Egyetemi doktori értekezés

AZ ÁLTALÁNOS ELOSZTÁSI (SZÁLLITÁSI) FELADAT MEGOLDÁSÁNAK ALGORITMUSA A SOR- ÉS OSZLOP-SAJÁTELEM FOGALOM FELHASZNÁLÁSÁVAL

Készitette:

Gyurkó György
matematika-fizika szakos
középiskolai tanár

Témavezetők:

Dr. Makay Árpád egyetemi docens
Dr. Megyesi László egyetemi docens

József Attila Tudományegyetem Természettudományi Kar
Bolyai Intézet

Szeged, 1982

TARTALOMJEGYZÉK

E L Ő S Z Ó

Az elosztási (szállitási) probléma egy speciális lineáris programozási feladat. Sajátosságai lehetővé teszik, hogy az egyébként megoldáshoz vezető szimplex transzformációkat - amelyeknek tárgya egy $(m+n) \times (m \times n)$ méretü mátrix - olyan gazdaságosabb operációkkal helyettesitsük, melyeket egy jelentősen kisebb - $m \times n$ méretü - u.n. disztribuciós táblán (továbbiakban: DT) folytatunk. Általában olyan müveletekről van szó, amelyeket a DT báziselemeiből - mint csomópontokból - álló gráfok felett definiálunk.

A megoldás elve - a disztribuciós eljárás - régóta egyszerü tananyag ([5] , [7]) , de a gyakorlati realizálását megneheziti, hogy a szóbanforgó gráfok nem közvetlenül adottak, azokat valahogy elő kell állitani. Még kisméretü feladatok "kézi" megoldása esetén e gráfok azonnal szembetünőek, de nagyobb feladatok számitógépi megoldásánál kijelölésük már igen komoly algoritmizálási probléma.

Szerencsére a feladat sajátosságaiból az is adódik, hogy a DT "útkereszteződéseiben" alkalmas "útjelző táblák" helyezhetők el. Az útjelzők szerepét Glover módszerénél (csak a klasszikus feladatra) a hármas-indexelés (pedecessor index method, triple-label method [3], [4]), Pogány Zsuzsánál (az általános feladatra) a négyes-indexelés ([6]) tölti be; ez azt jelenti, hogy a kérdéses gráfokat a DT soraihoz és oszlopaihoz rendelt index-hármasokkal, illetve index-négyesekkel irják le.

Az itt tárgyalt algoritmusban az útjelző funkciót a sor- és oszlop-sajátelem fogalom látja el. Eredeti megfogalmazásában (amely egy gyakorlati probléma megoldása kapcsán született) még nem esett szó sor- és oszlop-indexelésről, de a korábbi megoldásokkal való összevetés céljából itt meg fogjuk mutatni: a sajátelem fogalommal végzett müveletek az bizonyitják, mindig létezik a DT sorainak és oszlopainak olyan indexelése, hogy

- egy-egy indexet rendelünk minden sorhoz és oszlophoz,
- a disztribációs eljárás megfogalmazható ezen egyes-indexelés felett definiált könnyen programozható eljárások sorozataként.

Értelmezésbeli bizonytalanságok elkerülése céljából megjegyezzük: A disztribuciós eljárás maga is szimplex algoritmus, ha az utóbbin azt értjük, hogy egy - az n-dimenziós euklidészi téren értelmezett - lineáris függvény valamely poliédrikus halmazra vonatkozó feltételes szélsőértékhelyét (ha létezik) ilymódon keressük meg: a poliédrikus halmaz szomszédos csúcsain úgy haladunk végig, hogy közben a függvénynek a szomszédos csúcsokon felvett értékei monoton közelitenek a szélsőértékhez. A dolgozat viszont konkrétan az SPX1-SPX5 eljárást (lásd az 1.1. fejezetben), illetve annak kétfázisú változatát (lásd az 1.8. fejezetben) fogja szimplex algoritmusnak tekinteni, és ennek megfelelően állitja szembe a disztribuciós eljárással (lásd az 1.2. fejezetben).

1. RÉSZ

AZ ALGORITMUS LEIRÁSA

1.1. A LINEÁRIS PROGRAMOZÁSI FELADAT ÉS A SZIMPLEX ALGORITMUS

A sajátelemeken alapuló algoritmus egyes lépéseinek igazolása céljából gyakran fogunk hivatkozni a szimplex algoritmus analóg lépéseire, ezért röviden az utóbbi algoritmust is ismertetjük.

A fejezetben néhány fogalom definició nélkül és valamennyi állitás bizonyitás nélkül szerepel. A hiányzó értelmezések illetve bizonyitások egyszerüen nem képezik a mondanivaló lényegét, ugyanakkor a legtöbb lineáris algebrával vagy a lineáris programozás alapjaival foglalkozó szakkönyvben megtalálhatók.

1.1.1. Definició: Lineáris programozási feladaton (a továbbiakban: LP-feladat) a

$$(1.1.1)\quad \max \left\{ \underline{c}^{*}\underline{x} : \underline{x} \geqq \underline{0},\ \underline{A}\cdot\underline{x} \left[\leqq, =, \geqq\right] \underline{b} \right\}$$

szélsőértékproblémát értjük; ebben a $\underline{c}^{*}\underline{x}$ függvényt célfüggvénynek nevezzük. . (Itt és általában a továbbiakban $\underline{c}^{*}$, $\underline{x}$, $\underline{b}$ valós komponensü vektorok, $\underline{A}$ valós komponensü mátrix; $\underline{c}^{*}$, $\underline{A}$, $\underline{b}$ adottak, $\underline{x}$ ismeretlen. A $\left[\leqq, =, \geqq\right]$ szimbólum azt jelenti, hogy a $\underline{b}$ különböző komponenseinél különböző relációk teljesülését irhatjuk elő, de mindig a $\leqq$, $=$, $\geqq$ relációk valamelyikét.)

1.1.2. Definició A

$$(1.1.2.)\quad \max \left\{ \underline{c}^{*}\underline{x} : \underline{x} \geqq \underline{0},\ \underline{A}\,\underline{x} \leqq \underline{b}, \underline{b} \geqq \underline{0} \right\}$$

szélsőértékfeladatot normál lineáris programozási feladatnak (továbbiakban: NLP-feladat) nevezzük.

1.1.3. Definició: Az LP-feladat esetében teljes megoldásnak nevezünk minden olyan $y=\begin{Bmatrix} x \\ u \end{Bmatrix}$ vektort, amellyel: $[A\ E]\cdot y = b$, ugyanakkor x-et megoldásnak (vagy a teljes megoldás primális részének) mondjuk. Ha ilyenkor még

$$x \geqq 0, \quad A\,x\left[\leqq, =, \geqq\right] b \tag{1.1.3}$$

is teljesül, akkor az y vektort lehetséges teljes megoldásnak, az x vektort lehetséges megoldásnak nevezzük.

1.1.4. Definició: Az LP-feladat egy y teljes megoldását vagy annak primális részét bázismegoldásnak nevezzük, ha az $[A\ E]$ mátrix azon oszlopvektorainak rendszere, amelyekhez nem-zéró y -komponens tartozik lineárisan független.

1.1.1. Állitás: Az NLP-feladatnak mindig van lehetséges bázismegoldása, egy ilyen bázismegoldás éppen az $x = 0$.

Mivel a lehetséges megoldások halmaza konvex és a $c^{*}\cdot x$ függvény lineáris, ezért fennáll:

1.1.2. Állitás: Az (1.1.1) feladat esetén valamely lokális szélsőérték(hely) egyben totális szélsőérték(hely) is.

1.1.5. Definició: Az (1.1.1) feladat feltételes maximumhelye(i)t másképpen optimális megoldás(ok)nak nevezzük.

1.1.3. Állitás: Ha az (1.1.1) feladatnak van optimális megoldása, akkor van optimális bázismegoldása is.

1.1.6. Definició: Ha valamely bázisról úgy térünk át egy újabb bázisra, hogy az utóbbi csak egyetlen vektorban különbözik az előbbitől, akkor elemi bázistranszformációról (báziscseréről) beszélünk.

1.17. Definició: Legyen $b_1, b_2, \ldots, b_i, \ldots, b_n$ bázis az E_n euklidészi térben, továbbá

$\underline{c} \in E_n$;

$\underline{c} \neq \underline{b}_i \qquad i = 1,2,\dots,n$;

$\underline{c} = [c_1, c_2, \dots, c_i, \dots, c_n]$.

Ha egy bázistranszformáció során a $\underline{c}$ vektort az előző bázis $\underline{b}_i$ vektora helyébe cseréljük, akkor a $\underline{c}$ vektor c_i komponensét az <u>elemi bázistranszformáció generáló elemének</u> nevezzük.

1.1.4. <u>Állitás:</u> Legyen $\underline{b}_1, \underline{b}_2, \dots, \underline{b}_i, \dots, \underline{b}_n$ bázis az E_n euklidészi térben, valamint

$\underline{a}, \underline{c} \in E_n$;

$\underline{c} \neq \underline{b}_i \qquad i = 1,2,\dots,n$;

$\underline{a} = a_1\underline{b}_1 + a_2\underline{b}_2 + \dots + a_n\underline{b}_n$;

$\underline{c} = c_1\underline{b}_1 + c_2\underline{b}_2 + \dots + c_n\underline{b}_n$.

Ha most $c_i \neq 0$ és $\underline{c} \rightarrow \underline{b}_i$ cserével áttérünk a

$\underline{b}_1, \underline{b}_2, \dots, \underline{c}, \dots, \underline{b}_n$

bázisra, akkor erre vonatkozóan az $\underline{a}$ vektor igy irható fel:

$$(1.1.4) \quad \underline{a} = (a_1 - \delta c_1)\,\underline{b}_1 + (a_2 - \delta c_2)\,\underline{b}_2 + \dots$$
$$\dots + \underline{b} + \dots + (a_n - \delta c_n)\,\underline{b}_n \; ,$$

ahol $\delta = \dfrac{a_i}{c_i}$.

Most tekintsük a feladat együtthatóinak mátrixát (az egységmátrixszal kiegészitve):

$$(1.1.5) \qquad \underline{L}_0 = \begin{bmatrix} \underline{A} & E & \underline{b} \\ \underline{c}^* & \underline{0}^* & 0 \end{bmatrix}$$

A mátrix oszlopvektorai felfoghatók, mint a

$$\underline{B}_0 = \begin{bmatrix} \underline{E} & \underline{0} \\ \underline{0}^* & 1 \end{bmatrix}$$

mátrix meghatározta bázisban megadott vektorok.

1.1.5 Állitás: Legyen $\underline{B}$ olyan nem szinguláris mátrix, amelynek rangja egyenlő az $\underline{L}_0$ mátrix sorai számával. Ekkor az $\underline{L}_0$ mátrixnak a $\underline{B}$ mátrix meghatározta bázisra vonatkozó komponenseit az

$$(1.1.6) \qquad \underline{L} = \underline{B}^{-1}\underline{L}_0$$

formula adja.

A továbbiakban a $\begin{bmatrix} \underline{D} \\ \underline{d}^* \end{bmatrix}$ olyan mátrixot fog jelölni, amely oszlopvektorait az $\underline{L}_0$ mátrix

$$\begin{bmatrix} \underline{A} & \underline{E} \\ \underline{c}^* & \underline{0}^* \end{bmatrix}$$

blokkjából veszi, és vele a

$$(1.1.7) \qquad \underline{B} = \begin{bmatrix} \underline{D} & \underline{0} \\ \underline{d}^* & 1 \end{bmatrix}$$

mátrix nem szinguláris.

1.1.6. Állitás: Ha a $\underline{B} = \begin{bmatrix} \underline{D} & \underline{0} \\ \underline{d}^* & 1 \end{bmatrix}$ mátrix nem szinguláris, akkor fennáll:

$$(1.1.8) \qquad \underline{B}^{-1} = \begin{bmatrix} \underline{D}^{-1} & 0 \\ -\underline{d}^*\underline{D}^{-1} & 1 \end{bmatrix}$$

1.1.7. Állitás: Az (1.1.7) szerinti $\underline{B}$ mátrix által meghatározott bázisra vonatkozóan az $\underline{L}_0$ komponenseit a következő $\underline{L}$ mátrix szolgáltatja:

$$(1.1.9) \qquad \underline{L} = \begin{bmatrix} \underline{D}^{-1}\underline{A} & \underline{D}^{-1} & \underline{D}^{-1}\underline{b} \\ \underline{c}^* - \mathrm{d}^*\underline{D}^{-1}\underline{A} & -\underline{d}^*\underline{D}^{-1} & -\mathrm{d}^* \mathrm{D}^{-1}\underline{b} \end{bmatrix}$$

1.1.8. Állitás: Ha egy NLP-feladat esetén az (1.1.7) szerinti $\underline{B}$ mátrix által meghatározott bázisra vonatkozóan (1.1.9) szerint felirt $\underline{L}$ mátrixban

(1.1.10) $$\underline{D}^{-1}\underline{b} \geqq \underline{0},$$

akkor az $\underline{L}$ -ből kiolvasható a feladat egy lehetséges bázismegoldása: Mégpedig, ha az $\underline{A}$ blokk egy $\underline{a}_i$ oszlopvektora egyben a $\underline{D}$-nek a j. oszlopvektora, akkor $x_i = b_j$, ahol b_j a $\underline{D}^{-1}\cdot\underline{b}$ vektor j. komponense; különben $x_i = 0$. Ha ezeken felül még

(1.1.11) $$\left[\underline{c}^* - \underline{d}^* \underline{D}^{-1}\underline{A}\,, \quad -\underline{d}^* \underline{D}^{-1}\right] \leqq \underline{0}^*,$$

akkor a megoldás optimális.

Sokszor kényelmesebb lesz az NLP-feladat következő az (1.1.2)-vel egyenértékü megfogalmazására hivatkozni.

(1.1.12) $$\max\left\{\underline{c}^*\underline{x} : \underline{x} \geqq \underline{0}, \underline{u} \geqq 0, \left[\underline{A} \;\; \underline{E}\right] \cdot \begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix} = \underline{b}, (\underline{b} \geqq \underline{0})\right\}$$

1.1.8. Definició: Az (1.1.12) feladat esetében az

(1.1.13) $$L_t = \left\{\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix} : \begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix} \geqq \underline{0}\,, \left[\underline{A}\,, \underline{E}\right] \cdot \begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix} = \underline{b}\,, (\underline{b} \geqq \underline{0})\right\}$$

halmazt a feladat lehetséges teljes megoldásai halmazának nevezzük. Az $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ teljes megoldásban $\underline{x}$ a primális rész, $\underline{u}$ a kiegészitő rész.

Nyilvánvaló, hogy az L_t halmaz elemei primális részének halmaza azonos az

$$L = \left\{\underline{x}(\geqq \underline{0}) : \underline{A}\,\underline{x} \leqq \underline{b}, (\underline{b} \geqq \underline{0})\right\}$$

halmazzal.

Itt célszerü megfogalmazni az 1.1.8. állitásnak a teljes megoldásokra vonatkozó megfelelőjét is:

1.1.9. Állitás: Ha egy NLP-feladat esetén az (1.1.7) szerinti $\underline{B}$ mátrix által meghatározott bázisra vonatkozóan (1.1.9) szerint felirt $\underline{L}$ mátrixban (1.1.10) teljesül, akkor $\underline{L}$-ből kiolvasható a feladat egy $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ lehetséges teljes bázismegoldása: Mégpedig, ha az $\begin{bmatrix} \underline{A} & \underline{E} \end{bmatrix}$ mátrix i. oszlopvektora egyben a $\underline{D}$-nek a j. oszlopvektora, akkor az $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ i-edik komponense egyenlő a $\underline{D}^{-1} \cdot \underline{b}$ vektor j. komponensével, különben pedig az $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ i. komponense: 0. Ha ezeken felül még (1.1.11) is teljesül, akkor a megoldás optimális.

1.1.9. Definició: Az LP-feladat egy $\underline{y}$ teljes bázismegoldása esetén legyen Y az $\begin{bmatrix} \underline{A} & \underline{E} \end{bmatrix}$ mátrix azon oszlopvektorainak rendszere, amelyekhez nem-zéró $\underline{y}$-komponens tartozik; ha B az $\begin{bmatrix} \underline{A} & \underline{E} \end{bmatrix}$ oszlopvektorainak olyan maximális lineárisan független részrendszere, hogy $Y \subset B$, akkor B-t az $\underline{y}$-hoz tartozó (aktuális) bázisnak mondjuk.

1.1.10. Definició: Az LP-feladat egy $\underline{y}$ teljes bázismegoldását degeneráltnak mondjuk, ha benne a nem-zéró komponensek száma kevesebb az $\underline{A}$ mátrix sorainak számánál.

1.1.10. Állitás: Ha adott az (1.1.12) feladat egy nem degenerált $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ lehetséges teljes bázismegoldása, és ha a

$$(1.1.14) \qquad \begin{bmatrix} \underline{D} \\ \underline{d}^{*} \end{bmatrix}$$

mátrix az

$$\begin{bmatrix} \underline{A} & \underline{E} \\ \underline{c}^{*} & \underline{0} \end{bmatrix}$$

mátrixnak pontosan azokból az oszlopvektoraiból áll, amelyekre vonatkozóan az $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ komponensei pozitivek, akkor az (1.1.14) felhasználásával számitott (1.1.9) alakú $\underline{L}$ mátrixra teljesül (1.1.10), és $\underline{L}$-ből az 1.1.9. állitásnál leirt módon éppen az

$\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix}$ teljes bázismegoldás olvasható ki. Ha ez a megoldás optimális, pontosan akkor teljesül még (1.1.11) is.

Most pedig nézzük sorra a szimplex algoritmus lépéseit, egyelőre NLP-feladatra, degenerációmentes esetben:

SPX1. Vesszük a feladat egy (lehetséges) bázismegoldását, és felirjuk a hozzátartozó (1.1.9) alaku $\underline{L}$ mátrixot. (Kezdetben vehető az $\begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{b} \end{bmatrix}$ megoldás és az $\underline{L}_0$ mátrix. A továbbiakban pedig $\underline{L}$-ben nem kell a $\left[\underline{D}^{-1} \cdot \underline{A}\right]$ blokkot felirni, hanem ha abból valamely oszlopvektorra szükség lesz az SPX3 pontban, úgy azt — és csak azt — ott előállitjuk.)

SPX2. Ha az (1.1.11) feltétel teljesül, akkor az $\underline{L}$-ből egy optimális megoldás az 1.1.9. állitás szerint kiolvasható. — Ezzel az algoritmus befejeződik. (Az alternativ megoldások nem érdekelnek.) Ha (1.1.11) nem teljesül, akkor áttérünk az SPX3 pontra. (Az (1.1.10) feltétel teljesülését az SPX1 és SPX3 pontok automatikusan biztositják.)

SPX3. Kiszemeljük a

$$\left[\underline{c}^{*} - \underline{d}^{*}\,\underline{D}^{-1} \cdot \underline{A} \qquad -\underline{d}^{*}\,\underline{D}^{-1}\right]$$

vektor egy pozitiv komponensét. Ha ez $\hat{c}_{j_0}$, akkor vesszük a

$$\left[\underline{D}^{-1} \cdot \underline{A} \qquad \underline{D}^{-1}\right]$$

blokk j_0. oszlopvektorát (jelölje ezt: $\hat{\underline{a}}_{j_0}$):

$$\hat{\underline{a}}_{j_0} = \left[\hat{a}_{1,j_0}, \hat{a}_{2,j_0}, \ldots, \hat{a}_{m,j_0}\right]^{*}$$

Ha $\hat{a}_{i,j_0} \leq 0$, $i = 1,2,\ldots, m$, akkor a célfüggvény a lehetséges megoldások halmazán nem korlátos, — az algoritmus befejeződik. Különben kiválasztjuk az $\hat{a}_{i_0,j_0}$ generáló elemet az

(1.1.15)

$$\frac{\hat{b}_{i_0}}{\hat{a}_{i_0,j_0}} = \min\left\{\frac{\hat{b}_i}{\hat{a}_{i,j_0}} : \hat{a}_{i,j_0} = \underline{e}_i^* \cdot \hat{\underline{a}}_{j_0} ,\right.$$

$$\left. \hat{a}_{i,j_0} > 0,\ \hat{b}_i = \underline{e}_i^* \cdot \underline{D}^{-1} \cdot \underline{b} ,\ i = 1,2,\ldots, m\right\}$$

feltétel szerint. (Degenerációmentes esetben a választás egyértelmü.)

SPX4. A kiválasztott $\begin{bmatrix} \hat{\underline{a}}_{j_0} \\ \hat{c}_{j_0} \end{bmatrix}$ vektort becseréljük a

$\underline{B} = \begin{bmatrix} \underline{D} & \underline{0} \\ \underline{d}^* & 1 \end{bmatrix}$ bázisba az i_0-adik helyre, és az 1.1.4. állitás szerint kiszámitjuk az elemi bázistranszformáció utáni új bázisnak megfelelő

$$\begin{bmatrix} \underline{D}^{-1} \\ -\underline{d}^* \underline{D}^{-1} \end{bmatrix}$$

blokkot. (Az $\underline{L}$ mátrixból elegendő csak ezt a blokkot számitani, mert ennek és $\underline{L}_0$-nak a segitségével az $\underline{L}$ bármely további blokkja - amelyre az SPX2, SPX3 pontokban szükség van - már előállitható.)

SPX5. Visszatérünk az SPX2 pontra.

1.1.11. Állitás: Ha az NLP-feladatnak van optimális megoldása, akkor azt az SPX1-SPX5 algoritmus véges sok lépés után megadja (SPX2 pont), optimális megoldás hiányában az algoritmus - ugyancsak véges sok lépés után - az SPX3 pontban ér véget. (Degeneráció kizárva.)

A későbbiekben ki fogunk térni a nem - NLP - feladat és a degeráció esetére is. (Lásd az 1.7. és 1.8. fejezeteket.)

1.2. AZ ÁLTALÁNOS ELOSZTÁSI (SZÁLLITÁSI) FELADAT ÉS A DISZTRIBUCIÓS ELJÁRÁS

1.2.1. Definició: Az (1.1.1) feladatot általános elosztási (szállitási) feladatnak (a továbbiakban: AEF) nevezzük, ha az $\underline{A}$ mátrix felirható

(1.2.1) $$\underline{A} = \left[\underline{a}_{1,1}, \underline{a}_{1,2}, \dots, \underline{a}_{1,n}, \underline{a}_{2,1}, \dots, \underline{a}_{2,n}, \dots, \underline{a}_{m,1}, \dots, \underline{a}_{m,n}\right]$$

alakban, ahol az $\underline{a}_{i,j}$ oszlopvektorokra teljesül:

(1.2.2) $$\underline{a}_{i,j} = f_{i,j} \cdot \underline{e}_i + \underline{e}_{m+j}\,; \qquad f_{i,j} > 0\,;$$

$$i = 1,2,\dots,m\,; \qquad j = 1,2,\dots,n\,.$$

1.2.2. Definició: Valamely általános elosztási feladatot klasszikus elosztási (szállitási) feladatnak nevezzük, ha

(1.2.3) $$f_{i,j} = 1\,; \quad i = 1,2,\dots,m\,; \quad j = 1,2,\dots,n\,;$$

és a lehetséges megoldások halmaza speciálisan

(1.2.4) $$L = \left\{\underline{x} : \underline{x} \geqq \underline{0}\,, \quad \underline{A} \cdot \underline{x} = \underline{b}\,, (\underline{b} \geqq \underline{0})\right\}.$$

Jelölések: Az (1.2.1) mintájára célszerü a $\underline{c}^*$ komponenseit is igy indexelni.

$$\underline{c}^* = \left[c_{1,1}, c_{1,2}, \dots, c_{1,n}, c_{2,1}, \dots, c_{2,n}, \dots, c_{m,1}, \dots, c_{m,n}\right].$$

A müveletek könnyebben formalizálhatók, ha bevezetjük még a további jelöléseket:

$$(1.2.5) \qquad \underline{E} = \left[\underline{e}_1, \underline{e}_2, \dots, \underline{e}_{m+n}\right] =$$

$$= \left[\underline{a}_{1,n+1}, \dots, \underline{a}_{m,n+1}, \underline{a}_{m+1,1}, \dots, \underline{a}_{m+1,n}\right].$$

$$(1.2.6) \qquad \begin{array}{ll} c_{i,n+1} = 0 & i = 1,2,\dots,m; \\ c_{m+1,j} = 0 & j = 1,2,\dots,n; \end{array}$$

$$(1.2.7) \qquad f_{i,n+1} = 1 \qquad i = 1,2,\dots,m.$$

(Nem értelmezünk $f_{m+1,j}$ mennyiségeket.)

A (1.2.5) és (1.2.7) alapján a következő nyilvánvaló és az (1.2.2)-vel némileg analóg összefüggést kapjuk:

$$\underline{a}_{i,n+1} = f_{i,n+1} \cdot \underline{e}_i \qquad i = 1,2,\dots,m.$$

Egyelőre, csak olyan AEF-ek esetét tárgyaljuk, amelyekre a lehetséges megoldások halmaza:

$$L = \left\{ \underline{x} : \underline{x} \geqq \underline{0},\ \underline{A}\,\underline{x} \leqq \underline{b},\ (\underline{b} = \underline{0}) \right\},$$

vagyis az AEF egyben NLP-feladatot is jelent.

Az ilyen AEF-ekhez az NLP-feladat (1.1.12) átfogalmazása szerint tartozó $\underline{u}$ kiegészitő változót az (1.2.5) analógiájára komponenseivel igy célszerü jelölni:

$$(1.2.8) \qquad \underline{u} = \left[x_{1,n+1}, \dots, x_{m,n+1}, x_{m+1,j}, \dots, x_{m+1,n}\right]^{*}.$$

Vezessük még be a következő jelölést:

$$\underline{b} = \left[t_1, t_2, \ldots, t_m, r_1, r_2, \ldots, r_n \right]^{*} \tag{1.2.9}$$

1.2.3. <u>Definició:</u> <u>Disztribuciós táblán</u> (DT) az AEF $f_{i,j}$, $c_{i,j}$, t_i, r_j adatainak az (1.2.10) táblázatát értjük.

(1.2.10)

$f_{1,1}$ $c_{1,1}$	...	$f_{1,j}$ $c_{1,j}$	...	$f_{1,n}$ $c_{1,n}$	$f_{1,n+1}$ $c_{1,n+1}$	t_1
...	...	...	...	...	...	...
$f_{i,1}$ $c_{i,1}$		$f_{i,j}$ $c_{i,j}$		$f_{i,n}$ $c_{i,n}$	$f_{i,n+1}$ $c_{i,n+1}$	t_i
...	...	...	...	...	...	...
$f_{m,1}$ $c_{m,1}$	...	$f_{m,j}$ $c_{m,j}$	...	$f_{m,n}$ $c_{m,n}$	$f_{m,n+1}$ $c_{m,n+1}$	t_m
$c_{m+1,1}$	...	$c_{m+1,j}$	...	$c_{m+1,n}$		
r_1	...	r_j	...	r_n		

1.2.4. <u>Definició:</u> A DT m+1. sorát <u>fiktiv sornak</u> (FS); n+1. oszlopát <u>fiktiv oszlopnak</u> (FO) nevezzük; a DT megfelelő bejegyzéseit pedig <u>FS</u>-, illetve <u>FO-elemeknek</u> mondjuk.

Az (1.2.1) és (1.2.5) alapján kölcsönösen egyértelmü hozzárendelés áll fenn az AEF $\left[\underline{A}, \underline{E}\right]$ mátrixának oszlopvektorai és a DT (i,j) elemei (rovatai, bejegyzései) között. Ennek megfelelően - ha az nem okoz félreértést - nem is mindig fogunk különbséget tenni a DT (i,j) eleme és a neki megfelelő $\underline{a}_{i,j}$ vektor között. Ilyen alapon beszélni fogunk a DT vektorrendszeréről, - ezen az $\left[\underline{A}\ \ \underline{E}\right]$ oszlopvektorait kell érteni. Hasonlóan szó lehet a DT bázisáról, rangjáról, fiktiv vektorokról, stb. Az (1.2.1) definició és az itt tett megállapodások alapján máris mondható: a DT rangja m+n.

Az AEF megoldására szolgáló disztribúciós eljárás (továbbiakban: DE) itt leirt módozata - egyelőre NLP-feladatot képező AEF-et és degenerációmentes esetet feltételezve - a következő pontokból áll:

DE1. Felirjuk az AEF-hez tartozó DT-t. Véve a feladat egy lehetséges bázismegoldását, kijelöljük a DT-ben a báziselemeket. Felirjuk a báziselemekhez a bázismegoldás szerint tartozó $x_{i,j}$ együtthatókat (programváltozó értékeket).

DE2. Kijelöljük a sajátelemeket.

DE3. Meghatározzuk a potenciálokat.

DE4. Meghatározzuk a célfüggvényegyütthatók transzformált értékeit. Ha az (1.1.11) feltétel teljesül, akkor a DT-ből kiolvasható egy optimális megoldás,- ezzel az algoritmus befejeződik; különben áttérünk a DE5 pontra. (Az (1.1.10) feltételt a DE1 és a DE5 pontok automatikusan teljesitik.)

DE5. A célfüggvényegyütthatók transzformált értékei alapján kiválasztjuk a bázisba bevonandó vektort, kiszámitjuk annak az aktuális bázisra vonatkozó komponenseit, és kiválasztjuk a generáló elemet (szük keresztmetszet). Ha pozitiv generáló elem nem választható, úgy a célfüggvény a lehetséges megoldások halmazán nem korlátos, - az algoritmus befejeződik; különben a DE6 pontra térünk át.

DE6. A transzformációt végrehajtjuk.

DE7. Visszatérünk a DE2. pontra.

A DE1-DE7 pontok és az SPX1-SPX5 pontok között a következő összefüggés áll fenn. Az SPX1-nek megfelel a DE1; az SPX2-nek megfelelnek a DE2-DE4 pontok; továbbá az SPX3, SPX4, SPX5 pontoknak sorrendben megfelelnek a DE5, DE6, DE7 pontok.

A DE2-DE7 pontokat a későbbiekben részletesen tárgyaljuk; a DE1 pontra vonatkozóan pedig megjegyezzük, hogy az induló bázismegoldás kijelölésének egyik módja az 1.1.1. állitást használja ki. Esetünkre alkalmazva ez azt jelenti, hogy az induló bázisvektorrendszer éppen a fiktiv vektorok (FS,FO) rendszere lehet, eszerint a bázisvektorokhoz tartozó $x_{i,j}$ együtthatók igy alakulnak

$$\left[x_{1,n+1}, x_{2,n+1}, \dots, x_{m,n+1}, x_{m+1,1}, \dots, x_{m+1,n}\right]^{*} =$$
$$= \left[t_1, t_2, \dots, t_m, r_1, r_2, \dots, r_n\right]^{*}$$

(Az $\underline{x}$ megoldásvektor további komponensei nyilván zérusok.)

1.3. A SAJÁTELEMEK KIJELÖLÉSE

A korábbi megoldásokhoz képest itt a sajátelem fogalom jelenti a leglényegesebb különbséget; ráépitve, a disztribuciós eljárás szinte minden mozzanata egyszerüsödik. A sajátelem fogalom matematikailag lényegében egy a DT báziselemein értelmezett függvényt jelent, és mint ilyent, éppen a hozzárendelési algoritmussal fogjuk definiálni.

Magát a hozzárendelést kétféle felfogásban fogjuk értelmezni. Az egyik felfogás szerint egy

$$(1.3.1) \qquad f: B \longrightarrow \{S, O\}$$

hozzárendelésről beszélünk, ahol B a DT báziselemeinek halmaza, $\{S, O\}$ pedig a sor-sajátelem, illetve az oszlop-sajátelem tulajdonságok kételemü halmaza. Más tekintetben viszont célszerübb az

$$(1.3.2) \qquad g: B \longrightarrow \{1, 2, \ldots, m, m+1, \ldots, m+n\}$$

hozzárendelésről beszélni; ahol az i $i \leq m$ esetén a DT i. sorát, $i > m$ esetén a DT $(i-m)$. oszlopát szimbolizálja. A további tárgyalás során a két felfogás közül célszerüen hol egyik, hol másik kerül majd előtérbe.

