tanulmányok

51/1976

MTA Számitástechnikai és Automatizálási Kutató Intézet Budapest

MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

TIMER IDŐREDUKCIÓS PROGRAMCSOMAG

Irta:
BAKÓ ANDRÁS

Tanulmányok 51/1976.

A kiadásért felel:

DR VÁMOS TIBOR

ISBN 311 023 8

768145 MTA KESZ Sokszorosító. F. v.: Szabó Gyula

TARTALOMJEGYZÉK

BEVEZETÉS

A Tanulmányban a CPM/PERT/time feladat egy általánositott változatát irjuk le, amely speciális esetként tartalmazza az alapfeladatot.

A problémát a gyakorlat vetette fel, és a megoldási algoritmus sok helyen alkalmazható.

A Tanulmány 3 részből áll. Az első fejezet a feladat felvetését tartalmazza. A 2. fejezetben megadjuk a probléma matematikai megfogalmazását és lehetséges megoldási módszereit. A 3. fejezet tartalmazza a feladat megoldására készitett TIMER nevü programrendszer leirását és a müködéséhez szükséges vezérlőkártyákat.

1. PROBLÉMA FELVETÉSE

A tervütemezési modelleket a számitógépek nagyipari gyártásával egyidőben készitették. A CPM/TIME és a PERT módszereket a RAND Corporation-nál dolgozták ki 1950-ben. Az első publikált irodalmi ismertetések 1956-ban /Ford [20] / és 1957-ben /Minty [9] / jelentek meg.

A CPM/COST változatát nemsokkal a fenti publikációk megjelenése után hozták nyolvánosságra /Kelley [5] , Fulkerson [4] /.

A két alapmodel mellet egy sor további problémát vetett fel a gyakorlat. Ezek egyike az erőforrás allokálás, amelyre eddig pontos algoritmus nem ismert. Egy sor közelitő eljárás született a különböző gépekre, amelyek rendszerint elég jó heurisztikus megoldást adnak a probléma megoldására. Ismeretes a feladatra egy magyar eljárás is, az ERAL.

Az ütemezési alapfeladatnak további általánosabb megfogalmazásait publikálta Ehnaghraby [1] és mások [10, 11] . Elmaghraby általánosan fogalmazta meg a feladatot: az egyes események bekövetkezése függ attól, hogy előtte mely események fejeződtek be. Az általa közölt model elsősorban irányitási, kutatásszervezési problémák megoldására alkalmas.

A számitógépek gyártói jelenleg a CPM/TIME/COST és a PERT/TIME algoritmusok programjait rendszerint a szamitógépekkel együtt szállitják. Ezek a programok nagyméretü hálózatokat képesek kezelni, I/O rendszerük rugalmas, a dátumgeneráló rész tetszésszerinti ünnepek, periódikusan előforduló munkanapok megoldását teszi lehetővé. Ezenkivül a háló adatainek felujitása néhány kártyával lehetséges, ami a programok gyakorlati felhasználását hatásosabbá teszi. Ennek során a régi és az ujabb háló adatait is el lehet tenni, igy egy sor variáns kidolgozását végezhetjük el, és minden fontosabb változás felvitelével naprakész ütemezési adatokat

kaphatunk.

A tanulmányban ismertetünk egy gyakorlatban jól használható modelt, amelynek programrendszerét a 3. fejezetben közöljük.

Tegyük fel, hogy egy CPM /vagy PERT/ feladatnál a háló felépitésekor nem ismerjük az átfutási időt. Ez nagymértékü hálók esetén nem is szigoru megkötés, mivel a számszaki eredmények nagysága lehetetlenné teszi az átfutási /kritikus/ idő megállapitását számitógép igénybevétele nélkül. Azt azonban tudjuk, hogy a teljes feladatot mennyi idő alatt kell elvégezni. A számitógép által szolgáltatott T_1 teljes átfutási idő és az általunk megkövetelt T_2 idő általában eltér egymástól. Ha $T_1 \cong T_2$, akkor a feladat megoldásával készen vagyunk.

Rendszerint azonban $T_1 > T_2$ vagy $T_1 < T_2$. Az első esetben a tevékenységek idejét csökkenteni kell ugy, hogy a $T_1 \cong T_2$ feltétel fennálljon. A 2. esetben a csökkentés helyett a tevékenységi idők növelése szükséges.

A csökkentés módját a gyakorlati feladat adja meg. Az mindenesetre nyilvánvaló, hogy ha egy tevékenység elvégzési idejét csökkentjük, akkor az azt végzőket intenzivebb munkára késztetjük. Ezért igyekszünk ugy módositani a tevékenységi időket, hogy lehetőleg minden tevékenység idejét egyszerre csökkentsük.

A teljes csökkentés mértéke függ a tevékenységektől. Az alsó korlátot tevékenységenként meg kell adni, aminek nemnegativnak kell lennie. A feladat megoldhatóságának kritérizma, hogy az alsó korlátokkal, mint tevékenységi idővel ellátott hálóban a kritikus idő ne legyen nagyobb T_2-nél.

A gyakorlati feladatoknál a csökkentési függvényként első és másodfoku polinomokat alkalmazhatunk. Más függvényeket természetesen beépithetünk a gépi programba, de az eddigi tapasztalat alapján a fenti függvények kielégitik a gyakorlati feltételeket.

Abban az esetben, ha az alapfeladatot oldjuk meg és a tevékenységek végrehajtása során csuszás van, de az eredeti befejezési határidőt ennek ellenére tartani akarjuk, akkor ugyanezt a modelt kell alkalmaznink. Tudniillik a hátralevő tevékenységek végrehajtási idejét ekkor ugy kell módositani, hogy a hátralevő tevékenységek részhálójában a kritikus ut a megadott korlát alá csökkenjen.

A 2. fejezetben elmondjuk a probléma matematikai megfogalmazását és megoldási lehetőségeit. Mi a csökkentésre a 2.2 pontban leirt algoritmust használtuk.

A 3. fejezetben leirjuk a programok müködését és felépitését, amelyet a SZTAKI CDC 3300 számitógépére készitettünk.

Saját programrendszer kiépitésére azért volt szükség, mert az algoritmust végző gyári programcsomagokba sok módositást, átdolgozást kellett volna tenni. Igy például a CDC programcsomagjában a hiarerchikus visszakeresési rendszer megadására nem volt lehetőség. Hasonlóképp nem volt eléggé rugalmas a rekord szelektáló és a rekordon belüli adatrészhalmaz kiirását végző programrész. A gyári rendszer időtáblázata is kicsi volt, és a legkisebb megadható idő is tizedhét volt. Az általunk készitett rendszer az alább felsoroltakban több, mint az átlagos gyári rendszerek:

- több rendezési szempont visszakeresésénél;
- nagyobb naptár és kisebb lépésköz;
- időredukciós algoritmus;
- többkulcsu visszakeresési lehetőség.

Az itt közölt leirás a Bakó A.-Kovács A. által irt kutatási szerződést lezáró tanulmány alapján készült. A programokat Kas Péter és Király László készitették a szerző irányitásával. A feladatot és a modelt Dr Bauer Frigyes fogalmazta meg.

2. TERVÜTEMHÁLÓK KRITIKUS IDEJÉNEK REDUKCIÓJA

2.1. A feladat matematikai megfogalmazása

Jelöljük egy körmenetes hálózat pontjainak halmazát $X=\{x_i\}$ -vel, irányitott éleinek halmazát E-vel és legyen megadva az éleken a $\tau(x_i,x_j) \geq 0, (x_i,x_j) \in E$ tevékenységi idő. Legyen az (X,E,τ) tervütemháló /továbbiakban háló/ kezdőpontja s, végpontja t.

Az idő ütemezési feladat egy olyan $\mu(x) \geq 0$ ütemezést megadni, amelyre

$$\mu(x_j)-\mu(x_i) \geq \tau(x_i,x_j), (x_i,x_j) \in E \qquad /1/$$

és $\mu(t) - \mu(s)$ minimális.

Ha az $\mu(x_j)-\mu(x_i) = \tau(x_i,x_j)$, akkor az (x_i,x_j) élt kritikusnak nevezzük.
Legyen $P = (s = x_o, x_1, \ldots, x_m = t)$ egy az (X,E,τ) hálóban s-ből a t-be vezető ut. A $\lambda(P) = \sum_{i=1}^{m} \tau(x_{i-1},x_i)$ értéket a P ut hosszának vagy időtartamának nevezzük, a fenti P utak közül a leghosszabbat kritikus utnak nevezzük.

A feladatot potenciál módszerrel oldjuk meg, amely megadja a μ ütemezést és a leghosszabb utat. A maximális hosszuságu P ut $\lambda(P)$ hosszára és a minimális idő ütemezésre érvényes a következő összefüggés:

$$\lambda(P) = \min(\mu(t) - \mu(s)). \quad \text{Legyen} \quad T = \lambda(P).$$

Legyen adva egy $T_1 \leq T$ szám.

A feladat ugy módositani az (X,E,τ) háló τ függvényét, hogy

a./ az (X,E,τ') módositott háló T' kritikus idejére teljesüljön az alábbi egyenlőtlenség: $T' \leq T_1$,

b./ a csökkentés mértéke legyen egyenletes, azaz ha $e_1,e_2,\ldots,e_m$ a háló élei és a csökkentés rendre $k_1,k_2,\ldots,k_m$ lépéssel történik, ugy $\max(k_1,k_2,\ldots,k_m)$ legyen minimális;

c./ ne csökkentsünk többet, mint amennyi a feltétel eléréséhez szükséges.

