

Bevezetés

A lineáris programozás megalkotása a szovjet *Kantorovics* [9], továbbá két amerikai: *Dantzig* [5] és *Koopmans* [10] nevéhez fűződik. Kantorovics és Koopmans 1975-ben Nobel-díjat, Dantzig pedig 1976-ban az Amerikai Egyesült Államokban állami díjat kapott a lineáris programozás felfedezéséért és az ezzel kapcsolatos további alapvető tudományos eredményekért.¹

A lineáris programozás jellegét illetően sokféle vélemény alakult ki a keletkezése utáni években, és a vele megismerkedni szándékozók ma is felteszik a kérdést: mi is ez tulajdonképpen? Talán geometria, algebra, kombinatorika, numerikus módszer, függvénytan vagy számítástechnika? Sok jele van annak, hogy kezdetben elsősorban geometriai jellegű tudománynak tartották. Mielőtt Kantorovics [9] könyve 1939-ben megjelent, 1938-ban vitát rendeztek Leningrádban és geometriákat kértek fel bírálóknak. Dantzig [6] könyvének 24. oldalán ezt írja: „A convex poliéder csúcsain való haladás (mely egyébként a szimplex módszer alapja) gondolatát először elvetettük, mert intuitív alapon ineffektívnek látszott. Egy másik geometriában hatékonynak tűnt, ezért szerencsére kipróbáltuk és elfogadtuk”.

A következő szakasz szendvicsekészítésről szóló példájában látni fogjuk, hogy ha a feltételeknek megfelelő egyenlőtlenségeket ábrázoljuk a koordináta-rendszerben, a sík azon pontjainak összessége, amelyek eleget tesznek a feltételeknek, azaz az ún. megengedett megoldások egy convex sokszöget alkotnak, és hogy optimális megoldás mindig található e sokszög csúcsai között. E tények valóban azt sejtetik, hogy a geometriának nagy szerep jut a lineáris programozásban.

Később, amikor előtérbe került a lineáris programozási feladat megoldási módszerének végességi kérdése, kitűnt, hogy erre a geometriai jellegű megfontolások már nem alkalmasak és a kombinatorikai jellegű gondolatok kerültek előtérbe. Arról van szó ugyanis, hogy – amint ezt később látni fogjuk – a megoldási módszer során egyik tábláról² a másikra megyünk, bizonyos szabályok betartásával, és arra törekszünk, hogy a véges sok lehetséges tábla között egy számunkra alkalmasat találjunk. Ez pedig kombinatorikai jellegű gondolkodásmódot igényel, noha nem támaszkodunk meglevő kombinatorikai eredményekre. Az első végességi bizonyítást *Charnes* adta 1952-ben [3] cikkében, mely egy évvel később a [4] könyvben is megjelent. Egy további végességi bizonyítást tartalmaz *Dantzig, Orden* és *Wolfe* 1955-ben megjelent [7] dolgozata. Az utóbbinak egy elegánsabb változata található *Gale* 1960-ban megjelent [8] könyvében, továbbá a szerző [11] könyvében és [12] dolgozatában.

E két utóbb említett mű tárgyalásmódja mellőz minden megszorítást és egyben elemi, csak igen egyszerű lineáris algebrai fogásokat alkalmaz. A mostani cikkbe ezt építjük be úgy, hogy semmilyen matematikai előismeretet nem tételezünk fel.

Az angliai lineáris programozási iskola *Beale* [1] és a magyar származású *Vajda* [13] munkássága révén kifejlesztett egy „váltócentrikus”, függvénytan jellegű tárgyalásmódot. Ennek ma is sok híve van, legutóbb az 1976-ban Budapesten tartott IX. Nemzetközi Matematikai Programozási Szümpozion oktatási szekciójában *van de Panne* kardoskodott mellette. Ebben a cikkben elfogadjuk ugyan a változócentrikus tárgyalást, ám összeházasítjuk a kombinatorikai tárgyalással.

Mint hogy a gyakorlatban előforduló lineáris programozási feladatok általában nagy méretűek, a „hagyományos” numerikus módszerek nagy szerepet játszanak a tényleges megoldási módszer kialakítása során. Ennek számítógépes programját is meg kell alkotni, érthető tehát, hogy sokan a lineáris programozási feladat megoldását elsősorban a numerikus módszerek közé tartozónak, illetve számítástechnikai jellegűnek tekintik.

