk, hogy a primál lehetséges megoldások halmaza akkor és csakis akkor korlátos, ha a duál lehetséges megoldások halmazában minden változó és a duál slack változók minden halmazában minden slack változó korlátlan.

A következő tétel értelmében a primál lehetséges megoldások és a duál slack változók, illetve a duál lehetséges megoldások és a primál slack változók kapcsolatából megállapíthatjuk azt, hogy egy megoldás mikor optimális.

**Tétel.** *Az* $\underline{x}, \underline{y}$ *primál illetve duál lehetséges megoldások akkor és csakis akkor optimálisak, ha az* $A^T\underline{y} - \underline{c} = \underline{v}$*, illetve a* $\underline{b} - A\underline{x} = \underline{u}$ *slack változókra merőlegesek, azaz ha*

$$\underline{x}^T(A^T\underline{y} - \underline{c}) = 0 \qquad \textit{és}$$
$$\underline{y}^T(\underline{b} - A\underline{x}) = 0\ .$$

*Bizonyítás.* Előbb az elegendőséget vizsgáljuk. Ha fennállnak az előbbi ortogonalitási feltételek, akkor az

$$\underline{x}^T\underline{c} = \underline{x}^TA^T\underline{y} = \underline{y}^TA\underline{x} = \underline{y}^T\underline{b}$$

teljesül, azaz $\underline{x}$ és $\underline{y}$ valóban optimális megoldások, tehát $\underline{x} \in P^0$,$\underline{y} \in D^0$.

Ha viszont $\underline{x}$ és $\underline{y}$ optimálisak, akkor

$$\underline{x}^T\underline{c} = \underline{y}^T\underline{b}\ ,$$

továbbá $\underline{u} = \underline{b} - A\underline{x} \geq \underline{0}, \underline{v} = A^T\underline{y} - \underline{c} \geq \underline{0}$. Felhasználva, hogy $\underline{x} \geq \underline{0}$, $\underline{y} \geq \underline{0}$, $\underline{u} \geq \underline{0}$, $\underline{v} \geq \underline{0}$ és

$$\underline{y}^T\underline{u} + \underline{x}^T\underline{v} = \underline{y}^T\underline{b} - \underline{y}^TA\underline{x} + \underline{x}^TA^T\underline{y} - \underline{x}^T\underline{c} = 0,$$

kapjuk, hogy $\begin{array}{c}\underline{y}^T\underline{u} = 0\\ \underline{x}^T\underline{v} = 0\end{array}$, azaz $\begin{array}{c}\underline{y}^T(\underline{b} - A\underline{x}) = 0\\ \underline{x}^T(A^T\underline{x} - \underline{c}) = 0,\end{array}$ tehát teljesülnek az ortogonalitási feltételek. $\square$

Bizonyítás nélkül közöljük még az előbbi állítás egy erősebb változatát is:

Ha a primál-duál feladatpár lehetséges, akkor léteznek olyan $\underline{x}_0 \in P^0, \underline{y}_0 \in D^0$ primál illetve duál optimális megoldások is, amelyekkel

$$\underline{b} - A\underline{x}_0 + \underline{y}_0 > \underline{0}$$
$$-\underline{c} + A^T\underline{y}_0 + \underline{x}_0 > \underline{0}$$

teljesülnek. Ezen $\underline{x}_0$ és $\underline{y}_0$ természetesen teljesítik az előző tételben szereplő ortogonalitási feltételeket is. Az előző egyenlőtlenségek értelmében ahol az $\underline{u}_0 = \underline{b} - A\underline{x}_0$ vektornak 0 a komponense, ott az $\underline{y}_0$ vektornak pozitív és fordítva; továbbá ahol a $\underline{v}_0 = A^T\underline{y}_0 - \underline{c}$ slack vektornak 0 a komponense, ott $\underline{x}_0$-nak pozitív és fordítva.

## I.9. Érzékenység vizsgálat

A lineáris programozási problémákban szereplő alapadatok, az $A$ mátrix, a $\underline{b}$ és a $\underline{c}$ vektorok megváltozhatnak. A $\underline{b}$ kapacitásvektor, a $\underline{c}$ ár- vagy nyereségvektor a gyakorlati életben nem tekinthetők állandónak, de hasonló mondható az $A$ technológiai mátrixról is. A tervezés fázisában érdekes lehet az, hogy a megváltoztatott alapadatok útján adódó feladatoknak mik lesznek az optimális megoldásai. Ha csak egyetlen adat is változik meg, az már egy új problémát eredményez, ennek megfelelően más lehet az optimális megoldása, mint ami az eredetié volt. Emiatt meg kell oldanunk az új feladatot is.

Ebben a fejezetben azt vizsgáljuk, hogy egy megváltozott alapadatú problémának hogyan lehet úgy meghatározni az optimális megoldását, hogy ahhoz felhasználhassuk az eredetiét, azaz, hogy ne kelljen elölről kezdeni a feladat megoldását. Kíváncsiak vagyunk arra is, hogy a kapacitásvektorban ill. az ár- (nyereség-)vektorban milyen változásokat engedhetünk meg úgy, hogy a megváltoztatott feladat optimális megoldása megegyezzen az eredetivel, vagy hogy ezek a változások milyen mértékben változtatják meg az optimális megoldást.

A továbbiakban csak egyetlen, a $\underline{b}$ vagy a $\underline{c}$ vektorokban szereplő adat változásának a határát fogjuk vizsgálni. Mielőtt azonban elkezdenénk az érzékenységvizsgálatot, megadjuk a szimplex táblánknak egy másik, a korábbiakkal lényegében megegyező formáját.

Legyen a kiinduló feladatunk az

$$\begin{aligned} A\underline{x} &\leq \underline{b} \\ \underline{x} &\geq \underline{0} \qquad (\underline{b} \geq \underline{0}) \\ \max\{\underline{c}^T\underline{x}\} \end{aligned}$$

lineáris programozási feladat, ahol $A$ egy $m \times n$-es adott mátrix, a $\underline{b}$ $m$-, a $\underline{c}$ pedig $n$-dimenziós adott vektorok. Az $y_i \geq 0, i = 1, \ldots, m$ változókkal egészítsük ki a feltételrendszert egyenletrendszerré. Minthogy pedig $\underline{b} \geq \underline{0}$, válasszuk az $\underline{e}_i, i = 1, 2, \ldots, m$ egységvektorokat kiinduló bázisnak. Az ehhez tartozó szimplex táblánk:

Feltételezve, hogy a kiinduló feladatunknak létezik véges optimuma (különben nincs értelme az érzékenységvizsgálatnak), véges sok transzformációs lépés után eljuthatunk az optimális táblához:

ahol

$T = Y \cdot A$

$f_0 = \underline{c}_{B_0}^T \cdot \underline{x}_{B_0} = \underline{b}^T \underline{y}_0$ az optimum értéke

$\underline{x}_{B_0} = Y \cdot \underline{b}$ az optimális megoldás

$\underline{z}^T - \underline{c}^T = \underline{c}_{B_0}^T T - \underline{c}^T (\geq \underline{0})$

$\underline{y}_0 = \underline{c}_{B_0^T} Y \geq 0$ duál optimális megoldás.

A „jobboldal” érzékenységvizsgálata

Arra keressük a választ, hogyha a $\underline{b}$ kapacitásvektor valamelyik komponensét megváltoztatjuk, az milyen változást eredményez az optimális megoldásban. Vizsgáljuk az $i$-edik feltételt, amelynek $b_i$ jobboldala változzon meg $b_i + \lambda$-ra. (Az alapfeladatunk legyen most is $A\underline{x} \leq \underline{b}$, $\underline{x} \geq \underline{0}$, $\max\{\underline{c}^T\underline{x}\}$, $\underline{b} \geq \underline{0}$ feladat.)

Mivel a jobboldal vektora csak egyetlen elemében változik, így az $i$. egységvektort felhasználva az új feladat jobboldalát $\underline{b} + \lambda\underline{e}_i$ formában írhatjuk fel. Ez a változtatás a transzformációk során csak a megoldásoszlopban és a célfüggvényértékben eredményez változtatást, így az optimális megoldásra és az optimum értékére az

$$\underline{x}'_{B_0} = Y(\underline{b} + \lambda\underline{e}_i) = Y\underline{b} + \lambda Y\underline{e}_i = \underline{x}_{B_0} + \lambda\underline{y}_i$$
$$z'_0 = \underline{y}_0(\underline{b} + \lambda\underline{e}_i) = \underline{y}_0\underline{b} + \lambda\underline{y}_0\underline{e}_i = z_0 + \lambda y_{0i}$$

adódik. (Itt $\underline{y}_i$ az $Y$ mátrix $i$-edik oszlopa.) Az új megoldás a $\lambda$ azon értékei mellett lesz optimális, amelyekre

$$\underline{x}_{B_0} + \lambda\underline{y}_i \geq \underline{0}$$

teljesül, azaz amelyre $\underline{x}'_{B_0}$ megengedett megoldása a módosított feladatnak. Az előbbi vektori egyenlőtlenséget komponensekre is felírva kapjuk, hogy

$$x_{B_0j} + \lambda y_{ij} \geq 0 \qquad \forall j \in I_B\text{-re.}$$

Így ha valamely $y_{ij} > 0$, akkor $\lambda \geq \frac{-x_{B_0i}}{y_{ij}}$, ha pedig valamely $y_{ij} < 0$, akkor a $\lambda \leq \frac{-x_{B_0i}}{y_{ij}}$ kell, hogy teljesüljön. Az így adódó egyenlőtlenségből a $\lambda$ legkisebb, a másodikból pedig a $\lambda$ legnagyobb értéke határozandó meg:

$$\lambda_{\min} = \max\left\{\frac{-x_{B_0i}}{y_{ij}} \;\middle|\; y_{ij} > 0\right\}$$
$$\lambda_{\max} = \min\left\{\frac{x_{B_0i}}{-y_{ij}} \;\middle|\; y_{ij} < 0\right\}.$$

Ha nincs $y_{ij} > 0$, akkor $\lambda_{\min} = -\infty$, ha pedig nincs $y_{ij} < 0$, akkor $\lambda_{\max} = \infty$.

Amennyiben tehát

$$\lambda_{\min} \leq \lambda \leq \lambda_{\max},$$

akkor a szimplex tábla továbbra is optimális lesz.

Most külön is vizsgáljuk meg, hogy a $\lambda$ változtatásával hogyan változik a célfüggvény értéke.

A $z_0'$-re kapott eredmény alapján mondhatjuk, hogy a célfüggvényérték az $i$-edik duál változó optimális értékének $\lambda$-szorosával változik meg. Ha tehát a $\underline{b}$ vektor valamennyi komponensét bizonyos határok között megváltoztatjuk, akkor a célfüggvény értéke a $\underline{b}$ változása és az adott feltételhez tartozó duál változó optimális értékének a szorzatával változik meg.

A duál változó optimális értékét árnyékárnak, vagy elszámoló árnak szokás nevezni. Az árnyékár a célfüggvényváltozás és a jobboldal változása arányát mutatja. Más szavakkal: az árnyékár az egységnyi jobboldal változásra eső célfüggvényérték változása. A jobboldal megváltozásának és az árnyékárnak az előjele szabja meg a változás irányát.

Tekintsük a következő példát!

Egy vállalatnak kétféle termék gyártása felől kell döntenie, amelyeket három gépen kell előállítania. Az első termék egységének elkészítéséhez szükséges gépidők rendre $1, 2, 0$, a másodikéhoz pedig $3, 1, 2$. Az egyes gépek adott időszakra vonatkozó kapacitásai $30, 35, 25$, az egyes termékek eladásából származó haszon pedig – egységnyi termékre vonatkoztatva – 2 illetve 6 egység. Mi lesz az optimális termékösszetétel és hogyan befolyásolja a gépkapacitások megváltozása a vállalat optimális termékösszetételét?

Oldjuk meg először a feladatot!

Jelöljék $x_1, x_2$ a döntési változókat. Ekkor a feladat matematikai modellje:

$$\begin{aligned} x_1 + 3x_2 &\le 30 \\ 2x_1 + x_2 &\le 35 \\ 2x_2 &\le 25 \\ x_1, x_2 &\ge 0 \end{aligned}$$

$$\overline{\max\{2x_1 + 6x_2\}}$$

Ehhez a feladathoz a következő kiinduló szimplex táblát lehet elkészíteni:

Hajtsuk végre az $\underline{a}_2 \leftrightarrow \underline{e}_1$ cserével járó elemi bázistranszformációt, amely után a következő szimplex táblát nyerjük:

Ez már optimális tábla. A kiinduló (primál) feladat optimális megoldása: $x_1 = 0,\ x_2 = 10$. Az optimum értéke 60. A duál fealadat optimális megoldása $y_1 = 2,\ y_2 = 0,\ y_3 = 0$.

Megjegyzés: Amiatt, mert $\underline{a}_1$ alatt az utolsó sorban 0 áll, a feladatnak létezik alternatív optimális megoldása is. Ehhez eljutunk, ha végrehajtjuk az $\underline{a}_1 \leftrightarrow \underline{e}_2$ elemi bázistranszformációt. Az alternatív optimális megoldáshoz tartozó szimplex tábla:

Itt az optimális megoldás $x_1 = 15, x_2 = 5$, az optimum értéke 60. A duál feladat optimális megoldása itt is $y_1 = 2, y_2 = y_3 = 0$.

Vizsgáljuk most először azt, hogy mi lesz az optimális megoldás, ha az első gép kapacitását 30-ról $30 + \lambda$-ra változtatjuk.

Az elméleti vizsgálódásaink alapján már tudjuk, hogy az $Y$ első oszlopára illetve az első duálváltozó optimális értékére lesz szükségünk. A megoldásoszlopban a régi adatokhoz rendre hozzá kell adni az $\underline{y}_1 = \underline{e}_1$ oszlopbeli elemek $\lambda$-szorosát. Így az első optimális megoldás esetén az

$$\underline{x}'_{B_0} = \begin{pmatrix} 10 + \frac{\lambda}{3} \\ 25 - \frac{\lambda}{3} \\ 5 - \frac{2\lambda}{3} \end{pmatrix} \qquad f'_0 = 60 + 2\lambda$$

adódik. (Ugyanez a második optimális megoldás esetében az

$$f'_0 = 60 + 2\lambda, \qquad \underline{x}'_{B_0} = \begin{pmatrix} 5 + \frac{2\lambda}{5} \\ 15 - \frac{\lambda}{5} \\ 15 - \frac{4\lambda}{5} \end{pmatrix} \quad \text{lesz.)}$$

Minthogy $\underline{x}'_{B_0} \geq \underline{0}$ kell, hogy legyen, ezért

$$\begin{array}{lcl} 10 + \frac{\lambda}{3} \geq 0 & & 5 + \frac{2\lambda}{5} \geq 0 \\ 25 - \frac{\lambda}{3} \geq 0\,, & \text{vagy} & 15 - \frac{\lambda}{5} \geq 0 \\ 5 - \frac{2\lambda}{3} \geq 0 & & 15 - \frac{4\lambda}{5} \geq 0. \end{array}$$

Így az első esetben $\lambda_{\min} = -30, \lambda_{\max} = 7.5$; a második esetben pedig $\lambda_{\min} = -12.5, \lambda_{\max} = 18.75$.

Mindezek miatt az első esetben

$$x_1 = 0$$

$$x_2 = 10 + \frac{\lambda}{3}$$

lesz az optimális megoldás, az optimum értéke pedig $60 + 2\lambda$ mindaddig, amíg $-30 \leq \lambda \leq 7.5$.

(A második optimális megoldást felhasználva $x_1 = 15 - \frac{\lambda}{5}$,$x_2 = 5 + \frac{2\lambda}{5}$ adódik optimálisnak, optimum értéknek $60 + 2\lambda$, ahol $-12.5 \leq \lambda \leq 18.75$.)

Látható, hogy a $\lambda$ kapacitásváltozás mértékének kétszeresével változik meg az árbevétel. Ezt az arányossági tényezőt (ami az $y_1$ duálváltozó optimális értéke) azért nevezik árnyékárnak, vagy elszámoló árnak, mert az első gépet értékeli olyan szempontból, hogy annak egységnyi kapacitása mennyit ér, mekkora célfüggvény változást eredményez. Az első gép árnyékára tehát 2 egység.