A csökkentést két további függvény $\alpha \geq 0$, $\beta \geq 0$ segitségével végezzük. Az $\alpha(x_i,x_j)$ függvény az $(x_i,x_j) \in E$ élen a tevékenységi idő alsó korlátja, azaz $\tau'(x_i,x_j) \geq \alpha(x_i,x_j)$. A β függvény az un. csökkentési függvény, amelynek a következő az értelmezése: legyen az aktuális tevékenységi idő $\tau'(x_i,x_j)$ az $(x_i,x_j) \in E$ élen. A következő lépésben a csökkentés mértéke $\beta(\tau'(x_i,x_j))$, az uj tevékenységi idő pedig

$$\tau''(x_i,x_j) = \tau'(x_i,x_j) - \beta(\tau'(x_i,x_j)) \qquad /2/$$

A csökkentési feladat akkor oldható meg, ha az (X,E,α) háló T^x kritikus idejére teljesül a $T^x \leq T_1$ feltétel.

Jelöljük $\mu(x)$ -el a legkorábbi, $\delta(x)$-el a legkésőbbi időket, és $\gamma(x_i,x_j)$-vel a maximális időtartalékot $(\gamma(x_i,x_j)=\delta(x_j)-\mu(x_i)-\tau(x_i,x_j))$.

A fenti feladat a következő esetekben fordul elő a gyakorlatban:

a./ egy objektum megalkotásának az ideje kisebb, mint amennyi a hálóban a kritikus idő;

b./ az objektum megvalósitása során egy t időpontban a kritikus éleken lemaradás van, és a véghatáridőt tartani akarják.

Mint a későbbiekben látni fogjuk, a megadott algoritmusok alkalmasak mindkét feladat megoldására. Ráadásul az átfutási idők variálásával a különböző költségkihatások, ten-

nivalók elemzése is lehetséges.

2.2. Csökkentés a teljes hálón

Egy kézenfekvő és viszonylag egyszerü algoritmust mutatunk be a kritikus idő T_1 érték alá csökkentésére.

Az A algoritmus az alábbi lépésekből áll:

A1: Egy leghosszabb ut algoritmussal megvizsgáljuk a feladat megoldhatóságát az (X,E,α) hálózaton. Amennyiben $T^* > T_1$ ugy az algoritmus végetér, a feladat megoldhatatlan, egyébként A2-nél folytatódik.

A2: Legyen $\tau_o(x_i,x_j) = \tau(x_i,x_j), (x_i,x_j) \in E, k = 0$.

A3: Legyen $k := k+1$;
a háló minden (x_i,x_j) E élén módositjuk a tevékenységi időket az alábbiak szerint

$$\tau_k(x_i,x_j) = \begin{cases} \tau_{k-1}(x_i,x_j) - \beta(\tau_{k-1}(x_i,x_j)), \text{ ha} \\ \qquad \tau_{k-1}(x_i,x_j) - \beta(\tau_{k-1}(x_i,x_j)) > \alpha(x_i,x_j) \\ \alpha(x_i,x_j) \text{ egyébként.} \end{cases}$$

A4: Számoljuk ki az (X,E,τ_k) háló T_k kritikus idejét. Ha $T_k \leq T_1$ ugy menjünk A5-re, egyébként A3-ra.

A5: Az (X,E,τ_k) azon (x_i,x_j) éleire, amelyekre $\gamma_k(x_i,x_j) > 0$ növeljük a $\tau_k(x_i,x_j)$ értéket lépésenként a $\beta(\tau_\ell(x_i,x_j)), \ell = k-1, k-2, \ldots$ értékekkel egészen addig, mig az első ℓ^*-ra teljesül a $\gamma_{\ell^*}(x_i,x_j) \leq 0$ feltétel, és ekkor $\tau_k(x_i,x_j)$ érték helyett az élen a $\tau_{\ell^*+1}(x_i,x_j)$ értéket vesszük.

Könnyü belátni, hogy az A algoritmus az 1. pontban kitüzött feladat megoldását adja. Az a./ feltétel az algoritmus A4 pontja miatt teljesül. Az A3 lépésben elmondottak miatt kielégül a b./ feltétel. Az A5 lépésben minden fe-

lesleges csökkentést elhagyunk, ez teljesiti a c./ feltételt.

2.3. Redukálás a kritikus utak mentén

A csökkentési feladat megoldásának egy másik algoritmusát mondjuk el ebben a fejezetben. Az itt leirt gondolat kézenfekvő: egy-egy lépésben csak a kritikus uton /vagy az alternativ kritikus utakon/ és esetleg a kritikus uttól hosszban nem sokkal eltérő utakon csökkentünk. Ekkor természetesen uj kritikus utak léphetnek fel, a csökkentést a következő lépésben ezeken folytatjuk...

Összes utvonal meghatározása

Az alábbiakban leirunk egy olyan algoritmust, amely egy hálóban az összes utvonal meghatározására alkalmas. Itt az algoritmust csak az első k leghosszabb utvonal meghatározására használjuk. Az alábbi algoritmus lényegesen egyszerübb, mint a k-adik optimális utakat meghatározó egyéb algoritmusok, /Yen [12], Fox[3]/, mivel a hálóban ciklus nem lehet.

A közlendő algoritmus potenciál módszerrel dolgozik. A A háló minden $x_i \in X$ pontjához hozzárendelünk egy $R(x_i) = (r_1(x_i), r_2(x_i), \ldots, r_q(x_i))$ potenciálvektort, amely tartalmazza az összes lehetséges ut hosszát az s ponttól az x_i pontig.

Vezessük be a következő jelölést: jelöljük a $(v_1, v_2, \ldots, v_r)$ számsor p-edik legnagyobb elemét $\max_p(v_1, v_2, \ldots, v_r)$-rel. Legyen S azon pontok halmaza, amelyek potenciál vektorát már kiszámoltuk és legyen T=X-S.

A B algoritmus a következő lépésekből áll:

B1: Legyen $S = \{s\}$, $R(s) = 0$.

B2: Keressünk egy olyan $x_j \in S$ pontot, amelyre nincs olyan $(x_i,x_j) \in E$, hogy $x_i \in S$ /ilyen pont van: l. Klafszky [6] 195 o./.

B3: Számoljuk ki az $R(x_j) = (r_i(x_j))$ potenciálvektort az

$$r_i(x_j) = \max_{\substack{x_i \in S \\ (x_i,x_j) \in E}} \{r_p(x_i) + \tau(x_i,x_j)\}_{p=1}^{q} \qquad /3/$$

ahol q az i-edik pontban kapott potenciálvektor hossza.

B4: Ha $x_j = t$ készen vagyunk, egyébként folytassuk a B2 pontban.

Az alábbi tétel a B algoritmus helyességét mondja ki.

2.1. Tétel: Az $R(t)$ potenciálvektor az s pontból a t pontba vezető összes utvonal hosszát adja.

Bizonyitás: Elegendő azt megmutatni, hogy a k-adik lépésben kiszámolt $R(x_{ik})$ tartalmazza az összes s-ből x_{ik}-ba vezető ut hosszát. Ezt az alábbiakban mutatjuk meg.

Tegyük fel, hogy az S halmazba az X halmaz pontjai a következő sorrendben kerültek be: $x_1,x_2,\ldots,x_n$. Az első lépésben s-ből x_1-ig egy ut vezet, és az ezen ut hosszát tartalmazza $R(x_1)$. Tegyük fel, hogy a tétel igaz az $S=\{x_1,x_2,\ldots,x_{k-1}\}$ halmazra és az $R(x_1)$, $R(x_2),\ldots,R(x_{k-1})$ potenciálokra. A k-adik lépésben kiszámoljuk az $R(x_k)$ potenciált és tegyük fel, hogy van egy olyan $P=(x_{i1}=s,x_{i2},\ldots,x_{i_{r-1}},x_{i_r} = x_k)$ ut, amely hossza nincs az $R(x_k)$ vektorban. Az $R(x_k)$ elemeit ugy számoltuk ki, hogy vettük

az összes $r_j(x_\ell) + \tau(x_\ell, x_k)$ összegeket, ahol $x_\ell \in S$, $(x_\ell, x_k) \in E$.

De a fenti összegben a $\tau(x_{r-1}, x_k)$ szerepelt, tehát a hossz csak ugy hiányozhat, ha a $P_1 = P\text{-}x_k$ ut hossza nincs az $R(x_{i_{r-1}})$ vektorban - szemben az indukciós feltevéssel.

Q.e.d.

2.4. Kritikus utak meghatározása

Az egy kritikus ut meghatározására van jó eljárás, de bonyolult az alternativ utak meghatározása. A maximális időtartalékok segitségével egy redukált hálón könnyen meg tudjuk határozni az alternativ kritikus utakat és a 2. leghosszabb utakat is.

Hasonló gondolattal próbálkozott M a e s - T e n g e l s [8] a $\mu(x_j)-\mu(x_i)-\tau(x_i,x_j)$ feltételes időtartalékok felhasználásával, de algoritmusa hibás az első lépésről a 2. lépésre való áttérésnél /1.204.o./.

