http://www.math.bme.hu/~szantai

Az operációkutatás matematikai módszerei

Szántai Tamás

Bővített óravázlat

Budapest
1999

A bővített óravázlatot PRÉKOPA ANDRÁSnak a Bolyai János Matematikai Társulat kiadásában 1968-ban 440 példányban megjelent lineáris programozás jegyzete alapján készítettem (lásd [12]). Ez a jegyzet egy matematikus hallgatóknak tartott tanfolyam anyagából készült, így értelemszerűen több matematikai elméletet és részletes bizonyításokat is tartalmaz. Ezek jelentős része itt nem jelenik meg. Néhány, a tárgyalt tananyagot nem matematikus hallgatók számára is könnyebben érthetővé tevő szemléletmódot PRÉKOPA ANDRÁS két további dolgozatából vettem át (lásd [13] és [14]).

1. A lineáris programozás feladata, standard alakra transzformálás

1.1. A lineáris programozás feladata

A lineáris programozás feladata röviden úgy fogalmazható meg, hogy lineáris korlátozó feltételek mellett keresendő egy lineáris függvény (célfüggvény) szélsőértéke (minimuma vagy maximuma).

Néhány példa olyan gyakorlati problémára, amely matematikai modellezése ilyen típusú feladatra vezet.

Példa. Termékösszetétel optimalizálás

Tekintsünk egy gyárat, vagy annak egy jól körülhatárolható részlegét, amely n-féle terméket gyárt. Tegyük fel, hogy a termékek gyártásához m-féle erőforrás szükséges.

Legyenek ismertek a gyártási folyamat következő jellemzői:

a_{ij} - az i-edik erőforrásból a j-edik termék egységének az előállításához szükséges mennyiség,

b_i - az i-edik erőforrásból az optimalizálási időszakasz alatt rendelkezésre álló mennyiség,

c_j - a j-edik termék egységének a gyártási haszna.

A termékösszetétel optimalizálása ekkor abból áll, hogy az egyes gyártható termékekből annyit akarunk gyártani, amennyit a rendelkezésre álló erőforrások még lehetővé tesznek, miközben összességében a lehető legnagyobb gyártási hasznot kívánjuk elérni. Az így megfogalmazott probléma matematikai modellezéséhez be kell még vezetni az x_j döntési változókat:

x_j - a j-edik termékből az optimalizálási időszakasz alatt gyártandó mennyiség.

A fenti felsorolásokban az i index 1-től m-ig, a j index pedig 1-től n-ig fut.

Az így megfogalmazott problémát matematikailag az alábbi lineáris programozási feladattal lehet leírni:

$$
\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq & b_2 \\
\vdots & & \vdots & & \ddots & & \vdots & & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq & b_m \\
x_1 \geq 0, & & x_2 \geq 0, & & \dots & & x_n \geq 0 & & \\
(c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_{n)} & \rightarrow & \max
\end{array} \tag{1}
$$

Példa. Takarmányösszetétel optimalizálás

Legyen adott egy szarvasmarha állomány, amely takarmányozásához álljon rendelkezésünkre n-féle takarmány. Tegyük fel, hogy a takarmányozása során m-féle tápanyagból kell előírt mennyiséget mindenképpen juttatni az állatoknak.

Legyenek ismertek az alábbi adatok:

a_{ij} - a j-edik takarmányféleség egységében az i-edik tápanyag mennyisége,

b_i - az i-edik tápanyagból az optimalizálási időszak alatt mindenképpen elfogyasztandó mennyiség,

c_j -a j-edik takarmányféleség egységének bekerülési költsége.

A takarmányösszetétel optimalizálása ekkor abban áll, hogy a rendelkezésre álló takarmányféleségekből egy olyan keveréket állítsunk össze, amely a szarvasmarha állomány számára a figyelembe vett tápanyagok mindegyikéből tartalmazza a takarmányozási időszak alatt feltétlen kívánatos mennyiséget, miközben az összes takarmány bekerülési költsége a lehető legkisebb legyen. A probléma matematikai modelljének a leírásához most is be kell vezetni az x_j döntési változókat:

x_j - a j-edik takarmányféleségből az optimalizálási időszak alatt a szarvasmarha állománynak adandó mennyiség.

A fenti felsorolásokban ismét az i index 1-től m-ig, a j index pedig 1-től n-ig fut.

A fent megfogalmazott problémát most matematikailag az alábbi lineáris programozási feladattal lehet leírni:

$$
\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \geq & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \geq & b_2 \\
\vdots & & \vdots & & \ddots & & \vdots & & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \geq & b_m \\
x_1 \geq 0, & & x_2 \geq 0, & & \dots & & x_n \geq 0 & & \\
(c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_{n)} & \rightarrow & \min
\end{array} \tag{2}
$$

Vegyük észre, hogy az (2) feladat csak annyiban különbözik az (1) feladattól, hogy a kisebb vagy egyenlő típusú feltételek helyett nagyobb vagy egyenlő típusú feltételek szerepelnek és benne a célfüggvényt nem maximalizálni, hanem minimalizálni kell.

Megjegyezzük, hogy az (2) feladat esetén volna értelme akár kisebb vagy egyenlő típusú feltételeknek is, hiszen ezek olyan korlátozásokat írnának elő, hogy valamely tápanyagból egy adott mennyiségnél többet nem kaphat a szarvasmarha állomány. Sőt kifejezetten a korlátozások közé kívánkozik egy további, egyenlőség típusú feltétel, amely azt írná elő, hogy összesen milyen mennyiségű takarmányt kívánunk az állatoknak adni. A következő példa éppen ebben a tulajdonságban fog különbözni az előző kettőtől, benne ugyanis minden lineáris korlátozó feltétel értelemszerűen egyenlőség alakú kell, hogy legyen.

Példa. Szállítási feladat

Legyen adott m telephely és n felvevőhely. Tegyük fel, hogy a felvevőhelyeknek ugyanolyan árúra van szükségük, mint amilyen árút a telephelyeken tárolnak. Tegyük fel, hogy bármely telephelyről bármely felvevőhelyre lehetséges az árú szállítása, de természetesen más és más szállítási költséggel. A telephelyeken lévő árúkészletek mennyiségeire, a felvevőhelyek igényeinek a mennyiségeire, valamint a szállítási egységköltségekre vezessük be az alábbi jelöléseket:

a_i - az i-edik telephelyen lévő árúkészlet mennyisége,

b_j - a j-edik felvevőhely igényének a mennyisége,

c_{ij} - egységnyi árúnak az i-edik telephelyről a j-edik felvevőhelyre történő szállítási költsége.

Feltesszük még, hogy a telephelyeken összesen rendelkezésre álló árúmennyiség legalább annyi, mint a felvevőhelyek összes igénye, hiszen különben valamelyik felvevőhelyen mindenképpen hiány mutatkozna, amely költsége definiálásának a problémájával nem kívánunk jelenleg foglalkozni. Ekkor viszont azt is feltehetjük, hogy a telephelyek összkészlete pontosan egyenlő a felvevőhelyek összigényével, hiszen ellenkező esetben bevezethetnénk egy ún. virtuális felvevőhelyet, amely igénye éppen az összkészlet és az összigény különbsége és amelyre bármelyik telephelyről ingyenes a szállítás. A virtuális felvevőhelyre történő szállítást pedig úgy tekinthetnénk, hogy a szállítandó mennyiséget a telephelyen hagyjuk. Ekkor az új feladat nyilvánvalóan ekvivalens lenne az előzővel.

A szállítás optimalizálásának a problémája ekkor abban áll, hogy minimális szállítási összköltséggel elégítsük ki a felvevőhelyek mindegyikének az összes igényét, miközben minden telephelyről az összes készletet elszállítjuk. A probléma matematikai modelljének a felírásához most az x_{ij} döntési változókat célszerű bevezetni az alábbi módon:

x_{ij} - az i-edik telephelyről a j-edik felvevőhelyre szállítandó árú mennyisége.

Mint eddig mindig, most is a fenti felsorolásokban az i index 1-től m-ig, a j index pedig 1-től n-ig fut.

A fent megfogalmazott problémát most matematikailag az alábbi lineáris programozási feladattal lehet leírni:

$$
\begin{array}{ccccccccc}
x_{11}\,+ & \ldots + x_{1n} & & & & & = & a_1 \\
 & & \ddots & & & & \vdots & \\
 & & & x_{m1}\,+ & \ldots & + x_{mn} & = & a_m \\
x_{11}\,+ & & \ldots + x_{m1} & & & & = & b_1 \\
 & \ddots & \ddots & & \ddots & & \vdots & \\
 & x_{1n}\,+ & & & \ldots & + x_{mn} & = & b_n \\
x_{11} \geq 0, & & \ldots & & & x_{mn} \geq 0 & & \\
(c_{11}x_{11}\,+ & & \ldots & & & + c_{mn}x_{mn}) & \rightarrow & \min
\end{array} \tag{3}
$$

Meglehet, hogy első ránézésre zavaró lehet az x_{ij} döntési változók kettős indexezése, azonban ettől eltekintve az (3) feladat ugyanolyan lineáris programozási feladat, mint az (1) és az (2) feladatok voltak. Látható, hogy most minden lineáris korlátozó feltétel a megoldani kívánt problémából adódóan egyenlőség alakú.

1.2. Standard alakra transzformálás

Ahhoz, hogy a lineáris programozási feladatra megoldó algoritmust tudjunk kidolgozni, mindenekelőtt szükséges az, hogy a különböző lehetséges alakokat egységessé tegyük. Ehhez nyújt segítséget a standard alak fogalma. Ezt a következőképpen definiáljuk:

Definíció. Azt mondjuk, hogy a lineáris programozási feladat standard alakú, ha minden lineáris korlátozó feltétele egyenlőség alakú, minden változójára elő van írva a nemnegativitási korlátozás és a célfüggvénye maximalizálandó:

$$
\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 \\
\vdots & & \vdots & & \ddots & & \vdots & & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m \\
x_1 \geq 0, & & x_2 \geq 0, & & \ldots & & x_n \geq 0 & & \\
(c_1x_1 & + & c_2x_2 & + & \ldots & + & c_nx_{n)} & \rightarrow & \max
\end{array}
$$

Segédtétel. Minden lineáris programozási feladat ekvivalens átalakításokkal standard alakra transzformálható.

Bizonyítás. Két lineáris programozási feladatot akkor tekintünk ekvivalensnek, ha az egyik optimális megoldásának az ismeretében a másik optimális megoldásást közvetlenül meg tudjuk adni és ugyanez fordítva is igaz.

– *Kisebb vagy egyenlő alakú korlátozó feltétel egyenlőséggé alakítása.*

Tegyük fel, hogy az i-edik feltétel kisebb vagy egyenlő alakú:

$$a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i.$$

Ez a korlátozó feltétel egy $x^{(s)}_{n+i}$ nemnegatív *segédváltozó* bevezetésével ekvivalens módon helyettesíthető az alábbi egyenlőség alakú korlátozó feltétellel:

$$
\begin{array}{rcl}
a_{i1}x_1 + \ldots + a_{in}x_n + x^{(s)}_{n+i} & = & b_i \\
x^{(s)}_{n+i} & \geq & 0
\end{array} .
$$

– *Nagyobb vagy egyenlő alakú korlátozó feltétel egyenlőséggé alakítása.*

Ebben az esetben látszólag kétféleképpen is eljárhatunk. Megtehetjük, hogy a nagyobb vagy egyenlő alakú feltételt először szorozzuk mínusz eggyel, majd az előzőek szerint egy nemnegatív segédváltozó hozzáadásával hozzuk létre a kívánt egyenlőség alakú korlátozó feltételt. Másrészt azonban úgy is eljárhatunk, hogy közvetlenül a nagyobb vagy egyenlő alakú korlátozó feltételhez vezetünk be egy nemnegatív segédváltozót, melyet nyilván ki kell vonni a feltétel bal oldalából. Ha ezek után megszorozzuk a keletkezett egyenlőség alakú korlátozó feltételt, akkor látni fogjuk, hogy ugyanarra az eredményre jutottunk, mint a korábban leírt átalakítással.

– *Szabad változó helyettesítése nemnegatív változókkal.*

Ha egy változó, mondjuk x_j nincs nemnegativitással korlátozva, vagyis szabad változó, akkor előállítható $x_j = x_j^+ - x_j^-$ alakban, ahol $x_j^+ \geq 0$ és $x_j^- \geq 0$. Ez az előállítás nyilvánvalóan nem egyértelmű, azonban a helyettesítéssel keletkező és az eredeti feladat ekvivalenciáját nem befolyásolja az a további megszorítás, hogy x_j^+ és x_j^- egyikét mindig nulla értékűnek gondoljuk az előállításban. Majd azt is látni fogjuk, hogy ez a pótlólagos megkötés a standard alakú lineáris programozási feladatra kidolgozandó megoldó algoritmust sem fogja befolyásolni.

– *Minimalizálandó célfüggvény helyettesítése maximalizálandó célfüggvénnyel.*

Ha egy célfüggvényt minimalizálni kell, az nyilvánvalóan ekvivalens azzal, mint hogyha a célfüggvény mínusz egyszeresét kell maximalizálni. Ezért a minimalizálandó célfüggvény ekvivalens módon helyettesíthető a mínusz egyszeresének a maximalizálásával.□

Megmutatjuk még, hogy egy egyenlőség alakú korlátozó feltétel mindig helyettesíthető két kisebb vagy egyenlő típusú korlátozó feltétellel. Legyen például az i-edik korlátozó feltétel egyenlőség alakú:

$$a_{i1}x_1 + \ldots + a_{in}x_n = b_i.$$

Ekkor ez a feltétel ekvivalens módon helyettesíthető az alábbi kettővel:

$$\begin{array}{c} a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \\ a_{i1}x_1 + \ldots + a_{in}x_n \geq b_i \end{array}.$$

Ha ezután a második egyenlőtlenséget mínusz eggyel megszorozzuk, megkapjuk a kívánt ekvivalens helyettesítést. Ezzel azt bizonyítottuk be, hogy minden lineáris programozási feladat

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & \leq & b_2 \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & \leq & b_m \\ x_1 \geq 0, & & x_2 \geq 0, & & \ldots & & x_n \geq 0 & & \\ (c_1x_1 & + & c_2x_2 & + & \ldots & + & c_nx_{n)} & \rightarrow & \max \end{array}$$

alakúra hozható. A lineáris programozási feladatoknak ezt az alakját *primál alak*nak nevezzük.

2. A szimplex módszer alap algoritmusa

Ha tekintjük az n-dimenziós Euklideszi térben a primál alakúra transzformált lineáris programozási feladatot, akkor láthatjuk, hogy geometriailag a feladat azt jelenti, hogy m darab lineáris féltér közös részének a nemnegatív ortánsba eső részén kell azt a pontot megkeresni, amelyiken egy lineáris függvény (a célfüggvény) a lehető legnagyobb értéket veszi fel. Mivel pedig a nemnegatív ortáns maga is n darab féltér közös része és véges sok féltér közös része az n-dimenziós Euklideszi tér egy konvex poliéderét képezi, a lineáris programozás feladata geometriailag úgy is megfogalmazható, hogy keresendő egy lineáris függvény maximuma az n-dimenziós Euklideszi tér egy konvex poliéderén.

Sajnos a fent leírt geometriai kép alapján csak két- illetve háromváltozós lineáris programozási feladat megoldása képzelhető el, hiszen háromnál magasabb dimenziós Euklideszi térben a geometriai szemléletet elveszítjük.

A geometriai kép segít azonban abban, hogy áttekintsük a lineáris programozási feladat megoldásakor fellépő különböző lehetőségeket. Ezek a következők.

– *Nincs megoldása a feladatnak.*

  A lineáris programozási feladat korlátozó feltételrendszere ábrázolásakor nem alakul ki a nemnegatív ortánsban egy valódi konvex poliéder, azaz az első néhány ábrázolt féltér közös része teljes egészében a következőleg ábrázolandó féltérnek az ellenkező oldalára esik. Ekkor a lineáris korlátozó feltételek és a változókra kirótt nemnegativitási feltételek együttesen ellentmondanak egymásnak.

– *Nincs véges optimuma a feladatnak.*

  A lineáris programozási feladat korlátozó feltételrendszere ábrázolásakor nem korlátos konvex poliéder alakul ki és a célfüggvény növekedési iránya egybeesik a konvex poliéder "nemkorlátossági irányával". Ekkor a feladat célfüggvénye az adott korlátozások mellett tetszőlegesen nagy értékeket felvehet.

– *A feladat véges optimuma a konvex poliéder egy vagy több csúcspontján valósul meg.*

  A lineáris programozási feladatot geometriailag úgy oldhatjuk meg, hogy a megoldások által alkotott konvex poliéderen "áthúzzuk" a célfüggvény egyre nagyobb értékekhez tartozó szintvonalait és amikor elérjük azt a legnagyobb értéket, amelyhez tartozó szintvonalnak utoljára van közös pontja a konvex poliéderrel, ez a közös pont(ok) lesz(nek) a lineáris programozási feladat optimális megoldása. Fontos, és a matematika eszközeivel két, illetve három változósnál többváltozós lineáris programozási feladatokra is igazolható az a következtetés, hogy ha a lineáris programozási feladatnak létezik megoldása és véges optimuma, akkor a véges optimum mindig megvalósul a korlátozó feltételek által létrehozott konvex poliéder legalább egy csúcsán is.

– *Feleslegesen előírt lineáris korlátozó feltételei vannak a feladatnak.*

  A lineáris programozási feladat korlátozó feltételrendszere ábrázolásakor az első néhány ábrázolt féltér közös része teljes egészében része a következőleg ábrázolandó féltérnek. Ekkor az éppen ábrázolandó féltér olyan lineáris korlátozó feltételből származik, amely az összes addig ábrázolt korlátozó feltételnek következménye.

Nyilván célszerű az ilyen felesleges korlátozó feltételeket valami módon kiszűrni a lineáris programozási feladatból.