Vizsgáljuk most azt, hogy hogyan módosul az optimum értéke, illetve az optimális megoldás, ha a második gép kapacitását 35-ről $35 + \lambda$-ra változtatjuk. Mivel az első optimális megoldás esetén $\underline{y}_2 = \underline{e}_2$, ezért

$$\underline{x}'_{B_0} = \begin{pmatrix} 10 \\ 25 + \lambda \\ 5 \end{pmatrix}, \qquad f'_0 = 60.$$

(A második optimális megoldás esetén:

$$\underline{x}'_{B_0} = \begin{pmatrix} 5 - \frac{\lambda}{5} \\ 15 + \frac{3\lambda}{5} \\ 15 + \frac{2\lambda}{5} \end{pmatrix}, \qquad f'_0 = 60.)$$

Így ha a második gép kapacitását 35-ről $35 + \lambda$-ra változtatjuk, akkor az első esetben az optimális megoldás minden $\lambda \geq -25$ esetén $x_1 = 0$, $x_2 = 10$ lesz, amelyhez $f'_0 = 60$ lesz az optimális érték. (Mindez a második megoldásnál

$$x_1 = 15 + \frac{3\lambda}{5},\ x_2 = 5 - \frac{\lambda}{5},\ f'_0 = 60, \quad \text{ha } -25 \leq \lambda \leq 25.)$$

A második gép árnyékára mindkét optimális megoldás esetén 0, vagyis ezen gép kapacitásának az adott határon belüli megváltoztatása az árbevételt nem befolyásolja.

Végül vizsgáljuk meg, hogy hogyan változik meg az optimális megoldás és az optimum értéke, ha a harmadik gép kapacitását 25-ről $25+\lambda$-ra változtatjuk. Az első optimális megoldás esetén

$$\underline{x}'_{B_0} = \begin{pmatrix} 10 \\ 25 \\ 5+\lambda \end{pmatrix}, \qquad f'_0 = 60$$

lesz, míg a második esetében

$$\underline{x}'_{B_0} = \begin{pmatrix} 5 \\ 15 \\ 15+\lambda \end{pmatrix}, \qquad f'_0 = 60 \quad \text{adódik.}$$

Így az első esetben $x_1 = 0$, $x_2 = 10$, $f'_0 = 60$ lesz mindaddig, amíg $\lambda \geq -5$. Ugyanez a másofik optimális megoldás esetében $x_1 = 15$, $x_2 = 5$, $f'_0 = 60$, hacsak $\lambda \geq -15$. A harmadik gép árnyékára 0 lesz.

## A célfüggvény érzékenységvizsgálata

Most arra keressük a választ, hogy ha valamelyik döntési változó célfüggvénybeli együtthatóját megváltoztatjuk, az milyen változást eredményez az optimális megoldásban.

Vizsgáljuk a $j$-edik változót, amelynek $c_j$ az együtthatója. Ez változzon most meg $c_j+\lambda$-ra. Belátható, hogy ez a változás az optimális szimplex táblának csak az utolsó sorát érinti.

Tekintsük továbbra is az előző példát és vizsgáljuk a célfüggvényben szereplő első együttható változásának a hatását. Változzon tehát az első termék haszna 2-ről $2+\lambda$-ra. Ekkor a kiinduló szimplex tábla a következő lesz:

Ha $-2-\lambda<0$, azaz $-2<\lambda$, akkor $\underline{a}_1$ bevihető a bázisba az $\underline{e}_2$ helyébe. Az új bázishoz tartozó szimplex tábla:

Ha $-6+\frac{2+\lambda}{2}<0$, azaz $\lambda<10$, akkor $\underline{a}_2$ bevihető a bázisba:

Ez a tábla akkor lesz optimális, ha

$$2 - \frac{\lambda}{5} \geq 0 \quad \text{és} \quad \frac{3\lambda}{5} \geq 0, \qquad \text{azaz}$$

$0 \leq \lambda \leq 10$. Ekkor $x_1 = 15,\ x_2 = 5$ lesz az optimális megoldás, az optimum értéke pedig $f_0' = 60 + 15\lambda$ lesz. Így $\lambda_{\min} = 0,\ \lambda_{\max} = 10$. Amennyiben tehát $\lambda \in [0, 10]$, akkor az optimális megoldás a $\lambda$ értékétől függetlenül $x_1 = 15,\ x_2 = 5$.

Hasonlóan vizsgálhatnánk meg még a második célfüggvényegyüttható változásának hatását is, ezt azonban már az olvasóra bízzuk.

# II. Nemlineáris programozás

A nemlineáris programozási feladatokat általános alakban a következőképpen fogalmazhatjuk meg: a

$$\begin{aligned} g_i(\underline{x}) &\le b_i\ , \quad i = 1, 2, \ldots, m \\ \underline{x} &\in \mathbb{R}^n \end{aligned}$$

feltételek mellett határozzuk meg az $f(\underline{x})$ célfüggvény maximumát. (Az $\underline{x} \ge \underline{0}$ feltételt terszmészetesen figyelembe vehetjük a $g_i(\underline{x}) \le b_i$ feltételek között!)

Nem találtak olyan algoritmust, amellyel az ilyen általánosan megfogalmazott feladatokat meg lehetne oldani. Ugyanakkor azonban, ha a $g_i(\underline{x})$ és az $f(\underline{x})$ függvényekre bizonyos megszorításokat teszünk, már matematikailag kezelhető feladatokhoz jutunk. Ezek a megszorítások ugyan speciálissá teszik a problémát, de azok a gyakorlatban még jól használhatók lesznek. A gyakorlati életben előforduló nemlineáris programozási problémák legtöbbször ugyanis olyanok, hogy a feltételeknek eleget tevő vektorok összessége konvex poliédert határoz meg és csak az $f(\underline{x})$ célfüggvény nemlineáris. Ezeknél a problémáknál az esetek többségében a szimplex módszer valamelyik változatának alkalmazásával egy egzakt, vagy pedig közelítő megoldást nyerhetünk.

## II.1. Hiperbolikus programozás

A matematikai programozás azon speciális esetét, amikor a korlátozó feltételekben szereplő függvények lineárisak, a célfüggvény pedig lineáris törtfüggvény, hiperbolikus programozásnak nevezzük[1]. Ilyen tipusúak azok a gazdasági feladatok, amelyeknél a különböző aktivitások az erőforrásokból mennyiségükkel arányos részt kötnek le és amelyeknél egy fajlagos mutatószám extrémumát, rendszerint a hozamok és ráfordítások legkedvezőbb arányát kell meghatároznunk.

A hiperbolikus programozás feladatának általános megfogalmazása a következő: az

$$\begin{aligned} A\underline{x} &\le \underline{b} \\ \underline{x} &\ge \underline{0} \end{aligned}$$

[1]A hiperbolikus programozási feladatok megoldására szolgáló algoritmus kidolgozása Martos Béla nevéhez fűződik.

feltételek mellett meghatározandó a

$$\varphi(\underline{x}) = \frac{\underline{c}^T \underline{x} - c_0}{\underline{d}^T \underline{x} - d_0}$$

függvény maximuma.

A $\varphi(\underline{x})$ célfüggvénnyel kapcsolatban a következő megszorításokat tesszük:

- A $\left(\underline{d}^T \underline{x} - d_0\right)$-ról kikötjük, hogy a lehetséges megoldások halmazán nem konstans, ellenkező esetben ugyanis lineáris programozási feladatról lenne szó, azt pedig már tárgyaltuk.
- A $\left(\underline{c}^T \underline{x} - c_0\right)$-tól megköveteljük, hogy ne legyen a $\underline{d}^T \underline{x} - d_0$ függvény konstansszorosa, ellenkező esetben ugyanis a $\varphi(\underline{x})$ konstans, s így a feladat értelmetlen.
- Az általánosság megszorítása nélkül feltételezhetjük, hogy a lehetséges megoldások között vannak olyanok, amelyekre $\underline{d}^T \underline{x} - d_0 > 0$.

Először olyan hiperbolikus programozási feladatot tekintünk, amelyiknél a megengedett megoldások halmaza korlátos konvex poliéder (azaz konvex politop) és amelyiknél az összes lehetséges $\underline{x}$ vektor esetén a $\underline{d}^T \underline{x} - d_0 > 0$ teljesül. Ez esetben a szimplex módszer alkalmazhatóságát a következő tétel biztosítja:

Ha a

$$\varphi(\underline{x}) = \frac{\underline{c}^T \underline{x} - c_0}{\underline{d}^T \underline{x} - d_0}$$

függvénynek az értelmezési tartománya egy

$$A\underline{x} \leq \underline{b}$$
$$\underline{x} \geq \underline{0}$$

feltételek által meghatározott korlátos konvex poliéder és annak minden pontjában a $\underline{d}^T \underline{x} - d_0 > 0$ teljesül, akkor a $\varphi(\underline{x})$ függvénynek a konvex politop felett van véges maximuma és ezt felveszi a politopnak legalább egy csúcspontján.

Tekintsük most az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$
$$\max \left\{ \frac{\underline{c}^T \underline{x} - c_0}{\underline{d}^T \underline{x} - d_0} \right\}$$

feladatot. (Ha a kiindulási problémánk nem ilyen formájú, akkor a már tanult ekvivalens átlalakításokkal ilyen formájúra hozzuk.) Az egyszerűség kedvéért tételezzül fel, hogy az $m \times n$-es $A$ mátrix rangja $m$, s hogy az

$\underline{a}_1, \underline{a}_2, \dots, \underline{a}_m$ vektorok jelölik a kiindulási $B$ bázishoz tartozó vektorokat. Ezzel a $B$ bázissal a következő szimplex táblázatot készíthetjük el:

ahol

$$t_i = \left(\underline{c}_B^T \underline{x}_B - c_0\right) \cdot \left(\underline{d}_B^T \underline{f}_i - d_i\right) - \left(\underline{c}_B^T \underline{f}_i - c_i\right) \cdot \left(\underline{d}_B^T \underline{x}_B - d_0\right)$$

és $\underline{f}_i$ az $f_{1,i}, \ldots, f_{m,i}$ értékekből képzett azon vektor, amely az $\underline{a}_i$-nek az $\{\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_m\} = B$ bázisbeli előállításában szereplő együtthatóiból áll.

Minthogy feltevésünk szerint a feltételek egy korlátos zárt halmazt határoznak meg, s a célfüggvény ezen halmaz felett folytonos, a problémának van optimális megoldása. Próbáljuk most a feladatunkat szimplex módszerrel megoldani.

Tegyük fel, hogy még nem találtuk meg az optimális megoldást, és legyen $\underline{a}_k$ az a vektor, amelyet a bázisba be akarunk vinni. (A $k$ index meghatározására a későbbiek során térünk ki.) Meg kell határoznunk a

$$\delta = \min_{j: f_{j,k} > 0} \left\{ \frac{x_j}{f_{j,k}} \right\}$$

értékét, illetve azt a $j$ indexet, amelynél ez a minimum teljesül. Az $\underline{a}_k$ vektor bázisba való bevonása után, a transzformációs formulák alapján, a következőket kapjuk:

$$\begin{array}{llll} \text{a} & \underline{c}_B^T \underline{x}_B - c_0 & \text{értéke} & \underline{c}_B^T \underline{x}_B - c_0 - \delta\left(\underline{c}_B^T \underline{f}_k - c_k\right) \text{-ra,} \\[2ex] \text{a} & \underline{d}_B^T \underline{x}_B - d_0 & \text{értéke} & \underline{d}_B^T \underline{x}_B - d_0 - \delta\left(\underline{d}_B^T \underline{f}_k - d_k\right) \text{-ra} \end{array}$$

változik, így az új célfüggvényérték

$$\frac{\underline{c}_B^T \underline{x}_B - c_0 - \delta\left(\underline{c}_B^T \underline{f}_k - c_k\right)}{\underline{d}_B^T \underline{x}_B - d_0 - \delta\left(\underline{d}_B^T \underline{f}_k - d_k\right)}$$

lesz. Minthogy az új bázisra való áttérés akkor jelenti a maximumhoz való közeledést, ha ezen új célfüggvényérték nagyobb, mint a régi táblázatban szereplő, ezért

$$\frac{\underline{c}_B^T \underline{x}_B - c_0 - \delta\left(\underline{c}_B^T \underline{f}_k - c_k\right)}{\underline{d}_B^T \underline{x}_B - d_0 - \delta\left(\underline{d}_B^T \underline{f}_k - d_k\right)} - \frac{\underline{c}_B^T \underline{x}_B - c_0}{\underline{d}_B^T \underline{x}_B - d_0} > 0$$

kell, hogy teljesüljön. Ebből viszont következik, hogy

$$\frac{\delta \cdot \left[\left(\underline{c}_B^T \underline{x}_B - c_0\right) \cdot \left(\underline{d}_B^T \underline{f}_k - d_k\right) - \left(\underline{d}_B^T \underline{x}_B - d_0\right) \cdot \left(\underline{c}_B^T \underline{f}_k - c_k\right)\right]}{\left(\underline{d}_B^T \underline{x}_B - d_0\right) \cdot \left[\underline{d}_B^T \underline{x}_B - d_0 - \delta\left(\underline{d}_B^T \underline{f}_k - d_k\right)\right]} > 0.$$

Minthogy azonban

$$\delta > 0,\ \underline{d}_B^T\underline{x}_B - d_0 > 0,\ \underline{d}_B^T\underline{x}_B - d_0 - \delta\cdot\left(\underline{d}_B^T\underline{f}_k - d_k\right) > 0,$$

ezért az előbbi egyenlőtlenség a

$$\left(\underline{c}_B^T\underline{x}_B - c_0\right)\left(\underline{d}_B^T\underline{f}_k - d_k\right) - \left(\underline{d}_B^T\underline{x}_B - d_0\right)\left(\underline{d}_B^T\underline{f}_k - c_k\right) > 0 \text{ -ra},$$

azaz $t_k > 0$-ra redukálódik. Mindezek miatt a transzformációnál olyan $k$ indexet kell tekintenünk, amelyre $t_k > 0$.

A hiperbolikus programozásnál alkalmazott szimplex módszer tehát abban áll, hogy a pozitív $t_k$ értékkel rendelkező $\underline{a}_k$ vektort bevonjuk a bázisba. Ezt mindaddig megtesszük, amíg ilyen $\underline{a}_k$ vektor található. Ezt a transzformációt természetesen csak akkor tudjuk végrehajtani, ha az $\underline{f}_k$ vektornak van pozitív komponense, ami esetünkben a véges optimum létezése miatt mindig teljesül. Minthogy a megengedett megoldások által meghatározott konvex politopnak véges számú csúcspontja van, s hogy minden $t_k > 0$ értékkel jellemzett vektor bevonása növeli a célfüggvény értékét, ezért véges számú lépés után olyan extrémális ponthoz jutunk, amelyik optimális. Nyilvánvaló, hogy az optimális megoldás esetén minden $i$-re $t_i \leq 0,\ i = 1, 2, \ldots, n$ kell, hogy teljesüljön.

Az ilyen egyszerű hiperbolikus programozási feladatoknál a szimplex módszerrel való megoldáshoz szükséges műveletek száma csak kis mértékben tér el egy azonos méretű lineáris programozási feladat megoldásakor fellépő műveletek számától.

Az általános hiperbolikus programozási feladatnál nem tételezzük fel a feltételek által meghatározott konvex poliéder korlátosságát, valamint megengedjük, hogy a célfüggvény nevezője 0 értéket is felvegyen. Továbbra is feltesszük azonban, hogy a nevező nem konstans a megoldáshalmazon, így léteznie kell olyan $\underline{x}_0$ megoldásnak, amelyre $\underline{d}^T\underline{x}_0 - d_0 > 0$.

Mielőtt az általános hiperbolikus feladat megoldhatóságát vizsgálnánk, néhány fogalmat kell megadnunk.

Tekintsük az általános hiperbolikus programozási feladat egy olyan $\underline{x}_0$ megoldását, amelyben a $\varphi(\underline{x}_0)$ célfüggvény nevezője a $\underline{d}^T\underline{x}_0 - d_0 > 0$. Ilyen $\underline{x}_0$ a kikötéseink miatt létezik. Egy ilyen $\underline{x}_0$-hoz képest az általános hiperbolikus programozási feladatban szereplő feltételek által meghatározott konvex poliéder egy $\underline{x}$ pontját „jó pont"-nak fogjuk nevezni, ha vagy a $\underline{d}^T\underline{x} - d_0 > 0$, vagy pedig $\underline{d}^T\underline{x} - d_0 = 0$ és $\underline{c}^T\underline{x} - c_0 < 0$. Az említett általános hiperbolikus programozási feladat egy másik $\underline{x}$ pontját $\underline{x}_0$-hoz viszonyítva „rossz pont"-nak nevezzük, ha vagy $\underline{d}^T\underline{x} - d_0 < 0$, vagy pedig $\underline{d}^T\underline{x} - d_0 = 0$ és

$\underline{c}^T \underline{x} - c_0 > 0$. Végül egy $\underline{x}$ pontot „szinguláris pont”-nak nevezünk, ha $\underline{d}^T \underline{x} - d_0 = \underline{c}^T \underline{x} - c_0 = 0$.