A dolgozat e részében a szigorú matematikai formalizálás még nem segitené elő a téma egyszerübb kifejtését, ezért a sajátelemeket kijelölő algoritmusnak itt csak egy verbális leirását adjuk.

Vegyük a DT egy bázisát (ez mindig m+n elemből áll). A báziselemeket sor-sajátelem (SS), illetve oszlop-sajátelem (OS) tulajdonságokkal cimkézzük fel a következő algoritmus szerint:

S1. A DT FO-beli báziselemeit jelöljük ki SS-elemnek.

S2. A DT FS-beli báziselemeit jelöljük ki OS-elemnek.

S3. Minden nem-fiktiv sorban, amelyben már van SS-elem, az összes még jelöletlen báziselemet jelöljük ki OS-elemnek.

S4. Minden nem-fiktiv oszlopban, amelyben már van OS-elem, az összes még jelöletlen báziselemet jelöljük SS-elemnek. Ezután, ha van még jelöletlen báziselem valamely SS-elem sorában, akkor menjünk a S3 pontra.

S5. Minden nem-fiktiv sorban, amelyben még nincs SS-elem, és csak egy kijelöletlen báziselemet tartalmaz, ezt a jelöletlen báziselemet vegyük SS-elemnek.

S6. Minden nem fiktiv oszlopban, amelyben még nincs OS-elem, és csak egy jelöletlen báziselemet tartalmaz, ezt a jelöletlen báziselemet vegyük OS-elemnek. Végül, ha van még olyan nem-fiktiv sor, amelyben nincs SS-elem, és pontosan egy jelöletlen báziselemet tartalmaz, akkor menjünk az S5. pontra.

S7. Ha az eddigiekben nem jelöltünk ki minden báziselemet, akkor a jelöletlenek közül egyet vegyünk SS-elemnek, és menjünk az S3. pontra; egyébként az algoritmus befejeződik. (Itt egy önkényes választásra kaptunk lehetőséget, ezért az S1-S7 algoritmus nem egyértelmü. Ez azonban nem lehet hiba forrása, mert bevizonyitható, hogy a DE további mozzanatai közömbösek aziránt, hogy itt milyen választással éltünk.)

Megjegyzés: Ha az Si pont másképpen nem rendelkezik, akkor azt mindig az S(i+1) pont követi.

A S1-S7 pontok végrehajtására igen könnyen igen hatékony számitógépes program irható.

Látjuk, hogy a kijelölést (felcimkézést) nem egyetlen szabállyal irtuk le, valamint az egyes szabályok nem csak azt határozzák, hogy a bemenetükből milyen kimenet származik, hanem (az S1-S7-et követő megjegyzést is figyelembe véve) azt is, hogy utánuk melyik más szabály következik. Ezzel összhangban az S1-S7 algoritmust a 2. részben egy véges automata speciális müködéseként fogjuk újradefiniálni. (Lásd a 2.2.1-2.2.4. definiciókat!)

A 2. részben bebizonyitjuk, hogy a báziselemek nem helyezkedhetnek el akárhogyan a DT-ben, de mindig úgy helyezkednek el, hogy az S1-S7 algoritmus elvégezhető (2.2.1. állitás) és mindig teljesülnek a következők:

a/ minden báziselem felcimkéződik (2.2.1. állitás);

b/ egy báziselem vagy csak SS-elem vagy csak OS-elem lehet (2.2.2. állitás);

c/ a DT minden nem-fiktiv sorában pontosan egy SS-elem van (2.2.3. lemma);

d/ a DT minden nem-fiktiv oszlopában pontosan egy OS-elem van (2.2.3. lemma).

Most már az (1.3.1) szerinti f függvényt minden $(i,j) \in B$, báziselem esetén igy definiálhatjuk:

$$f(i,j) = \begin{cases} S, \text{ ha } (i,j) \text{ SS-elem;} \\ 0, \text{ ha } (i,j) \text{ OS-elem.} \end{cases}$$

Az (1.3.2) szerinti g függvény pedig:

$$g(i,j) = \begin{cases} i, \text{ ha } (i,j) \text{ SS-elem;} \\ m+j, \text{ ha } (i,j) \text{ OS-elem} \end{cases}$$

Az S1-S7 algoritmus c/, d/ tulajdonságai szerint a g függvény invertálható. Erre támaszkodva definiálunk egy w függvényt:

$$(1.3.B) \quad w(i) = \begin{cases} j, \text{ ha } i = m \text{ és } g(i,j) = i; \\ k, \text{ ha } i \;\; m \text{ és } g(k, i-m) = i. \end{cases}$$

$$i = 1, 2, \ldots, m+n.$$

A w függvény éppen az előszóban emlitett sor- és oszlop-indexelésnek felel meg; azt jelenti, hogy

- minden nem-fiktiv sorhoz a sorbeli SS-elem oszlopindexét,
- minden nem-fiktiv oszlophoz az oszlopbeli OS-elem sorindexét rendeljük.

Az 1.6. fejezetben látni fogjuk; ha valamely iterációs lépésben már adottak a sajátelemek, akkor a következő iterációs lépéshez tartozó sajátelemek az előbbiekből az S1-S7-nél egyszerübb módon származlathatók. Ugyanakkor az 1.2. fejezetben javasolt induló bázismegoldás esetén már az S1, S2 pontok kijelölik az összes sajátelemet. Az S1- S7 teljes terjedelmében tehát csak akkor válhat szükségessé, ha valamely más eljárással (pl. Vogel-Korda algoritmus) kapott lehetséges bázismegoldást az itt leirtak szerint kivánunk tovább javitani, - akkor is csak az első iterációs lépésben.

1.4. A POTENCIÁLOK MEGHATÁROZÁSA

A DE során is alkalmazzuk az (1.1.11) feltételt, ám a $\underline{D}^{-1}$ mátrixot nem ismerjük; az (1.1.9) szerinti $\underline{L}$ mátrix $\left[-\underline{d}^{*}\underline{D}^{-1}\right]$ blokkját azonban enélkül is kiszámithatjuk.

1.4.1. Definició: Egy adott bázismegoldás esetén a

$$\underline{d}^{*}\underline{D}^{-1} = \left[u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n\right] \tag{1.4.1}$$

vektor u_i, illetve v_j komponensét a DT i. sorához, illetve j. oszlopához tartozó potenciálnak nevezzük.

A továbbiakban az AEF $c_{i,j}$ együtthatóinak egy adott bázismegoldáshoz tartozó transzformált értékeit jelöljük $\hat{c}_{i,j}$-vel. Az (1.2.1), (1.2.2) és (1.4.1), valamint az 1.1.7. állitás alapján azonnal adódik:

1.4.1. Állitás: Minden nem-fiktiv (i,j) esetén

$$\hat{c}_{i,j} = c_{i,j} - (f_{i,j} u_i + v_j) .$$

Ugyancsak az 1.1.7. állitást összevetve az (1.2.5), (1.2.6) és (1.4.1) összefüggésekkel, kapjuk:

1.4.2. Állitás: Az FO-elemek esetén

$$\hat{c}_{i,n+1} = -u_i , \qquad i = 1,2,\ldots,m,$$

az FS-elemek esetén:

$$\hat{c}_{m+1,j} = -v_j , \qquad j = 1,2,\ldots,n.$$

Vegyünk még egy triviális állitást:

1.4.3. Állitás: Minden (i,j) báziselem esetén

$$\hat{c}_{i,j} = 0.$$

A potenciálok meghatározásának lényege az 1.4.1, 1.4.2. és az 1.4.3. állitások alapján felállitható egyenletrendszer megoldása; de a megoldás körülményei most kedvezőbbek mint általában. - A fiktiv báziselemeket tartalmazó nem-fiktiv sorok és oszlopok potenciáljai az S1, S2 pontok, valamint az 1.4.2. és az 1.4.3. állitások szerint azonnal adódnak (lásd a későbbiekben a P1, P2 pontokat). A már meghatározott potenciálokból az 1.4.1. állitás szerint további potenciálokat származtathatunk (lásd a P4, P5 pontokat). Figyelemre méltó, hogy az eljárás során az SS-elemek mindig sor-potenciált, az OS-elemek mindig oszlop-potenciált adnak meg. A fiktiv báziselemekhez tartozó potenciálokból kiindulva nem mindig határozható meg az összes potenciál. Látni fogjuk, hogy ilyenkor a bázis-elemek egy része u.n. hurko(ka)t alkot.

1.4.2. Definició: Huroknak nevezzük a DT báziselemeinek olyan nem üres halmazát, amely a DT minden sorából és oszlopából - ha onnan tartalmaz elemet - pontosan két báziselemet tartalmaz; viszont nincs olyan valódi részhalmaza, amelyre szintén igaz lenne ez az állitás.

Már a sajátelemek kijelölésekor megmondható, léteznek-e hurkok a DT-ben. Ugyanis a 2. részben be fogjuk bizonyitani: egy adott bázisvektorrendszer esetén a DT-ben pontosan akkor létezik hurok, ha valamely bázisvektort az S7 pont jelölte ki sajátelemnek; és pontosan annyi hurok létezik, ahány bázisvektort az S7 pont jelölt ki (lásd a 2.2.5. lemmát).

A hurok elemeinek megkeresését és egyvalamely eleméhez tartozó egyik potenciál megadását a HU1–HU3 hurok-algoritmus végzi; a hurokhoz tartozó további potenciálok a későbbiekben következő P4, P5 szabályok szerint határozhatók meg.

A hurok-algoritmus:

HU1. Nevezetes körülmény, hogy minden hurok tartalmaz (pontosan) egy S7 pont szerint kijelölt báziselemet (2.2.4. lemma a/ része); a hurok elemeinek felsorolása vele kezdhető. – Legyenek ennek indexei: i_0, j_1. A hurok elemei igy sorolhatók fel:

$$(i_0, j_1), \ (i_2, j_1), \ (i_2, j_3), \dots, \ (i_s, j_{s-1}) \ ;$$

ahol

$$(1.4.2) \qquad i_{2k} = w(m + j_{2k-1}) \ , \qquad j_{2k+1} = w(i_{2k})$$

és s az első olyan pozitiv index, amelyre:

$$(1.4.3) \qquad i_s = i_0$$

(s nyilván páros).

Megjegyzés: Az (1.4.2) egyenlőségek másképpen azt jelentik, hogy SS-elemtől az oszlopában levő OS-elemhez, illetve OS-elemtől a sorában levő SS-elemhez kell lépni. – Itt jelentkezik először a sajátelemeknek az előszóban már emlitett "útjelző" szerepe. (Igazolásul lásd a 2.2.5. lemmát!)

HU2. A HU1 pontban kijelölt hurok elemeit a kijelölés sorrendjében végigjárva, generálunk két véges számsorozatot:

$$p_0, p_1, \ldots, p_s ;$$
$$q_0, q_1, \ldots, q_s .$$

A sorozatok definiciója:

$$p_0 = 0,$$
$$q_0 = 1,$$

$$p_{2k} = \frac{c_{i_{2k}, j_{2k-1}} - p_{2k-1}}{f_{i_{2k}, j_{2k-1}}},$$

$$q_{2k} = - \frac{q_{2k-1}}{f_{i_{2k}, j_{2k-1}}},$$

$$p_{2k+1} = c_{i_{2k}, j_{2k+1}} - f_{i_{2k}, j_{2k+1}} \cdot p_{2k} ,$$

$$q_{2k+1} = - f_{i_{2k}, j_{2k+1}} \cdot q_{2k} .$$

(Számitástechnikai szempontból lényeges, hogy végül a $\{p_i\}$, $\{q_i\}$ sorozatokból csak az utolsó (s-edik) elemre van szükség. - Tehát, például a p_{i-1}-et csak addig kell tárolni, amig p_i-t nem határoztuk meg.)

HU3. Az (i_0, j_1)-hez tartozó u_{i_0} potenciál igy állitható elő:

$$u_{i_0} = \frac{p_s}{1-q_s} .$$

A HU1 pontot a 2. részben bebizonyitjuk(2.2.5. lemma). A HU2 és HU3 pontok viszont azonnal adódnak, ha a HU1-ben kijelölt hurok elemeire az 1.4.1. és az 1.4.3. állitások alapján felirható lineáris egyenletrendszert a Gauss módszerrel úgy oldjuk meg, hogy közben az egyenleteket a nekik megfelelő báziselemek HU1 szerinti sorrendjében vesszük figyelembe.

Ezek után a potenciálok meghatározását igy foglalhatjuk össze:

P1. Minden fiktiv SS-elem sorához rendeljük: $u_i = 0$.

P2. Minden fiktiv OS-elem oszlopához rendeljük: $v_j = 0$.

P3. Minden olyan SS-elem esetén, amelyet az S7 pontban jelöltünk, alkalmazzuk a hurok-algoritmust.

P4. Minden olyan SS-elem esetén, amelynek oszlopához már tartozik v_j (de a sorához még nem tartozik u_i), a sorához rendeljük:

$$u_i = \frac{c_{i,j} - v_j}{f_{i,j}} \quad . \tag{1.4.4}$$

P5. Minden olyan OS-elem esetén amelynek sorához már tartozik u_i (de az oszlopához még nem tartozik v_j) rendeljük:

$$v_j = c_{i,j} - f_{i,j} \cdot u_i \quad . \tag{1.4.5}$$

Ha végül maradt még meghatározatlan potenciál, akkor térjünk vissza a P4 ponthoz.

<u>Megjegyzések:</u> A P1, P2, P4, P5 pontokban i, j az SS-, illetve OS-elem indexei. Ha a Pk pont másképpen nem rendelkezik, akkor azt mindig a P(k+1) pont követi.

Ezzel túl vagyunk a DE3 ponton. A DE4 ponthoz már csak az 1.4.1 állitást kell alkalmazni, és utána az (1.1.11) feltétel vizsgálható. Ha az (1.1.11) feltétel teljesül, akkor a megoldást az aktuális bázishoz tartozó $x_{i,j}$ együtthatók szolgáltatják, az $\underline{x}$ megoldásvektor más komponensei zérusok. (Az (1.1.10) feltétel teljesülését külön nem kell vizsgálni, mert a DE1 pontban csak olyan induló bázismegoldást választunk és a DE5 pontban csak olyan bázismegoldásra térünk át, amely (1.1.10)-et kielégiti.)

Az 1.6. fejezetben látni fogjuk, hogy amint az S1-S7 algoritmust, ugyanúgy a P1-P5 algoritmust is általában csak egyszer kell az összes báziselemre alkalmazni, mert az egyes elemi bázistranszformációk alkalmával a potenciálok egy része többnyire változatlan marad. Az 1.2. fejezetben javasolt induló bázismegoldás esetén valamennyi potenciál zérus.

1.5. A BÁZISBA BEVONANDÓ VEKTOR (AKTUÁLIS BÁZISRA VONATKOZÓ) KOMPONENSEINEK MEGHATÁROZÁSA

A bázisba bevonandó elem (a fejezetben ezt $\underline{b}_0$ jelöli) aktuális bázisra vonatkozó komponenseinek meghatározásához először kijelölünk bizonyos gráfo(ka)t a DT-ben (oszlopút és /vagy sorút), amely(ek) a DT báziselemeiből és a $\underline{b}_0$-ból állnak; ezt követően a gráf(ok)hoz tartozó báziselemekhez u.n. kompenzációs együtthatókat rendelünk; majd a kompenzációs együtthatókból származtatjuk a $\underline{b}_0$ komponenseit.

Be kell vezetnünk néhány újabb fogalmat:

1.5.1. Definició: Legyen $\underline{b}_0, \underline{b}_1, b_2, \ldots$ a DT vektorainak (elemeinek) egy sorozata. A $\underline{b}_j$ tagot a sorozat hurokpontjának nevezzük, ha j a legkisebb olyan index, amelyre $\underline{b}_j = \underline{b}_i$ és $0 \leqq i < j$.

1.5.2. Definició: Legyen $\underline{b}_j$ a

$$\underline{b}_0, b_1, \ldots, \underline{b}_j, \ldots$$

sorozat hurokpontja, $\underline{b}_j = \underline{b}_i$, $0 < i < j$. A sorozat által generált csomón a

$$\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_{j-1}, \underline{b}_{i-1}, \underline{b}_{i-2}, \ldots, \underline{b}_0$$

sorozatot értjük.

A sorút, illetve az oszlopút egy-egy

$$\underline{b}_0, \underline{b}_1, \underline{b}_2, \ldots$$

véges sorozatot jelent, ahol $\underline{b}_i$ ($i > 0$) a DT báziseleme. Ez a véges sorozat másképpen egy leképezésnek tekinthető a természetes számok egy véges részhalmazából a $\{\underline{b}_0\} \cup B$ halmazba (B a báziselemek halmaza); és mint ilyent, magával a leképezési algoritmussal fogjuk definiálni. – Az itt leirt UT1–UT4 algoritmus <u>közvetlenül a sorútra vonatkozik</u>, de benne zárójelben elhelyeztük azokat a szövegrészeket, amelyeket az eredeti szöveg megfelelő helyeire helyettesitve az oszlopútra vonatkozó algoritmust kapunk.

Legyen adott a DT egy bázisa és rajta a sajátelemek egy kijelölése. Legyen $\underline{b}_0$ a DT olyan vektora, amely nem tartozik a bázishoz és nem FS-elem (nem FO-elem). Ekkor a $\underline{b}_0$-hoz tartozó <u>sorutat (oszloputat)</u> a következőképpen jelöljük ki:

UT1. Az út $\underline{b}_0$-lal kezdődik, $\underline{b}_1$ pedig legyen a $\underline{b}_0$ sorában (oszlopában) levő SS-elem (OS-elem); ha $\underline{b}_1$ fiktiv, akkor ez az út vége; különben a folytassuk az UT2 (UT3) pontban.

UT2. Ha már adott az út a $\underline{b}_j$ SS-elemig, akkor legyen $\underline{b}_{j+1}$ a $\underline{b}_j$ oszlopában levő OS-elem. Ha ez fiktiv, akkor ez az út vége; ha ez hurokpont, akkor következzék az UT4; egyébként folytassuk az UT3 szerint.

UT3. Ha már adott az út a $\underline{b}_j$ OS-elemig, akkor legyen $\underline{b}_{j+1}$ a $\underline{b}_j$ sorában levő SS-elem. Ha ez fiktiv, akkor ez az út vége; ha ez hurokpont, akkor következzék az UT4; különben folytassuk az UT2 szerint.

UT4. Ha már adott a $\underline{b}_0$, $\underline{b}_1$, b_2,..., $\underline{b}_j$ sorozat a $\underline{b}_j$ hurokpontig, legyen az út a sorozat által generált csomó.

Mint az látható, az UT1–UT4 algoritmussal nem-fiktiv $\underline{b}_0$-hoz sorutat és oszloputat is rendelhetünk, de FO-elemhez csak sorút, FS-elemhez csak oszlopút tartozik.

Ha azt is szükséges tudni, hogyan helyezkednek el a sor-, illetve oszlopút tagjai a DT-ben, akkor azokat célszerü a következők szerint indexelni:

- sorút esetén:

$$(i_0, j_{-1}),\ (i_0, j_1),\ (i_2, j_1),\ (i_2, j_3), \ldots$$

- oszlopút esetén:

$$(i_{-1}, j_0),\ (i_1, j_0),\ (i_1, j_2),\ (i_3, j_2), \ldots$$

Vegyük észre, hogy a hurokpontig az út kijelölése az (1.3.3) szerinti w függvénnyel igy irható le:

- sorút esetén:

$$i_{2k} = w(j_{2k-1} + m), \qquad k = 1, 2, \ldots;$$

$$j_{2k+1} = w(i_{2k}), \qquad k = 0, 1, 2, \ldots,$$

- oszlopút esetén:

$$j_{2k} = w(i_{2k-1}), \qquad k = 1, 2, \ldots;$$

$$i_{2k+1} = w(j_{2k} + m), \qquad k = 0, 1, 2, \ldots,$$

Miután meghatároztuk a sorutat és /vagy az oszloputat, ezek minden $\underline{b}_k \in B$ tagjához rendelünk egy c_k u.n. kompenzációs együtthatót. (Ennek semmi köze az (1.1.1) szerinti $\underline{c}^*$ vektorhoz.) A kompenzációs együtthatókat magával a hozzárendelési algoritmussal definiáljuk (E1,– E4):

E1. Fiktiv báziselemet tartalmazó sorút esetén a kompenzációs együtthatókat (c_i) a következő hozzárendeléssel kapjuk (i_0, j_{-1} ; a bázisba bevonandó elem indexei):

$$\underline{b}_1 = \underline{a}_{0,j_1} \longrightarrow c_1 = \frac{f_{i_0,j_{-1}}}{f_{i_0,j_1}} \;;$$

$$\underline{b}_2 = \underline{a}_{i_2,j_1} \longrightarrow c_2 = -c_1 \;;$$

...

$$\underline{b}_{2k} = \underline{a}_{i_{2k},j_{2k-1}} \longrightarrow c_{2k} = -c_{2k-1} \;;$$

$$\underline{b}_{2k+1} = \underline{a}_{i_{2k},j_{2k+1}} \longrightarrow c_{2k+1} = -\frac{f_{i_{2k},j_{2k-1}}}{f_{i_{2k},j_{2k+1}}} \cdot c_{2k} \;.$$

E2. Fiktiv báziselemet tartalmazó oszlopút esetén a hozzárendelés a következő:

$$\underline{b}_1 = \underline{a}_{i_1,j_0} \longrightarrow c_1 = 1 \;;$$

$$\underline{b}_2 = \underline{a}_{i_1,j_2} \longrightarrow c_2 = -\frac{f_{i_1,j_0}}{f_{i_1,j_2}} \cdot c_1 \;;$$

...

$$\underline{b}_{2k} = \underline{a}_{i_{2k-1},j_{2k}} \longrightarrow c_{2k} = -\frac{f_{i_{2k-1},j_{2k-2}}}{f_{i_{2k-1},j_{2k}}} \cdot c_{2k-1} \;;$$

$$\underline{b}_{2k+1} = \underline{a}_{i_{2k+1},j_{2k}} \longrightarrow c_{2k+1} = -c_{2k} \;.$$

E3. Fiktiv báziselemet nem tartalmazó sorút esetén előbb egy $\underline{b}_i \longrightarrow d_i$ hozzárendelést végzünk, ennek pontosan ugyanazok a szabályai mint az E1 pontban a $\underline{b}_i \longrightarrow c_i$ hozzárendelésnek. A d_i értékekből, most a c_i kompenzációs együtthatók így származtathatók:

$$c_i = d_i \cdot \alpha \qquad (i = 1,2,\ldots,s)\,, \tag{1.5.1}$$

ahol

$$\alpha = \frac{f_{i_0,j_{-1}}}{f_{i_0,j_1}\cdot d_1 + f_{i_s,j_{s-1}}\cdot d_s}\,, \tag{1.5.2}$$

és s a sorút $\underline{b}_0$-on kívüli tagjainak száma.

Megjegyzés: Itt és a továbbiakban a "tag" következetesen valamilyen sorozatban álló egyedet jelöl és lényegesen különbözik a "halmaz eleme" megjelöléstől. - Egy sorozat két különböző tagja lehet valamely halmaz egyazon eleme. - Így egy sorozat tagjainak száma általában nem egyenlő a sorozat tagjai halmazának elemszámával.

E4. Fiktiv báziselemet nem tartalmazó oszlopút esetén szintén előbb egy $\underline{b}_i \longrightarrow d_i$ hozzárendelést végzünk, aminek ugyanazok a szabályai, mint az E2-ben a $\underline{b}_i \longrightarrow c_i$ hozzárendelésnek. A d_i-ből c_i-t itt is (1.5.1) -hez hasonlóan kapjuk, de most

$$\alpha = \frac{1}{1 + d_s}\,. \tag{1.5.3}$$

Mármost legyen adott a DT egy bázisa és rajta a sajátelemek egy kijelölése. Legyen $\underline{b}_0$ a DT olyan vektora, amely nem tartozik az adott bázishoz. A 2. részben meg fogjuk mutatni, hogy $\underline{b}_0$-t a báziselemek következő lineáris kombinációja állitja elő:

- ha $\underline{b}_0$ FO-elem, úgy

(1.5.4) $$\underline{b}_0 = \sum_{k=1}^{s} c_k \cdot \underline{b}_k \quad ;$$

- ha $\underline{b}_0$ FS-elem, úgy

(1.5.5) $$\underline{b}_0 = \sum_{k=1}^{q} \hat{c}_k \cdot \hat{\underline{b}}_k \quad ;$$

- ha $\underline{b}_0$ nem fiktiv elem, úgy

(1.5.6) $$\underline{b}_0 = \sum_{k=1}^{s} c_k \cdot \underline{b}_k + \sum_{k=1}^{q} \hat{c}_k \cdot \hat{\underline{b}}_k \quad ;$$

ahol $\underline{b}_1, \underline{b}_2, \ldots, \underline{b}_s$ a $\underline{b}_0$-hoz tartozó sorút; $\hat{\underline{b}}_1, \hat{\underline{b}}_2, \ldots, \hat{\underline{b}}_q$ a $\underline{b}_0$-hoz tartozó oszlopút $\underline{b}_0$-on kivüli tagjai. Ha a $\underline{b}_0$-ot előállitó lineáris kombinációban az ugyanazon báziselemhez tartozó kompenzációs együtthatókat összevonjuk, akkor $\underline{b}_0$-nak épen az erre a báziselemre vonatkozó komponensét kapjuk. Miután a bázisba bevonandó $\underline{b}_0$ elem aktuális komponenseit igy meghatároztuk a szük keresztmetszet kiválasztása és a DE6 pont végrehajtása (ez most csak az $x_{i,j}$ programváltozók transzformációját jelenti) hasonlóan történhet, mint a szimplex algoritmusnál.

1.6. TOVÁBBI LÉNYEGES EGYSZERÜSITÉSEK

Ha valamely iterációs lépésben már adott a sajátelemek egy kijelölése és adottak a potenciálok, akkor a következő iterációs lépéshez (bázismegoldáshoz) tartozó sajátelemek, illetve potenciálok az előbbiekből az S1 - S7, illetve P1-P5 alapalgoritmusoknál egyszerübb algoritmussal származtathatók.

A sajátelemek származtatása:

SS1. Ha a bázisból kieső elem a sorúthoz tartozik, de az oszlopúthoz nem tartozik (vagy nincs oszlopút), akkor a bázisba éppen bevont elem $(\underline{b}_0)$ legyen SS-elem.

SS2. Ha a bázisból kieső elem az oszlopúthoz tartozik, de a sorúthoz nem tartozik (vagy nincs sorút), akkor a bázisba éppen bevont elem $(\underline{b}_0)$ legyen OS-elem.

SS3. Ha a bázisból kieső elem a sorúthoz is, oszlopúthoz is hozzátartozik, akkor csak az egyik utat vegyük figyelembe és aszerint az SS1 vagy az SS2 pontot alkalmazzuk.

SS4. Vegyük a $\underline{b}_0$-hoz tartozó sorutat, ha $\underline{b}_0$ SS-elem lett, különben pedig a $\underline{b}_0$-hoz tartozó oszlopútat. Vegyük az

$$(1.6.1) \qquad \underline{b}_0, \underline{b}_1, b_2, \dots, \underline{b}_s$$

sorozatot, ahol (1.6.1) azonos a megfelelő úttal, ha az út nem csomó; különben az út hurokpont előtti részét jelenti.

SS5. Az (1.6.1) sorozatban a régi (báziscsere előtti) SS-elemeket változtassuk át OS-elemmé, a régi OS-elemeket változtassuk át SS-elemmé; egészen addig, amig nem értünk a bázisból kiesett elemhez. Az (1.6.1) sorozatban a bázisból kiesett elem után álló, illetve az (1.6.1) sorozathoz nem tartozó báziselemek ugyanolyan sajátelemek maradnak, mint a báziscsere előtt.

Az S1–S7 algoritmus mellőzése felveti a kérdést: ezután hogyan dönthető el, hány hurok létezik a DT-ben? Tény, hogy az S7 pontra épülő kritériumtól elesünk. Csakhogy ez a probléma eljárásunk szempontjából teljesen közömbös, ugyanis a lényeges kérdés; az új báziselem bevonásával keletkezik-e új hurok? Ennek eldöntésére most az SS3 pontra épülő kritérium adható meg. – A 2. részben be fogjuk bizonyitani:

- Az új hurok keletkezésének szükséges és elegendő feltétele: a sajátelemek származtatásakor alkalmazni kellett az SS3 pontot (2.4.2.lemma).
- Az új hurok mindig tartalmazza az új báziselemet (2.4.2. lemma).

A potenciálok származtatása:

PP1. Ha az új báziselem fiktiv SS-elem, akkor a sorához tartozó új potenciál: $u_i = 0$; ha olyan nem-fiktiv SS-elem, amely nem tartozik hurokhoz, akkor a sorához tartozó új potenciált az (1.4.4) formula adja meg; amelyben v_j az új báziselem oszlopához tartozó régi potenciál. (i , j az új báziselem indexei.)

PP2. Ha az új báziselem fiktiv OS-elem, akkor az oszlopához tartozó új potenciál: $v_j = 0$; ha olyan nem-fiktiv OS-elem, amely nem tartozik hurokhoz, akkor az oszlopához tartozó új potenciált az (1.4.5) formula adja; ebben u_i az új báziselem sorához tartozó régi potenciál, (i,j az új báziselem indexei.)

PP3. Ha az új báziselemhez tartozik egy hurok, akkor (csak erre a hurokra) végrehajtjuk a hurok algoritmust; de a HU1 pont annyiban módosul, hogy a hurok elemeinek felsorolásakor a kezdőelemet igy választjuk:

a/ ha az új báziselem SS-elem, akkor legyen ez a kezdőelem,

b/ egyébként az új báziselem sorában levő SS-elem legyen a kezdőelem.

PP4. Minden olyan SS-elem esetén, amelynek az oszlopához már tartozik <u>új</u> v_j, de a sorához még nem tartozik <u>új</u> u_i, a sorához tartozó új potenciált az (1.4.4) formula adja; ebben v_j az oszlophoz tartozó <u>új</u> potenciált jelöli (i,j az SS-elem indexei.)

PP5. Minden olyan OS-elem esetén amelynek a sorához már tartozik <u>új</u> u_i, de az oszlopához még nem tartozik <u>új</u> v_j, az oszlopához tartozó új potenciált az (1.4.5) formula adja; ebben u_i a sorhoz tartozó <u>új</u> potenciált jelöli (i,j az OS-elem indexei.) Ha van még olyan SS-elem, amelyre a PP4 pont alkalmazható, akkor a folytatás a PP4-ben, különben a PP6 pont következik.

PP6. Minden olyan sor és oszlop esetén, amelyekhez az eddigi pontok nem rendeltek új potenciált, a régi potenciálok maradnak érvényben.

<u>Megjegyzés:</u> Ha a sajátelemek származtatásakor az SS3 pontban az SS1 szabályt alkalmaztuk, úgy a PP3 pontban az a/ verzió kerül sorra.

1.7. A DEGENERÁCIÓ PROBLÉMÁJA

A degeneráció (lásd az 1.1.10. definiciót) lényegében azt jelenti, hogy több különböző bázishoz ugyanaz a bázismegoldás tartozhat. Ez a számitásoknál elvileg bizonyos bázisok végtelen ciklusban való ismétlődéséhez vezethet. A degenerációval ilyenkor együtt jár az a jelenség, hogy valamely transzformációs lépésben nem tudunk egyértelmüen szük keresztmetszetet választani (lásd az SPX3 vagy a DE5 pontot); azaz az (1.1.15) feltétel vizsgálatakor van (legalább) két olyan i_1, és i_2 index amelyre az (1.1.15) feltétel egyszerre teljesül. Azonban az u.n. porturbációs eljárás (lásd [5]) ekkor is lehetővé teszi az egyértelmü választást, és egyben azt is biztositja, hogy a szimplex algoritmus degeneráció esetén is véges sok lépésben elvezet az optimális megoldáshoz (ha az létezik).