A szimplex módszert G. B. DANTZIG amerikai matematikus 1947-ben fedezte fel, azonban azt csak 1951-ben publikálta a T. C. KOOPMANS szerkesztésében megjelent cikkgyűjteményben (lásd [2]). Ez a módszer algebrai eszközökkel oldja meg a lineáris programozás feladatát és mint ilyen, nem korlátozódik csupán a két- illetve háromváltozós feladatokra.

Tekintsük a standard alakú lineáris programozási feladatot *vektoriális alak*ban:

$$\begin{gathered}\mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \ldots + \mathbf{a}_nx_n = \mathbf{b}\ (= \mathbf{a}_0)\\ \mathbf{x} \geq \mathbf{0}\\ \max\ \mathbf{c}'\mathbf{x}\end{gathered} \tag{4}$$

ahol

$$\mathbf{a}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}, j = 1, \ldots, n, \quad \mathbf{a}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

m-dimenziós, míg

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

n-dimenziós vektorok, a felső vessző pedig a transzponálás jele, azaz $\mathbf{c}'$ a $\mathbf{c}$ vektort mint sorvektort jelöli és $\mathbf{c}'\mathbf{x}$ a $\mathbf{c}$ és az $\mathbf{x}$ vektorok skaláris szorzata.

Definíció. Azt mondjuk, hogy az n-dimenziós $\mathbf{x}$ vektor a (4) lineáris programozási feladat *megoldásvektora*, ha kielégíti az $\mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \ldots + \mathbf{a}_nx_n = \mathbf{b}$ lineáris egyenletrendszert.

Definíció. Azt mondjuk, hogy az n-dimenziós $\mathbf{x}$ vektor a (4) lineáris programozási feladat *megengedett megoldásvektora*, ha kielégíti az $\mathbf{a}_1x_1 + \mathbf{a}_2x_2 + \ldots + \mathbf{a}_nx_n = \mathbf{b}$ lineáris egyenletrendszert és az $\mathbf{x} \geq \mathbf{0}$ nemnegativitási feltétel is teljesül rá.

Definíció. Az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ együttható oszlopvektorok közül kiválasztott $B = \{\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \ldots, \mathbf{a}_{i_r}\}$ oszlopvektor halmazt a (4) lineáris programozási feladat *bázis vektorrendszer*ének (röviden bázisának) nevezzük, ha a benne szereplő m-dimenziós oszlopvektorok lineárisan függetlenek és az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ együttható oszlopvektorok mindegyike ezek lineáris kombinációjaként előállítható. Célszerű külön jelölést bevezetni a bázis vektorrendszer indexhalmazára, ezért legyen $I_B = \{i_1, i_2, \ldots, i_r\}$. A maradék, bázisrendszeren kívüli oszlopvektorok indexhalmazára pedig vezessük be a $\{K = k_1, k_2, \ldots, k_t\}$ jelölést. Ekkor nyilvánvalóan $I_B \cup K = \{1, 2, \ldots, n\}$ és $r + t = n$.

Definíció. Azt mondjuk, hogy az n-dimenziós $\mathbf{x}$ vektor a (4) lineáris programozási feladat *B bázishoz tartozó bázis megoldásvektora*, ha olyan megoldásvektora annak, amely nembázis komponensei nulla értékűek, azaz $x_k = 0, k \in K$.

Megjegyezzük, hogy minden B bázishoz egyértelműen tartozik egy bázis megoldásvektor, melyet egyszerűen úgy lehet meghatározni, hogy megoldjuk a nulla értéken való rögzítések után fennmaradó

$$\mathbf{a}_{i_1} x_{i_1} + \mathbf{a}_{i_2} x_{i_2} + \ldots + \mathbf{a}_{i_r} x_{i_r} = \mathbf{b} \tag{5}$$

lineáris egyenletrendszert. Mivel a bázisbeli vektorok lineárisan függetlenek, a fenti lineáris egyenletrendszernek csak egy megoldása van. Célszerű a fenti egyenletrendszer megoldásvektorára bevezetni az $\mathbf{x}'_B = (x_{i_1}, x_{i_2}, \ldots , x_{i_r})$ jelölést.

Definíció. Azt mondjuk a (4) lineáris programozási feladat B bázisáról, hogy az *megengedett bázis,* ha a hozzá egyértelműen tartozó bázis megoldásvektor a feladat megengedett megoldásvektora. Ehhez nyilvánvalóan csak annak kell teljesülni, hogy $\mathbf{x}_B \geq 0$, hiszen a bázis megoldásvektor többi komponensére definíció szerint $x_k = 0, k \in K$.

Megjegyezzük, hogy amint azt a következő egyszerű példa is mutatja, a lineáris programozási feladat egy megengedett megoldásvektora egyszerre több bázishoz is tartozhat:

Példa. Tekintsük a következő háromváltozós lineáris programozási feladatot:

$$\begin{array}{ccccccc} x_1 & & & + & x_3 & = & 1 \\ & & x_2 & + & x_3 & = & 1 \\ x_1 \geq 0, & & x_2 \geq 0, & & x_3 \geq 0 & & \\ (x_1 & + & x_2) & & & \rightarrow & \max \end{array} \tag{6}$$

Ennek két különböző bázisa $B_1 = \{\mathbf{a}_1, \mathbf{a}_3\}$ és $B_2 = \{\mathbf{a}_2, \mathbf{a}_3\}$ és mindkettőhöz az $\mathbf{x}' = (0, 0, 1)$ megengedett megoldásvektor tartozik, mint bázis megoldásvektor.

Bizonyítás nélkül közöljük a következő igen fontos tételt, amely rávilágít a lineáris programozási feladat korábban adott geometriai szemléltetése és a szimplex módszer közötti kapcsolatra.

Tétel. A lineáris programozási feladat megengedett megoldásvektorai által alkotott konvex poliéder csúcspontjai és megengedett bázis megoldásvektorai egymásnak kölcsönösen egyértelmű módon megfeleltethetők.

Meg kell jegyezni, hogy a fenti tétel példán történő bemutatása azért nehéz, mert a geometriai szemléltetés csupán két- illetve háromváltozós primál alakú feladatra lehetséges, ezzel szemben a megengedett bázis megoldásvektorokat pedig a standard alakú feladatokkal kapcsolatban vezettük be. Tekintsük mégis a következő példát.

Példa. Legyen a vizsgált lineáris programozási feladat primál alakban az alábbi:

$$\begin{array}{ccccc} x_1 & + & x_2 & \leq & 3 \\ x_1 & & & \leq & 2 \\ & & x_2 & \leq & 2 \\ x_1 \geq 0, & & x_2 \geq 0, & & \\ (x_1 & + & 2x_2) & \rightarrow & \max \end{array}$$

Vezessük be az egyszerűség kedvéért most x_3, x_4 és x_5-tel jelölt segédváltozókat, akkor az ekvivalens standard alakú feladat a következő lesz:

$$\begin{array}{ccccccccccc}
x_1 & + & x_2 & + & x_3 & & & & & = & 3 \\
x_1 & & & & & + & x_4 & & & = & 2 \\
 & & x_2 & & & & & + & x_5 & = & 2 \\
x_1 \geq 0, & & x_2 \geq 0, & & x_3 \geq 0, & & x_4 \geq 0, & & x_5 \geq 0 & & \\
(x_1 & + & 2x_2) & & & & & & & \rightarrow & \max
\end{array}$$

Ennek a standard alakú lineáris programozási feladatnak még nem túl sok számolás árán számba tudjuk venni az összes bázisát. Ehhez tegyük azt, hogy járjuk végig az öt darab együttható oszlopvektorból kiválasztható $\binom{5}{3} = 10$ különböző vektor hármast, ellenőrizzük mindegyikre, hogy bázis-e, ha igen, akkor megengedett bázis-e. A megengedett bázisokhoz tartozó bázis megoldásvektorok mindegyikéhez hozzá kell tudni rendelni a primál alakú feladat megengedett megoldásvektorai által alkotott konvex poliéder egy és csak egy csúcsát.

A következő táblázat foglalja össze a számítások eredményeit:

$B_1 = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$	$x_1 = 2, x_2 = 2, x_3 = -1$	nem megengedett bázis
$B_2 = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4\}$	$x_1 = 1, x_2 = 2, x_4 = 1$	$\mathbf{x}' = (1, 2)$ csúcspont
$B_3 = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_5\}$	$x_1 = 2, x_2 = 1, x_5 = 1$	$\mathbf{x}' = (2, 1)$ csúcspont
$B_4 = \{\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4\}$	$\mathbf{a}_1 = \mathbf{a}_3 + \mathbf{a}_4$	nem bázis
$B_5 = \{\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_5\}$	$x_1 = 2, x_3 = 1, x_5 = 2$	$\mathbf{x}' = (2, 0)$ csúcspont
$B_6 = \{\mathbf{a}_1, \mathbf{a}_4, \mathbf{a}_5\}$	$x_1 = 3, x_4 = -1, x_5 = 2$	nem megengedett bázis
$B_7 = \{\mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$	$x_2 = 2, x_3 = 1, x_4 = 2$	$\mathbf{x}' = (0, 2)$ csúcspont
$B_8 = \{\mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_5\}$	$\mathbf{a}_2 = \mathbf{a}_3 + \mathbf{a}_5$	nem bázis
$B_9 = \{\mathbf{a}_2, \mathbf{a}_4, \mathbf{a}_5\}$	$x_2 = 3, x_4 = 2, x_5 = -1$	nem megengedett bázis
$B_{10} = \{\mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5\}$	$x_3 = 3, x_4 = 2, x_5 = 2$	$\mathbf{x}' = (0, 0)$ csúcspont

Megjegyezzük, hogy a fenti példából nem helyes azt a következtetést levonni, hogy számítsuk ki a megtalált csúcspontokon felvett célfüggvény értékeket, válasszuk ki közülük a legnagyobbat és már meg is oldottuk a lineáris programozási feladatot algebrai módon. Ez igaz ugyanis a fenti kisméretű feladat esetén, azonban reális méretű feladatok esetén a megoldásnak ez az útja nem járható.

Tekintsük ismét az általános (4) alakú lineáris programozási feladatot. Tegyük fel egyelőre, hogy adott ehhez a feladathoz egy $B = \{\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \dots, \mathbf{a}_{i_r}\}$ induló megengedett bázis. Ekkor az (4) lineáris programozási feladat oszlopvektorait a B bázisbeli vektorok lineáris kombinációjaként előállító együtthatókat az alábbi lineáris egyenletrendszerek megoldásával lehet előállítani:

$$\mathbf{a}_p = \sum_{i \in I_B} d_{ip} \mathbf{a}_i, p = 0, 1, 2, \dots, n, \tag{7}$$

ahol $\mathbf{a}_0$ ismét a $\mathbf{b}$ jobboldali vektort jelöli. Nyilván ez is előállítható a B bázis oszlopvektoraival, hiszen feltettük, hogy a B bázis megengedett, azaz létezik az (4) lineáris programozási feladatnak megoldásvektora. Az (7) lineáris egyenletrendszerek megoldásával kapott $d_{ip}, i \in I_B, p = 0, 1, 2, \dots, n$ számokkal definiáljunk további $d_{0p}, p = 0, 1, 2, \dots, n$ számokat az alábbi módon

$$d_{0p} = z_p - c_p = \sum_{i \in I_B} d_{ip} c_i - c_p, p = 0, 1, 2, \ldots, n. \tag{8}$$

A fenti képletekben feltettük, hogy $c_0 = 0$, ez nem befolyásolja a megoldani kívánt (4) lineáris programozási feladatot. Az (7) és (8) összefüggésekkel bevezetett $d_{ip}, i \in I_B, i = 0, p = 0, 1, 2, \ldots, n$ számokat foglaljuk táblázatba és ezt a táblázatot nevezzük az (4) lineáris programozási feladat B megengedett bázishoz tartozó szimplex táblázatának:

$$\begin{array}{c|c|cccc} B & \mathbf{a}_0 & \mathbf{a}_1 & \ldots & \mathbf{a}_n \\ \hline \mathbf{a}_{i_1} & d_{i_1 0} & d_{i_1 1} & \ldots & d_{i_1 n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{i_r} & d_{i_r 0} & d_{i_r 1} & \ldots & d_{i_r n} \\ \hline z-c & d_{00} & d_{01} & \ldots & d_{0n} \end{array} \tag{9}$$

A szimplex táblázatba foglalt számok szemléletes jelentése az, hogy a felső peremre írt $\mathbf{a}_p$ oszlopvektort a baloldali peremre írt, B bázist alkotó vektorok lineáris kombinációjaként a d_{ip} együtthatókkal lehet előállítani, minden p index esetén, $\{p = 0, 1, 2, \ldots, n\}$. Az utolsó, a baloldali peremen $z-c$-vel azonosított sor szemléletes származtatásához pedig célszerű még a szimplex tábla peremén külön feltüntetni a megfelelő célfüggvény együtthatókat:

$$\begin{array}{cc|c|cccc} & & 0 & c_1 & \ldots & c_n \\ & B & \mathbf{a}_0 & \mathbf{a}_1 & \ldots & \mathbf{a}_n \\ \cline{2-6} c_{i_1} & \mathbf{a}_{i_1} & d_{i_1 0} & d_{i_1 1} & \ldots & d_{i_1 n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{i_r} & \mathbf{a}_{i_r} & d_{i_r 0} & d_{i_r 1} & \ldots & d_{i_r n} \\ \cline{2-6} & z-c & d_{00} & d_{01} & \ldots & d_{0n} \end{array}$$

Ekkor a $z-c$-vel azonosított sor d_{0p} elemét úgy számíthatjuk, hogy az $\mathbf{a}_p$ alá írt d_{ip} együtthatókkal nem a B bázist alkotó vektorokat "kombináljuk lineárisan", hanem a peremre írt megfelelő célfüggvény együtthatókat és végül még a kapott összegből levonjuk a felső peremre írt c_p célfüggvény együtthatót. A szimplex tábla a következő fontos tulajdonságokkal rendelkezik:

1. Ha $p \in I_B$, akkor az $\mathbf{a}_p$ vektor előállítása a B bázist alkotó vektorok lineáris kombinációjaként triviális, azaz minden d_{ip} együttható nulla értékű, kivéve a $d_{pp} = 1$ értéket.

2. Ha $p \in I_B$, akkor a $z-c$-vel azonosított sorban d_{0p} is nulla értékű, hiszen az (8) számítás eredménye az előző tulajdonság figyelembevételével nullát eredményez.

3. A táblázatban $d_{i_1 0} \geq 0, \ldots, d_{i_r 0} \geq 0$, mivel ezek a számok a B megengedett bázishoz tartozó megengedett bázismegoldás báziskomponensei, hiszen az (5) és (8) definiáló lineáris egyenletrendszerek azonosak.

4. A táblázatban a d_{00} szám a B megengedett bázishoz tartozó megengedett bázismegoldáson a célfüggvény értéke, hiszen az (8) számítás eredménye $p = 0$-ra az előző tulajdonság értelmében pontosan ezt adja.

Az (9) szimplex táblának három típusát különböztetjük meg:

1. *Típus:* $d_{0p} \geq 0, p = 1, 2, \ldots n$.
2. *Típus:* Létezik $k \notin I_B$, hogy $d_{0k} < 0$, és egy ilyen k indexre $d_{ik} \leq 0$ minden $i \in I_B$ esetén.
3. *Típus:* Létezik $k \notin I_B$, hogy $d_{0k} < 0$, és minden ilyen k indexre létezik olyan $i \in I_B$, hogy $d_{ik} > 0$.

A szimplex tábla fenti három típusa egymást nyilván kizárja, és az összes lehetőséget magában foglalja. Az 1. típusú szimplex tábla esetén a következő tételt lehet bebizonyítani:

Tétel. Ha az (9) szimplex tábla 1. típusú, akkor a B megengedett bázishoz tartozó bázismegoldás a lineáris programozási feladat optimális megoldása.

Bizonyítás. Legyen $\mathbf{y}$ az (4) lineáris programozási feladat tetszőleges megengedett megoldásvektora, azaz legyen $\sum_{p=1}^{n} \mathbf{a}_p y_p = \mathbf{b}, \mathbf{y} \geq \mathbf{0}$. Ekkor $d_{0p} = z_p - c_p \geq 0, p = 1, 2, \ldots, n$ és $\mathbf{y} \geq \mathbf{0}$ miatt

$$\mathbf{c}'\mathbf{y} = \sum_{p=1}^{n} c_p y_p \leq \sum_{p=1}^{n} z_p y_p = \sum_{p=1}^{n} \left(\sum_{i \in I_B} d_{ip} c_i \right) y_p = \\ \sum_{i \in I_B} \left(\sum_{p=1}^{n} d_{ip} y_p \right) c_i = \sum_{i \in I_B} d_{i0} c_i = d_{00},$$

hiszen

$$\mathbf{b} = \sum_{p=1}^{n} \mathbf{a}_p y_p = \sum_{p=1}^{n} \left(\sum_{i \in I_B} d_{ip} \mathbf{a}_i \right) y_p = \sum_{i \in I_B} \left(\sum_{p=1}^{n} d_{ip} y_p \right) \mathbf{a}_i$$

és

$$\mathbf{b} = \sum_{i \in I_B} d_{i0} \mathbf{a}_i,$$

valamint a bázist alkotó vektorok lineáris függetlensége miatt minden $i \in I_B$ esetén

$$d_{i0} = \sum_{p=1}^{n} d_{ip} y_p,$$

d_{00} pedig a szimplex tábla 4. tulajdonsága szerint egyenlő a B megengedett bázishoz tartozó megengedett bázismegoldáson a célfüggvény értékével. □

A 2. típusú szimplex tábla esetén a következő tétel igazolható:

Tétel. Ha az (9) szimplex tábla 2. típusú, akkor az (4) lineáris programozási feladatnak nincs véges optimuma.