Legyen P egy tetszésszerinti ut az s pontból a t pontba, és legyen a hossza $\lambda(P)$. Bontsuk a P utat három részre $P = P_1 \cup (x_i, x_j) \cup P_2$, ahol P_1 s-ből x_i-be, P_2 x_j-ből t-be vezet. A fentieknek elegettevő P, P_1, P_2 és az (x_i, x_j) élre, valamint a T kritikus időre az alábbi lemma teljesül.

2.1. Lemma A P ut hossza és a T kritikus idő között az alábbi összefüggés van

$$\lambda(P) \leq \tau(\delta(x_j) - \mu(x_i) - \tau(x_i, x_j)) \qquad /4/$$

Bizonyitás: A lemma helyességét számolással könnyen igazolhatjuk:

$$\lambda(P) = \lambda(P_1) + \tau(x_i,x_j) + \lambda(P_2) \leq$$

$$\leq \mu(x_i) - \mu(s) + \tau(x_i,x_j) + \delta(t) - \delta(x_j) =$$

$$= \delta(t) - \mu(s) - (\delta(x_j) - \mu(x_i) - \tau(x_i,x_j))$$

De $\delta(t) = \mu(t)$ miatt a lemmát kapjuk, azaz

$$\lambda(P) \leq T-(\delta(x_j) - \mu(x_i) -\tau(x_i,x_j) = T -\gamma(x_i,x_j)$$

Q.e.d.

A 2.1 Lemmából következik néhány egyszerü észrevétel:

a./ Ha valamely élen $\gamma(x_i,x_j)>0$, ugy ez nem kritikus él;

b./ Ha valamely (x_i,x_j) -re $\gamma(x_i,x_j)=0$, ugy van rajta átmenő kritikus ut. Ugyanis az x_i-ig van telitett ut μ -ben s-ből, az x_j-től van telitett ut δ -ban t-ig. Igy a lemmában az egyenlőtlenség helyett egyenlőség áll.

c./ Ha $(x_i,x_j)\in E$ esetén $\gamma(x_i,x_j) = 0$, ugy $\delta(x_j) = \mu(x_j)$:

ugyanis $\delta(x_j) - \mu(x_i) - \tau(x_i,x_j) = 0$

és $\mu(x_i) - \tau(x_i,x_j) = \mu(x_j)$

d./ A kritikus ut mentén $\delta(x_i) = \mu(x_i)$.

Ez utóbbi megjegyzést fogjuk kihasználni az alternativ kritikus utak meghatározásához.

Tekintsük az (X,E') irányitott gráfot, ahol $E' = \{(x_i,x_j)|\gamma(x_i,x_j) = 0\}$. Az algoritmus során az (X,E') gráf minden x_i pontjához hozzárendelünk egy $\omega(x_i)$ számot, amely a különböző /telitett/ utak számát adja s-ből x_i-be.

Legyen S azon x_i pontok halmaza, amelyre az $\omega(x_i)$ értéket már meghatároztuk:

A C algoritmus a következő lépésekből áll:

C1: Legyen $S = \{s\}$, $\omega(s) = 1$.

C2: Válasszunk egy olyan $x_j \in S$ pontot, amelyre nincs olyan $(x_i, x_j) \in E$, hogy $x_i \in S$.

C3: Számoljuk ki az $\omega(x_j)$ értékét:

$$\omega(x_j) = \sum_{(x_i,x_j)\in E} \omega(x_i) \qquad /5/$$

C4: Ha $t=x_j$ készen vagyunk, ellenkező esetben folytatódik az eljárás a C2 pontban.

Az algoritmus konstrukciójából következik, hogy $\omega/t/$ a különböző kritikus utak számát adja.

A 2.1 lemmából az is következik, hogy ha valamely (x_i,x_j) esetén $T^* = \gamma(x_i,x_j) > 0$, ugy van egy olyan ut, amely hossza $T-T^*$. Ezt az észrevételt használhatjuk a 2. leghosszabb utvonal meghatározására.

Tegyük fel, hogy

$$\min_{\gamma(x_i,x_j)>0} \gamma(x_i,x_j), (x_i,x_j) \in E$$

az (x_{i_o},x_{j_o}) élre teljesül. $T_o = T-\gamma(x_{io},x_{jo})$ hosszuságu utvonal létezik, s-ből x_{io}-ig μ-ben telitett, x_{jo}-tól t-ig δ -ban telitett éleken megy át. Megmutatjuk, hogy ez a 2. leghosszabb utvonal éppen T_o hosszuságu.

2.2 Lemma: Nincs olyan ut, amelynek hossza T és T_o közé esik.

Bizonyitás: Vegyünk egy tetszésszerinti T^* hosszuságu P^* utat. Megmutatjuk, hogy vagy $T^* = T$ vagy $T^* \leq T_o$.

Ha a P^* ut minden egyes (x_i,x_j) élére $\gamma(x_i,x_j) = 0$, akkor ez a kritikus ut, igy $T^* = T$.

Ha van olyan $(x_i,x_j) \in P^*$, hogy $\gamma(x_i,x_j) > 0$, akkor a P^{**} ut T^{**} hosszára, amely s-től x_i^* -ig μ-ben telitett, x_j^* -tól t-ig δ-ban telitett, fennáll a $T^* \leq T^{**}$ egyenlőtlenség. Mivel $\gamma(x_o,y_o)$ minimális volt, igy $T^{**} \leq T_o$. Összefoglalva éppen a kivánt eredményt kaptuk, mivel

$$T^* \leq T - \gamma(x_i^*,x_j^*) \leq T - \gamma(x_{io},x_{jo}) = To$$ Q.e.d.

A fenti eredmény nem általánostiható a 3. legrövidebb utak meghatározására. Azaz általában nem igaz, hogy ha $\gamma(\bar{x}_i,\bar{x}_j)$ a maximális időtartalékok között a 3. legkisebb, akkor a 3. leghosszabb ut hossza $T-\gamma(\bar{x}_i,\bar{x}_j)$.

A fenti két módszert használhatjuk a kritikus ut csökkentésére. A B algoritmussal kiszámoljuk az első k utat, a C algoritmussal a leghosszabb utakat és a 2. leghosszabb utakat. Ezeken az utakon elvégezzük a kivánt csökkentést ugy, hogy ezen utak hossza T_1 alá kerüljön, és minden élen ugyanannyi lépésszámon csökkentsük. Az uj hálón az eljárást folytatjuk tovább egészen addig, mig az aktuális $\tau'(x_i,x_j)$ függvény eleget nem tesza 2.1 pontban megfogalmazott feltételeknek.

2.5. Számitástechnikai megjegyzések

A fejezetben bemutatjuk egy minimális tárolási igényt és maximális rugalmasságot nyujtó tárolási módszert.

Egy n pontból álló háló tárolási igénye n^2. Nagyméretü hálók ritkák /és az n^2 is nagy/, ezért a tömör tárolás helyett célszerü egy ujabb módszert bevezetni az adatok kezelésére. Ráadásul a hálók gyakorlati alkalmazásakor sokszor kell az alapadatokat módositani, éleket beiktatni, törölni stb. Az itt közölt tárolási módszer figyelembe veszi a fenti igényeket és kevés memóriát igényel. Ezen kivül igazodik az algoritmus jellegéhez is: a μ és δ potenciálok kiszámolásához az egy pontból kimenő és az egy pontba befutó élek egyaránt kellenek.

A tárolási módszer a ritka mátrixok tárolási módszeréhez hasonlit /l. Knuth [7], 299.o./ és duplán összekapcsolt lista strukturával dolgozik.

A módszer 2n+3m tárolóhelyet igényel, ahol n a pontok, m az élek száma. Legyenek $A=(a_i)$ és $B=(b_i)$ n hosszuságu vektorok és $T=(t_i)$, $K=(k_i)$ és $L=(l_i)$ m hosszuságu vektorok. Az a_i elem adja meg az i pontból kifutó első tárolt él τ_{ij} értékének a helyét a T vektorban, a b_k vektor ugyanezt a k pontba befutó első élre. Azt, hogy az i pontból kifutó első él végpontja hol van, a következőképp állapithatjuk meg: az L vektor l_i eleme azt mutatja, hogy a következő ebben a pontban befutó él τ_{ij} értéke hol van a T vektorban, az ezen a listán levő utolsó l_k értéke negativ és abszolut értéke a végpont indexe. A K vektor az L vektorhoz hasonló funkciót tölt be, csak a kimenő pontokra vonatkozóan.

Ez a tárolási módszer a módositás és tárolási hely szempontjából nagyon jó hatásfoku. Egy él kezdő és végpontjának meghatározásához $k^2/2$ összehasonlitására van szükség.

3. PROGRAMRENDSZER FELÉPITÉSE

A feladat megoldására kifejlesztett programrendszer COBOL nyelven megirt keretprogramból és általa hivott rutinokból, valamint önálló FORTRAN programból áll.

A matematikai müveleteket /csökkentés, kritikus ut és tevékenységi idők/ a CSOKK nevü FORTRAN rutin és a CPMTIME nevü FORTRAN program végzi.

