

TÖBBFOKOZATÚ SZÁLLÍTÁSI PROBLÉMA

KOVÁCS LÁSZLÓ BÉLA

Bevezetés

A többfokozatú szállítási probléma, amellyel ebben a cikkben foglalkozunk, abban különbözik a közönséges szállítási feladattól, hogy nemcsak két fajta telephelyet különböztetünk meg — az egyik, ahonnan, a másik, ahová szállítunk —, hanem közbülső helyek is vannak, amelyeken átmenő szállítás történik. Ezen közbülső helyek kapacitása korlátozott. A továbbiakban részletesen tárgyalt gabonaraktárak — malmok — lisztraktárak közötti szállítási problémán kívül ilyen problémára vezet például, ha a termelőhelyekről bizonyos árucikkeket elszállítanak a központi raktárakba, majd onnan második fokozatként az üzletekbe. További példaként megemlíjtük az olyan termékek esetét, amelyeket több különböző típusú üzemben munkálnak meg egymás után, és mindegyik fajta gyárból több van. Meghatározandó az összes gyárat magábafoglaló kooperációs program úgy, hogy a kombinált szállítási és termelési összköltség minimális legyen. (A termelési költséget hozzászámíthatjuk a megfelelő szállítási költséghez.)

Matematikailag a probléma a következő:

Keresendő az

$$x_{il} \geq 0, \quad y_{lj} \geq 0$$

$$\sum_{l=1}^n x_{il} = a_i \quad (i = 1, 2, \dots, m)$$

$$\sum_{i=1}^m x_{il} = \sum_{j=1}^p y_{lj} \leq k_l \quad (l = 1, 2, \dots, n)$$

$$\sum_{l=1}^n y_{lj} = b_j \quad (j = 1, 2, \dots, p)$$

rendszer kielégítő azon x_{il} , y_{lj} változó-értékek, melyek minimalizálják a

$$\sum_{i=1}^n \sum_{l=1}^m c_{il} x_{il} + \sum_{j=1}^p \sum_{l=1}^n d_{lj} y_{lj}$$

lineáris kifejezést. Az itt szereplő $a_i > 0$, $b_j > 0$, $k_l > 0$, c_{il} , d_{lj} mennyiségek adott konstansok, amelyekre a

$$\sum_{i=1}^m a_i = \sum_{j=1}^p b_j \leq \sum_{l=1}^n k_l$$

relációk teljesülnek. Itt a_i jelenti az i -edik termelőhely kínálatát, b_j a j -edik felhasználó fogyasztását és k_j a j -edik közbülső hely kapacitását.

A probléma jelentőségét mutatja, hogy már többen foglalkoztak a megoldásával. Az egyik megfogalmazás szerint, amikor az i -edik termelőhelyről az l -edik közbülső helyen keresztül a j -edik fogyasztóhoz x_{ilj} mennyiséget juttatunk. Ekkor az $x_{ilj} \geq 0$

$$\sum_i \sum_j x_{ilj} = a_i \quad \sum_i \sum_j x_{ilj} \leq k_l \quad \sum_i \sum_l x_{ilj} = b_j$$

$$\min \sum_i \sum_l \sum_j (c_{il} + c_{lj}) x_{ilj}$$

ún. multiindexes szállítási problémához jutunk. (L. például [5]) Ekkor azonban a változók száma mnp , ami viszonylag kis m , n és p esetén is igen nagy. A változók nagy száma miatt még az elektronikus géppel való megoldás is komoly problémát okoz. Például $m = n = p = 30$ esetén hazai gépekkel már nem tudnánk megoldani a problémát. A korábbi megfogalmazásban csak $mn + np$ változó szerepel!

A másik megoldás W. SZWARC lengyel matematikus [3] dolgozatában olvasható az ún. szállítási séma alkalmazásaként. Ebben negatív változók is lehetnek. A meg nem engedett szállításokat, például gabona a lisztraktárba, $+\infty$ ill. $-\infty$ megfelelő költségekkel zárja ki az optimumból. Ezek a lehetőségek a jelen tárgyalásban már nem is szerepelnek. További rövidítést jelent a pótváltozók egyszeri szerepeltetése a megoldó táblázatokban a Szwarc-modellben található kétszeres felhasználással szemben.

A bizonyításokat illetően nem SZWARC eredményeit használtuk fel, hanem a klasszikus szállítási probléma megfelelő tételeit általánosítottuk. A standard szállítási feladat elmélete megtalálható például HADLEY [2] és G. B. DANTZIG [1] könyvében. Az utóbbi könyv tartalmazza a teljesen általános gráffal jellemezhető rendszeren való szállítási problémát („transshipment problem”), azonban az éppen az általános volta miatt meglehetősen bonyolult.

A jelen cikk 1–2. pontja a feladat ismertetését és változatait tárgyalja. A 3. pont az előkészítő tételeket tartalmazza, melyek a klasszikus szállítási probléma megfelelő tételeinek általánosításai. Az algoritmus lényegének leírása a 4. pontban van, míg az 5–7. és 9. pontok kiegészítéseket tartalmaznak. Végül az egész eljárást egy numerikus példán szemléltetjük a 8. pontban.

1. Egy gyakorlati alkalmazás

Abból a célból, hogy a probléma lényegét és gyakorlati jelentőségét jobban láthassuk, kezdjük vizsgálódásainkat egy konkrét feladaton.

Néhány évvel ezelőtt az optimális gabona ill. liszt szállítástervezés kérdésével a Matematikai Kutató Intézethez fordultak a gazdasági szakemberek. Az áttekinthetőség kedvéért most leegyszerűsítve tárgyaljuk ezt a problémát.

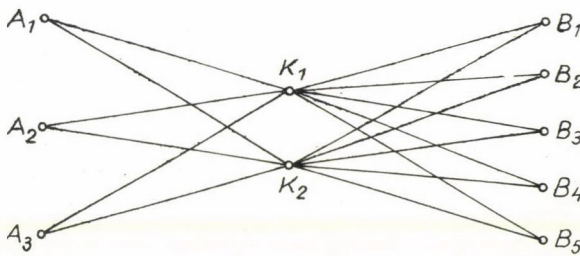
A feladatban m gabonaraktár, n malom és p lisztárolóhely szerepel. Ezeket rendre A_i , K_l és B_j -vel jelöljük. Adva van továbbá az i -edik gabonaraktárból az l -edik malomba való szállítás egységköltsége, c_{il} , és az l -edik malomból a j -edik lisztárolóhelyre való szállítás egységköltsége, d_{lj} . Megjegyzendő, hogy ekkor a malomba beérkező és onnan elszállított mennyiség

nem azonos, hiszen 1q búzából nem lesz 1q liszt, hanem csak kevesebb. Ezt a nehézséget elkerülhetjük úgy, hogy a lisztet is búza egységekben mérjük, azaz például egy egységnyinek nevezzük azt a lisztmennyiséget, ami 1q (általában egységnyi) búzából lesz. Ezen egységnyi liszt szállítási költsége lesz d_{lj} , és a lisztraktárak „igényét” is ebben az egységben állapítjuk meg. Az i -edik gabonátárolóhelyen a_i mennyiségű gabona áll rendelkezésre, a j -edik liszt-tárolóhely szükséglete pedig b_j . Ekkor a fenti megjegyzést figyelembe véve, az összes gabona mennyisége megegyezik az összes lisztkereslettel:

$$(1.1) \quad \sum_{i=1}^m a_i = \sum_{j=1}^p b_j.$$

Legyen továbbá az l -edik malom kapacitása k_l .

Készítsünk egy vázlatot, melyen ábrázoljuk a gabonaraktárakat (A_i , $i = 1, 2, \dots, m$), a malmokat (K_l , $l = 1, 2, \dots, n$) és a liszt-tárolóhelyeket (B_j , $j = 1, 2, \dots, p$), valamint a közöttük levő szállítási útvonalakat. Nyilvánvaló, hogy A_i -kből K_l -ekbe és K_l -ekből B_j -kbe vezetnek utak, de közvetlenül az A_i -kből B_j -kbe nem, hiszen gabonát egyenesen nem szállíthatunk a lisztraktárba. Az 1. ábrán az $m = 3$, $n = 2$ és $p = 5$ esetet vázoltuk.



1. ábra

Legyen az A_i -ből K_l -be szállított mennyiség x_{il} , K_l -ből B_j -be szállított mennyiség y_{lj} . Ekkor az A_i gabonaraktárból elszállított összes mennyiségnek egyenlőnek kell lennie az ott található mennyiséggel, azaz

$$(1.2) \quad \sum_{l=1}^n x_{il} = a_i \quad (i = 1, 2, \dots, m)$$

hasonlóan a különböző malmokból B_j -be szállított mennyiségnek egyenlőnek kell lennie az ottani kereslettel,

$$(1.3) \quad \sum_{l=1}^n y_{lj} = b_j \quad (j = 1, 2, \dots, p).$$

Továbbá minden malomból pontosan ugyanannyi lisztet vihetünk el (gabona-egységekben mérve!), amennyi gabonát odaszállítottunk, ezenkívül ez a „keresztúlszállított” mennyiség nem lehet nagyobb, mint az illető malom kapacitása, azaz

$$(1.4) \quad \sum_{i=1}^m x_{il} = \sum_{j=1}^p y_{lj} \leq k_l \quad (l = 1, 2, \dots, n).$$

Mivel az A_i -ből a K_l -be való szállítás egységköltsége c_{il} , a teljes mennyiség, x_{il} szállítási költsége $c_{il}x_{il}$. Minden gabonaraktár-malom viszonylatra ki-számolva az összköltség $\sum_{l=1}^n \sum_{i=1}^m c_{il}x_{il}$. Hasonlóan malom-lisztraktár szállítási összköltsége $\sum_{j=1}^p \sum_{l=1}^n d_{lj}y_{lj}$. Így a szállítások teljes költsége:

$$(1.5) \quad \sum_{l=1}^n \sum_{i=1}^m c_{il}x_{il} + \sum_{l=1}^n \sum_{j=1}^p d_{lj}y_{lj}.$$

A feladat tehát a következő: Keresendők olyan $x_{il} \geq 0$ és $y_{lj} \geq 0$ számok ($i = 1, \dots, m$; $l = 1, \dots, n$; $j = 1, \dots, p$), amelyek kielégítik az (1.2)–(1.4) feltételeket és emellett az (1.5) függvény a lehető legkisebb értéket vesz fel.

A most vázolt modell természetesen nem írja le pontosan a valóságot, bizonyos kiegészítésekre szorul. Például nem vettük figyelembe azt, hogy a malmok különböző költséggel őrlnek. Ezen könnyen segíthetünk oly módon, hogy az őrlési költséget hozzászámítjuk vagy az odaszállítás vagy az elszállítás költségeihez.