A lineáris programozás tehát, mint minden alkalmazott matematikai tudomány, sokarcú. Ám a bevezető jellegű oktatásban le kell egyszerűsíteni a dolgokat. Ennek sikerült megtalálni egy olyan lehetőségét, amely nem nélkülözi a matematikai egzakttságot. Kis méretű feladatokkal dolgozunk, ezért nincs szükségünk a kerekítési hibákra, az információátvitel és a műveletszám minimalizálására figyelő speciális eljárások beépítésére. Megmutatjuk, hogy egy alkalmas szabály betartásával véges sok lépésben el tudunk jutni egy optimális megoldáshoz.

A lineáris programozás feladata

Lineáris egyenlőtlenségekből és lineáris egyenlőségekből álló feltételeknek eleget tevő x_1, \dots, x_n rendezett szám n -esek közül ki kell választanunk olyan rendezett szám n -est, melyre egy előírt lineáris (első fokú) függvény a lehető legnagyobb, illetve a lehető legkisebb értékét veszi fel. Előfordulhat, hogy feltételeink ellentmondásosak, továbbá az is, hogy van ugyan a feltételeknek eleget tevő rendezett szám n -es, ám nincs véges nagyságú szélsőérték. E két esetben a gyakorlati feladatot rosszul fogalmaztuk meg. Ez a tény az esetek nagy részében nem dönthető el ránézéssel, hanem csak a megoldási módszer alkalmazása során tűnik ki. Ezért a feltételek összeférhetőségének és a véges szélsőérték létezésének az eldöntését is a lineáris programozás feladatához soroljuk.

Néhány, lineáris programozásra vezető gyakorlati feladat

A lineáris programozást igen széles körben alkalmazhatjuk. E néhány példa bemutatásával csupán ízelítőt adunk azokból. Az olvasó tetszésére bízunk, hogy ezek közül melyeket tanulmányozza át. A legelső, szendvicsekészítéssel

¹ A Bevezetés további részeit az utolsó bekezdés kivételével az olvasó elhagyhatja, ez nem akadályozza a megértést.

² Tábla helyett használják a táblázat szót is.

kapcsolatos feladat az egyetlen, amely nem hagyható el, ugyanis ezen illusztráljuk a lineáris programozás grafikus és algoritmikus megoldási módszereit. Az egyes feladatok megoldásait is közöljük; ezeket az olvasó a későbbiekben ismertető módszerrel maga is kiszámíthatja. A szénelosztás feladata esetében azonban ezt a módszert nem ajánljuk, erre a speciális feladatra gyorsabb módszer is van, amellyel ebben a cikkben nem foglalkozunk.

Szendvicskészítés. Egy társasági összejövetelre szendvicseket készítünk, mégpedig két fajtát. Ezek egy-egy darabja az alábbi összetételű:

I. fajta	II. fajta
2 dkg vaj	1 dkg vaj
2 dkg sonka	3 dkg sonka
3 dkg franciasaláta	2 dkg sajt
1 szelet kenyér	1 szelet kenyér

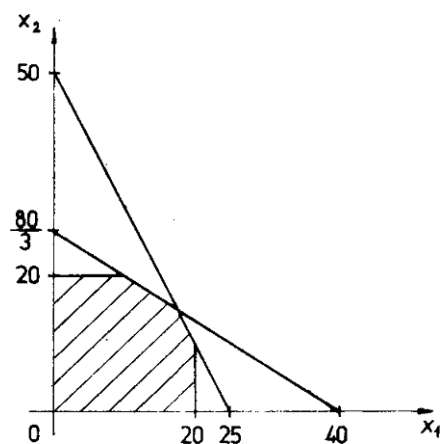
Kenyér van otthon bőven, ám a további összetevők mennyisége már korlátozott. Rendelkezésünkre áll:

50 dkg vaj
 80 dkg sonka
 60 dkg franciasaláta
 40 dkg sajt

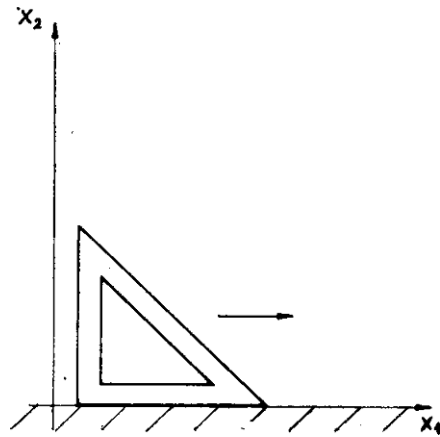
A célunk az, hogy a lehető legtöbb szendvicset készítsük. Ha x_1 , ill. x_2 jelöli az egyes szendvicsfajtákból készítendő mennyiséget, akkor feladatunk a következő:

$$\begin{array}{ll} \text{maximalizálandó} & (x_1 + x_2), \\ \text{feltéve, hogy} & 2x_1 + x_2 \leq 50, \\ & 2x_1 + 3x_2 \leq 80, \\ & 3x_1 \leq 60, \\ & 2x_2 \leq 40, \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{array}$$