Legyen

$$\left[d_{i_1,1}, d_{i_1,2}, \ldots, d_{i_1,m} \right],$$

illetve

$$\left[d_{i_2,1}, d_{i_2,2}, \ldots, d_{i_2,m} \right]$$

a $\underline{D}^{-1}$ mátrix (lásd 1.1. fejezet) i_1. , illetve i_2. sora. Legyen k a legkisebb olyan index amelyre

$$\frac{d_{i_1,k}}{\hat{a}_{i_1,j_0}} \neq \frac{d_{i_2,k}}{\hat{a}_{i_2,j_0}} ; \tag{1.7.1}$$

Ekkor a perturbációs eljárás szerint az i_1 és i_2 indexek közül azt választjuk, amelyhez az (1.7.1)-beli hányadosok közül a kisebb tartozik. - Egyébként azt, hogy az (1.7.1) nem - egyenlőség valamely k-ra biztosan előfordul, a $\underline{D}^{-1}$ mátrix nem - singuláris volta biztositja.

Itt azt mutatjuk meg, hogy a degeneráció kivédésének ez a módja a disztribuciós eljárás esetén is megoldható. - Bár a $\underline{D}^{-1}$ mátrixot nem vezetjük, de az, ha szükséges, bármikor előállitható, méghozzá a mátrixinvertálásnál egyszerübb módon. Ráadásul, általában az (1.7.1) már kis k-ra teljesül, és akkor elegendő a $\underline{D}^{-1}$ mátrix első k oszlopát előállitani.

Az 1.1.7. állitásból az adódik, hogy a $\underline{D}^{-1}$ oszlopai éppen az indulásnál $\underline{E}$ alakú mátrix oszlopvektorainak az aktuális bázisra vonatkozó komponenseit tartalmazzák. Az AEF esetén (1.2.5) alapján úgy is fogalmazhatunk, hogy a $\underline{D}^{-1}$ mátrix oszlopvektorait a fiktiv elemek adtuális bázisra vonatkozó komponensei adják. Ezek szerint, ha például ismerni kivánjuk a $\underline{D}^{-1}$ mátrix 1. oszlopát, akkor az $\underline{a}_{1,n+1}$ vektor aktuális bázisra vonatkozó komponenseit kell ugyanúgy meghatároznunk, mint az 1.5. fejezetben a bázisba bevonandó elem komponenseit.

1.8. AZ ÁLTALÁNOS LINEÁRIS PROGRAMOZÁSI (NEM NLP) FELADAT ESETE

Tekintsük az általános lineáris programozási feladatot a következő formában:

(1.8.1)

$$\max\left\{\underline{c}^{*}\underline{x}:\ \underline{x}\geqq\underline{0},\ \underline{A}_1\underline{x}\leqq\underline{b}_1,\ \underline{A}_2\underline{x}=\underline{b}_2,\ \underline{A}_3\underline{x}\geqq\underline{b}_3,\right.$$

$$\left.(\underline{b}_1\geqq\underline{0}\ \ \underline{b}_2\geqq\underline{0},\ \ \underline{b}_3\geqq\underline{0})\right\}.$$

Az (1.8.1) feltételhalmazát másképpen igy is megadhatjuk:

$$\begin{bmatrix} \underline{A}_1 & \underline{0} & \underline{E}_1 & \underline{0} & \underline{0} \\ \underline{A}_2 & \underline{0} & \underline{0} & \underline{E}_2 & \underline{0} \\ \underline{A}_3 & -\underline{E}_3 & \underline{0} & \underline{0} & \underline{E}_3 \end{bmatrix} \cdot \begin{bmatrix} \underline{x} \\ \underline{v} \\ \underline{u}_1 \\ \underline{u}_2 \\ \underline{u}_3 \end{bmatrix} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \underline{b}_3 \end{bmatrix}$$

(1.8.2)

$$\underline{x}\geqq\underline{0},\ \underline{v}\geqq\underline{0},\ \underline{u}_1\geqq\underline{0},\ \underline{u}_2=\underline{0},\ \underline{u}_3=\underline{0},$$

$$(\underline{b}_1\geqq\underline{0},\ \underline{b}_2\geqq\underline{0},\ \underline{b}_3\geqq\underline{0}\quad \text{konstansok})$$

Itt $\underline{E}_i$ olyan kxk méretü egységmátrixot jelent, amelynél k az $\underline{A}_i$ mátrix sorainak számával egyenlő. Az (1.8.2)-ből látható, hogy az (1.8.1) feladat egy megengedett bázismegoldása esetén a bázisban az

(1.8.3)
$$\begin{bmatrix} \underline{0} & \underline{0} \\ \underline{E}_2 & \underline{0} \\ \underline{0} & \underline{E}_3 \end{bmatrix}$$

blokk oszlopvektorai nem szerepelhetnek. Ezért egy lehetséges bázismegoldás előállitása most külön problémát képez. Ennek megfelelően az általános LP-feladat megoldása két fázisban történik:

1. fázis: az (1.8.1) feladat egy lehetséges bázismegoldásának megadása;

2. fázis: a megoldás további javitása.

Az 1. fázisban úgy járhatunk el, hogy megoldjuk az

(1.8.4)
$$\max \left\{ \underline{1}^* \begin{bmatrix} \underline{A}_2 & \underline{0} \\ \underline{A}_3 & -\underline{E}_3 \end{bmatrix} \cdot \begin{bmatrix} \underline{x} \\ \underline{v} \end{bmatrix} : \quad \underline{x} \geqq \underline{0}, \quad \underline{v} \geqq \underline{0}, \quad \underline{A}_1 \underline{x} \leqq \underline{b}_1, \quad \underline{A}_2 \underline{x} \leqq \underline{b}_2, \left[\underline{A}_3, -\underline{E}_3\right] \cdot \begin{bmatrix} \underline{x} \\ \underline{v} \end{bmatrix} \leqq \underline{b}_3 \right\}$$

normál feladatot, amelynek adatai (együtthatói) az (1.8.1) feladatéval azonosak.

Mivel
$$\underline{1}^* \cdot \begin{bmatrix} \underline{A}_2 & \underline{0} \\ \underline{A}_3 & -\underline{E}_3 \end{bmatrix} \cdot \begin{bmatrix} \underline{x} \\ \underline{v} \end{bmatrix} = \underline{1}^* \cdot \begin{bmatrix} \underline{A}_2 \cdot \underline{x} \\ \underline{A}_3 \underline{x} - \underline{E}_3 \underline{v} \end{bmatrix} \leqq \underline{1}^* \cdot \begin{bmatrix} \underline{b}_2 \\ \underline{b}_3 \end{bmatrix}$$

azaz az (1.8.4) feladat célfüggvénye korlátos, a keresett maximum mindig létezik. Ha még az is teljesül, hogy

$$(1.8.5) \qquad \max \underline{1}^* \cdot \begin{bmatrix} \underline{A}_2 & \underline{0} \\ \underline{A}_3 & -\underline{E}_3 \end{bmatrix} \cdot \begin{bmatrix} \underline{x} \\ \underline{v} \end{bmatrix} = \underline{1}^* \cdot \begin{bmatrix} \underline{b}_2 \\ \underline{b}_3 \end{bmatrix}$$

akkor az (1.8.4) feladat optimális megoldása egyben az (1.8.1) feladatnak egy lehetséges megoldása lesz. Ha (1.8.5) nem áll fenn, akkor az (1.8.1) feladat nem is oldható meg, - a feltételrendszere üres halmazt határoz meg.

A 2. fázisban a szimplex algoritmus igy módosul:

M1. Az SPX1 pontban az 1. fázis által adott lehetséges bázismegoldásból indulunk ki.

M2. Az SPX2 pontban valamely bázismegoldás optimális voltának eldöntésénél az (1.1.11) kritérium helyett az

(1.8.6)
$$\left[\underline{c}^{*}-\underline{d}^{*}\underline{D}^{-1}\underline{A} \qquad -\underline{d}^{*}\underline{D}^{-1}\begin{bmatrix}\underline{0}\\ -\underline{E}_3\end{bmatrix} \qquad -\underline{d}^{*}\underline{D}^{-1}\begin{bmatrix}\underline{E}_1\\ \underline{0}\end{bmatrix}\right] \leqq \underline{0}^{*}$$

teljesülését követeljük meg.

M3. A még javitható bázismegoldás esetén az SPX3 pontban csak az

$$\begin{bmatrix}\underline{A}_1 & \underline{0} & \underline{E}_1\\ \underline{A}_2 & \underline{0} & \underline{0}\\ \underline{A}_3 & -\underline{E}_3 & \underline{0}\end{bmatrix}$$

blokkhoz tartozó oszlopvektor vihető be a bázisba.

A nem NLP-feladatot képező AEF esetén az itt leirt kétfázisú szimplex algoritmushoz hasonló kétfázisú disztribuciós eljárást alkalmazhatunk. Az elv teljesen azonos csak a technikai részletekben van különbség.

A nem NLP feladat esetén a DT-ben valahogy jelölni kell, hogy az egyes feltételek "$\leqq$", "=" vagy "$\geqq$" tipusúak-e. Ezért minden sorhoz, illetve oszlophoz hozzárendelünk egy p_i, illetve egy q_j értéket. Ez a szám 0, ha a (sornak illetve oszlopnak) megfelelő feltétel "=" tipusú; 1, ha a feltétel "$\leqq$" tipusú; -1, ha a feltétel "$\geqq$" tipusú.

A DT azon fiktiv

$(i,\ n+1)$ $\qquad i=1,2,\dots,m;$

$(m+1,j)$ $\qquad j=1,2,\dots,n$

rovatai, amelyeknél $p_i=-1$, illetve $q_j=-1$ nemcsak az

$$\underline{a}_{i,n+1}=\underline{e}_i\ ,\qquad \underline{a}_{n+1,j}=\underline{e}_{m+j}$$

vektorokat képviselik, hanem a $-\underline{a}_{i,n+1}$, illetve a $-\underline{a}_{m+1,j}$ vektorokat is. Ezzel a DT rovatai és a feladat mátrixának oszlopvektorai közti kölcsönösen egyértelmü kapcsolat itt csorbát szenved, de ez nem okoz nehézséget, ha szem előtt tartjuk a következőket:

a/ Az ugyanazon fiktiv DT rovathoz tartozó két vektor közül mindig csak egyik szerepelhet az aktuális bázisban, különben a bázisvektorok rendszere lineárisan függő lenne, ami ellentmondás.

b/ Valamilyen jelzéssel megkülönböztetjük, hogy a bázisban levő DT rovat a hozzátartozó két vektor közül melyiket képviseli.

c/ A sajátelemek kijelölése és a potenciálok meghatározása szempontjából teljesen közömbös, hogy az ugyanazon fiktiv elemhez tartozó két vektor közül melyik áll a bázisban.

d/ Ha az $(i,n+1)$ rovathoz a $-\underline{a}_{i,n+1}$ vektor is hozzátartozik, akkor a 2. fázisban feltételezhető, hogy a bázisban az $(i,\ n+1)$ rovat a $-\underline{a}_{i,n+1}$ vektort képviseli. Az 1. fázis eredményes befejezésekor ugyanis degenerációmentes esetben a tiltott $\underline{a}_{i,n+1}$ vektor már automatikusan kiesett a bázisból; degeneráció esetén pedig csak $x_{i,n+1}=0$ programváltozó értékkel szerepelhet a bázisban, és igy a $-\underline{a}_{i,n+1}$ vektorral anélkül kicserélhető, hogy a programváltozók értékei, a potonciálok értékei és a sajátelem kijelölés változnának. Hasonlók mondhatók el megfelelő $(m+1,j)$ rovat esetén is.

e/ Az E1–E4 pontokban (1.5. fejezet) úgy járhatunk el, mintha minden fiktiv DT-elem, amely a bázisban áll, az $\underline{a}_{i,n+1}$ $(\underline{a}_{m+1,j})$ vektort képviselné, azután az igy előálló lineáris kombinációban azokat a komponenseket, amelyeknek valójában mégis a $-\underline{a}_{i,n+1}$ $(-\underline{a}_{i,m+1,j})$ vektor felel meg, (-1)-gyel szorozzuk.

f/ Ha olyan fiktiv DT-elemet vonunk be a bázisba, amelyhez $-\underline{a}_{i,n+1}$ vagy $-\underline{a}_{m+1,j}$ alakú vektor is tartozik, akkor mindig ezek a vektorok kerülnek a bázisba; azaz miután az E1-E4 pontok szerint az $\underline{a}_{i,n+1}$ vagy $\underline{a}_{m+1,j}$ vektor komponenseit meghatároztuk, ezek (-1)-szeresét vesszük.

A disztribuciós eljárás 1. fázisában az

(1.8.7) $$\underline{\tilde{c}}^{*} = \underline{1}^{*} \begin{bmatrix} \underline{A}_2 & \underline{0} \\ \underline{A}_3 & -\underline{E}_3 \end{bmatrix}$$

vektor $\tilde{c}_{i,j}$ komponenseit igy számithatjuk:

(1.8.8) $$\tilde{c}_{i,j} = \alpha \cdot f_{i,j} + \beta ,$$

ahol

$$\alpha = \begin{cases} 1, & \text{ha } i \neq m+1,\ j \neq n+1 \text{ és } p_i \neq 1 ; \\ -1, & \text{ha } j = n+1 \text{ és } p_i = -1 ; \\ 0 & \text{egyébként;} \end{cases}$$

$$\beta = \begin{cases} 1, & \text{ha } i \neq m+1,\ j \neq n+1 \text{ és } q_j \neq 1; \\ -1, & \text{ha } i = m+1 \text{ és } q_j = -1 ; \\ 0 & \text{egyébként.} \end{cases}$$

Mivel $\tilde{c}_{i,j}$ az $i,j, f_{i,j}, p_i, q_j$ adatokból egyértelmüen származtatható, a számitások során külön tárolására nincs szükség, hacsak a futási idő csökkentése nem indokolja. Az (1.8.7) vektor transzformációját ugyanúgy végezhetjük a potenciálok segitségével, mint ahogy azt már a $\underline{c}^{*}$ vektorra az 1.4. fejezetben leirtak. - Természetesen az 1. fázisban a potenciálok a $\tilde{c}_{i,j}$-kel számitandók. Egyebekre nézve a szimplex algoritmussal analóg módon járunk el.

A disztribuciós eljárás második fázisában a DE1. pontban az első fázis által szolgáltatott bázismegoldásból indulunk ki. A célfüggvény együtthatók $\hat{c}_{i,j}$ transzformált értékeire optimális megoldás esetén a következőknek kell teljesülni:

(1.8.9) $\hat{c}_{i,j} \leqq 0$ $\quad i=1,2,\dots,m;$ $\quad j=1,2,\dots,n;$

(1.8.10) $p_i \cdot \hat{c}_{i,n+1} \leqq 0$ $\quad i=1,2,\dots,m;$

(1.8.11) $q_j \cdot \hat{c}_{m+1,j} \leqq 0$ $\quad j=1,2,\dots,n.$

Az (1.8.9) -(1.8.11) feltételek az (1.8.6) feltételnek az AEF esetére való adaptálását jelentik. A szimplex algoritmus M3. módositásának itt az felel meg, hogy olyan $\underline{a}_{i,n+1}$ ($\underline{a}_{m+1,j}$) vektort soha nem vonunk be a bázisba, amelyre $p_i \neq 1$ ($q_j \neq 1$).
Persze a $-\underline{a}_{i,n+1}$ ($-\underline{a}_{m+1,j}$) vektor $p_i=-1$ ($q_j=-1$) esetén is bevonható a bázisba.

1.9. A KLASSZIKUS ELOSZTÁSI FELADAT ESETE

Tekintsük az 1.2.2. definició szerinti feladatot. Az (1.2.3) és (1.2.4) alapján azonnal adódik, hogy a feladatnak pontosan akkor van lehetséges megoldása, ha

$$\sum_{i=1}^{m} t_i = \sum_{j=1}^{n} r_j \ . \qquad (1.9.1)$$

(A t_i és r_j értelmezését lásd az 1.2. fejezetben (1.2.9) alatt.)

Az "=" tipusú feltételrendszer és (1.9.1) miatt a fiktiv elemektől itt eltekinthetünk, és nélkülük (1.2.3) miatt a feladat $\underline{A}$ mátrixának rangja m+n-1 lesz. - Igy már valamely nem-fiktiv sorhoz vagy oszlophoz nem tudunk sajátelemet rendelni. Ezen körülménynek megfelelően itt átfogalmazzuk a klasszikus feladat esetére a sajátelem kijelölés, valamint az oszlop- és sorút kijelölés algoritmusát.

<u>A sajátelemek kijelölése (adott bázisra):</u>

KS1. Minden sorban, amelyben még nincs SS-elem és csak egy kijelöletlen báziselemet tartalmaz, ezt a jelöletlen báziselemet jelöljük SS-elemnek.

KS2. Minden oszlopban, amelyben még nincs OS-elem és csak egy jelöletlen báziselemet tartalmaz, ezt a báziselemet jelöljük OS-elemnek. Ha maradt még jelöletlen báziselem, akkor térjünk vissza a KS1. pontra.

A sorút (oszloput) meghatározása:

KU1. Ha a bázisba bevonandó elem sorában (oszlopában) nincs SS-elem (OS-elem), akkor a sorút (oszlopút) üres.

KU2. Egyébként elindulunk a bevonandó elemtől a sorában (oszlopában) levő SS-elemhez (OS-elemhez) ; ha ennek oszlopában (sorában) már nincs OS-elem (SS-elem), akkor ez az út vége, különben folytatjuk a KU3 pont szerint.

KU3. Az SS-elemből az oszlopában levő OS-elemhez lépünk; ha ennek sorában már nincs SS-elem, akkor ez az út vége, különben folytatjuk a KU4 pont szerint.

KU4. Az OS-elemből a sorában levő SS-elemhez lépünk; ha ennek oszlopában már nincs OS-elem, akkor ez az út vége, különben folytatjuk a KU3 pont szerint.

A fiktiv elemek elhagyása után visszamaradó DT m+n-1 rangja miatt már csak, m+n-1 egyenletet tudunk felirni (az 1.4.1 és az 1.4.3. állitások alapján) az m+n számú ismeretlen pontenciálra. Az igy előálló határozatlan egyenletrendszer egy önkényesen választott

(1.9.2) $\quad u_1, u_2, \ldots, u_m, v_1, v_2, \ldots, v_n$

megoldása általában különbözik az 1.4.1. definició szerinti potenciáloktól, de mindig van olyan q konstans, hogy

$$u_i = u_i + q \quad ; \quad v_j = v_j - q \quad ; \quad i = 1, 2, \ldots, m \; ; \quad j = 1, 2, \ldots, n.$$

Igy az 1.4.1. összefüggést a $c_{i,j} = c_{i,j} - (u_i + v_j)$ alakban alkalmazva, az (1.9.2) éppúgy alkalmas a célfüggvény együtthatók transzformálásához, mint az (1.4.1).

Az (1.2.3)-ból az is adódik, hogy a kompenzációs együtthatók hozzárendelése sor- és oszlopút esetén is igy egyszerüsödik.

(1.9.3) $\quad c_{2k} = -1, \quad c_{2k+1} = 1.$

A KU1-KU4 és (1.9.3) alapján az is könnyen belátható, hogy ha (i,j) az oszlopútnak is és a sorútnak is tagja, akkor figyelmen kivül hagyható, mert a bázisba bevonandó vektor $\underline{a}_{i,j}$-re vonatkozó komponense zérus lesz.

1.10. A MUNKAELOSZTÁSI PROGRAM. A DE BEMUTATÁSA EGY KONKRÉT FELADATON

Az általános elosztási feladat legtipikusabb alkalmazási területe a munkaelosztási program (lásd [7]). Természetesen ezen felül még a gyakorlat számos más területén létezhetnek olyan jelenségek, amelyeknek az AEF lehet az alkalmas matematikai modellje.

Tegyük fel, hogy n számú különböző - de a termelésben egymást helyettesiteni tudó - gép áll rendelkezésünkre $t_1, t_2, \dots, t_m$ hasznos időalappal egy tervidőszakban. Ugyanakkor n számú különböző terméket kell gyártanunk $r_1, r_2, \dots, r_n$ rendelési mennyiségekkel.

Ha $f_{i,j}$ az i. gépen a j. termék egységének gyártási ideje, $c_{i,j}$ pedig a fajlagos nyereség (árbevétel - munkaelosztástól függő költség), valamint $x_{i,j}$ az j. termékből az i. gépen gyártott mennyiség, akkor a maximális nyereséget hozó programot a következő modell optimális megoldása adja.

$$x_{i,j} \geqq 0 \quad (i = 1,2,\dots,m; \quad j = 1,2,\dots,n)$$

$$(1.10.1) \qquad \sum_i x_{i,j} = r_j$$

$$(1.10.2) \qquad \sum_j f_{i,j} x_{i,j} \leqq t_i$$

$$\sum_i \sum_j c_{i,j} x_{i,j} \longrightarrow \text{maximum}$$

Ha történetesen az r_j mennyiségek nem szigorú - szerződésekben vagy tervelőirásokban rögzitett - számok, hanem valamilyen piackutatási becslés szerint várhatóan értékesithető mennyiségeket jelentenek, úgy (1.10.1)-től eltérően

$$\sum_i x_{i,j} \leq r_j$$

alakú feltétel is állhat. Hasonlóan , ha valamilyen különleges érdekből az i. gép kapacitását teljes egészében ki kell használni, úgy (1.10.2)-től eltérően

$$\sum_j f_{i,j} \cdot x_{i,j} = t_i$$

feltétel lehet kivánatos.

Az is előfordulhat, hogy a j_0. terméket az i_0. gépen valamilyen technológiai ok miatt nem lehet gyártani, ilyenkor az f_{i_0,j_0} fajlagos is értelmetlen. A különleges körülményt $f_{i_0,j_0} = 0$ választással jelölhetjük, ami azt fogja mutatni, hogy a DT (i_0, j_0) eleme minden tekintetben figyelmen kivül hagyandó. - Ugyan az AEF definiciójában szerepelt az $f_{i,j} \neq 0$ kikötés, de ezt csak az aktuális bázisban levő, illetve a bázisba bevonandó elemek vonatkozásában használtuk ki. Igy azzal a kiegészitéssel, hogy $f_{i,j} = 0$ fajlagossal jellemzett DT-elemet (vektort) soha nem vonunk be a bázisba; az optimálisság (1.8.9) szerinti vizsgálatainál az ilyen vektorokat figyelmen kivül hagyjuk; valamint a 2. rész állitásaiban (lemmáiban) - hallgatólagosan - mindig olyan vektort (DT-elemet) értünk, amit nem ilyen fajlagos jellemez; eddig ismertetett eljárásunk ezen különleges körülmény mellett is alkalmazható.

A sajátelem fogalmon felépitett eljárásunkat a következő feladaton fogjuk bemutatni:

$$x_{i,j} \geqq 0 \qquad (i=1,2,3,4; \qquad j=1,2,3,4,5)$$

$$5 \cdot x_{1,1} + 2 \cdot x_{1,2} + 4 \cdot x_{1,3} + 4 \cdot x_{1,4} + 8 \cdot x_{1,5} \doteq 500$$

$$5 \cdot x_{2,1} + x_{2,2} + 10 \cdot x_{2,3} + 5 \cdot x_{2,4} + 4 x_{2,5} \leqq 100$$

$$x_{3,1} + x_{3,2} + 4 x_{3,3} + 2 x_{3,4} + 2 x_{3,5} \leqq 520$$

$$x_{4,1} + 5 x_{4,2} + 10 x_{4,3} + 2 x_{4,4} \leqq 300$$

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} \leqq 100$$

$$x_{1,2} + x_{2,2} + x_{3,3} + x_{4,2} \leqq 400$$

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \geqq 100$$

$$x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} \leqq 200$$

$$x_{1,5} + x_{2,5} + x_{3,5} \leqq 50$$

$$4 x_{1,1} + 5 x_{1,2} + 10 x_{1,3} + 2 x_{1,4} + 10 x_{1,5} +$$
$$+ 6 x_{2,1} + 3 x_{2,2} + 8 x_{2,3} + 5 x_{2,4} + 5 x_{2,5} +$$
$$+ x_{3,1} + 2 x_{3,2} + 7 x_{3,3} + 4 x_{3,4} + 3 x_{3,5} +$$
$$+ 2 x_{4,1} + 8 x_{4,2} + 20 x_{4,3} + 3 x_{4,4} \longrightarrow \text{maximum}$$

A feladat mint modell a gyakorlatban leirhatja 4 alternativ gépkapacitás (mert 4 nem-fiktiv sora van) és 5-féle megrendelés, (mert 5 nem-fiktiv oszlopa van) kapcsolatát. A feladat az $x_{4,5}$ változót nem értelmezte, ami azt jelentheti, hogy az 5. megrendelés a 4. kapacitásból nem teljesithető.

Tegyük fel, hogy valamely iterációs lépésben eljutottunk az (1.10.3) bázismegoldáshoz.

(1.10.3)

5	2	4 $\boxed{25}$	4	8 $\boxed{50}$	1	= 500
5 $\boxed{20}$	1	10	5	4	1	100
1	1 $\boxed{400}$	4	2	2	1 $\boxed{120}$	520
1	5	10 $\boxed{30}$	2	0	1	300
$\boxed{80}$		$\boxed{45}$	$\boxed{200}$			
100	400	$\geqq$ 100	200	50		

$f_{i,j}$
$\boxed{x_{i,j}}$

Az (1.10.3) táblázatot (DT-t) már a fiktiv elemekkel (rovatokkal, vektorokkal) is kiegészitettük. A jobb alsó sarokban "kinagyitott" rovat szimbólumai magyarázzák a bejegyzések értelmét. - Például $f_{1,3} = 4$ és $x_{1,3} = 25$. Pontosan azok a rovatok vannak az aktuális bázisban, amelyeknél az $x_{i,j}$ értéket feltüntettük. Ez nem NLP-feladat, és a megoldás folyamatában még az első fázisnál - valamely lehetséges bázismegoldás keresésénél - tartunk.

Ugyanis:

$$\underline{1}^{*} \begin{bmatrix} \underline{A}_2 & \underline{0} \\ \underline{A}_3 & -\underline{E}_3 \end{bmatrix} \cdot \begin{bmatrix} \underline{x} \\ \underline{v} \end{bmatrix} =$$

$$= 4 \cdot 25 + 8 \cdot 50 + 25 + 30 = 555 < 600 = \underline{1}^{*} \cdot \begin{bmatrix} \underline{b}_2 \\ \underline{b}_3 \end{bmatrix}$$

(Az itt meg nem magyarázott szimbólumok jelentését lásd az 1.8. fejezetben!)

Az S1–S7 algoritmust esetünkre alkalmazva, kapjuk (a kijelölés sorrendjében):

$\underline{a}_{3,6}$: SS-elem S1 szerint;

$\underline{a}_{5,1}$, $\underline{a}_{5,3}$, $\underline{a}_{5,4}$: OS-elemek S2 szerint;

$\underline{a}_{3,2}$: OS-elem S3 szerint;

$\underline{a}_{1,3}$, $\underline{a}_{2,1}$, $\underline{a}_{4,3}$: SS-elemek S4 szerint;

$\underline{a}_{1,5}$: OS-elem S3 szerint.

Az eredményt az (1.10.4) táblázat mutatja, ennek peremein már a $\underline{\tilde{c}}^{*}$ (lásd /1.8.7/-et és /1.8.8/-at) vektor alapján számitott potenciálokat is feltüntettük. ⁎ jelöli a bázisba bevonandó rovatot. (A p_1 és q_3 értelmezését lásd az 1.8 fejezetben!)

(1.10.4)

$v_j \rightarrow$ 0 0 0 0 −2

$u_i \downarrow$							
1,25	5	2	SS 4	4	OS 8		$= (p_1 = 0)$
0	SS 5	1	10	5	4	1	
0	1	OS 1	⁎ 4	2	2	SS 1	
0,1	1	5	SS 10	2	0	1	
	OS		OS	OS		SS/OS $f_{i,j}$	

$(\overset{\geq}{q}_3 = -1)$

Példaként az u_1 és a v_3 potenciálok meghatározását mutatjuk be. Mivel a 3. oszlop sajáteleme: (5,3) fiktiv, igy a P2 szabály szerint $v_3=0$. Másrészt $\tilde{c}_{1,3}=f_{1,3}+1=5$ (lásd (1.8.8)-ban), igy a P4 szabály alapján

$$u_1=\frac{\tilde{c}_{1,3}-v_3}{f_{1,3}}=\frac{5-0}{4}=1{,}25\,.$$

Az (1.10.4) szerint a (3,3) rovat bázisba vonásával a program javitható, mert

$$\tilde{c}_{3,3}-f_{3,3}\cdot u_3-v_3=1-0-0>0\,.$$

Ekkor a sorút: (3,3), (3,6); az oszlopút: (3,3), (5,3).
- Bár a (3,3) oszlopában három báziselem is van, az oszlopút kijelölésénél mégsem szorulunk találgatásra; az (5,3) OS-elem tulajdonsága egyértelmüen eligazit.

A kompenzációs együtthatók (lásd az 1.5 fejezet E1, E2 pontjait):

$$\underline{a}_{3,6}\longrightarrow\frac{f_{3,3}}{f_{3,6}}=4,$$

$$\underline{a}_{5,3}\longrightarrow 1.$$

Ebből a szük keresztmetszetre kapjuk:

$$\min\left\{\frac{x_{3,6}}{4}(=30),\ \frac{x_{5,3}}{1}(=45)\right\}=\frac{x_{3,6}}{4},$$

azaz a (3,6) rovat esik ki a bázisból. Egyidejüleg a (3,3) elem $\hat{x}_{3,3}=x_{3,6}/4=30$ programváltozó értékkel lép be a bázisba, és az (5,3) elemre $\hat{x}_{5,3} = x_{5,3} - 30 = 15$ lesz a programváltozó új értéke. További (i,j) elemekre: $\hat{x}_{i,j} = x_{i,j}$.

A báziscsere után a sajátelemek vonatkozásában az SS1 és az SS5 pont (1.6. fejezet) szerint csak annyi történik, hogy a 3. sorban (3,6) helyett a (3,3) rovat lesz az SS-elem. Ugyanakkor a potenciálok közül is a PP1, PP5, és PP6 szabályok szerint csak u_3 és v_2 változik meg; új értékeik: $u_3 = 0{,}25$, $v_2 = -0{,}25$.

Az (1.8.5) feltétel még mindig, nem teljesül, viszont $-v_2 = 0{,}25 > 0$ miatt az (5,2) fiktiv rovat bázisba vonásával a program javitható. Ezt a transzformációt már nem részletezzük, (1.10.5) alatt látható az eredménye.

(1.10.5)

5 4	2 5	SS 4 10 25	4 2	OS 8 10 50	1 0
SS 5 6 20	1 3	10 8	5 5	4 5	1 0
1 1	SS 1 2 340	OS 4 7 45	2 4	2 3	1 0
1 2	5 8	SS 10 20 30	2 3	0 0	1 0
OS 0 80	OS 0 60	0	OS 0 200	0	

OS SS	$f_{i,j}$
$c_{i,j}$	$x_{i,j}$

Most már (1.8.5) teljesül, ugyanis

$$f_{1,3} \cdot x_{1,3} + f_{1,5} \cdot x_{1,5} \cdot x_{1,3} + x_{3,3} + x_{4,3} =$$

$$= 4 \cdot 25 + 8 \cdot 50 + 25 + 45 + 30 = 600 = t_1 + r_3 ,$$

azaz áttérhetünk az eljárás második fázisára.

Újra ki kell számolni a potenciálokat, mert az első fázis során $c_{i,j}$ helyett a $\tilde{c}_{i,j}$ (lásd az (1.8.7)-et és (1.8.8)-at) mennyiségekből származtattuk őket.

Az (1.10.5) bázis esetében a potenciálok a következő sorrendben számitódnak ki;

v_1, v_2, v_4 : P2 szerint;

u_2, u_3 : P4 szerint;

v_3 : P5 szerint;

u_1, u_4 : P4 szerint;

v_5 : P5 szerint.