2. A probléma változatai

A többfokozatú szállítási feladat felmerülhet a tervezés stádiumában is bizonyos ipari létesítményeknél vagy bizonyos termékeknel, amelyeket egymás után különböző helyeken különböző műveleteknek vetnek alá. A megművelés költségeit — amennyiben azok azonos típusú gyáranként is változnak — hozzászámíthatjuk ismét a megfelelő szállítási költségekhez.

Jelentsenek most az A_i -k gyárakat, ahol a munkadarabokon az első műveletet végzik. Ezután a munkadarabok a K_l -vel jelölt kooperáló gyárakba jutnak, majd átadják a kész termékeket a B_j -vel jelölt fogyasztóknak. Legyen az A_i üzem termelési kapacitása a_i ($i = 1, 2, \dots, m$), a K_l üzemeké k_l ($l = 1, \dots, n$), végül legyen a B_j fogyasztó igénye b_j . Ez a példa mindössze abban különbözik az 1. pontban tárgyalttól, hogy nem kívánjuk meg a

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

egyenlőséget, ugyanis lehet, hogy az A_i gyárak összkapacitása nagyobb, mint az összkéréslet (ekkor bizonyos gyárakban marad kihasználatlan kapacitás), azonban kisebb nem lehet, azaz a

$$(2.1) \quad \sum_{i=1}^m a_i \geq \sum_{j=1}^p b_j$$

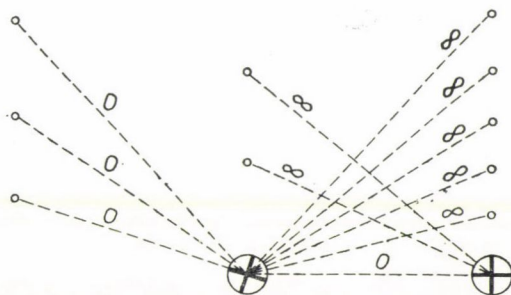
egyenlőtlenség fogja helyettesíteni az 1. pontban szereplő példában feltételezett (1.1) egyenlőséget. Szerencsére ez a probléma könnyen visszavezethető az előbbire. Vegyünk fel ugyanis egy fiktív gyárat a középső csoportban, valamint egy fiktív fogyasztót, ahova a többletkapacitásnak megfelelő mennyiséget szállítjuk, azaz legyen

$$k_{n+1} = b_{p+1} = \sum_{i=1}^m a_i - \sum_{j=1}^p b_j.$$

Természetesen most már

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

azonban azt még biztosítanunk kell, hogy ez a fiktív szállítás ne jelentsen többletköltséget, tehát ne zavarja meg az igazi szállítást, valamint azt, hogy a K_{n+1} közbülső (fiktív) állomásról ne szállíthassunk árut a $B_1, \dots, B_p$ valódi fogyasztókhoz, továbbá valódi árut ne szállíthassunk a fiktív fogyasztóhoz. Ezt a három kívánalmat könnyen kielégíthetjük, ha $c_{i,n+1} = d_{n+1,p+1} = 0$ ($i = 1, \dots, m$) egységköltséget választjuk a fiktív szállításokra, valamint $d_{n+1,j} = \infty$ ($j \neq p+1$) és $d_{l,n+1} = \infty$ ($l \neq n+1$) egységköltséget választjuk a meg nem engedett szállításokra. Az eddig elmondottak azonnal világossá válnak, ha egy pillantást vetünk a 2. ábrára, melyen ismét az $m = 3$, $n = 2$, $p = 4$ esetet vázoltuk, de bejelöltük a fiktív állomásokat és a hozzá tartozó egységköltségeket is. Az ábrán az eredeti utakat nem tüntettük fel, hogy elkerüljük a sok vonal okozta zavart. Természetesen a ∞ költségek helyett a gyakorlati számításoknál egy kellően nagyra választott költséget választunk, amely az optimális megoldásban lehetetlenné teszi a meg nem engedett szállítások részvételét.



2. ábra

Előfordulhat az is, hogy bizonyos gyárakat csak ezután fogunk létesíteni, így azok kapacitását tetszőlegesen nagyra tekinthetjük a számítások megkezdésekor. Könnyen elkerülhetjük az ebből adódó nehézségeket, ugyanis ha például K_l egy ilyen tetszőleges kapacitású gyár, akkor a $k_l = \sum_{j=1}^n b_j$ kapacitás felvételével nem korlátoztuk a K_l gyáron keresztülhaladó mennyiséget, mert a legrosszabb esetben a teljes kereslet ezen az üzemen mehet keresztül. A most elmondottak érvényesek a kiinduló, A_i gyárakra vagy raktárakra is. Megjegyzendő, hogy a $\sum_{i=1}^m a_i = \sum_{l=1}^n k_l = \sum_{j=1}^p b_j$ esetben, amikor biztosan nem lesz kihasználatlan kapacitás a feladat egyszerűen mint két egymásutáni szállítási probléma oldható meg. Az eddig elmondottak érvényesek akkor is, ha az $A_1, \dots, A_m$ kiinduló helyekről a $K_1, \dots, K_n$, majd innen az $L_1, \dots, L_r$ közbülső feldolgozó helyekre és innen a $B_1, \dots, B_p$ célállomásokra kerül az áru, sőt a közbülső fázisok száma még ennél is több lehet. (L. bővebben a 9. pontban.)

3. Előkészítő tételek

A megoldás alapvetően a közönséges szállítási feladat gondolatmenetét követi, így a megfelelő általánosabb tételeket kell először bebizonyítanunk.

Áttekinthetőség kedvéért foglaljuk össze a feladatot. Tegyük fel, hogy az (1.1) feltétel teljesül, azaz

$$(3.1) \quad \sum_{i=1}^m a_i = \sum_{j=1}^p b_j.$$

Keresendők az (1.2)–(1.4) feltételnek eleget tevő oly x_{il} , y_{lj} számok, melyre az (1.5) függvény a minimumát veszi fel.

Az (1.4) feltétel rendkívül kényelmetlen, alapjában két feltételt foglal össze. Az egyik azt jelenti, hogy a K_l -be beérkező összmennyiség megegyezik az onnan elszállított összmennyiséggel, azaz

$$(3.2) \quad \sum_{i=1}^m x_{il} = \sum_{j=1}^p y_{lj} \quad (l = 1, \dots, n)$$

a másik pedig azt, hogy ez a közös érték, azaz az áthaladó mennyiség nem több, mint a K_l pontbeli k_l kapacitás.

$$(3.3) \quad \sum_{i=1}^m x_{il} \leq k_l \quad (l = 1, \dots, n).$$

Ezt úgy is fogalmazhatjuk, hogy a K_l -ben mind a befutó, mind a kiszállított összmennyiség az adott k_l korlát alatt marad, és pedig ugyanannyival. Azaz a (3.2) és (3.3) együtt equivalentens az alábbi két egyenlőség rendszerrel:

$$(3.4) \quad \sum_{i=1}^m x_{il} + s_l = k_l \quad (l = 1, \dots, n)$$

$$(3.5) \quad \sum_{j=1}^p y_{lj} + s_l = k_l \quad (l = 1, \dots, n)$$

ahol $s_l \geq 0$ ($l = 1, \dots, n$).

Ennek alapján tehát a feladatunk a következőképpen írható fel, az (1.4)-et, ill. (3.2)–(3.3)-t a (3.4)–(3.5)-tel helyettesítve:

$$(3.6) \quad \sum_{i=1}^n x_{il} = a_i \quad (i = 1, 2, \dots, m)$$

$$(3.7) \quad \sum_{i=1}^m x_{il} + s_l = k_l \quad (l = 1, 2, \dots, n)$$

$$(3.8) \quad \sum_{j=1}^p y_{lj} + s_l = k_l \quad (l = 1, 2, \dots, n)$$

$$(3.9) \quad \sum_{l=1}^n y_{lj} = b_j \quad (j = 1, 2, \dots, p)$$

$$(3.10) \quad \begin{aligned} x_{il} &\geq 0 & (i = 1, 2, \dots, m) \\ y_{lj} &\geq 0 & (l = 1, 2, \dots, n) \\ s_l &\geq 0 & (j = 1, 2, \dots, p) \end{aligned}$$

$$(3.11) \quad \min \left(\sum_{i=1}^n \sum_{l=1}^m c_{il} x_{il} + \sum_{j=1}^p \sum_{l=1}^n d_{lj} y_{lj} \right).$$

A közönséges szállítási problémához hasonlóan ezt is könnyen ábrázolhatjuk egy egyszerű sémával, amely azonban két egymásutáni szállításról lévén szó két téglalap alakú táblázatból áll, ezek azonban nem függetlenek egymástól. Az összefüggést az s_i pótváltozók kettős szereplése jelenti, amely azt biztosítja, hogy pontosan ugyanannyit szállítsunk el egy közbülső állomásról, mint amennyit odaszállítottunk.

Az 1. táblázaton láthatók az említett résztáblázatok is, amelyeken könnyen látható, hogy a (3.6)–(3.10) követelmények itt azt jelentik, hogy soronként, illetőleg oszloponként összeadva a változókat a sor, ill. oszlop szélére írt számot kell megkapnunk.

Az egészet együtt tekintve is emlékeztet egy szállítási problémára, amely $(m + n) \cdot (n + p)$ változót tartalmaz, ezek közül azonban valójában csak $mn + n + np$ létezik. Természetesen a feladat megoldható volna úgy is, hogy ezeket a változókat hozzá vesszük, ∞ hozzárendelt költséggel, ezáltal azonban a változók száma $(m + n)(n + p) - (mn + n + np) = n^2 - n + mp$ -vel növekedne, amely különösen nagy n (vagy több mint két fokozatú probléma) esetén jelentős többletmunkát eredményezne.

Most vizsgáljuk meg a (3.6)–(3.11) lineáris programozási feladat mátrixát, azaz azt a mátrixot, amely a (3.6)–(3.9) egyenlőségek bal oldalán szereplő x_{ij} , y_{ij} és s_i ismeretlenek együtthatóiból áll. Könnyebb áttekinthetőség kedvéért, valamint, hogy a bizonyítások megértését megkönnyítsük, ezt a mátrixot részletesen kiírjuk:

(3.12)

[illegible]

α_1	x_{11}	x_{12}	$\dots$	x_{1n}
α_2	x_{21}	x_{22}	$\dots$	x_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
α_n	x_{n1}	x_{n2}	$\dots$	x_{nn}

k_1	s_1
k_2	s_2

 $\vdots$ k_n

s_n

 $k_1 \quad k_2$ k_n

y_{11}	y_{12}	$\dots$	y_{1p}
y_{21}	y_{22}	$\dots$	y_{2p}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_{n1}	y_{n2}	$\dots$	y_{np}

 $b_1 \quad b_2$ $\dots$ b_p

1. táblázat

A (3.12) összefüggésben az A mátrix többi helyén zérusok állnak.

