KFKI-71-73

Bagyinszki János

DIGITÁLIS RENDSZEREK OPTIMÁLIS HUZALOZÁSÁNAK SZÁMOLÓGÉPES TERVEZÉSE

Hungarian Academy of Sciences

CENTRAL RESEARCH INSTITUTE FOR PHYSICS

BUDAPEST

2017

KFKI-71-73

DIGITÁLIS RENDSZEREK OPTIMÁLIS HUZALOZÁSÁNAK SZÁMOLÓGÉPES TERVEZÉSE

Bagyinszki János

MTA. Központi Fizikai Kutató Intézete, Budapest
Elektronikus Főosztály

ABSTRACT

This paper deals with one of the most important chapters of automatic design of digital systems namely the optimization of back board wireing. In the theoretic part an algorithm is described giving a "good" initial placing of printed circuit boards and on the other hand a further algorithm is described issueing from this placing with the help of a given category iteration which creates the optimal placing from point of view of a target function. General composition assures utilizability of the above mentioned algorithms by the placing programme of the component parts at printed circuit board designe. After description of the computer programme made for the ICL-1905 in details the work discusses the possibility of generalization and further development as well.

KIVONAT

A dolgozatban a digitális rendszerek automatikus tervezésének egy fontos fejezetével, az optimális hátlaphuzalozással foglalkozunk. Az elméleti részben a kártyák egy "jó" kezdeti elrendezését adó algoritmust és egy, ebből adott tipusu iterációval elérhető, célfüggvényünkre nézve optimális elrendezést eredményező algoritmust irunk le. Az általános fogalmazás biztositja, hogy az algoritmusokat a nyomtatott-áramkör tervezés alkatrész-elrendező részprogramjaként is felhasználhassuk. Az ICT 1905 számológépre elkészitett program részletes leirása után foglalkozunk az általánositási és továbbfejlesztési lehetőségekkel is.

Резюме

Работа занимается одним из важнейших разделов автоматического проектирования цифровых систем именно оптимацией задних соединений. В теоретической части описывается алгоритм дающий "хорошее" начальное размещение сублоков и с другой стороны алгоритм достигаемый из этого размещения с помощью итерации данного типа производящий оптимальное размещение из начального с точки зрения целовой функции. Общее изложение обеспечивает использоваемость высших алгоритмов размещающей программой деталей при проектировании печатных плат. После подробного описания программы, подготовленной на ЭЦВМ типа ICT-I905 работа занимается и возможностями обобщения и дальнейшего развития.

1. BEVEZETÉS

1.1. A hagyományos mérnöki tervezésben nagy gyakorlattal rendelkező tervezők előtt ismeretes, hogy a tervezés egyes fázisai, vagy csaknem az egésze mechanikusan végezhető. A tervezés jellege meghatározza azokat a paramétereket és korlátokat, melyeket a tervezőnek a munka során figyelembe kell vagy lehet venni. Több paraméter esetén ezek együttes hatása áttekinthetetlenné válik /különösen, ha ezek nem függetlenek, esetleg egymásnak ellentmondó követelmények határozzák meg az optimumot vagy a megengedett megoldást/. Ezért a konstruktőr kénytelen iterativ tervezést alkalmazni. Ez abból áll, hogy egy kezdeti, intuitiv felvételből kiindulva elvégzi a szükséges számitásokat, s ha az eredmény nem esik a megengedett tartományba, módositott felvételből kiindulva ismétli meg az eljárást. Az egyes módositások lehetnek ugyancsak intuitiv jellegüek, de az előző eredmények tendenciájából is kiindulhatnak.

Az ilyen optimalizálási feladatok megoldására - nagy adathalmaz vagy sok iteráció esetén - igen előnyös a számológép használata. Ha az eredményből a módositásra maga a számológép következtet, /teljesen/ <u>automatizált tervezésről</u> beszélünk. Ha viszont /pl. bonyolultabb esetben, bizonyos szituációk esetén a döntést kilátástalan a gépre bizni/ olyan tipusu ember-gép együttmüködés jön létre, hogy bizonyos döntéseket az ember végez, egyébként a számológép végzi a közbülső, hosszadalmas számitásokat, <u>interaktiv tervezésről</u>, vagy számológép segitségével történő tervezésről /CAD/ beszélünk. Az ember a számológéppel a döntéseit valamilyen egyszerüen kódolt /probléma-orientált/ formában közli, s ugyanigy kapja a gép részéről a kérdéseket. Gyakorlati szempontból megkülönböztetett figyelmet érdemel az ember-gép kapcsolatban a vizuális kommunikáció <u>/grafikus-display/</u>.

A bonyolult digitális és analóg rendszerek tervezése területén egyaránt jelentős szerepet kap /hazánkban is a közeljövőben/ az előzőekben kiemelt mindkét számológépes tervezési mód. Ennek szükségszerüsége a rendszerek bonyolultsági fokának növekedési tendenciájából következik.

1.2. A nagy rendszerek /pl. gyorsmüködésü univerzális számológépek/ tervezésének és gyártásanak teljes automatizálása /bemenet: a tervezendő rendszerrel szemben támasztott igények, kimenet: pl. rajz és teljes dokumentáció, technológiai előirások, sőt esetleg programvezérelt szerszámgépek kimenete/ jelenleg és a közeljövőben nyilvánvalóan nem járható ut. Azonban a tervezés egyes fázisainak /rendszer-, logikai- és áramköri szinten való tervezés, analizis és diagnosztika, modellezés, áramkörkészlet tervezése, dokumentáció és rajzgépek alkalmazása, konstrukciós kérdések stb./ számológéppel való kivitelezése hazánkban is a jelen problémájává - és részben gyakorlatává - vált. A helyzet országos szintü felmérése helyett /ezt részben elvégezte pl. a Számitógép-technika '71 - Esztergom - konferencia/ csak a KFKI területére szoritkozva a következő saját /szerény/ eredményről számolhatunk be.

1967-ben elkészült két, a logikai függvényeket minimalizáló program /a különböző igények, feltételek kielégitése indokolta a kétféle megoldást/. Erről részletesen beszámoltunk korábban [4],[5],[6]. A kártyákból felépitett digitális rendszerek optimális kártyaelhelyezését és optimális hátlaphuzalozását is alátámasztottuk programrendszerrel, amely 1969-ben készült el. Ezt röviden tárgyalja [5], és alkatrész-elrendezés szempontjából - mint felhasználható részprogramot - emliti [7]. A részletes elemzés azonban éppen e dolgozat feladata.

Végül az áramkör-tervezés egy speciális területén, a nyomtatott kártyák optimális fólia-tervezésénél felmerülő problémák egy részének megoldását irja le a [7] dolgozat. Ebben gyakorlati szempontból is érdekesnek tünő uj elméleti eredményt adunk, azonban az algoritmus számológépre vitele nem történt meg.

1.3. A logikai tervezés - nagy rendszerek konstrukciójában - veszitett jelentőségéből a technológia megváltozása következtében /készen kapható nagyobb, integrált áramköri modulok megjelenése/, ill. főként a modulokat gyártó és tervező intézetekre szoritkozik. Ugyanakkor megjelent egy ujabb problémakör, amely fokozottabban kombinatorikus jellegü és általánosan igy körvonalazható: bizonyos, előirt követelményeknek eleget tevő elrendezések közül válasszunk ki /egy/ olyat, amely egy előre megadott célfüggvényre nézve optimális /diszkrét optimalizálás/.

Ide tartozik a jelen dolgozatban tárgyalt problémák jelentős része is. Mielőtt ennek részletes tárgyalását elkezdenénk, néhány mondatban a témával szoros kapcsolatban lévő területekről szólunk, ill. a tájékozódás megkönnyitése végett irodalomra hivatkozunk.

Nem foglalkozunk itt azzal az egyébként igen fontos kérdéssel, hogy adott alapáramkör-készlet /különböző integráltsági foku áramköri elemek, kártyák stb./ esetén hogyan célszerü meghatározni a rendszerhez felhasználandó /különféle előirt feltételeket teljesitő/ szükséges és elégséges halmazt /SE/. Ez olyan halmaz, amely minden logikai /és nem-logikai/ egységből elegendő számut tartalmaz, de bármely elemét elhagyva, ez a tulajdonsága megszünik.

Rögzitett SE halmaz esetén, a keresett logikai elrendezéssel ekvivalens elrendezések /pl. kapuk és lábak cseréje stb./ halmazából /LE halmaz/ meghatározandó egy olyan elem, amely egy előirt célfüggvényre nézve optimális. Ezek közül a lehetséges SE halmazok közül azt választjuk, amelyre az LE-optimumok közül a legkedvezőbb adódik. A probléma matematikai vonatkozása: olyan sorrendben előirt gyakorlati korlátozások megadása, amely ésszerü határok között /kombinatorikus, azaz diszkrét halmazon értelmezett függvények elmélete területén/ megoldást biztosit. Egy elvileg adott rendszer kártyákra bontásának problémája az előbbiekkel szoros kapcsolatban van; bizonyos részeredményt ad ezen a téren is a dolgozatunkban megadott algoritmus - alkalmas sulyozás esetén.
A következőkben /1., 2. és 3. rész/ az SE és LE halmazok rögzitettek.

1.4. A jelen dolgozatban tárgyalt permutációs jellegü matematikai probléma igy fogalmazható: adott egy $H=\{h_1,\dots,h_N\}$ N-elemü véges halmaz és az I index halmazán értelmezett rendezés. Jelölje a $P=(p_1,\dots,p_N)$ azt az I halmazon értelmezett permutációt, melyre $p_j \in I$ a j. pozicióban levő elemet jelöli. Bizonyos permutációkat kizárva, jelölje a megengedett permutációk halmazát π.

A π elemein értelmezett c/P/ célfüggvénye az optimum-probléma: keresendő olyan P megengedett permutáció, amelyre c/P/ minimumot vesz fel, azaz amelyre $c/P_0/ = \min \{c/P/ | P \in \pi\}$
Az ilyen tipusu problémáknak jelentős irodalma van, legismertebb az "utazó ügynök probléma" néven ismeretes speciális eset.
Különféle megoldásokat adó néhány módszer:

- elágazás és korlátozás módszere
- dinamikus programozás
- pszeudo-Boole függvények elméletén alapuló módszer [16, 3]
- "magyar módszer" [21,22,3,7,16,25]
- a diszkrét optimalizálás [30]-ban leirt eljárása.

A dinamikus programozás szempontjából a probléma igy fogalmazható: Jelölje $f_j(i_1, i_2, \ldots i_{j-1}; i)$ a minimális függvényértéket azon elrendezés esetén, amikor az i pont a j-edik helyen van és az $i_1, i_2, \ldots, i_{j-1}$ pontok megelőzik i-t. Ekkor a következő függvényegyenlet irható fel:

$$f_j(i_1, i_2, \ldots, i_{j-1}; i) = \min_{1 \le k \le j-1} \{d(i_k, i) + f_{j-1}(i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_{j-1}; i_k)\},$$

$j=1\ldots N$ és minden $(i_1, i_2, \ldots, i_{j-1}; i)$ sorozatra, ahol $d(i,j)$ a pontpárhoz rendelt távolságot jelenti.
Végül az alábbi táblázatban irodalmi utmutatást adunk a témával szoros kapcsolatban álló néhány területre:

Vékony- és vastagréteg-áramkörök:	[1,29,33]
Nyomtatott áramkör tervezése:	[1,3,7,9,11,12,18,20,23,24,28]
Dokumentáció és gépi rajzolás:	[2,11,32]
Grafikus display:	[10,23]
Alkatrész-elrendezés:	[3,4,11,17,18]
Áramkör-analizis:	[8]
Logikai tervezés:	[4,5,6,19]
Diszkrét optimalizálás	[3,7,16,20,24,30]
Permutációs problémák:	[3,5,14,17,21,22,25,26,27,31]
Rendező algoritmus:	[13,7]

Végezetül megköszönöm Király Péternének a program elkészitésén tulmenően, valamint Rosta Jánosnak a lektorálás során tett értékes megjegyzéseit.