Az általános hiperbolikus programozási feladat megoldhatóságával kapcsolatban a következő tételeket mondhatjuk ki:

- „Rossz eset” : ha a feladatnak van „rossz pont”-ja, akkor a $\varphi(\underline{x})$ célfüggvény felülről nem korlátos, azaz a feladatnak nincs véges optimuma.
- „Jó eset” : ha a konvex poliédernek csak „jó pont”-ja van, akkor vagy a $\varphi(\underline{x})$ célfüggvényérték felülről nem korlátos, vagy van a konvex poliédernek olyan extrémális pontja, amelyben a $\varphi(\underline{x})$ felveszi a maximális értékét.
- „Szinguláris eset” : ha a feltételek által meghatározott konvex poliédernek nincs „rossz pont”-ja, de van szinguláris pontja, akkor a $\varphi(\underline{x})$ célfüggvénynek vagy nincs maximum értéke, vagy pedig létezik olyan „jó pont”-ja, ahol a $\varphi(\underline{x})$ felveszi a maximumát.

Ezen tételeket felhasználva az általános hiperbolikus feladat megoldásának algoritmusa a következő lesz:

1. lépés: A lineáris programozásnál alkalmazott módszerek valamelyikével meghatározunk egy $\underline{x}_0$ kiinduló megengedett bázismegoldást, majd a 2. lépésnél folytatjuk az algoritmust.

2. lépés: Megvizsgáljuk a kiinduló megoldáshoz tartozó $\underline{d}^T \underline{x}_0 - d_0$ értéket. Ha $\underline{d}^T \underline{x}_0 - d_0 > 0$, akkor a 3. lépésnél folytatjuk az eljárást. Ha $\underline{d}^T \underline{x}_0 - d_0 < 0$, akkor $(-1)$-gyel bővítjük a $\varphi(\underline{x})$ célfüggvényt, ezzel elérve, hogy az új nevező pozitív legyen, majd a 3. lépésnél folytatjuk az eljárást. Ha $\underline{d}^T \underline{x}_0 - d_0 = 0$, akkor megvizsgáljuk, hogy egy bázistranszformáció során $\underline{d}^T \underline{x}_0 - d_0$ értéke növelhető, vagy csökkenthető. (Minthogy a $\underline{d}^T \underline{x}_0 - d_0$ nem konstans, a kettő közül valamelyiknek be kell következnie.) Végrehajtjuk a transzformációt, s ha növelhető, akkor a 3. lépésnél folytatjuk, ha pedig csökkenthető, akkor $\varphi(\underline{x})$-et $(-1)$-gyel bővítjük, majd szintén a 3. lépésnél folytatjuk.

3. lépés: Szimplex módszer alkalmazásával az

$$A\underline{x} = \underline{b}$$
$$\underline{x} \geq \underline{0}$$

feltételek mellett meghatározzuk a

$$\underline{d}^T \underline{x} - d_0$$

lineáris függvény minimumát.
Ha $\min\left\{\underline{d}^T\underline{x} - d_0\right\} > 0$, akkor a 5. lépésnél folytatjuk az eljárást.(„Jó eset").
Ha $\min\left\{\underline{d}^T\underline{x} - d_0\right\} = 0$, akkor a 4. lépésre térünk át.
Ha a minimumszámítás közben $\underline{d}^T\underline{x} - d_0 < 0$ adódik, akkor az eljárást megszakítjuk, a feladatnak nincs optimális megoldása. („Rossz eset").

4. lépés: Az

$$A\underline{x} = \underline{b}$$
$$\underline{d}^T\underline{x} = d_0$$
$$\underline{x} \geq \underline{0}$$

feltételek mellett meghatározzuk a $\underline{c}^T\underline{x} - c_0$ függvény maximumát. (Mivel ez is lineáris programozási feladat, szimplex módszerrel oldhatjuk meg.)
Ha a maximum értéke negatív, akkor az 5. lépésnél kell folytatnunk az algoritmust. („Jó eset").
Ha a maximum értéke 0, akkor a 6. lépésnél folytatjuk az eljárást. („Szinguláris eset").
Ha a maximum számítása közben $\underline{c}^T\underline{x} - c_0 > 0$ adódik, akkor az eljárást megszakítjuk, a feladatnak nincs véges optimuma. („Rossz eset").

5. lépés: A „jó eset"-nek megfelelő feladatot kell megoldani. Minthogy az eljárásunk közben többször is alkalmaztuk a szimplex módszert, ezért ennél a lépésnél célszerű kiinduló táblának azt tekinteni, amelynél a legnagyobb $\varphi(\underline{x})$ célfüggvényérték szerepelt. Ugyanazt a szimplex módszert fogjuk használni az optimális megoldás meghatározására, mint amit már a nemáltalános esetben is használtunk. Elkészítjük a szimplex táblázatot mostmár a hiperbolikus feladathoz és kiértékeljük azt.
Ha minden $k$-ra $t_k \leq 0,\ k = 1, 2, \ldots, n$, akkor már optimális megoldásnál vagyunk. Megadjuk az optimális megoldást, a hozzá tartozó optimum értéket, majd befejezzük az eljárást.
Ha létezik olyan $k$ érték, amelyre $t_k > 0$ és az $f_{ik} \leq 0$ teljesül minden $i \in I_B$-re, akkor az eljárás befejeződik, mert ilyenkor a $\varphi(\underline{x})$ célfüggvénynek nincs véges maximum értéke. Ez azonban a lineáris esettel ellentétben most kétféleképpen is bekövetkezhet. Az egyik esetben $\underline{d}_B^T\underline{f}_k - d_k = 0$. Ilyenkor a $\varphi(\underline{x})$ értéke minden határon túl növelhető, vagyis a célfüggvényérték felülről nem korlátos. A

másik esetben $\underline{d}_B^T\underline{f}_k - d_k < 0$. Ekkor a $\varphi(\underline{x})$ célfüggvény ugyan felülről korlátos (pontos felső korlátja a $\frac{\underline{d}_B^T\underline{f}_k - c_k}{\underline{d}_B^T\underline{f}_k - d_k}$ érték), de a pontos felső korlátnak megfelelő értéket a konvex poliéder egyetlen végesben fekvő pontjában sem veszi fel.

Ha létezik $t_k > 0$ és az $f_{ik}, i \in I_B$ értékek között minden ilyen $k$-ra van pozitív, akkor az ilyen tábla nem optimális, de azt sem állíthatjuk, hogy nincs véges optimum érték. Ilyenkor elemi bázistranszformációt hajtunk végre és újból az 5. lépésnél folytatjuk az eljárást.

6. lépés: Ez esetben a szimplex táblánk kétféle csoportba sorolható. Az egyik esetben van olyan $1 \leq k \leq n$, hogy $\underline{d}_B^T\underline{f}_k - d_k < 0$ és a

$$\frac{\underline{c}_B^T\underline{f}_k - c_k}{\underline{d}_B^T\underline{f}_k - d_k} \geq \frac{\underline{c}_B^T\underline{f}_j - c_j}{\underline{d}_B^T\underline{f}_j - d_j}$$

egyenlőtlenség teljesül minden olyan $j$-re, amelyre $\underline{d}_B^T\underline{f}_j - d_j < 0$, továbbá az $\underline{f}_k$ vektornak van pozitív komponense. Az ilyen indexű $\underline{a}_k$ vektort bevonva a bázisba optimális megoldáshoz jutunk, majd befejezzük az eljárást. A másik esetben a szimplex tábla olyan, hogy van olyan $1 \leq k \leq n$ index, amelynél

$$\underline{d}_B^T\underline{f}_k - d_k < 0,$$
$$\frac{\underline{c}_B^T\underline{f}_k - c_k}{\underline{d}_B^T\underline{f}_k - d_k} \geq \frac{\underline{c}_B^T\underline{f}_j - c_j}{\underline{d}_B^T\underline{f}_j - d_j} \text{ minden olyan}$$
$$j\text{-re, amelyre } \underline{d}_B^T\underline{f}_j - d_j < 0$$
$$\underline{f}_k \leq \underline{0}$$

Ez esetben az

$$(x_1 - \lambda \cdot f_{i_1,k} \cdots x_m - \lambda \cdot f_{i_m,k} 0 \cdots 0 \lambda 0 \cdots)^T$$

vektorsereg, amelyeknél a $k$. komponens $\lambda$, terszőleges $\lambda > 0$ mellett optimális megoldása a feladatnak. (Itt $I_B = \{i_1, i_2, \ldots, i_m\}$.) Ezzel az eljárás befejeződött.

Ez az algoritmus feltételezi, hogy a feltételek által meghatározott konvex poliéder nem üres halmaz. Ha a kiindulási megoldás meghatározására például a kétfázisú szimplex módszert alkalmazzuk, akkor megállapíthatjuk azt az esetet is, amikor a feladat feltételrendszere ellentmondásos, azaz a feladat megoldáshalmaza üres. Ez esetben nyilvánvalóan ezen megállapítással be is fejeződik az eljárás.

Az algoritmus továbbá feltételezi azt is, hogy a feltételek által meghatározott konvex poliédernek létezik legalább egy extrémális pontja. Ez csak abban a nagyon speciális esetben nem következik be, amikor a megoldáshalmaz tartalmaz egyenest is. Ilyenkor azonban bizonyítható, hogy a hiperbolikus programozási feladatnak nincs optimális megoldása.

## II.2. Kvadratikus programozás

Kvadratikus programozási feladatnak nevezzük azt a matematikai programozási feladatot, amelynél a

$$\begin{aligned} A\underline{x} &\leq \underline{b} \\ \underline{x} &\geq \underline{0} \end{aligned}$$

feltételek mellett kell meghatároznunk a

$$\varphi(\underline{x}) = \underline{c}^T \underline{x} + \frac{1}{2}\underline{x}^T D\underline{x}$$

kvadratikus függvény minimumát. Segédváltozók bevezetésével ezt a feladatot mindig át lehet alakítani olyanná, amelynél a feltételek csak egyenlőséget tartalmaznak, azaz az

$$\begin{aligned} A\underline{x} &= \underline{b} \\ \underline{x} &\geq \underline{0} \\ \min \{\varphi&(\underline{x})\} \end{aligned}$$

feladattá. A kvadratikus programozási feladatoknál az $A$ $m \times n$-es mátrix, a $D$ $n \times n$-es mátrix, a $\underline{b}$ $m$-dimeziós, a $\underline{c}$ pedig $n$-dimenziós vektorok mindegyike adott, az $\underline{x}$ $n$-dimeziós vektor az ismeretlen. A $D$ mátrixot, amelyről mindig fel fogjuk tételezni a szimmetrikusságot, hatékonysági mátrixnak nevezzük.

Minthogy a $\varphi(\underline{x})$ célfüggvény a lehetséges megoldások által meghatározott konvex poliéderen folytonos, ezért ha a konvex poliéder korlátos, akkor a feladatnak létezik optimális megoldása. Ha a konvex poliéder nem korlátos, de a felette tekintett $\inf\{\varphi(\underline{x})\}$ véges, akkor bizonyítható, hogy van olyan $\underline{x}^*$ pontja a konvex poliédernek, amelyre a

$$\varphi(\underline{x}^*) = \inf\{\varphi(\underline{x})\}$$

teljesül.

Tekintsük most tehát az

$$\begin{aligned} A\underline{x} &= \underline{b} \\ \underline{x} &\geq \underline{0} \\ \min\{\varphi(\underline{x})\} = \min &\left\{\underline{c}^T\underline{x} + \frac{1}{2}\underline{x}^T D\underline{x}\right\} \end{aligned}$$

kvadratikus programozási feladatot, amelynél legyen a $D$ mátrix pozitív definit. Ez a feltevés erősebb, mint a célfüggvény konvexitásának feltevése, ugyanis a $\varphi(\underline{x})$ függvény már akkor is konvex, ha a $D$ mátrix csak pozitív szemidefinit. Egy ilyen probléma megoldására szolgáló, Wolfe-tól származó algoritmus a következő tételen alapszik:

Ha az

$$\begin{aligned} A\underline{x} &= \underline{b} \\ \underline{x} &\geq \underline{0} \\ \min &\left\{\underline{c}^T\underline{x} + \frac{1}{2}\underline{x}^T D\underline{x}\right\} \end{aligned}$$

feladatnál a $D$ egy pozitív szemidefinit mátrix, és ha létezik egy $\underline{x}$ megoldáshoz olyan $\underline{v} \geq \underline{0}$ $n$-dimenziós, valamint $\underline{u}$ $m$-dimenziós vektor, hogy

$$\begin{aligned} \underline{v}^T\underline{x} &= 0 \\ D\underline{x} - \underline{v} + A^T\underline{u} + \underline{c} &= \underline{0}, \end{aligned}$$

akkor az $\underline{x}$ optimális megoldása a feladatnak.

Így tehát egy kvadratikus feladatnak megkapjuk az optimális megoldását, ha az

$$\begin{aligned} A\underline{x} &= \underline{b} \\ D\underline{x} - \underline{v} + A^T\underline{u} + \underline{c} &= \underline{0} \\ \underline{v}^T\underline{x} &= 0 \\ \underline{x} \geq \underline{0}, &\quad \underline{v} \geq \underline{0} \end{aligned}$$

rendszert megoldjuk. A feltételes szélsőértékszámítási problémánk helyett így egy speciális egyenletrendszer nemnegatív megoldását kell tehát meghatároznunk. A $\underline{v}^T\underline{x} = 0$ miatt ez a rendszer nem lineáris, így ezt sem könnyebb megoldani, mint az eredeti problémát.

Wolfe algoritmusa ezek után a következő lesz:

Jelöljön $\underline{w}$ egy $m$-dimenziós, $\underline{z}$ pedig egy $n$-dimenziós vektort és tekintsük az

$$\begin{gathered} A\underline{x} + \underline{w} = \underline{b} \\ D\underline{x} - \underline{v} + A^T\underline{u} + \underline{c} + \underline{z} = \underline{0} \\ \underline{v}^T\underline{x} = 0 \\ \underline{x} \geq \underline{0}, \quad \underline{w} \geq \underline{0}, \quad \underline{v} \geq \underline{0} \end{gathered}$$

rendszert, ahol $\underline{z}$-re nem írunk elő előjelkorlátozást. Ha ennek meg tudnánk adni egy olyan $\underline{x}_0, \underline{u}_0, \underline{v}_0, \underline{w}_0, \underline{z}_0$ megoldását, amelyben $\underline{w}_0 = \underline{0}, \underline{z}_0 = \underline{0}$, akkor $\underline{x}_0$ optimális megoldása lenne az eredeti problémánknak. (Az új $\underline{w}$ és $\underline{z}$ változók bevezetésére azért volt szükségünk, hogy könnyen meghatározhassuk a rendszer egy bázismegoldását.)

Induljunk ki az előbbi rendszer egy

$$(\underline{x}, \underline{u} = \underline{0},\ \underline{v} = \underline{0},\ \underline{w}, \underline{z})$$

megoldásából, amellyel a $\underline{v}^T\underline{x} = 0$ nyilvánvalóan teljesül. Minthogy a $\underline{z}$-re nem tettünk előjelkorlátozást, vezessük be a

$$\underline{z} = \underline{z}_1 - \underline{z}_2, \quad \underline{z}_1 \geq \underline{0}, \quad \underline{z}_2 \geq \underline{0}$$

jelölést. Ezt és az $\underline{u} = \underline{0}$, $\underline{v} = \underline{0}$ egyenlőségeket felhasználva az

$$\begin{gathered} A\underline{x} + \underline{w} = \underline{b} \\ D\underline{x} + \underline{z}_1 - \underline{z}_2 = -\underline{c} \\ \underline{x} \geq \underline{0}, \quad \underline{w} \geq \underline{0}, \quad \underline{z}_1 \geq \underline{0}, \quad \underline{z}_2 \geq \underline{0} \end{gathered}$$

adódik. A $\underline{b} \geq \underline{0}$ miatt (amit könnyen elérhetünk az eredeti problémánknál) ezen rendszerünknek az

$$\underline{x} = \underline{0}, \quad \underline{w} = \underline{b}, \quad z_{1j} = \begin{cases} -c_j & \text{, ha } c_j < 0 \\ 0 & \text{, ha } c_j \geq 0 \end{cases} \qquad z_{2j} = \begin{cases} 0 & \text{, ha } c_j < 0 \\ c_j & \text{, ha } c_j \geq 0 \end{cases}$$

megengedett bázismegoldása lesz. El szeretnénk érni, hogy $\underline{w} = \underline{0}$, $\underline{z}_1 = \underline{z}_2 = \underline{0}$ teljesüljön. Mindezek miatt az előbbi megoldásokból kiindulva először minimalizáljuk a

$$\sum_{j=1}^{n} w_j$$

célfüggvényt. Ez nyílvánvalóan egy lineáris programozási feladat, amit a szimplex módszer egyik változatával megadhatunk. Minthogy $\sum_{j=1}^{n} w_j \geq 0$, a feladatnak létezik optimális megoldása. A minimum értéke 0 kell, hogy legyen, különben a kiindulási feladatunknak nem lesz megoldása.