Az adatok felvitelét, módositását, a több kulcs szerinti visszakeresést és a különböző perifériákon való kiirást a CPM-MANAGER nevü COBOL keretprogram végzi. A programok aktivizálását a gépbe beolvasott vezérlőkártyákkal lehet elérni. A különböző paraméterek bevitele történhet távállomáson keresztül alfanumerikus display vagy kártyaolvasó segitségével, vagy a központi kártyaolvasón keresztül. Az eredmények kiiratása line printeren, permanens disk vagy mágnesszalag file-n, távállomáson keresztül display-en és lyukszalagon történhet.

A rendszer I/O bemenetét az 1.sz. ábrán mutatjuk be.

A különböző nyelvü programok egymással paraméter átadással, illetve külső file-kon keresztül kommunikálnak. A CPM-MANAGER keretprogram a hozzá kapcsolódó kis FORTRAN rutinokat paraméter átadással /az ENTER hivással/ aktivizálja. A COBOL keretprogram a CSOK és a CPMTIME programoknak /illetve programoktól/ permanens disk file-n adja át /illetve kapja/ az adatokat. A programok egymás közötti adatátvitelét a 2. sz. ábrán mutatjuk be.

3.1. Program Input-Output formátuma

3.1.1. Input adatok

Az input adatok egy részét adatkártyákon adjuk meg, más része a program vezérlőkértyáin kap helyet. Az alapadatok felvitele két lépésben történhet. A hálóhoz tevékenységenként meg kell adni a hozzá szükséges összes információt a szövegek kivételével. A szövegek felvitele opcionális, mivel bizonyos vezetési hálók esetén nem célszerü a tevékenységekhez tartozó szövegek felvitele. Ezért lehetőség van az alapadatok felvitele után közvetlenül a szövegek felvitelére, de ugy is lehet szervezni a felvitelt, hogy a tevékenységek szöveges felvitelét csak a szöveges listázások előtt vigyük fel, és a futás után automatikusan töröljük. Tevékenységenként az alábbi számszaki információkat kell megadni.

1 - 4 CH	tevékenység sorszáma
5 - 8 CH	kezdő esemény
9 -12 CH	befejező esemény
13 -17 CH	átfutási idő tizedórákban
18 -24 CH	csökkentési függvény együtthatói és alsó korlátja
25 -45 CH	tevékenységet koordináló vállalatok kódjai (7CH vállalatonként)
46 -79 CH	tevékenységet végző vállalatok kódszámai

A felvitel módját a felvivő és visszakereső rendszer leirásánál adjuk meg.

A tevékenységhez tartozó egy vagy két szövegkártyát a következő formában kell megadni:

<u>első kártya:</u>

1 - 4 CH	kezdő esemény száma

5 - 8 CH	befejező esemény száma
9 CH	üres
10 CH	kártyaszám, értéke 1
11 -80 CH	tevékenység szövege

második kártya:

1 - 4 CH	lásd 1. kártya
5 - 8 CH	lásd 1. kártya
9 CH	üres
10 CH	kártyaszám, értéke 2
11 -80 CH	lásd 1. kártya

A szövegkártyák felvitelének módját és az induló dátum megadását a következő paragrafusban mondjuk el.

A CPMTIME nevü FORTRAN program a csökkentés mértékét - ha ilyen van - egy külön kártyáról olvassa le, amelyen az első 6 CH a megkivánt átfutási időt adja meg tizedórákban.

3.1.2. Output formátuma

A kritikus utat számoló CPMTIME program kiirja táblázatos formában a hálóhoz tartozó fontosabb alapadatokat és az általa kiszámolt időtartartamakat. A táblázat egy sora a következő elemeket tartalmazza:

tevékenység sorszáma
kezdő esemény száma
befejező esemény száma
tevékenység időtartama
legkorábbi és legkésőbbi idők
szabad időtartalék
feltételes időtartalék
független időtartalék

A CPMTIME program által hivott CSÖKK rutin kiirja a tevékenység sorszámát, a kezdő és befejező esemény számát és az előző és a csökkentett tevékenységi időtartamot csökkentési lépésenként.

A CPM-MANAGER keretprogram rugalmas output táblázatok megadását teszi lehetővé. A listázás vonatkozhat az összes tevékenységek halmazára, vagy bizonyos szempontok szerint kiválaszthatjuk a tevékenységek egy részhalmazát.

Hasonlóan rugalmas az egy tevékenységhez tartozó adatok halmazának a kiirása is: tetszésszerinti részhalmaz kiirására is lehetőség van.

A kiirási lehetőségeket a program vezérlőkártyáinak leirásánál mondjuk el.

3.2. Belső adattárolás

A CSOKK és a CPMTIME programok FORTRAN adatfile-kkal dolgoznak.

Az alábbi FORTRAN formátumu adathalmazok vannak disken tárolva a futás alatt: háló számszaki alapadatai i,j,τ_{ij} sorozatként; a csökkentéshez szükséges függvények együtthatói és a tevékenységi idők alsó korlátai: a csökkentés legutolsó eredménye i,j,τ'_{ij} alakban; a négy alapidő és a négy időtartalék számszaki eredménye.

A COBOL keretprogram olvassa be, teszi el az alapadatokat és veszi át a FORTRAN programoktól a számitási eredményeket. Ezen adatokat egy file-ba /az un. T-FILE/ teszi el.

A T-FILE egy rekordja az alábbi alaku:

adatnév	méret	tipus	tartalom
KEL	5CH	numerikus	kezdőpont kódja
VÉL	5CH	numerikus	végpont kódja
IDK	5CH	N	tevékenységi idő
MINK	5CH	N	legkorábbi /kezdési/idő
MAXB	5CH	N	legkésőbbi /befejezési/ idő
MAXK	5CH	N	legkésőbbi kezdési idő
MINB	5CH	N	legkorábbi befejezési idő
TART	5CH	N	maximális időtartalék
SZAB	5CH	N	szabad időtartalék
FELT	5CH	N	feltételes időtartalék
FUGG	5CH	N	független időtartalék
NKØD	6CH	N	szövegkód
TELN	130CH	A	szöveg
FLAG	1CH	A	flag 1, ha kritikus él, 0 egyébként

adatnév	méret	tipus	tartalom
HATØ	17OX2CH	N	tevékenységet megrendelő vagy felügyelő szervek kódjai
PARG	3X7CH	N	tevékenységet végző vállalatok kódjai
A,B,C	3X4CH	N	függvény együtthatói és alsó korlátja

A szöveges kiiráshoz egy sor szöveg-konstansra van szükség. Ezek egy rész a T-FILE-on, más része pedig a programban nyert elhelyezést.

3.3. FORTRAN NYELVÜ PROGRAMOK ÉS RUTINOK

3.3.1. Időredukciót végző programrész

A CSOKK nevü FORTRAN program végzi az eredeti tevékenységi idők csökkenését. Feladata a kritikus ut hosszára a számitás eredményeképp kiadódó T időtartamot a megadott időkorlát alá szoritani - az éleken megadott idők csökkentésével. A csökkentés pontos leirását a matematikai algoritmus megadásánál adtuk meg. A CSOKK program egy csökkentési lépést végez a háló minden élén. A csökkentés mértékének ellenőrzését a CPMTIME nevü /szintén/ FORTRAN nyelvü/ rutin végzi a csökkentett időértékü hálózatra - kritikus ut kiszámitásával.

A 21 cimkéjü file-n az i,j,τ_{ij} értékek vannak egymásután, a hozzá tartozó első- illetve másodfoku polinom együtthatói a 23 file-n vannak rendre. A 21 és 23 file hossza 6000, ami maximálisan 2000 tevékenységből álló háló kezelését teszi lehetővé. A méretek ennek többszö-

rösére növelhetők a program néhány utasitásának egyszerü cseréjével.

A csökkentést egy függvény segitségével végezzük el /1. matematikai megfogalmazás/. A függvény alakjára semmiféle megkötés nincs. A gyakorlati feladatok megoldása azt mutatta, hogy a tevékenységi idők leszoritásának gyorsaságát és a különféle tevékenységek jellemzését jól meg lehet közeliteni első, vagy másodfoku polinommal. Ezért a programban is az x^2+bx+c illetve $bx+c$ alaku függvény szerepel, de tetszésszerinti függvényt be lehet épiteni a megfelelő paraméterek megváltoztatása után.

A C csökkentési eljárás a következő lépésekből áll:

C1: beolvassa az összes rekordot a memóriába a 21 illetve 23 file-król /a D illetve P vektorokba/;

C2: veszi az /i,j/ élt és csökkent:

$$\tau_{ij} := \tau_{ij} - a_{ij}\tau_{ij}^2 - b_{ij}\tau_{ij} - c_{ij}$$

ahol τ_{ij} a tevékenységi idő a_{ij}, b_{ij}, c_{ij} az /i,j/ élhez tartozó megfelelő polinom értéke;

C3: ha minden élre elvégezte a fenti csökkentést, akkor menjen C4-re, egyébként C2-re;

C4: kiirja az uj $\tau_{i,j}$ értékeket line printerre és a 21-es file-ra.