2. HUZALOZÁSI PROBLÉMA

Részben megbizhatósági és szervizproblémák szempontjából, részben müszaki /áramköri stb./ okokból kivánatos, hogy a kisebb standard egységekből/ modulok, kártyák stb./ felépitett rendszerek huzalozása a lehetőségekhez képest egyszerü legyen. A problémát egyszerübb formájában ugy lehet megfogalmazni, hogy a huzalozásnál az összhuzalhossz /vagy annak egy egyszerü függvénye/ legyen minimális, esetleg bizonyos korlátozások mellett. /Bizonyos meghatározott huzalok induktiv, kapacitiv vagy más szempontból nem kerülhetnek tulságosan közel egymáshoz, másrészt megadott időkésleltetések meghatározzák a késleltető vonal hosszát./ A probléma matematikai leirása előtt körvonalazzuk annak müszaki vonatkozásait. Adott térbeli elhelyezésben lévő "dobozok" tárolják a kártyákat, melyek kimenetei, lábai adott módon összehuzalozandók. A feladat a kártyák olyan elrendezése, melyre a "huzalhossz-függvény" - a korlátozó feltételek kielégitése mellett - minimális értéket vesz fel.

2.1. <u>Matematikai megfogalmazás</u>

Legyen adott az ℓ-dimenziós vektorok $\alpha=\{\alpha_1,\dots,\alpha_m\}$ halmaza, ahol $\alpha_i=(\alpha_{i_1},\dots,\alpha_{i_\ell})$ $/i = 1, \dots, m/$. Legyen $\overline{\alpha_{iu}\,\alpha_{jv}}$ összeköttetés sulya $c_{iu,jv} \geq 0$/$c_{iu,iu} = 0$ és rendszerint a $c = \|c_{iu,jv}\|$ összeköttetés-mátrix szimmetrikus/.
Legyen adott továbbá az ℓ-dimenziós vektorok egy $\beta=\{\beta_1,\dots,\beta_n\}$ halmaza, $\beta_i=(\beta_{i1},\dots,\beta_{i\ell})$, $i = 1,\dots,n$; $n \geq m$.
/Itt β a lehetséges kártyahelyek halmaza, rögzitett térbeli elrendezésben, α pedig az elhelyezendő kártyák halmaza./ A $D = \|d_{iu,jv}\|$ mátrix szintén szimmetrikus, $d_{iu,iu} = 0$, ahol $d_{iu,jv}=d(\beta_{iu},\beta_{jv})$ egy mérték: a β_{iu} és β_{jv} pontok összekötésének huzalhossz-függvénye. Ez lehet az euklidesi geometria távolságmértéke, lehet a <u>koordináta-különbségek abszolutértékeinek összege</u>, vagy más, bonyolultabb függvény. Jelölje S az E_m halmaz E_n halmazba való kölcsönösen egyértelmü leképezéseinek halmazát, ahol $E_r = \{1,\dots,r\}$. Minden $s \in S$ az $\alpha_1,\dots,\alpha_m$ vektorok egy lehetséges elhelyezését adja a β vektorhalmaz egy m elemü részhalmazán.

S elemszáma: $o(S) = \frac{n!}{(n-m)!}$.

Tetszőleges $s \in S$ -re jelöljön $A(s)$ egy huzalhossz-függvényt, amely a C és D mátrixok elemeinek függvénye.
Az alapprobléma: olyan s^* keresése, amelyre $A(s^*) = \min\{A(s) | s \in S\}$.
Néhány, a gyakorlatban fontos huzalhossz-függvény tipusra példa:

$$a/ \quad Z(s) = \sum_{i,j=1}^{m} f_{ij}(s) \ ,$$

és ennek speciális esete /különösen fontos a k = 0 és a k = 1 eset/:

$$b/ \quad Z_k(s) = \sum_{1 \le i < j \le m} \ \sum_{u,v=1}^{\ell} c_{iu,jv} \cdot (d_{s(i)u,s(j)v})^k$$

A továbbiakban viszonylag gyenge - a gyakorlatban többnyire jól teljesülő - feltételek fennállása esetén vizsgáljuk a probléma megoldását, amely adott kezdeti elrendezésből kiindulva annak javitását végzi el.

2.2. A huzalhossz-függvény egy széles osztályára vonatkozó algoritmus

Az $U = \{\alpha_{i_1}, \dots, \alpha_{i_t}\}$ halmazt független halmaznak nevezzük, ha az összeköttetés-mátrixban az U halmaznak megfelelő részmátrix a $||0||$mátrix, azaz a vektorhalmaz elempárjai nem tartalmaznak összekötött komponenspárt.
U maximális független halmaz, ha nem létezik olyan független halmaz, amelynek U valódi részhalmaza.

A továbbiakban tekintsük az U halmazt rögzitettnek, és indexhalmazát jelölje $T_1 = \{i_1, \dots, i_t\}$ Definiáljuk a $\Sigma \subset S$ halmazt: $\Sigma = \{s | s \in S, \ s(i) = \sigma_o(i), \text{ ha } i \in E_m \setminus T_1\}$ ahol σ_o. egy rögzitett elrendezés. A Σ halmaz tehát pontosan azokat a leképzéseket tartalmazza, amelyek a rögzitett $\sigma_o(i)$ elhelyezésbe visznek minden U-hoz nem tartozó elemet. /Σ az U és σ_o függvénye./

Célunk olyan $\sigma_1 \in \Sigma$ elrendezés meghatározása, amelyre $A(\sigma_1) = \min\{A(s) | s \in \Sigma\}$.Ha $A(s)$ a/-tipusu, $||f_{ij}||$ szimmetrikus mátrix és $c_{ij} = \sum_{u,v} c_{iujv} = 0$ esetén $f_{ij}(s) = 0$, akkor σ_1 meghatározása a hozzárendelési probléma megoldására vezethető vissza. /A visszavezetés általánosabb esetben is lehetséges, csak igy egyszerübben adódik./

Bontsuk fel a huzalhossz-függvényt a T_1 szerint három tag összegére:

$$A(s) = \sum_{i,j=1}^{m} f_{ij} = A_1(s) + A_2(s) + A_3(s) \quad , \qquad \text{ahol}$$

$$A_1(s) = \sum_{i\in T_1,\, j\in T_1} f_{ij} \;, \quad A_2(s) = \sum_{i\notin T_1,\, j\notin T_1} f_{ij} \;, \quad A_3(s) = \sum_{\substack{i\in T_1 \text{és} j\notin T_1 \\ \text{vagy} i\notin T_1 \text{és} j\in T_1}} f_{ij}$$

Itt nyilvánvalóan

$$A_1(s) = 0, \quad A_2(s) = \sum_{i\notin T_1,\, j\notin T_1} f_{ij}(s) = \sum_{i\notin T_1,\, j\notin T_1} f_{ij}(\sigma_o) = a \text{ konstans és}$$

$$A_3(s) = \sum_{i\in T_1,\, j\notin T_1} \left[f_{ij}(s) + f_{ji}(s)\right] = \sum_{i\in T_1,\, j\notin T_1} \left[f_{ij}(s(i).\sigma_o(j)) + f_{ij}(\sigma_o(j).s(i))\right] =$$

$$= \sum_{i\in T_1} F_{i,s(i)} \;,$$

ahol

$$F_{i,s(i)} = \sum_{j\notin T_1} \left[f_{ij}(s(i).\sigma_o(j)) + f_{ij}(\sigma_o(j).s(i))\right] = 2.\sum_{j\notin T_1} f_{ij}(s(i).\sigma_o(j)) \;.$$

Tehát $s\in\Sigma$ esetén $A(s) = a + \sum_{i\in T_1} F_{i,s(i)}$, és igy

$$A(\sigma_1) = \min\{A(s) \mid s\in\Sigma\} = a + \min_{s\in\Sigma} \sum_{i\in T_1} F_{i,s(i)} = a + \min_{s\in\Sigma} B(s),$$

ahol $B(s) = \sum_{i\in T_1} F_{i,s(i)}$.

A minimális s^* permutációt az $||F_{ij}||$ mátrixból az ismert "magyar módszer" adja (l. később).

Innen σ_1 meghatározása: $\sigma_1(i) = \begin{cases} \sigma_o(i), & \text{ha } i\notin T_1 \\ s^*(i), & \text{ha } i\in T_1 \;. \end{cases}$

Ezzel a feladatot a hozzárendelési probléma megoldására vezettük vissza.

2.3. A hozzárendelési probléma

A kombinatorikus jellegü problémák egy jelentős hányada az un. hozzárendelési problémához vezet. Egy nxn méretü mátrix átlóján a mátrix elemeiből alkotott olyan szám - n-est értünk, amely a mátrix minden sorából és oszlopából pontosan egy elemet tartalmaz. A hozzárendelési probléma matematikai megfogalmazása ezekután igy adható meg:

Az adott $A = ||a_{ij}||$nxn-es /nem-negativ, egész szám elemekből álló/ mátrixban meghatározandó az az átló, amelyben az elemek összege minimális. A megfelelő maximum-probléma erre visszavezethető: az $A' = ||a'_{ij}||$, $a'_{ij} = a - a_{ij}$ mátrixot vizsgáljuk, ahol $a = \max\{a_{ij}\}$

A probléma redukálását eredményezi a következő két megállapitás:

i./ Egy adott hozzárendelési problémából vele ekvivalens problémát nyerünk, ha a mátrix bármely vonalának /sorának vagy oszlopának/ elemeit egyidejüleg tetszőleges u számmal növeljük /u lehet negativ is; az elemek nem-negativ voltának megkötése nem lényeges./

Egy mátrix tetszőleges átlójából választott elemeket független elemeknek nevezzük.

ii. Tétel /Kőnig-Egerváry/: Az A négyzetes mátrix független zérus-elemeinek maximális száma megegyezik az összes 0-elemet lefedő vonalak minimális számával.

Könnyen belátható, hogy a hozzárendelési probléma megoldására az összes lehetséges eset áttekintése nem járható ut, mert nxn-es mátrixnál ez $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ esetet jelent, s ez már n = 10-re is 10^6 nagyságrendü és n-el exponenciálisan nő. Viszonylag nagy n értékekre is jól használható megoldást ad az u.n. magyar módszer.

A magyar módszert a program leirásánál tárgyaljuk részletesen, itt csak megjegyezzük, hogy az algoritmus alapja az i. és ii. megállapitások, melyek közül i. bizonyitása triviális, ii./ tétel bizonyitását illetően pedig az irodalomra hivatkozunk [21].