Jelölje tehát az előbbi feladat optimális megoldását az

$$(\underline{x}^*, \underline{w}^* = \underline{0}, \underline{z}_1^*, \underline{z}_2^*)$$

és legyen $\underline{z}^* = \underline{z}_1^* - \underline{z}_2^*$. Ez utóbbi vektornak már lehetnek negatív komponensei is. Jelölje $\underline{z}^{(0)}$ a $|z_j^*|, j = 1, 2, \ldots, n$ értékekből képezett vektort. Legyen továbbá $F$ egy olyan $n \times n$-es diagonális mátrix, amelynek $f_{jj}$ eleme -1, ha $z_j^* < 0$; 1, ha $z_j^* \geq 0$. Ezeket a jelöléseket felhasználva az $(\underline{x}^*, \underline{z}^{(0)})$ vektor az

$$A\underline{x} = \underline{b}$$
$$D\underline{x} + F\underline{z} = -\underline{c}$$
$$\underline{x} \geq \underline{0}, \quad \underline{z} \geq \underline{0}$$

rendszernek megoldása, s így az

$$(\underline{x}^*, \underline{u}^* = \underline{0}, \underline{v}^* = \underline{0}, \underline{z}^{(0)})$$

vektor megoldása lesz az

$$A\underline{x} = \underline{b}$$
$$D\underline{x} - \underline{v} + A^T\underline{u} + F\underline{z} = -\underline{c}$$
$$\underline{x} \geq \underline{0}, \quad \underline{v} \geq \underline{0}, \quad \underline{z} \geq \underline{0}$$

feladatnak, amelyre még a $\underline{v}^T\underline{x} = 0$ is teljesül. Határozzuk meg ezután ezen feltételrendszer mellett szimplex módszerrel a $\sum_{j=1}^n z_j$ függvény minimumát, ahol kiindulási bázismegoldásként az

$$(\underline{x}^*, \underline{u}^* = \underline{0}, \underline{v}^* = \underline{0}, \underline{z}^{(0)})$$

vektort használjuk. Ezen probléma megoldása során minden lépésnél a $\underline{v}^T\underline{x} = 0$ egyenlőség teljesülését is szem előtt kell tartani. Ezt úgy érhetjük el, hogy minden egyes transzformációs lépésnél biztosítjuk: ha az $x_k$ bázisváltozó, akkor a $v_k$ nem lehet az, és fordítva. Igazolható, hogy ha a $\underline{c} = \underline{0}$, vagy a $D$ pozitív definit mátrix, akkor a $\underline{z} = \underline{0}$ értékhez legfeljebb $\binom{3n}{n}$ lépésben eljuthatunk. A legutolsó lépésben adódó

$$(x, u, v, \underline{z} = \underline{0})$$

megoldás $\underline{x}$ része pedig optimális megoldása lesz az eredeti kvadratikus feladatunknak.

A Wolfe-algoritmus kiterjeszthető arra az esetre is, ha a $D$ mátrix pozitív szemidefinit, de megoldották ezt a problémát olyan esetekre is, amikor a célfüggvény kvázikonvex, illetve nemkonvex. Közelítő eljárásokat dolgoztak ki olyan kvadratikus programozási feladatokra, amelyeknél a feltételrendszer nemlineáris függvényeket is tartalmaz, illetve amelyeknél nem minden változótól van megkövetelve a nemnegativitási feltétel.

## II.3. Gradiens módszer

Ezzel a módszerrel olyan nemlineáris programozási feladatokat tudunk megoldani, amelyek célfüggvénye folytonosan deriválható és amelyeknél a feltételek által meghatározott konvex halmaz korlátos. Általános esetben a korlátozó feltételeknek nem szükségképpen kell lineárisnak lenniük, elegendő, ha csak konvex halmazt határoznak meg. A módszer során fogjuk használni a következő tételeket:

- Ha az $f(x)$, $\underline{x} \in \mathbb{R}^n$ folytonosan deriválható és az $\underline{x}_0$-ban az $\underline{f}'(\underline{x}_0)$ gradiens vektor nem zérus vektor, akkor létezik olyan $\delta$ pozitív skalár, hogy az
$$f\left(\underline{x}_0 + t\underline{f}'(\underline{x}_0)\right) > f(\underline{x}_0)$$
teljesül minden $0 < t < \delta$ értékre.
- Ha $f(x)$, $\underline{x} \in \mathbb{R}^n$ folytonosan deriválható és az $\underline{x}_0$-ban $\underline{f}'(\underline{x}_0) \neq \underline{0}$, akkor minden olyan $\underline{v}$ esetén, amelyre $\underline{f}'(\underline{x}_0)^T \underline{v} > 0$ teljesül, hogy létezik olyan $\delta > 0$ szám, hogy
$$f\left(\underline{x}_0 + t\underline{v}\right) > f(\underline{x}_0)$$
hacsak $0 < t < \delta$.

A módszerhez szükségünk lesz még a következő definicióra is:

A $\underline{v}^T \cdot \underline{f}'(\underline{x}_0) > 0$ feltételeknek eleget tevő olyan $\underline{v}$ vektorokat, amelyek egyben lehetségesek is, azaz $\underline{x}_0$-ból elmozdulva a $\underline{v}$ irányába van olyan vektor, amely a feltételek által meghatározott halmazban van, „hatékony irányok”-nak nevezzük.

Nyilvánvaló, hogy ha az $\underline{f}'(\underline{x}_0) \neq \underline{0}$ lehetséges irány, akkor az egyben hatékony irány is, emiatt szokás ezt a módszert a „hatékony irányok módszeré”-nek is nevezni.

Legyen most a feladat lineáris feltételrendszerű, azaz egy konvex poliéder felett kell meghatároznunk egy folytonosan deriválható függvény maximumát. Feltesszük, hogy a megoldáshalmaz korlátos, azaz konvex politop. Jelöljük a megoldáshalmazt $K$-val, így a feladat a
$$\max_{k} \{f(\underline{x})\}$$

formában írható fel, ahol $K = \{\underline{x} | A\underline{x} \leq \underline{b}, \underline{x} \geq \underline{0}\}$ korlátos. Legyen $\underline{x}_0 \in K$ tetszőleges. Tegyük fel, hogy $\underline{v}_0 = \underline{f}'(\underline{x}_0) \neq \underline{0}$. Amennyiben a $\underline{v}_0$ lehetséges irány, akkor abban az irányban $\underline{x}_0$-ból elindulva az $f(\underline{x})$ értéke növekedni kezd. Ebben az irányban addig mozdulunk el, amíg a növekedés tart, vagy el nem érjük a $K$ határát. Íly módon olyan $\underline{x}_1$-hez jutunk, amelyre $f(\underline{x}_1) > f(\underline{x}_0)$. Ezt követően $\underline{x}_1$ átveszi az $\underline{x}_0$ szerepét, majd az eljárást annyiszor ismételjük meg, ahányszor csak lehetséges. Abban az esetben, amikor egy adott lépésben az $\underline{f}'(\underline{x}_0)$ gradiens vektor nem lehetséges irány, akkor keresünk egy másik „hatékony irány"-t, amely átveszi a gradiens vektor szerepét. Az eljárást mindaddig folytatjuk, amíg a $K$ olyan pontjához nem jutunk, amelyben már nincs hatékony irány.

Ez a módszer sok tekintetben különbözik a szimplex módszertől. Egyik különbség az, hogy a feltételek által meghatározott halmaznak nem szükségképpen kell konvex politopnak lennie, elegendő, ha csak korlátos konvex halmaz. Ebből következik, hogy a hatékony irányok módszere szélesebb területen alkalmazható. Egy másik különbség, hogy a célfüggvénynek nem kell lineárisnak lennie, csak folytonosan differenciálhatónak. Harmadik különbség, hogy $\underline{x}_0$-nak nem kell bázismegoldásnak lennie. Meg kell azonban említenünk a módszer hátrányait is, amelyek miatt a szimplex módszert alkalmazzák minden olyan esetben, amikor a probléma azzal is megoldható. Egyik ilyen hátránya, hogy a hatékony irányok módszerével általában csak helyi minimumot, vagy maximumot tudunk megállapítani. A másik hátránya az, hogy az iterációs lépések száma nagyon nagy, sőt végtelen nagy is lehet. Ilyenkor a kapott eredmény csak közelítése lesz a tényleges optimumhelynek. Igaz viszont az, hogy a lépések számát növelve az approximáció pontossága tetszés szerint növelhető.

Vizsgáljuk most tehát meg, hogy hogyan néznek ki a módszer lépései az

$$\begin{aligned} A\underline{x} &\leq \underline{b} \\ \underline{x} &\geq \underline{0} \end{aligned}$$

$$\max\{f(\underline{x})\}$$

problémánál, ahol a feltételek egy konvex politopot határoznak meg, az $f(\underline{x})$ pedig folytonosan deriválható. Legyen az $A$ mátrix $m \times n$-es, a $\underline{b}$ vektor $m$-, az $\underline{x}$ pedig $n$-dimenziós vektor. Az

$$A^* = \begin{pmatrix} -E \\ A \end{pmatrix} \text{ mátrix és a } \underline{b}^* = \begin{pmatrix} 0 \\ \underline{b} \end{pmatrix}$$

$m + n$ dimenziós vektor bevezetésével a felétételeket

$$A^*\underline{x} \leq \underline{b}^*$$

alakban is felírhatjuk. Ahhoz, hogy a módszert „beindíthassuk”, ismernünk kell egy olyan $\underline{x}_0$ megoldásvektort, amelyre $\underline{f}'(\underline{x}_0) \neq \underline{0}$. A gyakorlatban elég sokszor ismert egy $\underline{x}_0$ kiindulási megoldás. Ha viszont ilyen nincs, akkor a kétfázisú szimplex módszert alkalmazhatjuk ennek meghatározására. Előfordulhat olyan probléma is, hogy van ugyan egy $\underline{x}_0$ kiindulási megoldásunk, de $\underline{f}'(\underline{x}_0) = \underline{0}$. Ebben az esetben már vagy optimális megoldásnál vagyunk (például ha $f(\underline{x})$ alulról konkáv), vagy pedig más kiindulási megoldást kell keresnünk. Ha már van olyan $\underline{x}_0 \in K$, amelyre $\underline{f}'(\underline{x}_0) \neq \underline{0}$ is teljesül, akkor meg kell határoznunk egy, az $\underline{x}_0$-hoz tartozó hatékony irányt is. Amennyiben $\underline{f}'(\underline{x}_0)$ lehetséges irány, akkor az egyben a leghatékonyabb irány is. Ez nyilván akkor teljesülhet, ha az

$$A^* \left(\underline{x}_0 + \lambda \underline{f}'(\underline{x}_0)\right) \leq \underline{b}^*$$

egyenlőség $\lambda > 0$ mellett is teljesül. Ebből átalakítással a

$$\lambda A^* \underline{f}'(\underline{x}_0) \leq \underline{b}^* - A^* \underline{x}_0$$

egyenlőtlenséghez jutunk, ahol $\underline{b}^* - A^* \underline{x}_0 \geq \underline{0}$. A baloldalon álló $A^* \underline{f}'(\underline{x}_0)$ vektor legalább egy pozitív komponenssel kell, hogy rendelkezzen, ellenkező esetben tetszőleges $\lambda > 0$ mellett teljesülne az egyenlőtlenség, ami a $K$ korlátosságának ellentmondana. Legyen például az $i$-edik komponense pozitív. Ekkor

$$\lambda \underline{e}_i^T A^* \underline{f}'(\underline{x}_0) \leq \underline{e}_i^T (\underline{b}^* - A^* \underline{x}_0)$$

adódik, amiből $\lambda$-ra a

$$\lambda \leq \frac{\underline{e}_i^T (\underline{b}^* - A^* \underline{x}_0)}{\underline{e}_i^T A^* \underline{f}'(\underline{x}_0)}$$

egyenlőtlenséghez jutunk. Így ha a

$$\min_i \left\{ \frac{\underline{e}_i^T (\underline{b}^* - A^* \underline{x}_0)}{\underline{e}_i^T A^* \underline{f}'(\underline{x}_0)} \;\middle|\; \underline{e}_i^T A^* \underline{f}'(\underline{x}_0) > 0 \right\} > 0,$$

akkor az $\underline{f}'(\underline{x}_0)$ gradiens vektor lehetséges irány, s egyben persze hatékony irány is.

Ha az előbbi minimum értéke 0, akkor hatékony irány csak az $\underline{f}'(\underline{x}_0)$-tól különböző valamelyik másik irány lehet. Mielőtt azonban a hatékony irány meghatározására szolgáló módszert ismertetni, megadjuk annak szükséges és elégséges feltételét, hogy az $\underline{x}_0$ megoldáshoz hatékony irány létezzen.

**Tétel.** *Az $\underline{x}_0$ megoldáshoz akkor és csakis akkor tartozik hatékony irány, ha*

$$\max \left\{ \underline{f}'(\underline{x}_0)^T \underline{x} | \underline{x} \in K \right\} > \underline{f}'(\underline{x}_0)^T \underline{x}_0$$

*ahol* $K = \{\underline{x} | A^* \underline{x} \le \underline{b}^*\}$.

Amennyiben tehát a nemzérus $\underline{f}'(\underline{x}_0)$ gradiens vektor nem lehetséges irány, az előbbi tétel szolgál egy hatékony irány meghatározására. Ehhez először meg kell oldani az

$$A^* \underline{x} \le \underline{b}^*$$
$$\max \{\underline{f}'(\underline{x}_0)^T \underline{x}\}$$

lineáris programozási feladatot, amelynek a konvex poliéder korlátossága miatt van optimális megoldása. Jelöljük ezt az optimális megoldást $\widehat{\underline{x}}$-al. Ezzel a következő két eset fordulhat elő:

- $\underline{f}'(\underline{x}_0)^T \widehat{\underline{x}} > \underline{f}'(\underline{x}_0)^T \underline{x}_0$. Ebben az esetben a $\underline{v} = \widehat{\underline{x}} - \underline{x}_0$ hatékony irány lesz.
- $\underline{f}'(\underline{x}_0)^T \widehat{\underline{x}} = \underline{f}'(\underline{x}_0)^T \underline{x}_0$. Ekkor $\underline{x}_0$-hoz nincs hatékony irány. Ilyenkor az $\underline{x}_0$ pontot stacionárius pontnak fogjuk nevezni.

Egy $\underline{x}_0$ stacionárius pont tehát olyan, hogy bármely $\underline{v} = \widehat{\underline{x}} - \underline{x}_0$ vektor esetén az $\underline{v}^T \cdot \underline{f}'(\underline{x}_0) \le 0$ teljesül. Az $\underline{x}_0$ akkor is stacionárius pont, ha $\underline{f}'(\underline{x}_0) = \underline{0}$, de akkor is, ha $\underline{f}'(\underline{x}_0) \ne \underline{0}$, ez az irány nem lehetséges és az $\underline{x}_0$-ban nem is létezik hatékony irány. A stacionárius pont ezen fogalmának hasonló szerepe van az optimumszámításban, mint az analízisbeli stacionárius pontfogalomnak. A definiciónkból ugyanis következik, hogy ha $f(\underline{x})$ folytonosan deriválható, akkor a $K$ felett csak stacionárius pontban lehet globális maximuma.

Ha az $\underline{x}_0$-hoz tartozik valamilyen $\underline{v}$ hatékony irány, akkor abban az irányban haladva a célfüggvény értéke növelhető. Ekkor még azt kell tisztáznunk, hogy meddig mehetünk el a $\underline{v}$ irányában úgy, hogy ne lépjünk ki a megoldáshalmazból és hogy növekedjen a célfüggvény értéke. Ezt a kérdést két paraméter értéke alapján válaszolhatjuk meg. Az egyik paraméter, mondjuk a $\lambda_0$, megmutatja, hogy a $\underline{v}$ irányban menve az $x_0 + \lambda \underline{v}$ vektorok közül melyik lesz az utolsó pont, amely még a $K$-hoz tartozik. Legyen $\lambda_0$ ezen értéke $\lambda_0'$, amelyre

$$\lambda_0' = \min_i \left\{ \frac{\underline{e}_i^T \cdot (\underline{b}^* - A^* \underline{x}_0)}{\underline{e}_i^T A^* \underline{v}} \;\middle|\; \underline{e}_i^T \cdot A^* \underline{v} > 0 \right\}.$$

A másik paraméter, amit $\lambda_1 \ge 0$ fog jelölni, azt mutatja meg, hogy az $f(\underline{x}_0 + \lambda_1 \underline{v})$ függvény az $\{\underline{x}_0 + \lambda_1 \underline{v} | \lambda_1 \ge 0\}$ halmaz mely pontjáig növekszik. Jelöljük $\lambda_1'$-vel $\lambda_1$-nek azt az értékét, amellyel meghatározott $\underline{x}_0 + \lambda_1' \underline{v}$ pontig növekszik az $f(\underline{x})$ értéke.

