

Az erőforrás allokáció SZELET módszere

Az építőiparban alkalmazott erőforrás allokáló módszerek és a SZELET

A hálós programozásban alkalmazott, ismert erőforrás allokáló eljárások lényege: az erőforrások időszakos elégtelenségének feloldása érdekében a technikailag párhuzamosan is elhelyezhető folyamatok közül halasszuk későbbre azt, amelyik bizonyos követelmények szerint legkevésbé kritikusnak minősül.

A kivitelező építőipar jellegzetességeihez alkalmazkodó ERALL [1] és VOP [2] üzemeltetésével szerzett tapasztalatok felhívják a figyelmet az adatok és a számítás pontosságának ellentmondására: a folyamatok erőforrásigényét hiába adják jó kipróbált normák, ha a rendelkezésre álló erőforrásokra gyakran csak 10—20% szórású becslés adható. Hiába a modell kifogástalan — és ennek megfelelően számításigényes — megoldása, ha a körülmények alakulása a feltételeket gyorsan változtatja (munkaerővándorlás, váratlan anyaghiány, jó szervezéssel sem elkerülhető szeles idő).

Gondolnunk kell maguknak a hálóknak a pontosságára is: hogy kezelhető méretűek maradjanak, számos összefüggés — elsősorban együtt futó tevékenységek közötti kapcsolat — kimarad belőlük. A hálók építői általában kihaznárlják a műszaki ütemező program tulajdonságait (pl.: kódolással jelezhető, hogy a tartalékidőt, ha van, a folyamat elé vagy mögé kérjük [3]). Ha e hálón kívül jelzett tulajdonságoktól eltekintve használjuk fel az olykor csak látszólagos tartalékidőket, az eredmény műszakilag értelmetlen lehet.

Olyan heurisztikus megoldást keresünk, mely

- érzéketlen a rövid időn belüli átütemezéssel feloldható erőforrás elégtelenségre,
- nem bolygatja a folyamatok egymáshoz viszonyított elhelyezkedését,
- sorban ütemezi a hálókat.

A rövid időszakon belül végrehajtandó átütemezés csak az operatív termelés-irányítás feladata lehet. A havi vagy negyedévi számítógépes feldolgozásokkal elsősorban arra kell törekedni, hogy az operatív termelésirányításra csak ilyen feladat maradjon.

A folyamatoknak az erőforrásokat is figyelő időzítése helyett a hálót láncá tömörítjük és a láncot ütemezzük. A csak műszakilag megalapozott előütemezés eredményét időegységenként (pl.: naponta) összevonjuk, így kapjuk a módszer névadóját: a szeletet. Az egyes szeletek időtartamának változtatásával próbáljuk a túlterheléseket felszámolni. A folyamatok végpontjaiként az operatív irányítás számára azokat az időpontokat ajánljuk, melyek a szeleteket az eredeti arányok szerint osztják.

A soros ütemezés mellett szól, hogy általános túlterhelés esetén (amit az építőipari kereslet és teljesítmény rossz mérlege mutat), így is bízhatunk a viszonylag bőséges erőforrások kihasználásában (a sor végén is lesznek olyan hálók, melyek lekötik) — lényegesen kisebb számításigénnyel.

A SZELET módszer gondolatmenete a következő:

I. Az allokálás időhorizontját szakaszokra osztjuk (pl. hónapokra, később negyedévekre), időszakonként megadunk a vizsgálandó erőforrásfajtákra két mennyiségi határt. Az első egy ideális érték, melyet a második, kemény korlátig akkor lépünk túl, ha ez kevés, az viszont elég a határidők betartásához. (Ha a második sem elég, akkor az ideális korlát betartásával igyekszünk mielőbb végezni.)

II. Sorra vesszük a műszakilag már ütemezett hálókat, s ha tehetjük, eredeti ütemezésüket meghagyva, kezdetüket megfelelően időzítve illesztjük a korábbiakhoz.

III. Ha az eredeti ütemezés erőforráshiány miatt nem tartható, akkor, — ha a háló alhálóból épül fel —, teljes alhálókat mozgatva igyekszünk a korlátokhoz igazodni.