2.4. Optimális független halmaz választásáról

Az előzőekben egy olyan algoritmust irtunk le, amely adott /rögzitett/ U független halmaz esetén egy, az U elemeinek permutálásával elérhető minimumot keres meg az A(s) huzalhosszfüggvényen /természetesen az összes permutáció előállításához szükségesnél lényegesen kisebb lépésszámban/. Az U halmazra való korlátozást a magyar módszer alkalmazhatósága indokolja, amely a t elemszámu halmazon $t!$ számu lehetséges permutáció helyett csak t^k számu permutációt vizsgál meg a minimumhely megkereséséhez /ahol $k < 5$ konstans: t-től független./

Az egyszerübb tárgyalásmód kedvéért ebben a részben gráfelméleti terminológiát használunk, s ezért az előző pontban tárgyalt modellt is megfogalmazzuk gráfnyelven. /A gráfelmélet alapfogalmai magyar nyelven megtalálhatók pl. [5]-ben/.

A rendszernek megfeleltetett G /X,Y/ gráf ponthalmaza $X = \alpha$ vektorhalmaz és az élhalmaz:

$$Y = \{y = (\alpha_i,\alpha_j) | c_{ij} = \sum_{u,v} c_{iu,jv} > 0\}$$

azaz, azon vektorpárok /kártyapárok/ halmaza, melyek össze vannak huzalozva. Itt c_{ij} az (α_i, α_j) élhez rendelt suly. /Az élein sulyozott gráfokat hálózatnak - network - szokás nevezni./

Problémánk szempontjából <u>optimális</u> az $U \subseteq X$ független ponthalmaz, ha egy, az $X \setminus U$ alaphalmazon rögzitett σ permutáció mellett létezik $s^* \in S$, amelyre $s \in S$ esetén $A_U(s^*) \leq A_{U'}(s)$ ahol U' tetszőleges független ponthalmaz. Az A_U /s/ huzalhosszfüggvényben az U index azt jelöli, hogy az $X \setminus U$ halmazra szoritkozva $s = \sigma$. Ilyen általánosságban nem ismeretes optimális U halmazt adó megoldás; az alábbiakban leirunk egy heurisztikus algoritmust, amely az esetek nagy részére az optimumot, vagy közel az optimumot adja. Ennek alapgondolata az, hogy lehetőleg kis elemszámu U független halmaz esetén minimális legyen a <u>"nem-mozgatott" élek</u> sulyösszege. Tekintsük a ponthalmaz $X = U \cup \Gamma/U/$ particióit, ahol U független halmaz, Γ/U/ azon pontok halmaza, melyek U valamely elemével a G gráf valamely élét alkotják.

Az algoritmus leirása előtt vezessük be a következő két jelölést:

$$c(H,K) = \sum_{P \in H} c(P,Q)$$ ahol H és K halmazok, c(P,Q) az összeköttetés-mátrix (P,Q) elemét jelöli;

$$c(P) = c(P,\Gamma(P)) = \sum_{Q \in \Gamma(P)} c(P,Q) \quad .$$

1. Algoritmus

1./ Képezzük a $c_i = \sum_j c_{ij}$ értékeket; $i = 1,\dots,n$.

2./ Legyen a $c_o = \max_i \{c_i\}$ sulyu $P_o \in X$ pont U eleme, ill. ha több ilyen van, ezek közül egy olyan P_o, amelyre $\max_{Q \in \Gamma(P_o)} c(Q)$ minimális. Képezzük az $X' = X \setminus (\overset{\circ}{\Gamma}(P_o) \cup U)$ halmazt.

3./ Ha $X' \neq \emptyset$, ismételjük meg $X := X'$-re a 2. lépést, egyébként U a keresett független halmaz.

2.5. Adott rendezési reláció szerinti rendezés

Nagy elemszám esetén a leirt algoritmus egy jó hatásfoku rendezést igényel. Ilyen például az alábbiakban megadott algoritmus, amely az elempárok összehasonlítása alapján index szerint monoton növekvő sorrendbe rendezi a halmazt. /Az algoritmus [13] felhasználásával, kis módositással adódott./

2. Algoritmus

1./ Képezzünk az m elemből $\left[\frac{m}{2}\right]$ számu diszjunkt párt, s az összehasonlitásban nagyobbnak bizonyult elemekre ismételjük meg az algoritmus 1. pontját /kivéve, ha $\left[\frac{m}{2}\right] = 2$, amikor az összehasonlitás a sorrendet is megadja./

2./ Az összehasonlitásokban kisebbnek bizonyult elemeket ugy illesztjük be egy már meglévő /a megfelelő nagyobb pártól jobbra levő/ k hosszúságu sorozatba, hogy a sorozat /egyik/ középső elemével összehasonlitva, az eredménytől függően a megfelelő félsorozatban ismételjük ezt az összehasonlitást, amig megtaláljuk az elem helyét a sorozatban.

3./ A beillesztendő elemek vizsgálatsorrendjére vonatkozó stratégia az, hogy a legkevesebb összehasonlitást igénylő elemeket illesztjük be, ezen belül az azonos számu összehasonlitást igénylők közül a leghosszabb sorba beillesztendővel kezdjük.

A maximális szükséges összehasonlitás- száma a következő adódik:

$$V(1)=0;\; V(2)=1;\; V(m)=\left[\frac{m}{2}\right] + V\left(\left[\frac{m}{2}\right]\right) + \sum_{i=2}^{\left[\frac{m+1}{2}\right]} W(i),$$ ahol W(i) a

$$\log_2(3i+1) - 1 \le W(i) < \log_2(3i-1)$$

egyenlőtlenség-pár egyetlen megoldása /könnyen belátható, hogy pontosan egyetlen megoldás létezik minden $i \geq 2$ egész számra/.

2.6. Kezdeti alkatrész-elrendezés konstruálása

Az eddigiekben nem volt szó a kiindulási elrendezés megadásáról, pedig az ott definiált értelemben optimális elrendezés eléréséhez jó kiindulási elrendezésre van szükség. Az alábbiakban leírunk egy induló-elrendezést megadó algoritmust, amely figyelembe veszi azt a szempontot, hogy a sok más kártyával összekötött kártya a szomszédaihoz közel, optimálisan legyen elhelyezve, másrészt viszonylag kis időráforditással akarjuk ezt elérni. /Ezért az elhelyezésnél csak "shift"-elést engedünk meg, a pontosabb elrendezés a főprogram feladata./

3. Algoritmus

1./ Képezzük a $c_i = \sum_j c_{ij}$ értékeket $i = 1,\dots,m$.
/Ez a lépés azonos a 1. Algoritmus 1. lépésével, ezért csak egyszer kell végrehajtani és tárolni./

2./ Helyezzük el a β lehetséges helyek "közepén" a $c_0 = \max \{c_i\}$ sulyu pontok egyikét, legyen az P_0. /A "közepén" kifejezést az általános esetben nem definiáljuk pontosan; az előzőekben definiált $d_{ij} = \sum_{u,v} d_{iu,jv}$ mérték esetén nevezzük pl. az i_0-dik pontot /α_{i_0}/ középsőnek, ha teljesül rá: $\max_j d_{i_0 j} = \min_i \max_j d_{ij}$/.

3./ Helyezzük P_0 ponthoz (egyik) legközelebbi pontba a $\Gamma(P_0)$ halmaz egy olyan Q_0 elemét, amelyre teljesül: $c(Q_0) = \max_{Q \in \Gamma(P_0)} c(Q)$.

4./ Ha már elrendeztünk q számu elemet, $q+1$-dik elemként válasszunk egy olyan R_0 elemet, amelyre teljesül /$q = 2,3,\dots,m-1$/:

$$c(\Omega_q \cap \Gamma(R_0), R_0) = \max_{R \in X \setminus \Omega_q} c(\Omega_q \cap \Gamma(R), R).$$

/Itt Ω_q az elrendezett q számu elem halmazát jelöli./
Az $R_0 = P_{q+1}$ elemet a lehetséges helyek közül egy olyan helyre tegyük, hogy a meglevő kártyákra nézve az A /s/ huzalhossz-függvény minimális értéket vegyen fel.

A következő lemma triviálisan adódik, az algoritmus tényleges kivitelezésénél azonban igen hasznos lehet a figyelembevétele.

Lemma: Minden lehetséges q értékre fennáll:

$$0 < \max_{R\in X\setminus\Omega_q} c(\Omega_q \cap I(R), R) \leq \max_{R\in X\setminus\Omega_q} c(R) \leq c_o .$$

3. A PROGRAM LEIRÁSA

A huzalozási probléma megoldását adó programrendszer három részből áll. Az első rész egy U független kártyahalmazt állit elő, a második részben az U halmazhoz tartozó távolságmátrixot képezzük. A távolságmátrix elemei a tényleges távolságok helyett más tulajdonságot is kifejezhetnek, igy a legkülönbözőbb feltételeket beépithetjük a programba. A harmadik rész lényegében a magyar módszer, s ez a többitől különválasztva is rendelkezésre áll, más, különféle permutációs jellegü /diszkrét/ optimumprobléma megoldására. /Neve= ETB1./

Az első részben a C mátrix olyan speciális realizálását adjuk, amely alapján a legkevesebb kártyával összekötött kártya választásával kezdi a független halmaz felépitését.

A kezdeti elrendezés kódját /szintén ALGOL/ Bagyinszki Jánosné készitette. /A programnév: EBB1./ Ezt különválasztva tároljuk, mert igy az ETB2 kód egy már meglévő /manuálisan elkészitett/ elrendezés javitására is használható, másrészt a továbbfejlesztéshez ez a forma előnyösebb.

A program adatai a következők:

a program neve:	ETB2
a program nyelve:	ALGOL /ICT - 1905/
a program helyfoglalása:	19200 szó
bemenet:	lyukszalag
eredmény:	sornyomtatón
elkészült:	1969 dec. /MTA-KFKI/
programozó:	Király Péterné

A futási időt nem adtuk meg, mert az lényegesen függ az adatoktól és széles skálán mozoghat.
Végül megjegyezzük, hogy a programrendszer minimális változtatással felhasználható integrált áramköri elemekből felépülő nyomtatott áramkörös kártyák tervezésének első fázisában /optimális alkatrészbeültetés [7]/. Ezt a változatot is elkészitettük.
Az egyes részek blokkdiagramját a Függelékben adjuk meg.

I. Független kártyahalmaz kiválasztása

Legyen a kártyák száma N és jelölje a kártyákat $K_1, K_2, \dots, K_N$. Rendeljünk minden K_i kártyához egy l_i számot, a K_i kártyával összekötött kártyák számát. Jelölje a K_i kártyával összekötött kártyákat $K_{j_1}, \dots, K_{j_{l_i}}$ és legyen $t_i = l_{j_1} + l_{j_2} + \dots + l_{j_{l_i}}$

1'. Algoritmus

Állitsuk elő a $K_1, K_2, \dots, K_N$ kártyáknak egy olyan K_{i_1}, $K_{i_2}, \dots, K_{i_N}$ permutációját, amelyre fennáll:

$$i_r < i_s, \text{ ha } 1./\ l_{i_r} < l_{i_s}, \text{ vagy}$$

$$2./\ l_{i_r} = l_{i_s} \text{ és } t_{i_r} \geq t_{i_s}$$

/A $t_{i_r} = t_{i_s}$ megengedésével a további vizsgálatok feleslegesen bonyolitó hatásának elkerülése a célunk; ez az egyenlőség ritkán fordul elő, másrészt a rendezésben azt tükrözi, hogy e kártyák sorrendje közömbös számunkra./

1. Algoritmus

Az U független halmaz elemeit a $K_{i_1}, K_{i_2}, \dots, K_{i_N}$ permutáció szerint választjuk ki, a következő módon:

1./ Kiválasztjuk az első /K_{i_1}/ elemet.
2./ Ha már r számu elemet kiválasztottunk, U r+1-edik elemének a soron következő elem választása után töröljük a megmaradó halmazból azokat az elemeket, amelyekkel ez össze van kötve.
3./ Az eljárás véget ér, ha a "választék-halmaz" üres.

