Kéri Gerzson

Módosított „stepping-stone“ algoritmus a szállítási probléma megoldására

1. *Bevezetés.* A szállítási probléma egyik megoldási módjául szolgál a „stepping-stone" algoritmus, vagy más elnevezéssel disztribuciós eljárás. Ez lényegében a lineáris programozás szimplex módszere, de nagymértékben kihasználja a szállítási probléma sajátos szerkezetében rejlő egyszerűsítési lehetőségeket. (Az utóbbi tény indokolja a külön elnevezést.) A „stepping-stone" algoritmus szokásos tárgyalásmódja a módszer kézi számolás esetén előnyös leírását tartalmazza. Dolgozatom célja kettős: egyrészt a „stepping-stone" módszernél alkalmazható további egyszerűsítésre kívánom felhívni a figyelmet, másrészt olyan irányban szándékozom kidolgozni az algoritmust, amely a szállítási táblától elszakadva, kézi számolás helyett a gépi megoldás szempontjait helyezi előtérbe. Ezenkívül kidolgozott példával illusztrálom a módosított „stepping-stone" módszer mindkét (kézi, ill. gépi megoldásra szánt változatát.

A fenti szakaszban említett egyszerűsítésre az a felismerés vezetett, hogy a $\delta_{ij} = z_{ij} - c_{ij}$ számok minden báziscsere alkalmával úgy transzformálódnak, hogy bizonyos sorokhoz hozzáadódik, egyúttal bizonyos oszlopokból levonódik ugyanaz a szám, mégpedig a bázisba bevonandó cellához tartozó δ_{ij} szám.

2. *Jelölések és definíciók.* A szállítási probléma matematikai megfogalmazásban:

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

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1, 2, \ldots, m) \tag{1}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1, 2, \ldots, n)$$

$$\left(\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \right)$$

$$x_{ij} \geq 0 \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$$

A szállítási tábla $m \cdot n$ téglalap alakban elrendezett cellából áll. Az (i, j) cellához hozzá van rendelve a c_{ij} célfüggvényegyüttható és a szállítási probléma mátrixának

$$\alpha_{ij} = e_i + e_{m+j}$$

oszlopvektora.

Cellagráf: Az $m \cdot n$ cella halmazának tetszőleges részhalmazát kiválasztjuk. Ezek a cellagráf pontjai. E részhalmaz két celláját akkor és csak akkor kötjük össze éllel, ha vagy egy sorban vagy egy oszlopban vannak. Az egy sorban levő cellákat összekötő éleket vízszintes, az egy oszlopban levő cellákat összekötő éleket függőleges vonalaknak képzeljük el.

Egyszerű út: olyan cellagráf, amely merőlegesen csatlakozó élekből és ezek végpontjaiból áll, továbbá minden sorból és minden oszlopból legfeljebb két cellát tartalmaz.

Egyszerű hurok: olyan egyszerű út, amelynek kezdő és végpontja azonos, kezdő és végéle pedig merőleges egymásra. (Így lényegtelen, hogy melyik cellát tekintjük kezdő és egyúttal végpontnak.)

Egy cellagráf *összefüggő*, ha az $m \cdot n$-es tábla minden sorában és minden oszlopában van hozzá tartozó cella, és a cellagráf bármely két cellája összeköthető egyszerű úttal.

Egy cellagráf *fa*, ha hurokmentes és összefüggő.

3. *A szimplex módszer alkalmazása.* Ismertnek tekintjük a következő fontosabb eredményeket:

A szállítási probléma mátrixának bizonyos α_{ij} oszlopvektorai akkor és csak akkor alkotnak bázist, ha a megfelelő (i, j) cellák a szállítási táblában olyan cellagráfot alkotnak, amely fa. Így egy cellarendszert *báziscellarendszernek* nevezünk, ha a cellagráfja fa.

Ha egy báziscellarendszert kibővítünk egy további cellával, akkor a kibővített cellarendszer gráfja tartalmaz egy és csak egy hurkot. Ez a hurok egyszerű és áthalad az újonnan bevont cellán.

