Írta:
MAROS ISTVÁN

OPERÁCIÓKUTATÁS INFORMATIKUSOKNAK

Egyetemi tananyag

TYPOTEX

2011

LEKTORÁLTA: Dr. Illés Tibor, Budapesti Műszaki és Gazdaságtudományi Egyetem Matematika Intézet Differenciálegyenletek Tanszék

TÁMOGATÁS:
Készült a TÁMOP-4.1.2-08/1/A-2009-0008 számú, „Tananyagfejlesztés mérnök informatikus, programtervező informatikus és gazdaságinformatikus képzésekhez” című projekt keretében.

ISBN 978-963-279-514-0

KÉSZÜLT: a Typotex Kiadó gondozásában
FELELŐS VEZETŐ: Votisky Zsuzsa
AZ ELEKTRONIKUS KIADÁST ELŐKÉSZÍTETTE: Juhász Lehel

KULCSSZAVAK:
operációkutatás, lineáris programozás, szimplex módszer, primál algoritmus, duál algoritmus, egészértékű programozás, hálózati optimalizálás, nemlineáris programozás.

ÖSSZEFOGLALÁS:
Az informatikai alkalmazások egy jelentős hányada a döntéstámogatás területére esik. Ez pedig jártasságot igényel a döntéstudományban, amit gyakran az operációkutatással azonosítanak. Jelen jegyzet célja, hogy az informatikus egyetemi hallgatókat segítse ezen jártasság megszerzésében az operációkutatás és az optimalizálás legfontosabb modelljeinek és módszereinek a megismertetésén keresztül. Ezen belül a lineáris, nemlineáris és egészértékű programozás, valamint a hálózati optimalizálás tárgyalására kerül sor olyan terjedelemben, ami egy féléves kurzus keretibe belefér.

Tartalomjegyzék

Jelölések

k, t	normál kisbetű: skaláris mennyiség
$i, \ldots, n$	(általában) egészértékű mennyiségek
$\mathbf{v}$	vastag kisbetű: (oszlop) vektor
v_i	$\mathbf{v}$ vektor i-edik komponense
$\mathbf{v}^T$	$\mathbf{v}$ vektor transzponáltja: sor vektor
$\mathbf{A}$	vastag nagybetű: mátrix
$\mathbf{a}^i$	$\mathbf{A}$ mátrix i-edik sora (sor vektor!)
$\mathbf{a}_j$	$\mathbf{A}$ mátrix j-edik oszlopa
a^i_j	$\mathbf{A}$ mátrix i-edik sorának j-edik eleme
m	(általában) egy mátrix sorainak a száma, vagy vektor dimenziója
n	(általában) egy mátrix oszlopainak a száma
$\mathbb{R}^m$	m dimenziós euklideszi tér
$\mathbb{R}^{m\times n}$	$m \times n$ dimenziós euklideszi tér
$\mathcal{F}$	kalligrafikus nagybetű: halmaz

Előszó

Ahogy az információs társadalom kibontakozik, egyre magasabb követelmények fogalmazódnak meg az informatikusokkal szemben. Az informatikai alkalmazások egyik iránya a döntéstámogatás területére esik. Ez pedig jártasságot igényel a döntéstudományban, amit gyakran az operációkutatással azonosítanak. Jelen jegyzet célja, hogy az informatikus egyetemi hallgatókat segítse ezen jártasság megszerzésében.

Mint látni fogjuk, az operációkutatás matematikai apparátussal dolgozik. Ennek megfelelően a jegyzet tanulmányozása feltételez bizonyos matematikai előismereteket. Ez elsősorban a lineáris algebrát jelenti, de szükség van a matematikai analízis (kalkulus) ismeretére is.

A jegyzet nem matematikus hallgatók részére készült, így nem törekszik arra, hogy minden matematikai levezetés tétel-bizonyítás formában kerüljön bemutatásra (bár lesz ilyen is). Ugyanakkor, a szereplő levezetések megfelelnek a szigorú matematikai követelményeknek. A részletezettség tekintetében az volt a szempont, hogy a legfontosabb algoritmusok elve és működése teljesen világos legyen. Így például a szimplex módszer tárgyalása kitér a részletekre is.

A jegyzet alapvető célja az, hogy a hallgató világos képet kapjon az operációkutatásban szereplő legfontosabb modellekről, módszerekről és képes legyen verbálisan megfogalmazott operációkutatás problémák matematikai alakba öntésére. Miután egy probléma többféleképpen is modellezhető, a hallgató olyan formát tud majd választani, aminek a megoldására van használható algoritmus. Ezért fontos a modellek és a megoldó algoritmusok ismerete.

Az operációkutatási algoritmusok igen számításigényesek, így szinte minden feladatot számítógépen kell megoldani. A számítógépes programok az elméleti algoritmusok implementációi. Az implementációs tevékenység külön tudománynak, bizonyos értelemben művészetnek minősül. A jegyzet nem tér ki ennek részleteire.

Az operációkutatás igen széles területet foglal magába. Ezek közül kellett kiválasztani azokat, amelyek a gyakorlatban a legfontosabbak, illetve leggyakrabban használatosak. A jegyzet a szerzőnek több évtizedes kutatói és oktatói tevékenységének tapasztalata és ismeretanyaga alapján készült. Valószínű, hogy a jövőben további szegmensekkel fog bővülni az itt tárgyalt anyag. Ilyen vonatkozásban a jegyzet használatával kapcsolatos tapasztalatok visszacsatolása a szerzőhöz nagy segítséget jelent.

Köszönet illeti a TÁMOP 4.1.2 A) tananyagfejlesztési pályázat anyagi támogatását, amely lehetővé tette a jegyzet elkészítését.

Veszprém, 2011 március

Maros István
egyetemi tanár

1. fejezet

Operációkutatás és optimalizálás

Az operációkutatásnak nincs általánosan elfogadott pontos definíciója. Leginkább az mondható el róla, hogy a gazdasági szervezetek irányításának tudományos alapokra helyezése abból a célból, hogy az irányítók a lehető legjobb döntéseket tudják meghozni. Ilyen értelemben az **operációkutatás** nem más mint **döntéstudomány**, pontosabban a döntéstámogatás tudománya, mert a végső döntést mindig egy felelős személy vagy testület hozza.

Az operációkutatás egy új tudomány, amely a második világháború ideje alatt született. Az angol katonai vezetés összehívott egy, főleg matematikusokból álló kutatócsoportot azzal a feladattal, hogy segítsenek bizonyos katonai műveletek (operációk) sikerének növelésében. Itt elsősorban a radar rendszerek hatékony használata (a radar akkoriban új eszköz volt), az utánpótlást szállító hajó konvojok biztonsága, a tengeralattjárók elleni harc hatékonyságának fokozása és a bombázó rajok méretének optimalizálása volt a fő feladat. Az angol terminológia (brit angol: operational research, amerikai angol: operations research) a katonai műveletek (operations) kutatására (research) utal. A magyar fordítás féloldalasra sikerült: az „operations”-ból *operáció* lett, míg a „research” pontos fordítással szerepel: *kutatás*. Mára az operációkutatás elnevezés olyan széles körben terjedt el, hogy annak megváltoztatása nem valószínű.

Az operációkutatásról az is elmondható, hogy segítségével olyan kompromisszumot lehet megtalálni egy szervezet (pl. vállalat) különböző részlegei között, ami a szervezet egésze szempontjából a legelőnyösebb. Megtalálni a „lehető legjobb”-at pedig nem más, mint felkutatni egy optimális megoldást.

Történetileg érdekes tény, hogy a számítógépek és az operációkutatás fejlődése milyen kölcsönösen megtermékenyítően hatottak egymásra. Közismert, hogy az optimalizálási feladatok megoldása nagyon számításigényes. Számítógép használata nélkül legfeljebb csak kisméretű tankönyvi feladatokat lehet megoldani. Az 1940-es évek végén az akkori néhány számítógépen az optimalizáló programok fogyasztották a gépidő nagy részét. Az operációkutatók a kezdeti sikereken felbuzdulva egyre nagyobb méretű feladatokat akartak megoldani. Erre azonban a számítógépek már nem voltak képesek a kicsi memóriaméret és az alacsony számítási sebesség miatt. Ez viszont ösztönzőleg hatott a hardware fejlesztésére. Ez a folyamat pozitív spirálként egészen az 1960-as évekig működött, amint az G.B. Dantzig egy későbbi tanulmányában kimutatta.

Ha az optimalizálásról úgy beszélünk, hogy „a dolgokat a lehető legjobban csinálni”, akkor ebben két lényeges szempontot is említünk. A „legjobb” csak akkor azonosítható, ha mérni tudjuk az események kimenetelének jóságát. A „lehetőleg” pedig arra utal, hogy nem lehet akármit csinálni, mert vannak korlátok, amiket nem lehet átlépni, vagyis bizonyos dolgokat nem lehet csinálni, bármennyire is szeretnénk.

Felmerül a kérdés, hogy könnyű vagy nehéz meghatározni egy optimális eseménysort, vagyis nehéz diszciplina-e az optimalizálás?

A válaszhoz jó tudni, hogy az optimalizálás ugyan egy önálló tudomány, de erősen támaszkodik más tudományokra. Ezek közül a legfontosabbak:

- matematika,
- számítástudomány,
- numerikus algebra,
- modellezés és logika.

Az operációkutatást és az optimalizálást a gyakorlati igények hozták létre és az elért eredmények a gyakorlatban nyernek alkalmazást. Az alkalmazások nagyon széles kört ölelnek fel.

2. fejezet

Alapfogalmak áttekintése

Ebben a fejezetben megfogalmazzuk az optimalizálás matematikai modelljét és tárgyaljuk a főbb modell típusokat.

2.1. Feladat megfogalmazása

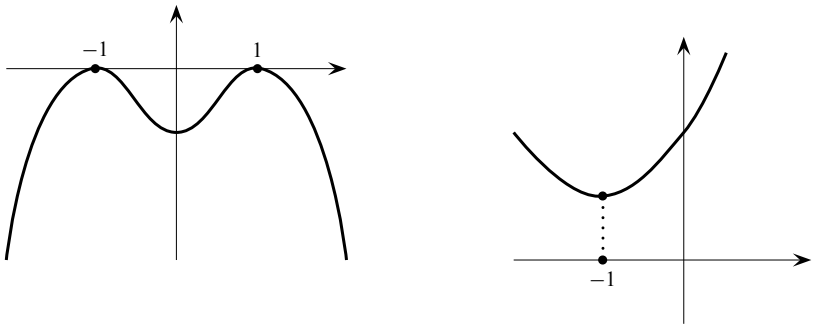
Az optimalizálás a hagyományos matematikai szélsőértékszámítás egy általánosítása. Egy (esetleg több) n-változós függvény szélsőértékét (maximumát vagy minimumát) keressük az n dimenziós térben, $\mathbb{R}^n$-ben, illetve annak egy részhalmazán, $\mathcal{F} \subseteq \mathbb{R}^n$. A függvényt **célfüggvény**nek nevezzük. A szélsőértékre mint **optimum**ra szokás hivatkozni. A függvény változói (ismeretlenjei) a **döntési** (más néven **tervezési**) **változók**.

Ha a teljes n dimenziós térben keressük as optimumot, akkor **feltétel nélküli optimalizálás**ról beszélünk. Amennyiben annak egy $\mathcal{F}$ valódi részhalmazán, akkor **feltételes optimalizálás**ról van szó.

Az $\mathcal{F}$ halmazt általában függvény-egyenlőségekkel és egyenlőtlenségekkel lehet megadni és rájuk **feltételi egyenlőség**, illetve **feltételi egyenlőtlenség** néven hivatkozunk. Bizonyos esetekben logikai kifejezések, illetve egyéb módok szükségesek $\mathcal{F}$ definiálására. Ha egyidejűleg több célfüggvényt próbálunk optimalizálni azonos feltételrendszer mellett, akkor **többszempontú optimalizálás**ról beszélünk.

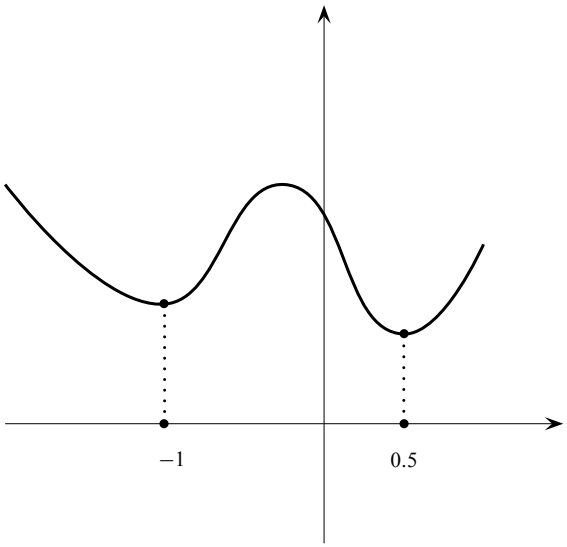
Az optimumot az $\mathcal{F}$ halmaz elemei közt keressük. $\mathcal{F}$ elemeit **megengedett megoldás**oknak nevezzük. Egy megengedett megoldás **optimális**, ha nincs nála jobb megoldás $\mathcal{F}$ elemei között.

Egy feladatnak lehet egy vagy több, esetleg végtelen sok optimális megoldása. Az is előfordulhat, hogy a célfüggvény minden határon túl javítható, vagyis a feladat megoldása nem korlátos. Amennyiben a feladat megfogalmazásában szereplő függvények közt van nem-lineáris, elképzelhető, hogy a függvénynek olyan szélsőértéke van, ami csak a környező pontokhoz képes a legjobb. Ezt **lokális optimum**nak nevezzük. Ha a szélsőérték az összes megengedett megoldás közt a legjobb, akkor a neve **globális optimum**.



(a) Több maximum. (b) Egyetlen minimum.

2.1. ábra. Szélsőértékek



2.2. ábra. Lokális illetve globális optimum (minimum)

2.2. Matematikai alak

Az egy célfüggvényes optimalizálási feladat általános alakja:

$$\min f(\mathbf{x}) \tag{2.1}$$

$$\mathbf{x} \in \mathcal{F} \subseteq \mathbb{R}^n \tag{2.2}$$

Minimalizálás helyett lehet maximalizálás is. A két feladat egymással ekvivalens, mert $\max f(\mathbf{x}) = -\min[-f(\mathbf{x})]$, vagyis a $-f(\mathbf{x})$ függvényt minimalizáljuk és a végén az eredmény előjelét megfordítjuk, ami által $f(\mathbf{x})$ maximumát kapjuk meg. Ha kiírjuk $\mathbf{x}$ komponenseit, akkor az $f(\mathbf{x}) = f(x_1, \ldots, x_n)$ alakot kapjuk.

A korábban megismert fogalmakat konkretizálva: (2.1)-et nevezzük célfüggvénynek, (2.2)-t pedig a megengedett megoldásokat definiáló feltételrendszernek.

Ha $\mathcal{F} = \mathbb{R}^n$, akkor feltétel nélküli optimalizálásról van szó. Ha $\mathcal{F} = \emptyset$ (üres halmaz), akkor a feladatnak nincs lehetséges megoldása, a feltételrendszer ellentmondásos.

2.3. Néhány nevezetes feladattípus

Az f függvény jellegétől és az $\mathcal{F}$ halmaz megadási módjától függően nagyon sokféle optimalizálási feladat adódik. Ebben a szakaszban bemutatunk néhányat a fontosabbak közül.

Eset 1 *Amennyiben $f(\mathbf{x})$ a változók lineáris függvénye, $f(\mathbf{x}) = c_1x_1 + \cdots + c_nx_n$ és a feltételek is lineáris függvények, akkor a* ***lineáris programozás****i modellt és feladatot kapjuk. Ez a modell alapvető fontosságú az optimalizálás elméletében és gyakorlatában, ezért a későbbiekben részletesen tárgyaljuk.*

Eset 2 *Ha $f(\mathbf{x})$ a változók kvadratikus függvénye, vagyis $f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}$, ahol $\mathbf{Q}$ egy szimmetrikus kvadratikus mátrix, és a feltételek lineáris függvények, akkor ezt* ***kvadratikus programozás****i feladatnak nevezzük.*

Belátható, hogy ha ez az $f(\mathbf{x})$ konvex függvény, akkor a minimuma egyben globális minimum az $\mathcal{F}$ tartományon.

Eset 3 *Tegyük fel ismét, hogy $f(\mathbf{x})$ a változók lineáris függvénye és a feltételek is lineáris függvények, úgy mint a lineáris programozási modell esetében. Most azonban kikötjük, hogy a döntési változók csak egész értéket vehetnek fel, vagyis az $\mathcal{F}$ halmaz azon pontokból áll, amelyek minden komponense egész értékű (rácspontok). Az ilyen feladatot* ***egészértékű lineáris programozás****i feladatnak nevezzük.*

Amennyiben a változók egy része folytonos, más része pedig egészértékű, akkor ez ***kevert egészértékű feladat****. Az angolban meghonosodott megnevezés Integer Linear Programming, illetve Mixed Integer Linear Programming alapján az ILP, illetve MILP rövidítéssel szokás hivatkozni ezekre a feladat és modell típusokra.*

Eset 4 *Ha $f(\mathbf{x})$ egy általános nemlineáris függvény és a feltételi függvények egy része egyenlőség, más része egyenlőtlenség, akkor a következő alakot kapjuk*

$$\min f(\mathbf{x}) \tag{2.3}$$
$$g_k(\mathbf{x}) \le 0, \qquad k = 1, \ldots, p, \tag{2.4}$$
$$h_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, m, \tag{2.5}$$

ahol $g_k(\mathbf{x})$ és $h_i(\mathbf{x})$ nemlineáris függvények. Ezt az alakot szokás általános ***nemlineáris programozási feladat****nak nevezni.*

Tömör alakban:

$$\min f(\mathbf{x}) \tag{2.6}$$
$$\mathbf{g}(\mathbf{x}) \le \mathbf{0}, \tag{2.7}$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{0}, \tag{2.8}$$

ahol $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_p(\mathbf{x}))$ és $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}))$. Amennyiben bizonyos változókra nem-negatívitási feltételt is meg kell adni, akkor azt a „g" függvények közt kell szerepeltetni. Például ha az $x_1 \ge 0$ feltételnek kell teljesülni, akkor azt $-x_1 \le 0$ formában adjuk meg.

3. fejezet

Lineáris optimalizálás korszerű módszerei

Az optimalizálási modellek között először a lineáris programozás (LP) tárgyalása következik. Az LP alapvető jelentőséggel bír az optimalizálás elméletében és gyakorlatában. Ennek több oka van. A legfontosabbak a következők.

1. LP közvetlenül alkalmazható minden olyan optimalizálási feladat esetén, amikor lineáris kapcsolat van a döntési változók között úgy a feltételekben, mint a célfüggvényben.

2. Az LP feladat megoldására hatékony és megbízható megoldó algoritmusok léteznek, amelyek képesek a gyakorlatban felmerülő igen nagy méretű feladatok megoldására is.

3. Az LP nagyon gyakran a háttérben dolgozó számítási gépezet sok egyéb (nemlineáris, egészértékű, sztochasztikus, stb.) optimalizálási feladat megoldó algoritmusában.

4. Számos, eredetileg nemlineáris feladat áttranszformálható ekvivalens LP feladattá és az LP feladat megoldásából vissza lehet számolni az eredeti feladat megoldását.

5. Sok nemlineáris feladatot jól lehet közelíteni alkalmasan választott LP modellel.

3.1. Egy termékválaszték optimalizálási példa

Az LP részletes tárgyalása előtt egy egyszerűsített optimális termékválasztási modellt mutatunk be. Ez, bár egyszerű, magában hordozza a valódi feladatok legfőbb ismérveit.

Tipikus ipari probléma egy optimális termékválaszték meghatározása, amely maximalizálja a hasznot egy adott tervezési időszakra.

Tegyük fel, hogy van egy jól menő gyár, amelyik 6 fő terméket tud előállítani, amiket Pr-1, …, Pr-6-tal jelölünk. Az előállításhoz felhasználnak élőmunkát, energiát és gépi megmunkálási időt. Ezek az erőforrások csak véges mennyiségben állnak rendelkezésre.

A gyár vezetése meg akarja határozni, hogy melyik termekből milyen mennyiséget termeljenek havonta, hogy a nyereség a maximális legyen, feltéve, hogy a piac bármekkora mennyiséget fel képes venni a termékekből.

A következő táblázat a gyártás technológiai feltételeit tartalmazza, vagyis hogy milyen erőforrásból mennyit kell felhasználni az egyes termékek egységnyi mennyiségének gyártásához, valamint az egyes erőforrásokból a havonta rendelkezésre álló mennyiséget (korlát).

	Pr-1	Pr-2	Pr-3	Pr-4	Pr-5	Pr-6	Korlát
Nyereség	5	6	7	5	6	7	
Energia	1	2	1	4	1	2	1080
Gépidő (óra)	2	3	2	1	1	3	1050
Élőmunka (óra)	2	1	3	1	3	2	1050

A döntési változók a termékek ismeretlen mennyiségei, melyeket $x_1, \ldots, x_6$-tal jelölünk.

A nyereséget a következő kifejezés írja le:

$$5x_1 + 6x_2 + 7x_3 + 5x_4 + 6x_5 + 7x_6$$

Ezt akarjuk maximalizálni figyelembe véve, hogy az erőforrások csak korlátolt mennyiségben állnak rendelkezésre:

$$\begin{aligned} 1x_1 + 2x_2 + 1x_3 + 4x_4 + 1x_5 + 2x_6 &\leq 1080 \\ 2x_1 + 3x_2 + 2x_3 + 1x_4 + 1x_5 + 3x_6 &\leq 1050 \\ 2x_1 + 1x_2 + 3x_3 + 1x_4 + 3x_5 + 2x_6 &\leq 1050 \end{aligned}$$

Miután negatív termelésnek nincs értelme, ki kell kötnünk, hogy a döntési változók csak nemnegatív értékeket vehetnek fel: $x_j \geq 0,\ j = 1, \ldots, 6$.

Ez egy egyszerű LP feladat, amit meg lehet oldani a később ismertetendő szimplex módszerrel. Optimális megoldásnak azt kapjuk, hogy $x_2 = 240,\ x_4 = 90,\ x_5 = 240$, és $x_1 = x_3 = x_6 = 0$, ami összesen 3330.00 egység nyereséget hoz.

Noha ez a feladat egyszerű, a megoldás értelmezése néhány érdekességre hívja fel a figyelmet.

- A legnyereségesebbnek látszó termékek (Pr-3 és Pr-6) nem szerepelnek az optimális megoldásban. Már itt is, de egy nagyobb feladat esetén még inkább igaz, hogy egy emberi döntéshozó nem könnyen jutna erre a következtetésre.
- Elegendő csak 3 féle terméket gyártani a maximális nyereség eléréséhez.
- Minden erőforrás teljes mértékben ki van használva (a feltételek egyenlőségre teljesülnek).

3.2. Az LP feladat általános alakja

Tegyük fel, hogy van n darab döntési változónk, $x_1, \ldots, x_n$ és m feltételünk. Először az LP feladat **standard alak**ját fogalmazzuk meg.

$$\begin{aligned} \min \quad & c_1x_1 + \cdots + c_nx_n \\ \text{feltéve, hogy} \quad & a_1^i x_1 + \cdots + a_n^i x_n = b_i \qquad i = 1, \ldots, m \\ & x_j \geq 0,\ j = 1, \ldots, n \end{aligned}$$

ahol c_j egy konstans, amely a j-edik változóhoz tartozik (és általában *költség együttható*nak nevezzük), a_j^i a j-edik változó együtthatója az i-edik feltételben, végül b_i az i-edik feltétel jobb-oldali konstansa.

Az első sor a **célfüggvény**, a második az **általános (közös) feltételek** együttese (amelyek mindegyike több változót tartalmaz), a harmadik pedig a változókra vonatkozó egyedi **nemnegativitási feltételek** kifejezése. A célfüggvény értékét gyakran z-vel jelölik, vagyis $z = c_1x_1 + \cdots + c_nx_n$.

A fenti feladat szummációs formája:

$$\begin{aligned} \min\ z = \quad & \sum_{j=1}^{n} c_j x_j \\ \text{feltéve, hogy} \quad & \sum_{j=1}^{n} a_j^i x_j = b_i, \quad i = 1, \ldots, m \\ & x_j \geq 0,\ j = 1, \ldots, n \end{aligned}$$

A közös feltételek együtthatóit mátrix alakban is fel lehet írni, ha az

$$\mathbf{A} = \begin{bmatrix} a_1^1 & \ldots & a_j^1 & \ldots & a_n^1 \\ \vdots & & \vdots & & \vdots \\ a_1^i & \ldots & a_j^i & \ldots & a_n^i \\ \vdots & & \vdots & & \vdots \\ a_1^m & \ldots & a_j^m & \ldots & a_n^m \end{bmatrix}$$

jelölést használjuk. Ezt a mátrixot tekinthetjük n oszlopvektor összességének:

$$\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_j \ldots \mathbf{a}_n],$$

ahol $\mathbf{a}_j$ jelöli a mátrix j-edik oszlopát (j alsó indexben), illetve m sorvektor összességének:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}^1 \\ \vdots \\ \mathbf{a}^i \\ \vdots \\ \mathbf{a}^m \end{bmatrix},$$

ahol $\mathbf{a}^i$ a mátrix i-edik sorát jelöli (i a felső indexben). Ez utóbbi segítségével az LP feltételek egyszerű skalár szorzatokkal is felírhatók:

$$\mathbf{a}^i\mathbf{x} = b_i, \quad i = 1, \ldots, m.$$

Meg kell jegyezni, hogy

- a célfüggvényt lehet maximalizálni is (amint arról korábban már szó volt a 2.2 szakaszban),
- egy akármilyen típusú feltételt egyenlőséggé lehet átalakítani (ennek részletei később következnek).