A programrendszer bemenete a következő:

$$N,\ h,\ g,\ Q,\ q,\ E,\ F,\ \alpha,\ \beta$$

ahol	N	egész,	a kártyák száma
	Q	egész,	az egy dobozban elhelyezhető kártyák max. száma
	q	egész,	egy kártyán levő "lábak" száma
	g	valós,	két szomszédos kártyahely távolsága
	h	valós,	egy kártya két szomszédos lába közötti távolság
	E	valós,	vizszintes irányban szomszéd dobozpár távolsága*
	F	valós,	függőleges irányban szomszéd dobozpár távolsága
	α	egész,	a kártyák közötti összeköttetéseket adja meg.
	β	egész,	a kártyalábak közötti összeköttetéseket adja meg.

(*"Távolság" alatt középpontok közötti távolság értendő, E kivételével, amely a két doboz legközelebbi kártyapárjának távolsága.)

Az α és β paraméterek részletes megadása a megfelelő részprogram leirásánál történik.

A következő adatok szükségesek a független halmaz kiválasztásához:

1./ a kártyák száma /N/

2./ a kártyák közötti összeköttetések

A kártya-összeköttetéseket az adatszalagon a következő formában adjuk meg:

$$\ell_1 \quad k_{11} \quad k_{12} \cdots\cdots\cdots k_{1\ell_1}$$

- - - - - - - - - - - - - - - - - - -

$$\ell_i \quad k_{i1} \quad k_{i2} \cdots\cdots\cdots k_{i\ell_i}$$

- - - - - - - - - - - - - - - - - - -

$$\ell_N \quad k_{N1} \quad k_{N2} \cdots\cdots\cdots k_{N\ell_N}$$

ahol az i-edik sor jelentése a következő:

a K_i kártya össze van kötve a K_{ki_1}, $K_{ki_2}, \ldots, K_{ki_\ell}$ kártyákkal.

A program a következő részekből áll:

1./ Az adatok beolvasása és egy $B = ||B_{i,j}||$ mátrix előállítása, melyre $B_{i,j} = \begin{cases} 1, \text{ ha } K_i \text{ össze van kötve } K_j\text{-vel} \\ 0, \text{ ha nincs összekötve } K_j\text{-vel} \end{cases}$

2./ A kártyák elrendezése ℓ szerint egy A (NxN-es) mátrixban a következőképpen:
Legyenek $K_{j1}, K_{j2}, \ldots\ldots K_{js}$ olyan kártyák, amelyekre $l_{j1} = l_{j2} = \ldots = l_{js} = i$. Akkor az A mátrix i-edik sora a következő lesz:

$$j_1, j_2, \ldots\ldots j_s, 0, \ldots, 0$$

3./ t_i előállitása $/i = 1,2,\ldots,N/$ az A mátrix soraiban az elemek átrendezése t_i csökkenő sorrendje szerint.

4./ A független kártyák /D halmaz/ kiválasztása az A mátrix alapján.

5./ A D halmaz elemeinek elrendezése növekvő sorrendben.

II. Távolságmátrix előállitása

A program második része az első részben kiválasztott független halmazhoz $/D_1, D_2, \ldots, D_L/$ tartozó távolságmátrixot állitja elő.

A W /L x L/-es távolságmátrix $W_{Di, Dj}$ elemét ugy kapjuk, hogy a K_{Di} kártyát a K_{Dj} kártya helyére téve kiszámitjuk a K_{Di} kártya bekötéséhez szükséges huzalhosszat. Ehhez ismernünk kell bizonyos paraméterek értékét /pl. kártyák közötti távolság, lábak száma, lábak közötti távolság stb./ és a lábak közötti összes összeköttetést. Az összeköttetéseket az adatszalagon a következő formában adjuk meg:

i. az első adat az aktuális kártya sorszáma negativ előjellel,

ii. a második adat az aktuális láb sorszáma,

iii. az utánuk következő adatpárok az aktuális lábbal összekötött kártyát, ill. lábat jelölik.

$$-1, r_1, v^1_{11}, v^1_{11}, v^1_{12}, v^1_{12}, \ldots, v^1_{1p^1_1}, v^1_{1p^1_1}$$

$$-1, r_2, v^2_{11}, v^2_{11}, v^2_{12}, v^1_{12}, \ldots, v^2_{1p^2_1}, v^2_{1p^2_1}$$

- -

- -

$$-N_1, r_{S_N}, v^{S_N}_{N1}, v^{S_N}_{N1}, v^{S_N}_{N2}, v^{S_N}_{N2}, \ldots, v^{S_N}_{Np_N}, v^{S_N}_{Np_N} \; .$$

Természetesen a távolságmátrix elemeinek kiszámitása függ a konkrét feladattól. A programnak ezt a részét a blokkdiagramban a $\boxed{W\ /m,\ p/\ \text{kiszámitása}}$ blokk jelzi.

III. Magyar módszer

A program harmadik része a hozzárendelési probléma megoldása a W távolságmátrixra a magyar módszer segitségével. Ennek a programrésznek már nincs szüksége uj adatra.

Az algoritmus a következő:

α. 1./ A W mátrix minden sorában kiválasztjuk a sor legkisebb elemét és ezt levonjuk a sor összes eleméből.

2./ Az uj mátrix minden oszlopában kiválasztjuk az oszlop legkisebb elemét, és azt levonjuk az oszlop összes eleméből.

Az 1 és 2. lépés után a mátrix minden sorában és oszlopában van legalább egy 0.

β. Az igy keletkezett 0-kból egy maximális független rendszert határozunk meg lefedő vonalak segitségével.

1./ A mátrix első sorától kezdve minden sorban megkeressük az első, még le nem fedett 0 elemet. Ha találunk ilyet, ✱-gal megjelöljük és oszlopát lefedjük.

2./ Ha már minden 0-t lefedtünk, a keresett maximális független rendszert a ✱-os 0-k adják. Ha van le nem fedett 0, akkor két esetet különböztetünk meg:

a./ A le nem fedett 0 sorában van ✱-gal jelzett 0. Ilyen esetben a le nem fedett 0-t ✱✱-gal megjelöljük, a ✱-os 0 oszlop lefedését megszüntetjük, és helyette a sort fedjük le. Utána visszatérünk β.2-re.

b./ A le nem fedett 0 sorában nincs ✱-os 0. Akkor az oszlopában biztosan van olyan ✱os 0, amely sorral van lefedve. Ezért a le nem fedett 0-ból kiindulva lépcsőzetesen haladunk a lefedő oszlopokon és sorokon, a X-os és XX-os O-kon keresztül mindaddig, amig egy olyan XX-os O-hoz nem jutunk, amelynek az oszlopában nincs X-os O. Az érintett X-os O-kat megszüntetjük, a kiinduló O-t és a XX-os O-kat X-os O-vá alakitjuk, és oszloppal lefedjük őket. Utána visszatérünk β.2-re.

γ.1./ Ha a kiválasztott független O-k száma = L, az algoritmus véget ér. Az eredmény a következő alakban jelenik meg:

$$
\begin{array}{ccc}
i_1 & & j_1 \\
i_2 & & j_2 \\
- - - - - & & - - - - - \\
- - - - - & & - - - - - \\
i_L & & j_L
\end{array}
$$

ahol az i-k a kártya sorszámát, a j-k a kártya uj helyét jelölik.

2./ Ha a kiválasztott független O-k száma kisebb mint L, akkor ujabb O-kat konstruálunk a következő módon:

a./ A mátrix le nem fedett elemei közül kiválasztjuk a legkisebbet, legyen ez p.

b./ A mátrix le nem fedett sorainak elemeiből levonjuk p-t.

c./ A mátrix lefedett oszlopainak elemeihez hozzáadjuk p-t.

Utána visszatérünk β -ra.

4. ÁLTALÁNOSITÁSOK ÉS TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

4.1. Az elméleti rész /2. pont/ tárgyalásánál nem tértünk ki arra az esetre, amikor az S leképezés-halmaz bizonyos elemeit a mellékfeltételek eleve kizárják. /A probléma megfogalmazásában ez szerepelt, azonban a választott megoldásba - magyar módszer - nem épitettük be az egyszerűség kedvéért./ Az ilyen irányu általánositásnak elvi akadálya nincs, a gyakorlati kivitelezés nem okoz nehézséget, sőt általában az algoritmus gyorsitását eredményezi. A magyar módszer gyakorlati kivitelezésében ez pl. olyan változtatással érhető el, hogy ha a megjelölt mátrixelemek egy tiltott kombinációja áll elő, a sor-oszlop lefedő vonalak alternáló kereső-sorozatában leállunk, sikertelennek tekintve ezt az ágat. /Itt azt használjuk ki, hogy általában degenerált - tehát bizonyos elemeket fixen hagyó - leképezések a tiltottak/. Megjegyezzük, hogy n-m fiktiv elem felvételével leképezés helyett közvetlenül permutációs probléma adódik, azonban a mátrix mérete megnő.

4.2. Egy másik tipusu gyakori feltétel az, hogy bizonyos távolságokra felső korlátot irunk elő, vagy esetleg azokat pontosan előirjuk. Hangsulyozni kivánjuk, hogy a problémakör matematikai jellegét az adja, hogy itt diszkrét halmazon értelmezett függvények szélsőérték-helyeit és szélsőértékeit keressük, s igy a matematikai analizis elemeiből ismert eljárás /több-változós függvények feltételes szélsőértékeinek meghatározása Lagrange-féle multiplikációs eljárással/ nem alkalmazható minden további nélkül, mert a függvények nem folytonosak. Mégis, vegyük észre, hogy az alapelv átmenthető olyan módon, hogy a λ multiplikátorokat elég nagyra választva /pl. két tetszőlegesen kiválasztott permutációhoz tartozó huzalhossz-függvények közül a nagyobbik értéke legyen λ / a feltételek a távolságmátrixba beépithetők, és ezzel a 2. részben tárgyalt - és megoldott - esetre vezettük vissza a problémát.

4.3. Ugyanitt jegyezzük meg: a szélsőérték és a függvény monotonitása definiciójának közvetlen következménye, hogy ha $\mu/x/$ az argumentumának szigoruan monoton növekvő függvénye, akkor tetszőleges $A/s/$ huzalhossz-függvény és a $\mu(A/s/)$ függvény szélsőértékhelyei azonosak.
Továbbá, minimumhely keresése maximumhely keresésére vezethető vissza $\nu(A/s/)$ transzformációval, ahol $\nu/x/$ tetszőleges szigoruan monoton csökkenő függvény.

4.4. A lehetséges megoldások közül a 2. részben egy olyan módszert választottunk, amelynek alapja egy /maximális/ független halmaz

kiválasztása és az ezen alkalmazott magyar módszer. /Adott kezdeti elrendezés esetén./ Egy, a gyakorlat szempontjából is lényeges általánositáshoz jutunk, ha a független halmaz helyett a független halmazok egy véges rendszerét választjuk ki. Egy rögzitett - és valamely megadott sorrendben felvett - független halmazrendszer esetén alkalmazzuk az alapalgoritmust a független halmazsorozat első elemére, majd a kapott lokális minimumot adó elrendezést kezdeti elrendezésnek tekintve, a sorozat következő elemével megismételjük az algoritmust.
A sorozat elemeit ciklikusan cserélve /az utolsó után ismét az első elemet vesszük következő független halmazként/, az eljárást addig folytatjuk, amig a sorozat elemszámával azonos hosszuságu olyan iteráció-sor adódik, amely a huzalhosszat változatlanul hagyta. Igy a huzalhossz-függvényértékek egy monoton csökkenő sorozatát nyerjük.