A szimplex módszer alkalmazása során adott báziscellarendszer esetén minden (i, j) cellához hozzárendelünk egy

$$\delta_{ij} = z_{ij} - c_{ij}$$

számot a következő módon: Ha (i, j) báziscella, akkor $\delta_{ij} = 0$. Egyébként tekintsük azt az egyszerű hurkot, amely áthalad az (i, j) cellán és rajta kívül csak báziscellákon halad át. Legyenek e hurok pontjai sorrendben

$$(i, j) = (i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_3), \ldots, (i_{k-1}, j_k), (i_k, j_k) = (i, j). \tag{2}$$

Ekkor

$$\delta_{ij} = c_{i_1, j_2} - c_{i_2, j_2} + c_{i_2, j_3} - \ldots + c_{i_{k-1}, j_k} - c_{ij}. \tag{3}$$

Ha most az adott báziscellarendszerből elhagyjuk az

$$(i_1, j_2), (i_2, j_3), \ldots, (i_{k-1}, j_k)$$

cellák valamelyikét,[1] mondjuk az (i_r, j_{r+1}) cellát, és bevonjuk helyette az (i, j) cellát, akkor újabb bázisrendszerhez jutunk. A stepping-stone algoritmus során mindig ilyen báziscseréket végzünk. A bázisba bevonandó cella tetszőleges olyan (i, j) cella lehet, amelyre a megfelelő δ_{ij} szám aktuális értéke pozitív. Ha $\delta_{ij} \leq 0$ minden (i, j) párra, akkor az optimális megoldásnál vagyunk. A bázisból kilépő cella (i_r, j_{r+1}) koordinátáit az

$$x_{i_r, j_{r+1}} = \min_{1 \leq s \leq k-1} x_{i_s, j_{s+1}} \tag{4}$$

[1] Itt csak az (i, j) cellától a hozzá tartozó hurok mentén páratlan távolságra levő báziscellák jöhetnek szóba, tehát a (2) alatti cellák közül minden második szerepel.

kritérium jelöli ki, ahol $x_{\alpha\beta}$ az (α, β) báziscellához tartozó bázisváltozó aktuális értéke. A bázisváltozók a báziscsere után a következő értékeket veszik fel:

$$
\begin{aligned}
x'_{i_s, j_{s+1}} &= x_{i_s, j_{s+1}} - x_{i_r, j_{r+1}} && (s = 1, 2, \ldots, r-1, r+1, \ldots, k-1, \\
x'_{i_s, j_s} &= x_{i_s, j_s} + x_{i_r, j_{r+1}} && (s = 2, 3, \ldots, k-1), \\
x'_{i_1, j_1} &= x_{i_r, j_{r+1}},
\end{aligned} \tag{5}
$$

és $x'_{\alpha\beta} = x_{\alpha\beta}$ minden más (α, β) báziscellára.

4. *A δ_{ij} számok megadása duál változók segítségével.* Tekintsük az

$$u_\alpha + v_\beta = c_{\alpha\beta} \tag{6}$$

egyenleteket minden olyan (α, β) párra, amelyre (α, β) báziscella. Legyenek az u_i, v_j számok $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ a (6) rendszer egy tetszőleges megoldása.

Ismeretes, hogy ekkor

$$\delta_{ij} = u_i + v_j - c_{ij}. \tag{7}$$

5. *A δ_{ij} számok transzformációja a báziscserék során.* Legyen H egy báziscellarendszer. Tekintsük a H-hoz tartozó u_i, v_j, δ_{ij} számokat, és legyen u'_i, v'_j, δ'_{ij} ezek új értéke a 3. szakaszban leírt báziscsere után, másszóval a H' báziscellarendszerben, amely abban különbözik H-tól, hogy az (i_r, j_{r+1}) cella helyett az (i_1, j_1) cellát tartalmazza.

Definiáljuk a bizonyos sor-, ill. oszlopindexekből álló C és D halmazokat a következőképpen:

a) Ha az i_r-edik sorban H csak az (i_r, j_{r+1}), azaz H' csak az (i_1, j_1) báziscellát tartalmazza ($r = 1$ szükségszerűen), akkor legyen $C = \{i_r\}$, $D = \varnothing$.

b) Ha a j_{r+1}-edik oszlopban H csak az (i_r, j_{r+1}) báziscellát tartalmazza, akkor legyen $C = \{1, 2, \ldots, m\}$, $D = \{1, 2, \ldots, j_{r+1} - 1,\ j_{r+1} + 1, \ldots, n\}$.