A gyakorlatban nemcsak egyenlőség feltételek és nem-negatív változók vannak. Éppen ezért célszerű az LP feladat általános alakját is felírni. Ez nem pusztán modellezési szempontból jelent majd könnyebbséget, hanem a megoldó algoritmus hatékonyságát is elő tudja segíteni.

A **célfüggvény** tartalmazhat egy c_0 additív konstanst, ami azonban az optimalizálás során nem játszik szerepet.

$$\min z = c_0 + c_1x_1 + \cdots + c_nx_n, \quad \text{vagy röviden: } c_0 + \sum_{j=1}^{n} c_jx_j \tag{3.1}$$

Az **általános feltételek** legalább két döntési változót tartalmaznak.

$$L_i \le \sum_{j=1}^{n} a_j^i x_j \le U_i, \quad i = 1, \ldots, m, \tag{3.2}$$

ahol L_i és U_i az i-edik feltétel által felvehető legkisebb, illetve legnagyobb érték (lehet $\mp\infty$ is).

A változók mindegyikének lehetnek **egyedi korlát**ai:

$$\ell_j \le x_j \le u_j \quad j = 1, \ldots, n, \tag{3.3}$$

ahol ℓ_j és u_j értelmezése az előbbihez hasonló, vagyis ℓ_j lehet $-\infty$ és u_j lehet $+\infty$.

A feltételek és egyedi korlátok struktúráját az alábbi formában lehet szemléltetni:

$$\begin{array}{c|ccccc|c}
 & x_1 & \ldots & x_j & \ldots & x_n & \\
L_1 & & \ldots & a_j^1 & \ldots & & U_1 \\
\vdots & & & \vdots & & & \vdots \\
L_i & a_1^i & \ldots & a_j^i & \ldots & a_n^i & U_i \\
\vdots & & & \vdots & & & \vdots \\
L_m & & \ldots & a_j^m & \ldots & & U_m \\
\hline
\ell_1 & 1 & & & & & u_1 \\
\vdots & & \ddots & & & & \vdots \\
\ell_j & & & 1 & & & u_j \\
\vdots & & & & \ddots & & \vdots \\
\ell_n & & & & & 1 & u_n
\end{array}$$

Attól függően, hogy a korlátok (akár közös, akár egyedi) közül melyek végesek illetve végtelenek, különböző változó illetve feltétel típusokat különböztetünk meg.

Változók egyedi korlátai

1. Ha $u_j = +\infty$ és ℓ_j véges, akkor (3.3) $\ell_j \le x_j$ alakra egyszerűsödik. Egy ilyen x_j-t *plusz típusú*, illetve *nemnegatív* változónak nevezünk és rá a PL címkével fogunk hivatkozni.

2. Ha $\ell_j = -\infty$ és u_j véges, akkor (3.3) az $x_j \le u_j$ alakot ölti. Egy ilyen x_j neve *mínusz típusú* változó és címkéje MI. Ezt könnyen PL típusúvá alakíthatjuk az $\bar{x}_j = -x_j$ és $\bar{\ell}_j = -u_j$ helyettesítéssel, ami által $\bar{\ell}_j \le \bar{x}_j$ adódik.

3. Ha ℓ_j és u_j mindegyike véges, akkor, egy (3.3)-t kielégítő x_j-t *korlátozott* változónak nevezünk és a BD címkével látjuk el (az angol „bounded" kifejezés alapján). Ennek egy speciális esete az, amikor $\ell_j = x_j = u_j$. Ezt *fix* vagy *rögzített* változónak nevezzük és a címkéje FX. Bár paradoxnak tűnik, hogy az ilyet miért hívjuk változónak, a későbbiekben látni fogjuk ennek az értelmét és hasznosságát.

4. Ha $\ell_j = -\infty$ és $u_j = +\infty$, akkor (3.3) az $-\infty \le x_j \le +\infty$ alakot ölti. Egy ilyen x_j neve *szabad* vagy (előjelben) *nem korlátozott* változó és a címkéje FR (angol: free). Egy szabad változót fel lehet írni két nemnegatív változó különbségeként is: $x_j = x_j^+ - x_j^-$, ahol $x_j^+, x_j^- \ge 0$. Ha ezt a helyettesítést használjuk, akkor az $\mathbf{A}$ mátrixot meg kell nagyobbítani egy oszloppal, mert $\mathbf{a}_j x_j = \mathbf{a}_j x_j^+ - \mathbf{a}_j x_j^-$ és az x_j^--nek megfelelő új oszlopvektor $-\mathbf{a}_j$.

Általános (közös) feltételek Az LP feladat megoldó algoritmusai megkövetelik, hogy a közös feltételek egyenlőség formában legyenek megadva. Az itt következőkben bemutatjuk, hogyan lehet egy tetszőleges általános feltételt ilyen alakra hozni. A különféle eseteket aszerint határozzuk meg, hogy az L_i és U_i korlátok közül melyek végesek, illetve végtelenek.

1. Ha $L_i = -\infty$ és U_i véges ($U_i < +\infty$), akkor: (3.2) a $\sum_{j=1}^{n} a_j^i x_j \le U_i$ alakot ölti, amit **kisebb vagy egyenlő**, vagyis „$\le$" típusú feltételnek nevezünk. Ezt könnyen átalakíthatjuk egyenlőséggé egy y_i nemnegatív változó bevezetésével. Egyidejűleg a $b_i = U_i$ helyettesítést is végrehajtjuk:

$$y_i + \sum_{j=1}^{n} a_j^i x_j = b_i, \quad \text{ahol} \quad y_i \ge 0.$$

   Az ilyen feltételek címkéje LE. Ez utóbbi az angol „Less than or Equal" (kisebb vagy egyenlő) kifejezésből származik.

2. Ha $U_i = +\infty$ és L_i véges ($L_i > -\infty$), akkor (3.2) a következő módon egyszerűsödik: $L_i \le \sum_{j=1}^{n} a_j^i x_j$, ami **nagyobb vagy egyenlő**, vagyis „$\ge$" típusú feltételként ismeretes és a GE címkével szokás rá hivatkozni. (Angol: „Greater than or Equal".) A két oldal -1-gyel történő beszorzása és a $b_i = -L_i$ jelölés bevezetése után az előző ($\le$) esetet kapjuk vissza: $\sum_{j=1}^{n} (-a_j^i) x_j \le b_i$, amit az ismert módon alakítunk át egyenlőséggé:

$$y_i + \sum_{j=1}^{n} (-a_j^i) x_j = b_i, \quad \text{ahol} \quad y_i \ge 0.$$

3. Ha mindkét korlát, L_i és U_i véges, akkor valójában két általános feltételünk van: $L_i \leq$ $\leq \sum_{j=1}^{n} a_j^i x_j$ és $\sum_{j=1}^{n} a_j^i x_j \leq U_i$. Ezek egyenértékűek egy általános és egy egyedi feltétellel: $y_i + \sum_{j=1}^{n} a_j^i x_j = U_i$, ahol $0 \leq y_i \leq (U_i - L_i)$. Ezt **intervallum**, illetve gyakran **range** típusú feltételnek nevezzük és RG címkével hivatkozunk rá. (Angol: „Range".) A $b_i = U_i$ és $r_i = (U_i - L_i)$ jelöléseket használva a feltételt átírhatjuk az

$$y_i + \sum_{j=1}^{n} a_j^i x_j = b_i, \quad 0 \leq y_i \leq r_i$$

alakra. Ez még akkor is igaz, ha $L_i = U_i$, amit **egyenlőség feltétel**nek nevezünk. Ekkor $y_i = 0$. Az ilyen feltétel címkéje EQ. (Angol: „Equality".)

4. Ha $L_i = -\infty$ és $U_i = +\infty$, akkor tetszőleges b_i-t választva formálisan azt írhatjuk, hogy

$$y_i + \sum_{j=1}^{n} a_j^i x_j = b_i, \quad \text{ahol } y_i \text{ szabad változó.}$$

Erre **nem-korlátozó feltétel**ként szokás hivatkozni, illetve rövidítve az NB címkével. (Angol: „Non-Binding".)

Következmény Minden általános feltétel a következő alakra hozható:

$$y_i + \sum_{j=1}^{n} a_j^i x_j = b_i, \ i = 1, \ldots, m \tag{3.4}$$

Az y_i $(i = 1, \ldots, m)$ változókat **logikai változók**nak, az x_j $(j = 1, \ldots, n)$ változókat pedig **strukturális változók**nak nevezzük.

Az előbb elmondottakat az alábbi **átalakítási szabályok**ban foglaljuk össze. Ezek eredményeként minden feltétel (3.4) alakú egyenlőség lesz.

1. Adjunk egy nemnegatív y_i logikai változót minden $\leq$ típusú feltétel baloldalához.

2. Szorozzuk meg minden $\geq$ feltételt -1-gyel (a jobboldali b_i elemet is beleértve), majd adjunk a baloldalhoz egy nemnegatív y_i logikai változót.

3. Legyen az intervallum típusú feltétel jobboldala az U_i felső korlát. Adjunk a baloldalhoz egy korlátos y_i logikai változót: $0 \leq y_i \leq U_i - L_i$.

4. Adjunk egy szabad (előjelben nem korlátozott) y_i logikai változót minden nem korlátozó (NB címkéjű) feltétel baloldalához. A b_i jobboldal értékeket tetszőlegesen lehet megválasztani.

5. Végül, az egyöntetűség érdekében, adjunk egy nulla értékű logikai változót minden egyenlőség típusú feltétel baloldalához.

1. Példa Alakítsuk át a következő LP feltételeket egyenlőségekké.

$$\begin{array}{rcrcrcrcr} & & 3x_1 & + & 2x_2 & + & 6x_3 & - & x_4 & \leq & 9 \\ & & x_1 & - & x_2 & & & + & x_4 & \geq & 3 \\ 2 & \leq & x_1 & + & 2x_2 & + & 3x_3 & - & 4x_4 & \leq & 15 \\ & & 2x_1 & - & x_2 & - & x_3 & & & \bowtie & 10 \\ & & x_1 & + & 3x_2 & - & x_3 & + & 6x_4 & = & 8 \end{array}$$

$$x_1 \geq 0,\ x_2 \leq 0,\ -4 \leq x_3 \leq -1,\ x_4 \text{ szabad}$$

A „$\bowtie$” szimbólum azt jelenti, hogy a baloldal és a jobboldal tetszőleges viszonyban állhat egymással: lehet kisebb, egyenlő, vagy nagyobb is (nem korlátozó feltétel). Ebben az esetben a hozzáadott logikai változó a jobboldal és a baloldal különbségét fogja mutatni.

1. feltétel: Adjunk egy nemnegatív y_1 változót a baloldalhoz (1. szabály).

$$y_1 + 3x_1 + 2x_2 + 6x_3 - x_4 = 9, \quad y_1 \geq 0$$

2. feltétel: Szorozzuk meg a feltétel mindkét oldalát -1-gyel és adjunk egy nemnegatív y_2 változót a baloldalhoz (2. szabály).

$$y_2 - x_1 + x_2 - x_4 = -3, \quad y_2 \geq 0$$

3. feltétel: Állítsuk a jobboldalt 15-re és adjunk egy y_3 korlátos logikai változót a baloldalhoz. (3. szabály).

$$y_3 + x_1 + 2x_2 + 3x_3 - 4x_4 = 15, \quad 0 \leq y_3 \leq 13$$

4. feltétel: Adjunk egy y_4 szabad változót a baloldalhoz (4. szabály).

$$y_4 + 2x_1 - x_2 - x_3 = 10, \quad y_4 \text{ szabad}$$

5. feltétel: Adjunk egy y_5 nulla értéken rögzített (fix) logikai változót a baloldalhoz (5. szabály).

$$y_5 + x_1 + 3x_2 - x_3 + 6x_4 = 8, \quad y_5 = 0$$

Végül a következő feltételrendszert kapjuk:

$$\begin{array}{rcrcrcrcrcr} y_1 & & & + & 3x_1 & + & 2x_2 & + & 6x_3 & - & x_4 & = & 9 \\ & y_2 & & - & x_1 & + & x_2 & & & - & x_4 & = & -3 \\ & & y_3 & + & x_1 & + & 2x_2 & + & 3x_3 & - & 4x_4 & = & 15 \\ & & y_4 & + & 2x_1 & - & x_2 & - & x_3 & & & = & 10 \\ & & y_5 & + & x_1 & + & 3x_2 & - & x_3 & + & 6x_4 & = & 8 \end{array}$$

$$y_1, y_2 \geq 0,\ 0 \leq y_3 \leq 13,\ y_4 \text{ szabad}, y_5 = 0$$

$$x_1 \geq 0,\ x_2 \leq 0,\ -4 \leq x_3 \leq -1,\ x_4 \text{ szabad}$$

Megjegyzés: nagyon fontos, hogy minden változó (strukturális és logikai) specifikálva legyen (utolsó két sor).

A (3.4) egyenleteket vektor formába átírva a következőt kapjuk

$$\sum_{i=1}^{m} \mathbf{e}_i y_i + \sum_{j=1}^{n} \mathbf{a}_j x_j = \mathbf{b}, \tag{3.5}$$

ahol $\mathbf{e}_i$ az i-edik egységvektor. Ugyanez mátrix formában:

$$\mathbf{I}\mathbf{y} + \mathbf{A}\mathbf{x} = \mathbf{b},$$

ahol $\mathbf{I}$ a megfelelő méretű (jelen esetben $m \times m$-es) egységmátrix.

MI típusú változók „megfordítása” Kényelmi okok miatt célszerű a nempozitív MI változókat nemnegatív PL tipusúakká transzformálni. Ez a következőképpen tehető meg.

Szorozzuk meg -1-gyel a $-\infty \le x_j \le u_j$ relációt, amiből

$$-u_j \le -x_j \le +\infty$$

adódik. Az $\bar{\ell}_j = -u_j$ és $\bar{x}_j = -x_j$ jelölést használva $\bar{\ell}_j \le \bar{x}_j \le +\infty$-t kapjuk, ahol $\bar{x}_j$ PL típusú változó. Miután $\bar{x}_j = -x_j$, a j-edik oszlopban minden a_j^i együtthatót meg kell szorozni -1-gyel, mert az oszlop most már $\bar{x}_j$-hez tartozik. Ha az LP feladatot megoldottuk, $\bar{x}_j$-t vissza kell transzformálni x_j-re az előjelének a megváltoztatásával.

Véges alsó korlátok A változók véges alsó korlátjait el lehet tolni a nullába. Ezáltal olyan alakot kapunk, amellyel a megoldó algoritmus leírása egyszerűbbé válik.

Ha ℓ_k véges és $\ell_k \le x_k$ akkor legyen $\bar{x}_k = x_k - \ell_k$. Ekkor $\bar{x}_k \ge 0$, és $x_k = \bar{x}_k + \ell_k$. Ha u_k is véges, akkor az szintén transzformálódik: $\bar{u}_k = u_k - \ell_k$. Behelyettesítve $x_k = \bar{x}_k + \ell_k$-t (3.5)-ba:

$$\sum_{i=1}^{m} \mathbf{e}_i y_i + \sum_{j \ne k} \mathbf{a}_j x_j + \mathbf{a}_k(\bar{x}_k + \ell_k) = \mathbf{b},$$

illetve

$$\sum_{i=1}^{m} \mathbf{e}_i y_i + \sum_{j \ne k} \mathbf{a}_j x_j + \mathbf{a}_k \bar{x}_k = \mathbf{b} - \mathbf{a}_k \ell_k$$

adódik. Ebből megfogalmazhatjuk a véges alsó korlátok eltolásának szabályát:

Egy x_k változóra érvényes ℓ_k véges alsó korlátot úgy lehet a nullába eltolni, hogy a k indexű oszlop ℓ_k-szorosát levonjuk a jobboldali $\mathbf{b}$ vektorból. $\mathbf{A}$ k-adik oszlopa változatlan marad, de az most $\bar{x}_k$-t reprezentálja. A feladat megoldása után x_k-t az $x_k = \bar{x}_k + \ell_k$ összefüggésből kell meghatározni.

2. Példa Az 1. példában az x_3 változónak véges alsó és felső korlátja van: $-4 \le x_3 \le -1$. Vezessük be az $\bar{x}_3 = x_3 + 4$ helyettesítést. Ekkor a 3. oszlop -4-szeresét le kell vonnunk a jobboldali vektorból:

$$\begin{bmatrix} 9 \\ -3 \\ 15 \\ 10 \\ 8 \end{bmatrix} - (-4) \begin{bmatrix} 6 \\ 0 \\ 3 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 33 \\ -3 \\ 27 \\ 6 \\ 4 \end{bmatrix}$$

$\bar{x}_3$ felső korlátja $\bar{u}_3 = -1 - (-4) = 3$ lesz. Végül tehát a transzformáció után $0 \le \bar{x}_3 \le 3$ lesz érvényben.

A példában van még egy MI típusú változó, x_2, amit PL-re változtathatunk oly módon, hogy a 2. oszlopban minden együttható előjelét az ellenkezőjére változtatjuk és megjegyezzük, hogy ez most $\bar{x}_2 = -x_2$-nek felel meg.

Változók új típusai Tegyük fel, hogy minden MI változó PL-re változtattunk és a véges alsó korlátokat a 0-ba toltuk. Ekkor, a változók (akár strukturális, akár logikai) érvényességi (megengedettségi) tartományuk alapján az alábbi módon kategorizálhatók:

Megengedettségi tartomány					Típus	Megnevezés	Címke
		y_i, x_j	$=$	0	0	Rögzített	FX
0	$\leq$	y_i, x_j	$\leq$	u_j	1	Korlátos	BD
0	$\leq$	y_i, x_j	$\leq$	$+\infty$	2	Nemnegatív	PL
$-\infty$	$\leq$	y_i, x_j	$\leq$	$+\infty$	3	Szabad	FR

(3.6)

Kapcsolat figyelhető meg a feltételek típusai és a hozzájuk rendelt logikai változók típusa közt az

$$y_i + \sum_{j=1}^{n} a_j^i x_j = b_i, \ i = 1, \ldots, m \tag{3.7}$$

összefüggésben. Nevezetesen:

y_i típusa	Feltétel típusa
0	Egyenlőség
1	Intervallum
2	$\geq$ vagy $\leq$
3	nem korlátozó.

Itt a feltétel típusa az eredeti (átalakítás előtti) típust jelenti. Az **LP standard alakja** eleve csupa nemnegatív (2-es típusú) változót és egyenlőség feltételt tartalmaz.

Az LP feladat új alakja Technikai szempontból nincs jelentősége, hogy a feladat megoldása során megkülönböztessük a strukturális és logikai változókat. Éppen ezért a

$$\min \quad \mathbf{c}^T\mathbf{x} \tag{3.8}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} + \mathbf{I}\mathbf{y} = \mathbf{b} \tag{3.9}$$

$$\text{és minden változó a 0–3 típusok valamelyikéhez tartozik} \tag{3.10}$$

alakot átírhatjuk egy egyszerűbb formára a következőképpen. Először is újra definiáljuk az $\mathbf{x}$ és $\mathbf{c}$ vektorokat

$$\mathbf{x} := \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}, \quad \mathbf{c} := \begin{bmatrix} \mathbf{c} \\ \mathbf{0} \end{bmatrix},$$

ahol $\mathbf{0}$ az m dimenziós nulla vektor. $\mathbf{x}$ és $\mathbf{c}$ új dimenziója $n := n + m$ lesz, vagyis a strukturális és logikai változók számának összege. Bevezetünk még egy jelölést: „type(x_j)” jelenti az x_j változó típusát, ami az előzőek alapján 0,1,2, vagy 3 lehet akár strukturális akár logikai változóról van szó. A (3.9) bal oldalán lévő mátrixot is újra definiáljuk:

$$\mathbf{A} := [\mathbf{A} \mid \mathbf{I}].$$

Így a (3.8) – (3.10) általános forma átírható a

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} && (3.11)\\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} && (3.12)\\ & \text{type}(x_j) \in \{0,1,2,3\},\ j = 1,\ldots,n && (3.13) \end{aligned}$$

alakra. A fentiekből látható, hogy a (3.12) alatti $\mathbf{A}$ mátrixnak több oszlopa van mint sora, hiszen tartalmaz egy $m \times m$-es egység mátrixot is.

Az eddigieket úgy foglalhatjuk össze, hogy minden LP feladat ilyen formára hozható, amire a továbbiakban a **számítási forma 1**-gyel fogunk hivatkozni.

3.3. A szimplex módszer alapjai

A lineáris programozási feladat legismertebb és legfontosabb megoldó algoritmusa a **szimplex módszer**, illetve annak különféle variánsai. Ez egy algebrai módszer, ami egy alulhatározott (több a változó, mint az egyenlőségek száma) lineáris egyenletrendszer megoldásai közül választja ki azt, amelyik az adott célfüggvény szempontjából a legjobb. Ezt részletesen is tárgyalni fogjuk. A módszer leírásához és megértéséhez szükség van néhány alapfogalom bevezetésére.

Egy n dimenziós $\mathbf{x}$ vektort, amely kielégíti a (3.12) egyenlőségeket, **megoldás**nak nevezünk. Ha egy megoldás teljesíti a (3.13) alatti egyedi korlátokat is, akkor a neve **megengedett megoldás**. A megengedett megoldások halmazát $\mathcal{X}$-szel jelöljük.

Egy megengedett $\mathbf{x}^*$ megoldás neve **optimális megoldás**, ha $\mathbf{c}^T\mathbf{x}^* \leq \mathbf{c}^T\mathbf{x}$ teljesül minden $\mathbf{x} \in \mathcal{X}$- re, vagyis semmilyen megengedett $\mathbf{x}$-re nem kapunk jobb célfüggvény értéket.

A (3.12)-ban szereplő $\mathbf{A}$ matrix rangja m, miután van benne egy $m \times m$-es egység mátrix. Ennek következtében az n oszlop közül ki lehet választani m lineárisan független vektort. Ezek együttesen egy $\mathbf{B}$ nem-szinguláris mátrixot alkotnak, amit **bázis**nak, vagy **bázis mátrix**nak hívunk. Az általánosság megszorítása nélkül feltehetjük, hogy ezek az oszlopok az $\mathbf{A}$ mátrix első m pozíciójába vannak permutálva. Ekkor $\mathbf{A}$-t felírhatjuk particionált alakban: $\mathbf{A} = [\mathbf{B} \,|\, \mathbf{R}]$, ahol $\mathbf{R}$ jelöli az $\mathbf{A}$ mátrix maradék (nem-bázis) részét.

Azokat az változókat, amelyek a bázisbeli oszlopokhoz tartoznak **bázis változók**nak nevezzük, míg a többit **nem-bázis változó**nak. A $\mathbf{c}$ és $\mathbf{x}$ vektorokat is ugyanígy particionálhatjuk. A bázis változók indexhalmazát B-vel, míg a nem-bázis változókét R-rel jelöljük. Ezek után (3.12)-et úgy is írhatjuk, hogy

$$\mathbf{A}\mathbf{x} = [\mathbf{B} \,|\, \mathbf{R}] \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_R \end{bmatrix} = \mathbf{b},$$

vagy

$$\mathbf{B}\mathbf{x}_B + \mathbf{R}\mathbf{x}_R = \mathbf{b}. \tag{3.14}$$

A célfüggvény particionált alakja:

$$\mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_R^T\mathbf{x}_R.$$

(3.14)-ból azt kapjuk, hogy $\mathbf{B}\mathbf{x}_B = \mathbf{b} - \mathbf{R}\mathbf{x}_R$ és, mivel $\mathbf{B}^{-1}$ létezik (hiszen $\mathbf{B}$ nem-szinguláris),

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x}_R). \tag{3.15}$$

$\mathbf{x}_B$-t a $\mathbf{B}$ bázishoz tartozó **bázis megoldás**nak nevezzük. Ha $\mathbf{x}_B$ megengedett, akkor **megengedett bázis megoldás**ként hivatkozunk rá.

(3.15) azt mondja, hogy a bázis megoldást egyértelműen meghatározzák a nem-bázis változók értékei. A nem-bázis változók értéke általában nulla. Ez esetben (3.15) $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$-re egyszerűsödik. Egy **bázis megoldás degenerált**, ha $\mathbf{x}_B$ legalább egy komponense nulla értéken van. Ennek lényegéről később lesz szó.

A különböző bázisok maximális száma roppant nagy lehet, ugyanis n elemből próbálunk kiválasztani m-et minden lehetséges módon és ezek száma

$$\binom{n}{m} = \frac{n!}{m!\,(n-m)!},$$

ami már $m = 50$ és $n = 100$ (nem egy nagy LP feladat) esetén is $\binom{100}{50} \approx 1.01 \times 10^{29}$, egy csillagászati szám!