Ha az említett $\{x_0 + \lambda_1 \underline{v} | \lambda_1 \geq 0\}$ halmazon az $f(\underline{x}_0 + \lambda_1 \underline{v})$ függvény monoton növekedve felülről nem korlátos, akkor $\lambda_1' = \infty$. Ezek után a

$$\lambda^* = \min\{\lambda_0', \lambda_1'\}$$

lesz a legnagyobb $\lambda$ érték, amely mellett $\underline{x}_0 + \lambda^* \underline{v} \in K$ és az $f(\underline{x})$ értéke is növelhető. Ezután az $\underline{x}_1 = \underline{x}_0 + \lambda^* \underline{v}$ veszi át az $\underline{x}_0$ szerepét és azzal folytatjuk tovább az eljárást, amíg lehet.

Az eljárás valamely $\underline{x}_k$ stacionárius pontnál ér véget, ha egyáltalán véges sok lépésben véget ér. Tisztáznunk kell még azt, hogy egy stacionárius pont milyen feltételek mellett lesz szélsőértékhely, illetve ha nem az, akkor hogyan folytassuk tovább az eljárást.

Ebből a célból vezessük be a

$$\underline{v}_k = \underline{f}'(\underline{x}_k)$$
$$K_0 = \{\underline{x} | \underline{x} \in K,\ \underline{v}_k^T \underline{x} = \underline{v}_k^T \underline{x}_k\}$$
$$\widetilde{K} = (K \backslash K_0) \cup \{\underline{x}_k\}$$

jelöléseket. A $K_0$ nem más, mint a

$$\max\{\underline{v}_k^T | \underline{x} \in K\}$$

lineáris programozási feladat optimális pontjainak halmaza.

Az $f(\underline{x})$ célfüggvényről fel fogjuk tételezni a szigorú konvexitást, eredményeinket erre az esetre adjuk meg.

Bebizonyítható, hogy ha $f(\underline{x})$ szigorúan konvex, akkor az $\underline{x}_k$ stacionárius pont csakis akkor jelent maximum helyet, ha a $K_0$-nak csak egy eleme van (maga az $\underline{x}_k$ pont). Ha $K_0$-nak egynél több eleme van, akkor $\underline{x}_k$ nem lehet maximum hely, ilyenkor az iterációt a $K_0$-nak $\underline{x}_k$-tól különböző pontjával kell folytatni, ugyanis egy ilyen ponthoz az $f(\underline{x}_k)$-nál nem kisebb célfüggvényérték tartozik. Ezt az eljárást is addig folytatjuk, amíg lehet.

## II.4. Dinamikus programozás

A dinamikus programozás, amelynek kidolgozása R. Bellmann nevéhez fűződik, elsősorban a gazdasági életben előforduló optimalizálási problémák megoldására alkalmas módszerek egyike. Ezeknél a problémáknál a célfüggvénynek szeparábilisnek, azaz egyváltozós függvények összegeként előállíthatónak kell lennie. Ezen módszer alkalmazása során azt a tényt használjuk ki, hogy az ilyen optimalizálási problémák $n$-lépéses döntési problémaként kezelhetők. Amikor ugyanis egy feladat egy lehetséges megoldását megadjuk, akkor az $x_1, x_2, \ldots, x_n$ változók értékeit választjuk ki a lehetséges értékek közül. Szeparábilis célfüggvény esetén ezt az értékadást megvalósíthatjuk $n$ egymást követő lépésben is, amelyek közül az elsőben az $x_1$, a másodikban az $x_2$, stb., az utolsóban az $x_n$ értékéről döntünk. Ezek a döntések nem függetlenek egymástól: a megelőző döntések sorozata mindig egyértelműen meghatározza a soron következő döntés előtti helyzetet. Az optimális döntéseknél ez méginkább igaz, ez fejeződik ki az ún. optimalizálási tételben:

Egy optimális politika csak optimális alpolitikákból állhat.

(Politikának az $n$ számú lépésben hozható döntések egy lehetséges sorozatát, alpolitikának pedig az első döntéssel kezdődő, egymást követő döntések egy olyan sorozatát nevezzük, amely része valamely politikának.)

Az optimalizálási tétel értelmében tehát egy $n$-lépéses döntési problémához tartozó valamilyen politika csak akkor lehet optimális, ha valamennyi alpolitikája is az. A dinamikus programozás alkalmazása során ezt úgy használjuk fel, hogy az $n$-lépéses probléma megoldásához kisebb lépésszámú döntési problémák optimális megoldásai segítségével jutunk el. Ha például tekintünk egy $k$ lépéses ($2 \leq k \leq n$) problémát, akkor ahhoz, hogy a hozzátartozó politika optimális lehessen, optimálisnak kell lennie az első $k-1$ lépéshez tartozó politikának is. Így a $k$-adik lépésben az optimális megoldást úgy kapjuk meg, hogy a megelőző $k-1$ döntés optimális értékének ismeretében számbavesszük a $k$-adik lépésben lehetséges összes döntést és kiválasztjuk azt, amelyik a $k$-lépéses probléma optimális megoldását adja.

A legáltalánosabb, dinamikus programozási módszerrel megoldható probléma modellje a következő: a

$$G_i(x_1, \ldots, x_n) \leq b_i, \qquad i = 1, 2, \ldots, m$$

feltételek mellett meghatározandó a

$$\max \text{ vagy } \min \left\{ \sum_{k=1}^{n} f_k(x_k) \right\}.$$

Ebben a modellben nem tettünk megszorítást sem a változók, sem a feltételek számát illetően (azaz $m$ és $n$ tetszőleges természetes számok lehetnek), sem pedig a feltételekben szereplő függvényekre. Ez azonban nem jelenti azt, hogy bármilyen alakúak is a $G_i(x_1, \ldots, x_n)$ függvények, a probléma mindig megoldható. Általánosan nem tudjuk meghatározni, hogy milyen $G_i$ függvények esetén lehet ezzel a módszerrel eljutni az optimális megoldáshoz. Biztos azonban, hogy a probléma megoldható, ha a $G_i$ függvények lineárisak. Ha ezek a feltételek nem teljesülnek, akkor a megoldás létezését nem tudjuk minden esetben garantálni. Ugyanakkor még az elvi megoldhatóság esetén sem lehet a gyakorlatban a feltételek és a változók száma tetszőleges nagy, mert a gyakorlati kivitelezésnek korlátot szab az az időtartam, amennyi idő alatt egy ilyen nagyméretű probléma elvileg megoldható.

Ezen jegyzetben az olyan egyfeltételes diszkrét feladat megoldását adjuk meg, amelyben a feltételi függvény lineáris. (Itt a diszkrétség az $x_k$ változókra vonatkozik, azok tehát csak diszkrét értékeket vehetnek fel.) Hasonlóan lehetne vizsgálni az egyfeltételes folytonos esetet is, azonban a változók folytonosságának megengedése újabb problémákat eredményezne az állapotfüggvények értékének meghatározásánál, ami tovább nehezítené a feladat megoldását. A „többfeltételes", „sokváltozós" problémák megoldása, mégha a feladat elvileg meg is oldható, gyakorlatilag kivitelezhetetlen. Így általában csak „kevés" lineáris feltétellel bíró, „kevés" diszkrét változót tartalmazó lehet a feladat, amelyben a változók csak „kevés értéket" vesznek fel, különben gyakorlatilag kivitelezhetetlen a feladat megoldása. (A „kevés" helyett nem lehet egy egyértelműen meghatározható számot mondani.)

Tekintsük tehát a

$$\sum_{j=1}^{n} a_j x_j \le b \qquad a_j > 0$$

$$x_j \ge 0 \text{ egész}, \quad j = 1, 2, \ldots, n$$

$$\max z = \max \left\{ \sum_{j=1}^{n} f_j(x_j) \right\}$$

feladatot. Az általánosság megszorítása nélkül feltehetjük, hogy az $a_j, j = 1, \ldots, n$, $b$ konstansok egészek. Jelölje $z^*$ a feladat optimális célfüggvényértékét, tehát

$$z^* = \max_{x_1,\ldots,x_n} \left\{ \sum_{j=1}^{n} f_j(x_j) \right\}.$$

Minthogy a célfüggvény szeparábilis, ezt a következő alakban is felírhatjuk:

$$z^* = \max_{x_n} \left\{ f_n(x_n) + \max_{x_1,\ldots,x_{n-1}} \left\{ \sum_{j=1}^{n-1} f_j(x_j) \right\} \right\}.$$

Ez utóbbi alapján a $z^*$ értékét tehát a következőképp határozhatjuk meg: kiválasztjuk az $x_n$ egy értékét és rögzítjük, majd a visszamaradó változók felett maximalizáljuk a maradék célfüggvényt. Ezt aztán az $x_n$ minden lehetséges értékére elvégezzük, majd az így kapott összegek közül kiválasztjuk a legnagyobbat, az lesz a $z^*$ értéke. (Látható, hogy ebben az eljárásban felhasználtuk az optimalitási tételt, hiszen $z^*$-ot úgy kaptuk meg, hogy az $n-1$ változós problémánál is optimumot számoltunk.)

Amikor az $x_n$-et valamilyen értéken rögzítettük, akkor a visszamaradó $x_1, x_2, \ldots, x_{n-1}$ változók értékeit olyan nemnegatív egészekre korlátozzuk, amelyek a

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

egyenlőtlenségnek tesznek elehet. Használjuk a

$$z^* = \Lambda_n(\xi_0), \quad \xi_0 = b$$
$$\xi_1 = b - a_n x_n$$
$$\Lambda_{n-1}(\xi_1) = \max_{x_1,\ldots,x_{n-1}} \left\{ \sum_{j=1}^{n-1} f_j(x_j) \right\}$$

jelöléseket, ahol a $\Lambda$ függvényeket állapotfüggvényeknek, a $\xi$ paramétereket pedig állapotparamétereknek nevezzük. Ezek segítségével az eredeti problémánk tehát a

$$\sum_{j=1}^{n-1} a_j x_j \leq \xi_1$$
$$z^* = \Lambda_n(\xi_0) = \max_{x_{n-1}} \{ f_n(x_n) + \Lambda_{n-1}(\xi_1) \}$$

formában írhatjuk fel. A $\Lambda_{n-1}(\xi_1)$ függvényt hasonlóan kaphatjuk meg:

$$\Lambda_{n-1}(\xi_1) = \max_{x_{n-1}} \left\{ f_{n-1}(x_{n-1}) + \max_{x_1,\ldots,x_{n-2}} \left\{ \sum_{j=1}^{n-2} f_j(x_j) \right\} \right\},$$

ahol az $x_1, x_2, \ldots, x_{n-2}$ változóknak ki kell elégíteniük a

$$\sum_{j=1}^{n-2} a_j x_j \leq \xi_1 - a_{n-1} x_{n-1}$$

feltételt. A

$$\xi_1 - a_{n-1} \cdot x_{n-1} = \xi_2 \quad \text{és a}$$
$$\Lambda_{n-2}(\xi_2) = \max_{x_1,\ldots,x_{n-2}} \left\{ \sum_{j=1}^{n-2} f_j(x_j) \right\}$$

jelöléseket használva a

$$\Lambda_{n-1}(\xi_1) = \max_{x_{n-1}} \{f_{n-1}(x_{n-1}) + \Lambda_{n-2}(\xi_2)\}$$

összefüggéshez jutunk. Ezzel az eljárással dolgozva a

$$\Lambda_0(\xi_n) = 0, \quad \xi_{n-k+1} = \xi_{n-k} - a_k x_k$$

jelöléseket felhasználva a

$$\Lambda_k(\xi_{n-k}) = \max_{x_k} \{f_k(x_k) + \Lambda_{k-1}(\xi_{n-k+1})\}$$
$$k = 1, 2, \ldots, n$$

egyenlőségek adódnak. Ezek után a számítási eljárás a következő lesz:

Először kiszámítjuk a $\Lambda_1(\xi_{n-1})$ függvényt. Ezt közvetlenül megkapjuk a $\xi_{n-1}$ minden lehetséges értéke esetén úgy, hogy az $x_1$ értékei a

$$0, 1, 2, \ldots, \left[\frac{\xi_{n-1}}{a_1}\right]$$

egész számok lehetnek. Egyidejűleg meghatározzuk azokat az $x_1(\xi_{n-1})$ értékeket is, amelyek maximálissá teszik az $f_1\big(x_1(\xi_{n-1})\big)$-et. Ezeket $\widehat{x}_1(\xi_{n-1})$-gyel fogjuk jelölni. Az állapotfüggvények definíciója értelmében tehát

$$\Lambda_1(\xi_{n-1}) = f_1\Big(\widehat{x}_1(\xi_{n-1})\Big).$$

Ezek után meghatározzuk a $\Lambda_2(\xi_{n-2})$ értékét minden egyes $\xi_{n-2}$ érték mellett. (Például a $\xi_{n-2}$ egy adott értéke mellett a $\Lambda_2(\xi_{n-2})$ meghatározásához az

$$\begin{gathered} f_2(0) + \Lambda_1(\xi_{n-2}) \\ f_2(1) + \Lambda_1(\xi_{n-2} - a_2) \\ \vdots \\ f_2\left(\left[\tfrac{\xi_{n-2}}{a_2}\right]\right) + \Lambda_1\left(\xi_{n-2} - \left[\tfrac{\xi_{n-2}}{a_2}\right] a_2\right) \end{gathered}$$

mennyiségeket kell kiszámítani és ezek közül a legnagyobb lesz az adott $\xi_{n-2}$ értékhez tartozó $\Lambda_2(\xi_{n-2})$ értéke.) A $\Lambda_2(\xi_{n-2})$ meghatározásával egyidejűleg meghatározzuk az $\widehat{x}_2(\xi_{n-2})$ értékeket is. Ezt az eljárást mindaddig folytatjuk, amíg végezetül megkapjuk a $\Lambda_n(\xi_0) = z^*$ értéket is.

Minden egyes lépésben célszerű a $\xi_{n-k}, \Lambda_k(\xi_{n-k})$ és az $\widehat{x}_k(\xi_{n-k})$ értékeit táblázatba foglalni, ugyanis az egymást követő lépések során mindig szükségünk lesz az előzőleg kiszámított $\Lambda_{k-1}(\xi_{n-k+1})$ értékekre. Előfordulhat az is, hogy a $\xi_{n-k}$ egy adott értéke esetén a $\Lambda_k(\xi_{n-k})$ nem csak egyetlen $\widehat{x}_k(\xi_{n-k})$

értékhez tartozik. Ilyenkor az összes $\widehat{x}_k(\xi_{n-k})$ értéket beírjuk a táblázatba ezekkel biztosítva azt, hogy valamennyi optimális megoldást megkaphassuk.

Miután valamennyi $\Lambda_k(\xi_{n-k})$ és $\widehat{x}_k(\xi_{n-k})$ értéket meghatároztuk ($k = 1, 2, \ldots, n-1$) és ismerjük a $z^* = \Lambda_n(\xi_0)$, valamint a $\widehat{x}_n(\xi_0)$ értékét is, meg kell határoznunk az $(x_1^*, x_2^*, \ldots, x_n^*)$ optimális megoldást is. Ez a következőképpen történhet:

Nyilvánvalóan $s_n^* = \widehat{x}_n(\xi_0) = \widehat{x}_n(b)$.