c) Ha az $(i_r,\ j_{r+1})$ cella sorában is, oszlopában is van további báziscella, akkor tekintsük azt a csak H celláiból álló maximális összefüggő H_1 cellagráfot, amely az (i_r, j_r) cellát tartalmazza, de az (i_r, j_{r+1}) cellát nem. Legyen most C és D a H_1 halmaz két vetülethalmaza, azaz

$$C = \{i : (i, j) \in H_1 \text{ valamely } j\text{-re}\},$$

$$D = \{j : (i, j) \in H_1 \text{ valamely } i\text{-re}\}.$$

Bebizonyítjuk, hogy az

$$
\begin{aligned}
u'_i &= \begin{cases} u_i - \delta_{i_1, j_1} & \text{ha } i \in C \\ u_i & \text{ha } i \notin C, \end{cases} \\
v'_j &= \begin{cases} v_j + \delta_{i_1, j_1} & \text{ha } j \in D \\ v_j & \text{ha } j \notin D \end{cases}
\end{aligned} \tag{8}
$$

számok az

$$u_\alpha + v_\alpha = c_{\alpha\beta} \qquad (\alpha, \beta) \in H' \tag{9}$$

rendszer egy megoldását alkotják.

A bizonyítás az *a)* esetben nagyon egyszerű. $(i, j) \in H \cap H'$ esetén ugyanis $i \neq i_r$, tehát $i \notin C$, $j \notin D$, így (8) és (6) miatt $u'_i + v'_j = u_i + v_j = c_{ij}$, az új (i_1, j_1) báziscellára pedig $i_1 = i_r \in C$, $j_1 \notin D$, tehát (8) és (7) miatt $u'_{i_1} + v'_{j_1} = (u_{i_1} - \delta_{i_1,j_1}) + v_{j_1} = c_{i_1,j_1}$.

A *b)* eset bizonyítása analóg. Végül a *c)* esetben $(i, j) \in H \cap H'$ típusú új báziscellákra $(i, j) \in H_1$ esetén $i \in C$ és $j \in D$, így (8) és (6) miatt $u'_i + v'_j = (u_i - \delta_{i_1,j_1}) + (v_j + \delta_{i_1 j_1}) = u_i + v_j = c_{ij}$, $(i, j) \notin H_1$ esetén viszont a H_1 gráf maximál-tulajdonsága miatt az (i, j) cellának sem a sorában, sem az oszlopában nem lehet H_1-hez tartozó cella, így $i \notin C$, $j \in D$, továbbá ismét (8) és (6) miatt $u'_i + v'_j = u_i + v_j = c_{ij}$. Belátandó még, hogy $u'_{i_1} + v'_{j_1} = c_{i_1,j_1}$. Mivel az (i_1, j_2) cellát H_1 tartalmazza, $i_1 \in C$. Másrészt $j_1 \notin D$. Tegyük fel ugyanis ennek ellenkezőjét, vagyis azt, hogy $j_1 \in D$. Ekkor azt a következtetést vonhatjuk le, hogy az i_1-edik sorban is, a j_1-edik oszlopban is van H_1-hez tartozó báziscella, $(i_1, j_1) \notin H_1$ miatt ezek egymástól különbözőek, H_1 összefüggő volta miatt összeköthetők egy csak H_1 celláin áthaladó úttal. Vegyük hozzá ehhez az úthoz az (i_1, j_1) cellát. Ekkor olyan hurokhoz jutunk, amely az (i_1, j_1) cellán kívül csak H-beli báziscellákat tartalmaz, de az (i_r, j_{r+1}) cellát nem tartalmazza. Ez ellentmond az (i_1, j_1) cellán és ezenkívül csak báziscellákon áthaladó hurok egyértelműségének. Következésképpen $j_1 \notin D$, továbbá (8) és (7) miatt $u'_{i_1} + v'_{j_1} = (u_{i_1} - \delta_{i_1,j_1}) + v_{j_1} = c_{i_1,j_1}$.

Beláttuk, hogy (8) tekinthető az u_i, v_j duál változók transzformációs formuláinak, következésképpen a δ_{ij} számok transzformációját a

$$\begin{aligned} &\delta'_{ij} = \delta_{ij} && \text{ha } i \in C \text{ és } j \in D, \\ & && \text{vagy } i \notin C \text{ és } j \notin D, \\ (10)\qquad &\delta'_{ij} = \delta_{ij} - \delta_{i_1,j_1} && \text{ha } i \in C \text{ és } j \notin D, \\ &\delta'_{ij} = \delta_{ij} + \delta_{i_1,j_1} && \text{ha } i \notin C \text{ és } j \in D \end{aligned}$$

formulák adják meg.