IV. Ha nem a III.-beli eset áll fenn, vagy pedig az igazítás után is marad akár erőforráshiány, akár határidő-túllépés, akkor a konfliktusokat időszakonként sorban próbáljuk rendezni.

Első eszközünk a gyorsítás: egy korábbi időszakba eggyel több szeletet ütemezünk, az ezt követők szeleteit korábbra csúsztatjuk.

V. Ha konfliktust találunk egy olyan időszakban, mely előtt nem gyorsíthatunk, akkor ha hiányzik erőforrás, lassítással próbálkozhatunk: az utolsó szeletet kihagyjuk ebből az időszakból, a következők szeleteit későbbre csúsztatjuk.

VI. A gyorsítást-lassítást — I-gyel összhangban — először az ideális korlátok betartása érdekében végezzük. Ha nem sikerül minden konfliktust megoldanunk, akkor megpróbálunk megoldást keresni a kemény korlát adta lehetőségekkel is élve. Ha az sem elegendő, akkor folytatjuk az ideális korlátokkal félbemaradt számítást (lényegében csak az erőforrás-korlátok betartása érdekében).

A SZELET módszer algoritmus

A SZELET módszer alkalmazása esetén elvégzendő legfontosabb teendők néhány eljárásba foglalhatók:

A) Szeletképzés

A még nem gyorsított-lassított háló vetületét az időtengelyen egyenlő hosszú, diszjunkt időintervallumokkal fedjük le. Az intervallumok hossza lehet hálónként eltérő (pl. a később kezdhető munkák hálójánál nagyobb), de osztója kell legyen a háló legkorábbi kezdéséig be nem fejeződő minden I. szerinti időszak hosszának. Ez a hosszúság a háló minden szeletének induló hossza.

A hálót alkotó tevékenységek erőforrásigényének időarányos részét számítjuk a vetületét lefedő intervallumokhoz tartozó szeletekhez. Ezeknek az időarányos részeknek összegei adják az egyes szeletek erőforrásigényét.

Ha egy intervallum belsejébe esik egy olyan esemény vetülete, melyhez legkorábbi kezdési vagy legkésőbbi befejezési igény tartozik, akkor ezt az igényt az intervallumhoz tartozó szelet elejére, ill. végére vonatkoztatjuk. (Ez feszítettebb az eredetinél.) Az intervallumok illeszkedési pontjaira eső legkorábbi kezdés kikötést a későbbinek, a legkorábbi befejezési kikötést pedig a korábbiinak megfelelő szeletre értjük. Így kaphatunk a szeletekhez kezdési, ill. befejezési időpontra vonatkozó korlátozást.

Az egyes intervallumokba pozitív hosszúságú vetülettel eső tevékenységek közül a legkevésbé rövidíthető, ill. a legkevésbé nyújtható változtathatósági viszonzyszáma adja meg, hogy hányadára nyomható össze, ill. hányszorosára nyújtható a megfelelő szelet. Ebből kapjuk meg a szelet minimális és maximális hosszát.

A szomszédos intervallumok mindegyikébe pozitív hosszúságú vetülettel eső tevékenységek nyújthatóságától függ, megszakítható-e a háló a megfelelő szelet-határon. Ha mind korlátlanul nyújtható, akkor megszakítható. (Következésképp: korlátlanul nyújtható szeletek között a háló megszakítható, de nem feltétlenül csak ott.)

Minden szelethez feljegyezzük a megfelelő intervallumban kezdődő és végződő vetülettel tevékenységeket, továbbá, hogy a végpont az intervallumot milyen arányban osztja. Az intervallum-határra eső kezdőpont az ott kezdődő, a végpont az ott végződő intervallumhoz számít.

Az *A)* eljárással összhangban: Szelet egy hálónak két — előre nem rögzített — időpont közötti része. Van, ill. lehet:

- időtartama (induló, minimális, maximális, tényleges),
- erőforrásigénye,
- korlát kezdetének és végének időpontjára,
- a szomszédjától elszakítható vagy el nem szakítható.

Az egyes időszakok mindig valahány teljes szeletet tartalmaznak.