3.3.2. Kritikus ut számitása

A CPMTIME nevü FORTRAN nyelvü rutin segitségével határozhatjuk meg a kritikus utat:

A 21 nevü file-n vannak a háló éleire vonatkozó adatok i,j,τ_{ij} sorrendban. A kritikus utat kétszer számoljuk ki: egyszer az s kezdő ponttól a t befejező pontig, majd a t ponttól visszafelé az s pontig. Az első eljárás során meghatározzuk a legkorábbi időket és a P vektorban tároljuk. A második /a t pontból visszafelé/ számolásnál meghatározzuk a legkésőbbi időket. /A legkorábbi idők azzal a tulajdonsággal birnak, hogy minden pontba vezet kritikus ut az s pontból, a legkésőbbi időkre vonatkoztatva pedig minden x ponttól a t-be vezet kritikus ut!/

A legkorábbi idők számitási algoritmusa a hálózatban található kört is kikeresi - ha ilyen van - és kiirja "A HÁLÓZAT CIKLIKUST TARTALMAZ" szöveget. Ebben az esetben az eljárás befejeződik, mivel körrel rendelkező hálózatban a leghosszabb ut feladat nem oldható meg.

Jelöljük a legkorábbi időket $\mu/x/$ -el, a legkésőbbi időket $\delta/x/$ -el.
A fenti idők meghatározása után kiszámolja az alábbi időtartalékokat is:

maximális	$\delta(y) - \mu(x) - \tau(x,y)$
szabad	$\mu(y) - \delta(x) - \tau(x,y)$
feltételes	$\mu(y) - \mu(x) - \tau(x,y)$
független	$\delta(y) - \delta(x) - \tau(x,y)$

Ezeknek jelentése a szokásos: maximális időtartalék akkor jön létre, ha a tevékenység legkorábban kezdődik és legkésőbben fejeződik be. Szabad időtartalék esetén a tevékenység legkésőbben kezdődhet és legkorábban kell befe-

jeződnie. Feltételes időtartalék esetén feltesszük, hogy a tevékenység legkorábban kezdődhet és az őt követő esemény legkorábbi időpontja nem ronthatja el. Független időtartaléknál amint az x-be érkező tevékenységek befejeződtek, kezdődhet a tevékenység, de a követő tevékenységre nincs tekintettel.

A fenti időtartalékok mellett két további időt a legkésőbbi kezdési $((\delta(y)-\tau(x,y))$, illetve a legkorábbi befejezési $(\mu(y)-\tau(x,y)$ időket is kiszámoljuk.

A négy alapidőt és a négy időtartalékot line printerre és a 24-es disk file-ra is kiirjuk, ahonnan majd T-filera is átiródik.

A futás során tömör formában kiirásra kerül a 2 potenciál vektor /legkorábbi és legkésőbbi idők/ és a megfelelő 2 cimkevektor is.

A kritikus idő redukciója a következőképp történik:

D1: a CPMTIME rutin beolvas egy kártyát, amely az általunk megadott kritikus idő hosszát tartalmazza órákban és tizedórákban /xxxxxy/ ahol xxxxx az órákat y a tizedórákat jelenti/;

D2: behivja a CSOKK nevü rutint, amely az ott elmondott egyetlen csökkentési lépést végrehajt;

D3: a CPMTIME ellenőrzi, hogy a csökkentett idejü hálózatban a kritikus ut hossza nem nagyobb-e az általunk megkivántnál; ha nagyobb a D2 pontban folytatódik az eljárás, egyébként a D4 pontban;

D4: a tevékenységi idők csökkentését befejeztük, és a 21 cimkéjü file tartalmazza a végeredményt.

3.3.3. Belső I/O rutinok

A COBOL keretprogram aktivizálja az egyes számitást végző rutinokat, beolvasást, kiirást végez stb. A fenti funkciók elvégzése során egy sor file tartalmát kell beolvasni illetve kiirni. Mivel az adatok számszaki részével FORTRAN nyelvü rutinok dolgoznak, igy a két nyelv közötti kommunikációra a FORTRAN nyelven irt file-ket csak ezen a nyelven irt programmal tudjuk beolvastatni és viszont. Az ENTER COBOL igével ezeket az I/O rutinokat aktivizáljuk.

A fenti I/O funkciókat látják el a LE1, FEL1, FEL2, FEL3 és LEREDU nevü rutinok.

3.4. Vezérlőprogram

A CPM-MANAGER nevü vezérlőprogram az input alapadatok felvitelére, a FORTRAN rutinokkal való kommunikációra és az eredmények adatbankszerü kezelésére szolgál. Mint azt a bevezetőben elmondtuk, az eredmények közölhetőek line printeren, vagy kártyán, vagy lyukszallagon, vagy on-line módon display-en. Ez utóbbi esetben egy-egy szelektálás után a tevékenységek egymás után jelennek meg a képernyőn, és addig ott vannak, mig egy vezérlőjellel tovább nem inditjuk a programot. A részletes leirást a fejezetben ismerjük.

3.4.1. Input vezérlőkártyák

Egy feladat alapadatait és a tevékenységekhez tartozó szöveges leirást a NETUP és TEXTUP vezérlőkártyákkal lehet gépre vinni.
A számszaki eredmények géprevitelét a NETUP vezérlőkártyával érhetjük el.

Ekkor a kártyák sorrendje a következő:

	1 2
1. kártya:	* NETUP
2. kártya:	az első tevékenység számszaki adatkártyája
3. kártya:	a második tevékenység számszaki adatkártyája

Az adatkártyák végét egy ENDATA kártya jelzi.

Szövegfelvitel esetén a kártyák sorrendje a következő:

	1 2
1. kártya:	* TEXTUP
2. kártya:	1. tevékenység 1. adatkártyája
3. kártya:	1. tevékenység 2. adatkártyája
4. kártya:	2. tevékenység 1. adatkártyája

Az adatkártyák végét egy ENDATA kártya jelzi.

3.4.2. Visszakeresési rendszer vezérlőkártyái

A visszakeresést és szelektálást a MINISORT nevü rutin végzi. Ez a rutin vezérli a megadott szempontok szerint visszakeresett rekordok /tevékenységek/ megadott formátumu listázását is.

A MINISØRT 4 vezérlőkártyát fogad el.
Ezek:

* LØØK a választási szempontok vezérlőkártyája,
* SØRT a feldolgozott információk rendezésének vezérlőkártyája,

/ØUTPUT az output információk vezérlőkártyája,
/TIMEBASE a naptári alapidő kártyája.

A fenti sorrendben egy-egy kártya egy-egy feladatot ir

elő a MINISØRT számára.

Párhuzamosan a MINISØRT rutinnal tetszőleges számu, a fenti kártyákkal specifikált feladatot tudunk elvégeztetni.

A LØØK vezérlőkártya irja elő a válogatás szempontjait. Két formáját alakitottuk ki:

a./ /LØØK, ALL

b./ /LØØK, $Kulcs_1$ relációjel, $literál_1$, $kulcs_2$, $r.jel_2$, $lit._2$,...,$kulcs_n$,$r.jel_n$,$lit._n$

ahol $kulcs_1$, $kulcs_2$,... $kulcs_n$ a 3.2 pontban leirt adatnevek valamelyike,
az $rjel_1$, $rjel_2$,...$rjel_n$ az =,<,> jelek valamelyike,
a lit_1, lit_2,... lit_n szám.

Az a./ tipusu vezérlőkártya hatása a MINISØRT az összes rekordot listázza, azaz szelektálás nem történik.

A b./ tipusu vezérlőkártya hatására pedig a MINISØRT visszakeresi azokat a rekordokat, melyek a specifikált feltételeket egyszerre kielégitik.

A SØRT vezérlőkártya irja elő, hogy a visszanyert rekordokra vonatkozó információk milyen sorrendben, mely adat szerint rendezve jelenjenek meg az output listán.
Formája:

/SØRT, kulcs,

ahol kulcs a 3.2. pontnak megfelelően a következő adatnevek valamelyike: KEL, VEL, IDOK, MINK, MAXB, TART, SZAB, FELT, FUGG, HATO, PARG.

Az ØUTPUT vezérlőkártya irja elő, hogy a visszanyert rekordokra vonatkozóan mely információk jelenjenek meg az output listán.

Formája kétféle lehet:

a./ /ØUTPUT, $kulcs_1$, $kulcs_2$,... $kulcs_n$,
ahol $kulcs_1$,... $kulcs_n$ a T-FILE-ban felsorolt adatnevek valamelyike;

b./ /ØUTPUT, ALL

Az a./ esetben a $kulcs_1$, ... $kulcs_n$ adatmezőkben tárolt információk,
b./ esetben a teljes rekordtartalom jelenik meg.

A CPM-MANAGER-be épitett MINISØRT elfogad egy negyedik vezérlőkártyát is. Ez a TIMEBASE vezérlőkártya, amely beállitja a CPM-MANAGER naptáron az un. alapidőt. A továbbiakban a megfelelő rutinok gondoskodnak arról, hogy a programok által használt abszolut idők a listákon az alapidőre vonatkoztatott relativ idővé transzformálódjanak.
Formája:

/TIMEBASE, xxyyzzvvww

ahol xx az év utolsó két számjegye
yy a hónap arab számjegyes megjelölése
zz a nap arab számjegyekkel
vv az óra
w a tizedóra

A /TIMEBASE kártyát a MINISØRT kártya után kell elhelyezni.

3.4.3. A vezérlőprogram felépitése

A vezérlőprogram COBOL nyelven iródott. Tartalmaz egy sor szekciót és ezek a szekciók a vezérlőkártyáknak megfelelően aktivizálódnak.