Most példaként a v_2, u_3, v_3 meghatározását kövessük végig:

$v_2 = 0$, mert (5,2) fiktiv 0S-elem;

$$u_3 = \frac{c_{3,2} - v_2}{f_{3,2}} = \frac{2-0}{1} = 2 ;$$

$$v_3 = c_{33} - f_{33} \cdot u_3 = 7 - 4 \cdot 2 = -1 ;$$

Az eredmény (már valamennyi potenciál):

$$u_i : \frac{11}{4} , \frac{6}{5} , 2, \frac{21}{10} ;$$

$$v_j : 0 , 0 , -1 , 0, -12 .$$

Az (1.10.5) bázismegoldás még nem optimális, mert a célfüggvényegyütthatók transzformált értékei között még van pozitiv, például:

$$\hat{c}_{3,5} = c_{3,5} - (f_{3,5} \cdot u_3 + v_5) = 3-(2 \cdot 2 - 12) = 11.$$

Tehát a DT (3,5) elemének bázisba vonásával javitható a program.

Az (1.10.6) tábla mutatja a (3,5)-höz tartozó sorút és oszlopút kijelölését. Folyamatos nyil jelzi a sorutat, szaggatott nyil az oszloputat; '*' jelzi a bázisba vonandó elemet.

(1.10.6)

		SS		OS	
SS					
	SS	OS		*	
		SS			
OS	OS		OS		

A sorút tagjai: (3,2) , (5,2) ; az oszlopút tagjai: (1,5), (1,3) , (3,3), (3,2) , (5,2). Mindkét út fiktiv elemben végződik. - A kompenzációs együtthatókat az E1, illetve E2 pontok szerint kell számitani.

Jelölje c_k a sorút k. tagjához tartozó kompenzációs együtthatót, hasonlóan $\hat{c}_k$ az oszlopút k. tagjához tartozó együtthatót.

A sorút tagjaihoz tartozó kompenzációs együtthatók számitása:

$$\underline{a}_{3,2} \longrightarrow c_1 = \frac{f_{3,5}}{f_{3,2}} = 2,$$

$$\underline{a}_{5,2} \longrightarrow c_2 = -c_1 = -2.$$

Az oszlopút tagjaihoz tartozó kompenzációs együtthatók számitása:

$$\underline{a}_{1,5} \longrightarrow \hat{c}_1 = 1,$$

$$\underline{a}_{1,3} \longrightarrow \hat{c}_2 = -\frac{f_{1,5}}{f_{1,3}} \cdot \hat{c}_1 = -2,$$

$$\underline{a}_{3,3} \longrightarrow \hat{c}_3 = -c_2 = 2,$$

$$\underline{a}_{3,2} \longrightarrow \hat{c}_4 = -\frac{f_{3,3}}{f_{3,2}} \cdot \hat{c}_3 = -8$$

$$\underline{a}_{5,2} \longrightarrow \hat{c}_5 = -c_4 = 8.$$

Igy az (1.10.5) bázis esetében az $\underline{a}_{3,5}$ vektort a következő lineáris kombináció állitja elő:

$$\underline{a}_{3,5} = \hat{c}_1 \underline{a}_{1,5} + \hat{c}_2 \cdot \underline{a}_{1,3} + \hat{c}_3 \cdot \underline{a}_{3,3} + (c_1 + \hat{c}_4)\,\underline{a}_{3,2} + (c_2 + \hat{c}_5)\,\underline{a}_{5,2} =$$

$$= h_{1,5}\,\underline{a}_{1,5} + h_{1,3}\,\underline{a}_{1,3} + h_{3,3}\,\underline{a}_{3,3} + h_{3,2}\,\underline{a}_{3,2} + h_{5,2}\,\underline{a}_{5,2} =$$

$$= \underline{a}_{1,5} - 2\underline{a}_{1,3} + 2\underline{a}_{3,3} - 6\underline{a}_{3,2} + 6\underline{a}_{5,2}$$

Ezek után már a szük keresztmetszet meghatározását és az $x_{i,j}$ programváltozók transzformációját úgyanúgy végezhetjük mint a szimplex algoritmusnál. A változásokat az (1.10.7) táblázatban foglaltuk össze.

(1.10.7)

(i,j)	$h_{i,j}$	$x_{i,j}$	$x_{i,j}$
(1,5)	1	50	40
(1,3)	-2	25	45
(3,3)	2	45	25
(3,2)	-6	340	400
(5,2)	$\boxed{6}$	$\boxed{60}$	
(3,5)			10

Mint látható, a szűk keresztmetszet (az (1.10.7)-ben külön bekeretezés jelzi) az (5,2) elemnél adódott, mert

$$Q = \frac{x_{5,2}}{h_{5,2}} = \min \left\{ \frac{x_{i,j}}{h_{i,j}} : h_{i,j} > 0 \right\}.$$

Igy lesz $\hat{x}_{5,2} = 0$, $\hat{x}_{3,5} = Q$, valamint a többi báziselem esetén:

$$\hat{x}_{i,j} = x_{i,j} - h_{i,j} \cdot Q$$

(Az (1.10.7)-ben nem szereplő báziselemek esetében nyilván $h_{i,j} = 0$.)

Ebben az iterációs lépésben az SS1-SS5 algoritmus csak a (3,5) és a (3,2) elemeket érinti, ha (5,2)-őt mint a sorút tagját tekintjük (SS3, SS1 pontok); a (3,5) SS-elem, a (3,2) OS-elem lesz; a többi báziselem megtartja a báziscsere előtti sajátelem tulajdonságát.

Az új bázis az (1.10.8) tábla szerint helyezkedik el a DT-ben. Látható, hogy (új) hurok keletkezett, ami összhangban áll a 2.4.2 lemmával.

(1.10.8)

		SS		OS	
SS					
	OS	OS		SS	
		SS			
OS			OS		

Határozzuk meg a (3,5) elemhez tartozó u_3 potenciált! A HU1. pont 1.6. fejezet szerinti módositásának megfelelően a következő sorrendben járjuk be a hurok elemeit:

(3,5), (1,5), (1,3), (3,3).

A HU2. pont szerinti számitások:

$$p_0 = 0, \qquad q_0 = 1,$$

$$p_1 = c_{3,5} - f_{3,5}\, p_0 = 3, \qquad q_1 = -f_{3,5}\, q_0 = -2,$$

$$p_2 = \frac{c_{1,5} - p_1}{f_{1,5}} = \frac{7}{8}, \qquad q_2 = -\frac{q_1}{f_{1,5}} = \frac{1}{4},$$

$$p_3 = c_{1,3} - f_{1,3}\, p_2 = \frac{13}{2}, \qquad q_3 = -f_{1,3}\, q_2 = -1,$$

$$p_4 = \frac{c_{3,3} - p_3}{f_{3,3}} = \frac{1}{8}, \qquad q_4 = -\frac{q_3}{f_{3,3}} = \frac{1}{4}.$$

A HU3. pont szerint:

$$u_3 = \frac{p_4}{1 - q_4} = \frac{1}{6}.$$

Az u_3 potenciálon kivül, a PP4 és PP5 pontok (1.6. fejezet) szerint, még a következő potenciálok változnak:

$$v_5 = c_{3,5} - f_{3,5}\, u_3 = \frac{8}{3},$$

$$u_1 = \frac{c_{1,5} - v_5}{f_{1,5}} = \frac{11}{12},$$

$$v_3 = c_{1,3} - f_{1,3} u_1 = \frac{19}{3},$$

$$u_4 = \frac{c_{4,3} - v_3}{f_{4,3}} = \frac{41}{10}.$$

$$v_2 = c_{3,2} - f_{3,2}\, u_3 = \frac{11}{6}.$$

A PP6. pont alapján az u_2, v_1, v_4 potenciálok megtartják a báziscsere előtti értéküket.

A program még mindig javitható: például

$$c_{1,2} = c_{1,2} - f_{1,2}\, u_1 - v_2 = 5 - \frac{11}{6} - \frac{11}{6} = \frac{8}{6} > 0.$$

A továbbiakban az eljárás az eddig elmondottak ismételt alkalmazását jelenti.

1.11. AZ ALGORITMUS GAZDASÁGOSSÁGA

A disztribuciós eljárás és a szimplex algoritmus hatékonyságának gyakorlati összehasonlitására PL/I forrásnyelven készültek számitógépes programok. Ugyanazokat a feladatokat mindkét eljárással lefuttattuk. Az output listákból kiderülnek a futási idők és az optimalizáláshoz szükséges iterációk száma.

Megjegyzés: Példaként mellékeltük egy feladat számitógépi táblázatait. Az 1. táblán látható a feladat inputja: a fajlagosok ($f_{i,j}$) ; a célfüggvény együtthatók ($c_{i,j}$) ; valamint a korlátok (t_i , r_j) a megfelelő feltételi relációkkal együtt. A példa egy 15 feltételes (M+N=15) feladatot mutat (a munkaelosztási probléma interpretációja szerint ez a 7 gépen 8 féle munka esetnek felel meg). A 2. táblán ennek a feladatnak a disztribuciós eljárás (ALTEOF névvel szignálva) és a szimplex algoritmus (LP440 névvel szignálva) szerinti megoldásai láthatók. (M és N jelentése a két algoritmus esetében különbözik; fennáll;

$$M_{LP440} = M_{ALTEOF} + N_{ALTEOF} ,$$

$$N_{LP440} = M_{ALTEOF} \cdot N_{ALTEOF} .)$$

A disztribuciós eljárás esetében a megoldási idő 15 másodperc, a szimplex algoritmus esetében 1 perc 15 másodperc volt.

- Az iterációk száma mindkét esetben: 20.

```
********       A L T E O F          ********        DATUM:82.04.09.    KEZDES:17.02.08.        ********

                    PROBA 7,8    (1,5),(2,5) KIESIK

A FELADAT INPUTJA:      M=  7      N=  8

A FAJLAGOSOK:

  1.SOR:
      1: 1.00000000E+00     2: 2.00000000E+00     3: 3.00000000E+00     4: 4.00000000E+00     6: 1.20000000E+01
      7: 1.10000000E+01     8: 1.00000000E+01
  2.SOR:
      1: 5.00000000E+01     2: 4.00000000E+00     3: 2.00000000E+01     4: 3.00000000E+00     6: 3.00000000E+00
      7: 2.00000000E+01     8: 1.10000000E+02
  3.SOR:
      1: 1.00000000E+00     2: 1.00000000E+00     3: 1.00000000E+00     4: 3.00000000E+01     5: 2.10000000E+01
      6: 5.00000000E+00     7: 7.00000000E+00     8: 9.00000000E+00
  4.SOR:
      1: 2.20000000E+01     2: 6.00000000E+00     3: 8.00000000E+00     4: 1.70000000E+01     5: 7.20000000E+00
      6: 1.00000000E+00     7: 1.00000000E+01     8: 1.72000000E+01
  5.SOR:
      1: 3.60000000E+00     2: 2.00000000E+00     3: 9.00000000E+00     4: 8.50000000E-01     5: 4.81000000E+01
      6: 3.00000000E+00     7: 8.00000000E+00     8: 1.05000000E+02
  6.SOR:
      1: 1.00000000E+01     2: 4.00000000E+00     3: 7.00000000E+00     4: 3.60000000E+01     5: 1.20000000E+01
      6: 5.00000000E+00     7: 6.00000000E+00     8: 7.20000000E+01
  7.SOR:
      1: 2.10000000E+02     2: 6.00000000E+00     3: 1.00000000E+00     4: 4.00000000E+00     5: 4.00000000E+00
      6: 7.00000000E+00     7: 2.00000000E+00     8: 6.00000000E+00

A CELFUGGVENY EGYUTTHATOI:

  1.SOR:
      1:-1.00000000E+00     2:-2.00000000E+01     3:-7.10000000E+01     4:-1.00000000E+01     6:-1.90000000E+01
      7:-1.10000000E+01     8: 1.00000000E+00
  2.SOR:
      1:-1.00000000E+00     2:-1.80000000E+01     3:-8.70000000E+01     4: 1.00000000E+00     6:-5.00000000E-01
      7:-9.20000000E+01     8: 1.00000000E+00
  3.SOR:
      1:-1.00000000E+01     2:-3.00000000E-01     3:-5.00000000E+00     4:-3.00000000E+01     5:-4.00000000E+00
      6:-7.52000000E+00     7:-8.00000000E+00     8:-4.00000000E+01
  4.SOR:
      1:-2.00000000E+01     2:-1.00000000E+01     3:-1.90000000E+01     4:-5.00000000E+01     5:-1.00000000E+00
      6:-2.00000000E+00     7:-1.00000000E+00     8:-6.00000000E+01
  5.SOR:
      1:-2.20000000E+00     2:-8.00000000E+00     3:-1.00000000E+00     4:-7.00000000E+01     5:-3.00000000E+00
      6:-4.20000000E+00     7:-1.30000000E+00     8:-5.60000000E+00
  6.SOR:
      1:-1.00000000E+00     2:-2.30000000E+00     3:-1.00000000E+00     4:-6.50000000E+00     5:-1.10000000E+01
      6:-1.62000000E+01     7:-1.00000000E+00     8:-1.00000000E+00
  7.SOR:
      1:-1.20000000E+01     2:-1.00000000E+00     3:-1.00000000E+00     4:-7.00000000E+01     5:-1.00000000E+00
      6:-2.00000000E+00     7:-3.00000000E+00     8:-4.00000000E+00

A KORLATOK:

  <=  1: 7.00000000E+04   <=  2: 5.00000000E+05   <=  3: 3.00000000E+04   <=  4: 3.00000000E+04   <=  5: 9.00000000E+03
  <=  6: 7.70000000E+04   <=  7: 1.00000000E+05   =   8: 2.00000000E+02   >=  9: 1.00000000E+01   >= 10: 1.00000000E+01
  >= 11: 2.00000000E+01   =  12: 1.00000000E+02   =  13: 5.00000000E+01   =  14: 7.60000000E+01   =  15: 8.70000000E+01
```

— 63 —

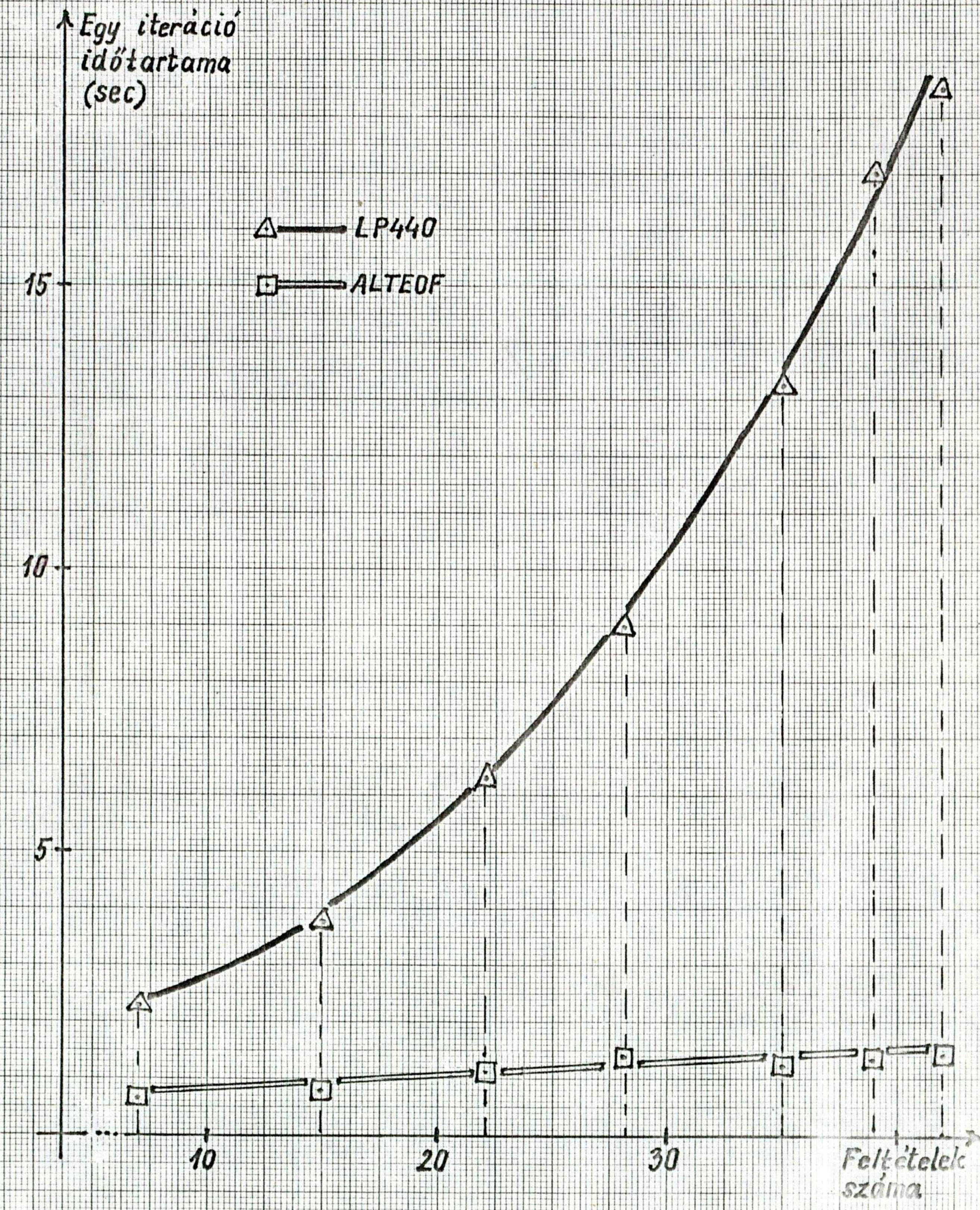
1. TÁBLA

2. TÁBLA

```
********          A L T E O F            ********          DATUM:82.04.08.     KEZDES:19.16.59.          ********

                  PROBA 7,8    (1,5),(2,5) KIESIK

A FELADAT OUTPUTJA:      M= 7      N= 8      ITERACIOK SZAMA=   10      OPTIMALIZALAS IDOTARTAMA=      15 SEC

OPTIMALIS MEGOLDAS

A (KIBOVITETT) X-VEKTOR:

     1   8: 8.70000000E+01        1   9: 1.91300000E+04        2   4: 1.66616667E+05        2   6: 5.00000000E+01
     3   2: 1.00000000E+01        3   9: 2.99900000E+04        4   5: 1.00000000E+02        4   7: 7.80000000E+01
     4   9: 2.85000000E+04        5   3: 1.00000000E+01        5   9: 8.91000000E+03        6   1: 2.00000000E+02
     6   9: 6.80000000E+04        7   9: 1.00000000E+05        8   4:-1.66596667E+05

A POTENCIALOK:

    2: 3.33333333E-01       8: 1.00000000E+00       9:-3.00000000E-01      10:-1.00000000E+00      12:-1.00000000E+00
   13:-1.50000000E+00      14:-1.00000000E+00      15: 1.00000000E+00

A CELFUGGVENY ERTEKE: 1.66687667E+05
```

```
***  LP440  ***                          PROBA 7,8    (1,5),(2,5) KIESIK

A FELADAT OUTPUTJA:      M= 15      N= 56      ITERACIOK SZAMA=   23      OPTIMALIZALAS IDOTARTAMA= 0. 1,15.

OPTIMALIS MEGOLDAS

A (KIBOVITETT) X VEKTOR:

     8: 8.700000E+01     12: 1.666167E+05     14: 5.000000E+01     18: 1.000000E+01     29: 1.000000E+02     31: 7.800000E+01
    35: 1.000000E+01     41: 2.000000E+02     57: 1.913000E+04     59: 2.999000E+04     60: 2.850000E+04     61: 8.910000E+03
    62: 6.800000E+04     63: 1.000000E+05     67:-1.665967E+05

AZ ARNYEKARAK:

     2: 3.333333E-01      8: 1.000000E+00      9:-3.000000E-01     10:-1.000000E+00     12:-1.000000E+00     13:-1.500000E+00
    14:-1.000000E+00     15: 1.000000E+00

A CELFUGGVENY ERTEKE: 1.666876667E+05
```

3. TÁBLA


Egy iteráció időtartama (sec)
LP440
ALTEOF
15
10
5
10
20
30
Feltételek száma

A feladatok méretét itt csak a feltételek számával jellemezzük. (A feladatokat úgy választottuk meg, hogy a változók száma közelitőleg arányos volt a feltételek számának négyzetével.) Mivel a feladat mérete az iterációk számát szigorúan nem determinálja, ezért elsősorban nem a teljes megoldási időre, hanem az egy iterációra eső átlagos időtartamra voltunk kiváncsiak.

A disztribúciós eljárás esetében egy-egy iteráció alkalmával a sajátelem-tulajdonságokat, a potenciálokat, és a programváltozókat kell transzformálni; ezen adathalmazok mérete a feltételek számával arányos. A szimplex algoritmus esetében viszont egy iterációs lépésben a feltételek számának négyzetével arányos méretü adathalmazt kell transzformálni (gondoljunk a bázisvektorok mátrixának inverzére). Ezért előzetesen úgy becsültük meg, hogy az egy iterációra eső átlagos időnek a feltételek számától a disztribuciós eljárás esetében lineárisan, a szimplex eljárás esetében négyzetesen kell függni. A különböző méretü feladatok megoldásához szükséges időket, iterációszámokat és az egy iterációra eső átlagos időtartamokat táblázatba foglaltuk, illetve a legutóbbinak a feltételek számától való függését grafikonon (3. tábla) is feltüntettük. Az elméleti becsléseket a tapsztalat igazolni látszik.

Felt. száma	Iterációk száma		Optimaliz. időtartama (sec)		Egy iterációra eső idő (sec)	
	LP 440	AL/TE OF	LP 440	AL/TE OF	LP 440	AL/TE OF
7	12	12	28	8	2,33	0,66
15	20	20	75	15	3,75	0,75
22	130	137	819	147	6,3	1,07
28	138	138	1242	193	9	1,4
35	102	102	1346	122	13,2	1,2
39	156	156	2652	203	17	1,3
42	96	96	1772	135	18,45	1,41

2. RÉSZ

AZ ALGORITMUS BIZONYITÁSA

2.1. ALAPVETŐ FOGALMAK (LÁNCOK, KÖRÖK, HURKOK, CSOMÓK)

A tárgyalt kijelentéseket állitás, illetve lemma nevekkel jelöltük. Az előbbi alatt olyan kijelentést értünk, amely a definiciókból és a vonatkozó feltételekből triviálisan következik; az utóbbi pedig olyan kijelentést jelöl, amely több-kevesebb bizonyitást igényel, illetve több korábbi kijelentés együttes figyelembe vételével látható be.

2.1.1. Lemma: Legyen $p > 1$ és $q > 1$ egész szám. Ha a DT p számú nem-fiktiv sorának és q számú nem-fiktiv oszlopának kereszteződésében szereplő elemek közül kiemelhető legalább $p+q+1$ számú különböző elem, akkor a kiemelt $\underline{a}_{i,j}$ vektorok rendszere lineárisan függő.

Bizonyitás: Az (1.2.2) és az (1.2.5)-ből következik, hogy a p számú sor és q számú oszlop kereszteződésében szereplő valamennyi (összesen $p \cdot q$ számú elem együttvéve kifejezhető e sorokhoz és oszlopokhoz tartozó, összesen $p+q$ számú fiktiv elem lineáris kombinációjaként; azaz a lemma szerinti $p+q+1$ elemü vektorrendszer rangja legfeljebb : $p+q$, és igy lineárisan függő. Qu. e. d.

2.1.1. Definició: A DT $\underline{a}$, $\underline{b}$ elempárjain értelmezzük a következő függvényt:

(2.1.1)

$$\varphi(\underline{a},\underline{b}) = \begin{cases} 1, & \text{ha } \underline{a} \neq \underline{b} \text{ és egy sorban vannak;} \\ -1, & \text{ha } \underline{a} \neq \underline{b} \text{ és egy oszlopban vannak;} \\ 0 & \text{egyébként.} \end{cases}$$

A $\varphi(\underline{a},\underline{b})$ értéket másképpen az $\underline{a}, \underline{b}$ pár fázisának nevezzük.

2.1.2. Definició: Ha $\{\underline{a}_k\}$ a DT vektorainak legalább kéttagú sorozata, és ennek két szomszédos $(\underline{a}_k, \underline{a}_{k+1})$ tagjára fennáll: $\varphi(\underline{a}_k, \underline{a}_{k+1}) \neq 0$, akkor az $\underline{a}_k, \underline{a}_{k+1}$ párt lépésnek nevezzük, és megfelelően sorfázisú ($\varphi = 1$) vagy oszlopfázisú ($\varphi = -1$) lépésről beszélünk.

A következő állitás a DT szerkezete(1.2.10), a φ értelmezése (2.1.1), valamint az FO-, FS-elemek értelmezése (1.2.4. definició) alapján látható be:

2.1.1. Állitás: Legyenek $\underline{a}, \underline{b}, \underline{c}$ a DT elemei, ekkor fennállnak:

a/ $\varphi(\underline{a}, \underline{b}) = \varphi(\underline{b}, a)$;

b/ ha $\varphi(\underline{a}, \underline{b}) = \varphi(\underline{b}, \underline{c}) \neq 0$, akkor
$\varphi(\underline{a}, \underline{b}) = \varphi(\underline{a}, \underline{c})$;

c/ ha $\varphi(\underline{a}, \underline{b}) = 1$ és $\underline{a}, \underline{b}$ közül pontosan egyik fiktiv, akkor a fiktiv vektor FO-elem;

d/ ha $\varphi(\underline{a}, \underline{b}) = -1$ és $\underline{a}, \underline{b}$ közül pontosan egyik fiktiv, akkor a fiktiv vektor FS-elem.

2.1.3. Definició: Láncnak nevezzük a DT vektorainak olyan - legalább kéttagú - sorozatát, amelynél teljesülnek:

a/ $\varphi(\underline{a}_1, \underline{a}_2) \neq 0$,

b/ bármely három szomszédos $\underline{a}_i, \underline{a}_{i+1}, \underline{a}_{i+2}$ tag esetén: $\varphi(\underline{a}_i, \underline{a}_{i+1}) = -\varphi(\underline{a}_{i+1}, \underline{a}_{i+2})$.

A lánc definiciójából azonnal adódik:

2.1.2. Állitás: Ha az

$$\underline{a}_1, \underline{a}_2, \dots, \underline{a}_k$$

sorozat lánc, akkor

a/ az $\underline{a}_k, \underline{a}_{k-1}, \ldots, \underline{a}_2, \underline{a}_1$ sorozat is lánc

b/ $1 \leq i < j \leq k$ esetén az

$\underline{a}_i, \underline{a}_{i+1}, \underline{a}_{i+2}, \ldots, \underline{a}_{j-1}, \underline{a}_j$

sorozat is lánc.

2.1.3. Állitás: Ha egy láncnak van többszörös tagja - azaz olyan $\underline{a}_j$, amelyre $\underline{a}_j = \underline{a}_i$ és $i \neq j$ -, akkor a láncnak legalább öt tagja van.

A lánc és a hurokpont definiciójából (1.5.1. definició) adódik:

2.1.4. Állitás: Legyen az

$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j, \ldots$

lánc hurokpontja $\underline{a}_j$, akkor teljesülnek a következők:

a/ az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_{j-1}$ tagok között nincs két azonos vektor;

b/ a láncnak csak egy hurokpontja van;

c/ csak egy olyan i $(1 \leq i < j)$ index van, amelyre $\underline{a}_i = \underline{a}_j$.

A 2.1.2. és a 2.1.3. állitásokból következik:

2.1.5. Állitás: Ha az $\underline{a}_1, \underline{a}_2, \ldots \underline{a}_j$, láncnak $\underline{a}_j$ hurokpontja, és $i < j$, $\underline{a}_i = \underline{a}_j$, akkor $j - i = 4$.

2.1.4. Definició: Az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j$ véges láncot körnek nevezzük, ha az utolsó tagja ($\underline{a}_j$) a hurokpontja, és $\underline{a}_j = \underline{a}_1$.

A kör definiciójából és a 2.1.2 állitás a/ részéből következik:

2.1.6. Állitás: Ha az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j$ sorozat kör, akkor az $\underline{a}_j, \underline{a}_{j-1}, \ldots, \underline{a}_2, \underline{a}_1$ sorozat is kör.

2.1.2. Lemma: Legyen $\underline{b}_1, \underline{b}_2, \dots, \underline{b}_j$ olyan kör, amelyben $\varphi(\underline{b}_1, \underline{b}_2) \neq \varphi(\underline{b}_{j-1}, \underline{b}_j)$, és legyen i: $1 < i < j$. Tekintsük a következő $\{\underline{a}_k\}$ sorozatot:

(2.1.2)

$$\underline{a}_1 = \underline{b}_i, \ \underline{a}_2 = \underline{b}_{i+1}, \dots, \underline{a}_{j-i+1} = \underline{b}_j \ (= \underline{b}_1),$$

$$\underline{a}_{j-i+2} = \underline{b}_2, \dots, \underline{a}_{j-1}, \underline{a}_j = \underline{b}_i \ (= \underline{a}_1).$$

Ekkor az $\underline{a}_k$ sorozat is kör, amelynek hurokpontja:

$$\underline{a}_j \ (= \underline{a}_1 = \underline{b}_i).$$

Bizonyitás: A 2.1.2. állitás szerint az

(2.1.3) $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{j-i+1}$;

(2.1.4) $\underline{a}_{j-i+2}, \dots, \underline{a}_{j-1}, \underline{a}_j$

sorozatok külön-külön láncot alkotnak. Miután

$$(0 \neq) \ \varphi(\underline{a}_{j-i}, \underline{a}_{j-i+1}) = \varphi(\underline{b}_{j-1}, \underline{b}_j) \neq$$

$$\neq \varphi(\underline{b}_1, b_2) = \varphi(\underline{a}_{j-i+1}, \underline{a}_{j-i+2}) \ (\neq 0),$$

igy a (2.1.3) sorozatot a (2.1.4) sorozattal folytatva megint láncot kapunk.

A 2.1.4. állitás a/ részéből és az $\{\underline{a}_k\}$ sorozat definiciójából az is adódik, hogy az

$$\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{j-1}$$

tagok közt nincs két azonos elem, és $\underline{a}_j = \underline{a}_1$. Tehát az $\{\underline{a}_k\}$ lánc egyben kör is. Qu. e.d.

2.1.3. Lemma: A 2.1.2. lemma feltételeinek eleget tevő kör a DT minden sorából és oszlopából – amelyből tartalmaz elemet – legalább két elemet tartalmaz.

<u>Bizonyitás:</u> Ha $1<i<j$, akkor a lánc definiciójából adódik:

$$\varphi(\underline{b}_{i-1}, \underline{b}_i) = -\varphi(\underline{b}_i, \underline{b}_{i+1}) \neq 0,$$

$$\underline{b}_{i-1} \neq \underline{b}_i ,$$

$$\underline{b}_i \neq \underline{b}_{i+1} .$$

Tehát a $\underline{b}_i$ sorából is oszlopából is van még további tagja a körnek.

Miután $\varphi(\underline{b}_1, \underline{b}_2) \neq \varphi(\underline{b}_{j-1}, \underline{b}_j)$,

a $\underline{b}_1 (= \underline{b}_j)$ sorára és oszlopára hasonlóan bizonyitható be az állitásunk. Q.u. e. d.

A lánc és hurokpont definiciójából, valamint a 2.1.4. állitás a/ pontjából adódik:

2.1.7. <u>Állitás:</u> Ha az $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_j, \dots$ láncnak $\underline{a}_j$ hurokpontja és $\underline{a}_j = \underline{a}_i$; $1<i<j$, akkor az

$$\underline{a}_{i-1}, \underline{a}_{j-1}, \underline{a}_j, \underline{a}_{i+1}$$

vektorok között nincs két azonos, továbbá a

$$\varphi(\underline{a}_{j-1}, \underline{a}_j) = \varphi(\underline{a}_j, \underline{a}_{i+1}),$$

$$\varphi(\underline{a}_{j-1}, \underline{a}_j) = \varphi(\underline{a}_j, \underline{a}_{i-1})$$

egyenlőségek közül pontosan egyik igaz.