B) Simítás

Akkor mondjuk, hogy egy háló alhálókból áll, ha bizonyos fiktív tevékenységeket elhagyva belőle, egymással össze nem függő részhálókra esik szét, és az e részhálókat, mint csúcspontokat, ill. az előbbi fiktív tevékenységeket, mint éleket tartalmazó gráfnak nincs irányított körútja. A fenti részhálókat nevezzük alhálóknak.

Az alhálók sorrendjét (sorszámozását) csak az említett fiktív tevékenységekkel adott, részben rendezés korlátozza.

A simító eljárással úgy próbáljuk elhelyezni az alhálókat, hogy a legnagyobb erőforrás-határ túllépés vagy kihasználás minimális legyen. Feltételezzük, hogy működik egy kiszolgáló eljárás, mely az alhálók sorozatához, ismerve az első elhelyezését, megadja a soronlevő szóabajövő kezdéseit. Szóabajön, a korábbiak helyzetétől is esetleg függő legkorábbi kezdés és a teljes időszaknyival későbbiek is, mindaddig, míg a többi alháló is határidő-tartóan elhelyezhető. (Ha vannak határidő-tartóan eleve el nem helyezhető, akkor azokat a tényleges simítás előtt legkorábbi kezdésükre helyezi.)

A különböző elhelyezések összehasonlítására szolgáló büntető függvény az időszakonkénti igények és a rendelkezésre álló erőforrások hányadosa exponenciális függvényeiből képzett összeg, erőforrásfajtánként súlyozva. Elég nagy alapot választva a legnagyobb kitevőjű tagokhoz képest a többi elhanyagolható; a büntető függvény csak a maximumra érzékeny.

Az összes szóabajóhető kezdésrendszert az esetek nagy száma miatt nem tudjuk értékelni, itt is soros eljárást választottunk. Rendre vesszük az alhálót, alhálónként a szóabajó kezdéseket. A következő alháló szóabajó kezdéseit már csak az előbbieknél a fenti függvénnyel értékelt, legfeljebb adott számnyi legjobb sorozatához nézzük meg. E szám megválasztásával a számítási idő befolyásolható. Első alkalmakkor még remény van a szeletenkénti ütemezés elkerülésére, ekkor érdemes viszonylag nagyobbban választani, később a végeredményt nem nagyon befolyásolhatja, célszerűbb kicsinek venni.

Sorozat	Érték	1.	2.	3.	n.
		Alháló kezdése			
0.					
1.					
2.					
s.					

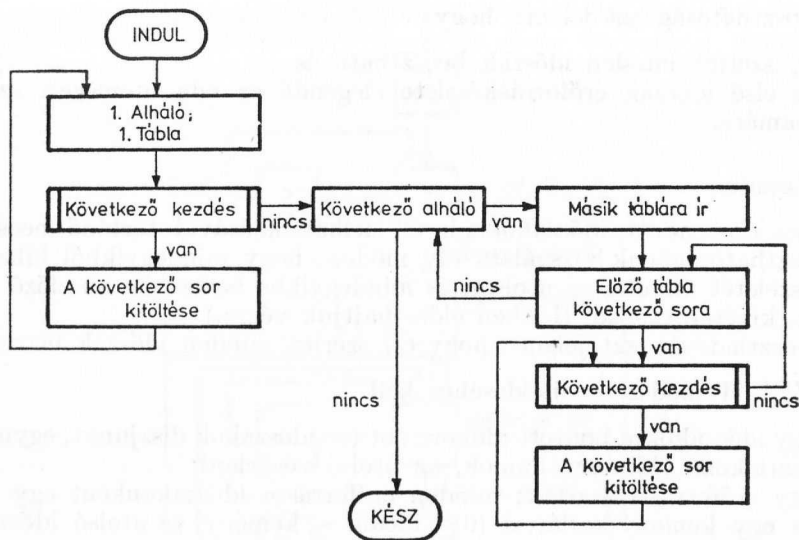
1. ábra

A számítást elvégezhetjük két, az 1. ábrán látható táblázat vezetésével. S sorozat számára tartunk helyet mindkettőben. A O . oszlopba az aktuális büntető függvény értékeit írjuk, a többiekbe a sorszámuknak megfelelő alháló kezdési időpontját.