Felmerül a kérdés, hogy milyen hosszu ez a sorozat, vagyis hány iterációs lépésben érhető el a minimumhely? Ez általánosságban nyilvánvalóan lényegesen függ a vizsgált hálózat topológiai strukturájától is, tehát minden lehetséges hálózatra külön kellene megadni, s ezért ez érdektelen. Érdekes, de nagyon nehéznek tünő probléma a topológiai strukturától való függés leirása, vagy legalábbis bizonyos korlátok megadása.

A kérdéskör ésszerübb megközelitése a sztochasztikus jellemzés, melynek modelljét az alábbiakban vázoljuk. Tételezzünk fel egyenletes eloszlást az n-szögpontu nem-izomorf erdők halmazán. Kérdés: véletlenszerüen kiválasztva egy elemet /egy hálózatot/, mi az iterációk számának várható értéke.

A kérdésfeltevésnek akkor van értelme, ha egyértelmüen rögzitjük (pl. algoritmikusan) az algoritmus hatékonyságának paramétereit, vagyis

a./ a kezdeti elhelyezést /U_1 esetén a bemeneti elrendezés/

b./ a felhasznált $U = \{U_1, U_2, \ldots, U_h\}$ sorozatot. /Ez utóbbi a független halmazrendszer választását és elemeinek sorrendjét irja elő./

Optimum-elvünknek megfelelően olyan U sorozatot keresünk, amelyre alkalmasan választott /adott/ kezdeti elhelyezés esetén az algoritmus hatékonyságától /iterációk száma/ és az elért lokális minimum /huzalhossz-függvényen/ értékétől függő célfüggvény minimumot vesz fel. /E célfüggvény lehet pl. a két változó homogén lineáris függvénye, alkalmas pozitiv együtthatópárral./

Megjegyezzük, hogy az /a./ - /b./ alatt leirt paraméterek a determinisztikus esetben is ugyanugy fellépnek.

4.5. Hasznos lenne, ha olyan U halmazrendszert találnánk, amelyre az elemek egyesitése a teljes kártyakészletet kiadja, azaz $\bigcup_{k=1}^{h} U_k = \alpha$. Továbbá megkivánjuk U-tól, hogy benne tetszőleges két, α_x, α_y elemhez tartozzon egy lánc abban az értelemben, hogy alkalmas $\{U_{z_1}, U_{z_2},\ldots,U_{z_w}\}$ részsorozatra létezik az α-beli elemek egy

$$\alpha_x = \alpha_{z_1}, \alpha_{z_2},\ldots,\alpha_{z_w}, \alpha_{z_{w+1}} = \alpha_y$$

lánca, ahol $\alpha_{z_e}, \alpha_{z_{e+1}} \in U_{z_e}$ $/e = 1,\ldots, w./$

Az a "sejtésünk", hogy a vázolt és az ezekhez "hasonló" optimumproblémák megoldása szoros kapcsolatba hozható Erdős Pál azon sejtésével, melynek bizonyitását adja a következő:

<u>Tétel</u> /Hajnal - Szemerédi/: Legyenek $\ell \geq 1$, $n = \ell \cdot m + r$ egész számok és $0 \leq r < \ell$ teljesüljön. Jelölje a G véges gráf P pontjának fokát /a ponthoz csatlakozó élek számát/ $\rho(P)$ és legyen a gráf pontjainak száma $|V(G)| = n$. Ha $\max_{P\in V(G)} \rho(P) \leq \ell$, akkor létezik a V(G) ponthalmaz egy olyan $V(G) = \bigcup_{i=1}^{\ell} V_i$ particiója, amelyre $V_i/i= 1,\ldots,\ell/$ nem tartalmaz G-beli élnek megfelelő pontpárt és teljesül rá:

$$|V_i| = \begin{cases} m+1, & \text{ha} \quad 1 \leq i \leq r \\ m, & \text{ha} \quad r < i \leq \ell \end{cases}$$

Ellenőrizhető, hogy a tétel feltételei esetünkben teljesülnek.

A megadott particióban szereplő V_i halmazok nyilvánvalóan független halmazrendszert alkotnak, azonban általában ez nem maximális rendszer /maximális rendszerre U_i és U_j nem feltétlenül diszjunktak/.
Sejtésünk arra vonatkozik, hogy a tételben szereplő particióból "egyszerüen megkonstruálható" az a maximális független halmazrendszer, amely optimális. /Bizonyos részletek tisztázatlansága miatt ezzel itt nem foglalkozunk; egy későbbi dolgozatban szeretnénk visszatérni rá./

Végül megjegyezzük, hogy a probléma feltételeinek elemzéséből adódik egy általánosabb megközelitési lehetőség. Ezt az alábbiakban vázoljuk.

Legyen α^* a kártyák egy halmaza és σ_0^* egy kezdeti elhelyezés. Keresendő olyan $\sigma_1^* \in \Sigma_1^*$ elrendezés, amelyre a huzalhosszfüggvény minimális és Σ_1^* azon elhelyezések halmaza, amelyre:

a./ minden $\alpha \setminus \alpha^*$ -beli elem fixpont

b./ α^* tetszőleges elempárja közti teljes huzalhossz konstans, azaz invariáns a megengedett leképezésekre.

Ha α^* független halmaz, speciális esetként az alapalgoritmus adódik. /Természetesen nincs akadálya az α^* sorozatra való általánositásnak sem./

4.6. Az eddig elmondott algoritmusok mindegyikét speciális esetként magában foglalja a következőkben körvonalazott algoritmus. Az egyszerüség kedvéért az alapalgoritmusra /2.rész/ vonatkozóan irjuk le.

Mig az alapalgoritmus kiindulási állapota feltételezi a galvanikusan összekapcsolt párok - vagyis a C mátrix - lerögzitését, itt csak az elvi áramköri szintet tekintjük rögzitettnek, vagyis a galvanikusan ekvivalens pontok halmazát adjuk meg.

Jelölje π azt az operátort, melynek hatása a hálózatra az, hogy tetszőleges állapotából az alapalgoritmussal /permutációsan/ elérhető minimális huzalhossz-értékkel rendelkező állapotba viszi a hálózatot. /Röviden: π operátorral való szorzás az alapalgoritmus alkalmazását jelenti./ Hasonlóan, τ jelölje azt az operátort, amely a rendszert bármely állapotából a galvanikusan ekvivalens áthuzalozással elérhető minimális huzalhossz-értékkel rendelkező állapotba viszi át.
Nyilvánvaló, hogy a minimális összhuzalhossz-esetnek megfelelő gráf erdő /speciális esetben fa/. A τ operátornak megfelelő algoritmus leirását és az ALGOL publikációs nyelven megadott programot a következő, 4.7. rész tartalmazza./

A π és τ operátorok definiciójából következik, hogy idempotens tulajdonságuak, vagyis fennáll:

$$\tau^2 = \tau, \qquad \pi^2 = \pi \ .$$

/Ennek jelentése pl. π -re: az alapalgoritmus eredményét használva az alapalgoritmus kezdeti elrendezésének, π másodszori alkalmazása változatlanul hagyja az elrendezést./

Megjegyezzük, hogy e két operátor nem felcserélhető, vagyis általában $\tau\pi \neq \pi\tau$.
Ezért - célszerüen τ operációval kezdve - a javasolt algoritmus a $\tau\pi\ \tau\pi\ldots$ tipusu operátor-szorzatnak megfelelő iteráció lesz, amely véges hosszuságu. Ugyanis az egyes szeleteknek megfelelő huzalhossz-érték sorozata monoton csökkenő /pontosabban, $\tau\pi$-lépésenként szigoruan monoton csökkenő/, nemnegativ elemekkel /tehát alulról korlátos/ és létezik egy pozitiv minimális csökkenés. Az iterációból ugy adódik az algoritmus, hogy definiáljuk a befejező lépést. Legyen a $\tau\pi\ \tau\pi\ldots$ végtelen szorzat valamely k-elemü Λ_k kezdőszelete olyan tulajdonságu, hogy

$$\Lambda_k\cdot\lambda_k = \Lambda_k, \quad \text{ahol} \quad \lambda_k = \begin{matrix} \tau\pi & \text{ha } k \text{ páros} \\ \pi\tau & \text{ha } k \text{ páratlan.} \end{matrix}$$

Az algoritmus k_o lépésben befejeződik, ha egyetlen $k' < k_o$ esetén sem teljesül az előbbi egyenlőség, és k_o-ra teljesül.

/Ez azt fejezi ki, hogy ha a rendszer állapotát a két algoritmus egyikével sem tudjuk megváltoztatni, akkor azok tetszőleges számu egymás utáni alkalmazása sem változtat azon./

Kérdés, mit tudunk mondani a k_o iterációszám értékéről, illetve paritásáról? Ezek nyilvánvalóan /a 4.4 pontban elmondottakhoz hasonlóan/ függenek a hálózat topológiai strukturájától, a kezdeti elhelyezéstől, a független halmaz /-rendszer/ választásától. Itt is érdekesebb a sztochasztikus jellemzés, továbbá adott topológia esetén a maximális k_o érték.

Anélkül, hogy kitérnénk a részletekre /az algebrai struktura vizsgálatára/, befejezésül megjegyezzük, hogy a $\{\Lambda_k\}$ operátorhalmaz algebrai értelemben véges félcsoport - mint az nyilvánvaló a definicióból.

<u>Megjegyzés:</u> Az ebben a pontban elmondottak változatlanul érvényben maradnak, ha τ helyett a τ' operátort definiáljuk, amelyre $\tau'=\tau.\pi'$, ahol π' a logikailag ekvivalens elemek /pl. ÉS-kapu bemenetei/ permutációin felvett minimumot adó /π-tipusu/ operátor. A $\{\pi'\}$ halmaz tehát a logikai szint automorfizmus-csoportja.

4.7. Ebben a részben a minimális kifeszitő /rész/-fát adó algoritmus leirását adjuk. Az algoritmus az irodalomból ismert /l.pl. [20], [3], [7], vagy Communications of the ACM 13 N.10 /1970./ 621, Algorithm. 399/, csak a teljesség kedvéért irjuk le; megadjuk az ALGOL publikációs nyelven irt programot is.

Minimális fa- adott ponthalmazzal és távolságmátrixszal

Adott a $\{P_1,\ldots,P_n\}$ ponthalmaz, valamint a $T = ||t_{ij}||$ nxn-es /távolság/-mátrix, ahol $t_{ij} \geq 0$ a $/P_i, P_j/$ pontpárhoz rendelt suly /a P_i és P_j pontok távolsága/. Konstruálandó olyan összefüggő gráf, melynek ponthalmaza az adott ponthalmaz, és az éleihez rendelt sulyok összege /a távolságösszeg/ minimális.