A feladat megoldását grafikusan a következő módon végezhetjük el. Az 1. ábrán a vonalkázott sokszög jelenti a sík azon pontjainak összességét, amelyek eleget tesznek a feladat feltételeinek. Ezek közül a legnagyobb $x_1 + x_2$ összeget szolgáltató pontot a következő módon választhatjuk ki: egy egyenlőszárú derékszögű háromszög alakú vonalzót végigcsúsztatunk egy másik vonalzón, mely az x_1 tengellyel egybeesik – a 2. ábrának megfelelő módon –, majd megállunk akkor, amikor a vonalzó átlójának és a vonalkázott sokszögnek még van közös pontja, de a vonalzó akármilyen kis továbbhaladása esetén ilyen közös pont már nincs.



1. ábra



2. ábra

Esetünkben ez az eljárás egyetlen pontot eredményez, mely a

$$\begin{aligned} 2x_1 + x_2 &= 50, \\ 2x_1 + 3x_2 &= 80 \end{aligned}$$

egyenesek metszéspontja. Innen adódik, hogy a legnagyobb $x_1 + x_2$ összeget akkor kapjuk, amikor ³ $x_1 = 17.5$ és $x_2 = 15$.

Az eredeti gyakorlati feladat megoldását csak közelítőleg kaptuk meg, hiszen 17,5 nem egész szám, fél szendvicset pedig nem készítünk. Van módszerünk arra is, hogy az x_1 , x_2 ismeretlenekkel kapcsolatos fenti feltételeken kívül még ezekre az egész értékűség követelményét is előírva, megoldjuk a feladatot. Ezzel a bonyolultabb kérdéssel e helyen nem foglalkozunk. Megjegyezzük, hogy az egész értékűség természetesen nem minden lineáris programozási feladatban jogos követelmény, és még ha jogos is, elhanyagolhatjuk, ha nem tartjuk lényegesnek a meghatározandó darabszámok ily módon adódó csökkenését vagy túllépését.

Ha az eredeti feladattal kapcsolatban az egész értékűséget megkívánjuk, akkor – amint az az előbbi grafikus módszerrel könnyen adódik – három, ugyanazt a maximumot szolgáltató megoldást nyerünk. Ezek a következők:

$$\begin{aligned} x_1 = 18, \quad x_2 = 14, \\ x_1 = 17, \quad x_2 = 15, \\ x_1 = 16, \quad x_2 = 16. \end{aligned}$$

Optimális termelési terv készítése. Egy teherautókat gyártó vállalat egyéves termelési tervet készít. Meg kell határozni, hogy ötféle autótípus mindegyikéből mennyit gyártsanak egy év folyamán. A gyárnak van két nagyobb üzemege: a karosszériaüzem és a motorüzem, továbbá öt kisebb szerelőcsarnoka, mindegyik autótípus számára egy-egy. A karosszéria- és a motorüzem egy időben csak egy-egy autótípushoz gyárt karosszériát, illetve motort, be kell osztani tehát, hogy az egy év időt hogyan használják fel az öt autótípus számára. A szerelőcsarnokok egymás munkáit nem tudják átvállalni.

Ha a karosszériaüzem csak egy autótípushoz gyártana karosszériát, akkor 10 000 db legyártásához rendre 1, 1.4, 2, 0.8 és 2.2 évre volna szükség.

Ha a motorüzem csak egy autótípushoz gyártana motort, akkor 15 000 db legyártásához rendre 1, 1.6, 3, 1 és 2.6 évre volna szüksége.

Az öt szerelőcsarnok kapacitása évente 7 500, 5 000, 1 000, 9 000, ill. 3000 autó összeszerelése.

Végül megadjuk, hogy az egyes autótípusok egy-egy darabjának eladása révén a gyár tiszta haszna ezer forintos egységben kifejezve 35, 45, 50, 30, illetve 40 egység.