Először az $1, 2, \dots, S$ sorba írunk; a O -ba csak akkor, ha az előbbieket mind foglaltak. Minden sorhoz, az aktuális alháló kezdő-időpontjának beírása után, kiszámítjuk a büntető függvény értékét is. A O . sort azonnal kicseréljük a legkedvezőtlenebbel (ha maga az, akkor hagyjuk helyben), s ezzel felszabadítjuk a következő számításhoz.

Az első alhálónál sorban kérjük a szóabajó kezdési időpontokat, az említett kiszolgáló eljárástól, s a hozzátartozó büntetéssel együtt beírjuk az 1. táblázat következő sorába (az $S + 1, S + 2, \dots$ -t a O -ba, és azonnal cseréljük a legkedvezőtlenebbel).

A további alhálónál váltakozva a $2, 1, 2, \dots$ táblázatot töltjük ki a másik alapján. Előlről foglaljuk el sorait (ha megtelt, akkor a O -ba írunk). Először a másik táblázat 1. sorozatához, majd a 2.-hoz, végül az S -hez kérjük a soronvő alháló szóabajó kezdési időpontjait (2. ábra).



2. ábra

C) Időszak-beosztás

Feladata egy megadott időszakba adott, egymást követő szeletek elhelyezése, ill. elhelyezésük lehetőségének vizsgálata. Ismeri az adott szeletek előtti és utáni szelet záró, ill. kezdő időpontját (időbeli szakadáskor van jelentősége).

A lehetőséget a következők betarthatósága jelenti:

- nem engedélyez szakadást ott, ahol az tilos;
- az időszakok hosszát a rá vonatkozó korlátok között tartja;
- betartja a kezdésekre vonatkozó alsó időpontkorlátokat.

Ha a lehetőség adott, akkor a szeletek elhelyeztetethetők a megadott időszakba. Olyan elhelyezést keresünk, mely egymás után érvényesíti a következő célokat:

- betartja a lehetőség-vizsgálatkor említett követelményeket;
- minimálja az időszakba eső határidők túllépésének darabszámát (ha nem tarthatjuk a határidőt, akkor a túllépés mértékének csökkentése már nem elsődleges cél);
- minimálja a szakadások számát;
- minimálja a szeletek időtartamának szórását.

D) Gyorsítás

Szomszédos, adott időszakoknak C) felhasználásával végzett beosztása, ill. beoszthatóságának vizsgálata olymódon, hogy mindegyik befogadja a következő időszak első szeletét, (kivéve az utolsó) és mindegyikből kihagyjuk első szeletét, kivétel az első. (Időben visszafelé hajtjuk végre.)

A beoszthatóság azt jelenti, hogy

- C) szerint minden időszak beosztható és
- az első időszak erőforráskészlete elegendő az oda ütemezett szeletek számára.

E) Lassítás

Szomszédos, adott időszakoknak C) felhasználásával történő beosztása, ill. beoszthatóságának vizsgálata oly módon, hogy mindegyikből kihagyjuk utolsó szeletét (kivétel az utolsó), és mindegyikbe bevesszük az előző utolsó szeletét, kivétel az első. (Időben előre hajtjuk végre.)

A beoszthatóság azt jelenti, hogy C) szerint minden időszak beosztható.

A SZELET módszer működéséhez kell

- egy időszakra bontott időhorizont (az időszakok diszjunkt, egymáshoz csatlakozó időintervallumok, az utolsó végtelen);
- egy erőforrás-választék; minden erőforrásra időszakonként egy ideális és egy kemény korláttal ($0 \leq \text{ideális} \leq \text{kemény}$, az utolsó időszakban mind végtelen);
- egy elő-ütemezett háló-sorozat (vagy akár vonalas ütemterv sorozat); csak a hálók tevékenységeivel foglalkozunk, melyeknek
 - = adott az időtartamuk, kezdésüknek a háló elejéhez viszonyított időpontja;
 - = adott nyújthatóságuk és összenyomhatóságuk mértéke;
 - = adott az erőforrás-választékba tartozó erőforrásokra vonatkozó igényük;
 - = kezdésére legkorábbi, befejezésére legkésőbbi időpont írható elő (legalább egy tevékenységre kell legkorábbi kezdési időpont);
- hálónként egy szelethossz, mely osztója minden olyan időszak hosszának, melybe e háló belenyúlhat.