Bizonyítás. Legyen $k \notin I_B$ egy, a 2. típusú szimplex táblának megfelelő index, azaz $d_{0k} < 0$, és $d_{ik} \leq 0$ minden $i \in I_B$ esetén. Legyen továbbá $\beta > 0$ tetszőleges paraméter. Ekkor az (4) lineáris programozási feladat egyenletrendszerének a következő átalakításából a feladat megengedett megoldásainak egy a $\beta > 0$ paramétertől függő serege olvasható le:

$$\begin{aligned}\mathbf{b} &= \sum_{i\in I_B} d_{i0}\mathbf{a}_i - \beta\mathbf{a}_k + \beta\mathbf{a}_k = \sum_{i\in I_B} d_{i0}\mathbf{a}_i - \beta\sum_{i\in I_B} d_{ik}\mathbf{a}_i + \beta\mathbf{a}_k = \\ &= \sum_{i\in I_B}\left(d_{i0} - \beta d_{ik}\right)\mathbf{a}_i + \beta\mathbf{a}_k\end{aligned}$$

Jelölje $\widetilde{\mathbf{x}}^\beta$ a $\mathbf{b}$ vektor fenti előállításából leolvasható megoldásvektor sereget. Ennek komponenseire:

$$\begin{aligned}\widetilde{x}_i^\beta &= d_{i0} - \beta d_{ik}, i \in I_B, \\ \widetilde{x}_k^\beta &= \beta, \\ \widetilde{x}_p^\beta &= 0, p \notin I_B, p \neq k.\end{aligned}$$

Megjegyezzük, hogy $\widetilde{\mathbf{x}}^\beta$ nyilvánvalóan nem bázis megoldásvektor, hiszen $k \notin I_B$ és $\widetilde{x}_k^\beta = \beta > 0$. Megmutatjuk, hogy $\widetilde{\mathbf{x}}^\beta \geq \mathbf{0}$, azaz megengedett megoldásvektor és a hozzátartozó célfüggvény értékek végtelenhez tartanak, ha β végtelenhez tart.

Mivel $\widetilde{x}_i^\beta = d_{i0} - \beta d_{ik} \geq 0, i \in I_B$, hiszen $d_{i0} \geq 0, \beta > 0, d_{ik} \geq 0, i \in I_B$ és $\widetilde{x}_k^\beta = \beta > 0$, azért valóban $\widetilde{\mathbf{x}}^\beta \geq \mathbf{0}$, a megfelelő célfüggvényértékekre pedig:

$$\begin{aligned}\mathbf{c}'\widetilde{\mathbf{x}}^\beta &= \sum_{i\in I_B}\left(d_{i0} - \beta d_{ik}\right)c_i + \beta c_k = \sum_{i\in I_B} d_{i0}c_i - \beta\sum_{i\in I_B} d_{ik}c_i + \beta c_k = \\ &= z_0 - \beta z_k + \beta c_k = d_{00} + \beta\left(z_k - c_k\right) = d_{00} + \beta d_{0k}.\end{aligned}$$

Ebből azonnal látható, hogy $\mathbf{c}'\widetilde{\mathbf{x}}^\beta \to \infty$, ha $\beta \to \infty$, hiszen $d_{0k} > 0$. □

Tétel. Ha az (9) szimplex tábla 3. típusú, akkor legyen $k \notin I_B$ **egy tetszőleges**, a 3. típusú szimplex táblának megfelelő index, azaz $d_{0k} < 0$, és legyen $j \in I_B$ **egy tetszőleges** olyan index, amelyre a

$$\min_{\substack{i \in I_B \\ d_{ik} > 0}} \frac{d_{i0}}{d_{ik}} = \frac{d_{j0}}{d_{jk}} \tag{10}$$

minimum megvalósul. Ekkor a $B_1 = B - \{\mathbf{a}_j\} + \{\mathbf{a}_k\}$ bázishoz is megengedett bázismegoldás tartozik és ezen a bázismegoldáson az (4) lineáris programozási feladat célfüggvény értéke **nem kisebb**, mint a B bázishoz tartozó megengedett bázismegoldáson volt.

Bizonyítás. Jelölje $I_{B_1} = I_B - \{j\} + \{k\}$ a B_1 bázis indexhalmazát. Állítsuk elő a B_1 bázishoz tartozó bázismegoldást. Ehhez vezessük be most is a $\beta > 0$ paramétert. Ezzel

$$\begin{aligned}
\mathbf{b} &= \sum_{i\in I_B} d_{i0}\mathbf{a}_i - \beta\mathbf{a}_k + \beta\mathbf{a}_k = \sum_{i\in I_B} d_{i0}\mathbf{a}_i - \beta \sum_{i\in I_B} d_{ik}\mathbf{a}_i + \beta\mathbf{a}_k = \\
&= \sum_{i\in I_B} \left(d_{i0} - \beta d_{ik}\right)\mathbf{a}_i + \beta\mathbf{a}_k,
\end{aligned}$$

ahol most a $\beta > 0$ paraméter értékét úgy célszerű megválasztani, hogy $j \in I_B$-re az $\mathbf{a}_j$ vektor együtthatója nulla legyen:

$$\begin{gathered}
d_{j0} - \beta d_{jk} = 0, \\
\beta = \tfrac{d_{j0}}{d_{jk}}.
\end{gathered}$$

Ekkor tehát azt írhatjuk, hogy

$$\mathbf{b} = \sum_{\substack{i\in I_B\\ i\neq j}} \left(d_{i0} - \frac{d_{j0}}{d_{jk}} d_{ik}\right)\mathbf{a}_i + \frac{d_{j0}}{d_{jk}}\mathbf{a}_k,$$

és a $\mathbf{b}$ vektornak ebből az előállításából pontosan a B_1 bázishoz tartozó $\mathbf{x}^{(1)}$ bázis megoldásvektor komponensei olvashatók le:

$$\begin{aligned}
x_i^{(1)} &= d_{i0} - \tfrac{d_{j0}}{d_{jk}} d_{ik}, i \in I_{B_1}, i \neq k, \\
x_k^{(1)} &= \tfrac{d_{j0}}{d_{jk}}, \\
x_p^{(1)} &= 0, i \notin I_{B_1}.
\end{aligned}$$

Meg kell mutatni, hogy ez megengedett bázis megoldásvektor, vagyis $\mathbf{x}^{(1)} \geq \mathbf{0}$ és rajta a célfüggvény értéke nem kisebb, mint a B bázishoz tartozó megengedett bázismegoldáson volt, vagyis $\mathbf{c}'\mathbf{x}^{(1)} \geq d_{00}$.

$\mathbf{x}^{(1)} \geq 0$, mert

$x_k^{(1)} = \frac{d_{j0}}{d_{jk}} \geq 0$, hiszen $d_{j0} \geq 0, d_{jk} > 0$,

$x_i^{(1)} \geq 0, i \in I_{B_1}, i \neq k$ pedig a következőképpen látható be:

ha $d_{ik} \leq 0$, akkor $x_i^{(1)} = d_{i0} - \frac{d_{j0}}{d_{jk}} d_{ik} \geq 0$, hiszen $d_{i0} \geq 0, d_{j0} \geq 0, d_{jk} > 0$,

ha $d_{ik} > 0$, akkor $x_i^{(1)} = d_{i0} - \frac{d_{j0}}{d_{jk}} d_{ik} \geq 0$ a $d_{ik} > 0$ értékkel történő leosztás és rendezés után azt jelenti, hogy $\frac{d_{i0}}{d_{ik}} \geq \frac{d_{j0}}{d_{jk}}$, ez viszont következik a $j \in I_B$index (10) kiválasztási szabályából.

A megfelelő célfüggvény értékekre pedig:

$$\begin{aligned}
\mathbf{c}'\mathbf{x}^{(1)} &= \sum_{\substack{i\in I_{B_1}\\ i\neq k}} \left(d_{i0} - \tfrac{d_{j0}}{d_{jk}} d_{ik}\right) c_i + \tfrac{d_{j0}}{d_{jk}} c_k = \sum_{\substack{i\in I_B\\ i\neq j}} \left(d_{i0} - \tfrac{d_{j0}}{d_{jk}} d_{ik}\right) c_i + \tfrac{d_{j0}}{d_{jk}} c_k = \\
&= \sum_{i\in I_B} \left(d_{i0} - \tfrac{d_{j0}}{d_{jk}} d_{ik}\right) c_i + \tfrac{d_{j0}}{d_{jk}} c_k = \sum_{i\in I_B} d_{i0}c_i - \tfrac{d_{j0}}{d_{jk}} \sum_{i\in I_B} d_{ik}c_i + \tfrac{d_{j0}}{d_{jk}} c_k = \\
&= z_0 - \tfrac{d_{j0}}{d_{jk}} z_k + \tfrac{d_{j0}}{d_{jk}} c_k = d_{00} + \tfrac{d_{j0}}{d_{jk}} \left(z_k - c_k\right) = d_{00} + \tfrac{d_{j0}}{d_{jk}} d_{0k}.
\end{aligned}$$

Mivel $d_{j0} \geq 0, d_{jk} > 0$ és $d_{0k} < 0$, $\mathbf{c}'\mathbf{x}^{(1)} \geq d_{00}$ következik. □

Megjegyzés. Megjegyezzük, hogy a bizonyításból az is leolvasható, hogy a célfüggvény értéke csak akkor maradhat változatlan, ha $d_{j0} = 0$. Az olyan bázisokat **degeneráltaknak** nevezzük, amelyekre a bázis megoldásvektor bázis indexhalmazhoz tartozó komponensei között is van nulla értékű. Tehát ha a B bázis nem degenerált, akkor a B_1 bázisra történő áttérés során a célfüggvény értéke határozottan nő.

Definíció. Szimplex módszernek nevezzük a standard alakra transzformált lineáris programozási feladat megoldására szolgáló azon algoritmust, amely a feladat egy megengedett bázisából kiindulva meghatározza a hozzá tartozó szimplex táblát. Ha a szimplex tábla 1. típusú, akkor leolvassa belőle a megengedett bázishoz tartozó bázis megoldásvektort, mint a lineáris programozási feladat optimális megoldásvektorát. Ha a szimplex tábla 2. típusú, akkor megállapítja, hogy a lineáris programozási feladatnak nincs véges optimuma. Ha pedig a szimplex tábla 3. típusú, akkor áttér az 1.3 Tételben bevezetett új B_1 bázisra, és megismétli a korábban mondottakat a B_1 bázissal. Mindezt addig teszi, míg 1, vagy 2. típusú szimplex táblához nem ér.

Az algoritmus további szemléletesebbé tétele céljából tekintsük ismét az (4) formában felírt, standard alakra hozott lineáris programozási feladatot, valamint a hozzá az (7) és (8) képletekkel bevezetett segéd mennyiségeket. Ekkor a bázisvektorok lineáris függetlenségének és a $\mathbf{b}$ jobboldali vektor alábbi két előállításának a felhasználásával

$$\begin{aligned}\mathbf{b} &= \sum_{p=1}^{n} \mathbf{a}_p x_p = \sum_{p=1}^{n} \left(\sum_{i \in I_B} d_{ip} \mathbf{a}_i \right) x_p = \sum_{i \in I_B} \left(\sum_{p=1}^{n} d_{ip} x_p \right) \mathbf{a}_i, \\ \mathbf{b} &= \sum_{i \in I_B} d_{i0} \mathbf{a}_i,\end{aligned}$$

a feladat lineáris egyenletrendszere a következő ekvivalens alakra hozható:

$$d_{i0} = \sum_{p=1}^{n} d_{ip} x_p, \forall i \in I_B.$$

Ha ebben figyelembe vesszük a $d_{ip}, i \in I_B, p = 1, 2, \ldots, n$ számok tulajdonságait, és kissé átrendezve részletesebben kiírjuk az egyenleteket:

$$\begin{array}{lllllllllllll} d_{i_1 0} & - & (& x_{i_1} & & & d_{i_1 k_1} x_{k_1} & + & \ldots & + & d_{i_1 k_t} x_{k_t} &) & = & 0 \\ d_{i_2 0} & - & (& & x_{i_2} & & d_{i_2 k_1} x_{k_1} & + & \ldots & + & d_{i_2 k_t} x_{k_t} &) & = & 0 \\ \vdots & & & & \ddots & & \vdots & & \ddots & & \vdots & & & \vdots \\ d_{i_r 0} & - & (& & & x_{i_r} & d_{i_r k_1} x_{k_1} & + & \ldots & + & d_{i_r k_t} x_{k_t} &) & = & 0 \end{array}$$

akkor ezt a lineáris programozási feladat egyenletrendszerének az $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ változókra nézve kifejezett alakjának nevezzük. Jelöljük z-vel a célfüggvényt és hozzuk azt is ennek megfelelő alakra:

$$-0 \quad -(c_{i_1} x_{i_1} - c_{i_2} x_{i_2} - \ldots - c_{i_r} x - c_{k_1} x_{k_1} \quad - \quad \ldots \quad - \quad c_{k_t} x_{k_t} \quad) \quad = \quad z \ .$$

z-nek ebben az alakjában az $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ kifejezett valtozók könnyen eliminálhatók. Ha ezt az eliminálást úgy hajtjuk végre, hogy a fenti kifejezett alakú egyenletrendszer nullát jelentő egyenleteit rendre $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$-rel szorozva a z-t definiáló egyenlethez adjuk, akkor könnyen látható, hogy a célfüggvény ekvivalens marad és benne éppen az (8) összefüggéssel definiált $d_{0p}, p = 1, 2, \ldots, n$ együtthatók jelennek meg. Igy ha a változók nemnegativitási feltételeit is kiírjuk, a teljes (4) lineáris programozási feladattal ekvivalens feladat:

$$
\begin{array}{llllll}
d_{i_1 0} - (& x_{i_1} & & & d_{i_1 k_1} x_{k_1} + \ldots + d_{i_1 k_t} x_{k_t}) = 0 \\
d_{i_2 0} - (& & x_{i_2} & & d_{i_2 k_1} x_{k_1} + \ldots + d_{i_2 k_t} x_{k_t}) = 0 \\
\vdots & & \ddots & & \vdots \qquad\qquad \vdots \qquad \vdots \\
d_{i_r 0} - (& & & x_{i_r} & d_{i_r k_1} x_{k_1} + \ldots + d_{i_r k_t} x_{k_t}) = 0 \\
 & x_{i_1} \geq 0, & x_{i_2} \geq 0, \quad \ldots & x_{i_r} \geq 0, & x_{k_1} \geq 0, \quad \ldots \quad x_{k_t} \geq 0 \\
d_{00} - (& & & & d_{0k_1} x_{k_1} + \ldots + d_{0k_t} x_{k_t}) = z \rightarrow \max
\end{array}
\tag{11}
$$

Ezt az alakot a teljes lineáris programozási feladat $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ változókra nézve kifejezett alakjának nevezzük. Ebből azonnal látható, hogy a korábban bevezetett szimplex tábla nem más, mint az ebből az egyenletrendszerből kigyűjtött $d_{ip}, i \in I_B, i = 0; p = 0, p \in I_B, p \in K$ együtthatók táblázata. Ez az észrevétel több szempontból is előnyös.

Először megmutatjuk, hogy a szimplex táblával kapcsolatban korábban bebizonyított három tétel milyen szemléletes jelentéssel bír.

Az (11) kifejezett alakú lineáris programozási feladatról azonnal leolvasható az a megoldás, melyben $x_k = 0, k \in K$, hiszen $x_i = d_{i0}, i \in I_B$ ekkor közvetlenül adódik. Ez pedig a B megengedett bázishoz tartozó bázismegoldás. Ezen a megoldáson a célfüggvény értéke nyilván $z = d_{00}$. Ha ennél nagyobb célfüggvényű megoldást keresünk, az úgy nyerhető, hogy az $x_k = 0, k \in K$ változók valamelyikének az értékét növeljük. Ha figyelembe vesszük a negatív zárójelezést, látható, hogy annak az x_k változónak az értékét kell növelni, amely mellett negatív d_{0k} együttható áll a célfüggvényben.

Ha negatív d_{0k} együttható nem létezik, azaz $d_{0p} \geq 0, p \in K$ (1. típusú szimplex tábla), akkor az (11) kifejezett alakról azonnal leolvasható, hogy $z \leq d_{00}$, hiszen elegendő csak az $x_p \geq 0, p \in K$ korlátozásokat tekinteni.

Ha létezik negatív d_{0k} együttható, akkor az x_k változó értékének a növelésével nő a célfüggvény z értéke és a kifejezett alaknak köszönhetően az ezáltal okozott változás minden lineáris egyenletben korrigálható az egyenlethez tartozó kifejezett változó értékének a változtatásával. Ennek az az előnye, hogy egy kifejezett változó értékének a változtatása egyetlen további egyenletben, még a célfüggvényt definiáló egyenletben sem okoz változást. Igy egyrészt az egyenletekben a változások jól kezelhetők, másrészt a célfüggvényben elért növekedés biztosan nem romlik el, amikor az egyenlőségek megőrzéséhez szükséges korrekciókat a kifejezett változókkal végrehajtjuk!

Gondoljuk meg, hogy az elmondottak szerint mit jelent egy $i \in I_B$ indexhez tartozó lineáris egyenletben az x_k változó értékének a növelése által okozott változás korrigálása az egyenlethez tartozó x_i kifejezett változó értékének a változtatásával? Ha az x_k változó d_{ik} együtthatója nulla, akkor nincs mit korrigálni; ha negatív, akkor a zárójelen belüli $d_{ik}x_k$ értékű változás csökkenés, így mindig korrigálható az x_i kifejezett változó értékének a növelésével; ha pedig pozitív, akkor a zárójelen belüli $d_{ik}x_k$ értékű változás növekedés, így korrigálás az x_i kifejezett változó értékének a csökkentésével hajtható végre, ennek pedig határt szab az, hogy x_i az induló d_{i0} értékéről csak nulláig csökkenthető, vagyis a korrigálás csak addig lehetséges, ameddig $d_{ik}x_k \leq d_{i0}$, azaz x_k értéke nem növelhető tovább, mint d_{i0}/d_{ik}.