A mátrix tehát csupa olyan oszlopból áll, amelyik két egyest tartalmaz. Ezenkívül két részmátrixa hagyományos szállítási feladatmátrix. Az $m \times n$ és $n \times p$ szállítási mátrixot az utolsó n oszlop kapcsolja össze.

Ezek után vizsgáljuk meg az A mátrix néhány tulajdonságát. Jelöljük az x_{il} , y_{lj} , s_i változók együttthatóiból álló vektorokat rendre $\mathbf{p}_i$, $\mathbf{q}_{lj}$ és $\mathbf{r}_i$ -l.

1. Tétel. Az A mátrix rangja, ha $n \geq 2$,

$$(3.13) \quad r(A) = m + 2n + p - 1.$$

Bizonyítás. Mivel a mátrix rangja a lineárisan független oszlopvektorok (vagy sorvektorok) maximális száma, könnyen beláthatjuk, hogy

$$(3.14) \quad r(A) < m + 2n + p.$$

Ehhez csak annyit kell igazolni, hogy az A mátrix $m + 2n + p$ számú sorvektora együttesen nem lineárisan független. Adjuk össze az A mátrix sorait úgy, hogy előzőleg az első m sort pozitív, a következő n sort negatív, az utána levő n sort ismét pozitív, majd az utolsó p sort negatív előjellel lássuk el, ahogy azt az A mátrix (3.12) felírásánál a zárójelbe tett előjelek mutatják. Az eredmény nullvektor, így a kérdéses mátrix sorvektorai nem lineárisan függetlenek, tehát a (3.14) egyenlőtlenséget beláttuk.

Most igazoljuk, hogy

$$(3.15) \quad r(A) > m + 2n + p - 2.$$

Jelöljük az A mátrix első mn oszlopából álló részmátrixot A_1 -gyel. Az első mn oszlop, mint már említettük, egy közönséges $m \times n$ méretű szállítási probléma mátrixa (ha az utolsó $n + p$ koordinátát, melyek mind zérusok, elhagynánk). Ismeretes, hogy az A_1 mátrix rangja $m + n - 1$. Ezt azonban könnyen ki is mutathatjuk. Az A_1 mátrix első m sorának pozitív és a következő n sorának negatív előjellel vett összege nullvektor, így $r(A_1) \leq m + n - 1$. Ha azonban kiválasztjuk az első n oszlopvektorát, valamint minden további n -es csoportból az első, könnyen belátható, hogy ezek lineárisan függetlenek, azaz $r(A_1) = m + n - 1$. Hasonlóan az A_2 mátrixra, mely az $mn + 1$ -edik oszloptól az

$mn + np$ -edik oszlopig terjed: $r(A_2) = n + p - 1$. Ez annyit jelent, hogy az A_1 és A_2 mátrixból kiválaszthatunk $m + n - 1$, ill. $n + p - 1$ lineárisan független oszlopvektort, amelyek természetesen együttesen is lineárisan függetlenek, hiszen amelyik sorban az egyik mátrixban a nemzérus elemek állnak, ott a másikban a zérus elemek, és viszont. Együttesen tehát van $m + 2n + p - 2$ lineárisan független oszlopvektorunk, amelyekről be kell látnunk, hogy az A mátrixra nézve nem alkotnak bázist, azaz van olyan oszlopvektor, amely nem fejezhető ki lineáris kombinációik segítségével. Ez az oszlopvektor nem kerülhet ki természetesen az A_1 vagy A_2 mátrixból, hiszen onnan egy-egy bázist választottunk. Ezért tekintsük kéldául az s_1 változó együttható-vektorát $\mathbf{r}_1$ -et, amely az $(m + 1)$ -edik és $(m + n + 1)$ -edik hely kivételével csupa zérust tartalmaz az említett pozíciókban pedig egyesek állnak. Jelöljük az A_1 -ből kiválasztott bázis indexhalmazát B_1 -gyel, az A_2 -ből kiválasztott bázisét B_2 -vel, a két bázis tehát

$$(3.16) \quad \mathbf{p}_{il} (i, l) \in B_1 \quad \text{és} \quad \mathbf{q}_{lj} (l, j) \in B_2.$$

Tegyük fel indirekt módon, hogy az

$$(3.17) \quad \mathbf{r}_1 = \sum_{(i,l) \in B_1} \lambda_{il} \mathbf{p}_{il} + \sum_{(l,j) \in B_2} \mu_{lj} \mathbf{q}_{lj}$$

előállítás mégis létezik, valamilyen λ_{il} és μ_{lj} konstansokkal. Az $m + 1$ -edik helyen szereplő egyes előállításában azonban csak az első összeg vesz részt, hiszen a $\mathbf{q}_{lj}$ vektorok első $m + n$ komponense zérus. Eszerint, ha (3.17) fennáll, akkor

$$(3.18) \quad \mathbf{e}_{m+1} = \sum_{(i,l) \in B_1} \lambda_{il} \mathbf{p}_{il}$$

ahol $\mathbf{e}_{m+1}(e_{m+1,1}; \dots; e_{m+1,m+2n+p})$ az $(m + 1)$ -edik egységvektort jelenti.

Minthogy a $\mathbf{p}_{il}$ vektorok mindegyikének pontosan két eleme 1-es, mégpedig egyik az első m elem, a másik pedig az $m + 1, \dots, m + n$ -edik elemek valamelyike, ezért a (3.18) vektoregyenlőség első m sorát összeadva

$$(3.19) \quad \sum_{(i,l) \in B_1} \lambda_{il} = \sum_{k=1}^m e_{m+n,k} = 0$$

az $m + 1, \dots, m + n$ -edik sorát összeadva pedig

$$(3.20) \quad \sum_{(i,l) \in B_1} \lambda_{il} = \sum_{k=m+1}^{m+n} e_{m+1,k} = 1.$$

Mivel (3.19) és (3.20) ellentmond egymásnak, befejeztük annak a bizonyítását, hogy $\mathbf{r}_1$ nem állítható elő az A_1 -ből választott $m + n - 1$ és az A_2 -ből választott $n + p - 1$ elemű, összességében is lineárisan független vektor segítségével, így beláttuk, hogy az $m + 2n + p - 2$ elemű lineárisan független vektorrendszer még nem bázis, azaz (3.15) igaz. Azonban (3.14) és (3.15) együtt a tétel állítását bizonyítja.

Megjegyzés. Ugyanúgy belátható, hogy az A mátrix bármelyik sorát elhagyva ismét egy olyan mátrixhoz jutunk, melynek rangja $m + 2n + p - 1$.

Egy mátrix minormátrixáról (A_k) beszélünk akkor, ha az eredeti mátrix tetszőleges számú sorát és ugyanannyi oszlopát kiválasztva, az ezek keresz-

teződéseiben álló elemek összességét tekintjük (az eredeti sorrendben!). A minormátrix determinánsát az eredeti mátrix minorának nevezzük.

2. Tétel. Az A mátrix minden minorának értéke 0 , $+1$ vagy -1 .

Bizonyítás. Az A mátrix minden oszlopában pontosan két egyes szerepel, ezért egy tetszőleges minorának minden oszlopában 0 , 1 vagy 2 egyes szerepelhet. A minorokat három csoportba oszthatjuk:

a) Van olyan oszlop, amelyben nincsen egyes, ekkor értéke 0 .

b) Minden oszlopában pontosan két egyes van. Ekkor könnyen beláthatjuk, hogy a minor értéke ismét zérus. Vizsgáljuk meg ugyanis, hogy melyik sora való az A mátrix első m sorából, ennek előjelét hagyjuk pozitívnak. A következő n sorból valót lássuk el negatív előjellel, a további n sorbeli előjele ismét pozitív marad, majd az utolsó p sorbeliét megint negatívra változtatjuk, mint az A mátrix (3.12) felírásánál látható. Ha az így előjelezett sorokat összeadjuk, zérus vektort kapunk, ugyanis minden egyes eredeti párja szerepel és a másik előjelű csoportba esik, így összegük zérus. Ezzel beláttuk, hogy az ilyen típusú minorok értéke is zérus.

c) Végül tekintsük azt az esetet, amikor bizonyos oszlopokban egy egyes van, a többiben kettő. Ekkor fejtük ki a determináns az egyedülálló egyesek szerint. Egy kifejtés után a minor abszolút értéke változatlan, legfeljebb az előjele változhat meg:

$$|A_k| = \pm |A_{k-1}|.$$

A kifejtést addig folytatjuk, amíg van olyan oszlop, amelyben egy egyes van. Így vagy eljutunk egy egyetlen 0 , ill. 1 -ből álló minorig, melynek értéke 0 , ill. $+1$ (ekkor az eredeti minor értéke ennek megfelelően 0 , ill. ± 1), vagy eljutunk az a), ill. b) esetek valamelyikéhez, amikor is az eredeti minor értéke 0 .

Mivel az a), b) és c) az összes esetet tartalmazza, a bizonyítást befejeztük.

Mint azt az 1. tételben láttuk, az A mátrix rangja $m + 2n + p - 1$, azaz a bázis, melynek segítségével A összes oszlopvektorát kifejezhetjük lineáris kombinációként, A pontosan ennyi oszlopvektorából fog állni. Jelöljük ismét az x_{il} , y_{lj} , s_l változók együtthatóvektorait rendre $\mathbf{p}_{il}$, $\mathbf{q}_{lj}$, $\mathbf{r}_l$ -vel. Legyen továbbá

$$(3.21) \quad G = \{g_{\alpha, \beta}\} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{m+2n+p-1}] \quad \begin{matrix} (\alpha = 1, 2, \dots, m+2n+p) \\ (\beta = 1, 2, \dots, m+2n+p-1) \end{matrix}$$

az A oszlopvektoraiból álló tetszőleges bázis. Ezen vektorok lineáris kombinációjaként kifejezzük az összes együttható vektort:

$$(3.22) \quad \begin{aligned} (a) \quad G \mathbf{h}_{il} &= \mathbf{p}_{il} & (b) \quad G \mathbf{z}_{lj} &= \mathbf{q}_{lj} & (c) \quad G \mathbf{w}_l &= \mathbf{r}_l \\ (i &= 1, 2, \dots, m; l = 1, 2, \dots, n; j = 1, 2, \dots, p), \end{aligned}$$

ahol $\mathbf{h}_{il}$, $\mathbf{z}_{lj}$ és $\mathbf{w}_l$ a megfelelő lineáris kombinációs együtthatók.