A CSOKK és CPMTIME FORTRAN rutinok a vezérlőprogramtól az adatokat a 3.3.3. pontban leirt FORTRAN nyelvü rutinokon keresztül kapják. A csökkentés és a kritikus ut számitás eredményeit ugyancsak ezek a rutinok adják át a keretprogramnak.

A különböző szempontok szerinti rendezést a COBOL SORT igével végezzük el. A visszakeresést külön szekció végzi a megfelelő vezérlőkártyák paramétereinek kiértékelése után.

A listázást REPORT-WRITER felhasználásával készitjük el.

A display kiiratást egy külön szekció végzi, a használatát a 3.4.5. részben mondjuk el.

3.4.4. Szolgáltatott információk rendszere és funkciója

A program összetettségéből is következik, hogy a felhasználás is sokrétü lehet. Ezt nemcsak a feladatrendszer sajátossága miatt alakitottuk igy, hanem biztositani kivántuk:

- a vezetési szinteknek megfelelően az információk menynyiségének differenciálását;
- az információk felhasználásának egyszerüségét /pl. csak azok az információk jelenjenek meg, amelyeket igényelnek/;

- az információk felhasználásának sokrétüségét /pl. a kért információk csak számszaki vagy alfanumerikus formában jelenjenek meg./

A programrendszer futtatásával az előző fejezetben tárgyalt vezérkártyák segitségével alapvetően kétféle formában jelentethetjük meg az eredményeket:

- táblázatos formában /csak numerikus jelek/,
- szabad formában /alfanumerikus jelek/.

3.4.4.1. Táblázatos formában listázott információk

Elsősorban a tevékenységek részletes elemzési céljára alakitottuk ki. A táblázatos formában megjelenő adatok előtt találjuk az un. azonositó paramétereket, amelyek a következők:

- XX. számu tevékenység tabló: a XX helyén megjelenő szám jelzi, hogy a táblázat adatai melyik összeg átfutási időérték figyelembevételével készült számitás eredményeit tartalmazzák. A programrendszer ugyanis lehetővé teszi, hogy a T-FILE-n tároljuk maximálisan 10 különböző össz átfutási értékhez tartozó számitás eredményeit. Egy kritikus utszámitás kb. 6 perc netto gépidőt igényel.

- A visszakeresés szempontjai: e szöveget követően megjelennek soronként azok a visszakeresési szempontok, amelyeket a /LØØK vezérkártyán a 3.2. pontban ismertetett T-FILE szerkezet "kulcsai" alapján rögzitettünk.

- Rendezési kulcs: e szöveget követően megjelennek a T-FILE szerkezetének és kulcsainak megfelelő szövegek,

amelyeket a /SØRT kártyával vezérlünk.

-Rekordok száma: szöveg után megjelenő szám jelzi, hogy az adott feldolgozásnál hány tevékenység felelt meg a kiválasztás paraméterenként rögzitett ismérveknek.

Amennyiben a hálódiagram valamennyi tevékenységének szövege vagy adatai kiirásra kerülnek, ugy itt azt a számot kapjuk, amely a hálódiagram tevékenységeinek számával egyenlő. Ez részben ellenőrzésre, részben pedig a visszakeresés szempontjainak részarányára ad megközelitő tájékoztatást.

Alapidő után következő számcsoport értékei jelzik a háló funkcióba lépésének kezdő értékét, feltüntetve az év, hó, napot, valamint az órát tizedóra pontossággal. Pl. ha az output táblán a következő számot találjuk: 70/01/01 00.0, akkor az 1970. jan. 1. 00 óra 00 percnek felel meg.

A fenti azonositó paraméterek után következik a táblázat maga. A táblázat egy sora az alábbi elemekből áll:

- KCS, amely a kezdő események száma,
- VCS, amely a befejező események száma,
- IDO, amely napban, órában tizedóra pontossággal jelzi az egyes tevékenységek adott vagy számitott átfutási időértékét attól függően, hogy a kritikus ut meghatározásánál előre rögzitettük-e az össz átfutási időértéket,
- MINKEZD, amely az egyes tevékenységek legkorábbi kezdési időpontját adja év, hó, nap, óra, tizedóra bontásban,

- MINBEF, amely az egyes tevékenységek legkorábbi befejezési időpontját mutatja a MINKEZD-nél leirt formában,

- MAXKEZD, amely a tevékenység maximális kezdési időpontját adja a MINKEZD-nél leirt formában,

- MAXBEF, amely az egyes tevékenységek maximális idejét mutatja a MINKEZD-nél elmondott formában,

- TARTIDO, a hálódiagram eljárásból megismerhető teljes tartalékidő értékét rögziti nap, óra, ill. tizedóra pontossággal. Ahol ezen oszlopon * jelet találunk, azok a kritikus tevékenységeket jelzik. Ezen tevékenységeknél ellenőrizhető, hogy

$$\text{MINKEZD} = \text{MAXKEZD} \quad \text{és}$$
$$\text{MINBEF} = \text{MAXBEF}.$$

- SZABIDO a szabadidő tartalékot adja meg a fent leirt formában,

- FELTIDO a feltételes időtartalékot adja meg a fent leirt formában,

- FUGGIDO a független időtartalékot adja meg a fent leirt formában.

3.4.4.2. Szabad formában listázott információk

Amint már e fejezet bevezetésében rögzitettük, a visszakeresett információk tartalmában nincs különbség, csupán a vezérlő kártyákon rögzitett kérdés-különbségeknek megfelelően különböznek egymástól a kiirt sorok. A táblázatos és szabad forma között csupán néhány különbséget találunk, az utóbbi esetben ugyanis:

- az egyes tevékenységek pontos szövegét is kiirja a kiiró berendezés,

- a végrehajtásért felelős vagy azt megrendelő szerveket is listázza a gép.

Hasonlóan a táblázatos formánál már ismertettekkel, ebben az esetben is megtaláljuk az un. azonositó paramétereket, amelyek felsorolásszerüen a következők:

- XX. számu TEVÉKENYSÉG TABLÓ:
- A VISSZAKERESÉS SZEMPONTJAI:
- RENDEZÉSI KULCS:
- REKORDOK SZÁMA:
- ALAPIDŐ:

Az azonositó paraméterek után következnek a vezérlő kártyával kiválasztott tevékenységekre vonatkozó információk az alábbi felépitésben:

- az első sorban található:
  a./ sor elején *, ami az adott tevékenység kritikus jellegét mutatja,
  b./ a kezdő és befejező események száma"-"jellel elválasztva,
  c./ IDO felirat és ezt követően az adott tevékenység előre rögzitett, vagy géppel számitott átfutási időértéke napban, órában tizedóra pontossággal megjelölve, majd ezt követően a /SØRT és /ØUTPUT vezérkártyák tartalmának megfelelően listázásra kerülnek a tevékenységre vonatkozó további jellemzők.

A tevékenység szövege

- VÁLLALATOK, ahol szövegesen vagy csak kódszámok formájában jelennek meg a végrehajtásért felelős szervek, szervezeti egységek,
  személyek

- MINIMALKEZDÉS:
- MINIMÁLBEFEJEZÉS:
- MAXIMALBEFEJEZÉS:
- MAXIMÁLKEZDÉS:
- TELJESTARTALÉK:
- SZABADIDŐ:

A kezdési és befejezési időpontok sorrendjét, darabszámát az előbbiekben már emlitett vezérkártyákon tetszés szerinti sorrendben, bármelyik kulcs elhagyásával vagy rögzitésével kérhetjük.

3.4.5. On-line rendszer

Az előre megadott szempontok szerint visszakeresett információk szabad formában listázandó része /l. 3.4.4.2/ display is megjelenithető ugy, hogy a képernyőn megjelenő információk vezérlését az egység billentyü rendszerén bonyolitjuk le.

A displayt közvetlen perifériaként tele fonvonalon keresztül, vagy távállomásként is rá lehet kötni a gépre. Telefonvonalon kipróbáltuk a rendszer a géptől mintegy 15 km távolságra és a gépi futás valamint az információk képi megjelenitése kifogástalan volt.
Display-ről vezérlőkértyák és adatok beolvasására, módositására is van elvileg lehetőség. A gyakorlati megvalósitás most folyik.

A display megnyitása operátorral való kommunikációval történik. A gép ekkor konzolon keresztül csatornát kér a display számára a következő szöveggel: "OPERATOR PLEASE TYPE TERMINAL-LINE NUMBER". A jelenlegi verzióban a le-

hetséges jó válasz az UTlO, UTO2, UTOO szövegek egyike. Ha az operátori üzenetet elfogadta a gép, akkor kiirja a THANK YOU szöveget. Ellenkező esetben ujra kiirja az "OPERATOR PLEASE..." szöveget és a konzolra kiirja uj sorba a jó válaszok valamelyikét, azaz az "UTlO vagy UTO2 vagy UTOO" szöveget.