Ha bizonyos hálókat alhálónként adunk meg, akkor a tevékenységek kezdetét az alháló elejéhez is viszonyítjuk és az alhálókat sorszámozzuk. Az elő-ütemezés minden alhálót időzít, pl. legkorábbi kezdésére.

A SZELET módszer az alábbiak szerint működik:

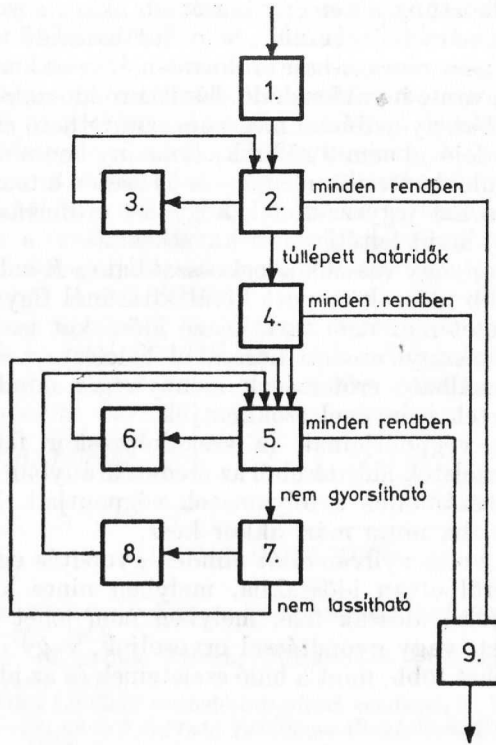
1. A soronlevő hálót először úgy helyezzük az időtengelyre, hogy minden tevékenységére teljesüljön az esetleges legkorábbi kezdési előírás, legalább egyikre pontosan; majd szeleteit képezzük A -val.

Rendelkezésre bocsátjuk időszakonként az ideális korlátból még megmaradt erőforrásokat.

2. Ha valamelyik szelet nem tartja befejezési határidejét, akkor 4 következik; egyébként megvizsgáljuk, elegendő erőforrás áll-e rendelkezésünkre minden időszakban a szeletek igényeinek kielégítésére. Ha igen, akkor 9 következik.

3. Minden szeletet annyival későbbre tolunk, amennyi az első olyan időszak hossza, melyben a hálónak van szelete, 2 következik.

4. Ha a háló alhálókból áll, akkor B -vel alhálóit újra időzítjük, szeleteit képezzük A -val, s ha így már mind tartja befejezési határidejét és az időszakok erőforrásai is elegendők, akkor 9 következik, egyébként 5 jön. Ha a háló nem



3. ábra

bomlik alhálókra, akkor visszaállítjuk (3 után) az 1-ben meghatározott időpontokat.

5. Megkeressük az első olyan, a keresésből 7-ben ki nem zárt, h. időszakot, melyben konfliktus van (akár véghatáridejét túllépő szelet, akár erőforrás-hiány). Ha ilyen nem találunk, akkor 9 következik. Ha találunk, akkor D -vel megnézzük, van-e h -t tartalmazó gyorsítható időszak-csoport. Ha nincs, akkor 7 következik.

6. Ha van, akkor vesszük a leghosszabbat és D -vel gyorsítjuk. 5 következik. (A leghosszabb időszak-csoport kiválasztásánál figyelembe vesszük az utolsó, a hálóból szeletet még nem tartalmazó időszakokat is az idővel ellentétes sorrendben, ha az utánuk következőben van annyi szelet, amennyi eredeti hosszával belefér, vagy ha a rendelkezésre álló erőforrás kevés a plusz egy szelet befogadásához.)

7. Ha az 5-ben kiválasztott h időszakban van véghatáridejét túllépő szelet, akkor (mivel a konfliktust itt már meg nem oldhatjuk)

a) ha még nem próbálkoztunk a kemény korlátok felhasználásával, rendelkezésre bocsátjuk a kemény korlátig még fennmaradt erőforrásokat és újra-kezdjük a 4 utáni állapotból, 5-től a számítást;

b) ha éppen a kemény korlát felhasználásával jutottunk ide, akkor vissza-állítjuk az ideális korlátokat és az azzal félbemaradt számítást úgy folytatjuk, mint eredetileg is tettük volna, ha már akkor sem lehetett volna félbeszakítani;

c) ha már próbálkoztunk a kemény korláttal, akkor a számítást nem lehet félbeszakítani; 5 számára feljegyezzük, hogy h-t határidő-túllépés miatt nem szabad kiválasztani; ha nincs h-ban erőforrás-hiány, akkor 5 következik.