Ha tehát $d_{ik} \leq 0, i \in I_B$ (2. típusú szimplex tábla), akkor x_k értéke a végtelenségig növelhető, vagyis látható, hogy a lineáris programozási feladatnak nincs véges optimuma.

Ha pedig létezik $d_{ik} > 0, i \in I_B$ (3. típusú szimplex tábla), akkor meg kell keresni az ilyen $i \in I_B$ értékekre a d_{i0}/d_{ik} hányadosok minimumát (lásd (10)), és x_k értékét erre a szintre szabad csak felemelni.

Ekkor, ha j egy olyan $i \in I_B$ indexet jelöl, amelyre a minimum megvalósult, x_j értékét nulla szintre kell csökkenteni és így az (11) kifejezett alakot át lehet alakítani olyanná, hogy az x_k változó legyen az x_j helyett kifejezett. Ez az átalakítás egy egyszerű eliminálással elvégezhető és az átalakítás után nyilvánvalóan olyan kifejezett alakot nyerünk, amelyből a $d_{ip}, i \in I_B, i = 0; p = 0, p \in I_B, p \in K$ együtthatókat kiszedve éppen a $B_1 = B - \{\mathbf{a}_j\} + \{\mathbf{a}_k\}$ megengedett bázishoz tartozó szimplex táblát nyerjük.

Ez az észrevétel segít abban, hogy felírjuk a szimplex tábla transzformációs formuláit, amikor a B megengedett bázisról a B_1 megengedett bázisra térünk át. Vezessük be a B bázishoz tartozó szimplex tábla soraira mint sorvektorokra a $\delta_i = (d_{i0}, d_{i1}, \dots, d_{in})$ és a B_1 bázishoz tartozó szimplex tábla soraira mint sorvektorokra a $\delta_i^{(1)} = \left(d_{i0}^{(1)}, d_{i1}^{(1)}, \dots, d_{in}^{(1)}\right)$ jelölést, akkor a szimplex táblák mögé képzeve a megfelelő kifejezett alakokat és azok Gauss-Jordan eliminációval történő transzformálási szabályait, azonnal kapjuk, hogy

$$\begin{aligned} \delta_k^{(1)} &= \tfrac{1}{d_{jk}}\delta_j, \\ \delta_i^{(1)} &= \delta_i - d_{ik}\delta_k^{(1)} = \delta_i - \tfrac{d_{ik}}{d_{jk}}\delta_j, i \in I_B, i \neq k, i = 0. \end{aligned} \tag{12}$$

Ugyanez komponensenként felírva:

$$\left.\begin{aligned} d_{kp}^{(1)} &= \tfrac{1}{d_{jk}}d_{jp} \\ d_{ip}^{(1)} &= d_{ip} - d_{ik}d_{kp}^{(1)} = d_{ip} - \tfrac{d_{ik}}{d_{jk}}d_{jp}, i \in I_B, i \neq k, i = 0 \end{aligned}\right\} p = 0, 1, 2, \dots, n. \tag{13}$$

A szimplex módszer alap algoritmusát ez a két transzformációs formula teszi teljessé.

3. A lexikografikus szimplex módszer

Az 2. szakaszban adott példa azt mutatta, hogy a szimplex módszer alap algoritmusának alkalmazásakor előfordulhat az a kellemetlen helyzet, hogy a bázisba bejővő, illetve a bázisból kimenő vektor kiválasztásának az algoritmusban meghagyott szabadságát olymódon tesszük egyértelművé, hogy degenerált bázisok során át, változatlan célfüggvény értékkel úgy halad az algoritmus, hogy visszajut egy olyan megengedett bázishoz, amelyiknél korábban már járt. Ekkor, ha nem változtatunk a választások egyértelműsítésén, a végtelenségig ebben a ciklusban fog az algoritmusunk keringeni. Ezt a helyzetet nevezzük a szimplex módszer alap algoritmusa **ciklizálás**ának.

Meg kell jegyezni, hogy a szimplex módszer alap algoritmusának a számítógépes megvalósításai esetén a ciklizálás gyakorlatilag sohasem tart a végtelenségig (a program futása felfüggesztéséig), hanem a program előbb, vagy utóbb automatikusan kiugrik belőle. Ennek az a magyarázata, hogy az azonos számítási ciklus nagyszámú ismétlése során a halmozódó számítási pontatlanságok miatt egyszer csak annyira torzulnak a gép által számított értékek, hogy az algoritmus a rögzített kiválasztási szabályok ellenére is más útra téved. A ciklizálást bizonyíthatóan elkerő algoritmus variáns kidolgozásának tehát sokkal inkább elméleti, mint gyakorlati a jelentősége. Hasonlóan fontos azonban a szimplex módszer alap algoritmusának olyan módosítása is, amely kellő védelmet tud nyújtani a számítási pontatlanságok halmozódása ellen. Az alap algoritmusnak ezt a változatát módosított szimplex módszernek nevezzük és egy későbbi szakaszban fogjuk tárgyalni.

A ciklizálás bizonyítható elkerülésére több módszer is létezik. Az elkövetkezőkben ismertetendő módszer neve **lexikografikus szimplex módszer.** A szimplex módszernek ez a változata először A. Charnes 1952-ben publikált cikkében (lásd [1]) jelent meg, ő azonban perturbációs módszernek nevezte azt. A módszerre 1955-ben új bizonyítást közölt G. B. Dantzig, A. Orden és P. Wolfe (lásd [6]), azonban az alábbiakban ismertetendő egyszerű tárgyalást Prékopa András írta le először [12] jegyzetében.

A módszer tárgyalását néhány definícióval kezdjük.

Definíció. Azt mondjuk, hogy egy n-dimenziós $\mathbf{x}$ vektor **lexikografikusan pozitív** (jelölése $\mathbf{x} \succ \mathbf{0}$), ha az $\mathbf{x}$ vektor első nullától különböző komponense pozitív.

Definíció. Azt mondjuk, hogy egy n-dimenziós $\mathbf{y}$ vektor **lexikografikusan nagyobb**, mint egy n-dimenziós $\mathbf{x}$ vektor, ha $\mathbf{y} - \mathbf{x} \succ \mathbf{0}$, vagy szavakkal, ha az $\mathbf{x}$ és $\mathbf{y}$ vektorok komponensenkénti összehasonlítása során az első nem azonosan egyenlő komponens párban az $\mathbf{y}$ vektor komponense nagyobb, mint az $\mathbf{x}$ vektor komponense.

Definíció. Azt mondjuk, hogy egy B bázishoz tatozó szimplex tábla lexikografikusan pozitív, ha a tábla $\delta_i, i \in I_B$ sorai, mint $n+1$-dimenziós sorvektorok lexikografikusan pozitívak.

Definíció. Lexikografikus kiválasztási szabály. A bázist elhagyó $\mathbf{a}_j$ vektor kiválasztásának az a módja, amikor a $j \in I_B$ indexet úgy választjuk ki, hogy

$$\operatorname*{lex\ min}_{\substack{i \in I_B \\ d_{ik} > 0}} \frac{1}{d_{ik}}\delta_i = \frac{1}{d_{jk}}\delta_j \tag{14}$$

teljesüljön.

Vegyük észre, hogy a lexikografikus kiválasztási szabály az (10) közönséges kiválasztási szabállyal azonos eredményre vezet, ha ez utóbbi egyértelműen választja ki a bázist elhagyó vektor j indexét. Vektorok lexikografikus minimum keresése ugyanis éppen úgy realizálható, hogy először az első komponensek minimumát keressük meg, ha az nem egyértelmű, akkor az addig minimumként szóbajöhető vektorok második komponenseire keressük a minimumot és így tovább. Az is látható, hogy ezzel az eljárással a bázist elhagyó vektor j indexének a kiválasztása mindig egyértelmű lesz. Ehhez elég az, hogy a szimplex tábla egyik sora sem számszorosa a másiknk, mivel mindig van a táblának egy teljes egységmátrix része.

Definíció. Lexikografikus szimplex módszer. A szimplex módszer alap algoritmusának olyan végrehajtása, amikor lexikografikosan pozitív szimplex táblából indulva, a bázist elhagyó vektort mindig a lexikografikus kiválasztási szabállyal határozzuk meg.

A lexikografikus szimplex módszer fenti definíciója csak akkor bír valódi tartalommal, ha megmutatjuk, hogy tetszőleges megengedett bázishoz tartozó szimplex tábla felírható lexikografikusan pozitív módon. Ez azonban így van, hiszen megengedett bázishoz tarozó szimplex tábla esetén a $\delta_i, i \in I_B$ vektorok első komponensei nemnegatívak, ezt követően pedig a lineáris programozási feladat változóinak alkalmas sorszámozásával mindig elérhető, hogy a megengedett bázisbeli oszlopvektorok legyenek elöl. Igy a szimplex tábla nemnegatív első oszlopát egy teljes egységmátrix követi, ez pedig elegendő a $\delta_i, i \in I_B$ vektorok lexikografikus pozitivitásához.

A lexikografikus szimplex módszerrel kapcsolatban a következő két állítást bizonyítjuk be.

Tétel. A lexikografikus szimplex módszer alkalmazása során minden szimplex tábla lexikografikusan pozitív.

Bizonyítás. Bár már láttuk azt, hogy bármely megengedett bázishoz tartozó szimplex tábla mindig felirható lexikografikusan pozitív módon, a tétel állítása mégis bizonyításra szorul, mivel a lexikografikus szimplex módszer alkalmazásakor nincs megengedve a változók sorszámozásának a megváltoztatása, azaz a szimplex tábla felső peremére írt oszlopvektorok sorrendje induláskor rögzítendő. Elegendő csak annyit megmutatni, hogy a lexikografikus szimplex módszer alkalmazásakor egyik szimplex tábláról a másikra áttérve a tábla lexikografikus $\delta_i, i \in I_B$ sorvektorai lexikografikusan pozitívak maradnak. Tekintsük ehhez a B bázisról a $B_1 = B - \{\mathbf{a}_j\} + \{\mathbf{a}_k\}$ bázisra történő áttérés (12) transzformációs formuláit. Ebben

$$\delta_k^{(1)} = \frac{1}{d_{jk}}\delta_j \succ \mathbf{0}, \text{ mivel } d_{jk} > 0 \text{ és } \delta_j \succ \mathbf{0},$$

a $\delta_i^{(1)}, i \in I_B, i \neq k$ sorvektorok lexikografikus pozitivitásának a bizonyításához pedig azt kell megmutatni, hogy

$$\delta_i - \frac{d_{ik}}{d_{jk}}\delta_j \succ \mathbf{0}, i \in I_B, i \neq k.$$

Ha $d_{ik} \leq 0$, akkor ez triviálisan igaz, hiszen $\delta_i \succ \mathbf{0}$,$d_{jk} > 0$ és $\delta_j \succ \mathbf{0}$.

Ha $d_{ik} > 0$, akkor pedig d_{ik}-val osztva és átrendezve az egyenlőtlenséget:

$$\frac{1}{d_{ik}}\delta_i \succ \frac{1}{d_{jk}}\delta_j$$

kell, hogy legyen. Ez viszont az (14) lexikografikus kiválasztási szabály alkalmazása miatt mindig teljesül. □

Tétel. A lexikografikus szimplex módszer alkalmazásakor a szimplex tábla utolsó sora, a δ_0 sorvektor iterációról iterációra lexikografikusan nő.

Bizonyítás. A B bázisról a $B_1 = B - \{\mathbf{a}_j\} + \{\mathbf{a}_k\}$ bázisra történő áttérés (12) transzformációs formulái a szimplex tábla utolsó sorának a transzformációjára azt adják, hogy

$$\delta_0^{(1)} = \delta_0 - \frac{d_{0k}}{d_{jk}}\delta_j.$$

Ezt átrendezve:

$$\delta_0^{(1)} - \delta_0 = -\frac{d_{0k}}{d_{jk}}\delta_j \succ \mathbf{0},$$

hiszen $d_{0k} < 0, d_{jk} > 0$ és $\delta_j \succ \mathbf{0}$. □

Az utoljára bizonyított tételből már egyszerűen következik a lexikografikus szimplex módszer végessége. Bár a célfüggvény értékének az iterációnkénti határozott növekedését továbbra se tudjuk biztosítani, az utolsó tétel értelmében a lexikografikus szimplex módszer során a szimplex tábla teljes utolsó sora viszont lexikografikusan nő. Ez elég ahhoz, hogy egyetlen olyan megengedett bázis se térhessen vissza, amelyen már jártunk korábban. Ez a megengedett bázisok véges száma miatt a lexikografikus szimplex módszer végességét biztosítja.

4. A kétfázisú szimplex módszer

A kétfázisú szimplex módszert G. B. Dantzig és A. Orden dolgozták ki a *Rand Corporation*–nél az 1950-es évek elején. Ebben a szakaszban az ő tárgyalásmódjuk szerint haladunk, melyet Prékopa András is követett [12] jegyzetében.

Az 2. szakaszban megismert algoritmussal csak akkor tudunk egy lineáris programozási feladatot megoldani, ha ismerjük a feladat egy megengedett bázisát. Bizonyos típusú feladatok esetén azok jellegéből következően könnyen adódik megengedett bázis, amelyből a szimplex algoritmussal történő megoldásuk elindítható. Ilyen például az szakaszban a termékösszetétel optimalizálására felírt (1) feladat. Ebben a feladatban könnyen elfogadható az a további feltétel, hogy minden i-re $b_i \geq 0$, hiszen azok az optimalizálási időszak alatt rendelkezésre álló erőforrás mennyiségeket modellezik. Minden lineáris korlátozó feltételhez bevezetve az $x_{n+i} \geq 0$ segédváltozókat, a standard alakra hozott feladatban éppen az ezekhez az új változókhoz tartozó egység oszlopvektorok fognak triviális induló megengedett bázist alkotni. Itt a triviális szó nem annyira a megengedett bázis triviális megtalálására, mint inkább arra utal, hogy ez az induló megengedett bázis még azzal a további jó tulajdonsággal is rendelkezik, hogy a benne szereplő oszlopvektorok megfelelő sorrendű felsorolása mellett a mátrixa egységmátrix. Ez a tulajdonsága pedig lényegesen leegyszerűsíti, triviálissá teszi a hozzátartozó szimplex tábla felírását, mivel az (7) előállításokban szereplő d_{ip} együtthatók számításához nem szükséges semmiféle lineáris egyenletrendszert sem megoldani, hiszen azok az előállítandó oszlopvektorok koordinátáival egyenlők.

Sajnos nem minden esetben ez a helyzet. Tegyük fel, hogy a standard alakra transzformált lineáris programozási feladatban a jobboldalon álló számok mind nemnegatívak. Ez nem jelenti az általánosság megszorítását, hiszen az egyenlőség kialakítása után, ha a jobboldalon negatív szám állna, szabad az egyenletet mínusz eggyel megszorozni. Ekkor meg kell vizsgálni az egyenletrendszer együtthatómátrixát. Ha létezik benne (akár szétszórtan is) egy teljes egységmátrix, akkor annak az oszlopvektorait megfelelő sorrendben véve egy bázisba, az triviális induló megengedett bázis lesz. Ha nem ez a helyzet, akkor adjunk minden feltételi egyenlőséghez egy nemnegatív értékeket felvevő, úgynevezett **mesterséges változó**t, majd egy előzetes fázisképpen minimalizáljuk ezek összegét, mint **mesterséges célfüggvény**t, bízva abban, hogy azok mind eltüntethetők. Ha az i-edik feltételi egyenegyenlőséghez bevezetett mesterséges változót $x_{n+i}^{(a)}$ vel jelöljük, és a minimalzálandó célfüggvényt maximalizálandóvá alakítjuk, akkor az előzetes fázisban megoldandó feladat a következő:

$$
\begin{array}{ccccccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & +x_{n+1}^{(a)} & & & & = b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & & +x_{n+2}^{(a)} & & & = b_2 \\
\vdots & & \vdots & & \ddots & & \vdots & & & \ddots & & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & & & & +x_{n+m}^{(a)} & = b_m \\
x_1 \geq 0, & & x_2 \geq 0, & & \dots & & x_n \geq 0, & x_{n+1}^{(a)} \geq 0, & x_{n+2}^{(a)} \geq 0, & \dots & x_{n+m}^{(a)} \geq 0 & \\
(& & & & & & & -x_{n+1}^{(a)} & -x_{n+2}^{(a)} & \dots & -x_{n+m}^{(a)}) \to \max &
\end{array}
$$

Ez egy standard alakú lineáris programozási feladat, melyet az előzőkben megismert algoritmussal meg tudunk oldani, hiszen a bevezetett mesterséges változókhoz tartozó egység oszlopvektorok a feladat triviális induló megengedett bázisát adják. A fenti feladatnak a szimplex módszer alap algoritmusával történő megoldását **a szimplex módszer első fázisának** nevezzük. Mivel nemnegatív változók negatív összege nullánál nagyobb nem lehet, az első fázis feladatnak mindíg van véges optimuma. A véges optimum értékét illetően azonban két esetet kell megkülönböztetni:

1. *Eset.* Az első fázis feladat optimum értéke negatív.

   Ekkor az eredeti, mesterséges változók nélküli lineáris programozási feladatnak nincs megengedett megoldása. Ha lenne ugyanis $\mathbf{x}^*$ megengedett megoldása, azaz $A\mathbf{x}^* = \mathbf{b}, \mathbf{x}^* \geq \mathbf{0}$ lenne, akkor erre az $\mathbf{x}^*$-ra és $\mathbf{x}^{(a)} \equiv \mathbf{0}$-ra $A\mathbf{x}^* + E\mathbf{x}^{(a)} = \mathbf{b}, \mathbf{x}^* \geq \mathbf{0}, \mathbf{x}^{(a)} \geq \mathbf{0}$ volna, azaz $\mathbf{x}^*$ és $\mathbf{x}^{(a)} \equiv \mathbf{0}$ az első fázis feladat egy nulla célfüggvényértékű megengedett megoldása volna. Ez pedig ellentmond annak, hogy az első fázis feladat optimum értéke negatív.

2. *Eset.* Az első fázis feladat optimum értéke nulla.