Az $x_n^*$ értékét felhasználva a $b - a_n x_n^*$ állapotparaméterértékkel egyértelműen meghatározódik az $(n-1)$-edik lépésbeli állapot. Minthogy az $n$-edik döntést megelőző állapotnak – az optimalizálási tétel értelmében – optimálisnak kell lennie, így az ahhoz tartozó célfüggvény értéke is maximális kell, hogy legyen. Ezt az értéket már ismerjük, ez éppen a $\Lambda_{n-1}(b - a_n x_n^*)$. Minthogy ezt az értéket az $\widehat{x}_{n-1}(b - a_n x_n^*)$ szolgáltatja, ezért

$$x_{n-1}^* = \widehat{x}_{n-1}(b - a_n x_n^*).$$

Hasonló meggondolással kapjuk, hogy

$$x_{n-i}^* = \widehat{x}_{n-i}\left(b - \sum_{j=0}^{i-1} a_{n-j} \cdot x_{n-j}^*\right),$$
$$i = 1, 2, \ldots, n-1.$$

## II.5. Sztochasztikus programozás

Ebben a jegyzetben csak vázoljuk a matematikai programozás ezen területét. A téma iránt komolyabban érdeklődőknek az irodalomjegyzékben szereplő, a sztochasztikus programozással foglalkozó munkákat ajánljuk áttanulányozásra.

A sztochasztikus programozás tárgyát olyan feltételes szélsőértékszámítási feladatok képezik, amelyekben a feltételi paraméterek, vagy a megoldás komponensei, esetleg mindkettő valószínűségi változók.

A sztochasztikus szélsőérték-feladatok vizsgálatának, kezelési módjai lényegesen különböznek attól függően, hogy a feladat feltételi paramétereire vonatkozó információt egyszerre, vagy részletekben két, vagy több ütemben kapjuk meg. A sztochasztikus modell felállításakor tehát fontos tudni azt, hogy egyetlen, nem korrigálható döntést kell-e hozni, vagy az információ felhalmozódásának mértéke szerint lehetséges a döntés egyszeri, vagy néhányszori kiigazítása. Ennek megfelelően beszélhetünk egy-, két-, vagy többlépcsős sztochasztikus programozási problémáról.

Amennyiben a feltételi paraméterek valószínűségi változók és ismert az együttes eloszlásuk, akkor a sztochasztikus feladatot vissza lehet vezetni olyan matematikai programozási feladatra, amelynek megoldása a sztochasztikus feladat megoldásával egybeesik. Az így származtatott matematikai programozási feladatot a sztochasztikus programozási feladat determinisztikus ekvivalensének nevezzük.

Az egylépcsős, vagy más elnevezéssel statikus sztochasztikus programozási feladatok természetes analógjai a determinisztikus szélsőérték problémáknak, amelyekben a döntést egyszer hozzuk és azt nem javítjuk. Tekintsük ezek közül például az

$$\begin{gathered} \underline{a}_i^T \underline{x} \geq \beta_i \qquad i = 1, 2, \ldots, m \\ \underline{x} \geq \underline{0} \\ \min \left\{ \underline{c}^T \underline{x} \right\} \end{gathered}$$

feladatot, ahol a $\beta_i$-k valószínűségi változók. Charnes, Cooper és Symonds ezt a

$$\begin{gathered} P\left(\underline{a}_i^T \underline{x} \geq \beta_i\right) \geq 1 - \varepsilon_i \qquad i = 1, 2, \ldots, m \\ \underline{x} \geq \underline{0} \\ \min \left\{ \underline{c}^T \underline{x} \right\} \end{gathered}$$

feladattá fogalmazták át, ahol az $\varepsilon_i > 0$ tetszőlegesen választott kicsiny számok. Ha a $\beta_i$ eloszlásfüggvényét $F_i$ jelöli, akkor a feltételeket az

$$\begin{gathered} F_i\left(\underline{a}_i^T \underline{x}\right) \geq 1 - \varepsilon_i \qquad i = 1, 2, \ldots, m \\ \underline{x} \geq \underline{0} \end{gathered}$$

alakban írhatjuk fel. Egy ilyen egylépcsős sztochasztikus programozási problémát valószínűséggel korlátozott sztochasztikus programozási problémának nevezzük. Az

$$\begin{gathered} F_i\left(\underline{a}_i^T \underline{x}\right) \geq 1 - \varepsilon_i \qquad i = 1, 2, \ldots, m \\ \underline{x} \geq \underline{0} \\ \min \left\{ \underline{c}^T \underline{x} \right\} \end{gathered}$$

feladat lesz az

$$\begin{gathered} \underline{a}_i^T \underline{x} \geq \beta_i \qquad i = 1, 2, \ldots, m \\ \underline{x} \geq \underline{0} \\ \min \left\{ \underline{c}^T \underline{x} \right\} \\ \beta_i\text{-k valószínűségi változók} \end{gathered}$$

sztochasztikus feladat determinisztikus megfelelője.

Az előbbi egylépcsős probléma „gyengesége" az, hogy nem az együttes bekövetkezés valószínűségét írja elő, hanem az egyes események valószínűségeit külön-külön, és nem veszi figyelembe a $\beta_i$ valószínűségi változók közötti kapcsolatot. Prékopa András számos olyan, a gyakorlatban is jól használható modellt konstruált, amelyekből ezeket a hiányosságokat kiküszöbölte. Egyik ilyen modellje például a következő:
megoldandó a

$$\begin{gathered} P(\underline{\beta}_1 \leq A\underline{x} \leq \underline{\beta}_2) \geq \varrho \\ g_1(\underline{x}) \geq 0 \\ \vdots \\ g_m(\underline{x}) \geq 0 \\ \min\{f(\underline{x})\} \end{gathered}$$

feladat, ahol $0 < \varrho < 1$; a $g_1(\underline{x}), \ldots, g_m(\underline{x}), -f(\underline{x})$ függvények pedig konkáv függvények.

A kétlépcsős sztochasztikus problémák fordulnak elő leggyakrabban a sztochasztikus programozás gyakorlati alkalmazásában, ugyanis hiányos információ feltételei közt felvetődő számos tervezési és irányítási problémában célszerű a döntés folyamatát két szakaszra bontani. Az „első lépcső"-ben egy olyan előzetes tervet választunk ki, amely a szükséges előkészítő munkálatok végrehajtását teszi lehetővé. A „második lépcső"-ben azokat az egyensúlytalanságokat kell felszámolni, amelyek a feladat véletlen feltételi paraméterei megvalósult értékeinek a megfigyelése után kerülnek a felszínre. Az „előzetes programot" és a „kompenzációs tervet" oly módon kell összehangolni, hogy ez biztosítsa a feladat megoldásának mindkét szakaszában felmerülő költségek összege várható értékének minimumát. (A megoldandó feladatban az „előzetes program" megválasztása magától érthetődően kell, hogy garantálja egy „kompenzálási terv" létezését.)

A kétlépcsős feladatok általánosításai a többlépcsős (dinamikus) sztochasztikus programozási feladatok. A tervezés és irányítás folyamatában gyakran van ugyanis lehetőség arra, hogy a feltételi paraméterek realizációjának sorozatát folyamatosan megfigyeljük és a tervet megfelelő módon helyesbítsük, ez pedig többlépcsős sztochasztikus programozási feladatot eredményez.

# III. Diszkrét programozás

Ebbe a témakörbe olyan matematikai programozási problémák tartoznak, amelyeknél a változók egy része (akár az összes is) csak véges sok, vagy megszámlálhatóan végtelen sok értéket vehet fel. Mind a korlátozó feltételekben szereplő függvények, mind pedig a célfüggvény lehetnek nemlineárisak is. Általános alakjuk az

$$\begin{aligned} f_i(\underline{x}, \underline{y}) \leq b_i \qquad & i = 1, 2, \ldots, m \\ \underline{y} \geq \underline{0}, \qquad & \underline{x} \in S \end{aligned}$$

feltételek mellett meghatátozandó az

$$f(\underline{x}, \underline{y})$$

függvény maximuma, ahol $\underline{x}$ és $\underline{y}$ $p$ illetve $q$ dimenziós vektorok, az $S$ halmaz pedig valamilyen diszkrét halmaz. A $p > 0$ és $q > 0$ esetén kevert (vegyes) diszkrét programozási, a $p > 0$ és $q = 0$ esetén pedig tiszta diszkrét feladatról beszélünk.

Az irodalomban ezen témakörre szokták még az „egészértékű programozás" elnevezést is használni. Ez az elnevezés azért tekinthető jogosnak, mert minden tiszta diszkrét programozási feladathoz meg lehet konstruálni olyan vele ekvivalens feladatot, amelyben a változók csak egész értékeket vehetnek fel. Ha ugyanis például egy $x_i$ változó az $a_1, a_2, \ldots, a_k$ értékeket veheti fel, akkor az

$$x_i = a_1\delta_1 + a_2\delta_2 + \cdots + a_k\delta_k$$

jelölést bevezetve, ahol $\delta_j$ csak 0 vagy 1 értéket vehet fel úgy, hogy $\sum_{j=1}^{k} d_j = 1$, a problémát visszavezettük egy olyanra, amelyben a diszkrét változók már csak egész értékeket vesznek fel.

Az előbbiek alapján könnyen beláthatjuk azt is, hogy minden diszkrét programozási feladathoz meg lehet konstruálni olyan vele ekvivalens feladatot is, amelyben a diszkrét változók csak 0-t, vagy 1-et vehetnek fel értékül. Az ilyen típusú egészértékű programozási feladatoknak nagy előnyük az, hogy számítógéppel történő megoldásuk során kis memóriakapacitást igényelnek, minthogy a változókat egy-egy bit is reprezentálhatja.

A diszkrét programozási feladatok megoldási módszerei között vannak olyanok, amelyek alkalmazásakor nincs szükségünk a megfelelő folytonos feladat megoldására. Ezeket szokás kombinatorikus módszereknek nevezni. Közéjük tartoznak a leszámlálási algoritmusok, a leszámlálási struktúrák, a korlátozás és szétválasztás módszere.

A továbbiakban ezeket a felsorolt kombinatorikus módszereket fogjuk tárgyalni.

## III.1. Leszámlálási algoritmusok

Tekintsük a $\max_{\underline{x}\in S}\{f(\underline{x})\}$, vagy a $\min_{\underline{x}\in S}\{f(\underline{x})\}$ feladatot, ahol $S$ az $n$-dimenziós, 0-1 komponensű vektorok halmazának valamely részhalmaza.

A leszámlálási algoritmusok lényege az, hogy az $S$ halmaz elemei közül az $S$ számosságához képest lényegesen kisebb számosságú részhalmazát kell előállítani és annak elemeit közvetlenül megvizsgálni. A meg nem vizsgált $S$-beli elemek olyanok lesznek, hogy azokban a célfüggvény értéke maximum feladat esetén nem nagyobb, mint a megvizsgáltakhoz tartozó célfüggvényértékek maximuma. (Minimum feladat esetén természetesen nem kisebb, mint a megvizsgáltakhoz tartozó célfüggvényértékek minimuma.)

Egyik leszámlálási algoritmus a Lowler-Bell algoritmus, amelyet az

$$f_{i1}(\underline{x}) - f_{i2}(\underline{x}) \geq 0 \qquad i = 1, 2, \ldots, m$$
$$\underline{x}^T = (x_1, x_2, \ldots, x_n),\ x_j = 0, \text{ vagy } 1, \qquad j = 1, 2, \ldots, n$$
$$\min\{f(\underline{x})\}$$

feladat megoldására használhatjuk, ahol $f_{i1}(\underline{x})$, $f_{i2}(\underline{x})$ és $f(\underline{x})$ függvények monoton nem csökkenőek. (Egy többváltozós függvényt monoton nemcsökkenőnek, vagy monoton növekvőnek nevezünk akkor, ha mindegyik változójában – mint egyváltozós függvény – monoton nemcsökkenő.)

A feladat megoldására szolgáló algoritmusnál használni fogjuk a következő fogalmat:

Legyenek $\underline{x}$ és $\underline{y}$ az $n$-dimenziós euklédeszi tér tetszőleges elemei. Azt mondjuk, hogy az $\underline{x}$ lexikografikus értelemben kisebb, mint az $\underline{y}$ (vagyis az $\underline{x}$ lexikografikus értelemben megelőzi az $\underline{y}$ vektort), ha

$$x_1 = y_1,\ x_2 = y_2, \ldots, x_{r-1} = y_{r-1},\ x_r < y_r\ ,$$
$$\text{ahol } 1 \leq r \leq n.$$

Ezt $\underline{x} \prec \underline{y}$-nal jelöljük. Ha $\underline{x}$ és $\underline{y}$ komponenesei csak 0-t, vagy 1-et vehetnek fel, akkor ezt a rendezést a következőképpen is definiálhatjuk: minden $\underline{x}$ $n$-dimenziós, 0-1 komponensű vektorhoz hozzárendelve az

$$s(\underline{x}) = x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + \cdots + x_n \cdot 2^0$$

számot azt mondjuk, hogy az $\underline{x}$ lexikografikus értelemben megelőzi az $\underline{y}$-t, ha $s(\underline{x}) < s(\underline{y})$. Ezt a rendezést szokás numerikus rendezésnek is nevezni.

A Lawler-Bell leszámlálási algoritmus a következő lemmán alapszik:

**Lemma.** *Jelöljük $\underline{x}^*$-gal az $\underline{x}$ után a numerikus rendezésben következő első olyan vektort, amelyre $\underline{x} \not< \underline{x}^*$ teljesül, azaz a „parciális rendezés” szerint már nem nagyobb $\underline{x}$-nél.*

Jelölje $\underline{x}^{*-}$ az $\underline{x}^{*}$-ot a numerikus rendezésben közvetlenül megelőző vektort. Végül jelölje $\widehat{\underline{x}}$ a numerikus rendezésben az $\underline{x}$-et megelőző vektorok közül azt, amelyhez az feladatnak az addigi legkisebb célfüggvényértéke adódott. Ekkor, ha az

a) $f(\underline{x}) \geq f(\widehat{\underline{x}})$
b) az $\underline{x}$ vektor a feladatnak megoldása
c) létezik olyan $i \in \{1, 2, \ldots, m\}$, amelyre $f_{i1}(\underline{x}^{*-}) - f_{i2}(\underline{x}) < 0$.

feltételek közül bármelyik teljesül, akkor a numerikus rendezésben az $\underline{x}$ és $\underline{x}^{*}$ között lévő vektorok vagy nem megoldásai a feladatnak, vagy pedig egyik sem lehet jobb megoldása, mint $\widehat{\underline{x}}$ és $\underline{x}$ közül a jobbik.

Ezen lemma alapján tehát+ az eljárást az a); b); és c) esetek bármelyikénél $\underline{x}$-et követően $\underline{x}^{*}$-nál folytathatjuk, hiszen az $\underline{x}$ és $\underline{x}^{*}$ közötti vektorok vagy egyáltalán nem megoldásai a feladatnak, vagy pedig nem tartozik hozzájuk kisebb célfüggvényérték, mint amilyen az $\underline{x}$-hez, vagy az $\widehat{\underline{x}}$ tartozott. Ha azonban az a); b) és c) esetek egyike sem teljesül, akkor az eljárást a numerikus rendezésben $\underline{x}$-et követő vektornál kell folytatnunk.

Az eljárás során többször is előfordulhat, hogy az éppen vizsgált $\underline{x}$ vektorról át kell térnünk az $\underline{x}^{*}$ vektorra. Kérdésként mindezek miatt felmerül az, hogy egy adott $\underline{x}$ vektorhoz hogyan lehet meghatározni az $\underline{x}^{*}$ vektort. Amennyiben az $n$-dimenziós $0-1$ komponensű vektorban létezik olyan $r, q$ indexpár, hogy

$$2 \leq r < q \leq n$$
$$x_{r-1} = 0, \quad x_r = x_{r+1} = \ldots = x_q = 1, \quad x_{q+1} = \ldots = x_n = 0,$$

akkor

$$\begin{aligned} x_i^* &= x_i, \quad \text{ha } i \leq r-2 \\ x_{r-1}^* &= 1 \\ x_j^* &= 0, \quad \text{ha } j \geq r. \end{aligned}$$

Ha pedig ilyen $r, q$ indexpár nem létezik, akkor $\underline{x}^{*}$ sem létezik és ilyenkor $\underline{x}^{*-} = (11\ldots1)^T$.

A leszámlálási algoritmusoknál, így a Lawler-Bell algoritmusnál is, az összes $0-1$ komponensű $n$ dimenziós vektort valamilyen módon előzetesen rendeznünk kell. Ilyen rendezés például a numerikus rendezés, de természetesen bármilyen más olyan rendezést is használhatunk, amelyet a változók sorrendjének felcserélésével és a felcserélt változó sorrend mellett alkalmazott numerikus rendezéssel nyerhetünk. Ezt természetesen az algoritmus használata előtt megtehetjük, de az eljárás során a változók sorrendje rögzített kell, hogy legyen.

Mindezek után a Lawler-Bell algoritmus lépései a következők lesznek.