Általános esetben az $\mathbf{A}$ mátrix tetszőleges oszlopai képezhetik a bázist. Ekkor a bázisváltozók indexhalmaza

$$\mathcal{B} = \{k_1, \ldots, k_m\}$$

Ennek jelentése: például, ha az első bázisvektor az $\mathbf{A}$ mátrix 24-ik oszlopa, akkor $k_1 = 24$.

A lineáris programozás egyik fő tétele azt mondja ki, hogy ha egy feladatnak van optimális megoldása, akkor van **optimális bázis megoldás**a is. Ebből következik, hogy a megoldás során elegendő bázis megoldásokat vizsgálni.

Ha több mint egy (mondjuk r) optimális megoldás van, akkor **alternatív optimum**ról beszélünk. Ilyen esetben az alternatív optimális megoldások ($\mathbf{x}$-ek) tetszőleges konvex lineáris kombinációja, azaz

$$\lambda_1\mathbf{x}_1 + \cdots + \lambda_r\mathbf{x}_r, \quad \lambda_i \geq 0,\ i = 1, \ldots, r, \quad \sum_{i=1}^{r} \lambda_i = 1$$

szintén optimális megoldás, vagyis végtelen sok optimális megoldás van, hiszen végtelen sok λ kombináció van, amelyek a konvex lineáris kombináció definícióját kielégíti. Ennek igazolása a következőképpen lehetséges.

Tegyük fel, hogy $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_r$ mindegyike optimális (nem feltétlenül bázis) megoldás azonos, z^* értékkel. Könnyen belátható, hogy ezek konvex lineáris kombinációja szintén megengedett megoldás és ezen a célfüggvény értéke:

$$\lambda_1\mathbf{c}^T\mathbf{x}_1 + \cdots + \lambda_r\mathbf{c}^T\mathbf{x}_r = (\underbrace{\lambda_1 + \cdots \lambda_r}_{=1})z^* = z^*,$$

amivel az állítást igazoltuk.

3.4. A primál szimplex módszer

A szimplex módszer egy iteratív numerikus eljárás, amely a degeneráció mentes esetben (egyetlen bázis sem degenerált) véges számú lépésben az alábbi kimenetelek egyikével áll meg:

1. A feladatnak nincs megengedett megoldása (a feltételek ellentmondást tartalmaznak, $\mathcal{X} = \emptyset$).
2. A feladat megoldása nem korlátos.
3. Egy optimális megoldást sikerült találni.

Két bázist **szomszédos**nak mondunk, ha csak egyetlen oszlop vektorban különböznek egymástól. A szimplex módszer szomszédos bázisokon halad úgy, hogy közben a célfüggvény értéke állandóan javul (néha változatlan marad, lásd degeneráció).

A szimplex módszer leírása általános esetre [2]-ben található meg.

Standard alak: optimalitási feltételek A továbbiakban az LP feladat standard alakjával foglalkozunk, vagyis:

$$\begin{aligned} \min z &= \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned}$$

Tegyük fel, hogy ennek a feladatnak ismerjük egy megengedett $\mathbf{B}$ bázisát. A bázison kívüli változók értéke most mind nulla. Emlékeztető: $\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x}_R)$. Ha ezt az $\mathbf{x}_B$-t behelyettesítjük a célfüggvény particionált $z = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_R^T\mathbf{x}_R$ alakjába, akkor $z = \mathbf{c}_B^T\mathbf{B}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x}_R) + \mathbf{c}_R^T\mathbf{x}_R$-t kapunk, amiből átrendezés után

$$z = \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_R^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{R})\mathbf{x}_R. \tag{3.16}$$

lesz. Ebben az egyenlőségben egy x_k (az $\mathbf{x}_R$ egyik komponense) bázison kívüli változó együtthatója

$$d_k = c_k - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{a}_k,$$

amit **redukált költség**nek nevezünk.

(3.16) azt mutatja, hogy ha x_k-t elmozdítjuk jelenlegi (nem-bázis) értékéről és ezt az elmozdulást t-vel jelöljük, akkor, a többi nem-bázis változó változatlanul hagyása mellett a célfüggvény td_k értékkel változik:

$$z(t) = z + td_k, \tag{3.17}$$

vagyis a változás mértékét a d_k redukált költség fejezi ki. Miután x_k-nak csak olyan elmozdulása jöhet szóba, hogy az új érték is megengedett legyen, ezért $t \geq 0$-nak van csak értelme. Ebből rögtön következik, hogy mi az az eset, amikor egyetlen változó megengedett elmozdulása sem tud javítani a célfüggvény értékén:

$$d_k \geq 0, \quad \text{minden } k \in \mathcal{R}. \tag{3.18}$$

Ez egyben az **optimalitás elégséges feltétel**ét jelenti a minimalizálási feladatra. Ennek igazolásához elegendő észrevenni, hogy x_k minden megengedett (nemnegatív) elmozdulására a célfüggvény (3.17) alapján nem javul.

Maximalizálási feladat esetén d_k-ra fordított irányú egyenlőtlenség áll fenn egy optimális bázisnál.

3.5. Báziscsere a szimplex módszerben

Az eddigiek alapján körvonalazható a szimplex módszer. Ez egy olyan iteratív eljárás, amely egyre jobb szomszédos bázis megoldásokon keresztül halad egy optimális megoldás felé. Ennek megfelelően, minden iteráció azzal kezdődik, hogy megvizsgáljuk az éppen aktuális bázist. Ha a nem-bázis változók redukált költségei kielégítik a (3.18) optimalitási feltételeket, akkor a mostani bázis optimális, az eljárás befejeződik.

Amennyiben a nem-bázis változók közt akár csak egy is van amelyik nem teljesíti (3.18)-at akkor a bázis nem teljesíti az optimalitás elégséges feltételét. A kritériumot nem teljesítő változók közül valamelyiket kiválasztjuk (hogy hogyan, arról később) és ezt **javító változó**nak, a hozzá tartozó oszlop vektort **javító vektor**nak nevezzük.

(3.17) alatt láttuk, hogy a célfüggvény javulása arányos a kiválasztott nem-bázis változó elmozdulásának t nagyságával. Kívánatos volna tehát t-nek minél nagyobb értéket adni. Mint látni fogjuk, ennek gátat szab, hogy míg növeljük t-t, addig a bázis változók is elmozdulnak. Egy ilyen elmozdulás csak akkora lehet, hogy minden bázis változó még megengedett értéken maradjon. Ha megtaláltuk a legnagyobb, még érvényes elmozdulást, akkor a kiválasztott javító változó ezen az értéken belép a bázisba valamelyik bázis változó helyére.

A báziscserének több követelményt kell teljesíteni, úgymint

1. A célfüggvény javuljon.
2. A belépő változó megengedett értéket vegyen fel.
3. A kilépő változó megengedett értéken hagyja el a bázist.
4. Minden bázis változó megengedett értéken maradjon.

Az első követelmény automatikusan teljesül, hiszen ennek alapján választjuk ki a belépő jelöltet. A többi 3 követelmény teljesülését az ún. hányados próba alkalmazásával tudjuk biztosítani. Ennek részletei rövidesen következnek.

Tegyük fel, hogy $d_q < 0$ alapján kiválasztottunk egy x_q javító változót a hozzá tartozó $\mathbf{a}_q$ javító vektorral együtt. A t javító elmozdulás tehát pozitív, a td_q szorzat negatív, a célfüggvény csökken.

Tegyük fel, hogy x_q elmozdul $x_q + t$-re (ami persze $x_q = 0$ miatt t-vel egyenlő). Ekkor a bázisváltozók is elmozdulnak t függvényében, hogy továbbra is teljesüljenek az alapegyenletek:

$$\mathbf{B}\mathbf{x}_B(t) + t\mathbf{a}_q = \mathbf{b}. \tag{3.19}$$

(3.19)-ből $\mathbf{x}_B(t)$ kifejezhető:

$$\mathbf{x}_B(t) = \mathbf{B}^{-1}\mathbf{b} - t\,\mathbf{B}^{-1}\mathbf{a}_q. \tag{3.20}$$

Az $\boldsymbol{\alpha}_q = \mathbf{B}^{-1}\mathbf{a}_q$, $\boldsymbol{\beta} = \mathbf{B}^{-1}\mathbf{b}$, $\boldsymbol{\beta}(t) = \mathbf{x}_B(t)$ jelölések bevezetésével (3.20)-ból az egyszerűsített

$$\boldsymbol{\beta}(t) = \boldsymbol{\beta} - t\boldsymbol{\alpha}_q. \tag{3.21}$$

alakot kapjuk. (3.21) i-edik komponense:

$$\beta_i(t) = \beta_i - t\alpha_q^i. \tag{3.22}$$

Célunk t legnagyobb értékének a meghatározása úgy, hogy az összes $\beta_i(t)$, $i = 1, \ldots, m$, bázisváltozó és x_q maga is megengedett értéken maradjon.

Az i-edik bázisváltozó megengedett értéken marad mindaddig, amíg t-re

$$\beta_i - t\alpha_q^i \geq 0 \tag{3.23}$$

teljesül.

(3.22)-ból látszik, hogy $\beta_i(t)$ egy csökkenő függvény, ha $\alpha_q^i > 0$. Ebben az esetben a legnagyobb érték, amit t felvehet: $t_i = \beta_i/\alpha_q^i$, ami a (3.23) egyenlőtlenségből adódik, ha $\beta_i - t\alpha_q^i = 0$. Nyilvánvalóan minden bázisváltozó, amelyhez tartozó $\alpha_q^i > 0$ megengedett értéken marad, ha t nem nagyobb, mint

$$\theta = \min\{t_i\} = \min\left\{\frac{\beta_i}{\alpha_q^i},\ \alpha_q^i > 0\right\}. \tag{3.24}$$

Ez a formula a hagyományos **hányados próba**.

Másrészt, ha $\alpha_q^i < 0$ akkor $\beta_i(t)$ egy növekvő függvény, így t növelésével a $\beta_i - t\alpha_q^i$ kifejezés végig pozitív marad, tehát nem korlátozza az x_q belépő változó értékét.

Ha (3.24)-ben a minimum a p-edik bázis pozíción vétetik fel, akkor az itt levő bázisváltozó eléri az alsó korlátját. Így az megengedett értéken ki tud lépni a bázisból. Helyére az x_q változó lép be a „megüresedő” p-edik pozícióra

$$\bar{x}_q = x_q + \theta \tag{3.25}$$

értéken.

Ha a minimum nem egyértelmű (több olyan pozíció is van, ahol a hányados egyenlő θ-val), akkor bármelyikhez tartozó változót választhatjuk kilépőnek.

Technikailag a (3.24) hányados próbát a következőképpen célszerű elvégezni. Vegyük az $\alpha_q^1, \ldots, \alpha_q^i, \ldots, \alpha_q^m$ elemeket ebben a sorrendben és számítsuk ki a

$$t_i = \frac{\beta_i}{\alpha_q^i}, \quad \alpha_q^i > 0 \tag{3.26}$$

mennyiségeket, majd vegyük azt a hányadost, amelyik a legkisebb. (Jelöljük ennek indexét p-vel.) Ez határozza meg x_q legnagyobb legnagyobb elmozdulását, ami mellett a bázisváltozók még megengedett szinten maradnak. Tehát:

$$\theta = \frac{\beta_p}{\alpha_q^p} = \min\{t_i\} = \min\left\{\frac{\beta_i}{\alpha_q^i}, \quad \alpha_q^i > 0\right\}. \tag{3.27}$$

Ha nincs egyetlen pozíció sem, amire $\alpha_q^i > 0$, akkor definíciószerűen $\theta = +\infty$ és ezesetben a megoldás nem korlátos. Egyébként $\theta < +\infty$ és ez a belépő változó elmozdulásának a **lépéshossz**a (3.21)-ban. Ennek következtében a bázisváltozók új értéke

$$\bar{\boldsymbol{\beta}} = \boldsymbol{\beta}(\theta) = \boldsymbol{\beta} - \theta \boldsymbol{\alpha}_q. \tag{3.28}$$

(3.17) alapján tudjuk, hogy a célfüggvény új értéke a θ lépéshossz után $\bar{z} = z + \theta d_q$.

A hányados próba alapján báziscsere történik: a p-edik bázisváltozó, β_p (aminek az $\mathbf{A}$ mátrixbeli indexe k_p) 0 szintre kerül és elhagyja el a bázist. A kilépő változó helyére belép a kiválasztott x_q és a p-edik pozíción ez lesz az új β_p bázisváltozó. x_q értékét (3.25) határozta meg, vagyis $\beta_p \equiv \bar{x}_q = x_q + \theta$, ami $x_q = 0$ miatt θ-val egyenlő.

Az új bázis $\bar{\mathcal{B}} = \mathcal{B} \setminus \{k_p\} \cup \{q\}$ lesz. (3.28) és (3.25) alapján az új bázismegoldás:

$$\bar{\beta}_i = \beta_i - \theta \alpha_q^i, \quad i = 1, \ldots, m, \; i \neq p, \tag{3.29}$$

$$\bar{\beta}_p = x_q + \theta. \tag{3.30}$$

A kilépő változó új értéke 0.

A célfüggvény új értékét (3.17)-ból kaphatjuk meg a t helyére θ-t helyettesítve:

$$\bar{z} = z(\theta) = z + \theta d_q. \tag{3.31}$$

Példa 1 *Tegyük fel, hogy adva van egy megengedett bázis a hozzátartozó megoldással:* $\boldsymbol{\beta} = \mathbf{x}_{\mathcal{B}}$, *továbbá egy javító változó,* $x_q = 0$, $d_q = -3$, *a célfüggvény érték,* $z = 12$, *és* $\boldsymbol{\alpha}_q$, *lásd az alábbi táblázat baloldalát. Határozzuk meg a hányadosokat, a kilépő változót, a belépő változó értékét, az új bázist és a célfüggvény új értékét.*

i	β_i	α_q^i	t_i
1	0	-1	–
2	8	2	$8/2 = 4$
3	6	3	$6/3 = 2$
4	2	0	–
5	5	1	$5/1 = 5$

A fentiekből, $\theta = \min\{4,2,5\} = 2$ *és* $p = 3$ *(a minimális hányadost meghatározó sor indexe). Vegyük észre, hogy a degeneráció ellenére a minimális hányados pozitív. Az új megoldás*

$$\boldsymbol{\beta}(\theta) = \boldsymbol{\beta} - 2 \times \boldsymbol{\alpha}_q = \begin{bmatrix} 0 \\ 8 \\ 6 \\ 2 \\ 5 \end{bmatrix} - 2 \begin{bmatrix} -1 \\ 2 \\ 3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \\ 2 \\ 3 \end{bmatrix}.$$

x_{k_3} *nullává vált és kilép a bázisból. Helyére belép* $x_q = 2$ *harmadik bázisváltozóként.*

$\bar{\mathcal{B}} = \mathcal{B} \setminus \{k_3\} \cup \{q\}$ *és* $\bar{\boldsymbol{\beta}} = \mathbf{x}_{\bar{\mathcal{B}}} = [2,4,2,2,3]^T$.

Az új célfüggvényérték (3.17) alapján $\bar{z} = z(\theta) = z + \theta d_q = 12 + 2(-3) = 6$.

3.6. A standard primál szimplex módszer lépései

Az eddigiek lehetővé teszik, hogy megoldjunk LP feladatokat a szimplex módszerrel, ha ismerünk egy megengedett $\mathcal{B}$ bázist. Feltételezzük, hogy a feladat a standard alakban van megadva. Fontos az algoritmus leírása utáni megjegyzések tanulmányozása, amik a gyakorlati számításhoz nyújtanak segítséget.

Az alábbi algoritmikus lépések a szimplex módszer úgynevezett **második fázis**át írják le ami akkor használható, ha ismerünk egy megengedett bázis megoldást. A későbbiekben szó lesz az **első fázis**ról is, amikoris az algoritmus egy megengedett megoldás megkeresését végzi el.

Standard primál szimplex módszer második fázisának lépései

0. lépés **Inicializálás.** Ellenőrzés: a kezdeti $\mathcal{B}$ bázis megengedett-e. Határozzuk meg $\mathbf{B}^{-1}$-t, valamint a hozzátartozó $\boldsymbol{\beta} = \mathbf{B}^{-1}\mathbf{b}$ megoldást és a $z = \mathbf{c}_B^T\mathbf{x}_B$ célfüggvényértéket.

1. lépés Számítsuk ki a $\boldsymbol{\pi}$ szimplex szorzót: $\boldsymbol{\pi}^T = \mathbf{c}_B^T\mathbf{B}^{-1}$.

2. lépés A szimplex szorzó segítségével meg tudjuk határozni a bázison kívüli változók redukált költségeit és ezáltal ellenőrizni tudjuk az optimalitási feltételek teljesülését: $d_j = c_j - \boldsymbol{\pi}^T\mathbf{a}_j$, $j \in \mathcal{R}$. Ha minden $\mathcal{R}$-beli változó teljesíti a rá vonatkozó feltételt, akkor a jelenlegi **megoldás optimális**. Ellenkező esetben van legalább egy változó, amelyik nem teljesíti. Általában több ilyen változó is akad. Ilyenkor bármelyiket ki lehet választani, hogy javító vektorként belépjen a bázisba. Célszerű azt választani, amelyik a legjobban megsérti az optimalitási feltételét. Legyen ez a változó x_q.

3. lépés Határozzuk meg az x_q-hoz tartozó oszlopvektor transzformált alakját: $\boldsymbol{\alpha}_q = \mathbf{B}^{-1}\mathbf{a}_q$.

4. lépés Hajtsuk végre a hányados próbát (3.27) alapján hogy megkapjuk a lépéshosszt, θ-t. Ha $\theta = \infty$ akkor a **megoldás nem korlátos**, az eljárás befejeződik. Ellenkező esetben megtaláltuk a kilépő változót: van báziscsere, a p-edik bázisváltozó távozik.

5. lépés Transzformációk. A célfüggvény: $\bar{z} = z + \theta d_q$. Az új megoldás:

$$\bar{\beta}_i = \beta_i - \theta\alpha_q^i, \quad \text{for } i = 1,\dots,m,\ i \neq p, \tag{3.32}$$

$$\bar{\beta}_p = x_q + \theta. \tag{3.33}$$

Meg kell még határozni az új bázis ($\bar{\mathbf{B}}$) inverzét is a következő iteráció számára. Ez legegyszerűbben az $\mathbf{B}^{-1}$ transzformálásával érhető el. Jelöljük $\mathbf{B}^{-1}$ sorait $\boldsymbol{\rho}^i$-vel, $i = 1,\dots,m$. Könnyen belátható, hogy az új inverz p-edik sora $\bar{\boldsymbol{\rho}}^p = (1/\alpha_q^p)\boldsymbol{\rho}^p$. Ennek segítségével a többi sor: $\bar{\boldsymbol{\rho}}^i = \boldsymbol{\rho}^i - \alpha_q^i\bar{\boldsymbol{\rho}}^p$ minden $i = 1,\dots,m$, $i \neq p$-re. $\bar{\mathbf{B}}^{-1}$ sorai tehát: $\bar{\boldsymbol{\rho}}^1,\dots,\bar{\boldsymbol{\rho}}^m$.

Megjegyzések.

1. Egy LP feladatot standard alakra (összes feltétel egyenlőség) a következőképpen lehet hozni.

(a) Minden $\leq$ feltétel baloldalához adjunk hozzá egy nemnegatív logikai változót.

(b) Minden $\geq$ feltételt szorozzunk be -1-gyel (ezáltal $\leq$ lesz) és adjunk a baloldalhoz egy nemnegatív logikai változót.

(c) Hagyjunk minden = feltételt változatlanul (ne adjunk hozzá semmilyen logikai változót).

2. Kis gyakorló feladatok esetén az **I** egység mátrix, amennyiben jelen van, lehet egy jó induló bázis. Ilyenkor $\mathbf{B}^{-1} = \mathbf{I}$, így semmit nem kell számolni, hiszen ekkor $\mathbf{x}_B = \mathbf{b}$ (vagyis a jobboldal) az induló bázismegoldás, amiről gyakran kiderül, hogy megengedett ($\mathbf{b} \geq \mathbf{0}$), így az algoritmus el tud indulni.

3. A fenti leírás általánosan felírt, akár nagyméretű LP feladatok megoldására is képes változata a szimplex algoritmusnak. Ennek neve **módosított szimplex módszer**. A Dantzig által kifejlesztett eredeti változat az ún. teljes **tabló transzformációs szimplex** (TTTS). Kisméretű feladatok esetén ezt a verziót célszerű használni. Ekkor az iterációkhoz szükséges minden információ azonnal (kiegészítő számítások nélkül) rendelkezésre áll. TTTS használata esetén az **A** mátrixot kibővítjük úgy, hogy a célfüggvényt hozzávesszük a mátrix utolsó ($m+1$-edik) soraként (ezt hívják teljes tablónak) és ezt is transzformáljuk minden báziscsere alkalmával. Ez a sor tartalmazza a redukált költségeket. Ennek a változatnak a használata egyszerű. A báziscserék során amikor a belépő q és kilépő p bázis indexet meghatároztuk, egyúttal ismertté válik az α_q^p elem, amit **pivot elem**nek, (néha generáló elemnek) hívnak. Az egész kibővített mátrixon végrehajtunk egy **Gauss-Jordan (G-J) eliminációt**, amit a pivot elem egyértelműen meghatároz. Ezután ismét minden adat rendelkezésre áll, ami a következő iterációhoz szükséges.

   Emlékeztetés céljából verbálisan megismételjük a G-J eliminációt. A pivot elemmel elosztjuk a pivot sor valamennyi elemét. Így α_q^p új értéke 1 lesz. A pivot sor alkalmas többszöröseit kivonjuk a többi sorból úgy, hogy a pivot elem fölött és alatt csupa nulla keletkezzék. Ezáltal a pivot oszlop egy egységvektor lesz.

   Az olvasó számára egy jó gyakorlat végiggondolni hogyan módosulnak a fent leírt algoritmusnak a lépései ha a TTTS verziót használjuk.

4. Ha az optimalitás ellenőrzése során több változó is megsérti az optimalitási kritériumot, akkor elvileg bármelyiket kiválaszthatjuk belépő változónak. Főleg a nagyméretű feladatok esetén azonban nagyon is fontos, hogy minden lépésben egy „jó” változót válasszunk ki, mert ezáltal a megoldás eléréséhez szükséges iterációk száma jelentősen csökkenhet. Sajnos, általában nehéz előre látni, hogy mi lesz egy jó jelölt. Erre nézve sok kritériumot dolgoztak ki, de egyik sem olyan, amelyik minden esetben a legjobb. Dantzig eredetileg azt javasolta, hogy az legyen a belépő, amelyik a legnagyobb mértékben sérti meg az optimalitási feltételt. Ezt a kiválasztást Dantzig szabálynak hívják és kézi számításokra ezt szokták használni.

5. Van egy érdekes jelenség a szimplex módszerben, amit **degeneráció**nak neveznek. Egy bázismegoldásról azt mondjuk, hogy degenerált ha legalább egy bázisváltozó nulla értéket vesz fel.

Például, tegyük fel, hogy a bázisváltozók x_5, x_2 és x_3 (ebben a sorrendben) és ezekre a nemnegatívitási korlátok vannak érvényben. Ha a változók értékei $x_5 = 1,\ x_2 = 2,\ x_3 = 0$, akkor ez a bázis megoldás degenerált, mivel $x_3 = 0$. Ha az értékek $x_5 = 2,\ x_2 = 1,\ x_3 = 3$ lennének, akkor a megoldás nem lenne degenerált.

A fő probléma a degenerációval az, hogy egy degenerált bázis esetén a θ lépéshossz esetleg (de nem feltétlenül) nullának adódhat és ilyenkor nem javul a célfüggvény (emlékeztetőül: $\bar{z} = z + \theta d_q$) bár a báziscserét ekkor is végre kell hajtani.

A szimplex módszer érzékeny a degenerációra, mert megtörténhet, hogy egymásután sok nem-javító lépésre kényszeríti az eljárást (angol neve: stalling), sőt még az is bekövetkezhet, hogy egy nem-javító iterációk sorozata végtelen sokszor ismétlődik, amit ciklizálásnak nevezünk. A gyakorlatban ciklizálás nem szokott előfordulni, de bizonyos típusú feladatok esetén a stalling nem túl ritka jelenség.

A fentiek jobb megértése érdekében most egy példát mutatunk be a standard szimplex módszer TTTS változatára.