Ha megkapja a megfelelő csatornaszámot, akkor kiirja a képernyőre a használati utasitást az alábbi szöveg formájában:

"UDVOZLOM A KEDVES FELHASZNALOT"
ISMERTETJUK A PARBESZED SZABALYAIT

5-FELE VALASZTASI LEHETOSÉG VAN:

S= INDULAS ILLETVE KEREM A KOVETKEZO KEREST
P= KEREM A KOVETKEZO REKORDOT
Y= KEREM A REKORD FOLYTATASAT*
F= FEJEZZUK BE
R= KEZDJUK ELOLROL"

* amennyiben a rekord nem fér el egy képernyőn /azaz 20 sorban/, akkor a rekordhoz tartozó további adatok az Y betü lenyomása után jelennek meg a képernyőn.

Ha vége van egy kérés teljesitésének, ugy az alábbi szöveg jelenik meg a képernyőn:

"AZ UTOLSO REKORDOT LATJA
A LEKERDEZESNEK VEGE
R BETUVEL UJRA INDITHAT EGYEBKENT
USSON BE EGY F BETUT
A VISZONTLATASOMRA"

A használati utasitás alapján a képernyő vezérlése egyszerü:

a./ "S" betüvel megjelenik az első kérés első rekordja

b./ egymásutáni "P" betük lenyomásával egymásután jelennek meg a kért rekordok

c./ "F" betü hatására az első kérés utolsó rekordját irja ki

d./ "S" betüvel a következő kérés első rekordja jelenik meg...

Az egység megnyitását és az I/O-ot a COMPASS nyelvü rutinok behivásával végezzük /TOPEN ill. IDISPLAY/.

A kiirás formája ekkor megegyezik a szabad formában listázott információknál elmondottakkal /l. 3.4.4.2. pont/. Minden kérés elején listázódnak a visszakeresési e listázási szempontok. Majd egymásután megjelennek a képernyőn az általunk kért tevékenységek általunk kért adatai.

A programrendszer rutinszerü alkalmazása céljából az alábbiakban ismertetünk egy feladatot, majd annak futásához elkészitendő kártyákat:

A feladat megfogalmazása:

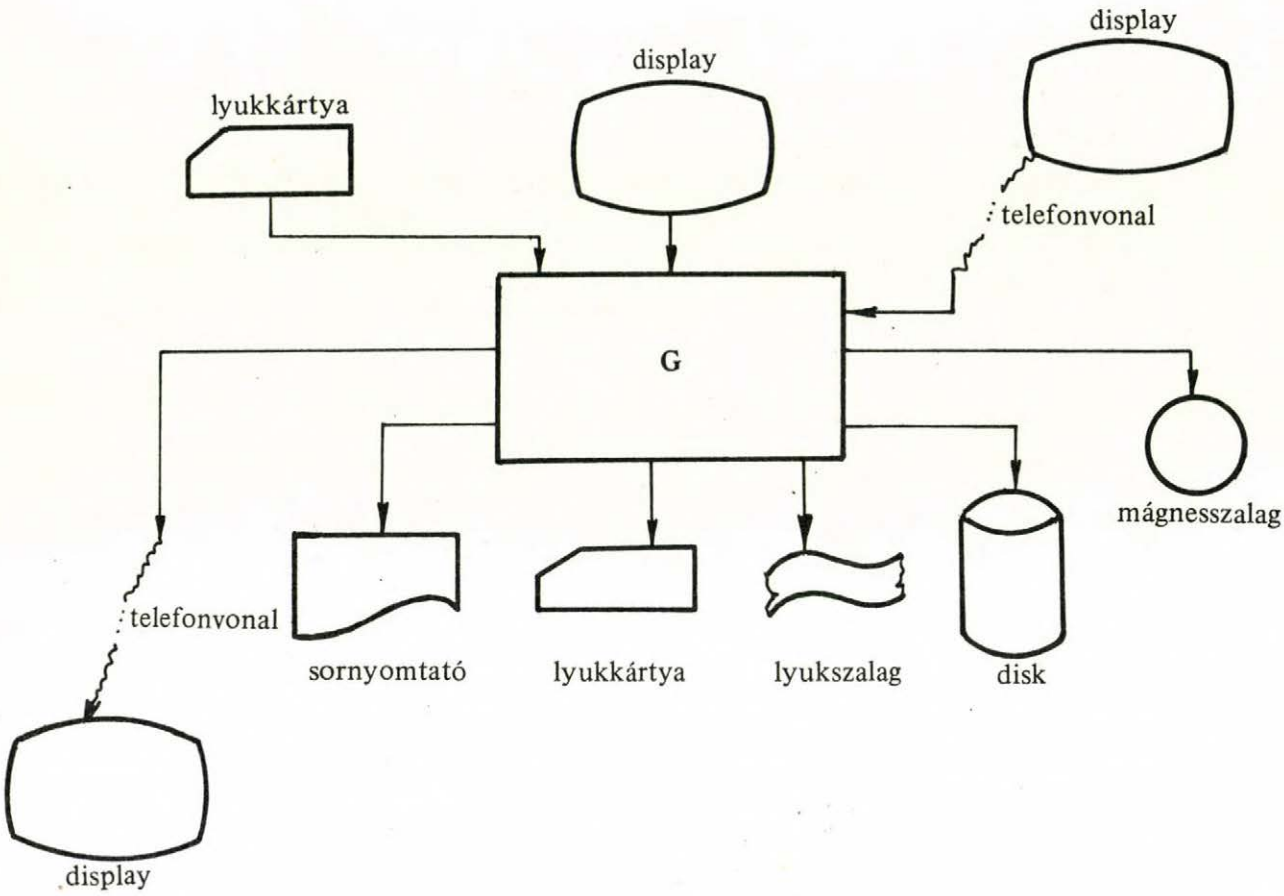
1./ Töltsük fel a T-FILE-t /szövegekkel is/

2./ Generáljuk a kritikus utakat, a tartalékokat, a kezdéseket

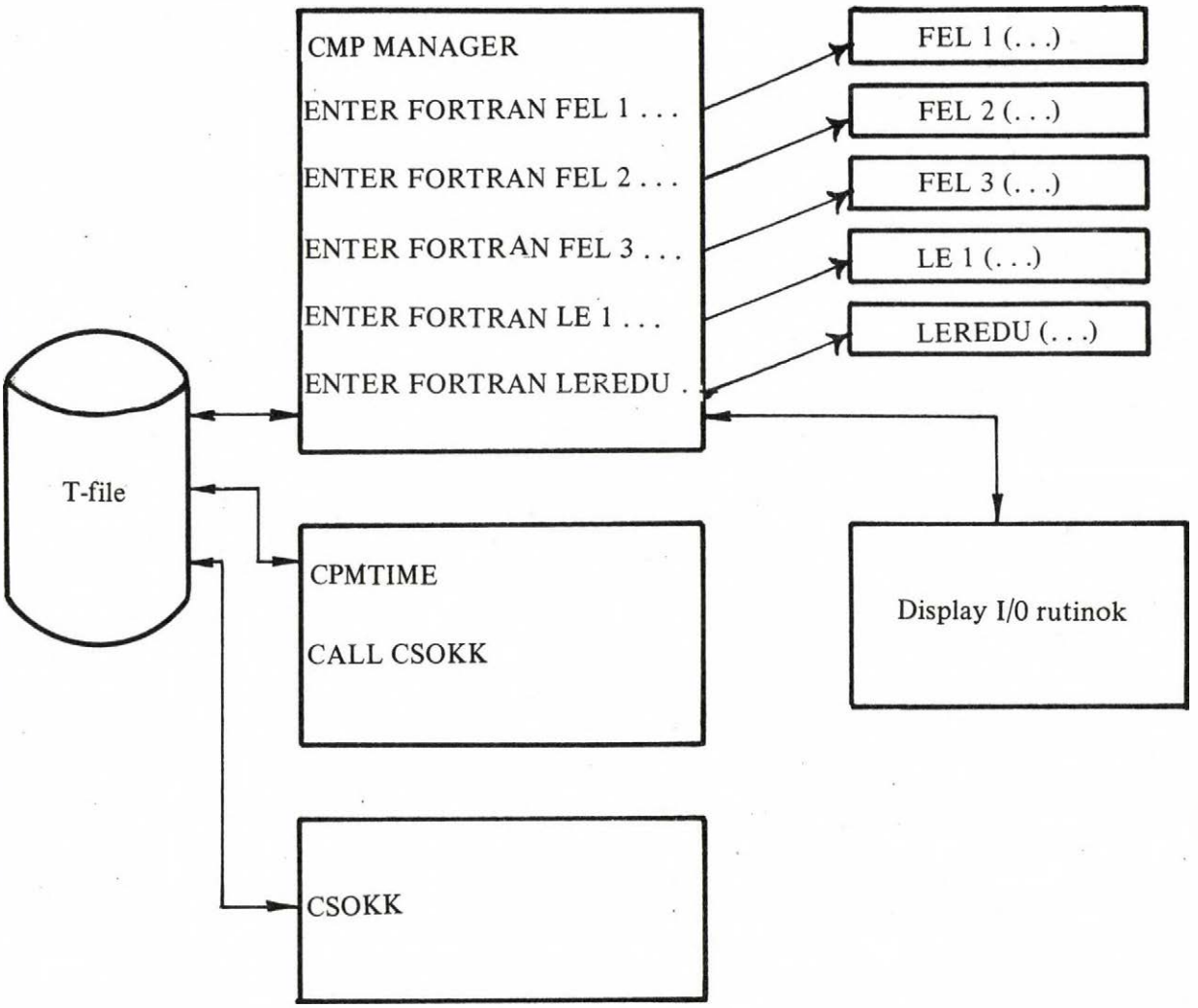
3./ 1974. jan. 1. 12 órára vonatkozóan listáztassuk ki az 01 kódszámu szervezethez tartozó tevékenységeket, ill. listáztassuk ki az 1974. jan. 15-én folyó tevékenységeket