Nem törődve azzal, hogy az autók száma csak egész szám lehet, feladatunkat az alábbi módon fogalmazzuk meg:

$$\begin{aligned} \text{maximalizálandó} \quad & (35x_1 + 45x_2 + 50x_3 + 30x_4 + 40x_5), \\ \text{feltéve, hogy} \quad & x_1 + 1.4x_2 + 2x_3 + 0.8x_4 + 2.2x_5 \leq 10000 \\ & x_1 + 1.6x_2 + 3x_3 + x_4 + 2.6x_5 \leq 15000 \end{aligned}$$

$$\begin{aligned} x_1 & \leq 7000 \\ x_2 & \leq 5000 \\ x_3 & \leq 1000 \\ x_4 & \leq 9000 \\ x_5 & \leq 3000 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0. \end{aligned}$$

³ A programozási nyelvekben elterjedt szokás szerint a számok felírásában tizedespontot és nem tizedesvesszőt használunk. (A szerk.)

Az első feltétel a karosszériaüzemmel, a második a motorüzemmel, a többiek az összeszerelő üzemekkel kapcsolatosak. A feladat részletes megoldásával nem foglalkozunk. Eredményként a következő mennyiségeket kapjuk:

$$x_1 = 2800, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 9000, \quad x_5 = 0.$$

Svájci családi óragyár. Jakab testvérével, Jánossal és annak fiával, Dániellel családi óragyárat létesít. Négyfajta órát gyárthatnak, ám nem tudják még, hogy melyikből mennyi legyen, mondjuk, egy hét termelése. Az alábbi táblázatban összefoglaltuk, hogy egy-egy óra gyártásához hány munkaórát használnak fel, továbbá mennyi a nyereségük egy-egy órán, végül pedig, hogy heti hány munkaórában termelnek. (Például az 1. fajta óra elkészítéséhez Jakabnak 2 órát, Jánosnak 1 órát és Dánielnek 3 órát kell dolgoznia. *A szerk.*)

1.	2.	3.	4.	Heti munka- órák száma	
fajta					
2	2	1	3	50	Jakab
1	2	1	4	60	János
3	1	3	1	40	Dániel
6	4	5	8		Nyereség

Kérdés, hogy hány darabot gyártsanak az egyes órafajtákból hetenként ahhoz, hogy a haszon a lehető legnagyobb legyen, ha még ismeretes az is, miszerint az első és a második fajtából együtt heti 6 darabnál többet nem tudnak eladni. Jelölje x_1, x_2, x_3, x_4 az egyes fajtákból egy hét alatt gyártandó mennyiségeket. A feladat az alábbi módon fogalmazható meg:

$$\begin{aligned} &\text{maximalizálandó} && (6x_1 + 4x_2 + 5x_3 + 8x_4), \\ &\text{feltéve, hogy} && 2x_1 + 2x_2 + x_3 + 3x_4 \leq 50, \\ &&& x_1 + 2x_2 + x_3 + 4x_4 \leq 60, \\ &&& 3x_1 + x_2 + 3x_3 + x_4 \leq 40, \\ &&& x_1 + x_2 \leq 6, \\ &&& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{aligned}$$

A feladat megoldásaként az alábbi számok adódnak:

$$x_1 = 30/11, \quad x_2 = 0, \quad x_3 = 70/11, \quad x_4 = 140/11.$$

Ezt úgy értelmezhetjük, hogy 11 hét alatt az első fajtából harmincat, a másodikból semmit, a harmadikból hetvenet és a negyedikből száznegyvenet kell gyártaniuk.

Optimális szénelosztás. Négy bányából öt erőműhöz szállítunk szenet. Egy alkalommal, amikor éppen egy elosztási tervet készítünk, az erőművek szénigényei a következők (10 tonnás egységben kifejezve):

Erőmű	1	2	3	4	5
Szénigény	10	20	30	30	20

Egyszerűség kedvéért tegyük fel, hogy a bányákban összesen ugyanannyi szén van, mint az igények összege, és pedig a következő megoszlásban (10 tonnás egységben kifejezve):

Bánya	1	2	3	4
Szénmennyiség	25	26	27	32

Feltesszük továbbá, hogy a szén szállítási költsége mindenütt arányos a szállított mennyiséggel. Egy tonna szén szállítási költségeit – valamilyen egységben kifejezve – az alábbi táblázat tartalmazza:

Bánya \ Erőmű	1	2	3	4	5
1	9	13	8	5	13
2	8	22	7	2	8
3	4	7	11	17	4
4	15	18	5	20	19

Kérdés, hogy honnan hová mennyi szemet szállítsunk, hogy a szállítási költségek összege a lehető legkisebb legyen?