   Ekkor minden mesterséges változó nulla értékű kell, hogy legyen az optimális megoldásban, hiszen nemnegatív változók összege (negatív összege) csak így lehet nulla. Ez ismét kétféleképpen lehetséges:

   (a) *Eset.* Nem maradt mesterséges változó az optimális bázisban.

   Ekkor az optimális bázisban m számú eredeti oszlopvektor van, így ezek az eredeti (mesterséges változók nélküli) feladat megengedett bázisát alkotják. Az optimális szimplex táblából a mesterséges változóknak megfelelő oszlopok elhagyhatók, a mesterséges célfüggvénynek megfelelő $z - c$ sor felváltható az eredeti célfüggvény hasonló sorával és az eredeti lineáris programozási feladat optimális megoldása meghatározható az előzőkben megismert szimplex módszer alap algoritmusával. Ezt a **szimplex módszer második fázisának** nevezzük.

(b) *Eset.* Maradt egy, vagy több mesterséges változó az optimális bázisban.

Mivel tudjuk, hogy a 2. esetben az optimális megoldásban minden mesterséges változó nulla értékű, azért ezek a mesterséges változók nulla szinten vannak az optimális bázisban. Ez a tény lehetőséget ad arra, hogy megkíséreljük őket kicserélni eredeti oszlopvektorokra. Ennek a technikáját itt nem részletezzük, csak annyit jegyzünk meg, hogy ilyenkor generáló elemként egyaránt lehet pozitív és negatív értékű elemet is választani. Előfordulhat azonban, hogy egy vagy több nulla szinten az optimális bázisban maradt mesterséges változóhoz tartozó szimplex tábla sorban már csak csupa nulla elemet találunk az eredeti oszlopvektorok alatt. Ilyenkor ez az utólagos cserélgetés elakad, ezek a mesterséges változók nulla szinten véglegesen bentmaradtak az optimális bázisban. Tehát ismét két esetet kell megkülönböztetnünk:

i. *Eset.* Minden mesterséges változót sikerült az optimális bázisból kivinni.

Ezzel visszajutottunk a 2a. esetre, ismét folytathatjuk az eredeti lineáris programozási feladat megoldását a második fázissal.

ii. *Eset.* Maradt egy, vagy több mesterséges változó véglegesen az optimális bázisban.

Ekkor az optimális bázisban m-nél kevesebb eredeti oszlopvektor maradt, ezért nem nyertünk egy teljes megengedett induló bázist az eredeti lineáris programozási feladathoz. Szerencsére meg lehet mutatni, hogy ilyenkor az optimális bázisban nulla szinten végleg bentmaradt mesterséges változók olyan feltételi egyenlőségekhez lettek bevezetve, amelyek az összes többi feltételi egyenlőségnek következményei. Az eredeti feladatból ezek a feltételi egyenlőségek ezért elhagyhatók, és így már annyi eredeti oszlopvektor maradt az optimális bázisban, amennyi egyáltalán maradhatott. Ezekből indulva a 2a. esethez hasonlóan folytatni lehet az eredeti feladat megoldását a második fázissal. A különbség csak anynyi, hogy most nem csak a mesterséges változókhoz tartozó oszlopokat, hanem a mesterséges változókhoz tartozó sorokat is el kell hagyni az első fázis feladat optimális szimplex táblájából. Ezután lehet venni az eredeti feladat célfüggvényének megfelelő $z - c$ sort, és ezzel folytatni az eredeti lineáris programozási feladat megoldását a második fázissal.

A szimplex módszer első és második fázisát együtt **kétfázisú szimplex módszer**nek nevezzük és amint láttuk, egy általános lineáris programozási feladat megoldásához tipikusan erre van szükség. A kétfázisú szimplex módszerre vonatkozóan két fontos észrevételt teszünk.

Az első az, hogy csak az egyszerűbb leírhatóság érdekében vezettünk be minden feltételi egyenlethez mesterséges változót. Nyilvánvalóan szükségtelen mesterséges változó bevezetése olyan feltételi egyenlethez, amelyben van olyan változó, hogy az csak abban az egyenletben szerepel egy (esetleg más pozitív) együtthatóval. Ekkor (az esetleges egytől különböző pozitív együtthatóval az egyenletet leosztva) ez a változó szerepeltethető az első fázis feladat triviális induló megengedett bázisában. Természetesen ilyenkor is csak a mesterséges változók negatív összegét kell maximalizálni. A feladat kézi megoldása során így eljárva sok számolást megtakaríthatunk magunknak.

A második észrevétel az, hogy a szimplex módszer alap algoritmusát mindig olyan feladatra alkalmazzuk, amelyben **a feltételi egyenletrendszer teljes sorrangú**, hiszen az első fázisban a mesterséges változók bevezetésével mesterségesen ilyenné tettük a megoldandó feladatot, a második fázisban pedig már elhagytuk az eredeti feltételi egyenletek közül az esetleges következmény feltételeket. Ezt az eddigi jelölés rendszerünkben azzal lehet kifejezésre juttatni, hogy r helyett m-et használhatunk a bázisvektorok felsorolásakor.

5. A módosított szimplex módszer

A módosított szimplex módszer kidolgozása G. B. Dantzig nevéhez kötődik ([4]), majd W. Orchard-Hays fejlesztette azt tovább (lásd [10] és [11]).

A kétfázisú szimplex módszer fontos következménye az, hogy a szimplex módszer alap algoritmusa alkalmazásakor mindig feltehetjük, hogy a megoldani kívánt lineáris programozási feladat feltételi egyenletrendszere teljes sorrangú, azaz a B megengedett bázisok m számú együttható oszlopvektorból állnak. Ha nem csak a bázisvektorok halmazát jelöljük ezentúl B-vel, hanem a belőlük összeállított $m \times m$ méretű mátrixot is, akkor B-ről tudjuk, hogy négyzetes és így a bázisvektorok lineáris függetlensége miatt invertálható mátrix. Gondoljuk meg, nem lehetne-e a szimplex módszer alap algoritmusának a végrehajtásában ezt az észrevételt felhasználni. Ez mindenekelőtt azzal az előnnyel járna, hogy felhasználhatóvá válnának a mátrix invertálásra korábban kifejlesztett, numerikusan stabil, megbízható számítógépes programok, illetve olyan magasabbszintű programnyelvek, amelyekben a mátrixok inverzét egyetlen utasítással nyerhetjük.

Először nézzük meg, hogy az aktuális megengedett bázis mátrixa B^{-1} inverzének ismeretében hogyan számíthatók a szimplex tábla felírásához szükséges $d_{ip}, i \in I_B, i = 0; p = 0, 1, 2, \ldots, n$ számok.

Jelölje a szimplex tábla oszlopaiban álló számokból (az utolsó, $z - c$-vel jelölt sorban lévőtől eltekintve) alkotott m-méretű oszlopvektorokat

$$\mathbf{d}_p = \begin{pmatrix} d_{i_1p} \\ d_{i_2p} \\ \vdots \\ d_{i_mp} \end{pmatrix}, p = 0, 1, 2, \ldots n.$$

Mivel

$$\mathbf{a}_p = \sum_{i \in I_B} d_{ip}\mathbf{a}_i = (\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \ldots, \mathbf{a}_{i_m}) \begin{pmatrix} d_{i_1p} \\ d_{i_2p} \\ \vdots \\ d_{i_mp} \end{pmatrix} = B\mathbf{d}_p, p = 0, 1, 2, \ldots n,$$

azért

$$\mathbf{d}_p = B^{-1}\mathbf{a}_p, p = 0, 1, 2, \ldots n.$$

Jelölje továbbá a célfüggvény B bázisnak megfelelő komponenseiből alkotott sorvektort:

$$\mathbf{c}'_B = (c_{i_1}, \ldots, c_{i_m}).$$

Ekkor az utolsó, $z - c$-vel jelölt sorban lévő $d_{0p}, p = 0, 1, 2, \ldots, n$ számok egyszerűen úgy számolhatók, hogy

$$\begin{aligned} d_{0p} &= z_p - c_p = \sum_{i \in I_B} d_{ip} c_i - c_p = (c_{i_1}, \ldots, c_{i_m}) \begin{pmatrix} d_{i_1 p} \\ d_{i_2 p} \\ \vdots \\ d_{i_m p} \end{pmatrix} - c_p = \\ &= \mathbf{c}'_B \mathbf{d}_p - c_p = \mathbf{c}'_B B^{-1} \mathbf{a}_p - c_p, p = 0, 1, 2, \ldots, n. \end{aligned}$$

A szimplex módszer alap algoritmusát végiggondolva, azonnal láthatjuk, hogy az aktuális megengedett bázis mátrixa inverzének rendelkezésre állásra mellett egy iterációs lépés végrehajtásához nincs szükség a teljes szimplex táblában foglalt információra. Ezt kihasználva az alábbi algoritmus-változat készíthető:

Explicit bázisinverz szimplex módszer

1. *Lépés.* Számítsuk sorra a $d_{0p} = c'_B B^{-1} a_p - c_p, p = 1, 2, \ldots, n$ számokat. Az első negatív értéket jelöljük d_{0k}-val és menjünk a 2. lépésre. Ha minden p-re $d_{0p} \geq 0, p = 1, 2, \ldots, n$, akkor álljunk le, az aktuális B megengedett bázis a feladat optimális bázisa.

2. *Lépés.* Számítsuk a $d_k = B^{-1} a_k$ vektort, ha ennek a komponenseire $d_{ik} \leq 0, i \in I_B$, akkor álljunk le, a lineáris programozási feladatnak nincs véges optimuma. Különben számítsuk a $d_0 = B^{-1} a_0$ vektort is és a $d_{ik} > 0, i \in I_B$ komponensekkel képezzük a d_{i0}/d_{ik} hányadosokat. Legyen $j \in I_B$ egy olyan index, amelyre a képzett hányadosok minimuma valósul meg. Menjünk a 3. lépésre.

3. *Lépés.* Hagyjuk el a bázisból az a_j vektort és vegyük hozzá az a_k vektort. Határozzuk meg a módosított bázismátrix inverzét és menjünk az 1. lépésre.

A számítások ílyen végrehajtásának nagy előnye a numerikus stabilitás, feltéve, hogy a 3. lépésben a módosított bázismátrix inverzét egy megbízható mátrix invertáló rutinnal végezzük. Ugyanez azonban az algoritmus hátránya is, hiszen előnytelennek tűnik az, hogy mintegy minden iterációban elfelejtsük a bázismátrix inverzét, és egy alig megváltozott (egy oszlopban különböző) bázismátrix inverzét úgy számítsuk újra, mintha az előző inverzét nem is ismertük volna. A következőkben megmutatjuk, hogy milyen kapcsolat van két olyan bázis mátrix inverze között, amelyek egymástól csak egy oszlopukban különböznek.

Legyen

$$B = \left\{ \mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_q-1}, \mathbf{a}_j, \mathbf{a}_{i_q+1}, \dots, \mathbf{a}_{i_m} \right\},$$

azaz az a_j vektor legyen a q-adik helyen a B bázisban és egy további a_k oszlopvektorra legyen

$$\mathbf{a}_k = \sum_{i \in I_B} d_{ik} \mathbf{a}_I, d_{jk} \neq 0.$$

Ekkor a B és a

$$B_1 = \left\{ \mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_q-1}, \mathbf{a}_k, \mathbf{a}_{i_q+1}, \dots, \mathbf{a}_{i_m} \right\}$$

bázisok mátrixai között a következő kapcsolat áll fent:

$$\begin{pmatrix} 1 & & & d_{i_1 k} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & d_{i_q-1,k} & & & \\ & & & d_{jk} & & & \\ & & & d_{i_q+1,k} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & d_{i_m k} & & & 1 \end{pmatrix} \left(\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_q-1}, \mathbf{a}_j, \mathbf{a}_{i_q+1}, \dots, \mathbf{a}_{i_m} \right) = \\ = \left(\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_q-1}, \mathbf{a}_k, \mathbf{a}_{i_q+1}, \dots, \mathbf{a}_{i_m} \right),$$

vagy tömörebben írva

$$\begin{pmatrix} 1 & & & d_{i_1 k} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & d_{i_q-1,k} & & & \\ & & & d_{jk} & & & \\ & & & d_{i_q+1,k} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & d_{i_m k} & & & 1 \end{pmatrix} B = B_1.$$

Ezt a mátrix egyenletet invertálva azt kapjuk, hogy

$$B_1^{-1} = \begin{pmatrix} 1 & & & -\frac{d_{i_1 k}}{d_{jk}} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & -\frac{d_{i_q-1,k}}{d_{jk}} & & & \\ & & & \frac{1}{d_{jk}} & & & \\ & & & -\frac{d_{i_q+1,k}}{d_{jk}} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -\frac{d_{i_m k}}{d_{jk}} & & & 1 \end{pmatrix} B^{-1}.$$

Az új bázis mátrix inverzét a régiből áttranszformáló, úgynevezett elemi transzformáló oszlop mátrixra bevezetve az

$$F_1 = \begin{pmatrix} 1 & & & -\frac{d_{i_1k}}{d_{jk}} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & -\frac{d_{i_q-1,k}}{d_{jk}} & & & \\ & & & \frac{1}{d_{jk}} & & & \\ & & & -\frac{d_{i_q+1,k}}{d_{jk}} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -\frac{d_{i_mk}}{d_{jk}} & & & 1 \end{pmatrix}$$

jelölést, végülis azt kaptuk, hogy

$$B_1^{-1} = F_1 B^{-1}.$$

A szimplex módszer alkalmazásakor vagy van triviális induló megengedett bázisunk, vagy a kétfázisú szimplex módszer első fázisát ugyancsak a triviális bázisból indítjuk. Ezért az induló bázismátrix minden esetben az egységmátrix, és így az s-edik iterációban az aktuális bázismátrix inverze a következő úgynevezett szorzat alakban áll elő:

$$B_s^{-1} = F_s F_{s-1} \cdots F_2 F_1.$$

A bázismátrix inverzének ezt az alakját az előbb leírt algoritmusban a következő számításokban használjuk. Az 1. lépésben a $\pi' = \mathbf{c}'_B B^{-1}$ úgynevezett árazóvektor számításakor

$$\pi' = \mathbf{c}'_B B^{-1} = \mathbf{c}'_B F_s F_{s-1} \cdots F_2 F_1, \tag{15}$$

a 2. lépésben pedig a $\mathbf{d}_k = B^{-1}\mathbf{a}_k$ és $\mathbf{d}_0 = B^{-1}\mathbf{a}_0$ transzformált vektorok előállításakor

$$\begin{aligned} \mathbf{d}_k &= F_s F_{s-1} \cdots F_2 F_1 \mathbf{a}_k, \\ \mathbf{d}_0 &= F_s F_{s-1} \cdots F_2 F_1 \mathbf{a}_0. \end{aligned} \tag{16}$$

Az (15) számítások egy sorvektor elemi transzformáló oszlopvektorokkal jobbról való szorzásának a sorozatából állnak, mely szorzások során az elemi transzformáló oszlopmátrixokat az előállítási sorrendjükkel ellentétes, visszafelé haladó sorrendben használjuk fel. Ezért ezt a számítás sorozatot **visszafelé haladó transzformáció**nak (angolul: "backward transformation", vagy rövidítve BTRAN) nevezzük. Ennek minden lépése során egy

$$\mathbf{x}' = (x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_m)$$

m-méretű sorvektort szorzunk jobbról egy, a j-edik oszlopvektorában nem triviális

$$\begin{pmatrix} 1 & & & \eta_1 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & \eta_{j-1} & & & \\ & & & \eta_j & & & \\ & & & \eta_{j+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & \eta_m & & & 1 \end{pmatrix}$$

elemi transzformáló oszlopmátrixszal. Ez a transzformáció azonnal láthatóan az $\mathbf{x}'$ sorvektort a következő sorvektorba transzformálja át:

$$(x_1, \ldots, x_{j-1}, \sum_{i=1}^{m} x_i \eta_i, x_{j+1}, \ldots, x_m). \tag{17}$$

Az (16) számítások egy oszlopvektor elemi transzformáló oszlopvektorokkal balról való szorzásának a sorozatából állnak, mely szorzások során az elemi transzformáló oszlopmátrixokat az előállítási sorrendjükkel megegyező, előre haladó sorrendben használjuk fel. Ezért ezt a számítás sorozatot **előre haladó transzformáció**nak (angolul: "forward transformation", vagy rövidítve FTRAN) nevezzük. Ennek minden lépése során egy

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{j-1} \\ y_j \\ y_{j+1} \\ \vdots \\ y_m \end{pmatrix}$$

m-méretű oszlopvektort szorzunk balról egy, a j-edik oszlopvektorában nem triviális

$$\begin{pmatrix} 1 & & & \eta_1 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & \eta_{j-1} & & & \\ & & & \eta_j & & & \\ & & & \eta_{j+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & \eta_m & & & 1 \end{pmatrix}$$

elemi transzformáló oszlopmátrixszal. Ez a transzformáció azonnal láthatóan az $\mathbf{y}$ oszlopvektort a következő oszlopvektorba transzformálja át:

$$\begin{pmatrix} y_1 + \eta_1 y_j \\ \vdots \\ y_{j-1} + \eta_{j-1} y_j \\ \eta_j y_j \\ y_{j+1} + \eta_{j+1} y_j \\ \vdots \\ y_m + \eta_m y_j \end{pmatrix}. \tag{18}$$

Mind az (17), mind pedig az (18) transzformációk elemi számításokat tartalmaznak, azok bármely magasszintű programozási nyelven néhány soros programrészlettel megvalósíthatók. Ezeket beépítve az explicit bázisinverz szimplex módszer algoritmusába nyerjük a módosított szimplex módszert:

Módosított szimplex módszer

1. *Lépés.* Számítsuk ki egy BTRAN transzformációval a $\pi' = \mathbf{c}'_B F_s F_{s-1} \cdots F_2 F_1$ árazóvektort, majd számítsuk sorra a $d_{0p} = \pi' \mathbf{a}_p - c_p, p = 1, 2, \ldots, n$ számokat. Az első negatív értéket jelöljük d_{0k}-val és menjünk a 2. lépésre. Ha minden p-re $d_{0p} \geq 0, p = 1, 2, \ldots, n$, akkor álljunk le, az aktuális B megengedett bázis a feladat optimális bázisa.