A feltételeket teljesitő fát adó algoritmus leirása előtt néhány fontos megjegyzést teszünk. /Hogy ez a gráf fa, az könnyen belátható./

Mindenekelőtt megjegyezzük, hogy a T mátrix meglehetősen tág határok között a pontpárok valamely tulajdonságát kifejező mennyiség. A mátrix elemei numerikus értékek, lehetnek speciálisan a pontpárok tényleges távolságai is, vagy azok egy szigoruan monoton növekvő, egyébként tetszőleges függvénye. Lehetséges azonban olyan általánositás is, amikor T nem szimmetrikus mátrix /$t_{ij} = t_{ji}$ nem teljesül minden indexpárra/, követelmény azonban, hogy $t_{ii} = 0$ legyen minden i-re. A nem szimmetrikus T mátrix gráfmegfelelője: irányitott kört nemtartalmazó, összefüggő, többszörös élek nélküli és hurokmentes gráf. /Két pont között ellentétes irányitással 2 él megengedett!/ Végül az algoritmus általánositható olymódon is, hogy minimális irányitatlan fát eredményezzen valamely irányitással. Fákat, mint komponenseket tartalmazó un. erdőre természetesen "megy" az algoritmus: a komponensekre függetlenül elvégzendő.

A "jó" algoritmus szükségessége nyilvánvaló, hiszen az n szögpontu fák száma n^{n-2} /Cayley tétele/, s ez nagy n-értékekre igen gyorsan nő n-nel. /A "jó" algoritmust az jellemzi, hogy az n^{n-2} számu eset áttekintése helyett egy lényegesen kisebb számu halmazból kiválasztja az optimális megoldást, vagy ezzel ekvivalens lépésszámmal jut az eredményhez. Pl. n^c számu esetet kell áttekinteni, ahol c n-től független konstans./

Az algoritmus leirása

1./ Rendezzük az $N = \binom{n}{2}$ számu élet monoton növekvő sorrendbe.
2./ $i = 1$; az első élet rajzoljuk be.

3./ i értékét növeljük 1-gyel.

Tekintsük az i. élet. A lehetséges 4 eset szerint folytatjuk.

a./ Az uj él 2 végpontjának egyike sem tartozik már meglevő részfa ponthalmazához. Az élet képezzük: uj részfa keletkezik.

b./ Az egyik pont valamely részfa ponthalmazának eleme, a másik nem: az élet képezzük, s ezzel az egyik részfa "megnő".

c./ A két pont már meglévő két különböző részfa ponthalmazának eleme: az élet képezzük, a két részfa összefüggő, egyetlen részfává egyesül.

d./ Mindkét pont ugyanabban a részfában van: nem képezzük az élet.

4./ A bekötött pontok száma <n : /3/-nál folytatjuk.

5./ Az algoritmus befejeződött: a konstruált fa minimális.

Megjegyzések:

1./ Az algoritmus első pontját a 2.5 rész, 2. algoritmus alapján hajtjuk végre /rendezés/.

2./ Az egyenlő hosszuságu élek sorrendje közömbös az eredmény szempontjából, azonban ez gyakran jól felhasználható az algoritmust részprogramként alkalmazó főprogramban, amely érzékeny lehet ezekre a permutációkra is.

3./ Az algoritmusban ténylegesen nem használjuk fel az n-2 leghosszabb élet.

4./ A pontok fokára vonatkozó bizonyos korlátozásokat is figyelembe vehetünk az algoritmusban, erre azonban itt nem térünk ki.

5./ Adott gráf minimális összefüggő részgráfját is meghatározhatjuk a leirt algoritmussal; a nem összekötött élek sulyát végtelennek választjuk. (Az itt következő algoritmus ezt teszi.)

Spanning tree

procedure spanning tree /v, e, I, J, p, T/; value v, e; integer v, e, p, integer array I, J, T; comment. Az eljárás egy adott G = (N, E) v pontu és e élü irányitatlan hurokmentes gráf egy T kifeszitő fáját /nem összefüggő G esetén erdőt/ határozza meg.
$(I[k], J[k]) \in E$, $k = 1,2,\ldots,e$ az $I[1:e]$ és $J[1:e]$ tömbökben.)

A kifeszitő erdő éleit a T [1: v-p] tömb tárolja, ahol p az erdő komponenseinek száma.

```
begin
  integer  i,j,c,n,r;
  integer array V [1:v]
  c: = n: = 0;
  for k: = 1 step 1 until v do V [k]: = 0;
  for k: = 1 step 1 until e do
  begin
    i: = I [k]; j: = I [k];
    if V [i] = 0 then
    begin T [k-n]: = k;
      if V [j] = 0 then V [i]: = V [j]: = c : = c+1
      else
      V [i]: = V [j]
    end
      else if V [j] = 0 then
    begin
      T [k-n]: = k; V [j]: = V [i]
    end
      else if V [i] ≠ V [j] then
    begin
      T [k-n]: = k; i: = V [i]; j: = V [j];
      for r: = 1 step 1 until v do
      if V [r] = j then V [r]: = i
    end graft
      else n: = n+1
  end edge;
    p:= v-e+n
end spanning tree
```

IRODALOMJEGYZÉK

1. Abos I., Bálint L., dr. Csurgay Á., Herpy M., Scsaurszki P.: Számitógépek alkalmazása a harmadik generációs áramkörök konstrukciós tervezésében. TKI-I-70-321-5.

2. Álló G., Csipka L., Sikolya Zs.: Nyomtatott áramköri kártyák automatikus elkészitése. Compcontrol 70 Conference, Miskolc, 1970.

3. Bagyinszki J.: Digitális rendszerek tervezésének Boole módszerei. BME Továbbképző Intézete, 1971. jegyzet.

4. Bagyinszki J., Viszt É.: Digitális rendszerek automatikus logikai tervezése. Számitógéptechnika 68'. Esztergom, 61-88. /Konferencia/

5. Bagyinszki J., Viszt É.: Digitális rendszerek számológépes logikai tervezésének elméleti és gyakorlati kérdései. BME Továbbképző Intézete, 1970. jegyzet.

6. Bagyinszki J.,Viszt É.: Digitális rendszerek logikai tervezéséről, KFKI Közl. 16 /1968/ 289-323.

7. Bagyinszki J.: Integrált áramköröket tartalmazó nyomtatott kártyák optimális fólia-tervezésének egy uj, gráfelméleti módszere. Számitógéptechnika '71, Esztergom /Konferencia/ 80-97

11. Fisk. C.J., Caskey, D.L., West, L.E.: ACCEL: Automated Circuit Card Etching Layout.Proc. IEEE 55/11 /1967/ pp. 1971-1982.

13. Ford, L.R., Johnson, S.M.: A Tournament Problem. Am.Math.Monthly, 66 /1959/ pp. 387-389.

14. Garside: R.G., Nicholson, T.A.J.: Permutation Procedure for the Backboard Wiring Problem. Proc. IEE, Vol.115. No.1. /1968./

15. Hajnal, A; Szemerédi, E.: Proof of a conjecture of P. Erdős. Coloquia Mathematica Societatis János Bolyai, 4. Combinatorial Theory and its Applications, Balatonfüred 1969; 601-623.

16. Hammer, P.L., Rudeanu, Sergiu: Boolean Methods in Operations Research, Springer-Verlag, 1968.

20. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proc. Am. Math. Soc. 7 /1956/ 1., 48-50.

21. Kuhn, H.W.: The Hungarian Method for the Assignment Problem, Naval. Res.Logist. Quart. 2 /1955/ 83-97.

22. Kuhn, H.W.: Variants of the Hungarian Method for Assignment Problems. Naval Res. Logist. Quart. 3 /1956/ 253-258.

24. Loberman, H., Weinberger, A.: Formal Procedures for Connecting Terminals with a Minimum Total Wire Length, Journal ACM 4 /1957/ 428-437.

25. Munkres, J.: Algorithms for the Assignment and Transportation Problems, Journal SIAM 5 /1957/ 32-38.

26. Nicholson, T.A.J: Permutation Procedure for Minimizing the Number of Crossing in a Network, Proc. Inst. Elec. Eng. 115 /1968/ No.1.

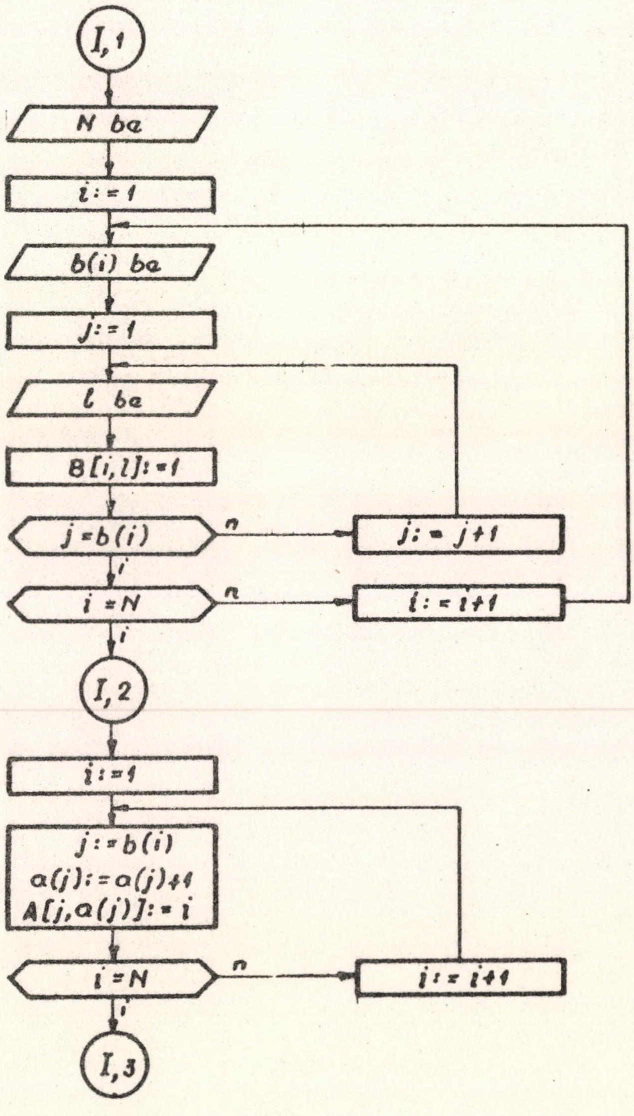
27. Nicholson, T.A.J: A Method for Optimizing Permutation Problems and its Industrial Applications, Combinatorial Mathematics and its Applications /Welsh/ Proc. Conf. Math. Int., Oxford 1969. 201-218.