Ha x_{ik} jelöli az i -edik bányából a k -adik erőműbe szállítandó mennyiséget (célszerű most kettős indexet használni), akkor a feladat a következő módon fogalmazható meg:

$$\begin{aligned} \text{minimalizálandó} \quad & (9x_{11} + 13x_{12} + 8x_{13} + 5x_{14} + 13x_{15} + \\ & + 8x_{21} + 22x_{22} + 7x_{23} + 2x_{24} + 8x_{25} + \\ & + 4x_{31} + 7x_{32} + 11x_{33} + 17x_{34} + 4x_{35} + \\ & + 15x_{41} + 18x_{42} + 5x_{43} + 20x_{44} + 19x_{45}), \end{aligned}$$

$$\begin{aligned} \text{feltéve, hogy} \quad & x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 25, \\ & x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 26, \\ & x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 27, \\ & x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 32, \\ & x_{11} + x_{21} + x_{31} + x_{41} = 10, \\ & x_{12} + x_{22} + x_{32} + x_{42} = 20, \\ & x_{13} + x_{23} + x_{33} + x_{43} = 30, \\ & x_{14} + x_{24} + x_{34} + x_{44} = 30, \\ & x_{15} + x_{25} + x_{35} + x_{45} = 20, \end{aligned}$$

$$x_{ik} \geq 0; \quad i = 1, 2, 3, 4; \quad k = 1, 2, 3, 4, 5.$$

A fenti megoldásban csupán a zérótól különböző x_{ik} mennyiségek megadására szorítkozunk. Ezek a következők:

$$\begin{aligned} x_{11} = 10, \quad x_{12} = 11, \quad x_{14} = 4, \quad x_{24} = 26, \\ x_{32} = 7, \quad x_{35} = 20, \quad x_{42} = 2, \quad x_{43} = 20. \end{aligned}$$

Menütervezési probléma. Fogyókúrás gyümölcskoktélt állítunk össze úgy, hogy előírjuk négy vitaminfajtából a koktélnél vitamintartalmának alsó szintjét, továbbá minimalizáljuk a szénhidrát össz mennyiségét. A gyümölcsök 100 grammjában levő vitamin- és szénhidrátmennyiségeket, továbbá az említett alsó szinteket az alábbi táblázat tartalmazza:

Vitamin \ Gyümölcs	Alma	Körte	Őszi- barack	Na- rancs	Banán	Füge	Szőlő	Görög- dinnye	Alsó szint
B_1	50	30	20	70	160	120	50	100	600 mikrogramm
B_2	50	30	20	40	80	80	50	0	750 mikrogramm
Nikotinsav	0.5	0.3	0.9	0.2	0.5	0	0.4	0	8 milligramm
C	5	5	7	40	10	5	5	200	35 milligramm
Szénhidrát	7	12	9	7	23	60	18	8	gramm

Jelölje x_1, \dots, x_8 az egyes gyümölcsfajtákból felhasználandó mennyiségeket 100 grammos egységekben kifejezve. Feladatunk a következő:

$$\begin{aligned} \text{minimalizálandó} \quad & (7x_1 + 12x_2 + 9x_3 + 7x_4 + 23x_5 + 60x_6 + 18x_7 + 8x_8), \\ \text{feltéve, hogy} \quad & 50x_1 + 30x_2 + 20x_3 + 70x_4 + 160x_5 + 120x_6 + 50x_7 + 100x_8 \geq 600, \\ & 50x_1 + 30x_2 + 20x_3 + 40x_4 + 80x_5 + 80x_6 + 50x_7 \geq 750, \\ & 0.5x_1 + 0.3x_2 + 0.9x_3 + 0.2x_4 + 0.5x_5 + 0.4x_7 \geq 8, \\ & 5x_1 + 5x_2 + 7x_3 + 40x_4 + 10x_5 + 5x_6 + 5x_7 + 200x_8 \geq 35, \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0, \quad x_6 \geq 0, \quad x_7 \geq 0, \quad x_8 \geq 0. \end{aligned}$$

A feladat megoldásaként a következő értékek adódnak:

$$\begin{aligned} x_1 = 14.71, \quad x_3 = 0.71, \\ x_2 = x_4 = x_5 = x_6 = x_7 = x_8 = 0. \end{aligned}$$

A lineáris programozási feladat megoldása

A lineáris programozási feladat megoldására mind a mai napig a Dantzig által 1951-ben közölt módszert alkalmazzák leginkább. Ennek egy egyszerű variánsát fogjuk ismertetni.