2. *Lépés.* Számítsuk ki egy FTRAN transzformációval a $\mathbf{d}_k = F_s F_{s-1} \cdots F_2 F_1 \mathbf{a}_k$ vektort, ha ennek a komponenseire $d_{ik} \leq 0, i \in I_B$, akkor álljunk le, a lineáris programozási feladatnak nincs véges optimuma. Különben számítsuk ki egy további FTRAN transzformációval a $\mathbf{d}_0 = F_s F_{s-1} \cdots F_2 F_1 \mathbf{a}_0$ vektort is és a $d_{ik} > 0, i \in I_B$ komponensekkel képezzük a d_{i0}/d_{ik} hányadosokat. Legyen $j \in I_B$ egy olyan index, amelyre a képzett hányadosok minimuma valósul meg. Menjünk a 3. lépésre.

3. *Lépés.* Hagyjuk el a bázisból az $\mathbf{a}_j$ vektort és vegyük hozzá az $\mathbf{a}_k$ vektort. Készítsünk el a $\mathbf{d}_k$ vektor komponenseiből egy új elemi transzformáló oszlopmátrixot, tároljuk a nemtriviális oszlopát, valamint annak oszlopindexét és menjünk az 1. lépésre.

Látható, hogy a módosított szimplex módszer "lelkét" a BTRAN és az FTRAN transzformációk jelentik. Sajnos azonban az is látható, hogy elveszítettük az explicit bázisinverz szimplex módszer numerikus stabilitását, hiszen a bázisba bejövő $\mathbf{a}_k$ és a bázist elhagyó $\mathbf{a}_j$ oszlopvektorok választásakor nem lehet arra ügyelni, hogy a d_{jk} **generáló**, vagy más szóval **pivot elem** a lehető legnagyobb értékű legyen, hanem azokat a szimplex módszer viszonylag merev szabályai szerint kell választanunk. Ugyanakkor az is jellemző a szimplex módszerre, hogy egyes oszlopvektorok többször kikerülnek, majd később ismét visszakerülnek a bázisba. Ez a többszörösen ismétlődő csere nagyon megnövelheti az aktuális bázismátrix inverzét szorzat alakban előállító elemi transzformáló oszlopmátrixok számát. Ez a két rossz tulajdonság indokolja azt, hogy a módosított szimplex módszer végrehajtása során időnként célszerű leállni, és egy úgynevezett **újrainvertálás**sal ismét előállítani az aktuális bázis inverzét. Ekkor, mivel ismerjük a szimplex iterációkkal elért bázist, annak az inverzét előlről indulva újra elő tudjuk úgy állítani, hogy a generáló elemeket nem a szimplex módszer szabályai szerint, hanem a numerikus stabilitás szempontjai szerint választjuk. Egy egy ilyen újrainvertálás számítási időt igényel ugyan, azonban az utána végrehajtott szimplex iterációk nem csak numerikusan lesznek pontosabbak, hanem a számítási sebességük is számottevően javul. A legnagyobb méretű lineáris programozási feladatok megoldásakor is az a célszerű, hogy mintegy 50 iterációnként újrainvertálást hajtsunk végre.

6. A duál szimplex módszer

A duál szimplex módszert C. E. Lemke dolgozta ki (lásd [9]).

Tekintsük példaként a következő lineáris programozási feladatot.

$$
\begin{array}{rcrcr}
2x_1 & + & x_2 & \geq & 1 \\
2x_1 & + & 5x_2 & \geq & 4 \\
-x_1 & + & x_2 & \geq & 0 \\
-x_1 & - & x_2 & \geq & -5 \\
-x_1 & - & 2x_2 & \geq & -4 \\
x_1 \geq 0, & & x_2 \geq 0 & & \\
(5x_1 & + & 4x_2) & \rightarrow & \min
\end{array}
$$

Szorozzuk az egyenlőtlenségeket mínusz eggyel, majd vezessük be az x_3, x_4, x_5, x_6, x_7 nemnegatív segédváltozókat, hogy a feltételi egyenlőtlenségekből egyenlőségeket hozzunk létre:

$$
\begin{array}{rrrrrrrcr}
-2x_1 & -\ x_2 & +\ x_3 & & & & & = & -1 \\
-2x_1 & -\ 5x_2 & & +\ x_4 & & & & = & -4 \\
x_1 & -\ x_2 & & & +\ x_5 & & & = & 0 \\
x_1 & +\ x_2 & & & & +\ x_6 & & = & 5 \\
x_1 & +\ 2x_2 & & & & & +\ x_7 & = & 4 \\
x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_5 \geq 0, & x_6 \geq 0, & x_7 \geq 0 & & \\
(5x_1 & +\ 4x_2 & & & & &) & \rightarrow & \min
\end{array}
\tag{19}
$$

Ha az eddig megismert szimplex módszerrel akarjuk megoldania (19) feladatot, akkor mivel a $B = \{\mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7\}$ triviális induló bázis nem megengedett, a kétfázisú szimplex módszer mindkét bázisára szükség van. Mégis írjuk fel a triviális induló bázisnak megfelelő szimplex táblát:

B	$\mathbf{a}_0$	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$
$\mathbf{a}_3$	-1	-2	-1	1	0	0	0	0
$\mathbf{a}_4$	-4	-2	-5	0	1	0	0	0
$\mathbf{a}_5$	0	1	-1	0	0	1	0	0
$\mathbf{a}_6$	5	1	1	0	0	0	1	0
$\mathbf{a}_7$	4	1	2	0	0	0	0	1
$z-c$	0	-5	-4	0	0	0	0	0

(20)

Ebből leolvasható a triviális induló bázis egy másik "jó tulajdonsága". Mivel az (19) lineáris programozási feladat célfüggvénye minimalizálandó, azért a $z - c$ különbségeknek nullánál kisebb vagy egyenlőnek kell lenni ahhoz, hogy egy bázis optimális legyen.

Ez pedig a (20) szimplex táblából leolvashatóan a triviális induló bázis esetében teljesül. Felmerül ezért az a gondolat, hogy a szimplex táblának ezt a "jó tulajdonságát" megőrizve próbáljuk meg elérni, hogy a bázis megengedettsége is teljesüljön, azaz $d_{i0} \geq 0, i \in I_B$ legyen.

Definíció. A B bázis **primál megengedett**, ha a hozzá tartozó szimplex táblában $d_{i0} \geq 0, i \in I_B$.

Definíció. A B bázis **duál megengedett**, ha a hozzá tartozó szimplex táblában minimalizálandó célfüggvény esetén $d_{0p} \leq 0, p \notin I_B$, maximalizálandó célfüggvény esetén $d_{0p} \geq 0, p \notin I_B$.

Az eddig tárgyalt szimplex módszert ezentúl **primál szimplex módszer**nek fogjuk nevezni, ennek a lényege az volt, hogy primál megengedett bázisból indulva, a primál megengedettség megőrzésével igyekezett elérni, hogy a bázis duál megengedett is legyen. Amennyiben ezt nem lehetett elérni, akkor be tudtuk látni azt, hogy a megoldani kívánt feladatnak nincs véges optimuma.

Ebben a szakaszban olyan szimplex módszert vezetünk be, amelyet **duál szimplex módszer**nek nevezünk, és amely lényege az lesz, hogy duál megengedett bázisból indulva, a duál megengedettség megőrzésével igyekszik elérni, hogy a bázis primál megengedett is legyen. Amennyiben ezt nem lehet elérni, akkor be fogjuk látni, hogy a megoldani kívánt feladatnak nincs megengedett megoldása.

Ehhez tekintsük az (20) szimplex tábla mögött rejlő kifejezett alakot:

$$
\begin{array}{rlllllllll}
-1- & (-2x_1 & -\;x_2 & +\;x_3 & & & &) & = & 0 \\
-4- & (-2x_1 & -\;5x_2 & & +\;x_4 & & &) & = & 0 \\
0- & (\;\;x_1 & -\;x_2 & & & +\;x_5 & &) & = & 0 \\
5- & (\;\;x_1 & +\;x_2 & & & & +\;x_6 &) & = & 0 \\
4- & (\;\;x_1 & +\;2x_2 & & & & & +\;x_7) & = & 0 \\
 & x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_5 \geq 0, & x_6 \geq 0, & x_7 \geq 0 & & \\
0- & (5x_1 & +\;4x_2 & & & & &) & = & z \\
 & & & & & & & & & \downarrow \\
 & & & & & & & & & \min
\end{array}
$$

és alakítsuk át azt a következő úgynevezett kibővített kifejezett alakra:

$$
\begin{array}{rcrcll}
\min\; z & = & 0 & - & (-5x_1 & -4x_2) \\
x_1 & = & 0 & - & (-x_1 & \;\;) \\
x_2 & = & 0 & - & (& -\;x_2) \\
x_3 & = & -1 & - & (-2x_1 & -\;x_2) \\
x_4 & = & -4 & - & (-2x_1 & -5x_2) \\
x_5 & = & 0 & - & (\;\;x_1 & -\;x_2) \\
x_6 & = & 5 & - & (\;\;x_1 & +\;x_2) \\
x_7 & = & 4 & - & (\;\;x_1 & +2x_2)
\end{array}
$$

Ebből az együtthatókat egy másik típusú, úgynevezett duál szimplex táblába gyűjthetjük ki

B		x_1	x_2
z	0	-5	-4
x_1	0	-1	0
x_2	0	0	-1
x_3	-1	-2	-1
x_4	-4	-2	-5
x_5	0	1	-1
x_6	5	1	1
x_7	4	2	2

Tekintsük most a duál szimplex táblát általános alakban. Legyen adott az (4) standard alakú lineáris programozási feladat, amelyben a célfüggvényt ne maximalizálni, hanem minimalizálni akarjuk. Ha ehhez a feladathoz adott egy $B = \{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_m}\}$ duál megengedett bázis, bevezethetjük az $I_B = \{i_1, \dots, i_m\}$ bázis és $K = \{k_1, \dots, k_t\}$ nem bázis index halmazokat, akkor az (7) és (8) összefüggésekkel definiálni tudjuk a $d_{ip}, i = 0, i \in I_B, p = 0, p \in K$ számokat. Ezekkel az előző számpéldának megfelelően a következő, úgynevezett **duál szimplex tábla** írható fel:

B		x_{k_1}	$\dots$	x_{k_t}
z	d_{00}	d_{0k_1}	$\dots$	d_{0k_t}
x_{k_1}	0	-1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
x_{k_t}	0	0	$\dots$	-1
x_{i_1}	d_{i_10}	$d_{i_1k_1}$	$\dots$	$d_{i_1k_t}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
x_{i_m}	d_{i_m0}	$d_{i_mk_1}$	$\dots$	d_{i_mkt}

(21)

Vezessük be az (21) duál szimplex tábla oszlopaira az alábbi vektor jelöléseket:

$$\mathbf{x} = \begin{pmatrix} z \\ x_{k_1} \\ \vdots \\ x_{k_t} \\ x_{i_1} \\ \vdots \\ x_{i_m} \end{pmatrix}, \mathbf{q}_0 = \begin{pmatrix} d_{00} \\ 0 \\ \vdots \\ 0 \\ d_{i_10} \\ \vdots \\ d_{i_m0} \end{pmatrix}, \mathbf{q}_{k_1} = \begin{pmatrix} d_{0k_1} \\ -1 \\ \vdots \\ 0 \\ d_{i_1k_1} \\ \vdots \\ d_{i_mk_1} \end{pmatrix}, \dots, \mathbf{q}_{k_t} = \begin{pmatrix} d_{0k_t} \\ 0 \\ \vdots \\ -1 \\ d_{i_1k_t} \\ \vdots \\ d_{i_mkt} \end{pmatrix}.$$

Ezekkel a duál szimplex tábla, illetve a megoldani kívánt lineáris programozási feladattal ekvivalens bővített kifejezett alak tömören úgy írható, hogy

$$\mathbf{x} = \mathbf{q}_0 - \sum_{p \in K} \mathbf{q}_p x_p, \tag{22}$$

illetve egy $j \in I_B$ indexhez tartozó nem triviális sorára

$$x_j = d_{j0} - \sum_{p \in K} d_{jp} x_p.$$

Vegyük észre, hogy a duál szimplex táblából a korábbinál is szemléletesebb módon olvasható le a B bázishoz tartozó bázismegoldás, hiszen ennek nem bázis komponenseire $x_p = 0, p \in K$ és így az (22) egyenlőségből annyi marad, hogy $\mathbf{x} = \mathbf{q}_0$, mely részletesen kiírva azt adja, hogy

$$\begin{array}{ccc} z & = & d_{00} \\ x_{k_1} & = & 0 \\ \vdots & \vdots & \vdots \\ x_{k_t} & = & 0 \\ x_{i_1} & = & d_{i_1 0} \\ \vdots & \vdots & \vdots \\ x_{i_m} & = & d_{i_m 0} \end{array}$$

Az (21) duál szimplex táblának is három típusát különböztethetjük meg:

1. *Típus:* $d_{i0} \geq 0$, minden $i \in I_B$ esetén.
2. *Típus:* Létezik $i \in I_B$, hogy $d_{i0} < 0$, és egy ilyen i indexre $d_{ip} \geq 0$ minden $p \in K$ esetén.
3. *Típus:* Létezik $i \in I_B$, hogy $d_{i0} < 0$, és minden ilyen i indexre létezik olyan $p \in K$, hogy $d_{ip} < 0$.

A duál szimplex tábla fenti három típusa egymást kizárja, és az összes lehetőséget magában foglalja. Az 1. típusú duál szimplex tábla esetén igaz a következő tétel (azt a primál szimplex módszer tárgyalásakor már bizonyítottuk):

Tétel. Ha az (21) duál szimplex tábla 1. típusú, akkor a B bázis nem csak duál, hanem primál megengedett is, és így a hozzá tartozó bázismegoldás a megoldani kívánt lineáris programozási feladat optimális megoldása.

A 2. típusú duál szimplex tábla esetén a következő tétel bizonyítható:

Tétel. Ha az (21) duál szimplex tábla 2. típusú, akkor a megoldani kívánt lineáris programozási feladatnak nincs (primál) megengedett megoldása.

Bizonyítás. Legyen $j \in I_B$ egy, a 2. típusú szimplex táblának megfelelő index, azaz legyen $d_{j0} < 0$, és $d_{jp} \geq 0$ minden $p \in K$ esetén. Tekintsük a megoldani kívánt lineáris programozási feladattal ekvivalens bővített kifejezett alak $j \in I_B$ indexhez tartozó nem triviális sorát:

$$x_j = d_{j0} - \sum_{p \in K} d_{jp} x_p. \tag{23}$$

Ez az egyenlőség nem teljesülhet nemnegatív x_j és $x_p, p \in K$ változókkal, hiszen a $j \in I_B$ index választása miatt az egyenlőség jobb oldala negatív.$\square$

Tétel. Ha az (21) duál szimplex tábla 3. típusú, akkor legyen $j \in I_B$ **egy tetszőleges**, a 3. típusú szimplex táblának megfelelő index, azaz $d_{j0} < 0$, és legyen $k \in K$ **egy**

tetszőleges olyan index, amelyre a

$$\min_{\substack{p \in K \\ d_{jp} < 0}} \frac{d_{0p}}{d_{jp}} = \frac{d_{0k}}{d_{jk}} \tag{24}$$

minimum megvalósul. Ekkor a $B_1 = B - \{\mathbf{a}_j\} + \{\mathbf{a}_k\}$ bázis is duál megengedett és a hozzá tartozó bázismegoldáson a célfüggvény értéke **nem kisebb**, mint a B bázishoz tartozó bázismegoldáson volt.

Bizonyítás. A bizonyításhoz először vezessük le a duál szimplex táblát alkotó $\mathbf{q}_0, \mathbf{q}_p$, $p \in K$ oszlopvektorok transzformációs formuláit. Jelölje $\mathbf{q}_0^{(1)}, \mathbf{q}_p^{(1)}, p \in K_1$ az új, B_1 bázishoz tartozó oszlopvektorokat, ahol $K_1 = K - \{k\} + \{j\}$. Ekkor mivel az (23) egyenlőségből x_k úgy állítható elő, hogy

$$x_k = \frac{1}{d_{jk}} \left(d_{j0} - \sum_{\substack{p \in K \\ p \neq k}} d_{jp} x_p - x_j \right),$$

ezt felhasználva

$$\begin{aligned}
\mathbf{x} &= \mathbf{q}_0 - \sum_{\substack{p \in K \\ p \neq k}} \mathbf{q}_p x_p - \mathbf{q}_k x_k = \mathbf{q}_0 - \sum_{\substack{p \in K \\ p \neq k}} \mathbf{q}_p x_p - \mathbf{q}_k \tfrac{1}{d_{jk}} \left(d_{j0} - \sum_{\substack{p \in K \\ p \neq k}} d_{jp} x_p - x_j \right) = \\
&= \left(\mathbf{q}_0 + \tfrac{d_{j0}}{(-d_{jk})} \mathbf{q}_k \right) - \sum_{\substack{p \in K \\ p \neq k}} \left(\mathbf{q}_p + \tfrac{d_{jp}}{(-d_{jk})} \mathbf{q}_k \right) x_p - \left(\tfrac{1}{(-d_{jk})} \mathbf{q}_k \right) x_j
\end{aligned}$$

és így a transzformációs formulák:

$$\begin{aligned}
\mathbf{q}_j^{(1)} &= \tfrac{1}{(-d_{jk})} \mathbf{q}_k \\
\mathbf{q}_p^{(1)} &= \mathbf{q}_p + \tfrac{d_{jp}}{(-d_{jk})} \mathbf{q}_k = \mathbf{q}_p + d_{jp} \mathbf{q}_j^{(1)}, \quad p \in K_1, p \neq j \\
\mathbf{q}_0^{(1)} &= \mathbf{q}_0 + \tfrac{d_{j0}}{(-d_{jk})} \mathbf{q}_k = \mathbf{q}_0 + d_{j0} \mathbf{q}_j^{(1)}
\end{aligned} \tag{25}$$

Ezekből

$$\begin{aligned}
d_{0j}^{(1)} &= \tfrac{1}{(-d_{jk})} d_{0k}, \\
d_{0p}^{(1)} &= d_{0p} + \tfrac{d_{jp}}{(-d_{jk})} d_{0k}, p \in K_1, p \neq j.
\end{aligned}$$

A B_1 bázis duál megengedettségéhez meg kell mutatni, hogy a fenti mennyiségek nem pozitívak. Mivel tudjuk, hogy $d_{jk} < 0$, vagyis $(-d_{jk}) > 0, d_{0k} \leq 0, d_{0p} \leq 0, p \in K_1, p \neq j$, azért csak $d_{0p}^{(1)}$ nem pozitivitása szorul bizonyításra, és az is csak azokban az esetekben, amikor $d_{jp} < 0$. Ekkor azonban a bizonyítandó $d_{0p}^{(1)} \leq 0$ egyenlőtlenséget átrendezve (és ügyelve arra, hogy a negatív d_{jp}-vel való átosztáskor az egyenlőtlenség iránya megfordul) azt kapjuk, hogy

$$\frac{d_{0p}}{d_{jp}} \geq \frac{d_{0k}}{d_{jk}}, p \in K_1, p \neq j, d_{jp} < 0,$$

vagy a korábbi bázis nem bázis index halmazával

$$\frac{d_{0p}}{d_{jp}} \geq \frac{d_{0k}}{d_{jk}}, p \in K, d_{jp} < 0$$

Ez pedig az (24) úgynevezett duál kiválasztási szabály alkalmazása miatt teljesül.