30. Reiter, S., Sherman, B.: Discrete Optimizing, Journal SIAM 13 /1965/ 864-889.

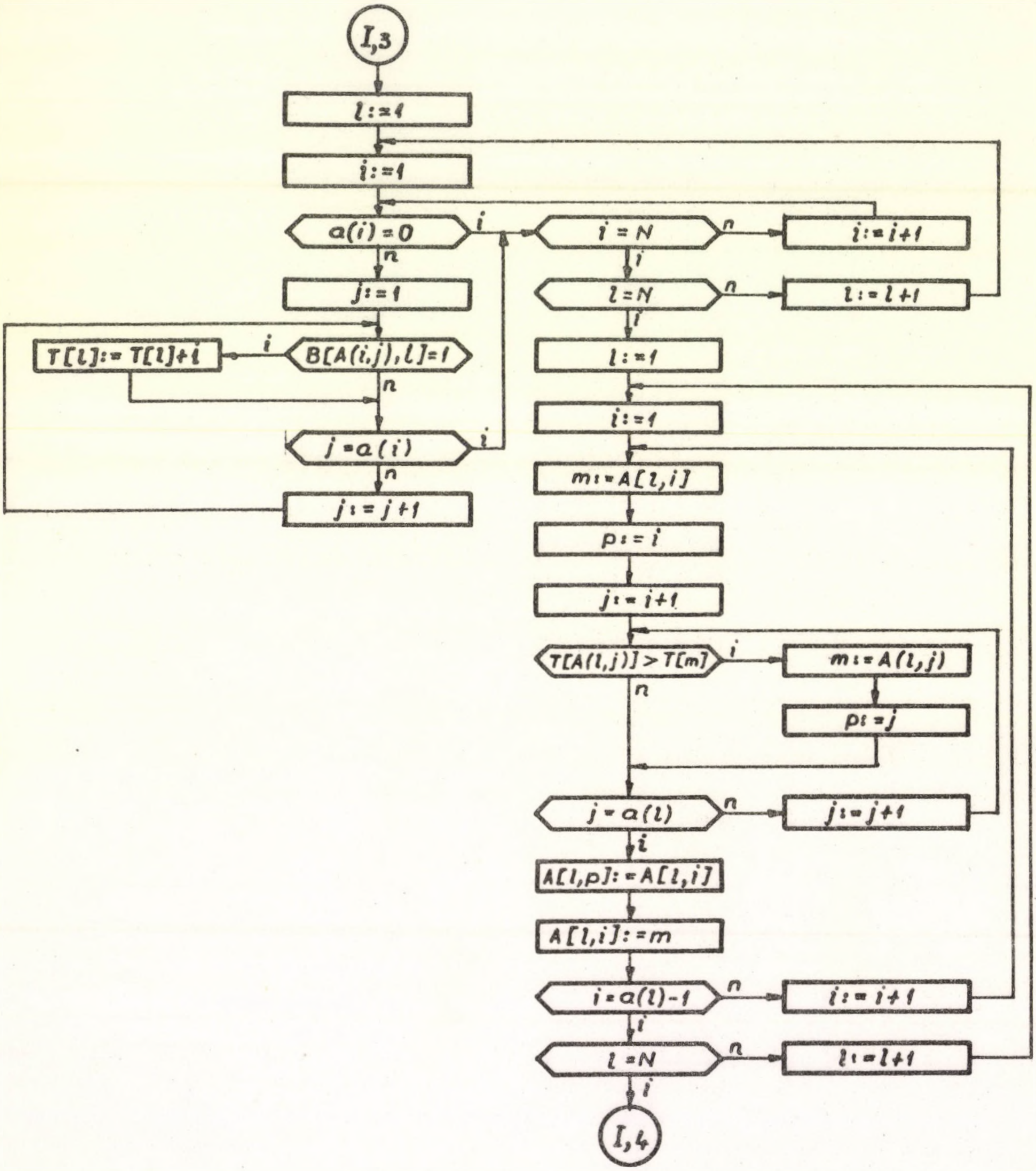
31. Steinberg, L.: The Backboard Wiring Problem: a Placement Algorithm. SIAM Rewiew, Vol. 3. No. 1. Jan. 1961.

A 8, 9, 10, 12, 17, 18, 19, 23, 28, 29, 32, 33 hivatkozások rendre a CAD Computer Aided Design, IEE Conference Publication No.51. Southampton, 1969: Bras, D; Carré, B.A; Cullyer, W.J; Flechter, A.J; Houghton, J; Huttley, N.A; Kaposi, A.A; Leevers, D.F.A; Pezé, F.A; Radley, P.E; Wise, D.J.K; Wood, J. szerzők dolgozatait jelölik.

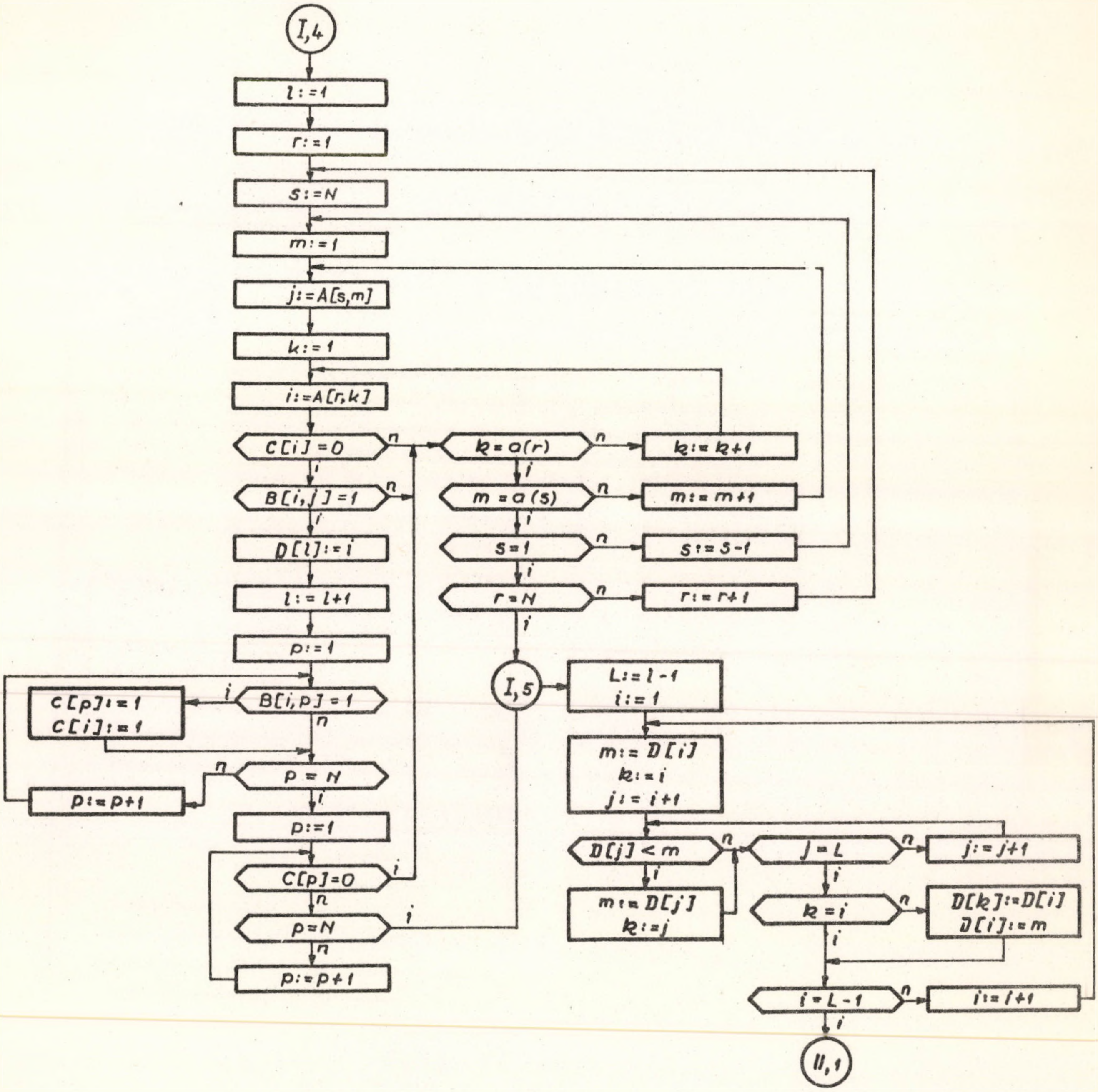
FÜGGELÉK



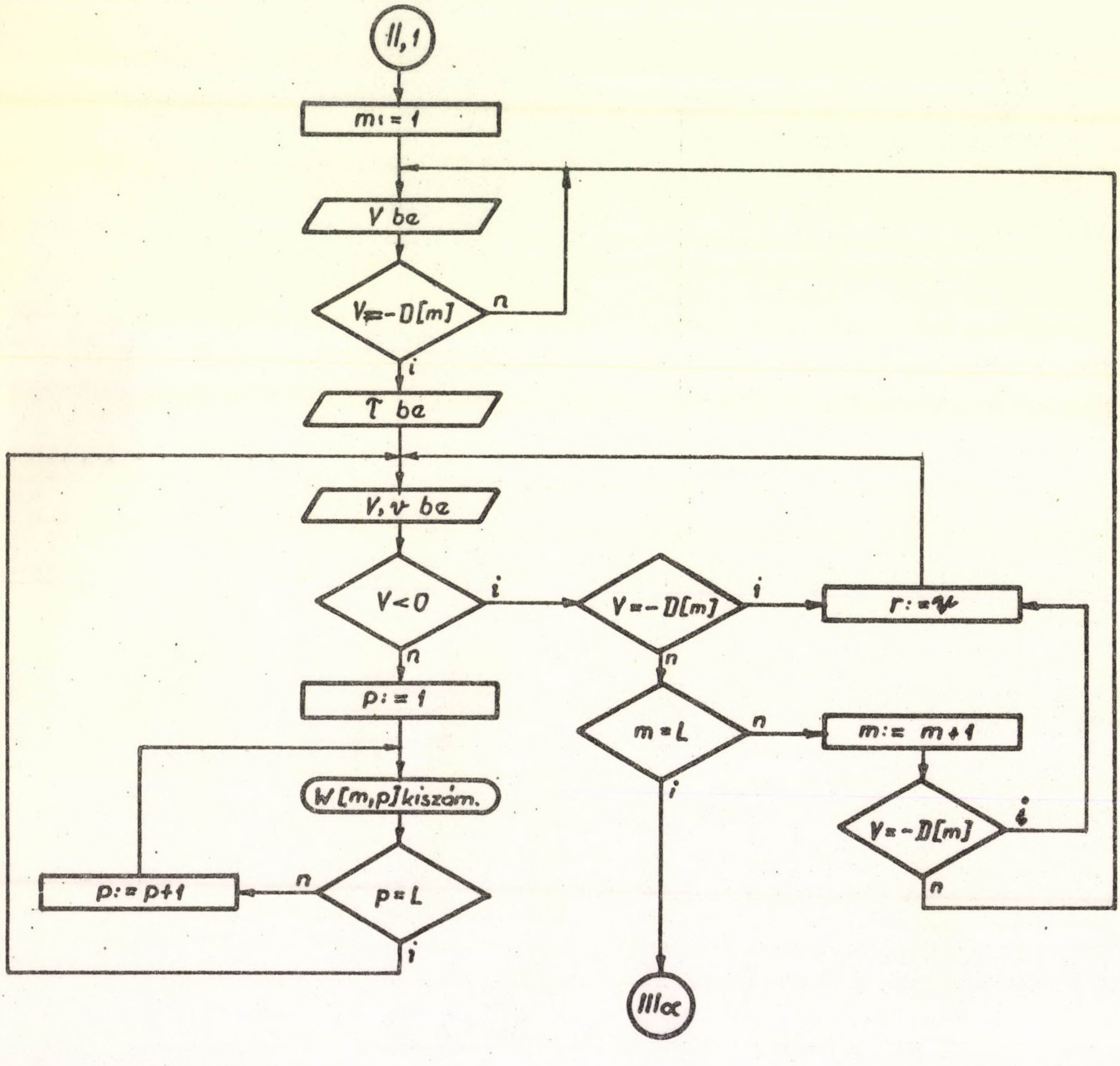
1. ábra


I,3
l:=1
i:=1
a(i)=0
i = N
i:=i+1
l=N
l:=l+1
j:=1
T[l]:=T[l]+1
B[A(i,j),l]=1
j=a(i)
j:=j+1
l:=1
i:=1
m:=A[l,i]
p:=i
j:=i+1
T[A(l,j)]>T[m]
m:=A(l,j)
p:=j
j=a(l)
j:=j+1
A[l,p]:=A[l,i]
A[l,i]:=m
i=a(l)-1
i:=i+1
l=N
l:=l+1
I,4