3. Tétel. Az $\mathbf{h}_{il}$, $\mathbf{z}_{lj}$ és $\mathbf{w}_l$ vektorok minden komponense 0 , $+1$ vagy -1 .

Bizonyítás. Az állítás helyességét csak $\mathbf{h}_{il}$ -re mutatjuk be, a másik két típusú vektor esetén a bizonyítás teljesen hasonlóan történik.

A $\mathbf{p}_{il}$ vektor pontosan két egyest tartalmaz, egyiket az i -edik, másikat az $(m+l)$ -edik helyen. Ha elhagyjuk a (3.22(a)) egyenletrendszer $(m+l)$ -edik

sorát, azaz a G mátrix $(m + l)$ -edik sorát (a kapott mátrixot H -val jelöljük) és a $\mathbf{p}_{il}$ vektor $(m + l)$ -edik komponensét, akkor a

$$(3.23) \quad H \mathbf{h}_{il} = \mathbf{e}_i$$

egyenletrendszerhez jutunk, ahol $\mathbf{e}_i$ az i -edik egységvektort jelenti. A H mátrix nonszinguláris ($|H| \neq 0$), ugyanis az 1. Tétel utáni megjegyzés szerint elhagyhatjuk az eredeti feladatból az A mátrix tetszőleges — jelen esetben az $m + l$ -edik — sorát, rangja továbbra is $m + 2n + p - 1$ lesz. Mivel a G -nek megfelelő H a redukált rendszer bázisa marad (az $m + l$ -edik komponensek elhagyása előtti előállítás fennáll a sor törlése után is), így rangja $m + 2n + p - 1$. Mivel ez a szám megegyezik sorainak, ill. oszlopainak számával, tehát H nonszinguláris, ahogy állítottuk, tehát létezik inverze. Így a (3.23) rendszerből:

$$(3.24) \quad \mathbf{h}_{il} = H^{-1} \mathbf{e}_i$$

$\mathbf{h}_{il}$ tehát a H^{-1} inverzmátrix i -edik oszlopát jelenti. Minthogy azonban H és minora az eredeti, A mátrixnak minora, $|H| = +1$, vagy -1 (H nonszinguláris, tehát $|H| \neq 0$), ezenkívül H összes minora is 0, $+1$ vagy -1 értékű. Az inverz mátrix elemeit azonban úgy kaphatjuk meg, hogy a megfelelő elemhez tartozó aldeterminánst osztjuk a teljes mátrix determinánsával. Innen az inverz mátrix minden eleme csak 0, $+1$ vagy -1 , ami érvényes $\mathbf{h}_{il}$ komponenseire is, és ezzel állításunkat igazoltuk.

Megjegyzés. Ismeretes, hogy a bázis segítségével történő előállítás egyértelmű, így a fenti tétel egyben azt is jelenti, hogy minden vektor csak úgy állítható elő, hogy a bázisból veszünk bizonyos vektorokat pozitív, másokat negatív előjellel, és összeadjuk őket.

4. Az algoritmus ismertetése

Tekintsünk most egy tetszőleges megengedett bázist. (Azaz egy olyan bázist, amely segítségével a (3.6)–(3.9) egyenletrendszer jobb oldalán álló vektort kifejezve a bázisvektorok együtthatójaként — azaz x_{il} , y_{lj} és s_l számára — nemnegatív értékek adódnak, amint azt a (3.10) feltételben megköveteltük.)

$$(4.1) \quad \mathbf{p}_{il} (i, l) \in B_1 \quad \mathbf{q}_{lj} (l, j) \in B_2 \quad \mathbf{r}_l \quad l \in B_3$$

ahol B_1 , B_2 és B_3 a bázisban levő $\mathbf{p}_{il}$, $\mathbf{q}_{lj}$, illetve $\mathbf{r}_l$ indexpárjainak, ill. indexeinek halmazát jelöli. Ezen bázis segítségével szeretnénk kifejezni az A mátrix egy tetszőleges oszlopvektorát, mondjuk $\mathbf{p}_{i_0 l_0}$ vektort. Legyen ez a keresett kifejezés az alábbi:

$$(4.2) \quad \mathbf{p}_{i_0 l_0} = \sum_{(i,l) \in B_1} \lambda_{il}^0 \mathbf{p}_{il} + \sum_{(l,j) \in B_2} \mu_{lj}^0 \mathbf{q}_{lj} + \sum_{l \in B_3} \nu_l^0 \mathbf{r}_l.$$

Tudjuk, hogy valamennyi λ_{il}^0 , μ_{lj}^0 , ν_l^0 csak 0, $+1$ vagy -1 lehet (3. Tétel). Ezenkívül a $\mathbf{p}_{i_0 l_0}$ vektort is ismerjük: az i_0 -adik és $m + l_0$ -adik eleme egyes, a többi zérus. Kell lenni tehát legalább egy olyan vektornak a (4.2) előállításban, amelynek i_0 -adik komponense szintén egyes. Ez a bázisváltozó, mint a (3.12)-ből könnyen kiolvasható, egy olyan változót jelent, amely $x_{i_0 l_0}$ -al egy

sorban van az 1. táblázatban. Legyen ez az oszlopvektor $\mathbf{p}_{i_0 l_1}$, a hozzátartozó együttható $\lambda_{i_0 l_1} = 1$. Ekkor azonban a (4.2) előállításban van egy -1 együtthatójú vektor, amelyiknek egy egyes komponense az $(m + l_1)$ -edik helyen van, mint $\mathbf{p}_{i_0 l_1}$ -nek, azaz $x_{i_0 l_1}$ -gyel egy oszlopban van a következő változó. Így például az alábbi előállítás adódhat:

$$(4.3) \quad \mathbf{p}_{i_0 l_0} = \mathbf{p}_{i_0 l_1} - \mathbf{p}_{i_1 l_1} + \mathbf{p}_{i_1 l_2} - + \dots - \mathbf{p}_{i_{\alpha} l_{\alpha}} + \mathbf{p}_{i_{\alpha} l_{\alpha}}.$$

Természetesen az előállításban szerepelhet $\mathbf{r}_l$ és $\mathbf{q}_{lj}$ típusú oszlopvektor is, például lehet, hogy a báziselőállítás ilyen alakú:

$$(4.4) \quad \mathbf{p}_{i_0 l_0} = \mathbf{p}_{i_0 l_1} - \mathbf{r}_{l_1} + \mathbf{q}_{l_1 j_2} - \mathbf{q}_{l_2 j_2} + \dots - \mathbf{q}_{l_{\alpha} j_{\alpha}} + \mathbf{r}_{l_{\alpha}} - \mathbf{p}_{i_{\alpha} l_{\alpha}} + \\ + \mathbf{p}_{i_{\alpha} l_{\alpha+1}} - + \dots - \mathbf{p}_{i_{\beta} l_{\beta}} + \mathbf{p}_{i_{\beta} l_{\beta}}.$$

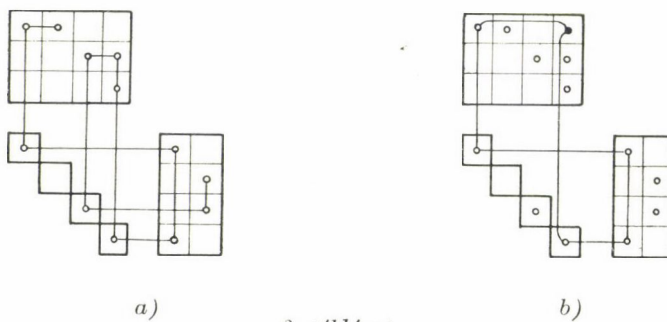
Hogy ezeket az előállításokat jobban megvilágítsuk, tekintsünk egy konkrét példát. Legyen $m = 3$, $n = 4$, $p = 2$. A bázis pedig álljon a következő vektorokból (B_1 , B_2 , B_3 definícióját lásd (4.1) után):

$$B_1 = \{(1, 1), (1, 2), (2, 3), (2, 4), (3, 4)\}$$

$$B_2 = \{(1, 1), (2, 2), (3, 2), (4, 1)\}$$

$$B_3 = \{1, 3, 4\}.$$

Az 1. táblázatnak megfelelő 2. táblázatban kis körrel jelöltük a bázis elemeit. Ha a sorok, illetve oszlopok mentén összekötjük a báziselemeket, látjuk, hogy nincs benne körút (2a. táblázat). Az együtthatómátrix segítségével (3. táblázat, ill. általánosan a (3.12) mátrix) könnyen belátható, hogy a körút a 2. táblázatban azt jelenti, hogy a megfelelő együtthatóvektorok lineárisan összefüggők, ha pedig nincs körút, akkor az együtthatóvektorok lineárisan függetlenek. Mivel a kijelölt változók együtthatóvektorai lineárisan függetlenek, és számuk $m + 2n + p - 1 = 12$, valóban bázist alkotnak.



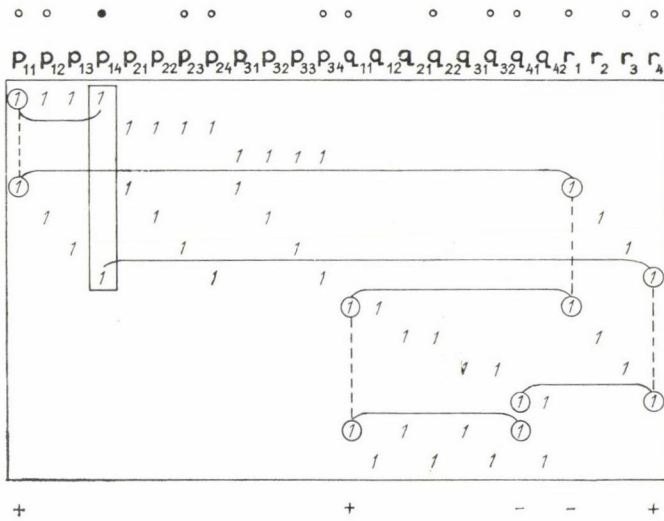
2. táblázat

Állítsuk elő például az x_{14} változó együttható vektorát. (Ezt a táblázatban ponttal jelöltük.) Mint a rajzból is világos, az előállítás a következő:

$$(4.5) \quad \mathbf{p}_{14} = \mathbf{p}_{11} - \mathbf{r}_1 + \mathbf{q}_{11} - \mathbf{q}_{41} + \mathbf{r}_4$$

Ugyanis a fenti megjegyzés értelmében ha találunk egy hurkot, amely tartalmazza az x_{14} változót, akkor a megfelelő együtthatóvektorok lineárisan össze-

függnek. Az együtthatómátrixból (3. táblázat) az is világos, hogy ekkor a (4.5) előállítás érvényes.