Végül az (25) transzformációs formulák utolsó tagjának az első komponense azt adja, hogy

$$d_{00}^{(1)} = d_{00} + \frac{d_{j0}}{(-d_{jk})} d_{0k},$$

és ebből mivel $d_{j0} < 0, (-d_{jk}) > 0$ és $d_{0k} \leq 0$, azért

$$d_{00}^{(1)} - d_{00} = \frac{d_{j0}}{(-d_{jk})} d_{0k} \geq 0, \tag{26}$$

amint azt állítottuk.□

A most bebizonyított tétel alapján pontosan ugyanúgy készíthető el a duál szimplex módszer alap algoritmusa, mint az a primál szimplex módszer esetén történt. Az utoljára bizonyított állítás szerint a minimalizálandó célfüggcényű lineáris programozási feladatot ekkor olyan iteráció sorozattal oldjuk meg, amikor a szomszédos, duálmegengedett bázisokon át haladva a célfüggvény értéke nem csökken. Ez első hallásra érthetetlennek tűnhet, azonban a jelenségnek igen egyszerű a magyarázata. A duál szimplex iterációk során egy bővebb halmazon a célfüggvény által felvett bizonyos értelmű minim értéket próbálunk szűkebb halmazon felvett minimummá kényszeríteni, aminek "az az ára", hogy az elérhető minimum érték nő. Ebből szemléletesen látható az is, hogy ha ez az eljárás sikertelen, az csak úgy lehet, hogy a megoldani kívánt lineáris programozási feladatnak nincs megengedett megoldása.

Természetesen a duál szimplex módszerre is ki kellene dolgozni egy kétfázisú változatot, illetve meg lehetne mutatni, hogy végrehajtható explicit bázisinverz, illetve módosított módon. Ezzel itt most nem foglalkozunk, ellenben megmutatjuk, hogy a duál szimplex módszer is végrehajtható lexikografikus módon. Erre szükség is van, hiszen az (26) összefüggésből látható, hogy most is előfordulhat, hogy iterációról iterációra nem nő a célfüggvény értéke, és ezáltal ciklikusan visszatérhetnek azok a duálmegengedett bázisok, amelyeken már jártunk. Ehhez elég annyi, hogy az (26) összefüggésben $d_{0k} = 0$ legyen, vagyis az aktuális duál megengedett bázis éppen a kiválasztott k indexre legyen úgynevezett **duál degenerált**.

Definíció. Azt mondjuk, hogy az (21) **duál szimplex tábla lexikografikusan negatív**, ha minden őt alkotó $\mathbf{q}_p, p \in K$ oszlopvektor lexikografikusan negatív.

Megjegyzés. Mivel B bázis duál megengedettsége miatt a $\mathbf{q}_p, p \in K$ oszlopvektorok első komponenseire $d_{0p} \leq 0, p \in K$ (ezért kellett minimalizálandó célfüggvényű feladatra tárgyalni a duál szimplex módszert!), és ezeket a komponenseket induláskor egy negatív egységmátrix rész követi a duál szimplex táblában, azért a duál szimplex módszer mindíg lexikografikusan negatív táblából indul.

Definíció. Lexikografikus duál kiválasztási szabály. Az (24) duál kiválasztási szabályt terjesszük ki a teljes oszlopokra úgy, hogy a $k \in K$ indexet úgy válasszuk ki, hogy

$$\operatorname*{lex\,min}_{\substack{p \in K \\ d_{jp} < 0}} \frac{1}{d_{jp}} \mathbf{q}_p = \frac{1}{d_{jk}} \mathbf{q}_k \tag{27}$$

legyen.

Megjegyzés. Most is megmutatható, hogy ezzel a $k \in K$ index kiválasztása egyértelművé válik, és ha az (24) közönséges kiválasztási szabály is egyértelmű kiválasztást ad, akkor a két szabály ugyanarra az eredményre vezet.

Definíció. Lexikografikus duál szimplex módszer. A duál szimplex módszer lexikografikusan negatív duál szimplex táblából indítva, és minden iterációban a lexikografikus duál kiválasztási szabályt alkalmazva.

Tétel. A lexikografikus duál szimplex módszer során minden duál szimplex tábla lexikografikusan negatív.

Bizonyítás. Elég egy iterációs lépésről megmutatni, hogy az megőrzi a duál szimplex tábla lexikografikus negativitását. Tekintsük ehhez az (25) duál transzformációs formulák közül a $\mathbf{q}_j^{(1)}$ $\mathbf{q}_p^{(1)}, p \in K_1, p \neq j$ oszlopvektorokra vonatkozókat. Mivel $(-d_{jk}) > 0, \mathbf{q}_k \prec \mathbf{0}, \mathbf{q}_p \prec \mathbf{0}, p \in K_1, p \neq j$, azért $\mathbf{q}_j^{(1)} \prec \mathbf{0}$, és $d_{jp} \geq 0$ esetén $\mathbf{q}_p^{(1)} \prec \mathbf{0}$ is egyszerűen igazolható. Ha pedig $d_{jp} < 0$, akkor kis átrendezéssel $\mathbf{q}_p^{(1)} \prec \mathbf{0}$ azt jelenti, hogy

$$\frac{1}{d_{jp}} \mathbf{q}_p \succ \frac{1}{d_{jk}} \mathbf{q}_k, p \in K_1, p \neq j, d_{jp} < 0,$$

vagy a korábbi bázis nem bázis index halmazával

$$\frac{1}{d_{jp}} \mathbf{q}_p \succ \frac{1}{d_{jk}} \mathbf{q}_k, p \in K, d_{jp} < 0.$$

Ez pedig az (27) alkalmazása miatt mindig teljesül.□

Tétel. A lexikografikus duál szimplex módszer alkalmazásakor a duál szimplex tábla első oszlopa, a $\mathbf{q}_0$ oszlopvektor iterációról iterációra lexikografikusan nő.

Bizonyítás. A B bázisból a $B_1 = B - \{\mathbf{a}_j\} + \{\mathbf{a}_k\}$ bázisra történő áttérés (25) ranszformációs formulái a duál szimplex tábla első oszlopának a transzformációjára azt adják, hogy

$$\mathbf{q}_0^{(1)} = \mathbf{q}_0 + \frac{d_{j0}}{(-d_{jk})} \mathbf{q}_k.$$

Ezt átrendezve:

$$\mathbf{q}_0^{(1)} - \mathbf{q}_0 = \frac{d_{j0}}{(-d_{jk})}\mathbf{q}_k \succ \mathbf{0},$$

hiszen $d_{j0} < 0, (-d_{jk}) > 0$ és $\mathbf{q}_k \prec \mathbf{0}$.□
Az utoljára bizonyított tételből már ugyanúgy egyszerűen következik a lexikografikus duál szimplex módszer végessége, mint ahogyan a közönséges, vagy más néven primál szimplex módszer esetén következett.

7. A lineáris programozás dualitás tétele

A lineáris programozás dualitás tételének megalkotását nem lehet egyértelműen egyetlen személy nevéhez kötni. G. B. Dantzig 1947-ben személyesen felkereste Neumann Jánost, akit akkor sokan a világ vezető matematikusának tartottak. Miután ismertette a lineáris programozás területén addig elért eredményeit, Neumann János felállt és jó másfél órás előadást tartott a lineáris programok elméletéről, amelyben észrevette a lineáris programozás jelentőségét a játékelméletben, megsejtette a dualitás tételt és bizonyítást is adott arra. Ezt ő maga sohasem publikálta, azonban G. B. Dantzig leírta azt és oktatta is a Pentagonban hivatali beosztottjainak. G. B. Dantzig visszaemlékezéseiből (lásd [5]) kiderül, hogy ő maga azért nem publikálta a dualitás tétel bizonyítását, mert azt Neumann Jánosnak tulajdonította. Igy történhetett, hogy a dualitás tétel első, írásban megjelent bizonyítása D. Gale, H. W. Kuhn és A. W. Tucker nevéhez fűződik (lásd [8]). Érdekességként megjegyezzük, hogy a játékelmélet és a lineáris programozás kapcsolatával foglalkozó, G. B. Dantzig által írt cikk (lásd [3]) ugyanabban a kötetben jelent meg, mint D. Gale, H. W. Kuhn és A. W. Tucker dolgozata.
Tekintsük a lineáris programozási feladatot az alábbi alakban

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq & b_2 \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq & b_m \\ x_1 \geq 0, & & x_2 \geq 0, & & \dots & & x_n \geq 0 & & \\ (c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n) & \to & \max \end{array} \tag{28}$$

és rendeljük hozzá a feladat soraihoz rendre az $y_1, \dots, y_m$ új változókat, melyekkel írjuk fel a következő lineáris programozási feladatot:

$$\begin{array}{ccccccccc} a_{11}y_1 & + & a_{21}y_2 & + & \dots & + & a_{m1}y_m & \geq & c_1 \\ a_{12}y_1 & + & a_{22}y_2 & + & \dots & + & a_{m2}y_m & \geq & c_2 \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots \\ a_{1n}y_1 & + & a_{2n}y_2 & + & \dots & + & a_{mn}y_m & \geq & c_n \\ y_1 \geq 0, & & y_2 \geq 0, & & \dots & & y_m \geq 0 & & \\ (b_1y_1 & + & b_2y_2 & + & \dots & + & b_my_m) & \to & \min \end{array} \tag{29}$$

A hozzárendelés szabálya tehát az, hogy az (28) feladat minden feltételi sorának az (29) feladat egy változója és ennek megfelelően az (28) feladat minden változójának az (29) feladat egy feltételi sora fele meg, azaz az (29) feladat együttható mátrixa a (28) feladat együttható mátrixának a transzponáltja; az (28) feladatban minden feltétel kisebb vagy egyenlő típusú, míg az (29) feladatban minden feltétel nagyobb vagy egyenlő típusú; az (28) feladat célfüggvény együtthatói az (29) feladat jobboldali állandói lettek; és fordítva, az (28) feladat jobboldali állandói az (29) feladat célfüggvény együtthatói lettek; végül pedig az (28) feladatban a célfüggvényt maximalizálni, míg az (29) feladatban a célfüggvényt maximalizálni kell. Ugyanez a két lineáris programozási feladat mátrixos alakban:

$$\begin{array}{rcl} A\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \\ \mathbf{c}'\mathbf{x} & \to & \max \end{array} \tag{30}$$

illetve

$$\begin{array}{rcl} A'\mathbf{y} & \geq & \mathbf{c} \\ \mathbf{y} & \geq & \mathbf{0} \\ \mathbf{b}'\mathbf{y} & \to & \min \end{array} \tag{31}$$

Azt mondjuk, hogy az (29), illetve az (31) feladat az (28), illetve az (30) feladat **duális feladata** (röviden: **duálja**). A kiinduló (28), illetve az (30) feladat okat pedig **primál feladat**nak nevezzük. Azt az eljárást, amellyel a két feladatot egymáshoz rendeljük **primál-duál megfeleltetés**nek nevezzük.

Először a következő egyszerű tételt bizonyítjuk be:

Tétel. A duál feladat duálja maga a kiinduló primál feladat.

Bizonyítás. Induljunk ki a

$$\begin{array}{rcl} A\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \\ \mathbf{c}'\mathbf{x} & \to & \max \end{array}$$

feladatból, ennek a duálja az

$$\begin{array}{rcl} A'\mathbf{y} & \geq & \mathbf{c} \\ \mathbf{y} & \geq & \mathbf{0} \\ \mathbf{b}'\mathbf{y} & \to & \min \end{array}$$

feladat. Ezt ekvivalens átalakítással ismét primál alakra hozva:

$$\begin{array}{rcl} -A'\mathbf{y} & \leq & -\mathbf{c} \\ \mathbf{y} & \geq & \mathbf{0} \\ (-\mathbf{b})'\mathbf{y} & \to & \max \end{array}$$

Ennek a feladatnak ismét képezhető a duálisa (jelölje $\mathbf{x}$ a duális változók vektorát):

$$\begin{array}{rcl} -A\mathbf{x} & \geq & -\mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \\ (-\mathbf{c})'\mathbf{x} & \to & \min \end{array}$$

Ez pedig a kiinduló feladattal azonos, hiszen ekvivalens átalakítással azt kapjuk, hogy

$$\begin{array}{rcl} A\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \\ \mathbf{c}'\mathbf{x} & \to & \max.\square \end{array}$$

Az (28) - (29), illetve az (30) - (31) primál-duál feladatpárra érvénnyes a következő fontos tétel:

Dualitás tétel. Ha egy primál-duál feladat pár egyikének létezik megengedett megoldása és véges optimuma, akkor ugyanez fennáll a másikra is, és a két feladat optimum értéke egyenlő.

Bizonyítás. A bizonyítást három lépésben végezzük el.

(i) Tekintsükaz (30) és az (31) primál-duál feladatpárt. Tegyük fel, hogy az (30) feladatnak van megengedett megoldása és véges optimuma. Hozzuk standard alakra a feladatot:

$$\begin{array}{ccccc} A\mathbf{x} & + & E\mathbf{u} & \leq \mathbf{b} & \\ \mathbf{x} & \geq \mathbf{0}, & \mathbf{u} & \geq \mathbf{0} & \\ (\mathbf{c}'\mathbf{x} & + & \mathbf{0}'\mathbf{u}) & \to & \max \end{array}$$

és oldjuk meg a lexikografikus szimplex módszerrel. Ha szükséges első és második fázison keresztül, ha nem, akkor csak a második fázis végrehajtásával el kell jutni egy $B = \{\mathbf{a}_{i_1}, \mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_m}\}$ optimális bázishoz, amely bázis bizonyos A-beli, úgynevezett tényleges oszlopvektorokból és bizonyos E-beli, úgynevezett segéd oszlopvektorokból fog állni. A módosított szimplex módszer optimalitás tesztjét visszaidézve, a B bázis optimalitása azt jelenti, hogy

$$\begin{array}{ccccccl} \mathbf{c}'_B B^{-1}\mathbf{a}_p & - & c_p & \geq & 0, & p = 1, \ldots, n, \\ \mathbf{c}'_B B^{-1}\mathbf{e}_q & - & 0 & \geq & 0, & q = 1, \ldots, m, \end{array} \tag{32}$$

hiszen szemben a módosított szimplex módszer tárgyalásakor megoldani kívánt feladattal, most a standard alakú feladat oszlopvektorai két nagy csoportra vannak bontva, az első csoportba az $\mathbf{a}_p, p = 1, \ldots, n$ tényleges oszlopvektorok, a második csoportba az $\mathbf{e}_q, q = 1, \ldots, m$ egységvektorok, mint segéd oszlopvektorok tartoznak.

Ha bevezetjük az $\mathbf{y}' = \mathbf{c}'_B B^{-1}$ jelölést, akkor erre az $\mathbf{y}$ vektorra (32) szerint

$$\begin{array}{cccl} \mathbf{y}'\mathbf{a}_p & \geq & c_p, & p = 1, \ldots, n, \\ \mathbf{y}'\mathbf{e}_q & \geq & 0, & q = 1, \ldots, m \end{array}$$

teljesül. Ebből kis átalakítással azt kapjuk, hogy

$$\begin{array}{ccc} \mathbf{a}'_1\mathbf{y} & \geq & c_1 \\ \vdots & & \vdots \\ \mathbf{a}'_n\mathbf{y} & \geq & c_n \\ y_1 & \geq & 0 \\ \vdots & & \vdots \\ y_q & \geq & 0 \end{array}$$

vagy ugyanez mátrix alakban

$$\begin{array}{lcl} A'\mathbf{y} & \geq & \mathbf{c} \\ \mathbf{y} & \geq & \mathbf{0} \end{array}$$

vagyis az $\mathbf{y}' = \mathbf{c}'_B B^{-1}$ vektor, az (31) duális feladat megengedett megoldása. Ezért az $\mathbf{y}$ vektort duál vektornak is nevezzük, és beláttuk, hogy a duál feladatnak is létezik megengedett megoldása.

(ii) Megmutatjuk, hogy a primál feladat bármely megengedett megoldásán a primál célfüggvény értéke sohasem nagyobb, mint a duál feladat bármely megengedett megoldásán a duál célfüggvény értéke.