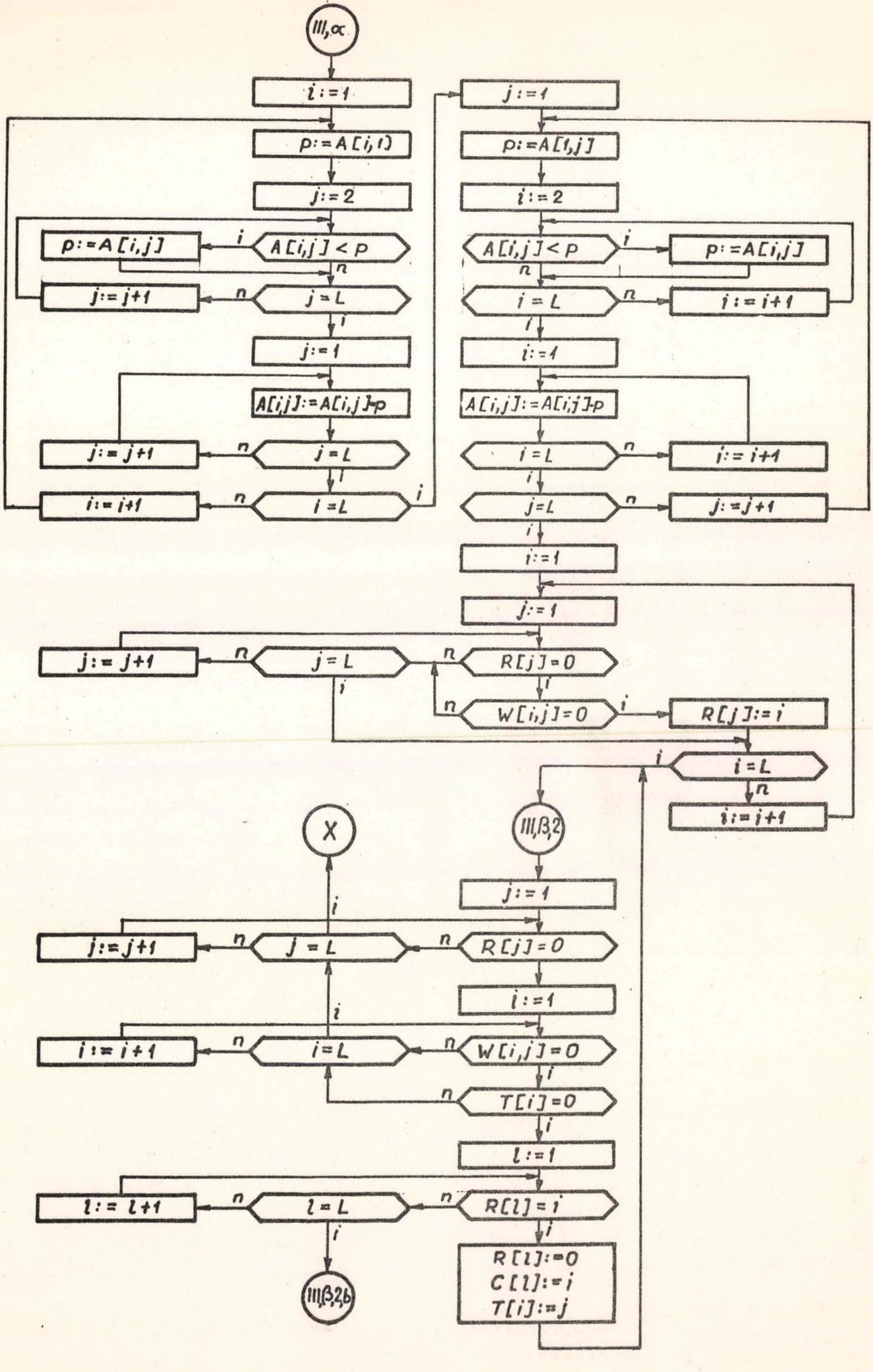
2. ábra


I,4
l:=1
r:=1
s:=N
m:=1
j:=A[s,m]
k:=1
i:=A[r,k]
C[i]=0
B[i,j]=1
D[l]:=i
l:=l+1
p:=1
B[i,p]=1
C[p]:=1
C[i]:=1
p=N
p:=p+1
p:=1
C[p]=0
p=N
p:=p+1
k=a(r)
k:=k+1
m=a(s)
m:=m+1
s=1
s:=s-1
r=N
r:=r+1
I,5
L:=l-1
i:=1
m:=D[i]
k:=i
j:=i+1
D[j]<m
m:=D[j]
k:=j
j=L
j:=j+1
k=i
D[k]:=D[i]
D[i]:=m
i=L-1
i:=i+1
II,1

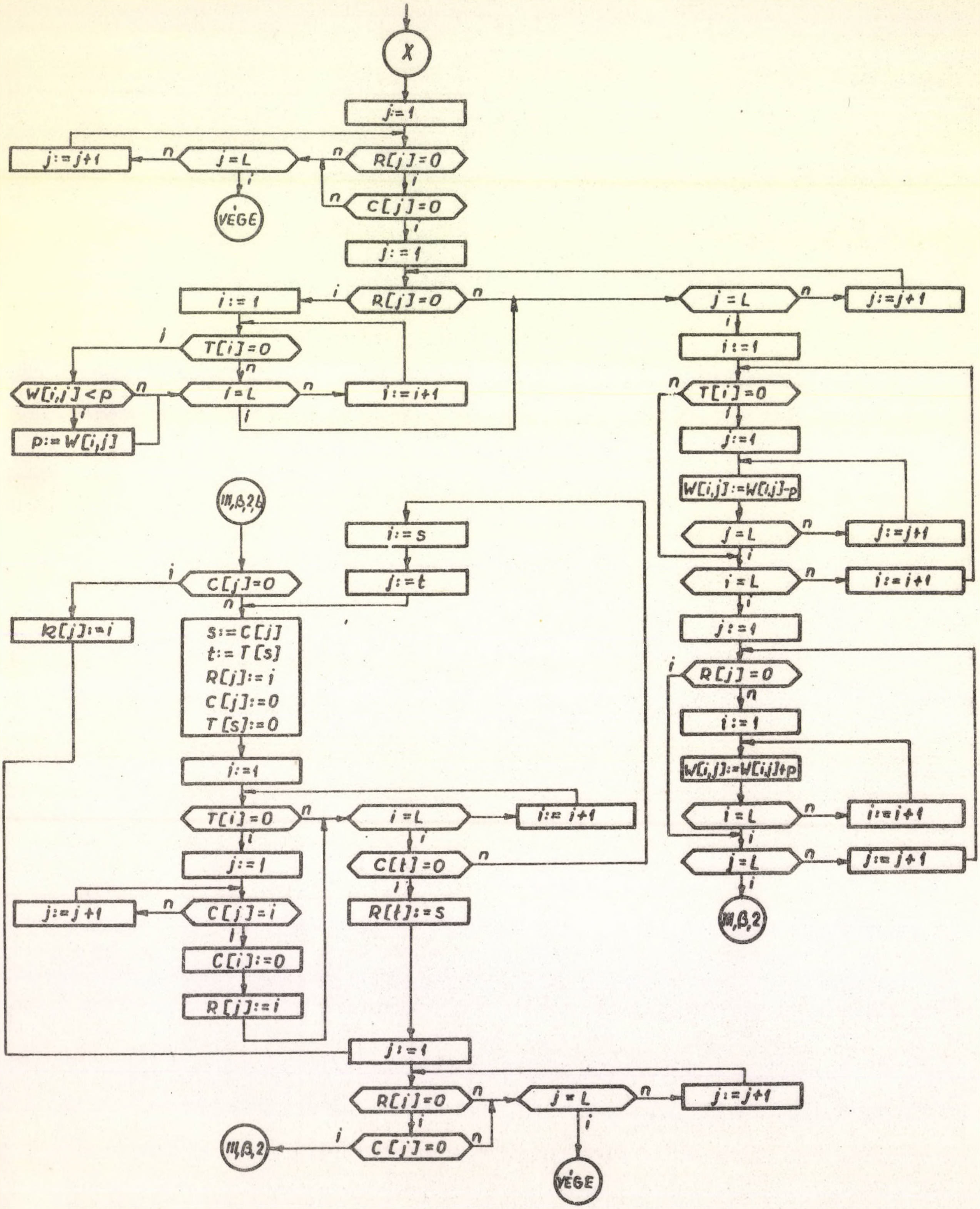
3. ábra


II,1
m:= 1
V be
V=-D[m]
n
i
T be
V, v be
V<0
i
n
V=-D[m]
i
n
r:= v
p:= 1
m=L
n
i
m:= m+1
W[m,p] kiszám.
V=-D[m]
i
n
p:= p+1
n
p=L
i
IIIα

4. ábra


III,α
i:=1
p:=A[i,1]
j:=2
A[i,j]<p
p:=A[i,j]
j=L
j:=j+1
j:=1
A[i,j]:=A[i,j]-p
j=L
j:=j+1
i=L
i:=i+1
j:=1
p:=A[1,j]
i:=2
A[i,j]<p
p:=A[i,j]
i=L
i:=i+1
i:=1
A[i,j]:=A[i,j]-p
i=L
i:=i+1
j=L
j:=j+1
i:=1
j:=1
R[j]=0
j=L
j:=j+1
W[i,j]=0
R[j]:=i
i=L
i:=i+1
X
III,β,2
j:=1
R[j]=0
j=L
j:=j+1
i:=1
W[i,j]=0
i=L
i:=i+1
T[i]=0
l:=1
R[l]=i
l=L
l:=l+1
R[l]:=0
C[l]:=i
T[i]:=j
III,β,2b

5. ábra


X
j:=1
R[j]=0
C[j]=0
j=L
j:=j+1
VÉGE
j:=1
R[j]=0
i:=1
T[i]=0
W[i,j]<p
p:=W[i,j]
i=L
i:=i+1
j=L
j:=j+1
i:=1
T[i]=0
j:=1
W[i,j]:=W[i,j]-p
j=L
j:=j+1
i=L
i:=i+1
j:=1
R[j]=0
i:=1
W[i,j]:=W[i,j]+p
i=L
i:=i+1
j=L
j:=j+1
III,B,2
III,B,2,b
C[j]=0
R[j]:=i
i:=s
j:=t
s:=C[j]
t:=T[s]
R[j]:=i
C[j]:=0
T[s]:=0
i:=1
T[i]=0
i=L
i:= i+1
C[t]=0
R[t]:=s
j:=1
C[j]=i
j:=j+1
C[j]:=0
R[j]:=i
j:=1
R[j]=0
j=L
j:=j+1
C[j]=0
III,B,2
VÉGE

6. ábra

61.929

Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Sándory Mihály, a KFKI
Elektronikus Tudományos Tanácsának elnöke
Szakmai lektor: Rosta János
Példányszám: 330, Törzsszám: 71-6176
Készült a KFKI sokszorosító üzemében,
Budapest
1971. december hó