Végül az olvasóra hagyom annak a bizonyítását, hogy a C, D halmazok az *a)*, *b)*, *c)* esetek bármelyikében elkészíthetők a következő algoritmikus úton: Jelöljük meg először az i_1-edik sort és az i_1-edik sorban levő (i_r, j_{r+1})-től különböző báziscellák oszlopait. Az általános lépés során jelöljük meg a közvetlenül előzőleg megjelölt oszlopokban levő báziscellák sorai közül azokat, amelyek még jelöletlenek, majd a most megjelölt sorokban levő (i_r, j_{r+1})-től különböző báziscellák oszlopai közül azokat, amelyek még jelöletlenek. Az eljárás véget ér, ha már nem tudunk ily módon újabb sort vagy újabb oszlopot megjelölni. A végeredményben megjelölt sorok, ill. oszlopok indexei fogják alkotni a C ill. D halmazt.

6. *Induló bázis keresése.* A szokásos módszernek egy olyan változatát alkalmazom, amelynél a báziscellákat úgy választjuk meg, hogy a hozzájuk tartozó c_{ij} értékek lehetőleg kicsik legyenek. Az u_i, v_j duál változók és a δ_{ij} számok kezdeti értékének meghatározása az induló bázis celláinak és a bázisváltozók értékének megadásával szimultán történik.

7. *Az algoritmus.* Újabb vagy az eddigiektől eltérő jelölések: a báziscellák regisztrálása az A_i, B_j halmazok segítségével történik, ahol

$$A_i = \{j : (i, j) \text{ báziscella}\} \quad (i = 1, 2, \ldots, m)$$

$$\text{és} \quad B_j = \{i : (i, j) \text{ báziscella}\} \quad (j = 1, 2, \ldots, n)$$

A bázisba bevonandó cella koordinátáit (r, s), a hozzátartozó hurok pontjait

$$(r, s) = (\alpha_1, \beta_{l1}), (\alpha_1, \beta_1), (\alpha_2, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_{l_1}, \beta_{l_1})$$

fogja jelölni; $(\alpha_{l_0}, \beta_{l_0})$ a bázisból kilépő cella. A kezdeti δ_{ij} értékek meghatározására szolgáló duál változók jelölésére elég két változó : u és v. A C_l, D_l halmazok sorozata a hurokkeresési eljárás számára szolgál.

Most következzenek az algoritmus lépései, az utasításokat zárójelen kívül, a magyarázatokat és megjegyzéseket szögletes zárójelek közt megadva. Más típusú zárójelek használata értelemszerű.

1° [*Báziskeresés és a δ_{ij} számok kezdeti értékének meghatározása.* r, s, C és D itt csak segédváltozók, a báziscserék során más, lényeges célra fogjuk használni ezeket a betűket.]

$A_i = \varnothing \ (i = 1, 2, \ldots, m)$, $B_j = \varnothing \ (j = 1, 2, \ldots, n)$

$c_{rs} = \min \{c_{ij} : i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n\}$ [r és s kijelölése számára. Ha a minimális c_{ij} nem egyértelmű, akkor mindegy hogy a lehetséges (r, s) párok közül melyiket választjuk.]

$C = \{r\}$, $D = \{s\}$, $A_r = \{s\}$, $B_s = \{r\}$

$a'_i = a_i \ (i = 1, 2, \ldots, m)$, $b'_j = b_j \ (j = 1, 2, \ldots, n)$

$\delta'_{ij} = -c_{ij} \ (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$

$x_{ij} = 0 \ (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$

$u = 0$, $v = c_{rs}$

2° $i = 1$, $j = 1$

3° $j = n$ esetén folytassuk 7°-nál, egyébként 4°-nél.

4° $a'_r < b'_s$ esetén folytassuk 6°-nál, egyébként 5°-nél.

5° $x_{rs} = b'_s$

$a'_r = a'_r - b'_s$

$\delta_{ks} = \delta_{ks} + v \ (k = 1, 2, \ldots, m)$

$j = j + 1$

$c_{rs} = \min \{c_{rl} : l \in \{1, 2, \ldots, n\} - D\}$ [r a régi, de s eddigi értéke törlendő, az utasítás s új értékének kijelölésére szolgál. Ha evvel s értékét nem egyértelműen jelöljük ki, akkor a lehetséges s-ek bármelyikét választhatjuk.]

$D = D \cup \{s\}$

$A_r = A_r \cup \{s\}$, $B_s = B_s \cup \{r\}$

$v = c_{rs} - u$

Térjünk vissza 3°-hoz.