Legyen $\mathbf{x}$ a primál feladat tetszőleges megengedett megoldása, azaz olyan, hogy $\mathbf{x} \geq \mathbf{0}$ és

$$A\mathbf{x} \leq \mathbf{b}, \tag{33}$$

valamint legyen $\mathbf{y}$ a duál feladat tetszőleges megengedett megoldása, azaz legyen $\mathbf{y} \geq \mathbf{0}$ és

$$A'\mathbf{y} \geq \mathbf{c}. \tag{34}$$

Ha az (34) egyenlőtlenséget az $\mathbf{c}' \leq \mathbf{y}'A$ transzponált alakban tekintjük, és megszorozzuk a nemnegatív $\mathbf{x}$ vektorral jobbról, valamint az (33) egyenlőtlenséget megszorozzuk balról a nemnegatív $\mathbf{y}' \geq \mathbf{0}'$ sorvektorral, akkor a következő két egyenlőtlenséget nyerjük:

$$\mathbf{c}'\mathbf{x} \leq \mathbf{y}'A\mathbf{x}$$

és

$$\mathbf{y}'A\mathbf{x} \leq \mathbf{y}'\mathbf{b} = \mathbf{b}'\mathbf{y}.$$

Ezek együtt bizonyítják az állításunkat. Vegyük észre, hogy ezzel azt is beláttuk, hogy a duál feladatnak is létezik véges optimuma, hiszen azt láttuk be, hogy a minimalizálandó duál célfüggvénynek bármly primál megengedett megoldáson felvett célfüggvényérték egy alsó korlátja.

(iii) Megmutatjuk, hogy létezik a duál feladatnak olyan megengedett megoldása, amelyen a duál célfüggvény értéke egyenlő egy primál megengedett megoldáson (szükségszerűen a primál feladat optimális megoldásán) felvett primál célfüggvény értékkel.

Tekintsük ehhez az (i) lépésben bevezetett $\mathbf{y}' = \mathbf{c}'_B B^{-1}$ duál vektort, ezen a duál célfüggvény értéke:

$$\mathbf{b}'\mathbf{y} = \mathbf{y}'\mathbf{b} = \mathbf{c}'_B B^{-1}\mathbf{b} = \mathbf{x}'_B\mathbf{b} = \mathbf{b}'\mathbf{x}_B.\square$$

Példa. Tekintsük a következő lineáris programozási feladatot:

$$\begin{array}{rcrcl} 2x_1 & + & x_2 & \leq & 1 \\ 2x_1 & + & 5x_2 & = & 4 \\ x_1 \geq 0 & & & & \\ (5x_1 & + & 3x_2) & \rightarrow & \max \end{array} \tag{35}$$

Ahhoz, hogy ennek a feladatnak fel tudjuk írni a duálisát, először primál alakúra kell transzformálni. A (35) feladat két helyen nem felel meg a primál alak követelményeinek. A második feltétele egyenlőség alakú, és az x_2 változóra nincs nemnegetivitás megkövetelve. Az egyenlőséget két egyenlőtlenséggel helyettesítve, az x_2 szabad változót pedig két nemnegatív változó (x_2^+ pozitív rész és x_2^- negatív rész) különbségeként felírva a következő feladatra jutunk:

$$\begin{array}{rcrcrcr} 2x_1 & + & x_2^+ & - & x_2^- & \leq & 1 \\ 2x_1 & + & 5x_2^+ & - & 5x_2^- & \leq & 4 \\ -2x_1 & - & 5x_2^+ & + & 5x_2^- & \leq & -4 \\ x_1 \geq 0, & & x_2^+ \geq 0, & & x_2^- \geq 0 & & \\ (5x_1 & + & 3x_2^+ & - & 3x_2^-) & \rightarrow & \max \end{array}$$

Ha az egyes feltételi sorokhoz rendre az y_1, y_2^+, y_2^- duál változókat rendeljük, akkor a duális feladat a következő lesz:

$$\begin{array}{rcrcrcr} 2y_1 & + & 2y_2^+ & - & 2y_2^- & \geq & 5 \\ y_1 & + & 5y_2^+ & - & 5y_2^- & \geq & 3 \\ -y_1 & - & 5y_2^+ & + & 5y_2^- & \geq & -3 \\ y_1 \geq 0, & & y_2^+ \geq 0, & & y_2^- \geq 0 & & \\ (y_1 & + & 4y_2^+ & - & 4y_2^-) & \rightarrow & \min \end{array}$$

Ebben figyelembe véve, hogy a második és a harmadik egyenlőtlenség alakú feltétel összevonható egyetlen egyenlőséggé, valamint, hogy a nemnegatív y_2^+ és y_2^- változók különbsége jelölhető egy olyan y_2 változóval amelyre nincs nemnegativitás megkövetelve, a következő ekvivalens feladatra jutunk:

$$\begin{array}{rcrcl} 2y_1 & + & 2y_2 & \geq & 5 \\ y_1 & + & 5y_2 & = & 3 \\ y_1 \geq 0, & & & & \\ (y_1 & + & 4y_2) & \rightarrow & \min \end{array} \tag{36}$$

Az (35) és (36), primál-duál feladatpárt alkotó lineáris programozási feladatokat összehasonlítva, azt a szabályt lehet megfogalmazni, hogy ha a primál feladatban egy feltétel egyenlőség alakú, akkor a megfelelő változó a duál feladatban nincs nemnegativitással korlátozva, illetve ha a primál feladat egy változója nincs nemnegativitással korlátozva, akkor a duál feladat megfelelő feltétele egyenlőség alakú. Ezt az észrevételt a következő tételben általánosan is bebizonyítjuk.

Tétel. Az alábbi két lineáris programozási feladat egymás duálisa:

$$\begin{array}{rcrcl} A_{11}\mathbf{x}_1 & + & A_{12}\mathbf{x}_2 & \leq & \mathbf{b}_1 \\ A_{21}\mathbf{x}_1 & + & A_{22}\mathbf{x}_2 & = & \mathbf{b}_2 \\ \mathbf{x}_1 \geq \mathbf{0} & & & & \\ (\mathbf{c}_1'\mathbf{x}_1 & + & \mathbf{c}_2'\mathbf{x}_2) & \rightarrow & \max \end{array} \tag{37}$$

$$
\begin{array}{lclcl}
A'_{11}\mathbf{y}_1 & + & A'_{21}\mathbf{y}_2 & \geq & \mathbf{c}_1 \\
A'_{12}\mathbf{y}_1 & + & A'_{22}\mathbf{y}_2 & = & \mathbf{c}_2 \\
\mathbf{y}_1 \geq \mathbf{0} & & & & \\
(\mathbf{b}'_1\mathbf{y}_1 & + & \mathbf{b}'_2\mathbf{y}_2) & \rightarrow & \min
\end{array}
\tag{38}
$$

Bizonyítás. Induljunk ki az (37) feladatból. Amint azt a korábbi számpéldában is tettük, helyettesítsük az egyenlőség alakú második feltételi sort két egyenlőtlenséggel, és írjuk fel az $\mathbf{x}_2$ szabad változó vektort két nemnegatív vektor változó ($\mathbf{x}_2^+$ pozitív rész és $\mathbf{x}_2^-$ negatív rész) különbségeként:

$$
\begin{array}{lclclcr}
A_{11}\mathbf{x}_1 & + & A_{12}\mathbf{x}_2^+ & - & A_{12}\mathbf{x}_2^- & \leq & \mathbf{b}_1 \\
A_{21}\mathbf{x}_1 & + & A_{22}\mathbf{x}_2^+ & - & A_{22}\mathbf{x}_2^- & \leq & \mathbf{b}_2 \\
-A_{21}\mathbf{x}_1 & - & A_{22}\mathbf{x}_2^+ & + & A_{22}\mathbf{x}_2^- & \leq & -\mathbf{b}_2 \\
\mathbf{x}_1 \geq \mathbf{0}, & & \mathbf{x}_2^+ \geq \mathbf{0}, & & \mathbf{x}_2^- \geq \mathbf{0} & & \\
(\mathbf{c}'_1\mathbf{x}_1 & + & \mathbf{c}'_2\mathbf{x}_2^+ & - & \mathbf{c}'_2\mathbf{x}_2^-) & \rightarrow & \max
\end{array}
$$

Vezessük be ennek a feladatnak a feltételi soraihoz rendre a nemnegatív komponensekből álló $\mathbf{y}_1, \mathbf{y}_2^+, \mathbf{y}_2^-$ duál változó vektorokat. A duális feladat ezekkel felírva:

$$
\begin{array}{lclclcr}
A'_{11}\mathbf{y}_1 & + & A'_{21}\mathbf{y}_2^+ & - & A'_{21}\mathbf{y}_2^- & \geq & \mathbf{c}_1 \\
A'_{12}\mathbf{y}_1 & + & A'_{12}\mathbf{y}_2^+ & - & A'_{22}\mathbf{y}_2^- & \geq & \mathbf{c}_2 \\
-A'_{12}\mathbf{y}_1 & - & A'_{12}\mathbf{y}_2^+ & + & A'_{22}\mathbf{y}_2^- & \geq & -\mathbf{c}_2 \\
\mathbf{y}_1 \geq \mathbf{0}, & & \mathbf{y}_2^+ \geq \mathbf{0}, & & \mathbf{y}_2^- \geq \mathbf{0} & & \\
(\mathbf{b}'_1\mathbf{y}_1 & + & \mathbf{b}'_2\mathbf{y}_2^+ & - & \mathbf{b}'_2\mathbf{y}_2^-) & \rightarrow & \min
\end{array}
$$

Ebben a feladatban ismét a korábbi számpéldához hasonlóan észre lehet venni, hogy a második és a harmadik egyenlőtlenségek egy egyenlőséggé vonhatók össze; az $\mathbf{y}_2^+$ és $\mathbf{y}_2^-$ nemnegatív komponensekből álló vektor változók különbsége pedig jelölhető egy olyan $\mathbf{y}_2$ vektor változóval amely komponenseire nincs nemnegativitás megkövetelve. Így ekvivalens feladatként az (38) lineáris programozási feladatra jutunk, és ez bizonyítja a tétel állítását. □

Ezek után általánosan is megfogalmazhatjuk a számpéldából korábban leolvasott szabályt:

Általános primál-duál megfeleltetési szabály. Ha a primál feladatban egy feltétel egyenlőség alakú, akkor a megfelelő változó a duál feladatban nincs nemnegativitással korlátozva, illetve ha a primál feladat egy változója nincs nemnegativitással korlátozva, akkor a duál feladat megfelelő feltétele egyenlőség alakú.

Megjegyzés. Mivel az (37) és (38) lineáris programozási feladatok az eredeti primál-duál megfeleltetés szerint is primál-duál feladatpárt alkotnak, azért a dualitás tétel állítása igaz az általános primál-duál megfeleltetés szerint egymáshoz rendelt lineáris programozási feladatpárokra is.

Az általános primál-duál megfeleltetés szerinti feladatpárra vonatkozó dualitás tétel egy szép alkalmazásaként megmutatjuk, hogyan bizonyítható be az a Farkas Gyula, kolozsvári matematikus által az 1800-as évek végén megfogalmazott, és 1901-ben publikált (lásd [7]), homogén lineáris egyenlőtlenség rendszerekre vonatkozó állítás, amely az optimalizálás elmélet egyik leggyakrabban alkalmazott alap eredményévé vált.

Farkas Gyula tétele. Legyenek $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_m$ az n-dimenziós Euklideszi tér tetsző-leges vektorai. Az ezekkel felírt

$$\begin{array}{rcl} \mathbf{g}_1'\mathbf{x} & \leq & 0 \\ \mathbf{g}_2'\mathbf{x} & \leq & 0 \\ \vdots & & \vdots \\ \mathbf{g}_m'\mathbf{x} & \leq & 0 \end{array} \tag{39}$$

homogén lineáris egyenlőtlenség rendszernek az n-dimenziós Euklideszi tér egy további $\mathbf{g}$ vektorával felírt

$$\mathbf{g}'\mathbf{x} \leq 0 \tag{40}$$

homogén lineáris egyenlőtlenség akkor és csak akkor következménye, ha léteznek olyan $\lambda_1 \geq 0, \lambda_2 \geq 0, \ldots, \lambda_m \geq 0$ valós számok, hogy ezekkel

$$\mathbf{g} = \lambda_1\mathbf{g}_1 + \lambda_2\mathbf{g}_2 + \ldots + \lambda_m\mathbf{g}_m. \tag{41}$$

Bizonyítás. Először megmutatjuk, hogy ha igaz az (41) előállítás, akkor az (39) homogén lineáris egyenlőtlenség rendszernek következménye az (40) homogén lineáris egyenlőtlenség.

Ekkor ugyanis az (40) homogén lineáris egyenlőtlenség egyszerűen "levezethető" az (39) homogén lineáris egyenlőtlenség rendszerből, és így annak nyilvánvalóan következménye:

$$\begin{array}{rcll} \mathbf{g}_1'\mathbf{x} & \leq & 0 & \quad / \cdot \lambda_1 \geq 0 \\ \mathbf{g}_2'\mathbf{x} & \leq & 0 & \quad / \cdot \lambda_2 \geq 0 \\ \vdots & & \vdots & \\ \mathbf{g}_m'\mathbf{x} & \leq & 0 & \quad / \cdot \lambda_m \geq 0 \\ \hline \mathbf{g}'\mathbf{x} & \leq & 0 & \end{array}$$

hiszen érvényes a $\mathbf{g}$ vektor (41) előállítása.

Másodszor megmutatjuk, hogy ha fordítva, az (39) homogén lineáris egyenlőtlenség rendszernek következménye az (40) homogén lineáris egyenlőtlenség, akkor igaz az (41) előállítás. Ehhez tekintsük a következő primál alakú lineáris programozási feladatot:

$$\begin{array}{rcl} G\mathbf{x} & \leq & \mathbf{0} \\ \mathbf{g}'\mathbf{x} & \rightarrow & \max \end{array} \tag{42}$$

ahol az $m \times n$ méretű G együttható mátrix sorait a $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_m$ vektorok, mint sorvektorok alkotják:

$$G = \begin{pmatrix} \mathbf{g}_1' \\ \mathbf{g}_2' \\ \vdots \\ \mathbf{g}_m' \end{pmatrix}.$$

Az (42) lineáris programozási feladatnak van megengedett megoldása (például az $\mathbf{x} \equiv \mathbf{0}$ vektor), és van véges optimuma, hiszen feltettük, hogy az (39) homogén lineáris egyenlőtlenség rendszernek következménye az (40) homogén lineáris egyenlőtlenség, azaz az (42) lineáris programozási feladat $\mathbf{g}'\mathbf{x}$ maximalizálandó célfüggvénye felülről korlátos a $G\mathbf{x} \leq \mathbf{0}$ feltétel mellett. Ezért a dualitás tétel értelmében ugyanez igaz a

$$\begin{array}{ccc} G'\mathbf{y} & = & \mathbf{g} \\ \mathbf{y} \geq \mathbf{0} & & \\ \mathbf{0}'\mathbf{y} & \rightarrow & \min \end{array} \tag{43}$$

duál feladatra is, és az optimumértékek egyenlők. Mivel a G' mátrixra

$$G' = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m)\,,$$

azért az, hogy az (43) lineáris programozási feladatnak létezik megengedett megoldása azt jelenti, hogy létezik $\mathbf{y}' = (y_1, \dots, y_m) \geq \mathbf{0}'$, amellyel $\mathbf{g} = y_1\mathbf{g}_1 + y_2\mathbf{g}_2 + \dots + y_m\mathbf{g}_m$. Ekkor pedig igaz (41) előállítás. □

Megjegyezzük, hogy a most bizonyított tétel fordítva is igaz abban az értelemben, hogy a dualitás tétel is könnyen bizonyítható lenne Farkas Gyula tétele segítségével.

Hivatkozások

[1] A. Charnes Optimality and degeneracy in linear programming, *Econometrica* **20** (1952) 160–170.

[2] G. B. Dantzig Maximization of linear functions of variables subject to linear inequalities, in: *Activity Analysis of Production and Allocation*, ed. T. C. Koopmans, John Wiley and Sons, Inc., New York, 1951.

[3] G. B. Dantzig A proof of the equivalence of the programming problem and the game problem, in: *Activity Analysis of Production and Allocation*, ed. T. C. Koopmans, John Wiley and Sons, Inc., New York, 1951.

[4] G. B. Dantzig Computational algorithm of the revised simplex method, RM-1266, The Rand Corporation, 1953.

[5] G. B. Dantzig Reminiscences about the origins of linear programming, *Operations Research Letters*, **1** 43–48. Magyar fordításban: *Szigma*, **XXIII** (1992) 45–54.

[6] G. B. Dantzig, A. Orden and P. Wolfe The generalized simplex method for minimizing a linear form under linear inequality restraints, *Pacific Journal of Mathematics* **5** (1955).

[7] J. Farkas Theorie der einfachen Ungleichungen, *Journal für die reine und angewandte Mathematik* **124** (1901) 1–24.

[8] D. Gale, H. W. Kuhn and A. W. Tucker Linear programming and the theory of games, in: *Activity Analysis of Production and Allocation*, ed. T. C. Koopmans, John Wiley and Sons, Inc., New York, 1951.

[9] C. E. Lemke The dual method of solving the linear programming problem, *Naval Research Logistics Quarterly* **1** (1954) 48–54.

[10] W. Orchard-Hays A composit simplex algorithm – II., RM-1275, The Rand Corporation, 1954.

[11] W. Orchard-Hays Background, development and extensions of the revised simplex method, RM-1433, The Rand Corporation, 1954.

[12] Prékopa András, *Lineáris programozás*, Bolyai János Matematikai Társulat, Budapest, 1968.

[13] Prékopa András, A lineáris programozás egy kombinatorikai jellegű tárgyalásmódja, *Matematikai Lapok* **22** (1971) 7–24.

[14] Prékopa András, A brief introduction to linear programming. *Mathematical Scientist* **21** (1996) 85–111.