A lánc és a hurokpont definiciójából, valamint a 2.1.2. állitás alapján belátható:

2.1.8. Állitás: Ha az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j, \ldots$ láncnak $\underline{a}_j$ hurokpontja, és $\underline{a}_j = \underline{a}_i$, $1 < i < j$, akkor az

$$(2.1.5) \qquad \underline{a}_1, \underline{a}_2, \ldots, \underline{a}_{j-1}, \underline{a}_{i-1}, \underline{a}_{i-2}, \ldots, \underline{a}_1,$$

$$(2.1.6) \qquad \underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j, \underline{a}_{i-1}, \underline{a}_{i-2}, \ldots, \underline{a}_1$$

sorozatok közül pontosan egyik lánc.

2.1.5. Definició: A 2.1.8. állitás (2.1.5) és (2.1.6) alakú sorozatai közül azt, amelyik lánc, az

$$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j$$

lánc által generált csomónak nevezzük.

Megjegyzés: Nyilvánvaló, hogy az UT1, UT2, UT3 pontokban (1.5. fejezet) egy láncot képezünk. Az ilyen tipusú láncokra szoritkozva, a 2.1.5. definició és az 1.5.2. definició ekvivalensek (2.3.1. lemma).

A 2.1.4. a/ állitás, a 2.1.5. állitás és a 2.1.5. definició alapján belátható:

2.1.9. Állitás: Ha az $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j, \ldots$ láncnak $\underline{a}_j$ hurokpontja, $\underline{a}_j = \underline{a}_i$, $1 < i < j$, akkor az

$$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j$$

lánc által generált csomónak az

$$(2.1.7) \qquad \underline{a}_{i+1}, \underline{a}_{i+2}, \ldots, \underline{a}_{j-1}$$

elemek csak egyszeres tagjai, és a (2.1.7) sorozat legalább négy tagból áll.

2.1.4. Lemma: Legyen

$$(2.1.8) \qquad \underline{b}_1, \underline{b}_2, \dots, \underline{b}_s$$

olyan lánc, amelyre fennáll:

(A) $\underline{b}_1$ és $\underline{b}_s$ fiktiv;

(B) a többi vektora nem fiktiv,

(C) van legalább egy nem fiktiv tagja.

Legyen G a lánc tagjainak halmaza. Ekkor a G vektorrendszer lineárisan függő.

Bizonyitás: Itt azt az esetet tárgyaljuk, amikor a lánc sorfázisban kezdődik és oszlopfázisban végződik. - Az ilyen szempont szerint létező további három esetben (oszlopfázis-sorfázis; sorfázis-sorfázis; oszlopfázis-oszlopfázis) a bizonyitás nem különbözik lényegesen az itt előadottól.

Esetünkben a sorozat következő átjelölése utal a lépések fázisára (i és j a DT-beli sor és oszlopindexek):

$$\underline{b}_1 = \underline{a}_{i_1, j_0}, \quad \underline{b}_2 = \underline{a}_{i_1, j_2}, \dots$$

$$\dots, \underline{b}_{2k-1} = \underline{a}_{i_{2k-1}, j_{2k-2}}, \quad \underline{b}_{2k} = \underline{a}_{i_{2k-1}, j_{2k}} \quad ;$$

$$k = 1, 2, \dots, s.$$

Mivel

$$\varphi(\underline{b}_{2k-1}, \underline{b}_{2k}) = 1,$$

$$\varphi(\underline{b}_{2k}, \underline{b}_{2k+1}) = -1,$$

$$\varphi(\underline{b}_{s-1}, \underline{b}_s) = -1,$$

igy $\underline{b}_s$-re a

$$\underline{b}_s = \underline{a}_{i_s, j_{s-1}} = \underline{a}_{i_{2q-1}, j_{2q-2}}$$

alak adódik ($q > 1$).

Végezzük el az 1.5. fejezet E2. pontjában leirt $\underline{b}_k \longrightarrow c_k$ hozzárendelést.

Az $f_{i,j} \neq 0$ tulajdonságból adódóan, igy csupa nem-zérus c_k együtthatót kapunk.

Legyen $1 \leqq k \leqq q$ egész szám, és

$$\underline{h}_{2k-1} = \sum_{r=1}^{2k-1} c_r \cdot \underline{b}_r \; . \tag{2.1.9}$$

Megmutatjuk, hogy

$$\underline{h}_s = \underline{h}_{2q-1} = \sum_{r=1}^{s} c_r \cdot \underline{b}_r = \underline{0} \; . \tag{2.1.10}$$

Ehhez előbb be kell látni, hogy $1 \leqq k < q$ esetén

$$\underline{h}_{2k-1} = c_{2k-1} \cdot f_{i_{2k-1}, j_{2k-2}} \cdot \underline{a}_{i_{2k-1}, n+1} \tag{2.1.11}$$

Az (A) miatt $\underline{b}_1$ fiktiv; (B), (C) miatt $\underline{b}_2$ nem fiktiv és $\varphi(\underline{b}_1, \underline{b}_2) = 1$. Igy a 2.1.1. c/ állitás szerint $\underline{b}_1 = \underline{a}_{i_1, n+1}$, és az (1.2.5), (1.2.7) összefüggéseket figyelembe véve, k=1 esetén (2.1.11) teljesül. Most $1 \leqq k < q-1$ esetére feltéve (2.1.11) fennállását, k+1-re kapjuk:

$$\underline{h}_{2k+1} = \underline{h}_{2k-1} + c_{2k} \cdot \underline{b}_{2k} + c_{2k+1} \cdot \underline{b}_{2k+1} =$$

$$= c_{2k-1} \cdot f_{i_{2k-1}, j_{2k-2}} \cdot \underline{a}_{i_{2k-1}, n+1} +$$

$$- \frac{f_{i_{2k-1}, j_{2k-2}}}{f_{i_{2k-1}, j_{2k}}} \cdot c_{2k-1} \cdot \left(f_{i_{2k-1}, j_{2k}} \cdot \underline{a}_{i_{2k-1}, n+1} + \underline{a}_{m+1, j_{2k}} \right) +$$

$$+ \frac{f_{i_{2k-1}, j_{2k-2}}}{f_{i_{2k-1}, j_{2k}}} \cdot c_{2k-1} \cdot \left(f_{i_{2k+1}, j_{2k}} \cdot \underline{a}_{i_{2k+1}, n+1} + \underline{a}_{m+1, j_{2k}} \right) =$$

$$= c_{2k+1} \cdot f_{i_{2k+1}, j_{2k}} \cdot \underline{a}_{i_{2k+1}, n+1} \; .$$

Tehát $1 \leq k < q$ esetén (2.1.11) fennáll. (Itt az 1.5. fejezet E2 pontjában levő hozzárendelési utasitás tulajdonságait, valamint az (1.2.3), (1.2.5) összefüggéseket sorozatosan kihasználtuk.)

Az (A) miatt $\underline{b}_s$ fiktiv; (B), (C) miatt $\underline{b}_{s-1}$ nem fiktiv és $\varphi(\underline{b}_{s-1}, \underline{b}_s) = -1$.

Igy a 2.1.1. állitás c/ része szerint $\underline{b}_s = \underline{a}_{m+1, j_{2q-2}}$.

Ezek után:

$$\underline{h}_s = \underline{h}_{s-2} + c_{s-1} \cdot \underline{b}_{s-1} + c_s \cdot \underline{b}_s =$$

$$= c_{2q-3} \cdot f_{i_{2q-3}, j_{2q-2}} \cdot \underline{a}_{i_{2q-3}, n+1} -$$

$$- \frac{f_{i_{2q-3}, j_{2q-4}}}{f_{i_{2q-3}, j_{2q-2}}} \cdot c_{2q-3} \cdot (f_{i_{2q-3}, j_{2q-2}} \cdot \underline{a}_{i_{2q-3}, n+1} +$$

$$+ \underline{a}_{m+1, j_{2q-2}}) +$$

$$+ \frac{f_{i_{2q-3}, j_{2q-4}}}{f_{i_{2q-3}, j_{2q-2}}} \cdot c_{2q-3} \cdot \underline{a}_{m+1, j_{2q-2}} = \underline{0}$$

Ezzel a (2.1.10) egyenlőséget beláttuk.

Még meg kell mutatni, hogy ebben a zéróvektort előállitó lineáris kombinációban valamely G-beli vektor nem-zéró együtthatóval szerepel. Mivel a c_k együtthatók zérustól különbözőek, igy ehhez elegendő belátni, hogy van olyan G-beli vektor, amely a (2.1.8) sorozatnak csak egyszeres tagja.

Ha a (2.1.8)-nak nincs hurokpontja vagy az kör, akkor az egyszeres tag létezése nyilvánvaló (lásd a 2.1.4. állitás a/ részét). Ha viszont (2.1.8) nem kör, de van hurokpontja, akkor tekintsük helyette az általa generált csomót.

A csomónak a 2.1.9. állitás szerint van egyszeres tagja, és az (A), (B), (C) feltevések rá is teljesülnek. Igy G helyett a csomó tagjainak H halmazára bizonyitható be a lineáris függőség, de ez a tulajdonság $H \subset G$ miatt G-re is teljesül. Q.u.e.d.

2.1.6. <u>Definició:</u> A DT valamely részhalmazára értelmezzük a T1–T5 tulajdonságokat, ezek:

T1. A halmaz nem üres.

T2. A halmaz a DT minden sorából és oszlopából – ha onnan tartalmaz elemet – pontosan két elemet tartalmaz.

T3. Nincs T1, T2-tulajdonságú valódi részhalmaza.

T4. A halmaz elemei báziselemek.

T5. A halmaz elemei nem-fiktivek.

Megjegyzés: A T4-tulajdonság pontosabban azt jelenti, hogy a halmaz elemeit egy kitüntetett bázisvektorrendszerből veszi. Tárgyalásunk során ez a kitüntetett rendszer az aktuális bázisvektorrendszer lesz; vagyis az a bázis, amely az éppen érvényes bázismegoldáshoz tartozik. – A továbbiakban ennek a bázisvektorrendszernek a jele: B.

Az 1.4.2. definició és a 2.1.6. definició alapján mondható:

2.1.10. <u>Állitás:</u> A DT egy részhalmaza pontosan akkor hurok, ha T1, T2, T3, T4 tulajdonságú.

Szükségünk lesz a következő – egyébként könnyen belátható állitásokra:

2.1.11. Állitás: Ha F T1-tulajdonságú halmaz, és minden olyan sorból és oszlopból, amelyből tartalmaz elemet, legalább két elemet tartalmaz, akkor annak tetszőleges $\underline{a}_1$ eleméhez van legalább egy

$$\underline{a}_1, \underline{a}_2, \underline{a}_3, \ldots \in F,$$

$$\varphi(\underline{a}_1, \underline{a}_2) = 1$$

tulajdonságú végtelen lánc és legalább egy

$$\underline{a}_1, \hat{\underline{a}}_2, \hat{\underline{a}}_3, \ldots \in F,$$

$$\varphi(\hat{\underline{a}}_1, \hat{\underline{a}}_2) = -1$$

tulajdonságú végtelen lánc.

2.1.12. Állitás: Ha F T1, T2 tulajdonságú halmaz, akkor annak tetszőleges $\underline{a}_1$ eleméhez van pontosan egy

$$\underline{a}_1, \underline{a}_2, \underline{a}_3, \ldots \in F,$$

$$\varphi(\underline{a}_1, \underline{a}_2) = 1$$

tulajdonságú végtelen lánc és pontosan egy

$$\underline{a}_1, \hat{\underline{a}}_2, \hat{\underline{a}}_3, \ldots \in F,$$

$$\varphi(\hat{\underline{a}}_1, \hat{\underline{a}}_2) = -1$$

tulajdonságú végtelen lánc.

2.1.5. Lemma: A 2.1.11. állitás feltételeinek eleget tevő T4 tulajdonságú halmaz T5 tulajdonságú is.

Bizonyitás: Legyen F a feltételeknek eleget tevő T4 tulajdonságú halmaz. Legyen $\underline{a}_1 \in F$ fiktiv. Vegyük az

$$\underline{a}_1, \underline{a}_2, \underline{a}_3, \ldots \in F$$

végtelen láncot. Feltehető, hogy $\underline{a}_2$ nem fiktiv, hiszen különben vehetünk egy másik

$$\underline{a}_1, \hat{\underline{a}}_2, \hat{\underline{a}}_3, \ldots \in F$$

láncot, amelyre $\varphi(\underline{a}_1, \hat{\underline{a}}_2) = -(\underline{a}_1, \underline{a}_2)$, és akkor $\hat{\underline{a}}_2$ biztosan nem fiktiv.

Megmutatjuk, hogy a lánc elemeiből származtatható a 2.1.4. lemma feltételeinek eleget tevő másik lánc. Ha valamely $s > 2$ esetén $\underline{a}_s$ fiktiv és $\underline{a}_i$ $(i = 2,3,\ldots,s-1)$ nem fiktiv, akkor az

$$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_s$$

lánc eleget tesz a 2.1.4. lemma feltételeinek. Más esetben legyen $\underline{a}_j$ a lánc hurokpontja. Ekkor az

$$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_j$$

sorozat vagy kör - de úgy ismét az előbbi eset áll elő; vagy ha nem kör, úgy véve az általa generált csomót, megint a 2.1.4.' lemma feltételeinek eleget tevő láncot kaptunk. Az elmondottakból következik, az F halmaz lineárisan függő; ami viszont ellentmond a T4 tulajdonságnak. Qu. e. d.

A 2.1.5. lemma speciális esetei:

2.1.13. <u>Állitás:</u> Minden T1, T2, T4 tulajdonságú halmaz egyben T5 tulajdonságú is.

2.1.14. <u>Állitás:</u> A huroknak nem lehetnek fiktiv elemei.

2.1.6. <u>Lemma:</u> Ha $\underline{b}_1, \underline{b}_2, \ldots,$ láncnak a hurokpontja az utolsó tag: $\underline{b}_j$; és a lánc egy T1, T2 tulajdonságú F halmazból veszi tagjait, akkor a/ a lánc kör, azaz $\underline{b}_j = \underline{b}_1$;

b/ $\varphi(\underline{b}_1, \underline{b}_2) \neq \varphi(\underline{b}_{j-1}, \underline{b}_j)$;

c/ a lánc tagjainak F_1 ($\subset F$) halmaza T1, T2, T3 tulajdonságú.

Bizonyitás: Ha a $\underline{b}_j$ hurokpont, akkor van olyan $(1 \leq) i (< j)$, amelyre $\underline{b}_j = \underline{b}_i$. Ha $i > 1$, akkor a 2.1.7. állitás második része szerint a DT-nek van olyan sora vagy oszlopa, amely a lánc három tagját tartalmazza. A 2.1.7. állitás első része szerint a lánc e három tagja a DT három különböző eleme. Az $i > 1$ feltevés tehát ellentmond annak, hogy a lánc tagjait egy T2 tulajdonságú halmazból veszi. Igy $\underline{b}_j = \underline{b}_1$, azaz a lánc kör.

Ha $\varphi(\underline{b}_1, \underline{b}_2) = \varphi(\underline{b}_{j-1}, \underline{b}_j)$, akkor az F halmaz T2 tulajdonsága és a már bebizonyitott a/ pont szerint $\underline{b}_2 = \underline{b}_{j-1}$, ami ellentmond a 2.1.4 állitás a/ részének, igy b/ bizonyitást nyert.

A 2.1.3. állitás szerint a lánc tagjainak F_1 halmaza nem üres (T1 tulajdonságú). A már bebizonyitott a/, b/ pontok és a 2.1.3 lemma alapján az F_1 a DT minden sorából és oszlopából – amelyből tartalmaz elemet – legalább két elemet tartalmaz, – az F T2 tulajdonsága szerint pedig legfeljebb két elemet tartalmaz. Tehát az F_1 T_2 tulajdonságú.

Legyen $F_2 \subset F_1$ T1, T2 tulajdonságú, és legyen $\underline{b}_i \in F_2$. Vegyük a $\{\underline{b}_k\}$ sorozat (2.1.2) szerinti átindexelését. – Ekkor $\underline{a}_1 (= \underline{b}_i) \in F_2$. Tegyük fel, hogy egy i $(1 \leq i < j)$ indexig fennáll

$$\underline{a}_1, \underline{a}_2, \ldots, a_i \in F_2 .$$

Ekkor az F_1 halmaz T2 tulajdonsága szerint az $\underline{a}_i$ sorában vagy oszlopában az $\underline{a}_i$-n kivül csak az $\underline{a}_{i+1}$ elem lesz F_1-ből.

Igy $F_2 \subset F_1$ miatt az F_2 csak úgy lehet T2 tulajdonságú, ha $\underline{a}_{i+1} \in F_2$. Tehát $F_1 \subset F_2$, azaz $F_1 = F_2$, azaz az F_1 halmaz T3 tulajdonságú. Qu. e. d.

2.1.7. Lemma: Legyen F T1, T2 tulajdonságú halmaz, akkor F-re teljesülnek a következő állitások:

a/ Az F halmaz tetszőleges $\underline{a}_1$ eleméhez tartozik pontosan egy $\{\underline{a}_k\}$: $\underline{a}_1$-ből sorfázisban induló kör és pontosan egy $\{\hat{\underline{a}}_k\}$: $\underline{a}_1$-ből oszlopfázisban induló kör, amelyre $\underline{a}_k \in F$, illetve $\hat{\underline{a}}_k \in F$ $(k = 1,2, \dots,)$. Az $\{\hat{\underline{a}}_k\}$ kör az $\{\underline{a}_k\}$ körnek a 2.1.6 állitás szerinti megforditása

b/ Legyen $\underline{a}_1 \in F$ tetszőleges elem és legyen az $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_j \in F$ sorozat egy kör. Az F halmaz pontosan akkor T3 tulajdonságú, ha

$$F = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{j-1}\}$$

c/ Az F halmaz tetszőleges $\underline{a}_1$ eleméhez pontosan egy olyan F_1 $(\subset F)$ T1, T2, T3 tulajdonságú halmaz létezik, amelyre $\underline{a}_1 \in F_1$.

d/ Az F halmaz bármely két $F_1 \neq F_2$ T1, T2, T3 tulajdonságú részhalmazára fennáll:

$F_1 \cap F_2 = \emptyset$, valamint $\underline{a} \in F_1$, $\underline{b} \in F \setminus F_1$ esetén $\varphi(a,b) = 0$.

e/ Az F halmaz pontosan egyféleképpen állitható elő T1, T2, T3 tulajdonságú részhalmazainak egyesitéseként. (Feltéve, hogy minden részhalmazt csak egyszer szerepeltetünk az unióban és a sorrendtől eltekintünk.)

Bizonyitás: Az $\{\underline{a}_k\}$ és az $\{\hat{\underline{a}}_k\}$ körök egzisztenciája és unicitása a 2.1.12. állitás és a 2.1.6. lemma a/ pontjából adódik. A 2.1.6. lemma b/ pontja szerint az $\{\underline{a}_k\}$ körnek a 2.1.6. állitásban szereplő megforditása oszlopfázisban indul $\underline{a}_1$-ből, igy ismét a 2.1.12. állitás szerint nem különbözhet az $\{\hat{\underline{a}}_k\}$ körtől. Ezzel az a/ pontot bebizonyitottuk.

A b/ pont a 2.1.4. állitás a/ pontjából, a 2.1.6. lemma c/ pontjából és a T3 tulajdonság definiciójából egyenesen adódik.

A c/ pont szerint az F_1 halmaz létezése az itteni a/ pontból és a 2.1.6. lemma c/ pontjából következik. Ha lenne még egy $F_2 \neq F_1$, $F_2 \subset F$ T1, T2, T3 tulajdonságú halmaz, amelyre $\underline{a}_1 \in F_2$, akkor az a/ pont szerint az $\underline{a}_1$-hez az F_1-ben, illetve az F_2-ben létezik egy $\underline{a}_1$-ből induló $\{\underline{a}_k\}$ kör, illetve egy $\{\hat{\underline{a}}_k\}$ kör. A b /pont szerint az F_1 éppen az $\{\underline{a}_k\}$ kör tagjainak halmaza, az F_2 éppen az $\{\hat{\underline{a}}_k\}$ kör tagjainak halmaza. $F_1 \neq F_2$ miatt a két kör különbözik, és a két kör nem lehet egymásnak a 2.1.6 állitásban szereplő megforditása sem. Ez ellentmond a már bebizonyitott a/ pontnak.

A d/ pont a c/-ből egyenesen következik - az F halmaz T2 tulajdonságát ismételten figyelembe véve.

Az e/ pont szintén a c/-ből következik: Abból, hogy minden $\underline{a}_i \in F$ elemhez létezik egy F_i T1, T2, T3 tulajdonságú halmaz, amelyre $\underline{a}_i \in F_i$; adódik, hogy F előáll az elemeihez tartozó T1, T2, T3 tulajdonságú részhalmazainak egyesitéseként. Most vegyük az F két

$$F = \bigcup_i F_i \qquad \text{és} \qquad F = \bigcup_i G_i$$

előállitását (F_i, G_i T1, T2, T3- tulajdonságú halmazok).

Minden F_i-hez vegyünk egy $\underline{a}_i$ ($\in F_i$) elemet. Az $\underline{a}_i$-hez létezik olyan $\underline{G}_j$ amelyre $\underline{a}_i \in G_j$. A c/ pont szerint $F_i = G_j$. Igy az $\{F_i\}$ halmazrendszer minden eleme előfordul a $\{G_i\}$ halmazrendszerben, de ennek a megforditása ugyanigy megmutatható. Tehát a két halmazrendszer azonos. Qu. e. d.

A 2.1.10. állitásból és a 2.1.7. lemma a/ és b/ pontjaiból adódik:

2.1.15. Állitás: Ha F hurok és $\underline{a}_1 \in F$, akkor van pontosan egy olyan

$$\underline{a}_1, \underline{a}_2, \dots, \underline{a}_j \in F, \qquad \varphi(a_1, \underline{a}_2) = 1 \qquad (\varphi(a_1, \underline{a}_2) = -1)$$

tulajdonságú kör, amelyre:

$$F = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{j-1}\} .$$

2.1.8. Lemma: Ha az $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_j, \dots$ láncnak az $\underline{a}_j$ hurokpontja, és $\underline{a}_j = \underline{a}_j = \underline{a}_i$, $1 < i < j$. valamint az

$$(2.1.12) \qquad \underline{a}_i, \underline{a}_{i+1}, \dots, \underline{a}_{j-1}, \underline{a}_j$$

tagok egy T_1, T_2-tulajdonságú halmaz elemei, akkor a lánc által generált csomó (2.1.5) alakú.

Bizonyitás: A 2.1.2. állitás szerint az

$$\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{j-1}$$

és az

$$\underline{a}_{i-1}, \underline{a}_{i-2}, \dots, \underline{a}_1$$

sorozatok külön-külön láncot alkotnak.

Megmutatjuk, hogy a két lánc (2.1.5) alakú összefüzése is lánc, azaz

(2.1.13)

$$0 \neq \varphi(\underline{a}_{j-2}, \underline{a}_{j-1}) = -\varphi(\underline{a}_{j-1}, \underline{a}_{i-1})$$

és ha még $i > 2$, akkor

(2.1.14)

$$-\varphi(\underline{a}_{j-1}, \underline{a}_{i-1}) = \varphi(\underline{a}_{i-1}, \underline{a}_{i-2}) \neq 0$$

A lánc definiciója alapján nyilvánvaló:

$$-\varphi(\underline{a}_{i-1}, \underline{a}_{i}) = \varphi(\underline{a}_{i}, \underline{a}_{i+1}) \neq 0.$$

A (2.1.12) láncra a 2.1.6. lemma b/ része alapján adódik, hogy az olyan kör, amelyben:

(2.1.15) $$\varphi(\underline{a}_{j-1}, a_j) \neq \varphi(\underline{a}_{i}, \underline{a}_{i+1}).$$

A φ értékek itt sem zérusok, mert az argumentumok lépések, igy (2.1.15)-ből adódik:

(2.1.16) $$\varphi(\underline{a}_{j-1}, \underline{a}_{j}) = \varphi(\underline{a}_{i-1}, \underline{a}_{i}).$$

Ebből pedig a 2.1.1. b/ állítás szerint:

(2.1.17) $$\varphi(\underline{a}_{j-1}, \underline{a}_{i-1}) = \varphi(\underline{a}_{j-1}, \underline{a}_{j}).$$

A lánc definiciója szerint:

(2.1.18) $$\varphi(\underline{a}_{j-2}, \underline{a}_{j-1}) = -\varphi(\underline{a}_{j-1}, \underline{a}_{j}).$$

A (2.1.17) és a (2.1.18) alapján (2.1.13) egyenesen következik.

A lánc definiciója alapján $i > 2$ esetén

(2.1.19) $$\varphi(\underline{a}_{i}, \underline{a}_{i-1}) = -\varphi(\underline{a}_{i-1}, \underline{a}_{i-2}),$$

a (2.1.16)-ból pedig megint a 2.1.1. b/ állitás szerint:

$$(2.1.20) \qquad \varphi(\underline{a}_{j-1}, \underline{a}_{i-1}) = \varphi(\underline{a}_{i-1}, \underline{a}_{i}) .$$

A (2.1.19)-ből és a (2.1.20)-ból már (2.1.14) egyenesen adódik. Qu. e.d.

2.1.9. Lemma: Legyen H olyan T1 tulajdonságú halmaz, amely a DT minden sorából és oszlopából - ha onnan tartalmaz elemet - legalább két elemet tartalmaz. Teljesülnek a következő állitások:

a/ Ha H elemei nem fiktivek(azaz H T5 tulajdonságú) és H legalább egy sorból vagy oszlopból kettőnél több elemet tartalmaz, úgy a H vektorrendszer lineárisan függő.

b/ Ha H T4 tulajdonságú halmaz, akkor egyben T2 tulajdonságú is.

Bizonyitás: Azoknak a soroknak, illetve oszlopoknak a száma, amelyekből a H elemeit veszi legyen p, illetve q. Ezen sorok közül a k.-ból ($k=1,2,\ldots,p$) tartalmazon α_k számú elemet; ezen oszlopok közül a k.-ból ($k=1,2,\ldots,q$) tartalmazzon β_k elemet. A feltételek szerint:

$$\alpha_k \geqq 2 \qquad (k=1,2,\ldots,p),$$

$$\beta_k \geqq 2 \qquad (k=1,2,\ldots,q).$$

Igy H elemeinek száma:

$$\frac{\sum_{k=1}^{p}\alpha_k + \sum_{k=1}^{q}\beta_k}{2} \geqq \frac{2p+2q}{2} = p+q$$

Ebből látszik, ha csak egyetlen esetben határozottan $\alpha_k > 2$ vagy $\beta_k > 2$, akkor a H halmaz elemeinek száma $p+q+1$. Igy az a/ állitás esetében teljesülnek a 2.1.1. lemma feltételei: H lineárisan függő.

A b/ állitás esetében a 2.1.5. lemma szerint a H minden eleme nem-fiktiv. Igy annak feltételezése, hogy H nem T2 tulajdonságú, az a/ állitás szerint ellentmondana a H T4 tulajdonságának. Qu. e.d.

A 2.1.3. lemmából a 2.1.9. lemmából és a 2.1.6. lemma c/részéből egyenesen adódik:

2.1.16. <u>Állitás:</u> Ha valamely - a 2.1.2. lemma feltételeinek eleget tevő - kör egy T4 tulajdonságú halmazból veszi tagjait, akkor a kör tagjainak halmaza T1, T2, T3, T4 tulajdonságú, azaz hurok.

2.1.10. <u>Lemma:</u> Legyenek F_1 és F_2 T1, T2, T5 tulajdonságú halmazok, továbbá a

(2.1.21) $\underline{b}_1, \underline{b}_2, \dots, \underline{b}_s$

sorozat olyan kör, amelyre $\underline{b}_1 \in F_1$, $\underline{b}_s \in F_2$. Legyen F_3 a (2.1.21) tagjainak halmaza; erre teljesüljön: $F_3 \setminus F_1 \neq \emptyset$ és F_3 T5 tulajdonságú. Ekkor a $H = F_1 \cup F_2 \cup F_3$ halmaz lineárisan függő.

<u>Bizonyitás:</u> A (2.1.21) láncban legyen $\underline{b}_i$ az első olyan tag, amelyre $\underline{b}_i \notin F_1$ (ilyen tag $F_3 \setminus F_1 \neq \emptyset$ miatt létezik). Mivel a (2.1.21) sorozat lánc, igy $\varphi(\underline{b}_{i-1}, \underline{b}_i) \neq 0$. Igy $\varphi(\underline{b}_{i-1}, \underline{b}_i) = 1$ esetén a $\underline{b}_i$ sorában ($\varphi(\underline{b}_{i-1}, \underline{b}_i) = -1$ esetén a $\underline{b}_i$ oszlopában) három különböző tagja lesz a H halmaznak. Ugyanis $\underline{b}_{i-1} \in F_1$ és F_1 T2 tulajdonsága miatt $\underline{b}_{i-1}$ sorában is oszlopában is van még $\underline{b}_{i-1}$-től különböző F_1-beli elem. Ugyanakkor a (2.1.21) sorozat lánc voltából adódik az is, hogy az F_3 minden elemének sorában és oszlopában legalább két különböző eleme van H-nak. Ezt az F_1 és F_2 T-tulajdonságával

összevetve a H-nak legalább két eleme van a DT minden olyan sorában és oszlopában, amelyben van eleme. Ehhez véve a $\underline{b}_i$-re tett megállapitásunkat, a 2.1.9. lemma a/ része szerint a H halmaz lineárisan függő. Qu. e. d.

2.2. AZ S1-S7 ALGORITMUS TULAJDONSÁGAI

Az 1. részben nem is egy olyan algoritmust irtunk le, amely több - egymást meghatározott rendben követő - rekurziós szabályból áll úgy, hogy a szabályok nem csak azt irják elő, bemenetükből milyen kimenet származik; hanem azt is, melyik más szabály követi őket. Ezért az ilyen algoritmusok matematikailag úgy is leirhatók, mint egy speciális algebrai automata speciális müködése. Erre nevezetes példa az S1-S7 eljárás.