6° $x_{rs} = a'_r$

$b'_s = b'_s - a'_r$

$\delta_{rl} = \delta_{rl} + u \ (l = 1, 2, \ldots, n)$

$i = i + 1$

$c_{rs} = \min \{c_{ks} : k \in \{1, 2, \ldots, m\} - C\}$ [s a régi, de r eddigi értéke törlendő, az utasítás r új értékének kijelölésére szolgál.]

$C = C \cup \{r\}$

$A_r = A_r \cup \{s\}$, $B_s = B_s \cup \{r\}$

$u = c_{rs} - v$

Térjünk vissza 3°-hoz.

7° $i < m$ esetén folytassuk 6°-nál, egyébként 8°-nál.

8° $x_{rs} = a'_r$

$\delta_{rl} = \delta_{rl} + u \ (l = 1, 2, \ldots, n)$
$\delta_{ks} = \delta_{ks} + v \ (k = 1, 2, \ldots, m)$

9° [*A bázisba bevonandó cella meghatározása.*]
$\delta_{rs} = \max \{\delta_{ij} : i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n\}$ [r és s kijelölésére]
$\delta_{rs} \leq 0$ esetén menjünk 34°-hez, egyébként 10°-hez.

10° [*Hurokkeresés és a bázisból kilépő cella meghatározása.*]
$C_1 = \{r\}$
$D_1 = A_r$
$l \ = 2$

11° $C_l = \cup \{B_j : j \in D_{l-1}\} - C_{l-1}$
$D_l = \cup \{A_i : i \in C_l\} - D_{l-1}$
$s \in D_l$ esetén folytassuk 13° nál, egyébként 12°-nél.

12° $l = l + 1$
Térjünk vissza 11°-hez.

13° $l_0 = l,\ l_1 = l$ [Értékük legfeljebb $\min \{m, n\}$.]
$\beta_l = s$
$\alpha_l \in B_{\beta_l} \cap C_l$ [Ez a metszet mindig egy elemből áll.]

14° $l = 1$ esetén folytassuk 17°-nél, egyébként 15°-nél.

15° $l = l - 1$
$\beta_l \in A_{\alpha_{l+1}} \cap D_l$
$\alpha_l \in B_{\beta_l} \cap C_l$ [Ezek is szükségképpen egy elemű halmazok.]
$x_{\alpha_l, \beta_l} \geq x_{\alpha_{l_0}, \beta_{l_0}}$ esetén folytassuk 14°-nél, egyébként 16°-nál.

16° $l_0 = l$
Menjünk vissza 14°-hez.
[A keresett hurok $(r, s) = (\alpha_1, \beta_{l_1}), (\alpha_1, \beta_1), (\alpha_2, \beta_1), \ldots, (\alpha_{l_1}, \beta_{l_1})$, a bázisba bekerül az (r, s) cella az $(\alpha_{l_0}, \beta_{l_0})$ cella helyett.]

17° [*A δ_{ij} számok transzformációja.*]
$l_0 \neq 1$ esetén folytassuk 20°-nál, egyébként 18°-nál.

18° $A_r \neq \{\beta_1\}$ esetén folytassuk 27°-nél, egyébként 19°-nél.

19° $\delta_{rj} = \delta_{rj} - \delta_{rs} \ (j = 1, 2, \ldots, n)$
Menjünk 32°-höz.

20° $l_0 \neq l_1$ esetén folytassuk 23°-nál, egyébként 21°-nél.

21° $B_s \neq \{\alpha_{l_1}\}$ esetén folytassuk 23°-nál,egyébként 22°-nél.

22° $\delta_{is} = \delta_{is} - \delta_{rs} \ (i = 1, 2, \ldots, m)$
Menjünk 32°-höz.

23° $C = C_1,\ D = D_1$
$l = 2$

24° $C = C \cup C_l$
$l = l_0$ esetén folytassuk 25°-nél, egyébként 26°-nál.

25° $C' = C_l$
Menjünk 28°hoz

26° $D = D \cup D_l$
$l = l + 1$
Térjünk vissza 24°-hez.

27° $C \ = \{r\},\ C' = \{r\},\ D = \varnothing$

28° $D' = \cup \{A_i : i \in C'\} - (D \cup \{\beta_{l_0}\})$
$D' = \varnothing$ esetén folytassuk 31°-nél, egyébként 29°-nél.

29° $D \ = D \cup D'$
$C' = \cup \{B_j : j \in D'\} - C$
$C' = \varnothing$ esetén folytassuk 31°-nél, egyébként 30°-nál.