1. lépés: Kiindulunk az $\underline{x} = \underline{0}$ vektorból. Amennyiben ez megoldás, akkor egyben optimális megoldás is. Ellenkező esetben a 2. lépésnél folytatjuk az eljárást.
2. lépés: Az $\underline{x}$ vektort helyettesítjük a numerikus rendezésben rákövetkezővel és a 3. lépésnél folytatjuk.
3. lépés: Kiszámítjuk a vizsgált $\underline{x}$-hez tartozó $f(\underline{x})$ függvényértéket. Amennyiben $f(\underline{x}) \geq f(\widehat{\underline{x}})$, a lemmánk a) része alapján a 6., különben pedig a 4. lépésnél folytatjuk az eljárást.
4. lépés: Megvizsgáljuk, hogy az $\underline{x}$ vektor megoldása-e a feladatnak. Ha igen, akkor a lemmánk b) része alapján az $\underline{x}$ vektor lesz az $\widehat{\underline{x}}$, a hozzátartozó $f(\underline{x})$ lesz az $\widehat{f} = f(\widehat{\underline{x}})$ és a 6. lépésnél, különben pedig az 5.-nél folytatjuk az algoritmust.
5. lépés: Megvizsgáljuk, hogy a lemmánk c) feltétele teljesül-e valamilyen $i \in \{1, \ldots, m\}$ esetén. Ha igen, akkor a 6., különben pedig a 2. lépésnél folytatjuk.
6. lépés: Megvizsgáljuk, hogy létezik-e $\underline{x}$-hez $\underline{x}^*$. Amennyiben igen, akkor képezzük $\underline{x}^*$-ot, ez veszi át $\underline{x}$ szerepét és a 3. lépésnél folytassuk az eljárást. Ha nem létezik $\underline{x}^*$, akkor vége van az eljárásnak. Ha az algoritmus lépései során találtunk megoldást, akkor az $\widehat{\underline{x}}$ helyén lévő lesz az optimális és $\widehat{f} = f(\widehat{\underline{x}})$ lesz az optimális érték. Amennyiben nem találtunk megoldást, akkor megállapítjuk, hogy a feladat feltételrendszere ellentmondásos.

## III.2. Leszámlálási struktúrák

A leszámlálási algoritmusoknál az összes $0-1$ komponensű vektort az algoritmus beindítása előtt rendeznünk kell és ezt a rendezést nem változtathatjuk meg a leszámlálási algoritmus alkalmazása során sem. Mindezek miatt nem tudjuk kihasználni az alkalmazás közben nyert információkat. Emiatt ugyanaz a leszámlálási algoritmus az egyik feladat esetén gyorsan, a másiknál pedig lassan szolgáltatja a megoldást. A feladat megoldása közben nyert információkat jól tudjuk hasznosítani az ún. leszámlálási strukturáknál.

A leszámlálási struktúrát jellemezhetjük azzal, hogy milyen rögzített változó értékek, vagy más feltételek mellett engedjük meg a nem rögzített változók sorrendjének megváltoztatását. Az adott leszámlálási strukturán alapuló algoritmusról akkor beszélünk, ha a megoldás egyes lépéseiben egy előre megadott elv szerint választjuk ki, hogy hol és hogy milyen permutációt alkalmazunk a változókra. Ez az elv olyan, hogy a feladat megoldása során az addig kapott eredményeket is és a feladat szerkezetét is figyelembe veszi. (Ha a permutációkat, valamint azokat a változó értékeket, amelyek rögzítése mellett hajtjuk végre a permutációkat, előre megadjuk, akkor egy leszámlálási algoritmushoz jutunk.)

A leszámlálási struktúrával minden egyes feladat megoldásakor az $n$-dimenziós $0-1$ komponensű vektorok más és más rendezését választjuk ki, a teljes rendezést pedig csak a feladat megoldása után kapjuk meg.

Tekintsük a

$$\min_{x \in S} \{f(\underline{x})\}$$

feladatot, ahol

$$S \subseteq D_n = \{\underline{x} | \underline{x}^T = (x_1, x_2, \ldots, x_n), x_i = 0 \text{ vagy } 1, i = 1, 2, \ldots, n\}$$

Legyen $\{j_1, j_2, \ldots, j_k\} \subseteq \{1, 2, \ldots, n\}$ és legyenek a $\delta_{j_i}, i = 1, 2, \ldots, k$, 0-val vagy 1-gyel egyenlő rögzített számértékek. A

$$\psi(S) = \{\underline{x} | \underline{x} \in S, x_{j_1} = \delta_{j_1}, \ldots, x_{j_k} = \delta_{j_k}\}.$$

Rendezett pszeudo-megoldásnál akkor beszélünk, ha a rögzített változók sorrendjét is megadjuk. Ezt $\varphi$-vel fogjuk jelölni. Például az előbbi $\psi$ pszeudo-megoldáshoz tartozó egyik rendezett pszeudo-megoldás a $\varphi = (x_{j_1} = \delta_{j_1}, \ldots, x_{j_k} = \delta_{j_k})$. Ez csak egy olyan szimbólum, amelyik kifejezi azt, hogy a változók rögzítését milyen sorrendben hajtottuk végre. Ehhez a $\varphi$ rendezett pszeudo-megoldáshoz tartozó pszeudo-megoldást $\{\varphi\}$-vel fogjuk jelölni. Nyilvánvaló, hogy mindazon rendezett pszeudo-megoldásokhoz, amelyek csak a változók rögzítésének sorrendjében térnek el egymástól, ugyanaz a pszeudo-megoldás fog tartozni.

Ha egy rendezett pszeudo-megoldásban valamely rögzített változó értékét nem változtathatjuk meg, akkor azt kötött változónak fogjuk nevezni és azt aláhúzással fogjuk jelölni. Például a

$$\varphi_1 = (x_{j_1} = \delta_{j_1}, \ldots, x_{j_k} = \delta_{j_k}, \underline{x_{j_{k+1}} = \delta_{j_{k+1}}})$$

rendezett pszeudo-megoldásban az $x_{j_{k+1}}$ rögzített változó egyben kötött változó is. Egy rögzített változó két ok miatt válhat kötötté valamely rendezett pszeudo-megoldásban. Egyik oka lehet az, hogy az őt megelőző rögzítések mellett az adott változó ellenkező értékéhez tartozó rendezett pszeudo-megoldást már megvizsgáltuk a megoldás során. A másik oka pedig az lehet, hogy az őt megelőző rögzítések mellett az adott változó ellenkező értékéhez tartozó pszeudo-megoldásnak $S$-sel alkotott metszete, azaz a $\psi(S)$ halmaz üres halmaz. Tehát például az előbbi $\varphi_1$ rendezett pszeudo-megoldásban az $x_{j_{k+1}}$ változó vagy amiatt kötött, mert a

$$\psi_2(S) = \{\underline{x}|\underline{x} \in S, x_{j_1} = \delta_{j_1}, \ldots, x_{j_k} = \delta_{j_k}, x_{j_{k+1}} = 1 - \delta_{j_{k+1}}\} = \emptyset,$$

vagy pedig mert a

$$\varphi_2 = (x_{j_1} = \delta_{j_1}, \ldots, x_{j_k} = \delta_{j_k}, x_{j_{k+1}} = 1 - \delta_{j_{k+1}})$$

rendezett pszeudo-megoldáshoz tartozó pszeudo-megoldást már vizsgáltuk az algoritmus során. Ilyenkor azt mondjuk, hogy a $\varphi_1$ rendezett pszeudo-megoldás a $\varphi$ közvetlen következménye.

Azt a rendezett pszeudo-megoldást, amelyben egyetlen változó sincs rögzítve, a $\varphi = (\emptyset)$ fogja jelölni. Nyilvánvaló, hogy $\{\varphi\} = \{()\} = D_n$.

Ahhoz, hogy egy leszámlálási strukturát felhasználhassuk a $\min_{\underline{x}\in S} f(\underline{x})$ feladat megoldásához, még bizonyos meggondolásokkal kell élnünk, amelyekben figyelembe tudjuk venni a feladat szerkezetét is. Ezeket az elveket általánosságban a következőképpen fogalmazhatjuk meg a teljesség igénye nélkül.

a) Ha a célfüggvény monoton nemcsökkenő függvény, akkor csak annyi változót rögzítünk 1 értéken, amennyit a korlátozó feltételek teljesítéséhez feltétlenül szükséges. A megmaradt változókat 0 értéken rögzítve lekötjük.

b) Ha egy vizsgált rendezett pszeudo-megoldáshoz tartozó pszeudo-megoldás nem tartalmaz egyetlen olyan $\underline{x}$ vektort sem, amely a korlátozó feltételeknek eleget tenne, akkor a rendezett pszeudo-megoldást „lezárjuk”, azaz csak a rendezett pszeudo-megoldásban szereplő rögzített változók alapján képezzük a következő rendezett pszeudo-megoldást. Nem kell tehát figyelembe venni a szabad változókat, mert ha azok közül bármelyiket is bármilyen értéken rögzítjük, akkor az eredeti pszeudo-megoldás részhalmazához jutunk, amiben tehát vagy nincs megoldása a feladatnak, vagy pedig nincs jobb, mint az eljárás során addig kapott legjobb megoldás.

c) Az $S$-et definiáló feltételekhez minden esetben hozzávesszük az ún. célfüggvény feltételt, amelyet az $f(\underline{x}) \leq k-1$ definiál, ahol $k$ az eljárás során a szóban forgó lépésig kapott legkisebb célfüggvényértéket jelöli. Kiinduláskor a $k$ értékét olyan nagynak kell választani, hogy ez a feltétel minden $S$-beli vektorra teljesüljön, azaz ne jelentsen megszorítást.

A $\min_{\underline{x}\in S} f(\underline{x})$ probléma megoldására szolgáló leszámlálási struktura lépései a következők lesznek.

1. lépés: Az $S$ halmazt definiáló feltételeket kiegészítjük az $f(\underline{x}) \leq k-1$ célfüggvény feltétellel, ahol a $k$ értékét először olyan nagynak választjuk, hogy ez a feltétel ne szűkítse az $S$ halmazt. Az új feltételrendszernek eleget tevő $\underline{x}$ vektorok összességét jelölje $S'$. Kiinduláskor nyilván $S' = S$, különben pedig $S' \subseteq S$. Megfogalmazzuk az algoritmus során figyelembe veendő elveket. Ezt követően tekintjük a $\varphi = (\emptyset)$ rendezett pszeudo-megoldást, amelynek megfelelő pszeudo-megoldás a $D_n$ halmaz lesz.
2. lépés: Képezzük a $\varphi$ közvetlen következményeit, majd a legutolsónak kapottat tekintjük $\varphi$-nek. A közvetlen következmény képzésekor az $S'$-t definiáló feltételrendszert kell tekintenünk. Bár lassabban, de az algoritmus akkor is működik, ha nem veszünk minden közvetlen következményt figyelembe. Mindezek miatt általában csak az „olcsón" (azaz rövid idő alatt) előállítható közvetlen következményeket szokás képezni.
3. lépés: Tekintjük a $\varphi$ rendezett pszeudo-megoldáshoz tartozó pszeudo-megoldásnak az $S'$ halmazzal alkotott metszetét, azaz a $\psi(S')$ halmazt. Amennyiben ez üres, akkor az 5. lépésnél, különben pedig a következőnél folytatjuk az eljárást.
4. lépés: Megnézzük, hogy a $\varphi$ rendezett pszeudo-megoldásban van-e szabad változó. Ha van, akkor megvizsgáljuk, hogy a feladattal kapcsolatban megfogalmazott elvek alapján ezeket, vagy a közülük valahányat le kell-e kötünk valamilyen értéken, vagy sem. Ha igen, akkor ezt megtesszük, majd az így kapott új $\varphi$ rendezett pszeudomegoldással a 2. lépésnél folytatjuk az eljárást. Amennyiben az elvek alapján nem kell lekötéseket tennünk, de $\varphi$-ben van szabad változó, akkor egy szabad változót valamilyen értékén rögzítjük, ezzel a rögzítéssel nyert új $\varphi$ rendezett pszeudo-megoldással a 2. lépésnél folytatjuk az algoritmust. Amennyiben $\varphi$-ben nincs szabad változó, akkor megkaptuk az $S'$ egy elemét, ami a célfüggvény feltétel miatt jobb megoldása a problémánknak, mint amilyent eddig előállítottunk.
5. lépés: Megvizsgáljuk, hogy hogy $\varphi$-ben van-e kötetlen változó. Ha nincs, akkor vége van az eljárásnak. Ha van, akkor $\varphi$-hez a következőképpen képezzük a $\varphi^*$-gal jelölt rendezett pszeudo-megoldást. Tekintjük a $\varphi$-ben lévő utolsó nem kötött rögzített változót. A $\varphi$-ben ezen változó előtt lévő rögzítéseket változatlanul hagyjuk, az utána

következőket elhagyjuk, a vizsgált változó értékét pedig ellenkezőjére változtatjuk és lekötjük. Az így nyert $\varphi^*$ rendezett pszeudomegoldást tekintjük a továbbiakban $\varphi$-nek, amivel a 2. lépésnél folytatjuk a megoldást.

## III.3. Korlátozás és szétválasztás módszere

Tekintsük a

$$\min_{\underline{x}\in S_0} f(\underline{x})$$
$$S_0 \subseteq D_n$$

feladatot. Tegyük fel, hogy adott egy alsó becslési eljárás, amellyel egy tetszőleges $S_i \subseteq S_0$ halmazon meg tudjuk határozni az $f(\underline{x})$ függvény egy alsó korlátját, amelyet $\underline{N}(S_i)$-vel fogunk jelölni. Feltesszük továbbá azt is, hogy adva van egy felbontási kritérium, amelynek segítségével a megoldások valamely $S_j$ részhalmazát diszjunkt részhalmazokra bonthatjuk.

Elsőként állapítsuk meg az $\underline{N}(S_0)$ értéket. Amennyiben ez függvényérték, akkor megtaláltuk az optimális megoldást. Ellenkező esetben az $S_0$ halmazt a felbontási kritérium alapján diszjunkt részhalmazokra bontjuk, majd az alsóbecslési eljárással minden részhalmaz felett meghatározzuk az $f(\underline{x})$ alsó becslését. Kiválasztjuk a legkisebb alsó becsléssel rendelkező részhalmazt. Amennyiben ez az alsó becslés függvényérték, akkor vége van az eljárásnak, hiszen $f(\underline{x})$-nek ez az alsó becslése lesz az optimum érték, az ezt szolgáltató $\underline{x}$ vektor pedig az optimális megoldás. Ellenkező esetben erre a részhalmazra alkalmazzuk a felbontási kritériumot, illetve az így származtatott új részhalmazokra az alsó becslési eljárást. Ezt az eljárást addig folytatjuk, amíg meg nem tudunk határozni egy olyan $\widehat{\underline{x}}$ megoldást, amelyhez tartozó célfüggvényérték megegyezik a $\widehat{\underline{x}}$ vektort tartalmazó részhalmaz felett az $f(\underline{x})$-re vonatkozó alsó becsléssel és ez az alsó becslés a legkisebb. Ez az $\widehat{\underline{x}}$ vektor nyilvánvalóan a feladat optimális megoldása lesz.

Az eljárást hasonóan vihetjük végbe maximum feladat esetén is. A különbség csak annyi, hogy az alsó becslési eljárás helyébe a felső becslési eljárás lép. Ez esetben az eljárás olyan $\widehat{\underline{x}}$ vektornál ér véget, amelynél olyan $f(\widehat{\underline{x}})$ függvényérték adódik, amely megegyezik az $\widehat{\underline{x}}$ vektort tartalmazó részhalmaz felett $f(\underline{x})$ felsőbecslésével, és ez a felsőbecslés a legnagyobb.

Alkalmazzuk a korlátozás és szétválasztás módszert a

$$\max_{\underline{x}\in S_0} f(\underline{x}) = \max_{\underline{x}\in S_0} \left\{ \sum_{j=1}^{n} c_j x_j \right\}$$

feladat megoldására, ahol

$$S_0 = \left\{ \underline{x} \middle| \sum_{j=1}^{n} a_j x_j \leq K, x_j = 0 \vee 1, c_j > 0, a_j > 0, j = 1, \ldots, n, K > 0 \right\}.$$

Az irodalomban az ilyen egyfeltételes, tiszta egészértékű programozási problémát „hátizsák problémának” nevezik. (Egy gyalogtúrázó számára $n$ féle használati tárgy áll rendelkezésre, amelyek közül az $i$-edik súlya $a_i$, használati értéke pedig $c_i$. Ha az elvihető tárgyak maximális összsúlyát $K$ jelöli, akkor nyilván azokat a tárgyakat kell magával vinnie, amelyek összsúlya nem haladja meg a $K$ értéket és összhasználati értékük maximális. Ez a probléma pontosan az előbbi feladattal modellezhető.)