A feladat feltételeinek eleget tevő x_1, \dots, x_n rendezett szám n -eseket *megengedett megoldásoknak* nevezzük. A továbbiakban feltesszük, hogy feladatunk olyan alakú, hogy csupán az $x_1 \geq 0, \dots, x_n \geq 0$ feltételekben áll egyenlőség, az egyéb feltételekben pedig egyenlőség, közelebből: lineáris egyenlőség. A szendvicses feladat numerikus megoldása előtt majd példát mutatunk arra, hogy mi módon lehet a feladatot a most említett, ún. kanonikus (egységesített) alakra hozni, ha az eredetileg nem volt ilyen. Még azt is feltesszük, hogy az egyenlőséges feltételek ún. *kifejezett alakúak* bizonyos „változókra” nézve, amin azt értjük, hogy mindegyik egyenlőséges feltételben van olyan

változó, mely a többiekben nem szerepel (másképpen kifejezve: a többiekben zéró együtthatóval szerepel). Ilyen pl. az alábbi feltételrendszer:

$$(1) \quad \begin{array}{rcl} 10 - x_1 & & - 3x_4 + 2x_5 = 0, \\ 15 & - x_2 & - 4x_4 + x_5 = 0, \\ 16 & & - x_3 + 2x_4 - 6x_5 = 0, \end{array}$$

mely kifejezett az x_1, x_2, x_3 változókra nézve. A kifejezett változókat más néven *bázisváltozóknak* is nevezzük. Még azt is feltesszük, hogy ha a bázisváltozóktól különböző többi változó helyébe zérót helyettesítünk, akkor az egyenlőségekből a bázisváltozók számára automatikusan adódó értékek nemnegatívak. A fenti példában pontosan ez a helyzet, ugyanis x_4 és x_5 helyébe zérót téve, az $x_1 = 10, x_2 = 15, x_3 = 16$ adódik.

Ha a kanonikus alakú feladat eredetileg nem ilyen, akkor is ilyen alakúra hozható, feltéve, hogy egyáltalán van megengedett megoldás, amint azt a későbbiekben megmutatjuk.

Most egy olyan feladattal foglalkozunk, amelyben maximumot keresünk. A minimum-feladat erre visszavezethető, ha a minimalizálandó kifejezést (-1) -gyel megszorozzuk. A maximalizálandó kifejezést *célfüggvénynek*, a legnagyobb célfüggvényértéket adó megengedett megoldást – ha ilyen egyáltalán van – *optimális megoldásnak* nevezzük. Általában ha egy halmaz minden eleméhez egyértelműen hozzárendelünk egy számot, akkor ezt a hozzárendelést *függvénynek* nevezzük. Most a feltételek által meghatározott x_1, \dots, x_n rendezett szám n -esek mindegyikéhez rendelünk hozzá egy-egy számot, ily módon alkotjuk meg a célfüggvényt. Pl. az (1) és az alábbi

$$(2) \quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$$

nemnegativitási feltételeknek eleget tevő x_1, x_2, x_3, x_4, x_5 rendezett számötösökön értelmezhetjük az alábbi célfüggvényt:

$$(3) \quad z = -6x_1 - x_2 + 5x_3 - 25x_4 + 37x_5.$$

Az (1) egyenlőségekből az x_1, x_2, x_3 változók számára adódó kifejezéseket (3) jobb oldalán a megfelelő helyekre behelyettesítve a célfüggvényre olyan alak adódik, amelyben x_1, x_2, x_3 már nem (vagy ha úgy tetszik, zéró együtthatóval) szerepel, mégpedig:

$$(4) \quad z = 5 + 7x_4 - 6x_5.$$

Azt a feladatot, amelyben a (4) célfüggvény maximalizálandó az (1) és a (2) feltételek mellett, az alábbi alakban írjuk fel:

$$(5) \quad \begin{array}{l} \text{maximalizálandó} \\ \text{feltéve, hogy} \end{array} \quad \begin{array}{rcl} 5 - (0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 - 7x_4 + 6x_5) = z, \\ 10 - (x_1 & & + 3x_4 - 2x_5) = 0, \\ 15 - (& x_2 & + 4x_4 - x_5) = 0, \\ 16 - (& & x_3 - 2x_4 + 6x_5) = 0, \\ x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0. \end{array}$$

A feladatot a konstansok és az együtthatók alábbi táblázatából egyértelműen rekonstruálhatjuk:

	x_1	x_2	x_3	x_4	x_5	
z	5	0	0	0	-7	6
x_1	10	1			3	-2
x_2	15		1		4	-1
x_3	16			1	-2	6

1. tábla

A táblában a zérók kiírása vagy elhagyása tetszés szerint történhet. A tábla bal oldalán a kifejezett változókat és a célfüggvény függő változóját írjuk fel, a tábla tetején pedig felsoroljuk valamennyi változót. Most két tételt említünk meg, amelyeket a lineáris programozási feladat megoldásakor használunk fel.