30° $C = C \cup C'$
Térjünk vissza 28°-hoz.
31° $\delta_{ij} = \delta_{ij} - \delta_{rs}$ $(i \in C,\ j \notin D)$
$\delta_{ij} = \delta_{ij} + \delta_{rs}$ $(i \notin C,\ j \in D)$
32° [*A báziscellák új rendszerének elkészítése.*]
$A_{\alpha l_0} = A_{\alpha l_0} - \{\beta_{l_0}\},\ B_{\beta l_0} = B_{\beta l_0} - \{\alpha_{l_0}\}$
$A_r = A_r \cup \{s\},\ B_s = B_s \cup \{r\}$
33° [*Az x_{ij} változók új értéke.*]
$x_{\alpha_l, \beta_l} = x_{\alpha_l, \beta_l} - x_{\alpha_{l_0}, \beta l_0}$ $(l = 1, 2, \ldots, l_o - 1, l_o + 1, \ldots, l_1)$
$x_{\alpha_{l+1}, \beta_l} = x_{\alpha_{l+1}, \beta_l} + x_{\alpha_{l_0}, \beta_{l_0}}$ $(l = 1, 2, \ldots, l_1 - 1)$
$x_{rs} = x_{\alpha_{l_0} \beta_{l_0}}$
$x_{\alpha_{l_0}, \beta_{l_0}} = 0$
Térjünk vissza 9°-hez
34° Az algoritmus vége.
[x_{ij} $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ az optimális megoldás.]

8. *Példa a módosított stepping-stone algoritmus alkalmazására.* Tekintsük az (1) szállítási problémát a következő adatokkal:

$m = 4$	$b_1 = 14$	$c_{11} = 10$	$c_{21} = 1$	$c_{31} = 4$	$c_{41} = 1$
$n = 7$	$b_2 = 30$	$c_{12} = 15$	$c_{22} = 5$	$c_{32} = 3$	$c_{42} = 2$
	$b_3 = 22$	$c_{13} = 3$	$c_{23} = 21$	$c_{33} = 24$	$c_{43} = 22$
$a_1 = 17$	$b_4 = 27$	$c_{14} = 4$	$c_{24} = 12$	$c_{34} = 7$	$c_{44} = 4$
$a_2 = 29$	$b_5 = 13$	$c_{15} = 0$	$c_{25} = 12$	$c_{35} = 1$	$c_{45} = 11$
$a_3 = 30$	$b_6 = 2$	$c_{16} = 13$	$c_{26} = 14$	$c_{36} = 23$	$c_{46} = 5$
$a_4 = 50$	$b_7 = 18$	$c_{17} = 13$	$c_{27} = 15$	$c_{37} = 1$	$c_{47} = 6$

Az algoritmus 1°—9° részét alkalmazva az

$A_1 = \{3, 5\}$, $A_2 = \{1, 3\}$, $A_3 = \{4, 6, 7\}$, $A_4 = \{1, 2, 4\}$,
$B_1 = \{2, 4\}$, $B_2 = \{4\}$, $B_3 = \{1, 2\}$, $B_4 = \{3, 4\}$, $B_5 = \{1\}$, B- $= \{3\}$,
$B_7 = \{3\}$

halmazokat és induló szállítási táblaként az 1. táblát kapjuk. Itt a bekeretezett számot tartalmazó

1. tábla

—27	—31	4	—18	13	−11	−33	17
12	—3	18	—8	6	6	−17	29
0	2	0	10	20	2	18	30
3	30	—1	17	7	15	−8	50
14	30	22	27	13	2	18	126

cellák alkotják az induló báziscellarendszert, a bázisváltozók értékével a körök belsejében. A cellák jobb felső sarkában a δ_{ij} értékek találhatók. A báziscellákhoz tartozó δ_{ij} számok és a bázison kívüli cellákhoz tartozó x_{ij} számok értéke természetesen 0. A c_{ij} célfüggvényegyütthatókra ezentúl már nem lesz szükségünk.