2.2.1. <u>Definició:</u> Legyen B a DT egy bázisvektorrendszere; m a DT nem-fiktiv sorainak, n a DT nem-fiktiv oszlopainak száma; M pedig

$$M = \{0, 1, 2, \ldots, m, m+1, \ldots, m+n\}$$

alakú halmaz. Ekkor a (P, Q, δ, λ) együttest <u>sajátelem automatának</u> nevezzük, ahol P az automata állapotainak

$$P = \{START, S1, S2, \ldots, S7, STOP\}$$

alakú halmaza; Q az összes

$$f: B \longrightarrow M$$

alakú függvény halmaza; továbbá δ, λ

olyan

$$\delta: (P \times Q) \longrightarrow Q,$$

$$\lambda: (P \times Q) \longrightarrow P,$$

függvények, amelyek a következő szabályokkal határozhatók meg $(f, g \in Q)$:

START:

$$\delta(\mathrm{START}, f) = g: \quad \underline{b}\,(\in B) \longrightarrow g(\underline{b}) = 0\,;$$

$$\lambda(\mathrm{START}, f) = S1.$$

S1:

$$\delta(S1, f) = g: \quad \underline{b} \longrightarrow g(\underline{b}) =$$

$$= \begin{cases} i, & \text{ha} \quad \underline{b} = \underline{a}_{i,n+1} \in B, \\ f(\underline{b}) & \text{egyébként.} \end{cases}$$

$$\lambda(S1, f) = S2.$$

S2: $\delta(S2, f) = g: \underline{b} \longrightarrow g(\underline{b}) =$

$$= \begin{cases} m+j, & \text{ha} \quad \underline{b} = \underline{a}_{m+1,j} \in B; \\ f(\underline{b}) & \text{egyébként.} \end{cases}$$

$$\lambda(S2, f) = S3.$$

S3:

$$\delta(S3, f) = g: \underline{b} \longrightarrow g(\underline{b}) =$$

$$= \begin{cases} m+j, & \text{ha} \quad \underline{b} = \underline{a}_{i,j} \in B, \;\; f(\underline{b}) = 0 \text{ és} \\ & \qquad \exists\, \hat{\underline{b}} \in B: \;\; \varphi(\underline{b}, \hat{\underline{b}}) = 1, \;\; f(\hat{\underline{b}}) = i; \\ f(\underline{b}) & \text{egyébként.} \end{cases}$$

$$\lambda(S3, f) = S4.$$

S4:

$$\delta(S4, f) = g: \underline{b} \longrightarrow g(\underline{b}) =$$

$$= \begin{cases} i, \;\text{ha} & \underline{b} = \underline{a}_{i,j} \in B, \;\; f(\underline{b}) = 0 \text{ és } \exists\, \hat{\underline{b}} \in B: \\ & \qquad \varphi(\underline{b}, \hat{\underline{b}}) = -1, \;\; f(\hat{\underline{b}}) = m+j\,; \\ f(\underline{b}) & \text{egyébként.} \end{cases}$$

$$\lambda(S4,f) = \begin{cases} S3, & \text{ha } \exists \underline{b} \in B \text{ és } \hat{\underline{b}} \in B: \\ & \varphi(\underline{b},\hat{\underline{b}}) = 1, \quad 0 = g(\underline{b}) < g(\hat{\underline{b}}) \leqq m; \\ S5 & \text{egyébként.} \end{cases}$$

S5:

$$\delta(S5,f) = g : \underline{b} \longrightarrow g(\underline{b}) =$$

$$= \begin{cases} i, & \text{ha } \underline{b} = \underline{a}_{i,j} \in B, \ f(\underline{b}) = 0 \text{ és } \forall \hat{\underline{b}} \in B, \\ & \varphi(\underline{b},\hat{\underline{b}}) = 1 \text{ esetén: } f(\hat{\underline{b}}) > m; \\ f(b) & \text{egyébként.} \end{cases}$$

$$\lambda(S5, f) = S6.$$

S6:

$$\delta(S6, f) = g : \underline{b} \longrightarrow g(\underline{b}) =$$

$$= \begin{cases} m+j, & \text{ha } \underline{b} = \underline{a}_{i,j} \in B, \ f(\underline{b}) = 0 \text{ és } \forall \hat{\underline{b}} \in B, \\ & \varphi(\underline{b},\underline{b}) = -1 \text{ esetén: } 0 < f(\hat{\underline{b}}) \leqq m; \\ f(\underline{b}) & \text{egyébként.} \end{cases}$$

$$\lambda(S6,f) = \begin{cases} S5, & \text{ha } \exists \underline{b} \in B: \ g(\underline{b}) = 0 \text{ és } \forall \hat{\underline{b}} \in B, \\ & \varphi(\underline{b},\underline{b}) = 1 \text{ esetén: } g(\hat{b}) > m; \\ S7 & \text{egyébként.} \end{cases}$$

S7.

$$\delta(S7,f) = g : \underline{b} \longrightarrow g(\underline{b}) =$$

$$= \begin{cases} i_0, & \text{ha } \underline{b} = \underline{a}_{i_0,j_0} \in B; \\ & i_0 = \min\{ i : \underline{a}_{i,j} \in B, \quad f(\underline{a}_{i,j}) = 0 \}, \\ & j_0 = \min\{ j : \underline{a}_{i_0,j} \in B, \quad f(\underline{a}_{i_0,j}) = 0 \}; \\ f(\underline{b}) & \text{egyébként.} \end{cases}$$

$$\lambda(S7, f) = \begin{cases} S3, & \text{ha } \exists \underline{b} \in B: \ f(\underline{b}) = 0; \\ STOP & \text{egyébként.} \end{cases}$$

2.2.2. Definició: A (P, Q, δ, λ) sajátelem automata rekurziv müködésén a

$$(p_1, f_1), \ (p_2, f_2), \ldots (p_s, f_s) \in (P \times Q)$$

sorozatot értjük, ahol

$$p_1 = START;$$

$$f_1 = \delta(START) = \delta(START, f_0),$$

$$f_0 \in B \quad \text{tetszőleges;}$$

$$p_i = \lambda(p_{i-1}, f_{i-2});$$

$$f_i = \delta(p_i, f_{i-1});$$

$$p_{s+1} = STOP.$$

2.2.3. Definició: Ha a DT egy adott B bázisa esetén a

$$(p_1, f_1), (p_2; f_2), \ldots, (p_s, f_s) \in (P \times Q)$$

sorozat a (P, Q, δ, λ) sajátelem automata rekurziv müködése, akkor a $g = f_s : B \longrightarrow M$ függvényt sajátelem kijelölésnek (felcimkézésnek) nevezzük.

2.2.4. Definició: Ha a DT egy adott B bázisa esetén a $g: B \longrightarrow M$ függvény sajátelem kijelölés és $\underline{b} \in B$, akkor $g(\underline{b}) \leq m$ esetén $\underline{b}$-t sor-sajátelemnek (SS-elemnek), $g(\underline{b}) > m$ esetén $\underline{b}$-t oszlop-sajátelemnek (OS-elemnek) nevezzük. Ha $g(\underline{b}) = i \leq m$, akkor azt mondjuk: $\underline{b}$ az i. sor sajátleme; illetve $g(\underline{b}) = m + j > m$ esetén azt mondjuk: $\underline{b}$ a j. oszlop sajátleme.

A 2.2.1.-2.2.4. definiciókban nem tettünk egyebet, mint az 1.3. fejezet S1-S7 algoritmusát formalizáltuk. - A 2.2.1. definicióban a szabályok ugyanolyan Sk jelölést kaptak, mint az 1.3. fejezetben.

(A 1.3. fejezetben az S7 szabály nem volt egyértelmü, az $\underline{a}_{i_0,j_0}$ önkényes megválasztására adott lehetőséget; itt ezt is egyértelmüvé tettük. Azonban $\underline{a}_{i_0,j_0}$-t másképpen is választhattuk volna. - Ezt azzal fogjuk demonstrálni, hogy a további bizonyitásokban sehol nem fogjuk kihasználni az itt megadott szabályt.)

A (P, Q, δ, λ) automata δ és λ függvényeinek tulajdonságaiból következik, hogy a rekurziv müködésében $f_k(\underline{b}) \geq f_{k-1}(\underline{b})$ ($\underline{b} \in B$, $k > 1$) ; és valahányszor még létezik olyan $\underline{b} \in B$, hogy $f_k(\underline{b}) = 0$, mindannyiszor lesz olyan $k' > k$, hogy $f_{k'}(\underline{b}) > 0$. A B halmaz véges volta alapján igy mondható:

2.2.1. <u>Állitás:</u> A sajátelem kijelölés a DT tetszőleges B bázisvektorrendszere esetén létezik; és a $g: B \to M$ sajátelem kijelölésre fennáll: $g(\underline{b}) > 0$ ($\underline{b} \in B$), azaz valamennyi báziselem kijelölődik.

Egyrészt, mert a g egyértékü függvény, másrészt a 2.2.4. definicióra és a δ függvény tulajdonságaira hivatkozva, mondható:

2.2.2. <u>Állitás:</u> Egy sajátelem kijelölés szerint minden báziselemre igaz: az vagy csak SS-elem vagy csak OS-elem. Ha $\underline{b}$ az i. sor (j. oszlop) sajáteleme, akkor $\underline{b}$ a DT i. sorában (j. oszlopában) helyezkedik el.

Mivel a sajátelem automata rekurziv müködésében a függvény tulajdonságai miatt fennáll: ha $f_r(\underline{b}) > 0$, akkor $r' > r$ esetén $f_{r'}(\underline{b}) = f_r(\underline{b})$, ezért:

2.2.3. <u>Állitás:</u> minden $\underline{b} \in B$ esetén csak egyetlen olyan ($\underline{b}$-től függő) r index van amelyre $f_{r-1}(\underline{b}) = 0$ és $f_r(\underline{b}) > 0$.

2.2.5. Definició: Legyen ϱ olyan

$$\varrho : B \longrightarrow \{2, 3, \ldots, s\}$$

függvény, amelyre:

$$\underline{b}\ (\in B) \longrightarrow r:\ f_{r-1}(\underline{b}) = 0 \text{ és } f_r(\underline{b}) > 0.$$

Ekkor a $\varrho(\underline{b}) = r$ számot a $\underline{b}$ <u>kijelöléséhez tartozó indexnek</u> nevezzük. Ha $\underline{b}, \hat{\underline{b}} \in B$ és $\varrho(\underline{b}) < \varrho(\hat{\underline{b}})$, akkor azt mondjuk, $\underline{b}$-t <u>korábban jelöltük sajátelemnek,</u> mint a $\hat{\underline{b}}$-t ($\hat{\underline{b}}$-t <u>később jelöltük sajátelemnek</u> mint a $\underline{b}$-t).

2.2.6. Definició: Ha a sajátelem automata rekurziv rekurziv müködésének $(p_{\varrho(\underline{b})}, f_{\varrho(\underline{b})})$ tagjában $p_{\varrho(\underline{b})} = Sk$, akkor azt mondjuk: a $\underline{b} \in B$ vektort az <u>Sk szabály jelölte</u> (vagy <u>az Sk szerint jelöltük ki</u>) sajátelemnek.

Az S1 szabály alkalmazása előtt minden $\underline{b} \in B$ báziselem jelöletlen ($f_1(\underline{b}) = 0$, ha $\underline{b} \in B$). Ugyanakkor a DT minden nem-fiktiv sorában (oszlopában) legfeljebb egy F0-báziselem (FS-báziselem) van, amit az S1 szabály (S2 szabály) SS-elemnek (OS-elemnek) jelöl ki, majd utána az S3 szabály (S4 szabály) szerint minden további báziselem a sorban (oszlopban) OS-elem (SS-elem lesz). Igy mondható:

2.2.4. Állitás: Azokban a nem-fiktiv sorokban (oszlopokban), amelyekben az S1 szabály (S2 szabály) szerint jelöltünk SS-elemet (OS-elemet), csak egy SS-elem (OS-elem) lesz a sajátelem kijelölés szerint.

Az S5 szabály (S6 szabály) alkalmazása előtt a megfelelő sorban (oszlopban) még nem volt SS-elem (OS-elem), és az S5 szabály (S6 szabály) a sor (oszlop) báziselemeit kimeriti, tehát:

2.2.5. Állitás: Az S5 szabály (S6 szabály) szerint kijelölt SS-elem (OS-elem) sorában (oszlopában) csak egy SS-elem (OS-elem) van a sajátelem kijelölés szerint.

Az S7 szabály alkalmazása előtt a megfelelő sorban még nem volt SS-elem, utána pedig az S3 szabály szerint a sor maradék báziselemeit OS-elemnek jelöljük, tehát:

2.2.6. Állitás: Az S7 szabály szerint kijelölt SS-elem sorában nincs másik SS-elem a sajátelem kijelölés szerint.

2.2.1. Lemma: Ha az S4 szabály (S3 szabály) valamely alkalmazását az S7 szabály nem előzte meg, akkor a S4 pontban (S3 pontban) kijelölt SS-elem (OS-elem) sorában (oszlopában) nincs másik SS-elem (OS-elem) a sajátelem kijelölés szerint.

Bizonyitás: Megmutatjuk, hogy az egy sorban két SS-elem feltételezése ellentmond a 2.1.4. lemmának (Az állitás OS-elemre vonatkozó részének bizonyitása érdemben nem különbözik az itt következőktől.)

Legyen (i_0, j_1) a lemma feltételeinek megfelelő SS-elem, és tegyük fel, hogy a sorában van még egy SS-elem: (i_0, j_{-1}). A 2.2.4. állitás, 2.2.5. állitás és a 2.2.6. állitás alapján adódik, hogy az (i_0, j_{-1})-et is csak az S4 szabály jelölhette ki. Annak feltételezése, hogy az (i_0, j_{-1}) jelölését az S7 pont alkalmazása megelőzte volna (azaz létezne r: $p_r = S7$, $r < \varrho(\underline{a}_{i_0, j_{-1}})$, azonnal ellentmondáshoz vezet. Ekkor ugyanis (i_0, j_{-1}) jelölése csak az (i_0, j_1) jelölése után történhetett $(\varrho(\underline{a}_{i_0, j_{-1}}) > \varrho(\underline{a}_{i_0, j_1}))$.

Igy

$$f_{\varrho(\underline{a}_{i_0,j_1})}(\underline{a}_{i_0,j_{-1}}) = 0$$

és akkor

$$p_{\varrho(\underline{a}_{i_0,j_1})+1} = \lambda(S4, f_{\varrho(\underline{a}_{i_0,j_1})-1}) = S3,$$

azaz az S3 szabály az (i_0,j_{-1})-et OS-elemnek jelöli, mielőtt az S7 szabályt érinthettük volna.

Most

$$p_{\varrho(\underline{a}_{i_0,j_1})} = S4$$

és a δ függvény S4 állapotbeli tulajdonságai miatt létezik az (i_0, j_1) oszlopában egy nála korábban kijelölt OS-elem; egy ilyent jelöljünk (i_2,j_1)-nek (tehát $\varrho(\underline{a}_{i_2,j_1}) < \varrho(\underline{a}_{i_0,j_1})$).

Fennáll:

$$(2.2.1) \qquad p_{\varrho(\underline{a}_{i_2,j_1})} = S2$$

vagy

$$(2.2.2) \qquad p_{\varrho(\underline{a}_{i_2,j_1})} = S3,$$

azaz (i_2, j_1) kijelölése csak az S2 vagy az S3 szabályok szerint történhetett. A (2.2.1) esetben megállunk; különben továbblépünk: veszünk egy olyan (i_2,j_3) SS-elemet, amelyet az i_2. sorban még az (i_2,j_1)-nél korábban kijelöltünk sajátelemnek. (Ez (2.2.2) és a δ függvény S3-beli tulajdonságai miatt létezik.) Az (i_2, j_3) jelölése csak az S1 vagy az S4 szabályok szerint lehetett; - most akkor lépünk tovább, ha (i_2,j_3)-at az S4 szerint jelöltük ki, különben megállunk. Igy haladva, felváltva SS és OS-elemeken, egy

(2.2.3)

$$\underline{a}_{i_0,j_1},\ \underline{a}_{i_2,j_2},\dots,\ \underline{a}_{i_{s-mod_2(s)},\ j_{s+mod_2(s)-1}}$$

láncot kapunk, amelynek utolsó tagját vagy az S1 vagy az S2 szabály szerint jelöltük sajátelemnek; igy a lánc vége fiktiv elem; a lánc többi tagja nem fiktiv (Itt $mod_2(s)$ az s/2 osztás maradéka: 0 vagy 1) ; s: a lánc tagjainak száma.) A lánc végessége abból adódik, hogy tagjai mind különbözőek (v.ö. a tagok választását a 2.2.3. állitással és a 2.2.5. definicióval), tehát a lánc elemeinek száma nem nagyobb, mint a báziselemek száma.

Hasonlóan kijelölhető egy lánc az $\underline{a}_{i_0,j_{-1}}$-ből kiindulva:

(2.2.4)

$$\underline{a}_{i_0,j_{-1}},\underline{a}_{i_2,j_{-1}},\dots,\underline{a}_{i_{-r+mod_2(r)},j_{-r-mod_2(r)+1}}.$$

Erre is elmondhatók ugyanazok, mint a (2.2.3) láncra.

Vegyük a (2.2.3) lánc megforditását (tagjait visszafelé soroljuk fel) és ezt folytassuk a (2.2.4) lánccal. A két lánc ilyen összefüsésével kapott sorozat is lánc, ami egyrészt a 2.1.2 állitás a/ részéből és abból adódik, hogy

$$\varphi(\underline{a}_{i_{-2},j_{-1}},\ \underline{a}_{i_0,j_{-1}}) = -1,$$

$$\varphi(\underline{a}_{i_0,j_{-1}},\ \underline{a}_{i_0,j_1}) = 1,$$

$$\varphi(\underline{a}_{i_0,j_1},\ \underline{a}_{i_2,j_1}) = -1.$$

A képzés módjából kapjuk, hogy ez a lánc eleget tesz a 2.1.4. lemma feltételeinek. Tehát e lánc tagjainak G halmaza lineárisan függő vektorrendszer. Ez azonban ellentmond annak, hogy a G elemei bázisvektorok ($G \subset B$). Qu.e.d.

2.2.7. <u>Definició:</u> Egy B bázis esetén tekintsük a sajátelem automata rekurziv müködését és legyen

$$H_r = \left\{ \underline{b} : \underline{b} \in B, \ f_{r-1}(\underline{b}) = 0 \right\}.$$

A $G(\subset B)$ halmazt <u>T6 tulajdonságúnak</u> mondjuk, ha van olyan r, hogy

$$G = H_r \ , \quad p_r = S7 \quad \text{és} \quad f_{r-1} \neq f_r \ .$$

Figyelembe véve, hogy az S7 szabály első alkalmazása előtt az S1, S2 szabályok már minden fiktiv báziselemet kijelöltek sajátelemnek, a definicióból következik:

2.2.7. <u>Állitás:</u> Minden T6 tulajdonságú halmaz egyben T1, T4, T5 tulajdonságú is.

A sajátelem automata λ függvényének tulajdonságaiból következik, hogy az S7 szabály első alkalmazása előtt, illetve az S7 szabály két különböző alkalmazása között az algoritmus mindig érinti az S3, S4, S5, S6 szabályokat. Az S3, S4 szabályok ciklikus alkalmazása után az SS-elemek sorában (OS-elemek oszlopában) minden báziselem ki lesz jelölve OS-elemnek (SS-elemnek). Utána az S5, S6 pontok ciklikus alkalmazása nem változtat ezen a helyzeten. Igy mondható:

2.2.8. <u>Állitás:</u> A DT azon soraiban (oszlopaiban), amelyekből a $G = H_r$ T6 tulajdonságú halmaz az elemeit veszi, az f_{r-1} függvény még nem jelöl SS-elemet (OS-elemet).

Az S5, S6 szabályok ciklikus alkalmazása után minden olyan sorban (oszlopban), amelyben még nincs SS-elem (OS-elem), de jelöletlen báziselemet tartalmaz a ciklus végén, legalább két jelöletlen báziselemnek kell lenni. Tehát:

2.2.9. <u>Állitás:</u> Egy T6 tulajdonságú halmaznak legalább két eleme van a DT minden olyan sorában és oszlopában, amelyben legalább egy eleme van.

Összevetve a 2.1.9. lemmát, a 2.2.7. állitást és a 2.2.9 állitást közvetlenül adódik:

2.2.10. <u>Állitás:</u> A T6 tulajdonságú halmaz egyben T2 tulajdonságú is.

2.2.8. <u>Definició:</u> Egy adott B bázis és (azon) egy sajátelem automata rekurziv müködése esetében jelölje G^{*} a következő halmazt;

$$G^{*} = H_r, \; p_r = S7, \; f_{r-1} \neq f_r$$

és minden $r' < r$ esetén $p_{r'} \neq S7$.

A G^{*} halmaz nyilván T6 tulajdonságú, másrészt a 2.2.7, 2.2.8 definiciók és az $f_k(\underline{b}) \geqq f_{k-1}(\underline{b})$ $(\underline{b} \in B, \; k > 1)$ tulajdonság alapján mondható:

2.2.11. <u>Állitás:</u> Ha $r < r'$, akkor

$$H_{r'} \subset H_r .$$

Továbbá, ha $\underline{b} \in B$ esetén $\rho(\underline{b}) = r$, akkor $\underline{b} \in H_r$ és $\underline{b} \notin H_{r'}$.
Speciálisan: Ha F T6 tulajdonságú halmaz, akkor $F \subset G^{*}$; továbbá minden $\underline{b} \in B$, $p_{\rho(\underline{b})} = S7$ esetén $\underline{b} \in G^{*}$.

A 2.2.11, a 2.2.7. és a 2.2.10. állitásokból, valamint a 2.1.7. lemma c/ részéből adódik:

2.2.12. Állitás: Ha $\underline{b} \in B$ és $p_{\varrho(\underline{b})} = S7$, akkor pontosan egy olyan F hurok létezik, amelyre: $F \subset H_{\varrho(\underline{b})}$ és $\underline{b} \in F$. Speciálisan: pontosan egy olyan F hurok létezik, amelyre: $F \subset G^{*}$ és $\underline{b} \in F$.

2.2.2. Lemma: Legyen H_r T6 tulajdonságú halmaz, $p_r = S7$; valamint legyen $\underline{a}_1 \in H_r$: $\varrho(\underline{a}_1) = r$ és F az $\underline{a}_1$-hez tartozó $\underline{a}_1 \in F$, $F \subset H_r$ tulajdonságokkal rendelkező hurok. Ekkor fennállnak:

a/ Vagy $F = H_r$ és $k > r$ esetén H_k nem T6 tulajdonságú; vagy $F = H_r \setminus H_{r'}$, ahol $r < r'$, $H_{r'}$ T6 tulajdonságú, de minden $r < k < r'$ esetén $p_k \neq S7$.

b/ Az $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_s \in F$, $\varphi(\underline{a}_1, \underline{a}_2) = 1$ tulajdonságú körre teljesül: páratlan (páros) indexü tagjai SS-elemek (OS-elemek) lesznek a sajátelem kijelölésben.

c/ Az F-hez tartozó minden sorban (oszlopban) pontosan egy SS-elem (OS-elem) lesz a sajátelem kijelölés szerint.

Bizonyitás: $\varrho(\underline{a}_1) = r$ esetén - $p_r = S7$ miatt - a 2.2.12. állitás szerint pontosan egy $F \subset H_r$ hurok létezik, amelyre: $\underline{a}_1 \in F$. Vegyük a 2.1.15. állitás szerint létező, egyetlen

$$(2.2.5) \qquad \begin{gathered} \underline{a}_1, \underline{a}_2, \dots, \underline{a}_s \in F, \\ \varphi(\underline{a}_1, \underline{a}_2) = 1, \\ F = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{s-1}\} \end{gathered}$$

tulajdonságú kört.

(Megjegyzés: A lánc, illetve a kör tulajdonságai miatt s páratlan.)

Mivel $p_r = S7$, igy $\underline{a}_1$ SS-elem lesz. A 2.2.8. állítás szerint a H_r-hez tartozó sorokban, illetve oszlopokban az f_{r-1} szerint még nincs jelölve SS-elem, illetve OS-elem. Igy a $p_{r+1} = S3$ szabály pontosan csak az $\underline{a}_2$ elemet jelöli ki OS-elemnek. (Csak az $\underline{a}_1$ sorában jelölhet sajátelemet és csak az $\underline{a}_2$-t a H_r T2-tulajdonsága miatt.) Most tegyük fel, hogy $1 \leq k < (s-1)/2$ esetén teljesülnek:

A(k) / $\quad H_r \setminus H_{r+2k} = \{\underline{a}_1, \dots, \underline{a}_{2k-1}, \underline{a}_{2k}\} \subset F$;

B(k) / Az $\{\underline{a}_1, \dots, \underline{a}_{2k-1}, \underline{a}_{2k}\}$ lánc páratlan (páros) indexü tagjai SS-elemek (OS-elemek).

C(k) / A lánchoz tartozó SS-elemek (OS-elemek) sorában (oszlopában) pontosan egy SS-elem (OS-elem) van az f_{r+2k-1} függvény szerint.

D(k) / Az f_{r+2k-1} függvény a H_{r+2k}-hoz tartozó sorokban még nem jelöl ki SS-elemet, a H_{r+2k}-hoz tartozó oszlopokban pedig - az $\underline{a}_{2k}$ oszlopát kivéve - még nem jelöl OS-elemet.
A láncban az utolsó lépésre: $\varphi(\underline{a}_{2k-1}, \underline{a}_{2k}) = 1$.

Az A/- D/ pontok $k = 1$ esetén nyilvánvalóan teljesülnek. $k+1$-re áttérve, a következőket mondhatjuk: $2k < s-1$ miatt vehetjük a (2.2.5) lánc következő

$$\underline{a}_{2(k+1)-1} \; (= \underline{a}_{2k+1})$$

elemét. Az A/ tulajdonság szerint erre még $f_{r+2k-1}(\underline{a}_{2k+1}) = 0$; a D(k) tulajdonság szerint, és mert a kör lánc: $-\varphi(\underline{a}_{2k}, \underline{a}_{2k+1}) = -1$. Igy - mert $\underline{a}_{2k}$ oszlopában már van OS-elem - az S4. szabály az $\underline{a}_{2k+1}$ tagot SS-elemnek jelöli ki. Ezután, hasonló indokok miatt, az $\underline{a}_{2(k+1)}$ tagot az S3 szabály OS-elemnek jelöli ki. Azt, hogy a f_{r+2k+1} függvény szerint csak ez a két újonnan kijelölt elem lesz az f_{r+2k-1} függvényhez képest, a $D(k)$ tulajdonság és a H_r T2 tulajdonsága biztositja. Tehát k+1 esetén teljesül:

$$A(k+1)/ \quad H_r \setminus H_{r+2(k+1)} = \{\underline{a}_1, \dots, \underline{a}_{2k+1}, \underline{a}_{2(k+1)}\} \subset F.$$

A $B(k+1)$ tulajdonság is fennáll. A $C(k+1)$ tulajdonság a $D(k)$ tulajdonságból következik. Mivel a (2.2.5) lánc kör és H_r T2 tulajdonságú, ezért $2(k+1) < s-1$ esetén a $D(k)$-ból $D(k+1)$ is következik. $2(k+1) = s-1$ esetén viszont az $\underline{a}_{2(k+1)}$ oszlopában is minden báziselemet kijelöl sajátelemmé a f_{r+2k+1} függvény.

Mármost $F = H_r \setminus H_{r+s-1}$, azaz $F \subset H_r$ és $H_{r+s-1} \subset H_r$ miatt:

$$(2.2.6) \qquad H_{r+s-1} = H_r \setminus F.$$

Ha $H_{r+s-1} = \emptyset$, akkor $F = H_r$.

$H_{r+s-1} \neq \emptyset$ esetén akár a $D(k)$ tulajdonságból az indukciós lépés során, akár a (2.2.6)-ból a 2.1.7. lemma d/ része szerint adódik, hogy az f_{r+s-2} függvény a H_{r+s-1}-hez tartozó sorokban (oszlopokban) még nem jelöl SS-elemet (OS-elemet). - Tehát az S7 szabály ujabb alkalmazását megelőzően az S3, S4 szabályok az f_{r+s-2} függvényen nem változtatnak. A H_r T2 tulajdonsága, a (2.2.6) és a 2.2.7. lemma e/ pontja szerint H_{r+s-1} is T2 tulajdonságú. - Tehát az S5, S6 szabályok sem változtatják meg az f_{r+s-2} függvényt az S7 szabály újabb alkalmazása előtt.
Igy $H_{r+s-1} = H_{r'}$, ahol $p_{r'} = S7$ (nem feltétlenül: $r' = r+s-1$): azaz $H_{r+s-1} \neq \emptyset$ esetén H_{r+s-1} T6 tulajdonságú halmaz.

Ezzel a lemma a/ részét bebizonyitottuk; a b/ része pedig a $B(k)$ tulajdonságból következik. A lemma c/ része abból adódik, hogy a H_r T2 tulajdonsága miatt a 2.2.7 lemma d/ része szerint

$\underline{a} \in F$, $\underline{b} \in H_{r+s-1}$ esetén $\varphi(\underline{a}, \underline{b}) = 0$; tehát az F-hez tartozó sorokban és oszlopokban a g sajátelem kijelölés is csak azokat a sajátelemeket jelöli ki, mint az f_{r+s-2} függvény. Qu.e.d.

2.2.3. Lemma: Ha adott a DT valamely bázisa, akkor azon a sajátelem kijelölés a DT minden nem-fiktiv sorában (oszlopában) pontosan egy SS-elemet (OS-elemet) jelöl ki.

Bizonyitás: A 2.2.4. és 2.2.5 állitások, valamint a 2.2.1. lemma biztositják, hogy az S7 szabály első alkalmazását megelőzően jelölt SS-elemek sorában legfeljebb egy SS-elem lesz (OS-elemek oszlopában legfeljebb egy OS-elem lesz). A 2.2.2. lemma c/ részének és a 2.1.7. lemma e/ részének összevetéséből adódik: ha az S7 pont kijelölt sajátelemet, akkor a G^* halmazhoz tartozó sorokban (oszlopokban) pontosan egy SS-elem (OS-elem) lesz a sajátelem kijelölés szerint. - Ezek szintéziseként kapjuk: A DT minden nem-fiktiv sorában (oszlopában) legfeljebb egy SS-elem (OS-elem) lesz a sajátelem kijelölés szerint.

Igy, ha p az SS-elemek száma, q az OS-elemek száma, akkor:

(2.2.7) $\quad p \leqq m, \quad q \leqq n.$

(Emlékeztetünk rá, hogy a fiktiv sorban (oszlopban) nincs SS-elem (OS-elem).) A 2.2.1. állitásból következik:

(2.2.8) $\quad p + q = m + n,$

ahol m:a nem-fiktiv sorok száma; n : a nem-fiktiv oszlopok száma. A (2.2.7)-ből és (2.2.8)-ból:

(2.2.9) $\quad p = m, \quad q = n.$

Igy az, hogy minden nem-fiktiv sorban (oszlopban) legfeljebb egy SS-elem (OS-elem) lehet a (2.2.9)-cel összevetve azt is jelenti, hogy minden nem-fiktiv sorban (oszlopban) legalább egy SS-elemnek (OS-elemnek) kell lenni. Qu.e.d.

A 2.1.16 és a 2.1.14. állitásokat figyelembe véve, teljes indukcióval - lépésenként a 2.2.3. lemmát felhasználva - megmutatható:

2.2.13. Állitás: Ha valamely - a 2.1.2. lemma feltételeinek eleget tevő - kör egy T4 tulajdonságú halmazból veszi elemeit, és SS-elemből (OS-elemből) sorfázisban (oszlopfázisban) indul, akkor a páratlan indexű tagjai SS-elemek (OS-elemek), a páros indexü tagjai OS-elemek (SS-elemek). - Feltettük, hogy az első tag indexe: 1.

A 2.2.1.- 2.2.5. definiciókból adódik:

2.2.14. Állitás: Ha valamely sajátelem kijelölés esetén $\underline{a}$, $\underline{b} \in B$ és $\underline{a}$: SS-elem, $\underline{b}$: OS-elem, akkor

$$\varrho(\underline{a}) \neq \varrho(\underline{b}) .$$

2.2.4. Lemma: Minden T1, T2, T4 tulajdonságú F halmaz esetében fennáll:

a/ ha $\underline{a} \in F$ és minden $\hat{\underline{a}} \in F$ esetén

$$\varrho(\underline{a}) \leq \varrho(\hat{\underline{a}}), \text{ akkor } p_{\varrho(\underline{a})} = S7;$$

b/ $F \subset G^{*}$.

Bizonyitás: Megmutatjuk, hogy a lemma feltételeinek eleget tevő $\underline{a} \in F$ elem nem lehet olyan SS-elem, amit az S1, S4 vagy S5 szabály jelölt volna sajátelemnek. - Hasonlóan mutatható meg, hogy $\underline{a}$ nem lehet olyan OS-elem sem, amit az S2, S3, S6 szabályok valamelyike jelölt volna sajátelemnek.

A 2.1.7. lemma a/ része szerint az $\underline{a}$-hoz létezik egy

$$(\underline{a} =)\ \underline{a}_1, \underline{a}_2, \dots, \underline{a}_s (= \underline{a}) \in F,$$

$$\varphi(\underline{a}_1, \underline{a}_2) = 1$$

tulajdonságú kör. A 2.1.6. lemma b/ része szerint és F T4 tulajdonsága miatt erre a körre alkalmazható a 2.2.13. állítás, amit a lánc illetve a kör definíciójával összevetve, kapjuk: $\underline{a}_2$ és $\underline{a}_{s-1}$ az $\underline{a}(=\underline{a}_1=\underline{a}_s)$ sorában és oszlopában levő OS-elemek. Az $\underline{a}$ választása és a 2.2.14. állítás szerint:

$$\varrho(\underline{a}_2) > \varrho(\underline{a}), \tag{2.2.10}$$

$$\varrho(\underline{a}_{s-1}) > \varrho(\underline{a}). \tag{2.2.11}$$

Az $\underline{a}$ elemet az S1 szabály azért nem jelölhette ki sajátelemmé, mert a 2.1.13. állítás szerint $\underline{a}$ nem-fiktiv DT-elem. Az S4 szabály azért nem jelölhette ki az $\underline{a}$ elemet, mert az a δ függvény S4-beli tulajdonságai szerint ellentmondana a (2.2.11) egyenlőtlenségnek; az $\underline{a}$ elem S5 szabály szerinti kijelölése pedig a (2.2.10) egyenlőtlenséggel ellenkezik.