Példa 2

$$\begin{array}{rrcrcrcl} \min z = & -x_1 & - & x_2 & - & 2x_3 & & \\ f.h. & 2x_1 & + & x_2 & + & x_3 & \leq & 1 \\ & & & x_2 & + & 2x_3 & \leq & 3 \\ & -x_1 & & & + & 3x_3 & \leq & 2 \\ & & & x_1, x_2, x_3 \geq 0 & & & & \end{array}$$

Először minden feltételt egyenlőséggé alakítunk át úgy, hogy mindegyikhez egy nemnegatív logikai változót adunk hozzá.

$$\begin{array}{rrcrcrcrcrcl} \min z = & -x_1 & - & x_2 & - & 2x_3 & & & & & \\ f.h. & 2x_1 & + & x_2 & + & x_3 & +y_1 & & & = & 1 \\ & & & x_2 & + & 2x_3 & & +y_2 & & = & 3 \\ & -x_1 & & & + & 3x_3 & & & +y_3 & = & 2 \\ & & & x_1, x_2, x_3 \geq 0, & & & y_1, y_2, y_3 \geq 0 & & & & \end{array}$$

Induló bázisnak a logikai változókhoz tartozó egység mátrixot lehet választani. Így az induló bázis megoldás: $y_1 = 1,\ y_2 = 3$ és $y_3 = 2$, ami nyilván egy megengedett megoldás, hiszen mindegyik logikai változó nemnegatív értéket vesz fel. A feladat tabló formáját úgy kapjuk, hogy a feltételi mátrixhoz hozzávesszük utolsó sorként a célfüggvényt, valamint utolsó oszlopként a jobboldalt. Ez utóbbi transzformált alakja tartalmazza a mindenkori bázis megoldást, $\mathbf{x}_B$-t. A $\mathbf{d}^T$ sor és $\mathbf{x}_B$ oszlop által meghatározott cellában a célfüggvény értéke szerepel. Ez a pozíció a (3.31) képlet alapján transzformálódik. Mindezek eredményeként az alábbi induló tablót kapjuk.

	x_1	x_2	x_3	y_1	y_2	y_3	$\mathbf{x}_B$
y_1	2	1	1	1			1
y_2	0	1	2		1		3
y_3	−1	0	3			1	2
$\mathbf{d}^T$	−1	−1	−2	0	0	0	0

A baloldali oszlop az egyes sorokhoz tartozó bázisváltozók listáját tartalmazza, ami az indulásnál $y_1 = 1,\ y_2 = 3$ *és* $y_3 = 2$. *A bázison kívüli változók most a strukturális változók:* $x_1,\ x_2$ *és* x_3 *és ezek nulla értéken vannak.*

A redukált költségeket tartalmazó sorból ($\mathbf{d}^T$*) látható, hogy valamennyi bázison kívüli változó megsérti az optimalitási feltételét. Válasszuk ezek közül az* x_2*-höz tartozó* $d_2 = -1$*-et noha nem ez mutatja a legnagyobb eltérést a* $d_j \geq 0$ *optimalitási feltételtől. Így most* x_2 *a javító (belépő) változó, vagyis* $q = 2$.

Végrehajtjuk a hányados próbát, hogy meghatározzuk a bázisból kilépő változót. Ez a (3.27) a táblázat alapján történik. A próbában résztvevő hányadosok: az első sorban $t_1 = 1/1 = 1$ *és a második sorban* $t_2 = 3/1 = 1$. *Ezek minimuma 1, ami aí első soron vétetik fel* $(p = 1)$ *és a lépéshossz* $\theta = 1$ *lesz. A pivot elem ennek a hányadosnak a nevezője, vagyis* $\alpha_q^p = 1$. *Ezek alapján az első bázisváltozó,* y_1 *lép ki, helyére bejön a kiválasztott* x_2. *Az alábbi tablón a pivot elemet bekereteztük.*

	x_1	x_2	x_3	y_1	y_2	y_3	$\mathbf{x}_B$
y_1	2	$\boxed{1}$	1	1			1
y_2	0	1	2		1		3
y_3	−1	0	3			1	2
$\mathbf{d}^T$	−1	−1	−2	0	0	0	0

A célfüggvény értéke a $\theta = 1$ *lépéshossz miatt nyilván* $\theta d_q = 1 \times (-1) = -1$*-gyel változik, vagyis* -1 *lesz. A tablón a Gauss-Jordan eliminációt az* α_2^1 *pivot elemmel végrehajtva az alábbit kapjuk (a bekeretezett* $\boxed{3}$*-at egyelőre hagyjuk figyelmen kívül):*

	x_1	x_2	x_3	y_1	y_2	y_3	$\mathbf{x}_B$
x_2	2	1	1	1	0	0	1
y_2	−2	0	1	−1	1	0	2
y_3	−1	0	$\boxed{3}$	0	0	1	2
$\mathbf{d}^T$	1	0	−1	1	0	0	−1

Ha most megvizsgáljuk a bázison kívüli változók ($x_1,\ x_3$ *és* y_1*) redukált költségét, azt tapasztaljuk, hogy csak* d_3 *negatív, így* x_3*-at választjuk javító változónak,* $q = 3$. *A* t_i *hányadosok most:* $t_1 = 1/1 = 1,\ t_2 = 2/1 = 2,\ t_3 = 2/3$. *Ezek minimuma* $\theta = 2/3$, *ami a harmadik soron vétetik fel, így* $p = 3$ *és a pivot elem a bekeretezett* $\boxed{3}$. *A célfüggvény megváltozása* $\theta d_q = 2/3 \times (-1) = -2/3$, *tehát az új értéke* $-1 - 2/3 = -5/3$. *Ha a kijelölt pivot elemmel végrehajtjuk a G-J eliminációt, a következő tabló adódik:*

	x_1	x_2	x_3	y_1	y_2	y_3	$\mathbf{x}_B$
x_2	7/3	1	0	1	0	−1/3	1/3
y_2	5/3	0	0	−1	1	−1/3	4/3
x_3	−1/3	0	1	0	0	1/3	2/3
$\mathbf{d}^T$	2/3	0	0	1	0	1/3	−5/3

Látható, hogy minden bázison kívüli változó (x_1, y_1, y_3) redukált költsége nemnegatív. Így a jelenlegi bázis megoldás teljesíti az optimalitási feltételeket. Az optimális megoldásban a strukturális változók értéke: $x_1 = 0$, $x_2 = 1/3$, $x_3 = 2/3$, a logikaiaké: $y_1 = 0$, $y_2 = 4/3$, $y_3 = 0$ és a célfüggvény értéke: $-5/3$.

Jó gyakorló feladat az olvasónak: oldja meg ugyanezt a feladatot úgy, hogy az első lépésben egy másik belépő változót választ. Eredményül ugyanezt kell kapnia.

3.7. Megengedett megoldás keresése

Az előbb ismertetett szimplex módszer működésének van egy nagyon fontos előfeltétele, éspedig az, hogy ismerünk egy megengedett bázist. Ha szerencsénk van, akkor a tisztán logikai változókból álló bázis (egységmátrix) ilyen. Ha nem, akkor szükség van egy szisztematikus eljárásra, ami talál egy ilyen bázist. Érdekes módon, egy kis módosítással maga a szimplex módszer használható erre a célra. Ezt hívjuk a szimplex módszer **első fázis**ának.

Az alapfeladat

$$\begin{aligned} \min z &= \mathbf{c}^T\mathbf{x} \\ \text{f.h.} \quad \mathbf{A}\mathbf{x} &= \mathbf{b}, \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned} \tag{3.34}$$

amiben jelen lehet egy egységmátrix is. Először tegyük fel, hogy ez a helyzet és ehhez a feladathoz keresünk egy $\mathbf{B}$ bázist, amellyel $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$. Ha $\mathbf{b} \geq \mathbf{0}$ akkor az egységmátrix teljesíti ezt a kritériumot (ugyanis $\mathbf{x}_B = \mathbf{I}^{-1}\mathbf{b} = \mathbf{I}\mathbf{b} = \mathbf{b} \geq \mathbf{0}$) és erről a bázisról el tud indulni a szimplex módszer. Ha van olyan feltételi sor, amelyre $b_i < 0$ akkor a feltételt -1-gyel beszorozva a jobboldal pozitív lesz. Ezesetben azonban a jelenlevő egységmátrix i-edik sorában $+1$ helyett -1 fog állni és így, ha bent lenne a bázisban, akkor a hozzátartozó logikai változó értéke negatív (nem megengedett) lenne. Ilyenkor a következő egyszerű gondolatmenet alkalmazható, amelyet rögtön finomítani is fogunk.

Szorozzuk meg minden $b_i < 0$ sort -1-gyel. Ettől minden feltétel változatlanul egyenlőség marad. Adjunk minden sorhoz egy x_{n+i} nemnegatív mesterséges változót, aminek oszlopa egy egységvektor (az i-edik sorhoz az i-edik egységvektor). Ezáltal a feladathoz hozzávettünk egy egységmátrixot, amihez tartozó bázismegoldás megengedett lesz. Formálisan a feltételek így alakulnak:

$$\sum_{j=1}^{n} a_j^i x_j + x_{n+i} = b_i, \quad i = 1, \ldots, m, \tag{3.35}$$

ahol $x_j \geq 0$ minden $j = 1, \ldots, n+m$-re. Ez a feladat különbözik az eredetitől, de megvan az az előnye, hogy egy megengedett bázis közvetlenül a rendelkezésre áll. Nevezetesen, a bázis a mesterséges változókhoz tartozó egységmátrix és a hozzá tartozó megengedett megoldás $x_{n+i} = b_i$, minden $i = 1, \ldots, m$-re.

(3.35) ekvivalens (3.34)-vel ha $x_{n+i} = 0$ minden $i = 1, \ldots, m$-re. Ezért célul tűzzük ki, hogy minden mesterséges változó legyen nulla, amit szimplex típusú báziscserék sorozatával kívá-

nunk elérni. Ez pedig a következő LP feladat megoldását jelenti:

$$\begin{aligned} \min \quad & z_a = \sum_{i=1}^{m} x_{n+i} \\ \text{f.h.} \quad & \sum_{j=1}^{n} a_j^i x_j + x_{n+i} = b_i, \quad i = 1, \ldots, m, \\ & x_j \geq 0, \quad j = 1, \ldots, n+m. \end{aligned} \tag{3.36}$$

Ez a feladat (3.34)-nek egy bővített változata. Mindkettőnek ugyanannyi (m) feltétele van, de a változók száma (3.36)-ben $n+m$. A legfontosabb különbség a célfüggvényben van, ami most a mesterséges változók összege. Az eredeti célfüggvény itt figyelmen kívül marad.

Ezt a feladatot meg tudjuk oldani a tárgyalt szimplex módszerrel, ha bázisváltozók indexhalmazának $\mathcal{B} = \{n+1, \ldots, n+m\}$-et és így bázisnak $\mathbf{B} = \mathbf{I}$-t választunk.

(3.36) elméleti minimuma nulla, hiszen nemnegatív változók összegének minimalizálásáról van szó. Éppen ezért az algoritmus minden esetben a kibővített feladat egy optimális megoldásának megtalálásával fejeződik be. Az optimális bázist $\mathcal{B}^*$-gal és az optimális célfüggvény értéket z_a^*-val jelöljük.

Ha $z_a^* > 0$, akkor a feltételeket csak úgy lehet kielégíteni, ha egy vagy több mesterséges változó pozitív értéket vesz fel. Ez azt jelenti, hogy ezeket a feltételeket az eredeti változókat megengedett értéken tartva nem lehet kielégíteni, más szóval a feladatnak nincs megengedett megoldása (a feltételek ellentmondóak).

Ha $z_a^* = 0$, akkor az összes mesterséges változó nulla értéken van az alábbi két lehetőség szerint: mesterséges változó (i) nincs a bázisban, (ii) van a bázisban de nulla értéken. Ha az (i) eset teljesül minden mesterséges változóra, akkor a bázis csupa eredeti változóból áll és a $\mathcal{B}^*$ bázis megengedett. Ekkor a mesterséges változókat el lehet felejteni és folytathatjuk a szimplex módszert az eredeti célfüggvénnyel, amely most az eredeti feladatot fogja megoldani.

Ha vannak olyan változók, amelyekre (ii) teljesül, akkor van még egy kis tennivaló. Ezen pozíciók indexhalmazát $\mathcal{D}$-vel jelöljük. Ekkor, miután a $\mathcal{D}$-hez tartozó bázisváltozók nulla értéken vannak, a bázis degenerált. Ha ezeken a pozíciókon tudunk pivot elemet választani, akkor a nulla értéken lévő mesterséges változók kilépnek a bázisból ahová többet nem térnek vissza, lásd (i) eset, és helyüket az eredeti feladat változói foglalják el a bázisban. Ehhez csak az kell, hogy minden $\mathcal{D}$-hez tartozó sorban találjunk egy bázison kívüli q indexhez tartozó $\alpha_q^p \neq 0$ elemet és ez legyen a pivot elem. Az ilyen báziscsere eredményeként a célfüggvény értéke nem változik, hiszen degenerált lépésről van szó (lépéshossz =0).

Egy apró módosítása a fenti lehetőségnek az, hogy elkezdjük a normál második fázisbeli iterációkat, de ügyelünk arra, hogy a 0 értéken lévő mesterséges változók továbbra is 0 értéken maradjanak, vagy lépjenek ki a bázisból. Ez úgy érhető el, hogy a hányados próbánál egy 0 értékű hányadost veszünk figyelembe minden olyan mesterséges változó sorában, ahol az $\alpha_q^i \neq 0$.

A gondolatmenet ígért finomítása a következő. Nem szükséges minden sorhoz hozzárendelni mesterséges változót. Elegendő csak az olyan sorokhoz, amelyeket átszoroztunk -1-gyel, illetve az olyanokhoz, amelyek eredetileg egyenlőség feltételek voltak (így nem rendeltünk hozzájuk logikai változót). A többi sorhoz a már jelenlevő egységmátrix logikai változóit

választhatjuk bázisváltozónak. Így csak annyival növeljük a változók számát, amennyivel feltétlenül szükséges. Természetesen így a célfüggvény is egyszerűbb lesz.

Gondot okozhat az átmenet a mesterséges feladatról az eredetire. Elsősorban az a kérdés, hogy mik lesznek a redukált költségek. Erre egy egyszerű megoldás az, hogy az igazi célfüggvényt a mesterséges feladatban is szerepeltetni kell mégpedig mint egy egyszerű feltételt, amit minden iteráció során ugyanúgy transzformálunk, mint a többi feltételt. Így a mesterséges feladat megoldása után a transzformált célfüggvény együtthatók, vagyis a redukált költségek rendelkezésre állnak.

3.8. Dualitás a lineáris programozásban

A dualitás egy általános elv, aminek a lineáris programozásban is nagy jelentősége van. Legegyszerűbb formájában a következőképpen néz ki.

Minden

$$\begin{aligned} \text{(P1)} \quad & \min \mathbf{c}^T\mathbf{x} \\ & \text{f.h.} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \qquad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

alakú LP feladathoz (amit most primal LP-nek nevezünk, definiálható egy vele szoros kapcsolatban lévő másik LP feladat, amelyet a primál LP duál párjának nevezünk és amelynek alakja

$$\begin{aligned} \text{(D1)} \quad & \max \mathbf{b}^T\mathbf{y} \\ & \text{f.h.} \quad \mathbf{A}^T\mathbf{y} \leq \mathbf{c}, \\ & \qquad \mathbf{y} \geq \mathbf{0} \end{aligned}$$

ahol $\mathbf{A} \in \mathbb{R}^{m\times n}$ és minden vektor kompatibilis dimenziójú.

Könnyen belátható, hogy a duál duálja az eredeti primál feladat, vagyis ez a megfogalmazás teljesen szimmetrikus. Azt is szokás mondani, hogy (P1) és (D1) egy primál-duál párt alkot. A primál változók a duál feltételekkel vannak kapcsolatban, míg a duál változók a primál feladat feltételeihez rendelődnek hozzá. A célfüggvény és a jobboldal vektor szerepet cserélnek és a feltételi mátrix a transzponáltjával szerepel a duálban.

Ha a primál a

$$\begin{aligned} \text{(P2)} \quad & \min \mathbf{c}^T\mathbf{x} \\ & \text{f.h.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \qquad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

alakban van megadva, akkor a duál

$$\begin{aligned} \text{(D2)} \quad & \max \mathbf{b}^T\mathbf{y} \\ & \text{f.h.} \quad \mathbf{A}^T\mathbf{y} \leq \mathbf{c}. \end{aligned}$$

Fontos észrevenni, hogy most a duál változók előjelben nem kötött (szabad) változók. Ezt az alakot le lehet vezetni a (P1)−(D1) kapcsolatból, figyelembe véve, hogy egy $\mathbf{a}^i\mathbf{x} = b_i$ egyenlőség helyettesíthető két egyenlőtlenséggel: $\mathbf{a}^i\mathbf{x} \geq b_i$ és $-\mathbf{a}^i\mathbf{x} \geq -b_i$ továbbá, hogy egy szabad változót fel lehet írni két nemnegatív változó különbségeként.

A formális hasonlóságon túl alapvető matematikai összefüggések vannak a primál és a duál között. Ezek közül bemutatunk néhány fontosat.

Gyenge dualitási tétel Legyen $\mathbf{x}$ egy tetszőleges megengedett megoldása (P2)-nek és $\mathbf{y}$ egy tetszőleges megengedett megoldása (D2)-nek. Ekkor a megfelelő célfüggvények közt az alábbi reláció áll fenn:

$$\mathbf{y}^T\mathbf{b} \leq \mathbf{c}^T\mathbf{x}.$$

Ennek bizonyítása igen egyszerű. Ha $\mathbf{x}$ és $\mathbf{y}$ a megfelelő feladat megengedett megoldásai, akkor $\mathbf{x} \geq \mathbf{0}$ és

$$\mathbf{b} = \mathbf{A}\mathbf{x} \quad \text{és} \quad \mathbf{y}^T\mathbf{A} \leq \mathbf{c}^T.$$

A baloldali egyenlőséget szorozzuk meg $\mathbf{y}^T$-tal, a jobboldali egyenlőtlenséget pedig $\mathbf{x}$-szel. A kettő összehasonlításával azt kapjuk, hogy

$$\mathbf{y}^T\mathbf{b} = \mathbf{y}^T\mathbf{A}\mathbf{x} \quad \text{és} \quad \mathbf{y}^T\mathbf{A}\mathbf{x} \leq \mathbf{c}^T\mathbf{x},$$

ami bizonyítja a tétel állítását. Verbálisan arról van szó, hogy egy tetszőleges duál megengedett megoldáshoz tartozó célfüggvény érték a primál célfüggvényt alulról korlátozza. Ez akkor is igaz, ha a (P1)–(D1) párról van szó. A bizonyítás ez esetben is hasonló és az olvasóra van bízva.

Az **erős dualitási tétel** azt mondja ki, hogy ha a primál-duál pár egyikének van megengedett megoldása és véges optimuma, akkor ugyanez áll a másikra is. Ezen belül, ha $\mathbf{B}$ egy optimális bázis a primálra és $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, akkor a célfüggvény értéke

$$z = \mathbf{c}_B^T\mathbf{x}_B = \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} = \boldsymbol{\pi}^T\mathbf{b}.$$

A duál megoldása $\mathbf{y}^T = \mathbf{c}_B^T\mathbf{B}^{-1} = \boldsymbol{\pi}^T$ és a duál optimum értéke

$$\mathbf{b}^T\mathbf{y} = \mathbf{b}^T\boldsymbol{\pi}$$

ami ugyanaz, mint a primálé.

Egy további tétel azt mondja ki, hogy ha a primál-duál pár egyikének a megoldása nem korlátos, akkor a másiknak nincs megengedett megoldása. Ennek belátása indirekt módon történhet. Tegyük fel, hogy a primál megoldása nem korlátos, a duálnak mégis van egy $\mathbf{y}$ megengedett megoldása. Ekkor, a gyenge dualitás tétel értelmében minden $\mathbf{x}$ primál megengedett megoldásra

$$\mathbf{y}^T\mathbf{b} \leq \mathbf{c}^T\mathbf{x},$$

ami lehetetlen hiszen $\mathbf{c}^T\mathbf{x} \to -\infty$. Tehát a duálnak ebben az esetben nem lehet megengedett megoldása. A másik állítás belátása hasonló módon történik.

A duál megengedettség Tekintsük a (P2)−(D2) párt és tegyük fel, hogy (P2) feladat mátrixa $\mathbf{A} \in \mathbb{R}^{m\times n}$, $m < n$, rangja m. Bevezetjük a (D2) duál logikai változóiból álló $\mathbf{w} = [w_1, \ldots, w_n]^T$ vektort. Így (D2) a következő alakot ölti:

$$\max \mathbf{b}^T\mathbf{y} \tag{3.37}$$

$$\text{f.h.} \quad \mathbf{A}^T\mathbf{y} + \mathbf{w} = \mathbf{c}, \tag{3.38}$$

$$\mathbf{w} \geq \mathbf{0}. \tag{3.39}$$

Legyen $\mathbf{B}$ az $\mathbf{A}$ egy bázisa. Nem kell, hogy primál megengedett legyen. (3.38)-at átrendezve $\mathbf{w}^T = \mathbf{c}^T - \mathbf{y}^T\mathbf{A}$-t kapunk, ami particionált formában

$$\mathbf{w}_\mathcal{B}^T = \mathbf{c}_\mathcal{B}^T - \mathbf{y}^T\mathbf{B}, \tag{3.40}$$

$$\mathbf{w}_\mathcal{R}^T = \mathbf{c}_\mathcal{R}^T - \mathbf{y}^T\mathbf{R}. \tag{3.41}$$

A (3.39) nemnegatívitási (és ebben az esetben a duál megengedettségi) követelmény particionált formában $[\mathbf{w}_\mathcal{B}^T, \mathbf{w}_\mathcal{R}^T]^T \geq \mathbf{0}$. Ha $\mathbf{y}^T = \mathbf{c}_\mathcal{B}^T\mathbf{B}^{-1}$-t választunk, akkor azt kapjuk, hogy

$$\mathbf{w}_\mathcal{B}^T = \mathbf{c}_\mathcal{B}^T - \mathbf{c}_\mathcal{B}^T\mathbf{B}^{-1}\mathbf{B} = \mathbf{0}, \tag{3.42}$$

$$\mathbf{w}_\mathcal{R}^T = \mathbf{c}_\mathcal{R}^T - \mathbf{c}_\mathcal{B}^T\mathbf{B}^{-1}\mathbf{R} = \mathbf{d}_\mathcal{R}^T \geq \mathbf{0}, \tag{3.43}$$

ahol $\mathbf{d}_\mathcal{R}$ jelöli a bázison kívüli primál változók redukált költségeiből álló vektort. Miután (3.42) minden bázis esetén teljesül és $\mathbf{y}$ előjelben nem korlátozott, a $\mathbf{B}$ bázis duál megengedett, ha kielégíti (3.43)-t. Ez viszont nem más, mint a primál optimalitási feltétel.

Összefoglalva A duál megengedettségi feltételek azonosak a primál optimalitási feltételekkel, továbbá a duál logikai változók azonosak a primál redukált költségekkel. Ennek szellemében w_j és d_j ugyanazt jelenti ($j \in \mathcal{N}$).

Következmény Ha egy primál megengedett bázis egyben duál megengedett is, akkor optimális mindkét feladatra.

3.9. A duál szimplex módszer

A duál szimplex módszer (DSM) a fenti ismereteken alapul. Elöljáróban leszögezzük, hogy a DSM is az eredeti feladaton dolgozik, de más szabályok alapján hajtja végre a báziscseréket. Azt is fontos tudni, hogy akár primál, akár duál szimplexszel oldunk meg egy feladatot, minden esetben megkapjuk a primál és a duál feladat megoldását is.

A DSM verbálisan a következőképpen írható le. Tegyük fel, hogy adva van egy $\mathbf{B}$ duál megengedett bázis ($\mathbf{d}_R \geq \mathbf{0}$). Ha ez egyben primál megengedett is ($\mathbf{x}_B \geq \mathbf{0}$), akkor optimális mindkét feladatra. Ellenkező esetben legalább egy primál változó negatív értéken van, mondjuk $x_{Bp} < 0$. Ezt a változót megengedetté változtathatjuk, ha kiléptetjük a bázisból (ezesetben az értéke ugyanis nulla lesz). A helyére belépő változót úgy határozzuk meg, hogy az új bázis duál megengedett maradjon. Ez a duál hányados próba alkalmazásával érhető el. Eközben a

duál célfüggvény is javul (esetleg változatlan marad). Ezt a lépéssorozatot addig ismételjük, amíg a bázis primál megengedetté nem válik.

A duál szimplex módszer algoritmikus lépései

Tekintsük az LP feladatot a $\min z = \mathbf{c}^T\mathbf{x},\ \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$ formában. Tegyük fel, hogy adva van egy $\mathbf{B}$ duál megengedett bázis, vagyis $\mathbf{d}_R \equiv \mathbf{w}_R \geq \mathbf{0}$, valamint $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$.

1. Válasszunk ki egy negatív értékű primál bázisváltozót: x_{Bp}. Ha ilyen nincs, akkor $\mathbf{B}$ optimális, stop.

2. Hajtsuk végre a **duál hányados próbát** $\mathbf{w}_R$-rel és $\boldsymbol{\alpha}^p$-vel (a transzformált tabló p-edik sorával), hogy meghatározzuk a belépő változó q indexét:

$$\theta_D = \frac{w_q}{\alpha_q^p} = \max\left\{\frac{w_j}{\alpha_j^p} : \alpha_j^p < 0,\ j \in R\right\}.$$

   Ha ilyen q nincs, akkor a duál feladat nem korlátos, tehát a primál feladatnak nincs megengedett megoldása, stop.

3. **Transzformáljuk** a megoldást és a tablót. Ez utóbbit a Gauss-Jordan elimináció alapján α_q^p-t használva pivot elemként. Tegyük fel, hogy $\mathbf{w}$ az $\mathbf{A}$ mátrix 0-ik sora, vagyis $a_{0j} = w_j,\ \forall j$. Ez ugyanúgy transzformálódik, mint a mátrix akármelyik másik sora.