3. táblázat

Ha a berajzolt utat követve az utolsó sorban levő előjelekkel összeadjuk a vektorokat, pontosan a $\mathbf{p}_{14}$ vektort kapjuk.

Természetesen a 2. táblázat kisebb mérete és jobb áttekintése miatt alkalmasabb számolásra, mint a 3.

Ha már egy tetszőleges, a bázison kívüli x_{ij} , y_{ij} vagy s_i változó együtt-hatávektorát előállítottuk, akkor következhet a simplex módszer általános tárgyalásában szereplő $c_{ij} - z_{ij}$ kiszámítása, amely megmutatja, hogy a megfelelő változót bevonjuk a bázisba, fog-e csökkenni a célfüggvény, vagy nem. A c_{ij} jelenti a kiválasztott változóhoz tartozó költséget, z_{ij} pedig a következőt:

$$(4.6) \quad z_{ij} = \mathbf{c}_B B^{-1} \mathbf{a}_{ij}$$

ahol $\mathbf{a}_{ij}$ a változó együtt-hatávektora, B a bázisvektorok együtteséből álló mátrix, $\mathbf{c}_B$ pedig a bázisbeli változókhoz tartozó költségek. Az erre vonatkozó részletek megtalálhatók pl. KREKÓ BÉLA [4] és HADLEY [2] könyvében. $B^{-1}\mathbf{a}_{ij}$ azonban nem más, mint a vizsgált változó együtt-hatávektorának kifejezése a bázisvektorokkal, jelen esetben 0, +1 és -1 számokból álló vektor. A korábban elmondottak szerint (lásd például (4.3)–(4.5) előállításokat) a +1 és -1 együtt-hatával szereplő változók a vizsgált változóval együtt, amelynek költségtényezője +1 együtt-hatával szerepel, együttesen egy hurkot alkotnak (például 2. táblázat). Könnyen belátható, hogy ez a hurok olyan tulajdonságú, hogy az egymásra következő élei mindig merőlegesek, ezenkívül minden sorból vagy oszlopból legfeljebb két változót tartalmaz, és a vizsgált bázison kívüli változó együtt-hatója úgy állítható elő, hogy az említett hurok mentén szereplő változók együtt-hatávektorait váltakozva +1 és -1-gyel szorozzuk és összeadjuk. A fenti 0, +1, -1 komponensekből álló vektort kell

megszorozni skalárisan a bázisváltozókhoz tartozó költségekkel, azaz például a (4.3) előállítás esetében

$$(4.7) \quad z_{i_0 l_0} = c_{i_0 l_1} - c_{i_1 l_1} + c_{i_1 l_2} - + \dots + c_{i_a l_a} + c_{i_a l_0}$$

vagy a (4.4) előállításnál, figyelembe véve, hogy az s_l pótváltozóhoz (kihasználatlan kapacitás) rendelt költség 0 az y_{lj} szállításokhoz rendelt költség pedig d_{lj} :

$$(4.8) \quad z_{i_0 l_0} = c_{i_0 l_1} - 0 + d_{l_1 j_2} - d_{l_2 j_2} + - \dots - d_{l_a j_a} + \\ + 0 - c_{i_a l_a} + - \dots - c_{i_\beta l_\beta} + c_{i_\beta l_0}.$$

Vagy a konkrét példában a (4.5) előállításnál

$$(4.9) \quad z_{14} = c_{11} - 0 + d_{11} - d_{41} + 0.$$

Az általános szimplex módszerből ismeretes, hogy ha minden változóra a $c_{ij} - z_{ij} \geq 0$, akkor az optimumnál vagyunk, amennyiben pedig van olyan változó, amelyre $c_{ij} - z_{ij} < 0$, akkor annak a bázisba való bevonásával a célfüggvény értéke csökkenthető, amennyiben degeneráció nem áll elő. Degenerációról beszélünk akkor, ha a bázisban szereplő $m + 2n + p - 1$ számú változó nem mind pozitív, hanem zérus is van közöttük. Minden oszlopvektorra a báziselőállítást természetesen csak $r(A) = m + 2n + p - 1$ lineárisan független oszlopvektor segítségével tudjuk elvégezni, tehát a bázisnak ennyi vektorból kell állni, még akkor is, ha némelyik bázisváltozó zérus. A degeneráció esetével a 7. pontban foglalkozunk.

Az eddig elmondottakra tekintsünk most egy számpéldát. Mint a fenti példában, legyen $m = 3$, $n = 4$, $p = 2$. A költségeket a 4. táblázatban tüntettük fel.

Legyenek továbbá az elszállítandó mennyiségek, a korlátok és a felvevőhelyek igényei rendre a következők:

$a_1 = 6$	$k_1 = 4$	
$a_2 = 6$	$k_2 = 4$	$b_1 = 6$
$a_3 = 3$	$k_3 = 6$	$b_2 = 9$
15	$k_4 = 7$	15

Könnyen belátható, hogy az 5. táblázaton feltüntetett

$$x_{11} = 2, \quad x_{12} = 4, \quad x_{23} = 5, \quad x_{24} = 1, \quad x_{34} = 3$$

$$s_1 = 2, \quad s_3 = 1, \quad s_4 = 3$$

$$y_{11} = 2, \quad y_{22} = 4, \quad y_{32} = 5, \quad y_{41} = 4$$

egy megengedett bázismegoldás.

Számítsuk ki például az x_{44} -re vonatkozó $c_{ij} - z_{ij}$ mennyiséget. Mint már korábban láttuk

$$(4.10) \quad p_{14} = p_{11} - r_1 + q_{11} - q_{41} + r_4$$

$$(4.11) \quad z_{14} = c_{11} - 0 + d_{11} - d_{41} + 0$$

5	4	9	1
3	2	3	2
4	2	1	5

0							
	0						
		0					
			0				
				0			

5	4
1	3
3	6
2	1

4. táblázat

6	②	4					
6			5	1			
3					3		

4	②					②	
4							4
6			1				5
7				③		④	

4 4 6 7 6 9

5. táblázat

és innen

$$c_{14} - z_{14} = c_{14} - c_{11} + 0 - d_{11} + d_{41} - 0 = 1 - 5 + 0 - 5 + 2 - 0 = -7$$

tehát az x_{14} változót érdemes bevonni a bázisba. Ha az x_{14} változó értékét egy egységgel növeljük, akkor a célfüggvény értéke 7 egységgel csökken. Jelöljük $\mathbf{b}$ -vel az $(a_1, a_2, a_3, k_1, k_2, k_3, k_4, k_1, k_2, k_3, k_4, b_1, b_2)$ komponensekből álló $m + 2n + p = 13$ dimenziós oszlopvektort. Ekkor a (3.6)–(3.9) egyenlőségek a jelen példánkban, a bázisvektorok segítségével felírva a következőképpen alakultak:

(4.12)

$$2\mathbf{p}_{11} + 4\mathbf{p}_{12} + 5\mathbf{p}_{23} + \mathbf{p}_{24} + 3\mathbf{p}_{34} + 2\mathbf{q}_{11} + 4\mathbf{q}_{22} + 5\mathbf{q}_{32} + 4\mathbf{q}_{41} + 2\mathbf{r}_1 + \mathbf{r}_3 + 3\mathbf{r}_4 = \mathbf{b}.$$

A (4.10) egyenlőséget nullára redukálva, és λ -val megszorozva hozzáadjuk (4.12)-hez, azaz a $\mathbf{p}_{14}$ oszlopvektort λ szinten behoztuk a $\mathbf{b}$ vektor előállításába:

$$(2 - \lambda)\mathbf{p}_{11} + 4\mathbf{p}_{12} + 5\mathbf{p}_{23} + \mathbf{p}_{24} + 3\mathbf{p}_{34} + (2 - \lambda)\mathbf{q}_{11} + 4\mathbf{q}_{22} + 5\mathbf{q}_{32} + (4 + \lambda)\mathbf{q}_{41} + (2 + \lambda)\mathbf{r}_1 + \mathbf{r}_3 + (3 - \lambda)\mathbf{r}_4 + \lambda\mathbf{p}_{14} = \mathbf{b}.$$

Korábban láttuk már, hogy a $\mathbf{p}_{14}$ vektor egységnyi bevonása (azaz x_{14} egységnyi növelése) 7 egységgel csökkenti a célfüggvényt. Tehát λ értékét minél nagyobbra kell választani. Ennek azonban határt szab, hogy minden változónak nemnegatívnak kell maradnia, tehát

$$2 - \lambda \geq 0, \quad 2 - \lambda \geq 0, \quad 3 - \lambda \geq 0.$$

Ennek alapján $x_{14} = \max \lambda = 2$ értéket vehet fel legfeljebb. Ha az új bázist kiszámítjuk, akkor a 6. táblázathoz jutunk. Figyelembe vettük azonban azt is,

0	4		2
		5	1
			3

4							
			1				
				1			

			4				
			5				
		6					

6. táblázat

hogy a bázisnak továbbra is $m + 2n + p - 1 = 12$ vektorból kell állnia, tehát mivel két bázisváltozó vált egyszerre zérussá, az egyiket (x_{11}) nulla szinten benthagytuk a bázisban.

A most elmondottak szó szerint megismételhetők teljes általánosságban, ha a **b** vektor komponensei helyébe a (3.6)–(3.9) egyenletrendszer jobb oldalait tesszük, a (4.10) bevonandó vektor előállítás helyébe pedig az általános (4.3) ill. (4.4) előállítást.