Az itt elmondottakból és a 2.2.1. állitásból következik, hogy az $\underline{a}$ elemet szükségképpen az S7 szabály jelölte ki sajátelemmé: $p_{\varrho(\underline{a})} = S7$.

Mármost, ha $\underline{a} \in F$ olyan elem, amelyre:

$$\varrho(\underline{a}) = \min\{\varrho(\hat{\underline{a}}) : \hat{\underline{a}} \in F\}$$

akkor $F \subset H_{\varrho(\underline{a})}$, $p_{\varrho(\underline{a})} = S7$, $f_{r-1}(\underline{a}) \neq f_r(\underline{a})$;

azaz F részhalmaza egy T6 tulajdonságú halmaznak; azaz a 2.2.11. állitás szerint $F \subset G^*$. Qu.e.d.

2.2.5. <u>Lemma:</u> A DT egy adott B bázisa esetén pontosan annyi hurok létezik a DT-ben, ahány $\underline{a} \in B$-re fennáll $p_{\varrho(\underline{a})} = S7$; továbbá, ha F hurok, $\underline{a} \in F$ és $\underline{a}$ SS-elem, akkor az F összes elemét, és csak az F elemeit felsorolhatjuk a következőképpen:

(A) $\underline{a}_1 = \underline{a}$;

(B) $\underline{a}_{2k}$: OS-elem és $\varphi(\underline{a}_{2k-1}, \underline{a}_{2k}) = -1$;

(C) $\underline{a}_{2k+1}$: OS-elem és $\varphi(\underline{a}_{2k}, \underline{a}_{2k+1}) = 1$,

ha $\varphi(\underline{a}_{2k}, \underline{a}_1) \neq 1$;

(D) $\underline{a}_{2k}$ a sorozat vége, ha $\varphi(\underline{a}_{2k}, \underline{a}_{2k+1}) = 1$.

Bizonyitás: A G^* halmaz T6 tulajdonsága – ennek következtében T1, T2, T4 tulajdonsága (2.2.7 és 2.2.10 állitások) – miatt a 2.1.7. lemma e/ része szerint pontosan egy olyan

(2.2.12) $\{F_1, F_2, \dots, F_q\}$

rendszere létezik az T1, T2, T3, T4 tulajdonságú F_k halmazoknak – tehát hurkoknak – amelyre:

$$\bigcup_{k=1}^{q} F_k = G^*.$$

Igy minden $F \subset G^*$ tulajdonságú F hurokhoz van olyan F_k a (2.2.12)-ben, amelyre $F = F_k$. – A 2.2.4. lemma b/ része szerint pedig minden F hurokra: $F \subset G^*$. – Tehát a G^* halmaz (amely definciója szerint pontosan akkor létezik, ha van $\underline{a} \in B$: $p_{\varsigma(a)} = S7$) részhalmazait képező hurkokon kivül más hurok nem létezik a DT-ben.

Mármost a 2.2.12. állitás szerint minden $\underline{a} \in B$, $p_{\varsigma(\underline{a})} = S7$ esetén pontosan egy olyan F hurok létezik, amelyre: $\underline{a} \in F$. A 2.2.2 lemma a/ részéből következik: ha $\underline{a}, \hat{\underline{a}} \in B$ $\underline{a} \neq \hat{\underline{a}}$ és $p_{\varsigma(\underline{a})} = S7$, $p_{\varsigma(\hat{a})} = S7$ esetén $\underline{a} \in F$, $\hat{\underline{a}} \in \hat{F}$ és F, $\hat{F}$ hurkok, akkor $F \neq \hat{F}$. Mindezeket összevetve: az $\underline{a} \in B$, $p_{\varsigma(\underline{a})} = S7$ tulajdonságú elemek száma nem több mint a hurkok száma.

A 2.2.4. lemma a/ része szerint minden F hurokhoz van olyan $\underline{a} \in B$ elem amelyre: $p_{\varrho(\underline{a})} = S7$. Másrészt, ha $F \neq \hat{F}$ hurkok és $\underline{a} \in F$, $\hat{\underline{a}} \in \hat{F}$, $p_{\varrho(\underline{a})} = p_{\varrho(\hat{\underline{a}})} = S7$, akkor $\underline{a} \neq \hat{\underline{a}}$. – Ugyanis $F \subset G^{*}$, $\hat{F} \subset G^{*}$ és $F \neq \hat{F}$ miatt a 2.1.7. lemma d/ része szerint $F \cap \hat{F} = \emptyset$. Kaptuk: a $F(\subset B)$ hurkok száma nem több mint az $\underline{a} \in B$, $p_{\varrho(\underline{a})} = S7$ tulajdonságú elemek száma. – A lemma első részét bebizonyitottuk.

A 2.1.15. állitás szerint van egy olyan

$$\underline{a}_1 (= \underline{a}),\ \underline{a}_2, \dots, \underline{a}_s (= \underline{a})$$

kör, amelyre $\varphi(\underline{a}_1, \underline{a}_2) = -1$ és

$$F = \{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{s-1}\}$$

A 2.1.17 lemma szerint ez éppen a 2.2.2. lemma b/ pontjában szereplő kör megforditása vagy a megforditás (2.1.2) szerinti átindexelése. A 2.2.2. lemma b/ részéből következik, hogy erre az $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_{s-1}$ sorozatra teljesülnek az (A) – (D) feltételek. Tehát az (A) –(D) pontokkal definiált algoritmussal az F halmaz elemeit fel lehet sorolni. Azt pedig, hogy ez az algoritmus mindig csak az F halmaz felsorolását eredményezi a következők garantálják: A sorozat első eleme adott. Minden nem-fiktiv sorban (oszlopban) pontosan egy SS-elem (OS-elem) van, amit a (B) és (C) pontokkal összevetve kapjuk: a sorozat minden tagja egyértelmüen meghatározza a következő tagot (ha létezik következő.) Ezeket a (D) ponttal összevetve, adódik, a sorozat utolsó tagja egyértelmüen meghatározott. Q.u.e.d.

Megjegyzés: Az, hogy az (A)–(B) algoritmust SS-elemmel inditjuk, lényegtelen megkötés, ami csak az itteni (A)–(B) pontok és az 1.4. fejezetbeli HU1–HU3 hurok-algoritmus egyszerübb megfogalmazását szolgálja. Viszont, miután első tagnak SS-elemet választottunk már lényeges, hogy abból oszlopfázisban lépünk tovább; mert csak oszlopra igaz, hogy abban csak egy OS-elem van,- egy sorban több OS-elem is lehet.

Ezzel a disztribuciós eljárásnak az 1.4. fejezetig leirt - korábbi megoldásokhoz képest új - elemeit bebizonyitottuk.

2. 3. A SOR- ÉS OSZLPÚTRA ÉS A KOMPENZÁCIÓS EGYÜTTHATÓKRA VONATKOZÓ TÉTELEK

Az 1. részben az algoritmus leirásánál még mellőzni kivántuk olyan - csak az elméleti bizonyitás szempontjából szükséges - fogalmak bevezetését, mint: sor-, oszlopfázis; lánc; kör; stb. Ezért az 1.5. fejezetben még sorozat által generált csomóról, a 2.1. fejezetben már lánc által generált csomóról beszéltünk. A két fogalom kapcsolatát itt világitjuk meg.

2.3.1. Lemma: Legyen $\underline{a}_j$ a DT-beli

(2.3.1) $\underline{b}_0, \underline{b}_1, \underline{b}_2, \dots, \underline{b}_j, \dots$

lánc hurokpontja, $\underline{b}_j = \underline{b}_i$, $0 < i < j$; valamint a lánc tagjaira teljesüljön még a $k = 1, 2, \dots, j$ esetekben:

(A) ha $\varphi(\underline{b}_{k-1}, \underline{b}_k) = 1$, akkor $\underline{b}_k$ SS-elem;

(B) ha $\varphi(\underline{b}_{k-1}, \underline{b}_k) = -1$, akkor $\underline{b}_k$ OS-elem.

Fennállnak:

a/ a következő

(2.3.2) $\underline{b}_i, \underline{b}_{i+1}, \dots, \underline{b}_j$

sorozat kör, és a tagjainak halmaza hurok;

b/ a (2.3.1) lánc által az 1.5.2. és a 2.1.5. definiciók szerint generált csomók azonosak.

Bizonyitás: A (2.3.2) sorozat a 2.1.2. állitás b/ része szerint lánc, de mert a $\underline{b}_j$ hurokpont és $\underline{b}_j = \underline{b}_i$, ezért a (2.3.2) egyben kör is.

Mivel $\underline{b}_i(=\underline{b}_j)$ vagy csak SS-elem vagy csak OS-elem, ezért az (A), (B) tulajdonságok alapján:

(2.3.3) $$\varphi(\underline{b}_{i-1}, \underline{b}_i) = \varphi(\underline{b}_{j-1}, \underline{b}_j).$$

A (2.3.1) sorozat lánc, ezért:

(2.3.4) $$\varphi(\underline{b}_{i-1}, \underline{b}_i) \neq \varphi(\underline{b}_i, \underline{b}_{i+1}).$$

Megjegyezzük, hogy $\underline{b}_{i-1}$ létezik, mert $i > 0$; a $\underline{b}_{i+1}$ létezését pedig a 2.1.5. állitás bizonyitja.

A (2.3.3) és (2.3.4) alapján:

$$\varphi(\underline{b}_i, \underline{b}_{i+1}) \neq \varphi(\underline{b}_{j-1}, \underline{b}_j),$$

azaz a (2.3.2) kör kielégiti a 2.1.2. lemma feltételeit, és - mert tagjai báziselemek - igy kielégiti a 2.1.16. állitás feltételeit is, azaz a kör tagjainak halmaza hurok. Ebből a 2.1.8. lemma szerint következik, hogy a (2.3.1) lánc esetében az 1.5.2. definició és a 2.1.5. definició ekvivalens. Q.u.e.d.

Az UT1-UT3 pontok alapján (1.5.fejezet) mondható:

2.3.1. <u>Állitás:</u> Az UT1-UT3 pontok által előállitott sorozatnak csak az utolsó tagja lehet fiktiv báziselem. - A sorozat hurokpontja nem lehet fiktiv. - A sorozat mindig véges: vagy fiktiv báziselemben vagy hurokpontban fejeződik be; legalább két tagja van.

2.3.2. Állitás Az UT1-UT3 pontok szerint előállitott sorozat SS-elembe mindig sorfázisban lép, OS-elembe mindig oszlopfázisban lép. Ha kettőnél több tagja van, akkor felváltva lép SS- és OS-elembe.

Ebből a 2.1.3. definició alapján adódik:

2.3.3. Állitás: Az UT1-UT3 pontok egy véges láncot állitanak elő.

Ebből a 2.1.5. állitás szerint adódik.

2.3.4. Állitás: Ha az UT1-UT3 pontok által előállitott sorozatnak $\underline{b}_j$ hurokpontja és $\underline{b}_j = \underline{b}_i$ $j > i$, akkor $j = i + 4$.

Mivel az UT1-UT3 pontok által előállitott sorozatban $\underline{b}_0$ nem báziselem, a többi tag viszont az, igy mondható:

2.3.5. Állitás: Ha az UT1-UT3 pontok által előállitott sorozatnak $\underline{b}_j$ hurokpontja és $\underline{b}_j = \underline{b}_i$, akkor $i > 0$.

A 2.3.1, 2.3.4. és 2.3.5 állitásokat összefoglalva mondható:

2.3.6. Állitás: Ha $\underline{b}_0$ nem FS-elem (nem FO-elem) és nem tartozik az adott bázishoz, akkor a $\underline{b}_0$-ból induló sorút (oszlpút) mindig létezik, - legalább két tagja van.

Az UT1. pont, a 2.3.1. állitás és a 2.3.3. állitás összevetéséből adódik.

2.3.7. Állitás: Az UT1-UT3 pontok által előállitott sorozatban minden szomszédos tagpár egy nem-fiktiv sorhoz vagy egy nem-fiktiv oszlophoz tartozó lépést alkot.

A 2.3.2. állitás, a 2.3.7. állitás és a 2.2.3. lemma alapján:

2.3.8. Állitás: A sorút (oszlopút) definiciója szerinti UT1-UT3 pontokban előállitott sorozatra $i = 1, 2, \ldots, j$ esetén fennáll: a $\underline{b}_{i-1}$ tag egyértelmüen meghatározza a $\underline{b}_i$ tagot.

A továbbiakban feltesszük, hogy $\underline{b}_0$ olyan DT-elem, amely nem tartozik az adott bázishoz, valamint ha sorútról (oszlopútról) van szó akkor $\underline{b}_0$ nem FS-elem (nem OS-elem).

A 2.3.4. állitásból következik:

2.3.9. Állitás: Ha a sorút (oszlopút) egy csomó, akkor első két tagja azonos az UT1-UT3 pontokban előállitott (és a csomót generáló) sorozat első két tagjával.

Igy mindig teljesül:

2.3.10. Állitás: A sorút (oszlopút) a $\underline{b}_0$-ból sorfázisban (oszlopfázisban) indul.

A 2.3.2. - 2.3.4. állitások, a 2.3.1. lemma és az 1.5.2. definició összevetéséből adódik:

2.3.11. Állitás: Ha a sorút (oszlopút) csomó, akkor az egyben egy véges lánc is.

Ezt a 2.3.3. állitással összevetve:

2.3.12. Állitás: A sorút (oszlopút) egy véges láncot jelent.

A 2.3.1, 2.3.2 és a 2.3.4. állitások, az UT1 és UT4 pontok és az 1.5.2. definició szerint:

2.3.13. Állitás: A sorút (oszlopút) vagy fiktiv báziselemben fejeződik be, - ilyenkor nincs két azonos tagja, és csak ilyenkor tartalmaz fiktiv báziselemet, és ilyenkor is csak ezt az egyet; vagy csomó, azaz sorfázisban (oszlopfázisban) visszatér a $\underline{b}_0$-hoz.

A 2.3.6. és a 2.3.8. állitások az UT4 pont és az 1.5.2. definició alapján mondható;

2.3.13. Állitás: Adott $\underline{b}_0$ elemhez pontosan egy sorút (oszlopút) létezik az adott sajátelem kijelölés mellett.

2.3.2. Lemma: Legyen adott a DT egy bázisa és rajta egy sajátelem kijelölés. Legyen $\underline{b}_0$ a DT olyan vektora, amely nem tartozik az adott bázishoz. Jelölje s a sorút, q az oszlopút

$\underline{b}_0$-on kivüli tagjainak számát; $\underline{b}_k$ a sorút, $\hat{\underline{b}}_k$ az oszlopút k. tagját; c_k legyen $\underline{b}_k$-hoz tartozó, $\hat{c}_k$ pedig a $\hat{\underline{b}}_k$-hoz tartozó – az E1–E4 pontok (1.5.fejezet) szerint számitott – kompenzációs együttható. Ekkor:

a/ $\underline{b}_0 = \sum\limits_{k=1}^{s} c_k \cdot \underline{b}_k$, ha $\underline{b}_0$ FO-elem;

b/ $\underline{b}_0 = \sum\limits_{k=1}^{q} \hat{c}_k \cdot \hat{\underline{b}}_k$, ha $\underline{b}_0$ FS-elem;

c/ $\underline{b}_0 = \sum\limits_{k=1}^{s} c_k \cdot \underline{b}_k + \sum\limits_{k=1}^{r} \hat{c}_k \cdot \hat{\underline{b}}_k$, ha $\underline{b}_0$ nem fiktiv elem.

Bizonyitás: Legyen H az a/ esetben a sorút, b/ esetben az oszlopút, c/ esetben a sor- és az oszlopút tagjainak halmaza (beleértve a $\underline{b}_0$ elemet is). Csak azt fogjuk részletesen megmutatni, hogy a H halmaz lineárisan függő.

Ha $\underline{b}_0$ nem-fiktiv és a hozzátartozó sorút is, oszlopút is csomó, akkor a 2.1.10. lemma alapján , minden más esetben pedig a 2.1.4. lemma alapján mutatható meg, hogy a H vektorrendszer lineárisan függő.

Ha $\underline{b}_0$ nem-fiktiv és a hozzátartozó sorút az UT1–UT3 pontok szerint előállitott $\underline{b}_0$, $\underline{b}_1, \dots, \underline{b}_j$ sorozat által (generált csomó; a hozzátartozó oszlopút pedig a $\underline{b}_0$, $\hat{\underline{b}}_1, \dots, \hat{\underline{b}}_j$ (UT1–UT3) sorozat által generált csomó akkor $\underline{b}_j = \underline{b}_i$ $0 < i < j$ és $\hat{\underline{b}}_{\hat{j}} = \hat{\underline{b}}_{\hat{i}}$, $0 < \hat{i} < \hat{j}$. Vegyük a $\underline{b}_i, \underline{b}_{i+1}, \dots, \underline{b}_j$ sorozatot, illetve a $\hat{\underline{b}}_{\hat{i}}$, $\hat{\underline{b}}_{\hat{i}+1}, \dots, \hat{\underline{b}}_{\hat{j}}$ sorozatot; a tagjaik halmaza legyen F_1, illetve F_2. A 2.3.2. állitás és a 2.3.1. lemma a/ része szerint F_1 és F_2 hurok. Ugyanakkor $\varphi(\underline{b}_0, \underline{b}_1) \neq \varphi(\underline{b}_0, \hat{\underline{b}}_1)$ miatt a

$$\underline{b}_{i-1}, \underline{b}_{i-2}, \ldots, \underline{b}_1, \underline{b}_0, \hat{\underline{b}}_1, \ldots, \hat{\underline{b}}_{\hat{i}-1}$$

sorozat lánc. Ezt a láncot a (2.1.21) láncnak megfeleltetve az F_1 és F_2 halmazokkal teljesülnek a 2.1.10. lemma feltételei; és mivel az itteni H halmaz éppen azonos a 2.1.10 lemma szerinti H halmazzal, igy H lineárisan függő vektorrendszer.

Legyen a $\underline{b}_0$ nem-fiktiv, és a hozzátartozó mindkét út fiktiv báziselemben fejeződjék be. Ekkor vegyük az egyik út megforditását, és ezt folytassuk a másik úttal. A két út ilyen összefüzésével kapott sorozat olyan lánc lesz, amelyre már alkalmazható a 2.1.4. lemma.

Legyen a $\underline{b}_0$ nem-fiktiv, és a hozzátartozó utak közül pontosan egyik fiktiv báziselemben fejeződjék be. Ekkor vegyük a fiktiv báziselemben befejeződő út megforditását, azt folytassuk a másik úttal, ezt ismét folytassuk az első úttal (most már nem a forditottjával). Ezekkel az összefüzésekkel megint a 2.1.4. lemmának eleget tevő láncot kapunk.

Legyen $\underline{b}_0$ fiktiv. Ekkor a hozzátartozó sorútra (oszlopútra) közvetlenül alkalmazható a 2.1.4. lemma.

Amit eddig bizonyitottunk: a H halmaz lineárisan függő vektorrendszer, azaz a $\underline{b}_0$ vektor kifejezhető a H-hoz tartozó bázisvektorok lineáris kombinációjaként.

Azt, hogy a $\underline{b}_0$ éppen az 1.5. fejezet E1-E4 pontjaiban meghatározott c_k ($\hat{c}_k$) kompenzációs együtthatók segitségével és éppen e lemma a/, b/, c/ pontjainak megfelelő formában irható fel, a 2.1.4. lemma bizonyitásában alkalmazott indukciós módszerrel láthatjuk be. - Az (1.5.2) és az (1.5.3) formulák nevezőjének nem-zéró voltát a bázisvektorrendszer lineáris függetlensége garantálja. Qu. e.d.

Legyen

(2.3.5) $$\underline{b}_0, \underline{b}_1, \underline{b}_2, \dots, \underline{b}_s$$

a $\underline{b}_0$-hoz tartozó sorútnak a hurokpontot megelőző részsorozata (ha nincs hurokpont, akkor ez maga a sorút) ; legyen

(2.3.6) $$\underline{b}_0, \hat{\underline{b}}_1, \hat{\underline{b}}_2, \dots, \hat{\underline{b}}_q$$

az oszlopútnak a hurokpontot megelőző részsorozata.

Nyilvánvaló, hogy a (2.3.5) és a (2.3.6) sorozatok tagjait egyaránt az UT1-UT3 pontok jelölték ki; igy a 2.3.8. állitás szerint mondható:

2.3.14. <u>Állitás:</u> Ha a (2.3.5) és a (2.3.6) sorozat $\underline{b}_i$, illetve $\hat{\underline{b}}_j$ tagjára fennáll: $\underline{b}_i = \hat{\underline{b}}_j$, akkor

a/ $\underline{b}_{i+1} = \hat{\underline{b}}_{j+1}$, feltéve hogy $i < s$ és $j < q$;

b/ $s-i \leq q-j$ esetén: $\underline{b}_{i+1} = \hat{\underline{b}}_{j+1}$, $\underline{b}_{i+2} = \hat{\underline{b}}_{j+2}, \dots, \underline{b}_s = \hat{\underline{b}}_{j+s-1}$;

c/ $s-i \geq q-j$ esetén: $\underline{b}_{i+1} = \hat{\underline{b}}_{j+1}$, $\underline{b}_{i+2} = \hat{\underline{b}}_{j+2}, \dots, \underline{b}_{i+q-j} = \hat{\underline{b}}_q$;

d/ ha $\underline{b}_s$ ($\hat{\underline{b}}_q$) fiktiv, és $\underline{b}_s = \hat{\underline{b}}_k$ ($\hat{\underline{b}}_q = b_k$), akkor $\underline{b}_s = \hat{\underline{b}}_q$.

2.3.3. <u>Lemma:</u> A (2.3.5) és (2.3.6) sorozatokkal kapcsolatban fennállnak a következő állitások:

a/ Ha $\underline{b}_s = \hat{\underline{b}}_k$ ($\hat{\underline{b}}_q = \underline{b}_k$) és $\underline{b}_s$ ($\hat{\underline{b}}_q$) nem fiktiv, akkor $\hat{\underline{b}}_q$ ($\underline{b}_s$) sem fiktiv.

b/ Ha $\underline{b}_s$ ($\hat{\underline{b}}_q$) fiktiv és a két útnak van közös tagja, akkor $\underline{b}_s = \hat{\underline{b}}_q$.

c/ Ha $\underline{b}_i$ ($\hat{\underline{b}}_i$) az oszlopútnak is (sorútnak is) tagja akkor a

$$\underline{b}_{i+1}, \dots, \underline{b}_s \qquad (\hat{b}_{i+1}, \dots, \hat{\underline{b}}_q)$$

tagok mind tagjai az oszlopútnak is (sorútnak is.)

d/ Legyen $\underline{b}_i$ a (2.3.5) sorozatban az első olyan tag, amelyre $i>0$ és $\underline{b}_i$ az oszlopútnak is tagja; valamint $\hat{\underline{b}}_j$ a (2.3.6) sorozatban az első olyan tag, amelyre $j>0$ és $\hat{\underline{b}}_j$ a sorútnak is tagja; továbbá $\underline{b}_i \neq \hat{\underline{b}}_j$. Ekkor a két út közös tagjai hurkot alkotnak, amelyben

$$\underline{b}_i = \hat{\underline{b}}_k,\ \underline{b}_{i+1} = \hat{\underline{b}}_{k+1}, \ldots$$

$$\ldots \underline{b}_{i+q-k} = \hat{\underline{b}}_q,\ \underline{b}_{i+q-k+1} = \hat{\underline{b}}_j,$$

$$\underline{b}_{i+q-k+2} = \hat{\underline{b}}_{j+1}, \ldots, \underline{b}_s = \hat{\underline{b}}_{k-1}.$$

Bizonyitás: Az a/ bizonyitása: Legyen $\underline{b}_s = \hat{\underline{b}}_k$, $k \leq q$. Az állitás $k=q$ esetén nyilvánvaló. Ha $k<q$ és $\underline{b}_s$ nem fiktiv, akkor az UT2-UT3 pontok szerint következő $\underline{b}_{s+1}$ tag hurokpontja a sorút képzésénél a UT1-UT3 pontok szerint előálló sorozatnak. De úgy $k<q$ miatt $\underline{b}_{s+1} = \hat{\underline{b}}_{k+1} = \underline{b}_j$, ahol $s-j \geq 3$ (2.1.5. állitás). Most ha $q-k-1 \leq s-j$, akkor $\hat{\underline{b}}_q = \underline{b}_{j+q-k-1}$ lesz a 2.3.14. c/ állitás szerint, azaz (a 2.3.1 állitást is figyelembe véve) a/ teljesül. Egyébként a (2.3.6) sorozatnak a 2.3.14. b/ állitás szerint van többszörös tagja, ami ellentmond (2.3.6) értelmezésének.

b/ bizonyitása: $\underline{b}_s = \hat{\underline{b}}_k$ $k \leq q$ esetén 2.3.14. d/ szerint az állitás igaz. Ha $\underline{b}_s$ fiktiv és $\hat{\underline{b}}_q = \underline{b}_k$, $k \leq s$, akkor ezen lemma a/ pontjából következik, hogy $\hat{\underline{b}}_q$ fiktiv; a 2.3.14. d/ állitásból pedig következik, hogy $\hat{\underline{b}}_q = \underline{b}_s$.

c/ bizonyitása: (Csak a sorút esetét bizonyitjuk.) Legyen $\underline{b}_i = \hat{\underline{b}}_j$. Igy $s-i \leq q-j$ esetén a 2.3.14. b/ állitás szerint igaz a c/ állitás is.

Legyen $s-i>q-j$. Ekkor a 2.3.14. c/ állitás szerint

$$\underline{b}_{i+2}=\hat{\underline{b}}_{j+2},\dots,\underline{b}_{i+q-j}=\hat{\underline{b}}_q .$$

Igy $\hat{\underline{b}}_q$ nem lehet fiktiv, mert úgy

$$(2.3.7) \qquad \underline{b}_{i+q-j}=\hat{\underline{b}}_q=\underline{b}_s$$

lenne az itteni b/ pont alapján; amiből a (2.3.5) értelmezése szerint: $s-i=q-j$ következik. Tehát az UT2-UT3 szerint az oszlopút képzésénél következő $\hat{\underline{b}}_{q+1}$ tag hurokpont, azaz $\hat{\underline{b}}_{q+1}=\hat{\underline{b}}_k$, $0<k<q$. Ekkor $k<j$, mert különben megint (2.3.7) következne, ami ellentmond az $s-i>q-j$ feltételnek. Most tekintsük a

$$\underline{b}_{i+q-j},\ \underline{b}_{i+q-j+1},\dots,\underline{b}_s$$

és a

$$\hat{\underline{b}}_k,\ \hat{\underline{b}}_{k+1},\dots,\hat{\underline{b}}_{j-1}$$

sorozatokat. Ha $s-i-q+j\leq j-1-k$, akkor a 2.3.14. b/ állitás szerint az itteni c/ állitás megint teljesül; ellenkezőleg

$$\underline{b}_i=\hat{\underline{b}}_j=\underline{b}_{i+q-k} \qquad (q>k),$$

ami ellentmond a (2.3.5) sorozat értelmezésének.

d/ bizonyitása: A c/ állitás alapján (2.3.5) és a (2.3.6) sorozatok közös tagjai:

$$\underline{b}_i,\underline{b}_{i+1},\dots,\underline{b}_s;$$

illetve

$$\hat{\underline{b}}_j,\hat{\underline{b}}_{j+1},\dots,\hat{\underline{b}}_q .$$

Ekkor $\hat{\underline{b}}_j=\underline{b}_k$ esetén $k>i$, és mert $s-i=q-j$, nyilván

$$(2.3.8) \qquad s-k<q-j.$$

Igy a 2.3.14. b/ állitás alapján:

$$(2.3.9) \qquad \hat{\underline{b}}_{j+1} = \underline{b}_{k+1}, \ldots, \hat{\underline{b}}_{j+s-k} = \underline{b}_s,$$

és a 2.3.8. állitás alapján:

$$(2.3.10) \qquad \underline{b}_s = \hat{\underline{b}}_{j+s-k} \neq \underline{b}_q \,.$$

Most (2.3.10) miatt a 2.3.14. d/ állitás szerint $\underline{b}_s$ nem lehet fiktiv. Igy a sorút képzésénél az UT1-UT3 pontok szerint előálló sorozatban a $\underline{b}_{s+1}$ tag hurokpont. Ekkor $\underline{b}_{s+1} = \underline{b}_{\tilde{k}}$, ahol $\tilde{k} \geq i$, mert a 2.3.8. állitás alapján $\underline{b}_{\tilde{k}} = \hat{\underline{b}}_{j+s-k+1}$, azaz $\underline{b}_{\tilde{k}}$ tagja az oszlopútnak is. Ugyanakkor határozottan a $k = i$ egyenlőségnek kell teljesülni, mert $\tilde{k} = i$ esetében egyrészt $j \neq j+s-\tilde{k}+1 < q$; másrészt (a 2.3.14. c/ állitás szerint) $\hat{\underline{b}}_j = \underline{b}_k = \hat{\underline{b}}_{j+s-\tilde{k}+1}$; azaz a (2.3.6) sorozatnak többszörös tagja lenne, ami ellentmond a sorozat értelmezésének. Ugyancsak a 2.3.14. c/ állitásból kapjuk, hogy

$$(2.3.11) \qquad \hat{\underline{b}}_{j+s-k+1} = \underline{b}_i, \ldots, \hat{\underline{b}}_q = \underline{b}_{k-1}.$$

Mindezek alapján a

(2.3.12)

$$\underline{b}_k, \underline{b}_{k+1}, \ldots, \underline{b}_s, \underline{b}_i, \ldots, \underline{b}_{k-1}, \underline{b}_k$$

sorozat lánc, sőt olyan kör, amelyre a 2.1.16 állitás alkalmazható; igy a tagjainak halmaza hurok. Ez a (2.3.9)-cel és a (2.3.11)-gyel együtt éppen a d/ állitást adja. Qu. e. d.

2.4. A SAJÁTELEMEK SS1.-SS5 PONTOK SZERINTI ÉS A POTENCIÁLOK PP1-PP5 PONTOK SZERINTI SZÁRMAZTATÁSÁRA VONATKOZÓ TÉTELEK

2.4.1. Definició: Legyen g olyan

$$g : B \longrightarrow \{1, 2, \ldots, m+n\}$$

függvény, amelynek értelmezési tartománya a teljes B bázisvektorrendszer; kölcsönösen egyértelmü; és $g(\underline{b}) = i \leq m$ esetén $\underline{b}$ a DT i. sorában, $g(\underline{b}) = m+j > m$ esetén $\underline{b}$ a DT j. oszlopában helyezkedik el. Ekkor a g függvényt kvázi sajátelem kijelölésnek nevezzük.

A 2.2.1. és a 2.2.2 állitásból, valamint a 2.2.3. lemmából adódik:

2.4.1. Állitás: A sajátelem kijelölés egyben kvázi sajátelem kijelölés is.

Ha a 2.2.4. definicióban sajátelem kijelölés helyett kvázi sajátelem kijelölést értünk, akkor a kvázi SS-elemek, illetve kvázi OS-elemek definicióját kapjuk. A 2.4.1 definicióból triviálisan adódik:

2.4.2. Állitás: A kvázi sajátelem kijelölésre, valamint a kvázi SS- és OS-elemekre érvényesek a 2.2.1. és 2.2.2 állitásokkal, valamint a 2.2.3. lemmával analóg állitások.