   Megoldás: Legyen $\theta_P = x_{Bp}/\alpha_q^p$ ($\equiv$ primál hányados.)

$$\bar{x}_{Bi} = x_{Bi} - \theta_P\alpha_q^i, \quad i \neq p,$$
$$\bar{x}_{Bp} = \theta_P.$$

   Tabló (Gauss-Jordan elimináció): Legyen $r_i = \alpha_q^i/\alpha_q^p,\ i \neq p$,

$$\bar{\alpha}_j^i = \alpha_j^i - r_i\alpha_j^p, \quad i \neq p,\ \forall j,$$
$$\bar{\alpha}_j^p = \alpha_j^p/\alpha_q^p.$$

   Visszatérés az 1. lépésre.

Megjegyzés Az ismertetett duál módszer a **duál második fázis**, ugyanis feltételeztük, hogy ismerünk egy duál megengedett bázist. Egy ilyen bázisnak a megkeresésére szolgál a **duál első fázis**. Ennek tárgyalása azonban túlmutat jelen jegyzet keretein a benne alkalmazott módszer nagyobb bonyolultsága miatt. További részletekért az érdeklődő olvasó figyelmébe ajánljuk [2]-t.

Példa 3 *Duál szimplex módszer.*

Tekintsük a következő (primál) LP feladatot:

$$\begin{array}{lrcrcrcrcr}
\min & 2x_1 & + & x_2 & + & 4x_3 & + & 2x_4 & & \\
f.h. & x_1 & - & 2x_2 & + & x_3 & + & x_4 & \leq & 3 \\
 & -x_1 & + & 2x_2 & - & x_3 & & & \leq & -1 \\
 & -x_1 & & & + & 4x_3 & - & 2x_4 & \leq & -2 \\
 & & & & x_j \geq 0,\ j = 1,\ldots,4 & & & & &
\end{array}$$

Miután minden feltétel `≤ típusú, a sorokhoz hozzáadott logikai változók 2-es (nemnegatív) típusúak. Ezeket most s_1, s_2, s_3-mal jelöljük. Ha a logikai változókhoz tartozó egységmátrixot választjuk induló bázisnak, akkor a hozzátartozó duál megoldás megengedett lesz, de a primál változók közt lesz negatív. Ennek következtében alkalmazhatjuk a duál szimplex algoritmust.

A példában a célfüggvényt nem a nulladik sorba írjuk, hanem a mátrix utolsó sora után. A tabló felesleges ismétlésének elkerülése érdekében minden tablóban bejelöltük a következő lépés pivot elemét. Amikor egy tablóra először nézünk rá, a bekeretezést ideiglenesen figyelmen kívül kell hagyni.

Az induló tabló:

B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	$\mathbf{x}_B$
s_1	1	−2	1	1	1			3
s_2	−1	2	−1	0		1		−1
s_3	−1	0	4	**−2**			1	−2
w	2	1	4	2	0	0	0	0

Most két primál változó (s_2 és s_3) is negatív, bármelyiket választhatjuk kilépőnek. Legyen ez s_3, vagyis a harmadik sor a pivot sor (miután s_3 a harmadik sor bázisváltozója), $p = 3$. A duál hányados próbában résztvevő pozíciók azok, amelyek negatív hányadost eredményeznek, vagyis az 1 és 4 oszlop, melyekkel a $2/(-1) = -2$ és $2/(-2) = -1$ hányadosok adódnak. Ezek maximuma -1 (negatív számokról van szó!), ami a 4. pozícióban vétetik fel. Ennek következtében x_4 fog belépni a bázisba a 3. pozíción és a pivot elem $\alpha_4^3 = -2$. Ha ezzel az elemmel végrehajtjuk a Gauss-Jordan eliminációt (ami egyébként a duál algoritmus 3. lépése) a következő tablót kapjuk:

B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	$\mathbf{x}_B$
s_1	$\frac{1}{2}$	−2	3	0	1		$\frac{1}{2}$	2
s_2	**−1**	2	−1	0		1	0	−1
x_4	$\frac{1}{2}$	0	−2	1	0	0	$-\frac{1}{2}$	1
w	1	1	8	0	0	0	1	−2

Látható, hogy a megoldás duál megengedett maradt. Most viszont már csak egy primál változó negatív ($s_2 < 0$), így ezt választjuk kilépőnek, pivot sor: $p = 2$. A duál hányados próbában az 1 és 3 oszlopok elemei vesznek részt és -1 illetve -8 hányadost adnak. Miután ezek maximuma -1, a pivot elem $\alpha_1^2 = -1$ és x_1 lép be a bázisba a 2. pozíción. A transzformált tabló:

B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	$\mathbf{x}_B$
s_1	0	−1	$\frac{5}{2}$	0	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$
x_1	1	−2	1	0		−1	0	1
x_4	0	1	$-\frac{5}{2}$	1	0	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$
w	0	3	7	0	0	1	1	−3

Most a megoldás egyaránt duál és primál megengedett, vagyis optimális. A primál strukturális változók értékei: $x_1 = x_{B2} = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = x_{B3} = \frac{1}{2}$, a logikaiaké $s_1 = x_{B1} = \frac{3}{2}$, $s_2 = s_3 = 0$. A duál megoldás definíció szerint $\mathbf{y}^T = \mathbf{c}_B^T \mathbf{B}^{-1} = [0, -1, -1]$. Ez utóbbi nem egyéb, mint

a logikai változókhoz tartozó redukált költségek -1*-szerese. Az optimális célfüggvény érték úgy a duálra mint a primálra* $= 3$.

Megjegyzés A teljes tabló transzformációs változat tartalmazza a mindenkori bázis inverzét is. A transzformált tabló ugyanis nem más, mint $\tilde{\mathbf{A}} = \mathbf{B}^{-1}\mathbf{A}$. Miután $\mathbf{A}$ tartalmazza a megfelelő dimenziójú egységmátrixot, ezért particionált alakban $\mathbf{A} = [\bar{\mathbf{A}} \mid \mathbf{I}]$. Ha ezt beszorozzuk $\mathbf{B}^{-1}$-zel akkor a transzformált tabló utolsó m oszlopa valóban a mindenkori bázis inverzét adja. Ezt a fenti példán könnyen azonosítani is lehet.

3.10. Belsőpontos algoritmusok

Az LP feladat megoldására a szimplex módszeren kívül létezik egy másik algoritmus család, a belsőpontos (BP) algoritmusok. Ezek alapvető módon különböznek a szimplextől. Míg a szimplex a megengedett tartomány határán (a konvex poliéder csúcsain és élein) halad, addig a BP algoritmusok a tartomány belsejében haladva érnek el egy optimális megoldást. A kidolgozott módszertan nemlineáris technikákat használ és minden iterációban egy nemlineáris egyenletrendszert old meg a Newton módszerrel.

A gyakorlati feladatok megoldása során a nagy lépéses primál-duál logaritmikus barrier módszer bizonyult a leghatékonyabbnak. Ennek részleteiről értékes információ található az [1] jegyzetben.

Érdekes tapasztalat, hogy a feladat méretétől függetlenül ez a módszer 40–80 iteráció alatt talál egy optimális megoldást. Természetesen az egy iterációra eső számítási munka a feladat méretével együtt jelentősen növekszik.

A BP algoritmusok nagyon nagy méretű feladatok esetén általában gyorsabbak, mint a szimplex. A gyakorlatban azonban mindkét módszerre szükség van, hiszen, mint látni fogjuk a kevert egészértékű feladatok megoldása során a szimplex nélkülözhetetlen.

4. fejezet

Kevert egészértékű optimalizálás

Ha egy lineáris programozási (LP) feladatban néhány változó csak egész értéket vehet fel akkor **kevert egészértékű LP feladat**ról beszélünk, amire az angol *Mixed Integer Linear Programming* elnevezés alapján MILP rövidítéssel szokás hivatkozni. Ha ez az összes döntési változóra elő van írva, akkor a feladat neve egészértékű LP feladat (Integer Linear Programming: ILP). Ilyenkor néha a hangsúly kedvéért ki lehet tenni, hogy *tiszta egészértékű LP*.

A MILP feladat általános alakja:

$$\min \ \mathbf{c}^T\mathbf{x} + \mathbf{h}^T\mathbf{y} \tag{4.1}$$

$$\text{f.h.} \quad \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b} \tag{4.2}$$

$$\mathbf{x},\ \mathbf{y} \geq \mathbf{0} \tag{4.3}$$

$$\mathbf{x} \text{ egészértékű } \ (\mathbf{x} \in \mathbb{Z}^n). \tag{4.4}$$

Igen gyakran ésszerű feltenni, hogy az egészértékű változóknak véges felső korlátjuk van, vagyis $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$.

4.1. MILP alkalmazások

Érdekes módon az ILP egy igen általános modellezési eszköz és gyakorlati jelentősége (értéke) messze túlmutat azon, amit az eredeti megfogalmazás sejtet. Eredetileg ugyanis az ILP-et olyan esetekben használták, amikor a termelési döntési változóknak értelemszerűen egésznek kellett lenni, mint például egész számú repülőgép, vagy ház, vagy éppen nukleáris erőmű. Ma már azonban ennél szélesebb alkalmazási kört figyelhetünk meg. Néhány példa az alábbi.

1. Tőke befektetés: választás befektetési alternatívák közt figyelembe véve a rendelkezésre álló pénz mennyiségét és a forgótőke igényt az adott időszakokban.

2. Kapacitásbővítés: termelési kapacitás bővítése diszkrét lépcsőkben (pl. új gépek beállításával) az igény és a többlet költségek figyelembevételével.

3. Termeléstervezés logikai feltételekkel, pl. ha egy bizonyos összetevő benne van a keverékben, akkor egy másiknak is benne kell (vagy éppen nem szabad benne) lenni a keverékben.

4. Hátizsák probléma: Adva van egy tartály (pl. hátizsák) ismert súly, vagy térfogat kapacitással, amit meg kell tölteni bizonyos árukkal, úgy hogy a tárolt cikkek összértéke maximális legyen. Ez a típusú feladat gyakran felmerül más feladatok részeként is.

5. Különféle ütemezési feladatok: munkák sorrendjének meghatározása a megmunkáló gépeken.

6. Elosztási és hozzárendelési problémák.

7. Repülőgép és személyzet járatokba történő beosztása úgy, hogy a kötelező pihenőidő be legyen tartva.

8. Fix költségű problémák: tervezési tevékenység olyan esetben, amikor a beindulásnak van egy egyszeri nagyobb költségigénye.

9. Hálózati optimalizálási problémák: egy hálózatban meg kell határozni azt a minimális költségű folyamot, amellyel egy bizonyos terméket (pl. víz, elektromosság) el lehet juttatni minden igénylőhöz. Ilyen feladatra vezethető vissza még a szállítási, átrakodásos szállítási (transshipment) és hozzárendelési probléma, valamint a maximum folyam és a legrövidebb út probléma is.

10. Utazó ügynök probléma: Egy körúttal meg kell látogatni m várost úgy, hogy az utazás összköltsége (ideje, távolsága) minimális legyen.

11. Problémák diszkrét (nem feltétlenül egész) változókkal.

12. Problémák diszjunktív feltételekkel, pl. vagy egyik, vagy egy másik feltételrendszert kell kielégíteni a megoldásnak (lehet mindkettőt).

13. És sok egyéb …

4.2. MILP problémák megoldása

A MILP problémák megoldása általában sokkal nehezebb, mint az LP problémáké. Ha csak néhány tucat egészértékű változó van jelen a feladatban a megoldó program akár napokig is futhat a sikeres befejezésig.

A megoldó algoritmusok többnyire úgy működnek, hogy először eltekintenek az (4.4) egészértékűségi követelménytől és megoldják a MILP feladat ún. LP relaxációját:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \mathbf{h}^T\mathbf{y} && (4.5)\\ \text{f.h.} \quad & \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b} && (4.6)\\ & \mathbf{x},\ \mathbf{y} \geq \mathbf{0} && (4.7) \end{aligned}$$

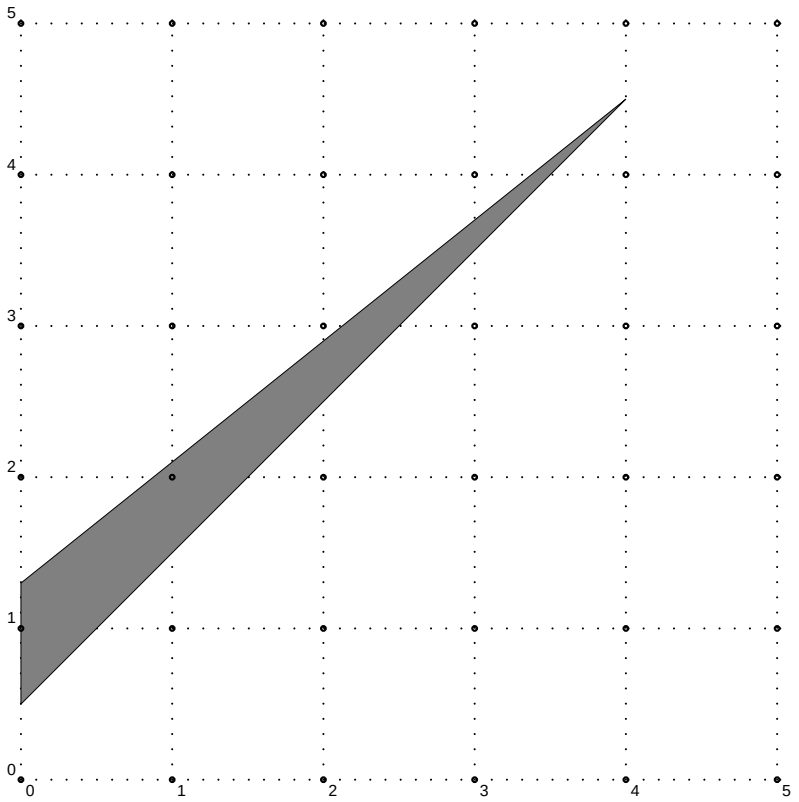
Ha az LP relaxáció megoldásában minden egész változó egész értéket vesz fel, akkor ez optimális megoldás az eredeti MILP problémára is. Bár azt hihetnénk, hogy ez egy ritka szerencsés eset, a valóság azonban más. Vannak olyan feladat típusok, amelyek matematikai tulajdonságai garantálják, hogy ez így lesz, amint ezt a későbbiekben látni fogjuk.

Ha nem az összes egészértékű változó vesz fel egész értéket, akkor a következő kézenfekvő gondolat az, hogy kerekítsük a tört értékeket a legközelebbi egészre. Ez elfogadható lehet, ha a megoldásban szereplő értékek elég nagyok . Ellenkező esetben a kerekítéssel elfogadhatatlan nem-megengedett megoldást kaphatunk, amint azt a következő példa mutatja.

Példa 4 *(H.P. Williams példája) :*

$$\begin{array}{lrcrcr} \max & x_1 & + & x_2 & & \\ f.h. & -2x_1 & + & 2x_2 & \geq & 1 \\ & -8x_1 & + & 10x_2 & \leq & 13 \\ & & & x_1,\, x_2 & \geq & 0 \\ & & & x_1,\, x_2 & \in & \mathbb{Z}. \end{array}$$

Az LP relaxáció optimális megoldása $x_1 = 4$ *és* $x_2 = 4.5$. *Ha* x_2*-t a legközelebbi egészre (4 vagy 5) kerekítjük, akkor nem-megengedett megoldást kapunk. A valódi egész megoldás* $x_1 = 1$ *és* $x_2 = 2$, *ami igen messze van az LP relaxált megoldástól. Az alábbi ábra jól érzékelteti a helyzetet.*



A MILP feladatok megoldására nincs olyan általános megoldó algoritmus, mint az LP-re a szimplex módszer. A leggyakrabban használt algoritmusok a **korlátozás és szétválasztás** (angolul : Branch-and-Bound, rövidítve : B&B) elve alapján működnek.

A B&B az egészértékű változók összes kombinációját megvizsgálja explicit vagy implicit módon és azonosítja a legjobbat. Az eljárás a megoldandó részfeladatokat egy **fa-struktúrá**val

reprezentálja és ennek a fának a bejárását végzi el. A model ismeretében a bejárás hatékonysága kedvezően befolyásolható oly módon, hogy a nem optimális, illetve nem-megengedett ágakat hamar azonosítani lehet és a keresést azokban az irányokban meg lehet szüntetni. Ezáltal a keresési tér nagy mértékben csökkenthető.

A gyakorlatban sok olyan ILP modell van, ahol a változók csak két értéket vehetnek fel: 0-t vagy 1-et. Az ilyen változókat **bináris változó**knak, a feladatokat pedig **0–1** (vagy **0/1**) **programozási feladat**oknak nevezzük.

A bináris változók általában „igen” (1), vagy „nem” (0) döntéseket jelentenek. De lehet velük változók vagy feltételek közti logikai kapcsolatokat is kifejezni. Ilyen értelemben a bináris változók nélkülözhetetlen modellezési eszközök.

A 0/1-es MILP feladat LP relaxációja

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \mathbf{h}^T\mathbf{y} \\ \text{f.h.} \quad & \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} = \mathbf{b} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \quad \mathbf{y} \ge \mathbf{0}, \end{aligned}$$

ahol $\mathbf{1}$ a csupa egyesekből álló vektor $\mathbf{1} = [1, \ldots, 1]^T$.

A gyakorlatban nagyon sok MILP feladatban bináris változók vannak. Hovatovább az is igaz, hogy majdnem minden értelmes MILP feladat transzformálható 0/1-es feladattá. Ez azonban nagyon fel tudja szaporítani a változók számát, ha nem ismerünk egy alacsony felső korlátot a változókra. Ezért ezt az eljárást a gyakorlatban általában nem célszerű használni.

4.2.1. Korlátozás és szétválasztás módszer

Az egyszerűség kedvéért tegyük fel, hogy egy olyan MILP feladatot akarunk megoldani, ahol minden egész változónak véges felső korlátja van. Ezt az alapfeladatot P_0-lal jelöljük. Először megoldjuk ennek LP relaxációját, LP(P_0)-t. Megjegyzendő, hogy miután LP(P_0) kevésbé van korlátozva, mint P_0, ezért LP(P_0) optimuma jobb (de legalább is nem rosszabb), mint P_0 optimuma.

Ha LP(P_0) optimális megoldásában $\mathbf{x}$ egy vagy több komponense nem egész értéken van (noha egész változó), kiválasztunk egy ilyen változót, mondjuk x_j-t és ennek segítségével az eredeti feladatot két részfeladatra, P_1-re és P_2-re bontjuk. Tegyük fel, hogy az LP(P_0) optimális megoldásában x_j értéke $\beta_j > 0$ nem egész érték. P_1 és P_2 származtatása:

$$P_1 := P_0 \ \text{ és } \ x_j \le \lfloor \beta_j \rfloor \tag{4.8}$$

$$P_2 := P_0 \ \text{ és } \ x_j \ge \lfloor \beta_j \rfloor + 1, \tag{4.9}$$

ahol $\lfloor \beta_j \rfloor$ a legnagyobb egész számot jelenti, ami β_j-nél még nem nagyobb. (Például, $\lfloor 4.99 \rfloor = 4$ és $\lfloor -4.99 \rfloor = -5$.) Nyilvánvaló, hogy P_0 bármely megoldása vagy P_1-nek vagy P_2-nek a megoldása.

A P_1 és P_2 definiálása során a P_0-hoz hozzávett feltételek egyedi feltételek, amiket a szimplex módszerben algoritmikusan lehet kezelni, nem pedig explicit feltételként.

Vegyük észre, hogy ha x_j bináris változó, akkor (4.8) és (4.9)-ben az új feltétel hozzáadása arra redukálódik, hogy rögzítjük x_j-t a 0 vagy 1 értéken.

Ezután megoldjuk LP(P_1)-t vagy LP(P_2)-t. Tegyük fel, P_1-et választottunk. LP(P_1) megoldását ugyanúgy értékeljük ki, mint LP(P_0) megoldását. Ha **x** még mindig nem egész, akkor P_1-et hasonló szellemben felbontjuk P_3-ra és P_4-re. És így tovább …

Ezzel az eljárással egy, az alproblémákat tartalmazó **bináris fa** épül fel, amelynek a levelei a függőben lévő (várakozó) csúcsok. Ez utóbbiak azok, amelyeket még meg kell oldani és ki kell értékelni.

Egy bizonyos szinten egy alprobléma megoldása egész lesz. Ez lehet is meg nem is az eredeti P_0 probléma optimális megoldása. Ha van még várakozó csúcs, akkor az összeset meg kell vizsgálni, mert lehet köztük olyan, amelyik jobb egész megoldást ad. Egy ismert egész megoldásnak, Z célfüggvény értékkel, azonban óriási jelentősége van. Segítségével drámai módon lehet csökkenteni az új csúcsok generálását. Ugyanis,

ha egy csúcshoz tartozó LP z optimumára az teljesül, hogy $z > Z$, akkor ennek az LP-nek és a belőle származtatott, még jobban korlátozott LP-knek sem lehet jobb az optimuma, ezért az egész ág, ami ebből a csúcsból ered, kizárható a további vizsgálatokból.

4.2.2. Korlátozás és szétválasztás (B&B) lépései

A B&B algoritmus formális leírásához szükség van néhány jelölésre. Z jelenti az algoritmus során az eddigi legjobb egész megoldáshoz tartozó célfüggvény értéket, mig $\mathcal{W}$ jelöli a várakozó csúcsokban lévő alfeladatok halmazát.

0. lépés: Oldjuk meg LP(P_0)-t. Ha **x** egészértékű, akkor egyben optimális megoldás P_0 számára is, az eljárás befejeződik.

   Különben, végezzük el a kezdeti beállításokat: $Z = +\infty$, $k = 0$, $\mathcal{W} := \{P_0\}$ és jelöljük LP(P_0) optimális célfüggvény értékét z_0-lal.

1. lépés: Válasszunk ki egy P_t alfeladatot $\mathcal{W}$-ből úgy, hogy $z_t < Z$, ha van ilyen. ellenkező esetben az eljárás befejeződik. Az eddig talált legjobb közbülső megoldás optimális. Ha ilyen megoldás eddig nem volt, akkor a feladatnak nincs megengedett egész megoldása.

2. lépés: LP(P_t) optimális megoldásában van olyan változó, ami tört értéket vesz fel noha egésznek kellene lennie. Válasszunk ki egy ilyet, mondjuk x_j-t, aminek az értéke β_j.

   Definiáljunk két alproblémát, P_{k+1}-et és P_{k+2}-t (utód csúcsok): az egyiknél P_t-hez hozzáadjuk az $x_j \le \lfloor \beta_j \rfloor$ feltételt, a másiknál pedig az $x_j \ge \lfloor \beta_j \rfloor + 1$ feltételt.

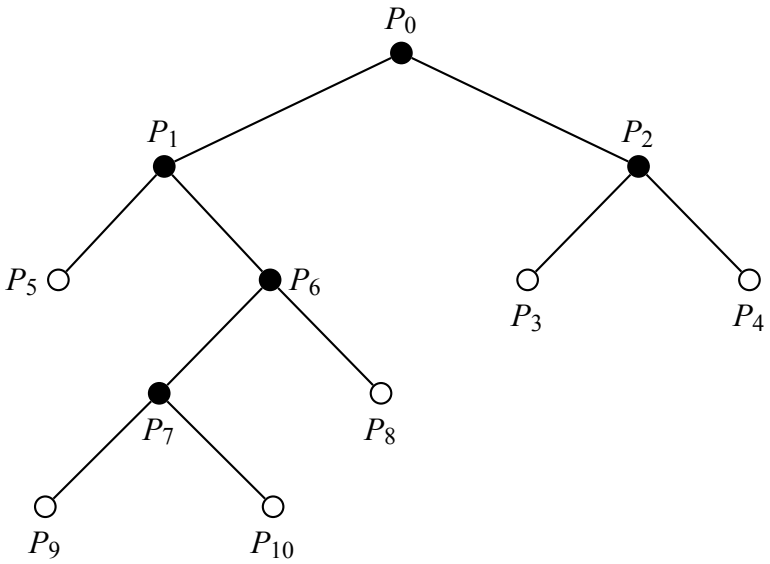
3. lépés: Oldjuk meg az LP(P_{k+1}) és LP(P_{k+2}) LP relaxációkat.

   Ha találunk egy új egész megoldást, ami jobb az eddigi legjobbnál, akkor ezt eltároljuk és a hozzátartozó célfüggvény érték lesz Z új értéke.

   Ha $z_{k+1} > Z$ vagy $z_{k+2} > Z$ teljesül, a megfelelő alfeladatot törölhetjük az összes lehetséges utódjával együtt. Ugyanígy járunk el, ha LP(P_{k+1})-nek vagy LP(P_{k+2})-nek nincs megengedett megoldása.

Adjuk P_{k+1}-et és/vagy P_{k+2}-t $\mathcal{W}$-hez ha a megfelelő alfeladatnak van megengedett megoldása és az LP optimum (z_{k+1} és/vagy z_{k+2}) kisebb, mint Z.