1. tétel. Ha a tábla felső sorában álló számok – az elsőt nem számítva – mind nemnegatívak, akkor a kifejezett változókat a mellettük álló konstansokkal, a többi változót 0-val egyenlővé téve, a feladat optimális megoldását nyerjük.

A tétel igaz voltát az (5) feladat egy módosítottján szemlélhetjük. A módosítás abban áll, hogy az x_4 együtthatóját a célfüggvényben megváltoztatjuk, -7 helyett 7 -et írunk a zárójelen belül, miáltal az új célfüggvény a következő lesz:

$$z = 5 - 7x_4 - 6x_5.$$

Mínt hogy $x_4 \geq 0$, $x_5 \geq 0$, azonnal adódik, hogy $z \leq 5$; és a $z = 5$ célfüggvényértéket valóban megkapjuk, ha $x_4 = x_5 = 0$, továbbá $x_1 = 10$, $x_2 = 15$, $x_3 = 16$.

2. tétel. Ha a tábla felső sorában – az első számot nem számítva – van olyan negatív szám, hogy az alatta álló számok mind nem pozitív, akkor a célfüggvény nem korlátos felülről, azaz akármilyen nagy számnál nagyobb értéket felvesz alkalmas megengedett megoldás esetén.

Ezt a tételt is az (5) feladat egy módosítottján szemléltetjük. A módosítás most abban áll, hogy az első és a második egyenlőséges feltételben a zárójelen belül x_4 együtthatóját 3 helyett (-3) -ra, illetve 4 helyett (-4) -re változtatjuk. Az így nyert feladatban választunk egy tetszőleges x_1, x_2, x_3, x_4, x_5 megengedett megoldást; x_4 -et növelve, az egyenlőségek érvényben tarthatók az x_1, x_2, x_3 bázisváltozók értékeinek alkalmas megválasztásával, miközben a nem bázisváltozók értékét nem változtatjuk (most az x_5 tartozik ebbe a kategóriába). Eközben a célfüggvény értéke minden határon túl nő, vagyis tényleg nem korlátos felülről.

A fenti két tételben két különböző típusú tábláról van szó. Van egy harmadik típus is, melyben a felső sorban az első számot kihagyva – van negatív szám és az alatta álló számok között van legalább egy pozitív. Mindegyik tábla a három kategória valamelyikébe tartozik. Az (5) feladathoz tartozó 1. tábla a harmadik csoportba tartozik.

Ekkor eljárásunk a következő: a bázisváltozók valamelyikét megszüntetjük mint bázisváltozót, helyette egy olyan változó, mely eddig nem volt bázisváltozó, azzá válik. Az új bázisváltozó céljára minden olyan változó megfelel, melynek a célfüggvényben a kerek zárójelen belüli együtthatója negatív. Ezek közül tetszőlegesen kiválasztunk egyet *bejövő* változó gyanánt; ám a kimenő *változó* meghatározása már nem tetszőleges. Ezt úgy kell megválasztanunk, hogy a célfüggvényben szereplő konstans lehetőleg növekedjék, de legalábbis ne csökkenjen, a feltételek konstansai pedig maradjanak nemnegatív az új bázisváltozók esetében. Mindkét célt elérjük, ha vesszük azokat az egyenlőséges feltételeket, amelyekben a bejövő változó – zárójelen belüli – együtthatója pozitív, majd meghatározzuk mindegyik ilyen egyenlőségen belül a konstans tag és a bejövő változó együtthatójának a hányadosát és amelyik egyenlőség esetében ez a hányados a legkisebb, annak a bázisváltozója lesz a kimenő változó. Ezek után az előbbi módon kiválasztott egyenlőséges feltételt a bejövő változó együtthatójával végigosztjuk, majd a célfüggvényből és a többi egyenlőséges feltételből ugyanezt a változót kiküszöböljük az új bázisváltozóhoz tartozó egyenlőséges feltétel alkalmas konstansszorosainak hozzáadásával. Megjegyzendő, hogy az új feltételek a régiekkel egyenértékűek, a célfüggvény pedig semmit nem változik értékében, csak alakjában, hiszen csupán zérót adtunk hozzá.