Vegyük észre, hogy a hurkok létezésére és számára vonatkozó megállapitások kivételével, a disztribuciós eljárás lépéseit illetően a sajátelem kijelölésnek csak a 2.2.1 és a 2.2.2. állitásokban és a 2.2.3. lemmában foglalt tulajdonságait használtuk ki. Igy az eljárás ezen részeire vonatkozóan egy kvázi sajátelem kijelölés a 2.4.2. állitás szerint helyettesitheti a sajátelem kijelölést. - Nevezetesen a 2.3. fejezet minden állitása (lemmája) igaz kvázi sajátelem kijelölés esetében is. Ami a hurkok létezésére és számára vonatkozó kritériumot (2.2.5. lemma) illeti: az adott iterációs lépéshez tartozó potenciáloknak az előző iterációs lépéshez tartozókból való származtatásához csak azt szükséges tudni, keletkezett-e új hurok a legutolsó báziscsere alkalmával. E megfontolások szerint a sajátelem kijelölés és a kvázi sajátelem kijelölés céljainknak egyformán megfelelnek, ha már adottak a báziscsere előtti sajátelemek és potenciálok. Igy elegendő azt megmutatnunk, hogy az SS1-SS5. algoritmus kvázi sajátelem kijelölést határoz meg, illetve a "kvázi" jelzőt a továbbiakban el is hagyjuk.

(Megjegyzés: a "kvázi" jelző elhagyására több jogunk is van, mint amennyit az eddigiekben megindokoltunk. A 2.2.1. definició S7 szabályában az i_0, j_0 megválasztására adott utasitást ugyanis sehol nem használtuk ki, helyette más utasitást is előirhattunk volna. Igaz a következő állitás: Tetszőleges g kvázi sajátelem kijelölés esetén van az S7 szabálynak olyan g-től függő átfogalmazása, hogy az S1-S7 szabályok szerint előállitott sajátelem kijelölés azonos lesz g-vel. - Ennek bizonyitása itt csak felesleges kitérőt jelentene.)

2.4.1. <u>Lemma:</u> Az SS1 - SS5. algoritmus (kvázi) sajátelem kijelölést határoz meg.

<u>Bizonyitás:</u> A bizonyitást arra az esetre végezzük el, amikor $\underline{b}_0$ - az új báziselem - SS-elem lesz. - OS-elem esetében a bizonyitás hasonló.

Elegendő megmutatni, hogy az újonnan kijelölt SS-elemek sorában csak egy SS-elem lesz, illetve az új OS-elemek oszlopában csak egy OS-elem lesz. - Hiszen az előző sajátelem kijelölés 2.2.3 lemma szerinti tulajdonságai miatt, ha SS1-SS5 után például valamely sorban két SS-elem lenne, úgy azok közül legalább egy újonnan (az SS1-SS5 pontok szerint) kijelölt elem

Legyen

(2.4.1) $\underline{b}_0, \underline{b}_1, \underline{b}_2, \dots, \underline{b}_s$

az SS4 pont szerinti sorozat. Tegyük fel, hogy a $\underline{b}_i$ tag esik ki a bázisból. Nyilván $0 < i \leq s$, azaz a $\underline{b}_s$ tag már vagy kiesett a bázisból, vagy nem változtatja meg a sajátelem tulajdonságát. A $\underline{b}_0$ sorában nem lehet két SS-elem, mert abban az előző sajátelem kijelölés szerint csak egy SS-elem volt: $\underline{b}_1$. Ez vagy kiesett a bázisból vagy OS-elemmé változott. Az sem fordulhat elő, hogy a $\underline{b}_0$ sorában levő a (2.4.1) sorozathoz tartozó OS-elem vált volna át SS-elemmé, mert úgy az csak $\underline{b}_s$ lehetne, ami - mint láttuk - ellentmondás.

Tegyük fel, hogy a $\underline{b}_j$ $(j<i)$ tagig igaz, hogy az új SS-elemek (OS-elemek) sorában (oszlopában) csak egy SS-elem (OS-elem) van. Ha $\underline{b}_{j+1}$ kiesett a bázisból, akkor a lemma teljesült. Különben tegyük fel, hogy $\underline{b}_{j+1}$ SS-elemből vált OS-elemmé (ellenkezőleg a bizonyitás hasonló). Az új OS-elem oszlopában a régi OS-elem a $\underline{b}_{j+2}$ volt. Ez vagy kiesett a bázisból vagy SS-elemmé változott át. Az indukciós feltevés szerint $k<j+1$ esetén $\underline{b}_k$ nem lehet a $\underline{b}_{j+1}$ oszlopában levő OS-elem. Az sem fordulhat elő, hogy a $\underline{b}_{j+1}$ oszlopában levő, a (2.4.1) sorozatban a $\underline{b}_{j+1}$ után álló, másik SS-elem is OS-elemmé vált volna át, mert úgy az csak $\underline{b}_s$ lehetne, ami ellentmondás. Qu. e. d.

2.4.2. Lemma: Egy báziscsere alkalmával a DT-ben pontosan akkor keletkezik új hurok, ha az SS1-SS5 algoritmusban alkalmazni kell az SS3 pontot (azaz a sorút és az oszlopút közös tagja esik ki a bázisból). Az új hurok mindig tartalmazza az új báziselemet.

Bizonyitás: Az, hogy az új hurok mindig tartalmazza az új báziselemet, triviális állitás. Ebből az is következik, hogy csak egy új hurok keletkezhet. - Egyebekre nézve előbb tegyük fel, hogy új hurok keletkezett. Ekkor a 2.1.14. állitás szerint $\underline{b}_0$ - az új báziselem - nem fiktiv, azaz volt sorút is, oszlopút is. Tegyük fel, hogy a sorút valamelyik tagja esett ki a bázisból (ellenkezőleg a bizonyitás hasonló). Vegyük az oszlopútnak a hurokpontot megelőző részsorozatát (ha nincs hurokpont, akkor magát az oszloputat):

$$(2.4.2) \qquad \underline{b}_0, \hat{\underline{b}}_1, \hat{b}_2, \ldots, \hat{b}_q.$$

Ad abszurdum tegyük fel, hogy a sorútnak és az oszlopútnak nincs közös tagja (I); vagy a sorútnak olyan tagja esik ki a bázisból, ami egyidejüleg nem tagja az oszlopútnak (II). Ekkor a (2.4.2) sorozat minden $\hat{\underline{b}}_k$ $(0<k\leqq q)$ tagja ugyanolyan sajátelem marad a báziscsere után, mint amilyen előtte volt. - Ez az (I) esetben az SS5-ből közvetlenül adódik, a (II) esetben pedig a 2.3.3. lemma c/ pontja és az SS5 összevetéséből következik. Ezek után összehasonlitva az UT1-UT3 pontokat a HU1 algoritmussal, kapjuk: a (2.4.2) sorozat részsorozata a HU1. algoritmussal előállitható sorozatnak (aminek a tagjai hurkot alkotnak). Következésképpen $\hat{\underline{b}}_q$ nem fiktiv (lásd a 2.1.14. állitást). A (2.4.2) értelmezése alapján az UT2, UT3 szerint következő $\hat{\underline{b}}_{q+1}$ tag hurokpont (az UT1-UT3 szerinti sorozatban), azaz

$\hat{\underline{b}}_{q+1} = \hat{\underline{b}}_k$, $0 < k < q+1$. Ekkor a 2.1.6. lemma c/ része és a 2.3.3. állitás szerint a

$$\hat{\underline{b}}_k, \hat{\underline{b}}_{k+1}, \ldots, \hat{\underline{b}}_q$$

sorozat tagjainak halmaza hurok, ami ellentmond a hurok T3 tulajdonságának. - Ugyanis ez a hurok $k > 0$ miatt valódi részhalmaza lenne az új huroknak.

Most tegyük fel, hogy a sorútnak és az oszlopútnak egy közös tagja esett ki a bázisból; megmutatjuk, hogy a $\underline{b}_0$ elemet tartalmazó (és ezért új) hurok keletkezett.

Legyen

(2.4.3) $\underline{b}_0, \underline{b}_1, \underline{b}_2, \ldots, \underline{b}_s$

a sorútnak a hurokpontot megelőző részsorozata. Legyen $\underline{b}_i$ a (2.4.3)-ban az első olyan tag, amelyre: $\underline{b}_i = \hat{b}_j$, $i > 0$. Ekkor a 2.3.2. állitás szerint:

(2.4.4) $\varphi(\underline{b}_{i-1}, \underline{b}_i) = \varphi(\hat{\underline{b}}_{j-1}, \hat{\underline{b}}_j)$.

A (2.3.10) állitás szerint:

(2.4.5) $\varphi(\underline{b}_0, \underline{b}_1) \neq \varphi(\underline{b}_0, \hat{\underline{b}}_1)$.

A (2.4.4)-ből a 2.1.1. állitás szerint:

(2.4.6) $\varphi(\underline{b}_{i-1}, \underline{b}_i) \neq \varphi(\underline{b}_{i-1}, \hat{\underline{b}}_{j-1})$,

(2.4.7) $\varphi(\underline{b}_{i-1}, \underline{b}_{j-1}) = \varphi(\hat{\underline{b}}_{j-1}, \hat{\underline{b}}_j)$.

A (2.4.5), (2.4.6) és (2.4.7) összevetéséből és $\underline{b}_i$ választásából adódik, a

(2.4.8)

$$\underline{b}_0, \underline{b}_1, \ldots, \underline{b}_{i-1}, \hat{\underline{b}}_{j-1}, \hat{\underline{b}}_{j-2}, \ldots, \underline{b}_0$$

sorozat kör. Legyen $\hat{\underline{b}}_j$ a (2.4.2)-ben az első olyan tag, amely egyidejüleg a (2.4.3)-nak is tagja. Ekkor a (2.4.8) kör kielégiti a 2.1.16. állitás feltételeit, azaz a tagjainak halmaza (új hurok).

Most tegyük fel, hogy a (2.4.2) sorozatban $\hat{\underline{b}}_k (\neq b_j)$ az első olyan tag, amely egyidejüleg a sorútnak is tagja. Ekkor a 2.3.3. lemma d/ része szerint a két út közös tagjai;

$$\underline{b}_i, \underline{b}_{i+1}, \dots, \underline{b}_s ;$$

illetve

$$\hat{\underline{b}}_k, \hat{\underline{b}}_{k+1}, \dots, \hat{\underline{b}}_q$$

hurkot alkotnak. És fennáll:

(2.4.9)
$$\underline{b}_i = \hat{\underline{b}}_j, \quad \underline{b}_{i+1} = \hat{b}_{j+1}, \dots,$$
$$\dots, \underline{b}_{i+q-j} = \hat{\underline{b}}_q, \underline{b}_{i+q-j+1} = \hat{\underline{b}}_k, \dots$$
$$\dots, \underline{b}_s = \hat{\underline{b}}_{j-1} .$$

Ha most a bázisból a

$$\underline{b}_i = \hat{\underline{b}}_j, \dots, \underline{b}_{i+q-j} = \hat{\underline{b}}_q$$

elemek valamelyike esik ki, akkor a (2.4.8) sorozatról ismét elmondhatók: a sorozat kör; a tagjainak halmaza (új) hurok.

Tegyük fel, hogy a bázisból a

(2.4.10)
$$\underline{b}_{i+q-j+1} = \hat{\underline{b}}_k, \dots; \underline{b}_{s-1} = \hat{\underline{b}}_{j-2}, \underline{b}_s = \hat{\underline{b}}_{j-1}$$

elemek valamelyike esik ki. Tekintsük a

(2.4.11)
$$\underline{b}_0, \hat{b}_1, \dots, \hat{\underline{b}}_{k-1}, \underline{b}_{i+q-j}, \underline{b}_{i+q-j-1}, \dots, \underline{b}_1, \underline{b}_0$$

sorozatot. A (2.4.4)-gyel, (2.4.6)-tal és (2.4.7)-tel analóg egyenlőségeket belátva, a $\hat{\underline{b}}_k$ választásából és a (2.4.10)-re tett feltevésből adódik, a (2.4.11) sorozat olyan kör, amelynek tagjai (új) hurkot alkotnak. Qu.e.d.

2.4.3. Lemma: Minden $\underline{b}_1$ báziselemre igaz a következő állitások közül pontosan egy:

a/ a $\underline{b}_1$ fiktiv;

b/ van pontosan egy olyan $\underline{b}_1$-ből induló, báziselemekből álló lánc, amelynek pontosan az utolsó tagja fiktiv;

c/ a $\underline{b}_1$ eleme pontosan egy huroknak;

d/ van pontosan egy olyan $\underline{b}_1$-ből induló nem-fiktiv báziselemekből álló sorozat, amelynek tagjai nem tartoznak hurokhoz, de létezik pontosan egy olyan hurok, amelynek valamely elemével a sorozatot folytatva, láncot kapunk.

Bizonyitás: Ha a $\underline{b}_1$ fiktiv, akkor rá a b/ vagy a d/ állitások már azért nem teljesülhetnek, mert úgy a 2.1.4. lemma szerint a bázisvektorrendszer lineáris függősége következne, ami ellentmondás; a c/ állitás a/-val egyidejü fennállása pedig a 2.1.14. állitás szerint lehetetlen.

A továbbiakban tegyük fel, hogy a $\underline{b}_1$ nem fiktiv. Belőle kiindulva alkalmazzuk az UT2, UT3 pontokat. A kapott

(2.4.12) $$\underline{b}_1, \underline{b}_2, \dots, \underline{b}_s$$

sorozatra teljesülnek a 2.3.1, 2.3.2, 2.3.3, 2.3.4, 2.3.7. és 2.3.8. állitások.

Tegyük fel, hogy $\underline{b}_s$ fiktiv. Ekkor a (2.4.12) sorozat a 2.3.3. állitás szerint lánc, és a 2.3.1. állitás szerint pontosan az utolsó tagja fiktiv. Ha most lenne egy másik (báziselemekből álló)

(2.4.13) $$\underline{b}_1, \hat{\underline{b}}_2, \dots, \hat{\underline{b}}_r$$

lánc, amelynek szintén pontosan az utolsó tagja fiktiv, abból megint a 2.1.4. lemma szerint következne ellentmondás. (Ha ugyanis $\hat{\underline{b}}_k$ a (2.4.13) láncnak az első olyan tagja, amelyre

$\hat{\underline{b}}_k \neq \underline{b}_k$, akkor $k = 2$ esetben a

$$\underline{b}_s, \underline{b}_{s-1}, \dots, \underline{b}_2, \underline{b}_1, \hat{\underline{b}}_2, \dots, \hat{\underline{b}}_r$$

sorozatot, egyébként pedig a

$$\underline{b}_s, \underline{b}_{s-1}, \dots, \underline{b}_k, \hat{\underline{b}}_k, \dots, \hat{\underline{b}}_{r-1}, \hat{\underline{b}}_r$$

sorozatot megfeleltetve a (2.1.8)-nak, teljesülnek a 2.1.4. lemma feltételei.) Végeredményben, ha $\underline{b}_s$ fiktiv, úgy a b/ állitás igaz. A c/ vagy a d/ állitás egyidejü fennállása ismét a 2.1.4 lemma szerint vezet ellentmondáshoz.

A továbbiakban feltesszük, hogy a $\underline{b}_s$ nem fiktiv, azaz (a 2.3.1. állitás szerint) hurokpont. Egyelőre tegyük még fel, hogy $\underline{b}_1 = \underline{b}_s$. Ekkor a 2.3.4. állitás szerint a $\underline{b}_2$ és $\underline{b}_{s-1}$ tagok léteznek, és az UT2, UT3 pontok, illetve a 2.3.2. állitás alapján

$$(0 \neq)\, \varphi(\underline{b}_1, \underline{b}_2) \neq \varphi(\underline{b}_{s-1}, \underline{b}_s) \;\; (\neq 0).$$

Tehát a (2.4.12) lánc most olyan kör, amelyre teljesülnek a 2.1.2. lemma feltételei, és - mert tagjai báziselemek - a 2.1.16. állitás feltételei is. Következik, hogy a (2.4.12) tagjainak halmaza hurok. A 2.2.4. lemma b/ része, a 2.2.8 definició, a 2.2.10. állitás és a 2.1.7. lemma d/ pontja alapján más hurok $\underline{b}_1$-et nem tartalmazhatja. Mindezekből következik: ezesetben a c/ állitás teljesül. A d/ pont egyidejü fennállása ellentmond a 2.1.10. lemmának. (Ha ugyanis a két (különböző) hurkot a 2.1.10. lemma szerinti F_1, F_2 halmazoknak, az összekötő láncot pedig a (2.1.21) láncnak feleltetjük meg, úgy a bázisvektorrendszer lineáris függőségét kapjuk.)

A továbbiakban tegyük fel, hogy $\underline{b}_s = \underline{b}_i$, $1 < i < s$. Megmutatjuk, hogy ekkor a d/ állitás teljesül. A 2.3.2. állitás, a 2.3.3. állitás és a 2.3.1. lemma a/ része szerint a

(2.4.14) $\underline{b}_i$, $\underline{b}_{i+1}$, $\underline{b}_s$

sorozat kör, a tagjainak halmaza hurok. A

(2.4.15) $\underline{b}_1, \underline{b}_2, \dots, \underline{b}_{i-1}$

sorozat tagjai nem tartoznak a hurokhoz, de a sorozatot a hurok $\underline{b}_i$ elemével folytatva láncot kapunk (2.3.3. állitás). Most tegyük fel, hogy van egy

(2.4.16) $\underline{b}_1, \hat{\underline{b}}_2, \dots, \hat{\underline{b}}_{j-1}$

sorozat, amelynek tagjai nem fiktiv báziselemek; és van egy F hurok, amelynek valamely elemét a (2.4.16)-hoz mint $\hat{\underline{b}}_j$ tagot hozzávéve, láncot kapunk. Ad abszurdum tegyük még fel, hogy vagy a (2.4.16) sorozat különbözik a (2.4.15) sorozattól vagy az F hurok a (2.4.14) szerinti huroktól. Előbbi esetben legyen $\hat{\underline{b}}_k$ a (2.4.16) első olyan tagja, amelyre: $\hat{b}_k \neq \underline{b}_k$. - Ha k = 2, akkor a

$$\underline{b}_i, \underline{b}_{i-1}, \dots, \underline{b}_2, \underline{b}_1, \hat{\underline{b}}_2, \dots, \hat{\underline{b}}_j$$

sorozatot, egyébként pedig a

$$\underline{b}_i, \underline{b}_{i-1}, \dots, \underline{b}_k, \hat{\underline{b}}_k, \hat{\underline{b}}_{k+1}, \dots, \hat{\underline{b}}_j$$

sorozatot megfeleltetve a (2.1.21)-nek, a két (nem feltétlenül különböző) hurkot pedig a 2.1.10. lemma F_1, F_2 halmazainak, teljesülnek a 2.1.10. lemma feltételei, ami ellentmondás.

Hátra van még az az eset, amikor a (2.4.15) és a (2.4.16) sorozatok azonosak, de az F és a (2.4.14) szerinti hurok különbözik. Ha $i = j > 2$, akkor a lánc értelmezése szerint:

(2.4.17) $0 \neq \varphi(\underline{b}_{i-1}, \underline{b}_i) = \varphi(\underline{b}_{i-1}, \hat{\underline{b}}_i) \neq 0$,

ám a 2.2.4. lemma b/ részének, a 2.2.8. definiciónak, a 2.2.10. állitásnak és a 2.1.7. lemma d/ pontjának összevetéséből kapjuk, hogy (2.4.17) ellentmondás. Természetesen a (2.4.17) $i=j=2$ esetében is ugyanigy ellentmondást jelent, de ekkor fennállhat még a

$$0 \neq \varphi(\underline{b}_1, \underline{b}_2) \neq \varphi(\underline{b}_1, \hat{\underline{b}}_2) \neq 0$$

lehetőség. Most a két hurkot megint megfeleltetve a 2.1.10. lemmánál tárgyalt F_1, F_2 halmazoknak a $\underline{b}_2$, $\underline{b}_1$, $\hat{\underline{b}}_2$ láncot pedig (2.1.21)-nek a 2.1.10 lemma alapján ismét ellentmondás adódik. Tehát $\underline{b}_s = \underline{b}_i$, $1 < i < s$ esetben a d/ állitás tagadása mindenképpen ellentmondáshoz vezetett. Q.u.e.d.

Az 1.4.1.-1.4.3. állitásokban szereplő összefüggések alapján egy adott bázisvektorrendszer esetén a nem-fiktiv bázisvektorokra vonatkozóan.

(2.4.18) $\qquad f_{i,j} \quad u_i + v_j = c_{i,j}$

alakú egyenletek, a fiktiv bázisvektorokra pedig

$$u_i = 0 \qquad (\text{FO-elemek}),$$

illetve

$$v_j = 0 \qquad (\text{FS-elemek})$$

alakú összefüggések adódnak. Az $(m+n)$-elemü bázisvektorrendszer esetén igy $(m+n)$ egyenletből álló egyenletrendszert kapunk a potenciálokra. A bázisvektorrendszer lineáris függetlenségéből következik, hogy az egyenletrendszert pontosan egy megoldásvektor elégiti ki. Tehát:

2.4.3. <u>Állitás:</u> A DT egy adott bázisára vonatkozóan az 1.4.1-1.4.3. állitások szerinti összefüggések a potenciálokat egyértelmüen meghatározzák.

A P1-P5 és a HU1-HU3 pontokban (1.4. fejezet) szereplő valamennyi rekurziós formula az 1.4.1-1.4.3. állitások véges sok alkalmazásával származtatható. Következésképpen:

2.4.4. Állitás: A DT egy adott bázisvektorrendszere esetén a P1-P5 algoritmus (1.4.1. definició szerinti) potenciálokat állit elő.

A P4-P5 pontokról a lépésszám szerinti indukcióval belátható:

2.4.5. Állitás: A P4-P5 pontok azoknak és csak azoknak az SS-, illetve OS-elemeknek a sorához, illetve oszlopához rendelnek potenciált, amelyeket báziselemekből álló lánc köti össze egy olyan báziselemmel, melyre a P1-P3 pontok valamelyike vonatkozott.

A 2.4.3. lemma szerint lehetséges négy esetet a 2.4.5. állitással összevetve, adódik:

2.4.6. Állitás: Egy adott bázisvektorrendszer esetén a P1-P5 algoritmus az összes potenciált előállitja.

A hurok-algoritmus, valamint a P4-P5 pontok alapján belátható:

2.4.7. Állitás: Valamely hurok soraihoz, illetve oszlopaihoz tartozó potenciálok csak a hurokban levő báziselemekhez tartozó $c_{i,j}$ és $f_{i,j}$ értékektől függnek.

Ugyancsak a P4-P5 pontokból következik:

2.4.8. Állitás: Ha a DT egy adott bázisvektorrendszere esetén az

$$\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_n$$

sorozat báziselemekből álló olyan lánc, amelyben $\underline{a}_n$ fiktiv, de a többi tag nem fiktiv, akkor az $\underline{a}_1$ SS-elem (vagy OS-elem) sorához (vagy oszlopához) tartozó potenciál csak a sorozat tagjaihoz tartozó $c_{i,j}$, $f_{i,j}$ együtthatóktól függ.

A hurok-algoritmusból és a P4-P5 pontokból kapjuk:

2.4.9. Állitás: Ha az $\underline{a}_1$ SS-elemhez (OS-elemhez) van olyan

$$\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n$$

báziselemekből álló sorozat, amely valamely hurok valamely elemével úgy folytatható, hogy láncot kapjunk, akkor az $\underline{a}_1$ sorához (oszlopához) tartozó potenciál csak a hurokban és a láncban levő báziselemekhez tartozó $c_{i,j}$, és $f_{i,j}$ együtthatóktól függ.

2.4.4. Lemma: A PP1-PP6 algoritmus (1.6. fejezet) adott bázisvektorrendszer esetén az 1.4.1. definició szerinti potenciálokat állitja elő, és valamennyit előállitja.

Bizonyitás: A 2.4.4. és a 2.4.6 állitások szerint elegendő megmutatni, hogy a PP1-PP6 algoritmus az adott bázis esetén pontosan azokat az u_i, v_j értékeket állitja elő, amelyeket a P1-P5 algoritmus is származtatna (feltéve, hogy ismertek a legutolsó elemi bázistranszformáció előtti pontenciálok).

Ha az új bázisvektor fiktiv, akkor a P1, illetve P2 pontokat összevetve a PP1, illetve PP2 pontokkal, kapjuk: mindkét algoritmus ugyanolyan potenciált rendel az új báziselemhez. A lépésszám szerinti indukcióval belátható, hogy ilyenkor a PP4-PP5 pontok az összes olyan és csak olyan nem-fiktiv SS-elem (OS-elem) sorához (oszlopához) rendelnek potenciált, amelyet báziselemekből álló lánc köt össze az új báziselemmel; és ugyanazokat az u_i, v_j értékeket származtatják, mint a P4-P5 pontok.

Ha az új báziselem egy új hurokhoz tartozik, akkor az új hurok valamely SS-elemének sorához tartozó új potenciált a hurok-algoritmussal pontosan úgy határozzuk meg mint a P1-P5 algoritmusnál, csupán a kiindulásul választott báziselem lehet más, mint ott. A hurok-algoritmus induló elemének változtatása azonban a származtatott potenciálok értékét nem befolyásolja, hiszen az algoritmus egy adott hurok esetén mindig ugyanazon – (2.4.18) alakú egyenletekből álló – egyenletrendszer megoldását jelenti, amely egyenletrendszer (a 2, 4, 3. állitás alapján) a potenciálokat egyértelmüen meghatározza. A lépésszám szerinti indukcióval ilyenkor azt kapjuk, hogy a PP4-PP5 pontok az összes olyan és csak olyan (nem-fiktiv) SS-elem (OS-elem) sorához (oszlopához) rendelnek potenciált, amelyet báziselemekből álló lánc köt össze a hurok-algoritmus kezdő elemével; és ugyanazokat az u_i , v_j értékeket származtatják, mint a P4-P5 pontok.

Ha az új báziselem nem fiktiv és nem tartozik hurokhoz, akkor rá a 2. 4. 3. lemma b/ és d/ állitásai közül pontosan egyik teljesül (éppen a 2.4.3. lemma szerint). Ha feltesszük, hogy az új báziselem SS-elem (ellenkezőleg a bizonyitás hasonló), azaz az utolsó báziscsere alkalmával a sorút valamely tagja esett ki a bázisból, akkor a 2. 4. 3. lemma b/ állitása szerinti lánc (ha ez az állitás teljesül) éppen az oszlopúttal azonos; a d/ állitás szerinti sorozat pedig (ha ez az állitás teljesül) éppen az – egy hurkot bejáró – oszlopútnak a hurkon kivüli tagjaiból áll. – Mindkét esetben következik (a 2.4.8. állitás, illetve a 2.4.7. és 2.4.9 állitások alapján), hogy az új báziselem oszlopához tartozó potenciált ugyanazok a (2. 4. 18) alakú egyenletek, és ugyanazok a $c_{i,j}$, $f_{i,j}$ együtthatók határozzák meg, mint a bázistranszformáció előtt; azaz a régi oszloppotenciál marad érvényben. Ebből az új báziselem sorához tartozó potenciált a PP1 szabály ugyanúgy származtatja, mint a P4 szabály. A lépésszám szerinti indukcióval most az látható be, hogy a PP4-PP5 szabályok az összes olyan és csak olyan (nem-fiktiv) SS-elem (OS-elem) sorához (oszlopához) rendelnek új potenciált, amelyek nem tartoznak az oszlopúthoz, és báziselemekből álló lánc köti össze őket az új báziselemmel; és a PP4-PP5 pontok a szóbanforgó báziselemek vonatkozásában ugyanazokat az u_i, v_j értékeket származtatják, mint a P4-P5 pontok.

Jelölje H azoknak az SS- és OS-elemeknek a halmazát amelyeknek sorához, illetve oszlopához a PP1-PP5 pontok nem rendeltek új potenciált. Az eddig elmondottakból következnek:

a/ A H elemei mind szerepeltek a báziscsere előtti bázisban is.

b/ Ha a H valamely elemére a 2.4.3. lemma b/ állitása teljesül, akkor a lánc minden tagja eleme a H halmaznak.

c/ Ha a H valamely elemére a 2.4.3. lemma c/ állitása teljesül, akkor a hurok részhalmaza a H halmaznak.

d/ Ha a H valamely elemére a 2.4.3 lemma d/ állitása teljesül, akkor a megfelelő sorozat minden tagja és a megfelelő hurok minden eleme a H halmazhoz tartozik.

Az a/-d/ pontokat a 2.4.7.-2.4.9. állitásokkal összevetve, kapjuk: a H elemeinek sorához, illetve oszlopához tartozó potenciálok értéke ugyanaz lesz a báziscsere után is, mint előtte volt. - Ezt összevetve a PP6 ponttal, következik, hogy a PP6 pont a H elemeinek és csak a H elemeinek sorához, illetve oszlopához rendel potenciált; és a H elemeihez a PP1-PP6 algoritmus ugyanazokat az u_i, v_j értékeket rendeli mint a P1-P5 algoritmus. Q.u.e.d.

IRODALOMJEGYZÉK

[1] Dantzig G. B.: "Linear Programming and Extensions", Princeton University Press, Princeton, New Yersey (1963).

[2] Dantzig G. B.: "Lineare Programmierung und Erweiterungen" Berlin - Heidelberg - New York (1966).

[3] Glover F. - Klingman D.: "Locating stepping-stone paths in distribution problems via the predecessor index method", Transportation Science 4 (2) (1970) 220-225.

[4] Glover F. - Karney D. - Klingman D.: "The augmented predecessor index method for locating stpping - stone paths and assigning dual prices in distribution problems", Transportation Science 6 (2) (1972) 171-180.

[5] Krekó B.: "Lineáris programozás", Közgazdasági és Jogi Könyvkiadó, Budapest (1966)

[6] Pogány Zs.: "An algorithm for solving the generalized transportation problem", 8th IFIP Conference on Optimization Techniques, Würzburg, W. Germany, 5.-9. September 1977, ESZK/2 Models and Methods I., Budapest (1978)

[7] Varga J.: "Gyakorlati programozás I"., Tankönyvkiadó, Budapest (1968).

Gyurkó György

ALGORITHM OF THE SOLUTION OF THE GENERAL DISTRIBUTION PROBLEM USING THE LINE- AND THE COLUMN-EIGENELEMENT CONCEPTS

/ Abstract /

The general distribution /transportation/ problem /abbreviated GDP/ is a special linear programming task. They can be solved with the help of the Dantzig's method, however, the inherent particularities of the GDP enable the solution of the simplex transformations by means of considerably economical operations - with the aid of the so called distribution procedure /abbreviated DP/.

The DP usually means methods defined above graphs consisting of the base elements - as nodes - of the so called distribution table /abbreviated DT/. In the case of simpler task these graphs can be seen at once. With computer backed solution of more complex task, however, their designation involves rather serious problems of algorithmisation. Luckily, it also comes from the particular features of the problem that - for definition of the graphs - suitable "signposts" can be located in the "crossroads" of the DT.

In the algorithm, described here, the role of the "signposts" is fulfilled by the line- and column-eigenelement /abbreviated LE and CE element/ concepts. This assembly of concepts mathematically means a function interpreted above the base element of the DT. - To put it shortly, in each /non-fictive/ line of the DT there is exactly one LE element and in every /non-fictive/ column there is exactly one CE element. - Each step of the solution means some readily programmable

recursive procedure defined above the eigenelements.

Example: When determining the components of the element /DT-vector/ to be included in the base, the so called line path and/or column path - formed by the element to be included in the base and by the base elements - plays decesive role. A line path can be designated as follows: from the element to be included in the base one has to step to the EL element being in its line; otherwise from the EL element one has to step to the CE element in its column, from the CE element one has to step to the LE element in its line, etc... As it could be seen here the eigen-element concept makes it possible that the n-th member of the path looked for unambiguously points at the /n+1/st member.

The DP, described here, and the Dantzig's method are of equal value as far as the GDP but the former require much smaller sets of data and the time, spent on the solution of the problem, is much shorter, too.