Részletesebben kifejtve, kiszámítjuk minden változóhoz a $c_{ij} - z_{ij}$ értéket. Az általános lineáris programozásból ismeretes (l. id. művek), hogy a bázisváltozókhoz tartozó $c_{ij} - z_{ij} = 0$. Ha minden $c_{ij} - z_{ij} \geq 0$, akkor a vizsgált bázismegoldás optimális. Ha találunk olyan változót, amelyhez tartozó $c_{ij} - z_{ij} < 0$, akkor ezt a változót vonjuk be a bázisba. Így a célfüggvény értéke, amennyiben degeneráció nem lép fel (l. 7. pont), csökken. A báziscsere úgy történik, hogy elvégezzük a bevonandó változó előállítását a bázisváltozók segítségével, ill. a fent említett — a bevonandó változót tartalmazó — hurkot. Figyelembe véve, hogy ezekben az előállításokban egy, az előállítandó vektort tartalmazó, de egyébként csupa bázisvektorból álló hurok mentén váltakozva + és – előjelet vesznek fel az egyes vektorok, az új bázis meghatározása igen könnyű. Az új változó meghatározandó értékét jelöljük λ -val, akkor a hurokban rákövetkező változó értéke $-\lambda$ -val változik, majd a következő ismét $+\lambda$ -val stb. Mivel a hurok egymásra következő élei egymásra merőlegesek (azaz ha egy változót az előzővel egy sorban választottunk, akkor a következőt vele egy oszlopban választjuk stb.), könnyen belátható, hogy visszaérve az új változó másik szomszédja is $-\lambda$ értékkel változik, vagyis mindegy, hogy a hurkon milyen irányban indulunk el. A maximális javítást adó, tehát a legnagyobb λ értéke ezek alapján úgy határozható meg, hogy a változók nem-negativitása miatt azon régi változók közül, amelyekhez $-\lambda$ járult, a legkisebbet vesszük. Egyúttal meghatároztuk a kieső bázisváltozót is. Ha egyszerre több bázisváltozó válik zérussá, akkor csak egyet hagyunk el, a többit zérus szinten a bázisban hagyjuk.

5. Induló bázismegoldás keresése

A fenti eljárást csak akkor alkalmazhatjuk, ha már egy megengedett bázismegoldás áll a rendelkezésünkre. Most bemutatunk egy módszert induló bázismegoldás keresésére. A közönséges szállítási problémára vonatkozó bázis-kereső eljárások csak akkor használhatók minden további nélkül, ha külön-külön kereshetünk $m + n - 1$, ill. $n + p - 1$ elemű bázisokat, azaz mivel összesen $m + 2n + p - 1$ elemű a bázis, ez csak úgy lehetséges, hogy csak egy s_i legyen a bázisban. Ennek azonban az a feltétele, hogy

$$(5.1) \quad \sum_{l=1}^n k_l - \sum_{i=1}^m a_i = s_k \leq k_n \leq \max_{1 \leq l \leq n} k_l$$

teljesüljön. Ha ez teljesül, akkor a két szállítási problémát külön megoldjuk, és az optimális megoldásokra vonatkozóan folytatjuk együttesen a korábban elmondott eljárást. Ha az (5.1) feltétel nem teljesül, akkor a következőképpen járhatunk el: Az s_i pótváltozókat rendre a lehető legnagyobbak választjuk,

míg a kihasználatlan kapacitásokat teljesen szétosztjuk közöttük, úgy, hogy legfeljebb egy legyen töredékesen, azaz

$$\sum_{i=1}^n k_i - \sum_{i=1}^m a_i = s_1 + s_2 + \dots + s_h + s_{h+1}$$

$$s_1 = k_1, s_2 = k_2, \dots, s_h = k_h, s_{h+1} \leq k_{h+1}$$

$$s_l = 0 \quad l > h + 1.$$

Ez annyit jelent, hogy az első $m \times n$ méretű probléma redukálódik egy $m \times (n - h)$ méretűre, mivel az első h oszlop összege $k_l - s_l = 0$ ($l \leq h$). Hasonlóan a másik feladat redukálódik egy $(n - h) \times p$ méretűre. A két feladatban a szokásos módszerek egyikével — például a északnyugati sarok módszerrel (l. például HADLEY [2]) — keresünk egy-egy $m + (n - h) - 1$, ill. $(n - h) + p - 1$ elemű bázist. A bázishoz vesszük még az s_l ($l = 1, 2, \dots, h + 1$) változókat. A megmaradó

$$(m + 2n + p - 1) - (m + n - h - 1 + n - h + p - 1 + h + 1) = h$$

számú bázisváltozót, melyek nulla szinten lesznek a bázisban, úgy választjuk, hogy a bázisváltozók együttható vektorai összességükben lineárisan függetlenek legyenek.

Természetesen most is kezdhethetjük a két részprobléma külön-külön történő megoldásával, és csak az optimumok elérése után folytatva együtt, azonban minél szélsőségesebb az eloszlás, annál kevésbé valószínű, hogy az együttes optimumnak akár csak a közelébe is jutottunk.

Az eljárást a 8. pontban egy számpéldán is bemutatjuk.

6. A potenciál módszer

A hosszadalmas hurok keresések helyett — mint a közönséges szállítási problémánál — könnyen kidolgozható egy egyszerűbb mechanikus eljárás z_{ij} kiszámítására.

Ismeretes a szimplex módszer általános tárgyalásából, hogy $c_{ij} - z_{ij} = 0$ minden bázisváltozóra vonatkozóan fennáll. Rendeljünk hozzá a többfokozatú szállítási probléma minden sorához ill. oszlopához egy u_i , ill. v_j potenciált, mégpedig úgy, hogy minden elemre a megfelelő sor és oszlop potenciálok összege szolgáltatja a z_{ij} mennyiséget, azaz

$$(6.1) \quad z_{ij} = u_i + v_j \quad \begin{pmatrix} i = 1, 2, \dots, m, \dots, m + n \\ j = 1, 2, \dots, n, \dots, n + p \end{pmatrix}.$$

Később belátjuk, hogy ez lehetséges. Ilyen választás mellett a bázisváltozókra $c_{ij} - z_{ij} = 0$ miatt

$$(6.2) \quad c_{ij} = u_i + v_j \quad (i, j) \in B_0$$

ahol a B_0 a bázishoz tartozó indexpárokat jelenti és

$$(6.3) \quad \begin{aligned} c_{ij} &= d_{i-m, j-n}, & \text{ha } i > m; & \quad j > n \\ c_{ij} &= 0, & \text{ha } i = m + l; & \quad j = l \quad (l = 1, 2, \dots, n). \end{aligned}$$

A (6.2) egyenletrendszer segítségével, mivel c_{ij} költségek ismertek, a z_{ij} mennyiségek kiszámíthatók. Minthogy az egyenletek száma egyenlő a bázisváltozók számával, $(m + 2n + p - 1)$ -gyel, az ismeretlenek száma pedig ennél eggyel nagyobb, egy potenciált (az u_i és v_j ismeretlenek bármelyikét) tetszőlegesen megválaszthatunk. Minthogy a bázisváltozók között nincsen hurok, a tetszőlegesen megválasztott potenciál segítségével a (6.2) egyenletrendszerből minden potenciál egyértelműen meghatározható. Ezek segítségével a (6.1) egyenlőségek alapján a z_{ij} , valamint a $c_{ij} - z_{ij}$ indikátorok is meghatározhatók. Természetesen a modell szerkezete miatt (lásd pl. 1. táblázat) csak az

$$(6.4) \quad \begin{array}{ll} i \leq m & j \leq n \\ i = m + l & j = l \quad (l = 1, 2, \dots, n) \\ i > m & j > n \end{array}$$

eseteknek megfelelő $c_{ij} - z_{ij}$ mennyiségeket kell kiszámítani, mert csak ezek jelentenek tényleges változókat. Emiatt a táblázatot összevonhatjuk, lásd 8. pont.

Azt kell még belátnunk, hogy a (6.1) összefüggés a bázison kívüli változókra is igaz. (A bázisváltozók esetében azért nem kell erről meggyőződni, mert úgy határoztuk meg z_{ij} mennyiségeket, hogy (6.1) teljesüljön.)

Ezt az állítást könnyen beláthatjuk, ha figyelembe vesszük a z_{ij} (4.7), ill. (4.8) előállítását. Ezek most a (6.3) egyszerűsítés miatt minden esetben így írhatók:

$$(6.5) \quad z_{ij} = c_{ik} - c_{lk} + c_{lm} - + \dots - c_{pr} + c_{pj},$$

ahol az összes szereplő $c_{\alpha\beta}$ költségek bázisváltozókhoz tartoznak, tehát igaz rájuk az, hogy

$$(6.6) \quad c_{\alpha\beta} = u_{\alpha} + v_{\beta}.$$

Felhasználva (6.6) előállítást, (6.5) így írható:

$$z_{ij} = (u_i + v_k) - (u_l + v_k) + (u_l + v_m) - + \dots - (u_p + v_r) + (u_p + v_j).$$

A zárójelek felbontása és az összevonások elvégzése után a kívánt

$$z_{ij} = u_i + v_j$$

összefüggést kapjuk a (6.4) esetekben, hiszen csak ekkor létezik a felhasznált (6.5) előállítás. Azonban ez az összes változót jelenti a probléma szerkezete miatt. Tehát a bázison kívüli változókra is igazoltuk a (6.1) összefüggést, amivel a módszer teljes.

7. Degeneráció

Degenerációnak nevezzük azt az esetet, amikor a bázisban szereplő változók között zérus is van. Másképpen fogalmazva, amikor az A mátrix rangjánál, jelen esetben $m + 2n + p - 1$ -nél kevesebb pozitív változó van.

Például a korábban felírt példában a 6. táblázat degenerációt mutat. Ha például az x_{21} változót akarjuk bevonni, amelyhez a 4. költségtáblázat alapján a

$$c_{21} - z_{21} = 3 - 2 + 1 - 5 = -3$$

negatív érték tartozik, akkor azonnal látszik, hogy a

$$P_{21} - P_{24} + P_{14} - P_{11} = 0$$

előállítás miatt csak

$$\min(x_{24}, x_{11}) = \min(1, 0) = 0$$

szinten vonhatjuk be, azaz a célfüggvény értéke változatlan.

Eddig a ciklus, azaz a javítást nem hozó lépések ciklikus ismétlődésének elkerülését az biztosította, hogy a célfüggvény értéke minden lépésben csökkent. Degeneráció esetében nem ez a helyzet.

Bár a gyakorlati esetekben még nem fordult elő ciklus a szállítási problémák megoldása során, azonban nem biztos, hogy ez nem következhet be.