Az általánosság megszorítása nélkül feltehetjük, hogy az $x_1, x_2, \ldots, x_n$ változók már úgy vannak indexelve, hogy a

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$$

teljesül. A felbontási kritériumot a következőképpen definiálhatjuk: Kiválasztjuk a felbontandó $S_i$ részhalmazban lévő $\underline{x}$ vektorok legkisebb indexű szabad változóját, majd az $S_i$-t ennek alapján két részre osztjuk. Az egyikbe tartoznak azok az $S_i$-beli vektorok, amelyeknél az említett szabad változó 1-et, a másikba pedig azok, amelyeknél 0-t vesz fel értékül.

Meg kell még adnunk a részhalmazokhoz tartozó felsőbecslések megállapítására szolgáló eljárást is. Egy

$$S_i = \{\underline{x} | \underline{x} \in S_0, x_1 = \delta_1, \ldots, x_s = \delta_s\}$$

részhalmazhoz tartozó $\overline{N}(S_i)$ felsőbecslés meg fog egyezni a

$$\sum_{j=s+1}^{n} a_j x_j \leq K - \sum_{i=1}^{s} a_i \delta_i$$

$$\max \left\{ \sum_{j=1}^{s} c_j \delta_j + \sum_{i=s+1}^{n} c_i x_i \right\}$$

feladat célfüggvényének felső becslésével:

$$\overline{N} = \sum_{j=1}^{s} c_j \delta_j + \sum_{j=s+1}^{q} c_j + \frac{K - \sum_{j=1}^{s} a_j \delta_j - \sum_{j=s+1}^{q} a_j}{a_{q+1}} c_{q+1},$$

ahol a $q$ index meghatározására a

$$\sum_{j=s+1}^{q} a_j \leq K - \sum_{j=1}^{s} a_j\delta_j < \sum_{j=s+1}^{q+1} a_j$$

egyenlőtlenségrendszer szolgál.

Ha egy $S_i = \{\underline{x}|\underline{x} \in S_0, x_1 = \delta_1, \ldots, x_n = \delta_n\}$ halmazban a rögzített változók a $K$ kapacitást annyira kihasználják, hogy az $x_{s+1}, \ldots, x_{r-1}(r-1 \leq n)$ szabad változók már csak 0-t vehetnek fel értékül, mert

$$a_j > K - \sum_{i=1}^{s} a_i\delta_i, \qquad j = s+1, \ldots, r-1,$$

akkor az $\overline{N}(S_i)$ meghatározása előtt ezeket a változókat 0 szinten rögzítjük.

Egy adott lépésben azt az $S_i$ halmazt kell tovább bontani, amelyhez a legnagyobb felső becslés tartozik.

A felsőbecslési eljárással kapcsolatosan megjegyezzük, hogy a lehetségesek közül azt választjuk, amely a többihez képest a legpontosabb felsőbecslést szolgáltatja. Alapvető követelmény a felsőbecslési eljárással szemben az, hogy egyelemű részhalmaz esetén függvényérték legyen a felsőbecslés.

# IV. Számítógépes programcsomagok a lineáris programozási feladatok megoldására

Az Operációkutatás az alkalmazott matematika egyik legdinamikusabban fejlődő ága. A gyakorlatban felmerülő problémák újabb és újabb megoldási módszerek kidolgozását igénylik. A bonyolult operációkutatási problémák megoldásában fontos szerepe van a mérnök, a közgazdász szakambereknek, akik a megoldandó műszaki-gazdasági problémákat jól ismerik; a matematikusoknak, akik a problémák matematikai modelljét szolgáltatják; a statisztikusoknak, akik az adatokat összegyűjtik, illetve rendszerezik és a számítástechnikai szakembereknek, akik a problémák megoldását a számítógép segítségével meghatározzák.

Ebben a fejezetben olyan programcsomagokat tárgyalunk, amelyek könynyen hozzáférhetők, általában több operációkutatási problématipus megoldására használhatók, könnyen kezelhetők, a kapott eredmények pedig széleskörű felhasználást biztosítanak az alkalmazott modell kiértékelésekor, vizsgálatakor. A SOLVER-t és a WinGULF-ot elsősorban lineáris, hiperbolikus és diszkrét problémák megoldására célszerű használni. A SAS/OR ugyanakkor minden operációkutatási problémára tartalmaz megoldási módszert, bár használata komolyabb előtanulmányozást igényel. Ezen fejezetben az első két programcsomagot mutatjuk be a teljességre nem törekedve.

## IV.1. A SOLVER

A Microsoft EXCEL 5.0 táblázatkezelő szoftver optimalizálását végző makrója a SOLVER. Ez természetesen nemcsak optimalizálási feladatok megoldására alkalmas, hanem arra is, amikor a (cél)függvénynek egy konkrét értékét szeretnénk elérni, továbbá használhatjuk egyenletek és egyenletrendszerek megoldására is.

A szélsőérték számításkor feltételes és feltétel nélküli problémák egyaránt megoldhatók vele. Az optimalizálási problémák célfüggvénye és korlátozó feltételei lehetnek lineárisak és nemlineárisak is. A lineáris programozási feladatok megoldására a SOLVER a szimplex módszert használja. A lineáris egyenletrendszer mogoldását pivotálással (bázistranszformációval) végzi. A nemlineáris problémákat a hatékony irányok módszerével oldja meg.

A SOLVER-ben használt paraméterek értékétől függően ezzel a programcsomaggal többek között lineáris, nemlineáris illetve egész értékű programozási problémákat oldhatunk meg. Az egész értékű programozási feladatok megoldására a „korlátozás és szétválasztás" módszert használja ez a programcsomag. A paraméterek beállításával az is elérhető, hogy a megoldás során az egyes lépésekben kapott eredményeket is megismerhessük. A feladatot megoldva eredmény, érzékenység és határok jelentést is kérhetünk.

Az általunk használatos SOLVER-rel 200 döntési változónál nem többet igénylő problémákat tudunk kezelni, de létezik nagyméretű feladatok megoldására alkalmas változata is.

A SOLVER-t az ESZKÖZÖK menü SOLVER parancsával lehet elindítani. Ha az ESZKÖZÖK menüben nem szerepel a SOLVER parancs, akkor az ESZKÖZÖK menü MAKROBEÉPÜLŐ parancsa segítségével a beépíthető makrók megtekinthetők. Ha itt szerepel a SOLVER, akkor azt kijelöljük, beépítjük, ezt követően pedig használjuk.

Az operációkutatási probléma megoldásakor első lépésként meg kell terveznünk, hogy a döntési változók, a korlátozó feltételek és a célfüggvény tárolására mely cellákat fogjuk használni. A döntési változókat tartalmazó cellákat módosuló celláknak szokás nevezni. Ezek száma nem haladhatja meg a 200-at. A célfüggvényt tartalmazó cellát, amelyből csak egyetlen lehet és amely csak képletet tartalmazhat, célcellának nevezzük. A célcellában lévő képlet a módosuló celláktól függő képlet kell, hogy legyen. A korlátozó feltételekből maximálisan 500 adható meg. Ezeket a feltételeket többféleképpen is definiálhatjuk a SOLVER-ben. Egyik lehetőség az, amikor a feltételeket definiáló, a feltételek bal oldalán szereplő képleteket egy-egy cellába beírjuk.

A SOLVER nem engedi meg, hogy a feladat megoldása során 1000 cellánál többet használjunk.

Miután a feladatot rögzítettük, az ESZKÖZÖK/SOLVER menüponttal a feladat megoldása elindítható. Az indítást követően a képernyőn megjelenik a SOLVER PARAMÉTEREK párbeszédablak. Ennek segítségével adhatjuk meg, hogy mely cellák lesznek a módosuló cellák, melyik lesz a célcella és hol lesznek a korlátozó feltételek. Ezen párbeszédablak célcella mezőjében kell megadni a célfüggvényt tartalmazó cella címét, hivatkozását. Ezt vagy beírással, vagy pedig egérrel való kijelöléssel tehetjük meg. A LEGYEN mezőben dönthetünk arról, hogy maximum, vagy minimum feladatot akarunk-e megoldani, vagy pedig a célfüggvény egy konkrét értékét akarjuk-e elérni. A MÓDOSULÓ CELLÁK mezőben a döntési változók celláit kell megadni. Ezt elvégezhetjük beírással, egérrel való kijelöléssel, vagy az AJÁNLAT gomb segítségével, amely használatakor a SOLVER-re bízzuk a módosuló cellák kijelölését. A KORLÁTOZÓ FELTÉTELEK ablakban adhatjuk meg a feltételeket. A FELVESZ gomb megnyomására megjelenik a KORLÁTOZÓ FELTÉTEL FELVÉTELE párbeszédablak. A CELLAHIVATKOZÁS mezőben a korlátozó feltétel baloldalát tartalmazó cellának a címét kell megadni beírással, vagy egérrel való kijelöléssel. A középen lévő legördülő menü segítségével választhatjuk ki a feltételnek megfelelő relációt. Az INT reláció megadásával lehet biztosítani a döntési változók egészértékűségét. A KORLÁTOZÓ FELTÉTEL mezőben kell megadni a feltétel jobboldalát. Itt megadhatunk egy számértéket, a korlátot tartalmazó cella címét, vagy egy kifejezést. Miután megadtunk egy korlátozó feltételt, a FELVESZ gomb lenyomásával ez a feltétel bekerül a feltételek közé. Az összes feltétel felvitele után az OK gombbal térhetünk vissza a SOLVER PARAMÉTEREK párbeszédablakhoz. A BEÁLLÍTÁS gomb megnyomásával a SOLVER BEÁLLÍTÁSOK párbeszédablak jelenik meg, amlynek segítségével a probléma megoldása során a SOLVER által használt paramétereket lehet megváltoztatni. Itt dönthetünk többek között arról is, hogy lineáris programozási feladatot oldunk-e meg, vagy pedig más tipusú a megoldandó feladat. A LÉPÉSENKÉNT KIJELZI beállítás lehetővé teszi, hogy a megoldás egyes lépéseinél megálljon a program futása és kiíródjanak a részeredmények. Az ALAPHELYZET gomb igénybevételével elérhetjük, hogy a SOLVER visszaállítsa a paramétereket alaphelyzetbe.

Amikor a FÁJL/MENTÉS parancsot kiadjuk, akkor az EXCEL a SOLVER PARAMÉTEREK párbeszédablak legutolsó beállításait a munkalaphoz csatolja és a munkafüzet mentésekor megőrzi. A munkafüzet legközelebbi megnyitásakor és a SOLVER elndításakor az egyes munkalapok beállításai automatikusan betöltődnek.

A MEGOLDÁS gomb megnyomására a SOLVER elindítja a probléma megoldását. A problémát megoldva megjelenik a SOLVER EREDMÉNYEK párbeszédpanel és ennek a tetején a SOLVER valamelyik végrehajtási üzenete. Ezen párbeszédpanel megjelenésével egyidejűleg megjelennek a munkalapon a módosuló cellák, a célcella és a feltételcellák megoldáshoz tartozó értékei.

Amennyiben a SOLVER megoldást talált, akkor az eredményekről különböző jelentések készíthetők. Ezeket az EXCEL a munkafüzet egy-egy munkalapján jelentethetjük meg, amiket ki is nyomtathatunk.

Az eredményjelentés a célcellamezőben megadott cellát és a módosuló cellákat sorolja fel feltüntetve azok eredeti és végső értékét. Ezen jelentésben szerepelnek a korlátozó feltételek és azok adatai is. Az ÁLLAPOT oszlopban az ÉPPEN azt jelenti, hogy a feltétel egyenlőséggel teljesült, a BŐVEN pedig azt jelzi, hogy a fetétel két oldala nem egyezik meg. Az ELTÉRÉS oszlop a felételel két oldalának a különbségét mutatja.

Az érzékenység jelentés azt mutatja be, hogy a célcella mezőben megadott képlet vagy a korlátozó feltételek kis változásaira a megoldás mennyire érzékeny. A lineáris programozási feladatok esetében a MÓDOSULÓ CELLÁK címszó alatt a célfüggvény érzékenységvizsgálatának, a KORLÁTOZÓ FELTÉTELEK címszó alatt pedig a jobboldal érzékenységvizsgálat eredményét közli a SOLVER. A MEGENGEDHETŐ NÖVEKEDÉS és CSÖKKENÉS a célfüggvény együtthatók illetve a jobboldal változásának megengedhető mértékét jelenti. Az ÁRNYÉKÁR a feltétel jobb oldalának egységnyi növekedésére eső célfüggvény változását mutatja.

A határok jelentés a célcella mezőben megadott cellát és a módosuló cellákat sorolja fel, valamint megadja értéküket, alsó és felső korlátjukat, valamint a célfüggvény értékét. Az ALSÓ HATÁR egy változó cellában lévő legkisebb felvehető érték akkor, amikor az összes többi változó cella értéke rögzített és teljesíti a feltételeket. A FELSŐ HATÁR hasonló feltételek mellett a változó cella által felvehető legnagyobb értéket jelenti. A CÉLEREDMÉNY a célcella mezőben megadott cella értéke a változó cella alsó iletve felső határértékénél.

## IV.2. WinGULF

Ez a programcsomag az internetről szabadon letölthető. Többek között alkalmazható lineáris, hiperbolikus programozási feladatok megoldására.

Egy lineáris programozási probléma megoldásához indítsuk el a WinGULF programcsomagot és nyissunk meg egy új feladat-ablakot (FILE-NEW).

A képernyőn ennek hatására megjelenő ablak alsó részében található status sorban állítsuk át a FROW és FCOL értékét 0-ra. Ezzel olyan munkafelülethez jutunk, amelyen minden pozició elérhető és szerkeszthető állapotba kerül. Az adatok bevitele után a szimplex módszert a RUN menüpont segítségével indíthatjuk el. Ügyelni kell arra, hogy a statussorban lévő cél MAX legyen.

Ha a program kiszámította a feladat optimális megoldását, akkor a képernyőn olyan párbeszédablak jelenik meg, amelyben információk találhatók az eljárás menetéről illetve a kapott eredményről. Ezen ablak jobb alsó részén lévő MAKE REPORT pont segítségével egy részletes riportot jeleníthetünk meg a feladattal kapcsolatban. A riport első részében olyan általános információk kerülnek megadásra, mint a futtatás dátuma, a feladat neve és tipusa, a célfüggvény maximumát, vagy minimumát határoztuk-e meg, hogy milyen módszert használtunk a feladat megoldására, valamint, hogy milyen méretű volt a feladat. A riport következő része az optimális megoldás kereséséről szóló olyan statisztikai információkat tartalmaz, hogy mikor indult a program, az egyes fázisok iterációiban mekkora volt a célfüggvény értéke, valamint mennyi az iterációk száma. A következő, az Aktivities című rész tartalmazza az optimális megoldás leírását. A Value oszlop tartalmazza a változók értékeit az optimális megoldásban.

A riport a Constraints címmel arról ad részletes információt, hogy a feltételrendszer milyen összetételű, valamint arról, hogy bizonyos jobboldali értékek változtatásával miként változik az optimum értéke. Az első két oszlopban szerepelnek a feladat feltételeinek a sorszámai, illetve azonosítói. A harmadik oszlopban a Less (kisebb-egyenlő), az Equal (egyenlő) és a Greater (nagyobb-egyenlő) relációk fordulhatnak elő. A Slack oszlop a feltételek bal és jobboldala különbségeit adja meg az optimális megoldás esetében.

A riport utolsó két fejezete a feladat érzékenységvizsgálatát tartalmazza. A Stability for Objective részben találhatjuk meg az optimális megoldás stabilitási tartományait a célfüggvényben szereplő együtthatókra vonatkozóan, míg a riport utolsó fejezete Stability for constraints címmel arról ad tájékoztatást, hogy az optimális megoldás mennyire érzékeny a jobboldali értékek megváltoztatására.

Mintapéldát és annak eredményeit az Operációkutatás 1. példatárban találhat az olvasó.

# Irodalomjegyzék

[1] *Prékopa András*, Lineáris programozás, Bolyai J. Matematikai Társulat kiadványa, 1968

[2] *Kovács László Béla*, A diszkrét programozás kombinatorikus módszerei, Bolyai J. Matematikai Társulat kiadványa, 1969

[3] *Martos Béla*, Hiperbolikus programozás, Matematikai Kutató Intézet Közleményei, 5. kötet.

[4] *Dr. Nagy Tamás*, Operációkutatás, Miskolci Egyetemi Kiadó, 1999

[5] *Bajalinov Erik - Imreh Balázs*, Operációkutatás, Polygon, Szeged, 2001

[6] *Wolfe*, The Simplex Method for Quadratic Programming, Econometria, 1959, 382–398