Növeljük meg a k számlálót annyival, amennyi (0, 1, vagy 2) új csúcsot adtunk $\mathcal{W}$-hez és térjünk vissza az 1. lépéshez.



Fontos észrevenni, hogy az újonnan keletkező alproblémák erősebben korlátozottak, mint az elődeik, így ezen problémák LP relaxációinak optimális megoldása rosszabb (nem jobb) értéket szolgáltat.

Miután az egész változók korlátosak, az algoritmus végül befejeződik, mert ahogy megyünk lefelé a fán úgy az egész változók korlátai egyre szorosabbá válnak, míg végül fix egész értékre szűkülnek.

Fontos megjegyezni, hogy az újonnan keletkező alproblémákat nem kell teljesen elölről kezdve megoldani. A közvetlen előd feladat ismert optimális bázisa ugyanis duál megengedett bázis a származtatott problémákra. Így erről a bázisról indulva a duál algoritmus használható és általában nagyon hamar be is fejeződik. Ennek oka az, hogy az alfeladatok csak nagyon picit térnek el az elődtől, ezért jó esély van arra, hogy az optimális megoldások is közel lesznek egymáshoz. A primál szimplex erre nem lenne képes, mert az említett bázis nem primál megengedett, így primál első fázisra lenne szükség, aminek során viszont az igazi célfüggvény nagyon el tud romlani. Tapasztalat szerint a duál tized vagy század annyi iterációval találja meg az alfeladat optimumát.

4.2.3. Néhány megjegyzés a B&B-ről

A korlátozás és szétválasztás módszerében az egyes lépéseknél van bizonyos szabadsági fok, ahol több lehetőségből választhatunk. Ezt a rugalmasságot ki lehet használni az algoritmus finom hangolására. Az, hogy mit választunk a lehetőségek közül, alapvetően befolyásolja az algoritmus teljesítményét (sebességét). Sajnos, nem ismert olyan stratégia, ami mindig a legjobb lenne. Azt azonban jó tudni, hogy a jó stratégia feladat függő és még egy feladat megoldása során is változhat. Néhány lehetőség az alábbi.

Választás a várakozó csúcsok közül Néhány sikeres stratégia: (i) az a csúcs, melyhez tartozó célfüggvény érték a legjobb, (ii) először az azonos szinten lévő csúcsok vizsgálata, vagy (iii) a legmélyebben lévő csúcsból még mélyebbre menni.

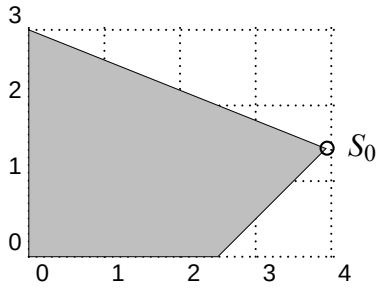
Elágaztató változó kiválasztása Ez még több variációs lehetőséget kínál, pl. egy tört értékű változó (i) a legjobb célfüggvény együtthatóval, (ii) amelyik nagyjából félúton van két egész érték közt, (iii) amelyet valamilyen hasznossági függvény „ajánl", (iv) amelyik egy „fontos változó" a modellben (pl. 0/1 típusú döntés). Többnyire ez az utolsó a leghatékonyabb, de ez feltételezi a modell ismeretét. Ennélfogva, a modell tulajdonosának vannak a legjobb esélyei, hogy egy MILP feladatot hatékonyan meg tudjon oldani. Ez általában úgy történik, hogy megadunk egy listát, ahol a változók a fontosságuk sorrendjében vannak felsorolva (prioritási lista) és az algoritmust felkészítjük ennek a listának a felhasználására.

4.2.4. Példa a B&B algoritmusra

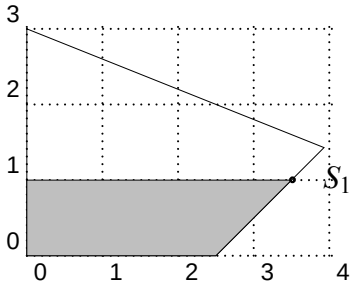
Tekintsük a következő feladatot.

$$\begin{aligned} \min z = \quad & -3x_1 \quad - \quad 4x_2 \quad + \quad 20 \\ \text{s.t.} \quad & \frac{2}{5}x_1 \quad + \quad x_2 \quad \leq \quad 3 \\ & \frac{2}{5}x_1 \quad - \quad \frac{2}{5}x_2 \quad \leq \quad 1 \\ & x_1, x_2 \geq 0 \text{ és egészértékű.} \end{aligned}$$

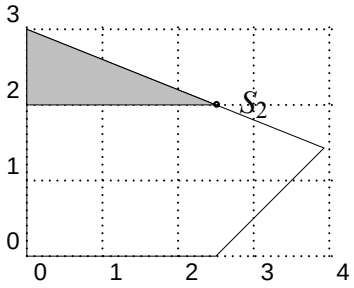
Ezt a feladatot jelöljük P_0-lal. Ennek az LP relaxációját (LP(P_0)) megoldjuk (satírozott rész a megengedett megoldások halmaza):



Optimális megoldás S_0, ahol $x_1 = 3\frac{13}{14}$, $x_2 = 1\frac{3}{7}$ és $z_0 = 2\frac{1}{2}$, vagyis mindkét változó tört értéket vesz fel. Kiválasztjuk x_2-t elágaztató változónak és a definiálunk két alfeladatot: $P_1 := P_0$ és $x_2 \leq 1$; $P_2 := P_0$ és $x_2 \geq 2$. Először megvizsgáljuk P_1-et.

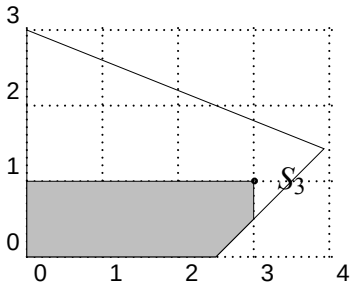


Optimális megoldás S_1, ahol $x_1 = 3\frac{1}{2},\ x_2 = 1$ és $z_1 = 5\frac{1}{2}$. Ez várakozó csúcs lesz, mert x_1 nem egész értékű. Várakozó (függőben lévő) csúcsok: $\mathcal{W} = \{P_1\}$.
Most megvizsgáljuk P_2-t, aminek definiciója: $P_2 := P_0\ \ \&\ \ x_2 \geq 2$.



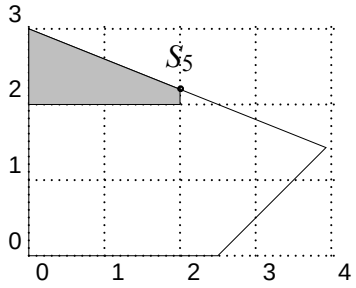
Optimális megoldás S_2, ahol $x_1 = 2\frac{1}{2},\ x_2 = 2$ és $z_2 = 4\frac{1}{2}$. Miután x_1 nem egész értékű, P_2 is felkerül a várakozó csúcsok listájára, $\mathcal{W} = \{P_1, P_2\}$.
Most kiválasztjuk P_1-et és ezen belül elágaztató változónak x_1-et. Ezáltal definiálunk két alproblémát, amelyek helyettesítik P_1-et: $P_3 := P_1$ és $x_1 \leq 3$ valamint $P_4 := P_1$ és $x_1 \geq 4$.

Először a $P_3 := P_1\ \ \&\ \ x_1 \leq 3$ feladatot vizsgáljuk meg.

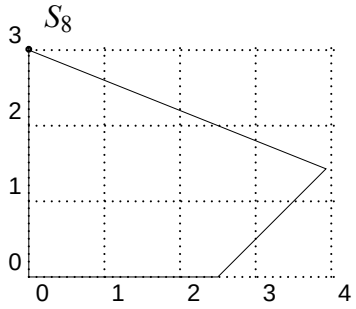


Optimális megoldás S_3, ahol $x_1 = 3,\ x_2 = 1$ és $z_3 = 7$. Ez egy **egész megoldás** és $Z = z_3 = 7$.

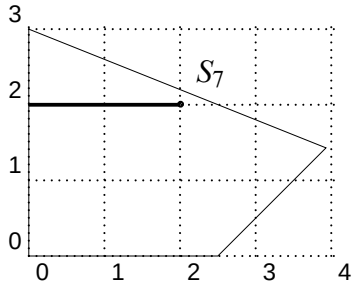
Most a $P_4 := P_1\ \ \&\ \ x_1 \geq 4$ feladatot vizsgáljuk meg. Rögtön látható, hogy P_4-nek **nincs megengedett megoldás**a, hiszen x_1 nem lehet nagyobb, mint 3.93 (lásd LP(P_0)). Így egyetlen függő csúcs marad: $\mathcal{W} = \{P_2\}$.
Vegyük elő P_2-t és ágaztassunk x_1-en. A keletkező két alfeladat, amelyek helyettesítik P_2-t: $P_5 := P_2$ és $x_1 \leq 2$, valamint $P_6 := P_2$ és $x_1 \geq 3$.
Tekintsük először $P_5 := P_2\ \ \&\ \ x_1 \leq 2$-t:



Optimális megoldás S_5, ahol $x_1 = 2,\ x_2 = 2\frac{1}{5}$ és $z_5 = 5\frac{1}{5}$. Ez nem egész megoldás, mert x_2 tört értéken van, így P_5 várakozó csúcs lesz, $\mathcal{W} = \{P_5\}$.
Ha vesszük $P_6 := P_2\ \&\ x_1 \geq 3$-t akkor a korábbi ábrákon látható, hogy x_1 nem lehet nagyobb, mint 3 ha $x_2 \geq 2$. Ebből kifolyólag P_6-nak nincs megengedett megoldása.
P_5 marad az egyetlen választásunk. Ha elágaztatunk x_2-n, a keletkező két alfeladat: $P_7 := P_5$ és $x_2 \leq 2$; $P_8 := P_5$ és $x_2 \geq 3$.
Vizsgáljuk előbb $P_8 := P_5\ \&\ x_2 \geq 3$-t



S_8-ban ennek egyetlen megengedett megoldása van, ami egyben optimális is erre a feladatra nézve. Itt $x_1 = 0,\ x_2 = 3$ és $z_8 = 8$. Ez ugyan egész megoldás, de a célfüggvény érték, 8, ami rosszabb, mint az eddigi legjobb, 7. Így Z nem változik.
A maradék csúcs, amit még meg kell vizsgálni $P_7 := P_5$ és $x_2 \leq 2$.



A megengedett megoldások halmaza a vastagított egyenes szakasz. Az optimális megoldás S_7, ahol $x_1 = 2,\ x_2 = 2$; és $z_7 = 6$. Ez egy egész megoldás és a célfüggvény értéke 6, ami jobb az eddigi legjobbnál. Ezért $Z := z_7 = 6$.
Miután nincs több függő csúcs, az eljárás befejeződik. Az **optimális megoldás**: $x_1 = 2,\ x_2 = 2$; és $Z = 6$.

4.3. MILP algoritmusok képességei

Jelenleg (2010 december) nem ismeretes egy „legjobb”, általános célú MILP megoldó. A legsikeresebb megoldó algoritmusok a korlátozás és szétválasztás (B&B) elve alapján működnek. Az első LP relaxációt meg lehet oldani a primál vagy duál szimplexszel (esetleg az itt nem tárgyalt belsőpontos módszerekkel). A generált alfeladatok megoldása a duál szimplexszel történik.

A MILP megoldók érzékenyek az egész értékű változók számára. Például, ha van egy modell, amely tartalmaz 100 darab 0/1 változót, akkor az ezekkel képezhető összes kombinációk száma $2^{100} \approx 10^{30}$. Egy B&B módszer ezeket mind ellenőrzi (a legtöbbjüket implicit módon). Attól függően, hogy milyen elágaztatási stratégiát használunk, a számítási munka nagyon különböző lehet. Egy jó stratégia megtalálásának lehetősége a modell tulajdonos kezében van, aki tudja, hogy melyek azok a fontos változók, amelyeken érdemes ágaztatni.

A modellek megfogalmazása is óriási hatással lehet a megoldás hatékonyságára. Az ún. szorosan megfogalmazott feladatokat könnyebb megoldani, mint a kevésbé szorosakat. Bizonyos algoritmikus technikák képesek egy megadott feladatot szorosabb alakra hozni, de így is a modellezőé a főszerep.

Nagyon kevés általános célú MILP megoldó van (kommerciális, illetve kutatói). Ezek többnyire egy kiválóan elkészített LP megoldó körül jöttek létre. A duál szimplex megléte kötelező az alfeladatok hatékony megoldása céljából, mert a számítási idő nagy része ($\approx 95\%$) ezzel telik el.

Néhány száz bináris (és sok folytonos) változóból és több tízezer feltételből álló MILP feladatok megoldása reális feladat, ha a modell jól van megfogalmazva. A megoldási idők nagyon nagy mértékben szóródnak, még az is előfordulhat, hogy belátható időn belül nem kapunk megoldást. A jól képzett modellezőknek azonban jó esélye van a valóságban előforduló MILP feladatok megoldására.

5. fejezet

Hálózati optimalizálás

A hálózati modellek igen népszerűek az operációkutatásban. Ennek több oka is van, melyek közül a legfontosabbak az alábbiak:

- Sok gyakorlati döntési probléma jól modellezhető hálózatokkal.
- A hálózatokat jól lehet vizuálisan megjeleníteni és a kapott eredményeket könnyű értelmezni.
- Rendkívül hatékony megoldó algoritmusok léteznek hálózati optimalizálási feladatok megoldására.

A hálózatokban bizonyos pontok közt folyam valósul meg. Ez lehet konkrét fizikai, de akár logikai folyam is. Néhány példa:

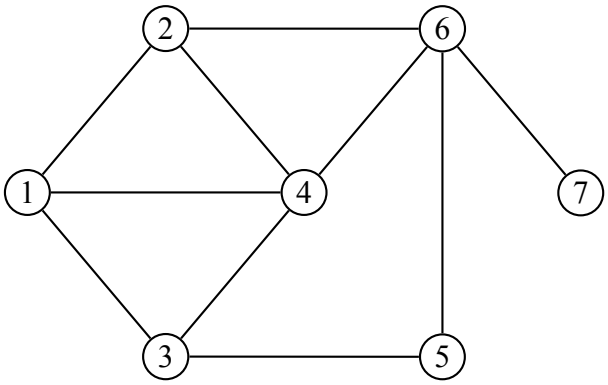
Fizikai folyam:

- elektromos áram,
- kőolaj, gáz, víz,
- információ,
- árucikkek, járművek, utasok,
- pénz,
- …

Logikai folyam:

- Hozzárendelés.
- Hálózaton belül úthosszúság meghatározása.
- …

A legtöbb hálózati folyam probléma egyszerűen csak LP feladat, így az LP algoritmusok használhatók a megoldásra. Ezek azonban speciális szerkezetű LP feladatok, amelyek megoldására a szimplex módszer erre a típusra adaptált változata 50–200-szor gyorsabb, mintha



5.1. ábra. Egy irányítatlan gráf 7 csúcsponttal és 10 éllel.

a módszert eredeti formájában használnánk. Ezen túlmenően, a hálózati szimplexnek további kedvező tulajdonságai is vannak.

Mik a forrásai a hatékonyság ilyen mértékű növekedésének? Erre a kérdésre megint csak egy listás felsorolással lehet legtömörebben válaszolni:

- algoritmikus fejlesztések a probléma speciális struktúrájának a kihasználására,
- A számítástudomány eredményeinek a használata, úgymint:
  - korszerű adatstruktúrák,
  - hatékony keresési, rendező és összeválogató algoritmusok,
  - korszerű fa-bejáró és listakezelő algoritmusok.

5.1. Gráfelméleti alapfogalmak

A **hálózat**okat **gráf**okkal lehet a legjobban reprezentálni. A továbbiak jobb megértése érdekében néhány gráfelméleti alapfogalom összefoglalójával kezdjük a tárgyalást.

Gráfok körében két fő típust különböztetünk meg: *irányítatlan* és *irányított* gráfok.

Irányítatlan gráf Csúcsok (csomópontok) és élek halmaza, $G = (N, E)$, ahol N jelöli a csúcsok (angolul *nodes*) és E az élek (angolul *edges*) halmazát.

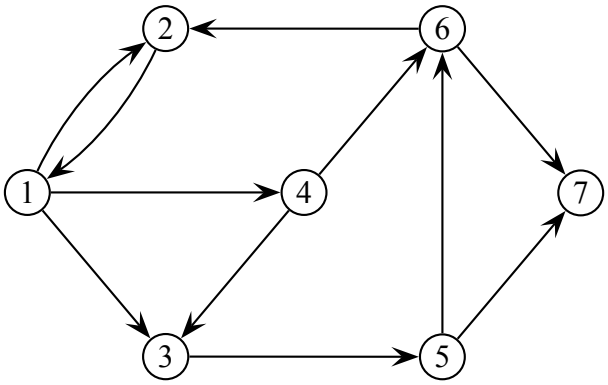
Az éleket egy (i, j) számpárral jelöljük, és azt mondjuk, hogy ez az i és j ($i \neq j$) csúcsokat összekötő irányítatlan él. Az (i, i) típusú, önmagába visszatérő él (self-loop) nincs megengedve.

Egy példa az irányítatlan gráfra a 5.1 ábrán látható.

A továbbiakban még a következő fogalmak játszanak szerepet a hálózati optimalizálásban.

Út: egymástól különböző csúcsok $i_1, \ldots, i_q$ sorozata úgy, hogy minden (i_k, i_{k+1}) egy létező él E-ben, $k = 1, \ldots, q-1$.

Kör: egy olyan út, amelynek kezdő és vég csúcsa ugyanaz a csúcs, $i_1 = i_q$.



5.2. ábra. Egy irányított gráf 7 csúcsponttal és 11 éllel.

Összefüggőség: Azt mondjuk, hogy G összefüggő, ha tetszőleges $i,j \in N,\ i \neq j$ csúcs párra létezik út i-ből j-be.

Irányított gráf Csúcsok (csomópontok) és irányított élek halmaza, $G = (N, A)$, ahol N a csúcsok és A az irányított élek (angolul *arcs*) halmaza. Most (i,j) egy rendezett pár, ami egy i-ből kiinduló és j-be befutó irányított élt jelöl. Az irányított éleket néha iveknek is szokás mondani.

További jelölések:
$O(i)$ azon élek végpontjai, amelyek az i-edik csúcsból indulnak ki. Formálisan: $O(i) = \{j \in \in N : (i,j) \in A\}$.
Ezzel párhuzamosan definiáljuk az $I(i)$ halmazt, ami az i-be befutó élek kezdőpontjainak a halmaza, formálisan: $I(i) = \{j \in N : (j,i) \in A\}$.

A 5.2 ábrán egy irányított gráfot mutatunk be. Érdekesség, hogy az 1 és 2 csúcsok között két él is van, fordított irányítással.

Az irányítatlan gráfokra fentebb definiált fogalmaknak van az irányított gráfokra vonatkozó változata.

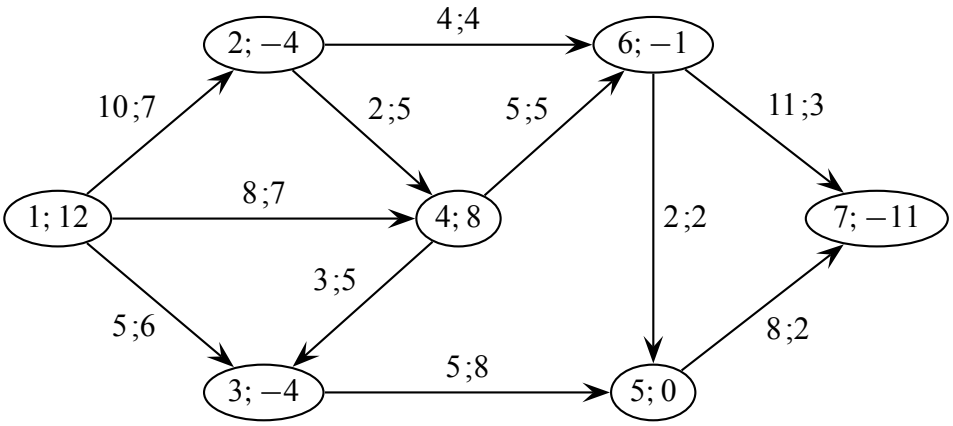
Út: egymástól különböző $i_1, \ldots, i_q$ csúcsok és a hozzájuk tartozó irányított élek $a_1, \ldots, a_{q-1}$ sorozata. Azt mondjuk, hogy

$$\text{egy } a_k \text{ él} \begin{cases} \textbf{előre mutató (forward arc)}, & \text{ha } a_k = (i_k, i_{k+1}), \\ \textbf{visszafelé mutató (backward arc)}, & \text{ha } a_k = (i_{k+1}, i_k). \end{cases}$$

Például, a 5.2 ábrán 1, (1,3),3, (4,3),4, (4,6),6 egy út, melyben van egy visszafelé mutató él, (4,3).

Irányított út: egy olyan út, amely csak előremutató élt tartalmaz. Például, a 5.2 ábrán egy út 4-ből 7-be: 4, (4,3), 3, (3,5), 5, (5,7),7.

Kör: egy olyan út, amelynek kezdő és végpontja ugyanaz a csúcs, $i_1 = i_q$. Például, egy 4-ből induló kör: 4, (4,6),6, (6,2),2, (2,1),1, (1,4), 4.



5.3. ábra. Példa egy hálózati optimalizálási feladatra. A csomóponti címkék tartalma: $(i; b_i)$*, az éleken lévő címkék:* $(u_{ij}; c_{ij})$

Összefüggőség: egy irányított gráfot összefüggőnek mondunk, ha az élek irányításának elhagyásával keletkező irányítatlan gráf összefüggő.

5.2. Az általános hálózati folyam probléma

A továbbiakban elsősorban irányított gráfokkal foglalkozunk, mert ezek a legalkalmasabbak a hálózati folyam probléma tárgyalására. Feltesszük tehát, hogy egy $G = (N, A)$ irányított gráffal van dolgunk. A hálózattal kapcsolatban az alábbi mennyiségeket definiáljuk:

$n = |N|$ a csomópontok száma,
$m = |A|$ az élek száma,
b_i előjeles igény az i ($i \in N$) csomópontban.

Az előjeles igény értelmezése:

$b_i > 0$: forrás csomópont,
$b_i < 0$: igény (nyelő) csomópont,
$b_i = 0$: átmenő csomópont.

Az $(i,j) \in A$ élekre a következőket definiáljuk:

x_{ij}: az i-ből a j csomópontba áramló folyam mennyisége (él változó), aminek az értéke a meghatározandó ismeretlen,
u_{ij}: az (i,j) él kapacitása,
c_{ij}: annak a költsége, hogy egy egységnyi folyamot mozgatunk az (i,j) élen.

5.2.1. A hálózati folyam probléma megfogalmazása

Verbálisan úgy fogalmazhatjuk meg a hálózati folyam problémát, hogy a forrás csomópontokból ($b_i > 0$) az összes mennyiséget el kell szállítani, a nyelő csomópontok ($b_i < 0$) összes igényét ki kell elégíteni és az átmeneti csomópontokban ($b_i = 0$) nem lehet semmilyen mennyiséget felhalmozni. Mindezek a megkötések maguk után vonják a $\sum_{i=1}^{n} b_i = 0$ feltételi követelmény teljesülését. Más szóval, a feladat csak akkor oldható meg ha ez teljesül. Ebből a

megfogalmazásból látszik, hogy akárhány forrás és akárhány nyelő is lehet a rendszerben mindaddig, amig ez az egyenlőség teljesül.

Ha a $\sum_{i=1}^{n} b_i = 0$ egyenlőség nem áll fenn, akkor igen egyszerű módon elérhetjük a teljesülést. Ha $\sum_{i=1}^{n} b_i > 0$ (vagyis ellátási többlet van), akkor egy fiktív nyelőt veszünk hozzá a rendszerhez és minden forrást összekötünk ezzel a nyelővel. Ezen élekhez tartozó c_{ij} költség 0 lesz, a nyelő igénye pedig éppen a többlet. Ha $\sum_{i=1}^{n} b_i < 0$, (hiány) akkor egy fiktív „forrást" definiálunk, amelynek kapacitása éppen a hiány lesz. Most is összekötünk minden nyelőt ezzel a forrással. A c_{ij} együtthatók beállításánál azonban vigyázni kell. Ugyanis, ha egy igényt a fiktív forrásból „elégítünk ki", akkor valójában nem elégítjük ki azt az igényt. Ilyenkor úgy célszerű a c_{ij}-ket meghatározni, hogy a hiányból fakadó veszteség minél kisebb legyen.