9°-nél folytatva az algoritmust, max $\delta_{ij} = 20 = \delta_{35}$, tehát $r = 3$, $s = 5$. Most a 10—13° közötti lépéseket elvégezve

$$C_1 = \{3\},\quad D_1 = \{4, 6, 7\},$$

$$C_2 = \{4\},\quad D_2 = \{1, 2\},$$

$$C_3 = \{2\},\quad D_3 = \{3\},$$

$$C_4 = \{1\},\quad D_4 = \{5\}.$$

Láthatjuk, hogy $s \notin D_1, D_2, D_3$, de $s \in D_4$ így $l_1 = 4$, $\beta_4 = 5$, és $\alpha_4 = 1$. A 14—17° lépéseket alkalmazva

$$\beta_3 = 3,\ \alpha_3 = 2,\ x_{23} = 18 > x_{15} = 13 \quad \text{miatt} \quad \text{marad } l_0 = 4,$$

$$\beta_2 = 1,\ \alpha_2 = 4,\ x_{41} = 3 < x_{15} = 13 \quad \text{miatt} \quad l_0 = 2,$$

$$\beta_1 = 4,\ \alpha_1 = 3,\ x_{34} = 10 > x_{41} = 3 \quad \text{miatt} \quad \text{marad } l_0 = 2.$$

Következik a C, D halmazok elkészítése a 17—29° lépések szerint. $l_0 \neq l$ és $l_0 \neq l_1$ miatt 23°-hoz ugorhatunk. A 25°-höz érés pillanatában

$$C = \bigcup_{l=1}^{l=l_0} C_l = C_1 \cup C_2 = \{3, 4\},$$

$$D = \bigcup_{l=1}^{l=l^0-1} D_l = D_1 = \{4, 6, 7\},$$

ezután 25° és 28—29° szerint $C' = \{4\}$, majd $D' = \{2\}$, $D = \{2, 4, 6, 7\}$ és $C' = \varnothing$. Végeredményben $C = \{3, 4\}$ és $D = \{2, 4, 6, 7\}$. Most a 31—33° utasítások formái szerint elkészítjük az

$$A_1 = \{3, 5\},\ A_2 = \{1, 3\},\ A_3 = \{4, 5, 6, 7\},\ A_4 = \{2, 4\},$$

$$B_1 = \{2\},\ B_2 = \{4\},\ B_3 = \{1, 2\},\ B_4 = \{3, 4\},\ B_5 = \{1, 3\},$$

$$B_6 = \{3\},\ B_7 = \{3\}$$

halmazokat és az új szállítási táblát (2. tábla)

2. tábla

−27	−11	7	2	10	9	−13
14	17	15	12	6	26	3
−20	2	−20	7	3	2	18
−20	30	−21	20	−13	15	−8

A következő bázisba bevonandó cella a (2, 6) cella, a hozzá tartozó hurok a

(2, 6), (2, 3), (1, 3), (1, 5), (3, 5), (3, 6)

cellákból áll. A bázisból kilép a (3, 6) cella, amely egyetlen báziscellája a 6-ik oszlopnak. Eszerint $C = \{1, 2, 3, 4\}$, $D = \{1, 2, 3, 4, 5, 7\}$ és következik a 3. szállítási tábla.

3. tábla

−27	−11	9	2	8	−17	−13
14	17	13	12	6	2	3
−20	2	−20	7	5	−26	18
−20	30	−21	20	−13	−11	−8

A bázisba bevonandó (2, 2) cellához tartozó hurok a

(2, 2), (2, 3), (1, 3), (1, 5), (3, 5), (3, 4), (4, 4), (4, 2)

cellákból áll. A bázisból kilép a (3, 4) cella. A (3, 4) cellát nem tartalmazó, de a (3, 5) cellát tartalmazó báziscellákból álló maximális összefüggő H_1 cellagráf a

(3, 5), (3, 7), (1, 5), (1, 3), (2, 3), (2, 1), (2, 6)

cellákból áll, tehát $C = \{1, 2, 3\}$ és $D = \{1, 3, 5, 6, 7\}$.

Az újabb szállítási tábla a 4. tábla.

4. tábla

−27	−28	16	−15	1	−17	−13
14	7	6	−5	6	2	3
−20	−15	−20	−17	12	−26	18
−3	23	−4	27	4	6	9

Most bejön a bázisba a (4, 7) cella az (1, 5) cella helyett. A C és D halmazoknak a jelölési algoritmussal való elkészítéséhez elsőként az 1. sort és a 3. oszlopot kell megjelölnünk. Ezután a 2. sort és az 1., 2. és 6. oszlopot, majd a 4. sort és a 4. oszlopot fogjuk megjelölni. Mivel újabb sort már nem tudunk megjelölni a kritériumunk alapján, végeredményben $C = \{1, 2, 4\}$, $D = \{1, 2, 3, 4, 6\}$. Következik az 5. tábla

5. tábla

−27	−28	17	−15	−9	−17	−22
14	8	5	−5	−3	2	−6
−11	−6	−11	−8	13	−17	17
−3	22	−4	27	−5	6	1

Most a bázisba csak a (4, 6) cella vonható be, a (2, 6) cella helyett. $C = \{1, 2, 3, 4\}$, $D = \{1, 2, 3, 4, 5, 7\}$. A következő 6. tábla már az optimális megoldáshoz tartozó szállítási tábla.