A feladathoz tartozó vezérkártyák és azok kötelező sorrendje:

```
$JØB...
$SCHED      Operációs rendszer vezérlőkártyái
$*DEFCØ,CM...
$X,CM       Leforditott programot beteszi a gépbe
*NETUP
Adatkártyák
ENDATA
*TEXTUP
Szövegkártyák
ENDATA
* CPM
Hálózati paraméterek
*MINISØRT
/TIMEBASE, 74/01/01/12.0
/LØØK, HATØ=1
/SØRT,KEL
/ØUTPUT,ALL
/LØØK,INK  74/01/15/00.0, MAXB 74/01/15/00.0
/SØRT,KEL
/ØUTPUT,KEL,VEL,IDOK,MINK,MAXB,PARC,HATØ
77          Operációs rendszernek jelzés: futás vége
78
```



1. ábra
I/O lehetőségek



2. ábra
Adatkommunikáció a különböző modulok között

IRODALOMJEGYZÉK

[1] Elmaghraby, S.E., "An Algebra for the Analysis of Generalized Activity Networks", Manag. Sci., 10/1964/, 326-352.

[2] Ford, L.R., "Network Flow Theory", The Rand Corporation, Paper P-923, jul. 4 /1956/.

[3] Fox, B.L., "Calculating kth Shortest Path", Infor 11./1973/, 66-70.

[4] Fulkerson, D.R., "A Network Flow Computation for Project Cost Curves", Manag. Sci. 7/1961/, 167-178.

[5[Kelley, J.E., "Critical Path Planning and Scheduling Mathematical Basis", Opns Res. 9/1961/, 296-320.

[6] Klafszky, E., "Hálózati folyamatok", Bólyai J. Mat. Társulat /1969/, pp. 263.

[7] Knuth, E., "The Art of Computer Programming", Vol.1. Addision-Wesley P.C. 1968, pp.634.

[8] Maes, M. - Tengels, "Potentially Critical Path in Indeterminate Times Schedulling Graph", Proceeding of Project Planning by Network Analysis, North-Holland Publ. Cy., Amsterdam /1969/ p. 202-206.

[9] Minty, G.J., "A Comment on the Shortest Route Problem", Opns, Res.5/1957/, 725-742.

[10] Pritsker, A.A.B., "GERT: Graphical Evaluation and Review Technique Part I. Fundamental, Part II., Probabilistic and Industrial Engeneering Application", J. of Ind. Eng. 17/1966/.

[11] Pritsker, A.A.B., "The Status of GERT", Proceedings of Project Planning by Netw. Analysis 147-153/1969/.

[12] Yen, J.Y., "Finding the K Shortest Loopless Path in a Network", Management Sci., 17/1971/, 712-716.

A TANULMÁNYOK sorozatban eddig megjeletek:

1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundás, $\{\wedge, \vee, \neg\}$ vagy {NOR} vagy {NAND} bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállitására

2/1973 Вашкеви Иштван: Расчленение многосвязных промышленных процессов с помощью вычислительных машин

3/1973 Ádám György: A számitógépipar helyzete 1972 második felében

4/1973 Bányász Csilla: Identification in the Presence of Drift

5/1973* Gyürki J.-Laufer J.-Girnt M.-Somló J.: Optimalizáló adaptiv szerszámgépirányitási rendszerek

6/1973 Szelke E.-Tóth K.: Felhasználói Kézikönyv /USER MANUAL/ a folytonos Rendszerek Szimulációjára készült ANDISIM programnyelvhez

7/1973 Legendi Tamás: A CHANGE nyelv/multiprocesszor

8/1973 Klafszky Emil: Geometriai programozás és néhány alkalmazása

9/1973 R.Narasimhan: Picture Processing Using Pax

10/1973 Dibuz Á.-Gáspár J-Várszegi S.: MANU-WRAP hátlaphuzalozó MSI-TESTER integrált áramköröket mérő, TESTOMAT-C logikai hálózatokat vizsgáló berendezések ismertetése

11/1973 Matolcsi Tamás: Az optimum-számitás egy uj módszeréről

12/1973 Makroprocesszorok, programozási nyelvek. Cikkgyüjtemény az NJSzT és SzTAKI közös kiadásában. Szerkesztette: Legendi Tamás

13/1973 Jedlovszky Pál: Uj módszer bonyolult rektifikáló oszlopok vegyészmérnöki számitására

14/1973 Bakó András: MTA kutatóintézeteinek bérszámfejtése számitógéppel

15/1973 Ádám György: Kelet-nyugati kapcsolatok a számitógépiparban

16/1973 Fidrich I.-Uzsoky M.: LIDI-72 listakezelő rendszer a Digitális Osztályon, 1972. évi változat

17/1974 Gyürki József: Adaptiv termelésprogramozó rendszer /APS/ termelőmühelyek irányitására

18/1974 Pikler Gyula: MINI-számitógépes interaktiv alkatrész-programiró rendszer NC szerszámgépek automatikus programozásához

19/1974 Gertler,J.-Sedlak,J.: Software for process control

20/1974 Vámos,T.-Vassy,Z.: Industrial Pattern Recognition Experiment - A Syntax Aided Approach

21/1974 A KGST I. -15-1.: "Diszkrét rendszerek automatikus vezérlése" c. témában 1973. februárban rendezett szeminárium előadásai

22/1974 Arató,M.-Benczur,A.-Krámli,A.-Pergel,J.: Stochastic Processes, Part I.

23/1974 Benkó S.-Renner G.: Erősen telitett mágneses körök számitógépes tervezési módszerei

24/1974 Kovács György-Franta Lászlóné: Programcsomagok elektronikus berendezések hátlaphuzalozásának tervezésére

25/1973 Járdán R. Kálmán: Háromfázisu tirisztoros inverterek állandosult tranziens jelenségei és belső impedanciája

26/1974 Geregely József: Numerikus módszerek sparse mátrixokra

27/1974 Somló János: Analitikus optimalizálás

28/1974 Vámos Tibor: Tárgyfelismerési kisérlet nyelvi módszerekkel

29/1974 Móricz Péter:Vegyészmérnöki számitási módszerek fázisegyensulyok és kémiai egyensulyok vizsgálatára

30/1974 Vassy, Z.-Vámos,T.: The Budapest Robot - Pragmatic Intelligence

31/1975 Nagy István: Frekvenciaosztásos kőzépfrekvenciás inverterek elmélete

32/1975 Singer D.,Borossay Gy.,Koltai T.: Gázhálozatok optimális irányitása különös tekintettel a Fővárosi Gázmüvek hálózataira

33/1975 Vámos, T.-Vassy,Z.: Limited and Pragmatic Robot Intelligence
Mérő,L.-Vassy,Z.: A Simplified and Fastened Version of the Hueckel Operator for Finding Optimal Edges in Pictures
Галло В.: Программа для распознавания геометрических образов, основанная на лингвистическом методе описания и анализа геометрических структур

34/1975 László Nemes: Pattern Identification Method for Industrial Robots by Extracting the Main Features of Objects

35/1975 Garádi - Krámli - Ratkó - Ruda: Statisztikai és számitástechnikai módszerek alkalmazása kórházi morbiditás vizsgálatokban

36/1975 Renner Gábor: Elektromágneses tér számitása nagyhőmérsékletü anyagban

37/1975 Edgardo Felipe: Specification problems of a process control display

38/1975 Hajnal Andrásné: Nemlineáris egyenletrendszerek megoldási módszerei

39/1975* A. Abd El-Sattar: Control of induction motor by three phase thyristor connections in the secondary circuit

40/1975 Gerhardt Géza: QDP Grafikus interaktiv szubrutinok a CDC 3300-GD'71 grafikus konfigurációra

41/1975 Arató,M.-Benczur,A.-Krámli,A.-Pergel,J.: Stochastic Processes, Part II.

42/1975 Arató M.: Fejezetek a matematikai statisztikából számitógépes alkalmazásokkal

43/1975 Matavovszky Tibor - dr. Pásztorné, Varga Katalin: Programrendszer Boole-függvény együttes egyszerüsitésére vagy minimalizálására

44/1975 Bacsó Nándorné: Pneumatikus áramköri hazardok

45/1975 Varga András: Ellenpárhuzamos félvezetőpárokkal vezérelt aszinkronmotoros hajtások számitási módszerei

46/1976 Galántai Aurél: Egylépéses módszerek lokális hibabecslései

47/1976 Abaffy József: A feltétel nélküli függvényminimalizálás kvadratikus befejezésü módszerei

48/1976 Strehó Mária: Stiff tipusu közönséges differenciálegyenletek megoldásáról

49/1976 Gerencsér László: Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel

50/1976 Robert Treer: A syntax macro definition language

A *-gal jelölt kivételével a sorozat kötetei megrendelehtők az Intézet könyvtáránál /Budapest, XIII. Victor Hugo u. 18-22./

Jelen dolgozat az 5.10.5 számu intézeti témában került kidolgozásra