A hálózati folyam probléma formális megfogalmazása a verbális leírást követi. Először felírjuk a csomópontokban a folyammegmaradás feltételét, vagyis hogy egy csomópontba befutó és onnan kiáramló folyamok összege egyenlő. Itt természetesen be kell számítani a csomópont saját igényét vagy feleslegét. Szokás ezt—elektromos analógia alapján—Kirchhoff törvénynek is nevezni.

$$b_i + \sum_{j \in I(i)} x_{ji} = \sum_{j \in O(i)} x_{ij}, \quad \forall i \in N \tag{5.1}$$

Miután az egyes élekre kapacitáskorlátok vannak érvényben, ezt is figyelembe kell venni:

$$0 \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in A$$

Végül megfogalmazzuk a célfüggvényt

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

Ami az összköltség minimalizálását írja elő. A fentiekből egy kis átrendezéssel kapjuk a **minimális költségű hálózati folyam, MKHF** (angolul *minimal cost network flow*, MCNF) probléma általános alakját:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

feltéve, hogy

$$\sum_{j \in O(i)} x_{ij} - \sum_{j \in I(i)} x_{ji} = b_i, \qquad \forall i \in N$$

$$0 \le x_{ij} \le u_{ij}, \qquad \forall (i,j) \in A$$

Ez nem más, mint egy LP probléma, úgymint

$$\begin{aligned} \min \ & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \ & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{u} \end{aligned}$$

Itt az **A** mátrix a hálózat (gráf) ún. **él-csúcs incidencia mátrix**a. A mátrix oszlopai a gráf éleit reprezentálják. Egy (i,j) élhez tartozó oszlopvektor definíciója a következő:

$+1$	az i pozícióban	(honnan)
-1	a j pozícióban	(hová)
0	a többi helyen.	

Ide tartozik még a feladat megoldhatóságának feltétele, miszerint $\sum_{i \in N} b_i = 0$.

Példaként a 5.4 ábrán bemutatjuk a 5.3 ábrán található hálózati feladatot LP megfogalmazásban az incidencia mátrixszal.

	Élek										
Csúcsok	(1,2)	(1,3)	(1,4)	(2,4)	(2,6)	(3,5)	(4,3)	(4,6)	(5,7)	(6,5)	(6,7)	b_i
1	1	1	1									12
2	-1			1	1							-4
3		-1				1	-1					-4
4			-1	-1			1	1				8
5						-1			1	-1		0
6					-1			-1		1	1	-1
7									-1		-1	-11
u_{ij}	10	5	8	2	4	5	3	5	8	2	11	
c_{ij}	7	6	7	5	4	8	5	5	2	2	3	

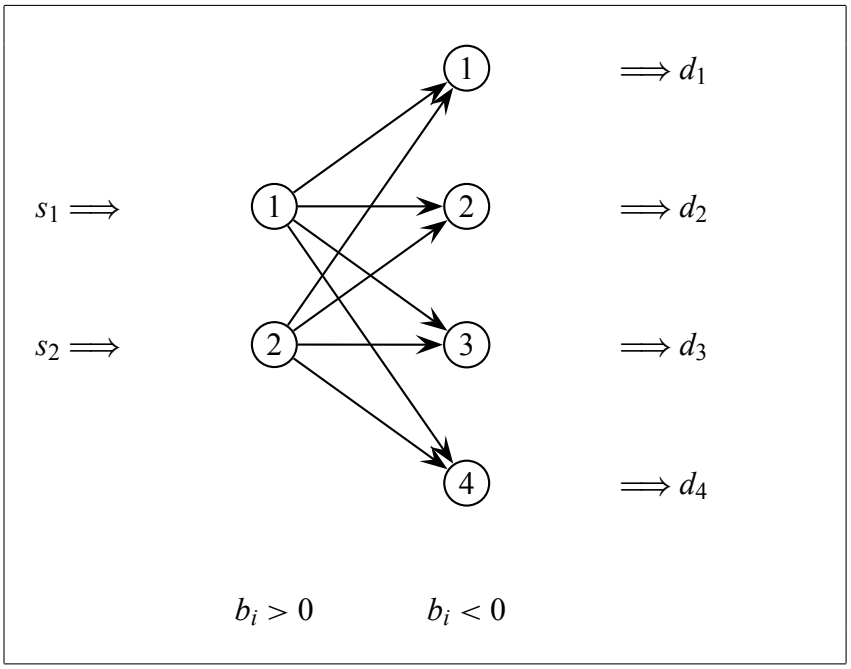
5.4. ábra. A 5.3 ábrán található minimum költségű hálózati folyam probléma mátrix alakja

5.2.2. Hálózati folyam probléma tulajdonságai

A hálózati folyam problémák legfontosabb matematikai tulajdonsága, az egészértékűség, ami abban áll, hogy amennyiben az összes b_i és u_{ij} egészértékű, akkor minden megengedett (és így az optimális) bázis megoldás komponensei egész számok. Mindez abból következik, hogy az **A** incidencia mátrix totálisan unimoduláris. Ez utóbbi tulajdonság fennállását általában igen nehéz bizonyítani, de van rá egy könnyen ellenőrizhető elégséges feltétel, ami viszont itt teljesül.

A további fontos tulajdonságok röviden.

- Ha feladat minden adata, beleértve a c_{ij} célfüggvény együtthatókat, egész értékű, akkor nemcsak a primál, hanem a duál megoldás is egész értékű.
- Az **A** mátrix minden **B** bázisa trianguláris, ami azt vonja maga után, hogy nincs szükség a bázis explicit $\mathbf{B}^{-1}$ inverzére.



5.5. ábra. A szállítási feladat folyam problémaként értelmezve.

- Ha az **B** bázishoz tartozó al-gráfot tekintjük, belátható, hogy az minden esetben egy T fa.

- A szimplex iteráció lépései, amelyek a bázis inverzzel ($\mathbf{B}^{-1}$) vannak definiálva, a bázis fa reprezentációján hajthatók végre igen kevés számítási munkával.

- Az összes számítási munka egész aritmetikával végezhető el, aminek következtében nincs számítási hiba, az eredmény teljesen pontos lesz.

Mindezek igen vonzóvá teszik a hálózati optimalizálási modelleket. Miután a gyakorlatban egy optimalizálási feladatot nagyon sok változatban szokás megoldani (alternatívák vizsgálata), ezért nagy jelentősége van, ha pontos megoldást lehet kapni, méghozzá nagyon gyorsan.

Jelen jegyzet keretei nem teszik lehetővé, hogy a hivatkozott hatékony algoritmus bemutatásra kerüljön. Ez ugyanis egy bonyolult logikájú eljárás. Fontos viszont, hogy az olvasó tudjon ennek az algoritmusnak a létezéséről, ami **hálózati szimplex** néven ismeretes, és szükség esetén biztonsággal tudjon hálózati modelleket megfogalmazni, tudva, hogy ezek (még nagy méretekben: több tízezer csomópont, több százezer él) is jól megoldhatók.

Szó volt róla, hogy a bemutatott hálózati feladat egy általános modell. Sok fontos modell osztály ennek a speciális esete. A továbbiakban arról adunk számot, hogyan lehet ezeket az általános alaknak tekinteni. Ez esetben ugyanis a hatékony hálózati szimplex alkalmazható a megoldásukra.

5.2.3. A szállítási feladat

Ez a modell arról szól, hogy valamilyen terméket, vagy anyagot el kell szállítani m raktárból (gyárból) n felvevőhelyre. Ismertek a forrásnál rendelkezésre álló mennyiségek, a felvevőhelyek (fogyasztók) igényei, továbbá az egységnyi szállítási költség minden viszonylatban. Formálisan:

m a forrás helyek száma
n a felvevő helyek száma
s_i rendelkezésre álló mennyiség az i ($i = 1, \ldots, m$) forrásnál
d_j igény a j-ik ($j = 1, \ldots, n$) fogyasztónál
c_{ij} szállítási egység költség az (i, j) viszonylatban
x_{ij} az i-edik forrásból a j-edik fogyasztóhoz szállított (ismeretlen) mennyiség.

Ebben az esetben a feladat megoldhatóságának feltétele az, hogy a források és a fogyasztói igények összege egyenlő legyen, vagyis $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$.

Az elérendő cél az, hogy minden igényt kielégítsünk a lehető legkisebb (minimális) összköltséggel. Ez az alábbi optimalizálási feladat megoldásával érhető el.

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

feltéve, hogy

$$\sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1, \ldots, n$$

$$\sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$$

$$x_{ij} \geq 0, \ \forall (i, j)$$

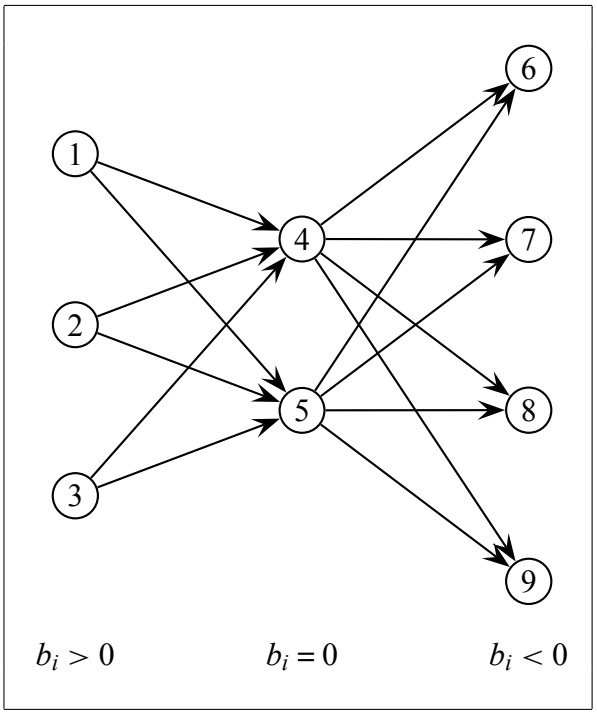
Ennek a feladatnak van egy változata, amikor néhány (vagy az összes) viszonylatban a változóra egyedi felső (kapacitás) korlát van előírva. Ezt nevezzük **kapacitás korlátos szállítási feladat**nak. Ez tehát azt jelenti, hogy a fenti szállítási feladathoz hozzávesszük a

$$0 \leq x_{ij} \leq u_{ij}, \quad \text{bizonyos, vagy az összes } (i, j)\text{-re}$$

feltételeket.

Most nézzük meg, hogy lesz ebből minimális költségű hálózati folyam probléma. Először is definiálunk egy *páros gráf*ot. Az egyik csoportba a forrás csúcsokat, a másikba a nyelő csúcsokat tesszük, ahogy azt a 5.5 ábra mutatja. Minden forrást összekötünk minden nyelővel, de a forrásokon, illetve nyelőkön belül nincs összeköttetés.

Ilyen értelmezés mellett ennek a feladatnak a megoldására alkalmazni lehet az idézett hálózati szimplex algoritmust.



5.6. ábra. Az átrakodásos szállítási feladat átfogalmazása MKHF problémává.

Van azonban a szimplex módszernek külön a szállítási feladat megoldására kidolgozott változata is, ami viszonylag egyszerű és kézi számolásra is alkalmas, noha ezt csak demonstrációs célból szokták végigcsinálni.

5.2.4. Átrakodásos szállítási feladat

Az átrakodásos szállítási (transshipment) feladat a szállítási feladat általánosításának tekinthető. Feltételezzük, hogy egy bizonyos terméket különböző helyeken állítanak elő. A terméket először közbülső tárolókba (pl. nagykereskedelmi raktárakba) szállítják, ahonnan (esetleg későbbi időpontban) tovább szállítják a végső felhasználóknak.

Ez a feladat típus is felírható minimum költségű hálózati problémaként, ahogy azt a 5.6 ábrán bemutatjuk. Itt világosan látható, hogy a 3 termelő egységből, 2 közbülső raktárból és 4 végfelhasználóból álló rendszer úgy is tekinthető, mint egy 9 csúcsot és 14 élt tartalmazó MKHF probléma.

A feladat további variációja a **kapacitás korlátos átrakodásos szállítási feladat**, ahol az élekre felső korlátot lehet megadni. Mindez nem változtat azon, hogy továbbra is MKHF feladattal állunk szemben.

5.2.5. Hozzárendelési probléma

Ezt a feladat típust többféleképpen szokták bevezetni. Az általunk követett változat szerint adva van m ember és ugyanannyi (m) feladat. Ahhoz, hogy az i-edik ember alkalmas legyen a j-edik feladat ellátására fel kell őt készíteni, ami c_{ij} költséggel jár.

Célunk az, hogy minden embert rendeljünk hozzá valamelyik feladathoz úgy, hogy ennek az összköltsége minimális legyen.

Könnyen látható, hogy ez egy elég széles körben használható modell, hiszen az „ember” helyére be lehet helyettesíteni járművet, gépet, stb. Sőt, lehet olyan helyzet is, amikor a célfüggvényt maximalizálni akarjuk, pl. amikor a c_{ij} együtthatók valami hasznosságot fejeznek ki.

Két további ésszerű feltételezéssel élünk: (i) minden embert pontosan egy feladathoz rendelünk hozzá, és (ii) minden feladathoz pontosan egy embert rendelünk hozzá.

A feladat konkrét megfogalmazásához bevezetjük az x_{ij} döntési változókat a következőképpen:

$$x_{ij} = \begin{cases} 1, & \text{ha az } i\text{-edik ember hozzá van rendelve a } j\text{-edik feladathoz,} \\ 0, & \text{egyébként.} \end{cases}$$

Ezek után a feladat:

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

feltéve, hogy

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1,\ldots,m$$

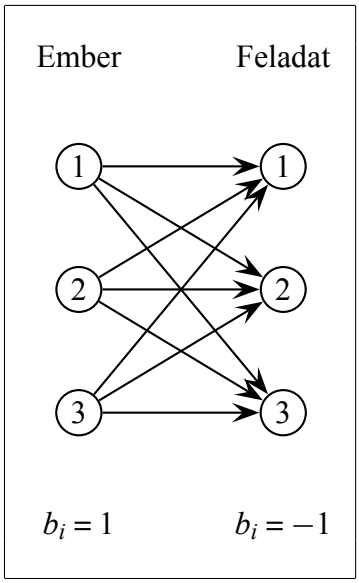
$$\sum_{j=1}^{m} x_{ij} = 1, \quad i = 1,\ldots,m$$

$$x_{ij} \in \{0,1\}, \text{ minden } (i,j) \text{ párra.}$$

A feltételek első csoportja azt írja elő, hogy minden munkához legyen hozzárendelve ember, a második csoport pedig azt, hogy minden ember legyen hozzárendelve valamelyik munkához. Az $x_{ij} \in \{0,1\}$ feltétel garantálja, hogy minden szummában egy és csak egy x_{ij} változó értéke lesz 1. Ez egy egészértékűségi megkötés. Emiatt ez egy egészértékű (bináris) LP feladat lesz. Ha ezt a B&B módszerrel próbálnánk megoldani, nagyon fáradságos lenne. Miután azonban a hozzárendelési feladat a szállítási feladat speciális esete (lásd 5.7 ábra), ezért ez egyben egy MKHF feladat is. Ennek következtében a feltételi mátrix totálisan unimoduláris, tehát az LP relaxáció optimális megoldása egészértékű, így egyben megoldása a hozzárendelési feladatnak.

5.2.6. Maximális folyam probléma

A hálózatot irányított gráfként kezeljük, $G = (N, A)$. Most azonban két csúcsnak kitüntetett szerepe van:



5.7. ábra. A hozzárendelési probléma értelmezése MKHF feladatként. Látszik, hogy ez a szállítási feladat speciális esete: minden forrásban egy egység áll rendelkezésre és minden nyelő igénye egy egység.

s : forrás csúcs,
t : nyelő csúcs,

az összes többi csúcs átmenő csúcs ($b_i = 0$). Az (i,j) él kapacitása korlátozott: $0 \le x_{ij} \le u_{ij}$, ahol azonban néhány (de nem az összes) $u_{ij} = +\infty$ is lehet.

A feladat az, hogy megtaláljuk a hálózat **maximális áteresztő kapacitás**át, vagyis a maximális folyamot, amit s-ből t-be át lehet nyomni a hálózaton keresztül. A feltételek megegyeznek a hagyományos folyam probléma feltételeivel, vagyis a folyam megőrzés (Kirchhoff törvény) minden csúcsban, a folyam nem-negatívitása minden él mentén és az él-kapacitások figyelembe vétele.

Formálisan a feladat:

$$\max\ b_s$$

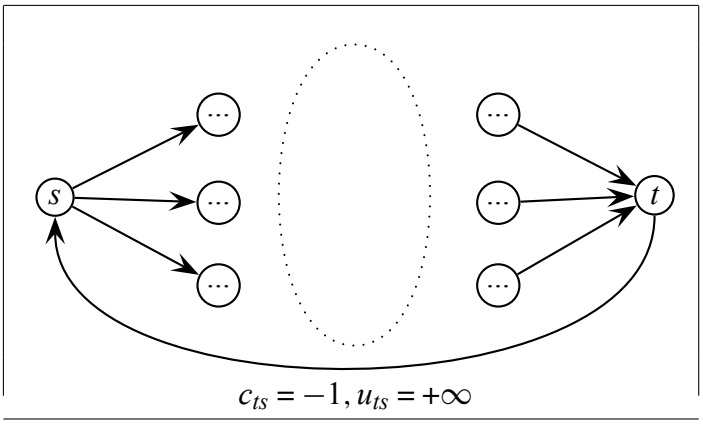
feltéve, hogy

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ b_t &= -b_s \\ b_i &= 0, \qquad i \neq s,t \\ 0 \le x_{ij} &\le u_{ij}, \end{aligned}$$

minden $(i,j) \in A$ párra.
Figyeljük meg, hogy most b_s nem konstans, hanem egy változó (akárcsak b_t).

Az eddigiektől eltérően, most kell egy kis átalakítást végrehajtanunk, hogy ebből is minimum költséges hálózati probléma legyen.

1. Először is, legyen $c_{ij} = 0$ minden (i,j) hálózati élre.



5.8. ábra. A maximális folyam probléma értelmezése MKHF problémaként.

2. Kössük össze a nyelőt és a forrást egy (t,s) éllel, melynek kapacitása $+\infty$ és a költség együtthatója egy negatív szám, mondjuk $c_{ts} = -1$.

3. Legyen $\bar{A} = A \cup \{(t,s)\}$ az egy éllel kibővített hálózat és jelölje x_{ts} az extra változót.

Az így keletkezett hálózatra egy példa 5.8 ábrán látható.

Ha minimalizáljuk a $\sum_{(i,j)\in\bar{A}} c_{ij}x_{ij}$ célfüggvényt (ami nem más, mint $-x_{ts}$), akkor a (t,s) él negatív célfüggvény együtthatója kényszeríteni fogja, hogy a (t,s) élen minél nagyobb mennyiség folyjon át a t csúcsból (és így s-ből is). Az egyéb körülmények megegyeznek a maximális folyam problémánál tárgyaltakkal.

5.2.7. Legrövidebb út probléma

Tegyük fel, hogy adva van egy irányított gráf, $G = (N,A)$, $|N| = n$, $|A| = m$. Most c_{ij} az (i,j) él hosszúságát jelöli minden $(i,j) \in A$ élre.

Egy útnak a hossza azon élek hosszának összege, amelyeken az út során áthaladtunk.

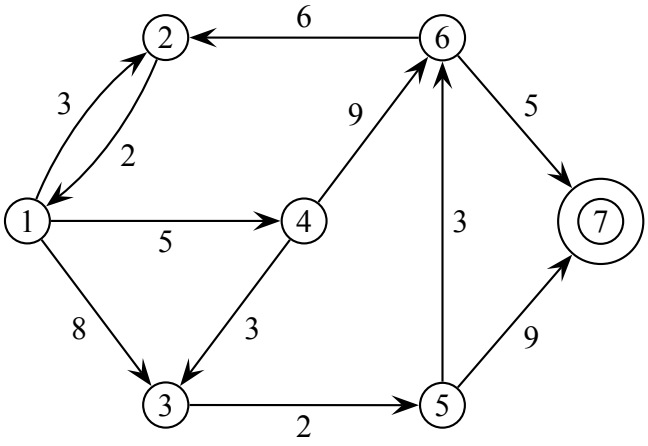
A **legrövidebb út** egy i és k csúcspont közt: egy olyan út, amelynek a hossza a legrövidebb az i-ből k-ba vezető összes út közül.

Ez egy alapprobléma a gráfelméletben. Éppen ezért már régóta foglalkoznak vele és van is a megoldásra hatékony algoritmus Dijkstra (1959) jóvoltából.

Mi egy kicsit általánosabb feladatot fogalmazunk meg. Ahelyett, hogy meghatároznánk a legrövidebb útat A két csúcsa között, egyidejűleg meghatározzuk a **legrövidebb utat az összes csúcstól** a gráf egy kijelölt csúcsához, mondjuk az n-edikhez. Ezt az alábbi feltételezések mellett tesszük meg, az MKHF módszertan segítségével.

1. Minden csúcsból létezik út n-be.

2. $O(n) = \emptyset$, vagyis n-ből nincs kifelé menő irányított él.



5.9. ábra. Keressük a legrövidebb útat a hálózatban az összes csúcsból a 7 csúcsba.

Ebből úgy lesz minimum költségű hálózati folyam probléma, ha néhány adatot speciálisan definiálunk.

Minden $i \neq n$ csúcsponthoz +1 egységnyi forrás mennyiséget rendelünk hozzá és az n jelű csúcshoz egy $-(n-1)$ értékű igényt. Az élek kapacitása $+\infty$.

Formálisan:

$$\begin{aligned} u_{ij} &= +\infty, && \forall (i,j) \in A, \\ b_i &= 1, && i \neq n, \\ b_n &= -(n-1). \end{aligned}$$

Ezek alapján a megoldandó MKHF probléma a következő:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

feltéve, hogy

$$\begin{aligned} \sum_{j\in O(i)} x_{ij} - \sum_{j\in I(i)} x_{ji} &= b_i, && \forall i \in N \\ x_{ij} &\geq 0, && \forall (i,j) \in A \end{aligned}$$

A feladatot megoldjuk a hálózati szimplex módszerrel. Az előzőekből tudjuk, hogy a megoldás egészértékű lesz. Az optimális bázishoz tartozó élek egy fát alkotnak, melynek gyökerében n van. Ebben a fában szerepel az összes csúcs és minden $i \neq n$ csúcsból egyetlen út vezet a gyökérbe és ez a legrövidebb út i-ből n-be.

6. fejezet

Bevezetés a nemlineáris optimalizálásba

A nemlineáris programozás (amit újabban *nemlineáris optimalizálás*ként is említenek) (2.3)–(2.5) alatti alakját fogjuk részletesebben vizsgálni. Emlékeztetőül, az idézett alak a következő:

$$\min f(\mathbf{x}) \tag{6.1}$$

$$g_k(\mathbf{x}) \leq 0, \qquad k = 1, \ldots, p, \tag{6.2}$$

$$h_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, m, \tag{6.3}$$

ahol $g_k(\mathbf{x})$ és $h_i(\mathbf{x})$ nemlineáris függvények és $\mathbf{x} \in \mathbb{R}^n$. A megengedett megoldások halmaza tehát

$$\mathcal{F} = \{\mathbf{x} : g_k(\mathbf{x}) \leq 0,\ k = 1, \ldots, p,\ h_i(\mathbf{x}) = 0,\ i = 1, \ldots, m.\}$$

6.1. Konvexitás, konkavitás

Konvexitás és konkavitás alapvető jelentőségű fogalmak a nemlineáris optimalizálásban. Ezért először ezekkel foglalkozunk.

Azt mondjuk, hogy egy **tartomány konvex**, ha bármely két pontját összekötő egyenes is a tartományhoz tartozik. A tartomány **nem konvex**, ha van két olyan pontja, hogy az azokat összekötő egyenesnek vannak tartományon kívüli pontjai. A fogalmak illusztrálása a 6.1 ábrán látható.

Egy $y = f(x)$ függvényről azt mondjuk, hogy a **függvény konvex** egy egydimenziós R intervallumon, ha tetszőleges $x_1, x_2 \in R$ pontokra és λ skalárra, ahol $0 \leq \lambda \leq 1$, teljesül, hogy

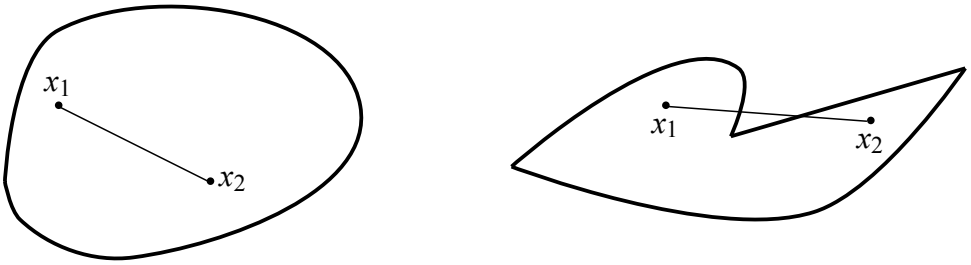
$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2). \tag{6.4}$$

Ha az $x(\lambda) = \lambda x_1 + (1-\lambda)x_2$ jelölést alkalmazzuk, akkor (6.4) így is írható:

$$f(x(\lambda)) \leq \lambda f(x_1) + (1-\lambda) f(x_2).$$

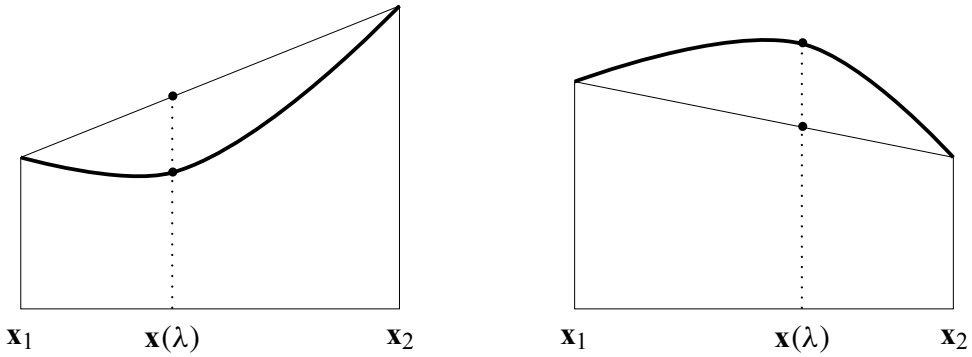
A **konkáv függvény** definiciója hasonló szellemben adható meg, csak az egyenlőtlenség iránya fordított:

$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda) f(x_2).$$



6.1. ábra. Egy konvex és egy konkáv tartomány (halmaz).