Bár a fentiek alapján nagyjából világos, hogy hogyan nyerjük az új táblát a régitől, a teljesség kedvéért ezt a szabályt röviden összefoglaljuk. Ha az x_k változó bejön a bázisba, x_j pedig kimegy a bázisból, akkor:

az új táblában x_k sorát úgy nyerjük, hogy a régi táblában az x_j -nek megfelelő sort végigosztjuk azzal az elemével, amelyik az x_k oszlopában áll;

az új tábla minden egyéb sorát úgy származtatjuk, hogy a neki megfelelő, régi táblabeli sor x_k oszlopában álló elemével az új tábla x_k -nak megfelelő sorát elemenként végigszorozzuk, majd az így kapott sort elemenként levonjuk a régi táblabeli sorból.

Alkalmazzuk a fent leírt lépést az (5) feladatra. Csak egy negatív szám van a tábla felső sorában az utolsó öt pozícióban, ez x_4 -hez tartozik, tehát x_4 bejön. Mínt hogy

$$\min(10/3, 15/4) = 10/3$$

és a minimumot az első egyenlőséges feltételben kapjuk, ezért x_1 a kimenő bázisváltozó. Az új feladatot szükségtelen kiírni, elég az új tábla megadására szorítkozni, ez pedig a következő:

	x_1	x_2	x_3	x_4	x_5	
z	85/3	7/3	0	0	4/3	
x_4	10/3	1/3	0	0	1	-2/3
x_2	5/3	-4/3	1	0	0	5/3
x_3	68/3	2/3	0	1	0	14/3

2. tábla

Mínt hogy a felső sor utolsó öt pozíciójában minden szám nemnegatív, optimális megoldáshoz jutottunk, amely a következő:

$$x_1 = x_5 = 0, \quad x_4 = 10/3, \quad x_2 = 5/3, \quad x_3 = 68/3.$$

Ezt a módszert nevezik Dantzig módszerének.

Ha Dantzig módszerének végrehajtása során a tábla bal felső sarkában álló szám állandóan növekszik, akkor soha nem térhet vissza a bázisváltozók egy olyan összessége, mely korábban már szerepelt. A kifejezett változók ugyanis a teljes táblát és így a bal felső sarokban álló elemet is egyértelműen meghatározzák. Mínt hogy pedig csak véges sok különböző módon lehet a kifejezett változókat kiválasztani, az eljárásnak véges sok lépés után be kell fejeződnie, vagy

az 1., vagy a 2. tételnek megfelelő tábla elérésével. Ha a 2. tételnek megfelelő táblához jutottunk, akkor nyilván rosszul fogalmaztuk meg az eredeti gyakorlati feladatot, ahhoz kell tehát visszatérnünk.

A bejövő változó oszlopának és a kimenő változó sorának kereszteződésében álló elemet szokás *sarokelemnek* nevezni. Ennek a pozíciónak a megjelölése egyértelműen meghatározza, hogy milyen számítások révén jutunk el az új táblához. Ezért az általános szokásnak megfelelően a sarokelemet mindig bekarikázzuk.

Prékopa András

IRODALOM

- [1] *E. M. L. Beule*: Cycling in the dual Simplex Algorithm, *Naval Research Quarterly Logistics*, 2/1955/269–276.
- [2] *R. G. Bland*: New Finite Pivoting Rules for the Simplex Method, *Mathematics of Operations Research*, 2(1977)103–107.
- [3] *A. Charnes*: Optimality and Degeneracy in Linear Programming, *Econometrica*, 20 (1952) 160–170.
- [4] *A. Charnes, W. W. Cooper and A. Henderson*: An Introduction to Linear Programming, Wiley, New York, 1953.
- [5] *G. B. Dantzig*: Maximization of a Linear Function of Variables Subject to Linear Inequalities, a [10] cikkgyűjteményben 339–347.
- [6] *G. B. Dantzig*: Linear Programming and Extensions, Princeton University press, Princeton, New Jersey, 1963.
- [7] *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) 183–195.
- [8] *D. Gale*: The Theory of Linear Economic Models, McGraw-Hill, New York, 1960.
- [9] *L. V. Kantorovics*: *Matematicheszkije Metodi v Organizacii i Planirovanii Proizvodstva*, L. G. U. 1939.
- [10] *T. C. Koopmans (szerkesztő)*: Activity Analysis of Production and Allocation, Wiley, New York, 1951.
- [11] *Prékopa A.*: Lineáris Programozás I., Bolyai János Matematikai Társulat, 1968.
- [12] *Prékopa A.*: A lineáris programozás egy kombinatorikai jellegű tárgyalásáról, *Matematikai Lapok*, 22 (1971) 7–24.
- [13] *S. Vajda*: The Theory of Games and Linear Programming, Wiley, New York, 1956.