6. tábla

−27	−28	17	−15	−9	−23	−22
14	10	5	−5	−3	−6	−6
−11	−6	−11	−8	13	−23	17
−3	20	−4	27	−5	2	1

(*Beérkezett:* 1970. *III.* 24.)

IRODALOM

[1] HADLEY, G.: Linear Programming. Reading, 1962. Addison-Wesley.

[2] DANTZIG, G. B.: Linear Programming and Extensions. Princeton, 1963. University Press.

[3] KREKÓ, B.: Lineáris programozás. Budapest, 1962. Közgazdasági és Jogi Könyvkiadó.

[4] PRÉKOPA, A.: Lineáris programozás I. Az operációkutatás matematikai módszerei c. tanfolyam jegyzete. Bolyai János Matematikai Társulat, 1968.

MODIFIED STEPPING-STONE ALGORITHM TO SOLVE THE TRANSPORTATION PROBLEM

It is possible to reduce the amount of computations in the stepping stone algorithm (or distribution procedure) which is one of the methods to solve the well known transportation problem, if in the course of the basis changes the transformation of figures δ_{ij} is carried out as follows: Let H be a basis cell system. Let us consider figures δ_{ij} belonging to H and let δ'_{ij} be their new value after drawing cell (i_1, j_1) into and withdrawing cell (i_o, j_o) from the basis. Let us form sets C and D composed of certain row and column indexes in the following algorithmic way: Let us first mark row i_1 and the columns of basis cells in row i_1 which differ from (i_o, i_o) In the course of the general step, let us mark those rows of the basis cells in the already marked columns which are not yet marked, and now in the newly marked rows the basis cell columns which differ from (i_o, j_o) which are not marked.

The procedure comes to an end when it is no longer possible to mark a new row or a new column in this way, and in the final result the indices of the marked rows and columns will form the sets C and D. The transformation of figures δ_{ij} are as follows:

$$\delta'_{ij} = \delta_{ij} \text{ if } i \in C \text{ and } j \in D$$

$$\text{or } i \notin C \text{ and } j \notin D,$$

$$\delta'_{ij} = \delta_{ij} - \delta_{i_1 j_1} \text{ if } i \in C \text{ and } j \notin D,$$

$$\delta'_{ij} = \delta_{ij} + \delta_{i_1 . j_1 .} \text{ if } i \notin C \text{ and } j \in D.$$

The author sets the above-described method of transformation of figures δ_{ij} into a detailed algorithm.

ВИДОИЗМЕНЕННЫЙ „STEPPING – STONE" АЛГОРИТМ ДЛЯ РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ

Возможно сокращение расчётов при использовании stepping stone алгорифма (или по другому: дистрибуционного метода) для решения транспортной задачи, если при замене базисов преобразование чисел δ_{ij} производим следующим образом.

Пусть H система базисных элементов. Рассмотрим числа δ_{ij}, принадлежащие множеству H и пусть δ'_{ij} их новые значения после включения элемента (i_1, j_1) в базис и после вывода из базиса элемента (i_0, j_0). Построим множества C и D, состоящие из каких-то индексов смрочек и столбцов, по следующему алгоритму: Отметим сначала строчку i_1, и в строрке j_1 столбцы базисных элементов, отличных от (i_0, j_0). На общем шаге отметим в только что отмеченных столбцах те базисные элементы, которые ещё не отмечены, далее во вновь отмеченных строчках те столбцы базисных элементов, которые ещё не отмечены.

Метод кончается, если таким способом уже не можем обозначить новую строчку или новый столбец, и в результате индексы отмеченных строк и столбцов дают множества C и D.

Формулы преобразования чисел δ_{ij}:

$$\delta'_{ij} = \delta_{ij} \text{ если } i \in C \text{ и } j \in D$$

$$\text{или } i \notin C \text{ и } j \notin D$$

$$\delta'_{ij} = \delta_{ij} - \delta_{i_1,j_1} \text{ если } i \in C \text{ и } j \in D$$

$$\delta'_{ij} = \delta_{ij} + \delta_{i_1,j_1} \text{ если } i \notin C \text{ и } j \notin D$$

Автор вставляет изложенный способ преобразования чисел в подробно разработанный алгоритм.