A degeneráció csak akkor következhet be, ha az a_i és a $k_l - s_l$ mennyiségek már egy részhalmazának összege is megegyezik.*

Tudjuk, hogy

$$(7.1) \quad \sum_{i=1}^m a_i = \sum_{l=1}^n (k_l - s_l).$$

Azonban előfordulhat, hogy létezik egy oly indexhalmaz pár, hogy

$$(7.2) \quad \begin{aligned} I &\subset \{1, 2, \dots, m\} \\ L_1 &\subset \{1, 2, \dots, n\} \end{aligned}$$

mégis

$$(7.3) \quad \sum_{i \in I} a_i = \sum_{l \in L_1} (k_l - s_l).$$

Hasonlóan előfordulhat, hogy

$$(7.4) \quad \sum_{l \in L_2} (k_l - s_l) = \sum_{j \in J} b_j, \quad \text{ahol} \quad \begin{aligned} L_2 &\subset \{1, 2, \dots, n\} \\ J &\subset \{1, 2, \dots, p\}. \end{aligned}$$

Ezt elkerülendő, ha degeneráció lép fel, illetőleg ha degenerációs ciklussal találkozunk, akkor bevezetjük az

$$(7.5a) \quad a_1 + p\varepsilon, a_2 + p\varepsilon, \dots, a_m + p\varepsilon$$

$$(7.5b) \quad k_1, k_2, \dots, k_n + m\varepsilon$$

$$(7.5c) \quad b_1 + m\varepsilon, b_2 + m\varepsilon, \dots, b_p + m\varepsilon$$

új szállítandó mennyiségeket ill. korlátokat.

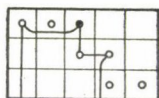
Található olyan kis ε_0 , hogy $\varepsilon < \varepsilon_0$ esetén a (7.5) rendszerrel már nem következhet be (7.3) vagy (7.4). Ugyanis az a_i , $k_l - s_l$, b_j számokból csak véges sok részhalmaz választható ki, az ε pedig végtelen sok értéket vehet fel.

* A degeneráció ugyanis azt jelenti, hogy van zérus értékű is a bázisváltozók között. Ha ezt az elemet elhagyjuk, a bázis legalább két részre bomlik, úgy, hogy a különböző csoportokból nem választható úgy ki egy-egy bázisváltozó, hogy azok egy sorban vagy egy oszlopban legyenek. Ellenkező esetben ugyanis a zérus értékű bázisváltozó elhagyása előtt a bázisban hurok lett volna, ami lehetetlen a bázisváltozók együtthatóinak lineáris függetlensége miatt.

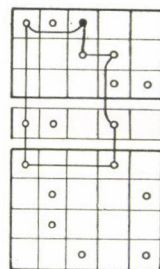
Feltéve tehát, hogy $0 < \varepsilon < \varepsilon_0$ tetszőlegesen kicsiny pozitív szám, az új rendszerben degeneráció nem léphet fel, tehát ciklus sem. Az ε értékét nem is szükséges konkrétan megadni. A konkrét számításokra vonatkozóan lásd pl. HADLEY [2] 299–304. oldal.

8. Egy numerikus példa

A továbbiakban az eljárás megvilágítása céljából kiszámítunk egy szám-példát. Legyen $m = 3, n = 5$ és $p = 4$. Jobb helykihasználás és áttekinthetőség kedvéért az eddig tárgyalt típusú 7. táblázat helyett a 8. táblázat formájában fogunk számolni, ami abban különbözik, hogy az előző táblázat második mátrixának transzponáltját vettük, és hozzáillesztettük — az s_h pótváltozók számára fenntartott egy sor kihagyásával — az előző mátrixhoz. Egy hurok, mint azt a táblázatokban feltüntettük, abban különbözik a korábbitól, hogy az első mátrixot továbbra is csak függőlegesen hagyhatjuk el, azonban a második mátrixba is csak függőlegesen léphetünk be a megfelelő pótváltozó közbeiktatásával.



7. táblázat



8. táblázat

A kidolgozandó feladat költségmátrixa legyen a következő (9. táblázat.)

a_i						
5	5	2	3	7	4	
7	7	8	2	9	5	
4	6	3	8	5	4	
	6	8	5	3	6	k_e
	0	0	0	0	0	
	7	8	5	9	4	7
	9	9	3	6	7	2
	2	4	7	5	6	3
	3	2	5	8	6	4
						b_j

9. táblázat

Az első mátrix bal oldalán az a_i kínálatokat, a második mátrix jobb oldalán a b_j keresleteket és végül a két mátrix között a közbülő állomások korlátait tüntettük fel. Természetesen az s_i pótváltozókhoz tartozó költségek

zérusok, ezt tüntettük fel a két mátrix közötti második sorban. Mivel a változók sorösszegei változatlanok (a melléírt a_i vagy b_j számmal egyenlők) szabad a 9. táblázat minden sorából tetszőleges számot levonni, ez az egyes megengedett programokhoz rendelt célfüggvény értékek sorrendjét nem változtatja meg, a jobb (olcsóbb) program a levonások utáni mátrixszal számolva is jobb lesz. Jelöljük ugyanis x_{ij} -vel az i -edik sor j -edik ismeretlenét ($i = 1, 2, \dots, m + p + 1$; $j = 1, 2, \dots, n$) és c_{ij} -vel a megfelelő költséget. Használjuk továbbá az $a_{m+j+1} = b_j$ ($j = 1, 2, \dots, p$) jelölést, és $\{x'_{ij}\}$, valamint $\{x''_{ij}\}$ jelentse a feladat két tetszőleges megengedett megoldását, amelyekre

$$(8.1) \quad z' = \sum_i \sum_j c_{ij} x'_{ij} < \sum_i \sum_j c_{ij} x''_{ij} = z''.$$

Vonjuk le most a c_{ij} költségmátrix i -edik sorának minden eleméből egy M_i számot ($i = 1, 2, \dots, m + p + 1$), amely természetesen negatív is lehet. Ekkor azt kell belátnunk, hogy az $\{x'_{ij}\}$ és $\{x''_{ij}\}$ programokra az új költségekkel a (8.1) összefüggésben szereplő irányú egyenlőtlenség áll fenn. Ezt az alábbi átalakítással mutatjuk ki, felhasználva, hogy $\sum_{j=1}^n x_{ij} = a_i$ minden megengedett megoldásra:

$$(8.2) \quad \sum_i \sum_j (c_{ij} - M_i) x'_{ij} = z' - \sum_i M_i (\sum_j x'_{ij}) = z' - \sum_i M_i a_i$$

$$(8.3) \quad \sum_i \sum_j (c_{ij} - M_i) x''_{ij} = z'' - \sum_i M_i (\sum_j x''_{ij}) = z'' - \sum_i M_i a_i.$$

Tehát minden programhoz tartozó célfüggvényérték ugyanazzal a konstanssal változott, amivel állításunkat beláttuk. Megjegyzendő, hogy ugyanezt az oszlopokkal nem tehetjük, mivel az oszlop összeg nem állandó. Vonjuk le tehát a 9. táblázat minden sorából az ott szereplő legkisebb elemet:

5	3	0	1	5	2
4	5	6	0	7	3
7	3	0	5	2	1
6 8 5 3 6					
0 0 0 0 0					
3	4	1	5	0	7
6	6	0	3	4	2
0	2	5	3	4	3
1	0	3	6	4	4

10. táblázat

5	5				
4	1	3			
7		5	2		
6 8 2 0 0 $k_j - s_j$					
0 0 3 3 6 s_j					
6	1				7
	2				2
	3				3
	2	2			4

11. táblázat

5					
1	3				
	5	2	0		
		3	3	6	
6	1				0
	2				
	3				
	2	2			

$$z = 94$$

12. táblázat

Az 5. pontban leírtak szerint keresünk egy induló bázist (11. táblázat), figyelembe véve, hogy az itt használt jelölésekkel

$$(8.4) \quad \sum_{i=1}^m x_{ij} = k_j - s_j = \sum_{i=m+2}^{m+p+1} x_{ij} \quad (s_j = x_{m+1,j}).$$

Ezt kiegészítjük két további, zérus szinten bevont változóval, hogy bázist kapjunk (12. táblázat). Az x_{34} és x_{55} bevonása azért célszerű, mert így bázist

kapunk és kis költségű változókat vontunk be ($c_{14} = 2$ és $c_{55} = 0$). A 12. táblázat pedig mutatja, hogy az így kiválasztott változók között nincs hurok, tehát valóban bázist kaptunk.

Most külön-külön megoldjuk a 12. táblázat két, szaggatott vonallal leválasztott szállítási problémáját a 4. pontban leírt hurok módszerrel:

I.

3	0	1
5	6	0
3	0	5

c_{ij}

5	•	
1	3	
	5	2

2	3	
4		•
	5	2

4	1	•
2	•	2
•	7	•

II.

3	4	1
6	6	0
0	2	5
1	0	3

c_{ij}

6	1	
	2	•
	3	
	2	2

6	1	•
		2
	3	
	4	0

6	1	0
		2
•	3	
	4	

3	4	0
•	•	2
3	•	•
•	4	•

12. táblázat

Most megoldjuk az együttes optimalizálási feladatot az így kapott részoptimumok mint induló bázis segítségével. Az első lépésben hurok módszerrel bevonjuk az s_1 pótváltozót, majd potenciál módszerrel (l. 6. pont) folytatjuk a megoldást.

III.

3	0	1	5	2
5	6	0	7	3
3	0	5	2	1
0	0	0	0	0

3	4	1	5	0
6	6	0	3	4
0	2	5	3	4
1	0	3	6	4

c_{ij}

III./1

4	1		
2	2		
7	0		
•	3	3	6

3	4	0	0
	2		
3			
4			

12. táblázat

A következőkben egy táblázatsor egy lépést jelent. Mindig felírjuk a jelenlegi bázisváltozókhoz tartozó c_{ij} költségeket, és az $u_i + v_j = z_{ij}$ összefüggések segítségével $u_1 = 0$ -ból kiindulva megkapjuk a potenciálokat, majd a $c_{ij} - z_{ij} = c_{ij} - (u_i + v_j)$ összefüggés segítségével kiszámítjuk a $c_{ij} - z_{ij}$ indikátorokat. Ha ezek között van negatív, akkor a megfelelő változót bevonjuk a bázisba. Ezt a számítást természetesen a pótváltozókra is elvégezzük. A fenti hurok segítségével bevont változóval kapjuk a következő megengedett bázismegoldást. (A megoldások alatt mindig feltüntetjük a célfüggvény hozzátartozó értékét). Az eljárást az optimumig folytatjuk. (Minden $c_{ij} - z_{ij} \geq 0$.)