Rögtön látható, hogy ha f konvex függvény akkor $-f$ konkáv.



(a) Konvex függvény. (b) Konkáv függvény.

6.2. ábra. Konvex, illetve konkáv függvény a húr alatt, illetve fölött van.

A konvex és konkáv függvény fogalmának illusztrálása a 6.2 ábrán található.

A konvexitásból számos előnyös tulajdonság származik. Ezek közül a legfontosabb a következő:

Egy konvex függvény konvex halmazon vett lokális minimuma egyben globális minimum is.

A nemlineáris optimalizálás leggyakrabban használt modellje a **konvex optimalizálás**, amikoris minimalizálunk egy $f(\mathbf{x})$ konvex függvényt egy $\mathcal{F}$ konvex halmazon.

6.2. Feltétel nélküli optimalizálás

A nemlineáris optimalizálás egyszerű esete az, amikor $\mathcal{F} = \mathbb{R}^n$, vagyis egy függvény minimumát az egész n dimenziós térben keressük (nincsenek feltételek). Az ilyen feladatokat meg-

oldó algoritmusok azért is fontosak, mert a feltételes nemlineáris feladatok megoldásában is jelentős szerepet játszanak.

Egy minimum pont jellemzésére az egyváltozós függvények esetére definiált tulajdonságokat ültetjük át a többdimenziós esetre.

Definíció Egy $\mathbf{x}^*$ pont az $f(\mathbf{x})$ függvény **lokális minimum**a, ha $\mathbf{x}^*$ kis $\epsilon > 0$ sugarú környezetében levő $\mathbf{x}$-ekre $f(\mathbf{x}^*) < f(\mathbf{x})$. A minimum globális, ha ez a reláció teljesül tetszőleges $\mathbf{x}$-re.

Definíció Egy $f(\mathbf{x}) = f(x_1, \ldots, x_n)$ függvény **gradiens**e a parciális deriváltakból álló sor vektor:

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right].$$

A ∇ szimbólum kiejtése **nabla**. Noha a gradiens komponensei függvények, egy tetszőlegesen adott $\mathbf{x}$ pontban egy numerikus komponensű vektor lesz belőle.

Példa 5 *Legyen* $f(\mathbf{x}) = (x_1^2 + 2x_2)^2 - 3x_3^3$. *Ekkor a gradiens*

$$\nabla f(\mathbf{x}) = \left[4x_1^3 + 8x_1x_2,\ 4x_1^2 + 8x_2,\ -9x_3^2 \right].$$

A $(0, 1, -1/3)$ *pontban a gradiens* $[0, 8, -1]$.

A következő tételt bizonyítás nélkül ismertetjük.

Tétel Ha $\mathbf{x}^*$ lokális minimuma egy folytonosan deriválható $f(\mathbf{x})$-nek, akkor teljesül, hogy $\nabla f(\mathbf{x}^*) = \mathbf{0}$, vagyis ebben a pontban a gradiens a null-vektor:

$$\nabla f(\mathbf{x}^*) = \left[\frac{\partial f(\mathbf{x}^*)}{\partial x_1}, \frac{\partial f(\mathbf{x}^*)}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x}^*)}{\partial x_n} \right] = \mathbf{0}.$$

Ez a tétel egy szükséges feltételt határoz meg a minimumhelyre. Szokás ezt az **optimalitás elsőrendű feltétel**ének is nevezni, mert az elsőrendű deriváltak szerepelnek benne. Ha f konvex, akkor ez a feltétel elégséges is.

A másodrendű optimalitási feltételek megfogalmazásához szükség van az f függvény Hesse mátrixának definiálására.

Hesse mátrix Ha $f \in C^2$ (vagyis léteznek a másodrendű deriváltak), akkor az f függvény Hesse mátrixa az $\mathbf{x}$ pontban egy $n \times n$-es mátrix, melynek komponensei f-nek a másodrendű parciális deriváltjai. A mátrixot $\nabla^2 f(\mathbf{x})$-szel, vagy $\mathbf{H}(\mathbf{x})$-szel szokás jelölni:

$$\mathbf{H}(\mathbf{x}) = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right]_{i,j=1}^{n}.$$

Miután

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$$

nyilvánvaló, hogy a Hesse mátrix szimmetrikus. A fődiagonálisban szereplő $\dfrac{\partial^2 f}{\partial x_i \partial x_i}$ komponenst általában $\dfrac{\partial^2 f}{\partial x_i^2}$-tel jelöljük.

Noha a Hesse mátrix komponensei függvények, egy tetszőlegesen adott $\mathbf{x}$ pontban egy numerikus komponensű mátrix lesz belőle.

Példa 6 *Legyen f ugyanaz a függvény, mint a gradiens példában, vagyis $f(\mathbf{x}) = (x_1^2 + 2x_2)^2 - 3x_3^3$. A Hesse mátrix képzéséhez felhasználjuk a gradiens ismert alakját. A mátrix első sorát úgy kapjuk, hogy a gradiens első komponensének vesszük az x_1, x_2 majd x_3 szerinti parciális deriváltját. A többi sor értelemszerűen hasonló módon adódik.*

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 12x_1^2 + 8x_2 & 8x_1 & 0 \\ 8x_1 & 8 & 0 \\ 0 & 0 & -18x_3 \end{bmatrix}.$$

A $(0, 1, -1/3)$ pontban a Hesse mátrix

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 6 \end{bmatrix}.$$

Egy n változós $f(\mathbf{x})$ függvény parciális deriváltjaival jól lehet jellemezni a függvény extremális (minimum, vagy maximum) pontjait. Ismeretes, hogy egy $\mathbf{x}^*$ pont extremális csak akkor lehet, ha a gradiens ebben a pontban a null-vektor (szükséges feltétel). Bizonyítható, hogy ha ebben a pontban a Hesse mátrix létezik (nem szinguláris) és pozitív definit, akkor $\mathbf{x}^*$ a függvény lokális minimumhelye. Hasonlóképpen, ha a Hesse mátrix létezik és negatív definit, akkor itt f-nek lokális maximuma van.

6.2.1. Iteratív módszerek kereső iránnyal

Sok iteratív módszer, melyek az $f(\mathbf{x})$ függvényt minimalizálják, a következőképpen működik. Először meghatároznak egy $\mathbf{d}$ kereső irányt, amely mentén f elkezd csökkenni és a jelenlegi megoldásból ebben az irányban mozdulnak el. Az elmozdulás α lépéshosszát általában úgy határozzák meg, hogy az $f(\mathbf{x})$ függvény a lehető legnagyobb mértékben javuljon ebben az irányban. Így az alábbi sorozatot generálják:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k.$$

Ezek a módszerek tehát iterációnként két fő lépésből állnak: (i) keresési irány meghatározása és (ii) lépéshossz meghatározása.

6.2.2. Newton módszere $f(\mathbf{x})$ minimalizálására

Newton módszerének az alapja az, hogy a minimalizálandó $f(\mathbf{x})$ függvényt ($f \in C^2$) lokálisan egy kvadratikus függvénnyel közelítjük és ezt a kvadratikus függvényt egzakt módon minimalizáljuk. Ennek értelmében az $\mathbf{x}_k$ pont közelében $f(\mathbf{x})$-et a csonkított másodrendű Taylor

sorral közelítjük (amelyből hiányzik a hibatag)

$$f(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k).$$

A szükséges feltétel a lokális minimumra az, hogy $f(\mathbf{x})$ gradiense a null-vektor legyen:

$$\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}_k) + (\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}(\mathbf{x}_k) = \mathbf{0}^T,$$

amiből

$$\mathbf{x} = \mathbf{x}_k - \mathbf{H}^{-1}(\mathbf{x}_k)\left(\nabla f(\mathbf{x}_k)\right)^T.$$

Ebből a következő iteratív eljárás származtatható:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}^{-1}(\mathbf{x}_k)\left(\nabla f(\mathbf{x}_k)\right)^T,$$

ami a Newton módszer tiszta formája.

Ha $\{\mathbf{x}_k\} \to \mathbf{x}^*$ és $\mathbf{x}^*$ pontban a Hesse mátrix $\mathbf{H}(\mathbf{x}^*)$ pozitív definit akkor $f(\mathbf{x})$-nek lokális minimuma van $\mathbf{x}^*$-ben. Ennek a módszernek kitűnő (kvadratikus) konvergencia tulajdonságai vannak a lokális minimum közelében. Ahhoz, hogy szélesebb körben is konvergáljon, további finomítások szükségesek. Ezek általában a lépéshossz megválasztásra vonatkoznak, ugyanis a fenti formában a lépéshossz implicit módon egységnyinek van definiálva.

6.2.3. Kvadratikus alak (függvény)

A kvadratikus alak egy valós értékű kvadratikus függvénye az $\mathbf{x}$ vektornak:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c \tag{6.5}$$

ahol $\mathbf{A} \in \mathbb{R}^{m\times m}$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^m$ és c egy skalár (szám).

A kvadratikus alak vizsgálata azért indokolt, mert a nemlineáris programozásban előforduló függvényeket többnyire azok kvadratikus tagú Taylor sorfejtésével közelítik.

Megkísérelhetjük minimalizálni $f(\mathbf{x})$-t úgy, hogy a gradienst nullával tesszük egyenlővé: $f'(\mathbf{x}) = \nabla f(\mathbf{x}) = \mathbf{0}$. Meg lehet mutatni, hogy

$$(\nabla f(\mathbf{x}))^T = \frac{1}{2}\mathbf{A}^T\mathbf{x} + \frac{1}{2}\mathbf{A}\mathbf{x} - \mathbf{b}. \tag{6.6}$$

Ha $\mathbf{A}$ szimmetrikus, vagyis $\mathbf{A}^T = \mathbf{A}$, akkor (6.6) a

$$(\nabla f(\mathbf{x}))^T = \mathbf{A}\mathbf{x} - \mathbf{b} \tag{6.7}$$

alakra egyszerűsödik. Ha (6.7)-t $\mathbf{0}$-val tesszük egyenlővé, akkor $\mathbf{A}\mathbf{x} = \mathbf{b}$ adódik.

Könnyen belátható, hogy az $\mathbf{A}$ konstans mátrix az $f(\mathbf{x})$ kvadratikus alak Hesse mátrixa minden $\mathbf{x}$-ben. Tehát, ha $\mathbf{A}$ pozitív definit, akkor az $\mathbf{A}\mathbf{x} = \mathbf{b}$ megoldása $f(\mathbf{x})$ globális minimumát szolgáltatja. Fordítva is igaz, ha $\mathbf{x}$ minimalizálja $f(\mathbf{x})$-et, akkor egyidejűleg megoldja az $\mathbf{A}\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszert is.

Miután $\nabla f(\mathbf{x})$ abba az irányba mutat, amerre az $f(\mathbf{x})$ a leggyorsabban növekszik, $-(\nabla f(\mathbf{x}))^T = \mathbf{b} - \mathbf{A}\mathbf{x}$ az $f(\mathbf{x})$ leggyorsabb csökkenésének az iránya.

6.2.4. Legmeredekebb csökkenés módszere

A nemzetközi szakirodalomban ez a módszer a **method of steepest descent** (MSD) néven ismeretes. Ez egy kereső irányos iteratív módszer, amely a (6.5) kvadratikus forma minimalizálára szolgál.

A k-adik iterációban MSD azt az irányt választja, amerre az f függvény a leggyorsabban csökken, ez pedig negatív gradiens: $-(\nabla f(\mathbf{x}_k))^T = \mathbf{b} - \mathbf{A}\mathbf{x}_k$. Ez viszont nem más, mint a jelenlegi megoldás behelyettesítéséből adódó különbség vektor $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ (reziduum). Vagyis a kereső irány a reziduum vektor. A következő megoldást tehát így számíthatjuk ki:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k.$$

Ebben az irányban f az α_k lépéshossz függvénye:

$$f(\mathbf{x}_k + \alpha_k \mathbf{r}_k).$$

Ezt akarjuk úgy meghatározni, hogy f a legnagyobbat változzon. Ha az (α_k szerinti egyváltozós) iránymenti deriváltat nullával tesszük egyenlővé, akkor meghatározhatjuk α_k legkedvezőbb értékét.

$$\frac{df(\mathbf{x}_{k+1})}{d\alpha_k} = f'(\mathbf{x}_{k+1})\frac{d\mathbf{x}_{k+1}}{d\alpha_k} = \nabla f(\mathbf{x}_{k+1})\mathbf{r}_k = -\mathbf{r}_{k+1}^T\mathbf{r}_k = 0.$$

Ez utóbbi azt mondja, hogy a legjobb α_k esetén $\mathbf{r}_{k+1}$ ortogonális (merőleges) lesz $\mathbf{r}_k$-ra. Ennek alapján ki tudjuk számítani α_k-t. Kiindulva $\mathbf{r}_{k+1}^T\mathbf{r}_k = 0$-ból, az itt mellőzött levezetés után azt kapjuk, hogy

$$\alpha_k = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T\mathbf{A}\mathbf{r}_k}.$$

Ezek alapján a MSD algoritmus (első változata) a következő:

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$$
$$\alpha_k = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T\mathbf{A}\mathbf{r}_k}$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{r}_k.$$

Ez a képletsor akár rögtön programozható is. Az eljárás úgy indul be, hogy választunk egy tetszőleges induló megoldást, $\mathbf{x}_0$-t, hozzá $k = 0$-t és az algoritmus máris definiálva van. Miután ez egy végtelen sorozatot generál, kell egy megállási szabály. A gyakorlatban bevált kritérium az, hogy ha két egymást követő iteráció során a megoldás alig javul, akkor az eljárás befejeződik. Pontosabban, legyen $\epsilon > 0$ egy kicsi szám (a megkívánt pontosság). Ha az

$$e = \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{1 + \|\mathbf{x}_k\|} < \epsilon,$$

egyenlőtlenség teljesül, akkor az algoritmus véget ér és az utoljára kapott megoldást tekintjük a feladat (közelítő) megoldásának. ϵ értékét a megkívánt pontosságtól függően 10^{-10} és 10^{-6} közt célszerű választani.

6.2.5. Konjugált gradiens módszer

Noha a MSD eljárás a gyakorlatban igen jól működik ha **A** pozitív definit, bizonyos feladatok, illetve induló pontok esetén lassan konvergál. Miután az egymást követő kereső irányok egymásra merőlegesek, az irányok ismétlődhetnek, ami cikk-cakkozáshoz vezet egyre kisebb javulással. Ennek orvoslására alkalmas a **konjugált irányok módszere**, amit gyakran **konjugált gradiens módszer (CG)** néven is említenek. Ennek alapja a konjugáltság fogalma.

Konjugáltság A konjugáltság fogalma a vektorok merőlegességének (ortogonalitásának) általánosítása. Két m dimenziós vektor, $\mathbf{u}$ és $\mathbf{v}$ merőleges, ha a skalár szorzatuk nulla, $\mathbf{u}^T\mathbf{v} = 0$. Az egység mátrix beillesztésével: $\mathbf{u}^T\mathbf{I}\mathbf{v} = 0$. Ha $\mathbf{I}$-t helyettesítjük egy $m \times m$-es szimmetrikus $\mathbf{A}$ mátrixszal akkor azt mondjuk, hogy $\mathbf{u}$ és $\mathbf{v}$ A-ortogonális, vagy konjugált ($\mathbf{A}$-ra nézve) ha

$$\mathbf{u}^T\mathbf{A}\mathbf{v} = 0.$$

Példa 7 *Legyen*

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \text{ és } \mathbf{u}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \mathbf{v}_1 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \text{ ekkor } \mathbf{u}_1^T\mathbf{A}\mathbf{v}_1 = 0,$$

vagyis $\mathbf{u}_1$ *és* $\mathbf{v}_1$ *konjugált, A-ortogonális (de nem merőleges). Hasonlóképpen,* $\mathbf{u}_2 = [2,-3]^T$ *és* $\mathbf{v}_2 = [1,0]^T$ *szintén A-ortogonális de* $\mathbf{u}_3 = [1,0]^T$ *és* $\mathbf{v}_3 = [0,1]^T$ *nem.*

Meg lehet mutatni, hogy ha $\mathbf{A}$ pozitív definit, akkor létezik m darab A-ortogonális (konjugált) vektor $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{m-1}$, úgy hogy $\mathbf{d}_i^T\mathbf{A}\mathbf{d}_j = 0$ minden $0 \le i,j \le m-1$, $i \neq j$-re. Ezek valójában konjugált irányok, hiszen a vektorok tetszőleges skalárszorosai szintén konjugáltak.

Ezeket az irányokat használva, szükségünk van még az α_k lépéshosszra. Most is az iránymenti derivált nullává tételével kapjuk meg, hogy

$$\alpha_k = \frac{\mathbf{d}_k^T\mathbf{r}_k}{\mathbf{d}_k^T\mathbf{A}\mathbf{d}_k}$$

A konjugált irányok meghatározása a Gram-Schmidt eljárás segítségével érhető el hatékonyan. Szerencsére ezeket az irányokat nem kell előre meghatározni, mert lehetőség van mindig csak a következő irány kiszámítására. Ezt beépítve, a CG módszer egyszerű változata a következő (értelmezés a SDM-nél leírtak szerint):

$$\begin{aligned}
\mathbf{d}_0 = \mathbf{r}_0 &= \mathbf{b} - \mathbf{A}\mathbf{x}_0 \\
\alpha_k &= \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{d}_k^T\mathbf{A}\mathbf{d}_k} \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k\mathbf{d}_k \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{d}_k \\
\beta_{k+1} &= \frac{\mathbf{r}_{k+1}^T\mathbf{r}_{k+1}}{\mathbf{r}_k^T\mathbf{r}_k} \\
\mathbf{d}_{k+1} &= \mathbf{r}_{k+1} + \beta_{k+1}\mathbf{d}_k.
\end{aligned}$$

Ezzel a módszerrel lehet megtalálni $f(\mathbf{x})$ minimumát. Bizonyítható, hogy ha $\mathbf{A}$ pozitív definit, akkor CG nem több, mint m lépésben megtalálja a minimumot. A megállási szabály a MSD módszernél használttal azonos lehet.

6.3. Feltételes optimalizálás

Ha $f(\mathbf{x})$-et úgy kell minimalizálni, hogy $\mathbf{x}$-nek bizonyos feltételeket is ki kell elégíteni, akkor feltételes optimalizálásról beszélünk. Először azt az esetet tárgyaljuk, amikor a feltételek egyenlőségek:

$$\begin{aligned} h_1(\mathbf{x}) &= 0 \\ h_2(\mathbf{x}) &= 0 \\ &\vdots \\ h_m(\mathbf{x}) &= 0, \end{aligned}$$

vagy vektor jelöléssel $\mathbf{h}(\mathbf{x}) = \mathbf{0}$.

6.3.1. Elsőrendű szükséges feltételek a szélsőértékre

Legyen az $\mathbf{x}^*$ pontban az f függvénynek lokális szélsőértéke a $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ feltételek mellett. Tegyük fel továbbá, hogy ebben a pontban a feltételek gradiensei lineárisan függetlenek ($\mathbf{x}^*$ egy reguláris pont). Ekkor létezik egy $\boldsymbol{\lambda} \in \mathbb{R}^m$ úgy, hogy

$$\nabla f(\mathbf{x}^*) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}^*) = \mathbf{0}.$$

A bizonyítástól itt eltekintünk. Megjegyzendő, hogy itt $\nabla \mathbf{h}(\mathbf{x}^*)$ a feltételi függvények gradienseiből álló mátrix, így ez a feltétel n egyenlőséget jelent. Ha ezekhez hozzávesszük a $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ feltételi egyenlőségeket, akkor egy $n+m$ (többnyire nemlineáris) egyenletből és $n+m$ változóból álló rendszert kapunk, ami elvileg megoldható, sőt, nem túl bonyolult függvények esetében gyakorlatilag is. Általában iteratív módszerek használatosak. A megoldás egy olyan pont (stacionárius pont), amelyben f-nek lokális szélsőértéke lehet.

Szokás definiálni a rendszer **Lagrange függvényét**:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}).$$

Azonnal látszik, hogy a Lagrange függvény értéke minden megengedett pontban (ahol $\mathbf{h}(\mathbf{x}) = \mathbf{0}$) megegyezik f értékével. A $\boldsymbol{\lambda}$ vektor komponenseit **Lagrange szorzó**knak (multiplikátoroknak) hívjuk. $L(\mathbf{x}, \boldsymbol{\lambda})$ segítségével az előbbi szükséges feltételek a

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{0} \\ \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{0}. \end{aligned}$$

alakban írhatók fel, ahol $\nabla_{\mathbf{x}}$ az $\mathbf{x}$, a $\nabla_{\boldsymbol{\lambda}}$ pedig az $\boldsymbol{\lambda}$ szerinti parciális deriváltat jelöli. Azt, hogy egy stacionárius pont szélsőérték hely-e, a pontban definiált érintősík vizsgálatával lehet eldönteni.

Ha a feltételek közt vannak **egyenlőtlenségek is**, akkor a helyzet valamivel bonyolultabb.

Definíció Legyen $\mathbf{x}^*$ egy megengedett pont, azaz

$$\mathbf{h}(\mathbf{x}^*) = \mathbf{0} \quad \text{és} \quad \mathbf{g}(\mathbf{x}^*) \leq \mathbf{0},$$

továbbá legyen $\mathcal{K}$ az indexhalmaza azon egyenlőtlenség feltételeknek, amelyek ebben a pontban aktívak, vagyis $g_k(\mathbf{x}^*) = 0$. Azt mondjuk, hogy $\mathbf{x}^*$ a feltételek **reguláris pont**ja, ha a $\nabla h_i(\mathbf{x}^*),\ i = 1, \ldots, m$ és $\nabla g_k(\mathbf{x}^*),\ k \in \mathcal{K}$ gradiens vektorok lineárisan függetlenek.

Ebben az esetben a rendszer Lagrange függvénye:

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}).$$

6.3.2. Karush-Kuhn-Tucker (KKT) feltételek

Legyen $\mathbf{x}^*$ a

$$\min f(\mathbf{x}) \tag{6.8}$$

$$f.h. \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \qquad \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \tag{6.9}$$

feladat egy lokális szélsőértéke és tegyük fel, hogy $\mathbf{x}^*$ egy reguláris pontja a feltételeknek. Ekkor létezik egy $\boldsymbol{\lambda} \in \mathbb{R}^m$ vektor és egy $\boldsymbol{\mu} \in \mathbb{R}^p,\ \boldsymbol{\mu} \geq \mathbf{0}$ vektor úgy, hogy

$$\nabla f(\mathbf{x}^*) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}^*) + \boldsymbol{\mu}^T \nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0} \tag{6.10}$$

$$\boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}^*) = 0. \tag{6.11}$$

Ez a feltételes szélsőérték létezésének **elsőrendű szükséges feltétel**e a (6.8) – (6.9) általános esetben. Vegyük észre, hogy $\boldsymbol{\mu} \geq \mathbf{0}$ miatt (6.11) csak úgy teljesülhet, ha a skalárszorzat minden tagja egyenlő nullával, vagyis $\mu_k g_k(\mathbf{x}^*) = 0,\ k = 1, \ldots, p$. Ez azt jelenti, hogy ha $g_k(\mathbf{x}^*) < 0$, vagyis a feltétel nem aktív ebben a pontban, akkor a hozzátartozó Lagrange szorzó nulla. Továbbá, ha egy Lagrange szorzó pozitív, akkor a neki megfelelő feltétel aktív, $g_k(\mathbf{x}^*) = 0$.

Konvex programozási feladatok esetén a KKT feltételek teljesülése szükséges és elégséges feltétele a minimum helynek.

Egy konkrét feladat megoldása során az aktív feltételek több kombinációjával kísérletezhetünk. Ez azt jelenti, hogy $\boldsymbol{\mu}$ bizonyos komponenseit nullára rögzítjük (inaktívvá tesszük) és a maradék rendszert megoldjuk minden alkalommal ellenőrizve az adódó μ_k Lagrange szorzók előjelét. Ha azok nem-negatívak, akkor találtunk egy KKT pontot.

Irodalomjegyzék

[1] Illés T., Nagy M., és Terlaky T., *Belsőpontos algoritmusok*, pp. 1230–1297. Iványi Antal (alkotó szerkesztő), Informatikai Algoritmusok II., ELTE Eötvös Kiadó, Budapest, 2005.

[2] Maros I., *Computational Techniques of the Simplex Method*, Kluwer Academic Publishers, 325 pages, 2003.

Tárgymutató