III./2

4	1		
		4	
7		0	

2		1	3	6
---	--	---	---	---

1	4	2		0
		2		
3				
	4			

 $z = 33$

$u_1 = 0 \ -3 \ 2 \ -1 \ 3$

3	3	0		
-2			0	
3		0	2	

0		0	0	0
---	--	---	---	---

3	3	4	1	0
2			0	
0	0			
-1		0		

$v_j/u_i: 0 \ 1 \ -2 \ 1 \ -3$
($i > 5$)

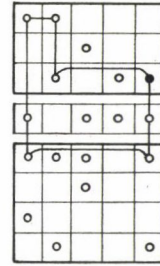
$u_i \ (i \leq 5)$

0	0	-4	3	-4
7	11	0	10	2
0	0	0	0	(-5)

0	2	0	0	0
---	---	---	---	---

0	0	0	1	0
4	3	0	0	5
0	1	7	2	7
2	0	6	6	8

$c_{ij} = z_{ij}$



III./3

3	2		
		4	
6		0	1

3		1	3	5
---	--	---	---	---

	4	2		1
		2		
3				
	4			

 $z = 28$

3	3	0		
3			0	
3		0	2	1

0				
---	--	--	--	--

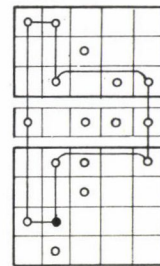
-2		4	1	0
-3			0	
0	0			
-6		0		

$0 \ 6 \ 3 \ -1 \ 2$

0	0	1	1	1
2	6	0	3	2
0	0	5	0	0

0	0	0	0	0
---	---	---	---	---

5	0	0	8	0
9	3	0	7	5
0	(-4)	2	4	2
7	0	6	13	8



III./4

	5		
		4	
3		0	4

6		1	3	2
---	--	---	---	---

	1	2		4
		2		
0	3			
	4			

 $z = 16$

$0 \ 1 \ 1 \ 3 \ 2$

-1		0		
-1			0	
-1		0	2	1

0		0	0	0
---	--	---	---	---

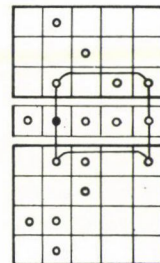
2		4	1	0
1			0	
0	0	2		
-2		0		

$0 \ 2 \ -1 \ -3 \ -2$

4	0	1	3	1
6	6	0	5	2
4	0	5	0	0

0	(-3)	0	0	0
---	------	---	---	---

1	0	0	4	0
5	9	0	5	5
0	0	6	6	6
3	0	6	11	10



III./5

	5			
		4		
	2		0	5

6	1	1	3	1
---	---	---	---	---

		2		5
		2		
0	3			
	4			

 $z = 13$

$0 \ -2 \ -2 \ 0 \ -1$

2		0		
2			0	
2		0	2	1

0	0	0	0	0
---	---	---	---	---

-1			1	0
-2			0	
0	0	2		
-2		0		

$0 \ 2 \ 2 \ 0 \ 1$

1	0	1	3	1
3	6	0	5	2
1	0	5	0	0

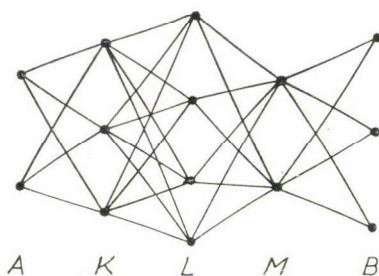
0	0	0	0	0
---	---	---	---	---

4	3	0	6	0
8	6	0	5	5
0	0	3	3	3
3	0	3	8	5

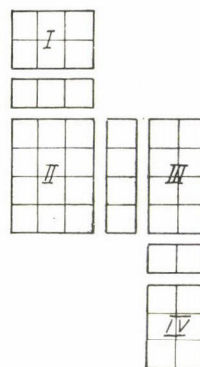
Minden $c_{ij} - z_{ij} \geq 0$ Optimum!

9. Megjegyzések

Ez a megoldási módszer értelemszerűen kettőnél több fokozatú szállítási probléma esetén is alkalmazható. Például a 3. ábrán látható feladathoz tartozó mátrix elrendezést a 4. ábra mutatja, ilyen elrendezésben kell megadni a szállítási költséget is, és a változók aktuális értékeit. Az első csoportból (A) a másodikba (K) szállítandó mennyiségeket kifejező változók az I mátrixba kerültek. A második csoportból (K) a harmadikba (L) szállítandó mennyiségeket a II . mátrixba írjuk stb.



3. ábra



4. ábra

Minden egyes közbülső csoportba (K , L , M) tartozó állomáshelynek adott kapacitása van. Az egyes mátrixok közé kerülnek azok a pótváltozók, amelyek azt mutatják, hogy az egyes helyeken mennyivel kevesebbet szállítunk keresztül, mint az ottani kapacitás.

A dolgozatban leírt algoritmus változtatás nélkül alkalmazható. Most az egyes hurkok természetesen több mátrixra is kiterjedhetnek, azonban minden mátrixból csak a vele szomszédosba mehetünk át és csak egy közöttük levő pótváltozón keresztül.

A vizsgált feladat felmerülhet egészen más jellegű problémánál is. Például, ha egy üzem termelését akarjuk megszervezni, amelyben bizonyos munkafolyamatok elvégzésére több gép is alkalmas, azonban a megmunkálás ideje vagy költsége nem azonos. Ekkor az egyes szállítási fokozatokat az azonos munkát végző gépek jelentik, amelyeket megmunkálási sorrendben vázolunk a 3. ábrához hasonlóan. A szállítási költségekhez (amely esetleg elenyészően kicsi lehet a többi költséghez képest) hozzászámítjuk a megfelelő termelési költséget is. Ekkor a feladat megoldásaként megkapjuk, hogy melyik gépen milyen mennyiségű anyagot munkáljunk meg, hogy az egyes gépkapacitásokat ne haladja meg, és a teljes termelési költség minimális legyen. (Költség helyett az időt is számolhatjuk.)

Természetesen előfordulhat, hogy egy későbbi fázisban a gyártmány (termék) ugyanoda kerül, ahol már egyszer volt, azaz a 3. ábrán látható különböző pontok azonos várost, gyárat vagy gépet reprezentálnak. Ez a megoldás szempontjából semmi problémát nem okoz. Például, ha egy gabona-tároló helyen őrlésre is lehetőség van, akkor a megfelelő szállítási költség zérus.

(Beérkezett: 1964. december 5.)

IRODALOM

- [1] DANTZIG, G. B.: *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey 1963.
- [2] HADLEY, G.: *Linear Programming*. Addison-Wesley, Reading, Massachusetts. 1962.
- [3] SZWARC, W.: *The Applications of the Transportation Scheme*. Carnegie Institute of Technology, Research Papers 1963.
- [4] KREKÓ B.: *Lineáris programozás*. Közgazdasági és Jogi Könyvkiadó, 1962.
- [5] SCHELL, E. D.: „Distribution of a Product of Several Properties” Proceedings of the Second Symposium in Linear Programming. (Directorate of Management Analysis and National Bureau of Standards, Washington, D. C.) 1955. 615—642.

ЗАДАЧА МНОГОСТУПЕНЧАТОГО (ТРАНЗИТНОГО) ТРАНСПОРТА

Л. Б. КОВАЧ

Резюме

В статье рассматривается задача транспорта такого в котором имеются не только исходные и конечные станции, а также промежуточные места через которых совершается транспорт. Точнее, имеются исходные станции $A_1, \dots, A_m$, промежуточные пункты: $K_1, \dots, K_n; L_1, \dots, L_r; \dots$ и места назначения: $B_1, B_2, \dots, B_p$. Условия задачи следующие:

(а) Исходная станция A_i предлагает количество $a_i (i = 1, \dots, m)$; спросом места назначения B_j является $b_j (j = 1, \dots, n)$ а емкостью промежуточных пунктов являются $k_s (s = 1, \dots, n)$, $l_t (t = 1, \dots, r)$ ит. д.

(б) Перевозку какого-нибудь места можно производить только до какого-нибудь места последующей группы. Специфические расходы для допустимых перевозок заданы. Задачей является сделать минимальными полные расходы для перевозок. Эту проблему можно было разрешить, если представить ее в виде мультииндексной задачи (см. например [5]) или, если исключить как перевозки с бесконечными транспортными расходами все недопустимые перевозки как например в статье [3]. Для обоих методов получается сравнительно очень большое число переменных. Например, если имеются m исходные станции, p места назначения и в отдельных группах $n_1, n_2, \dots, n_q$ промежуточные станции, тогда при применении первого метода в выкладках участвуют $mn_1 \dots pn_q$ переменные, а при втором методе число переменных равно $(m + n_1 + \dots + n_q)(n_1 + \dots + n_q + p)$. В настоящей статье представляется алгоритм, основывающийся на классической транспортной задаче, такой что в нем число переменных равно $mn_1 + n_1 + n_1n_2 + \dots + n_2 + \dots + pn_q$.

Проблема переходит из задачи из практики: Требуется перевозить зерно в мельницы из амбаров, а муку из мельниц к мест назначения так, чтобы суммарная затрата средств для этой цепи была минимальной. (Количество муки измеряется в единицах зерна). Применение результатов к этой задаче, а также к другим задачам рассматриваются в статье подробно.

За исключением п. 9 в статье рассматривается случай трех групп — места исходные, места назначения и одна группа промежуточных станций, однако алгоритм применим без всяких изменений также и в случае неличия многих групп промежуточных станций.

TRANSIT TRANSPORTATION PROBLEM

by

L. B. KOVÁCS

Summary

This paper deals with a transportation problem which has not only origins and destinations but some other places in which the transportation passes through. More exactly there are origins $A_1, \dots, A_m$, one or more group of other places $K_1, \dots, K_n; L_1, \dots, L_r; \dots$ and destinations $B_1, \dots, B_p$. The conditions of the problem are the following:

(a) The supply of the origin A_i is a_i ($i = 1, \dots, m$), the demand of the destination B_j is b_j ($j = 1, \dots, n$); the capacities of the other places k_s ($s = 1, \dots, n$), l_t ($t = 1, \dots, r$) etc.

(b) It is allowed to transport from any place only to the next group of places. The unit costs of the permitted shipments are given.

The problem is to minimize the total transportation cost. It is possible to solve this problem as a multi-index problem. (see for example [5]) or by excluding the non-permitted transportations with ∞ costs as in [3]. In both these methods the number of variables are relatively very large. For example if there are m origins p destinations and $n_1, n_2, \dots, n_q$ places in the other groups then the first method needs for $mn_1 \dots n_q p$ variables and the second one needs for $(m + n_1 + \dots + n_q)(n_1 + \dots + n_q + p)$ variables. The present paper guarantees an algorithm — based on the classical transportation problem — having $mn_1 + n_1 + n_1 n_2 + n_2 + \dots + n_q p$ variables.

The problem appeared as a practical task of minimal total cost transportation of corn from granaries to mills and flour from mills to the destinations. (The flour is measured in corn measure.)

The above application and other ones are written in the paper in details.

Except the point 9 the paper deals with the case having three groups at places — origins, destinations and one group of intermediate places — but the algorithm is suitable without any changes for the case having more groups of places.