Megnézzük E -vel, van-e h-val kezdődő, lassítható időszak-csoport (ide akkor jut a vezérlés, ha h-ban gyorsítással meg nem szüntethető erőforrás-hiány van és szeleteinek határidejével nem törődünk). Amennyiben nincs, ismét az előbbi a), b) és c) esetet különböztetjük meg, a) és b) esetén a teendő is változatlan, c) esetén 5 számára azt jegyezzük fel, hogy h-t erőforrás-hiány miatt nem szabad kiválasztani, majd feltétlenül 5 következik.

8. Amennyiben van, úgy vesszük a leghosszabbat és E -vel lassítjuk. 5 következik. (A leghosszabb időszak-csoport kiválasztásánál figyelembe vesszük az első, a hálóból szeletet már nem tartalmazó időszakot is, ha a megelőzőben van annyi szelet, amennyi eredeti hosszával belefér).

9. A még felhasználható erőforrások mennyiségét minden időszakban az oda ütemezett szeletek igényével csökkentjük.

A tevékenységek végpontjainak (a szeletképzéskor feljegyzett) szeleten belüli helyéből és a szeletet időzítéséből az eredeti arányban osztó pontok meghatározásával visszaszámoljuk a folyamatok végpontjait. Ha van még háló, akkor 1 következik, ha nincs már, akkor kész.

A módszer véges volta nyilvánvaló: minden gyorsítás egy további szeletet juttat a háló elejéről olyan időszakba, melyben nincs konfliktus; minden lassítás lépés az utolsó időszak felé, melyben nem lehet erőforrás-hiány (a határidő-túllépéseket, vagy gyorsítással orvosoljuk, vagy eltekintünk tőlük). E lépésekből nem lehet több, mint a háló szeleteinek és az időszakok számának szorzata.

Kiegészítő megjegyzések:

Ha minden hálót egyetlen összefoglaló háló alhálójaként kezeljük, akkor a simító eljárás maga is megvalósít egy, a tevékenységek időtartamát nem változtató, a bevezetőben említett követelményeknek megfelelő erőforrás-allokálást. Ebben az esetben célszerű megőriztetni a kezdés-sorozatokat táblánkénti legjobb értékelését: ha a végül legjobbaknak mutatkozóak mindig távol álltak a kiselejtezéstől, akkor ez a tény növeli bizalmunkat — előfordulhat, hogy az összes lehetséges esetet megvizsgálva is ugyanazt az eredményt kaptuk volna.

Megtehetjük, hogy az összes háló minden szóbajöhető kezdését mégegyszer megvizsgáljuk (közben a többieket a legjobb sorozat szerint rögzítve); ha nem akad jobb megoldás: lokális optimumot találtunk.

A simító eljárás önálló alkalmazásával elsősorban a kezdési, másodsorban a befejezési határidők betartása érhető el. (Ha nem mondanak ellent az előütemezésnek, mindet tartani fogja.) Következő szempont csupán a relatív erőforráshatár túllépések maximumának leszorítása.

A szelet módszer a valóságos lehetőségeket jobban megközelítő eszközeivel (a tevékenységek időtartama nem merev) jobb eredményt ígér. Betartja a legkorábbi kezdésekre vonatkozó kikötéseket, a tevékenységek időtartamát csak a megengedett mértékben változtatja, megszakítást csak ott tervez, ahol az műszakilag és szervezésileg megengedett. Felvehető pl. a meg nem szakítható hálórészrel párhuzamosan egy erőforrást nem igénylő, összenyomható,

nagy mértékben nyújtható, de meg nem szakítható fiktív tevékenység. Igyekeznek betartani az erőforráshatárokat, túlterheléssel nem kezd, de megszakíthatatlanság miatt a későbbi időszakok esetleg már meglévő erőforráshiányát fokozhatja (a lehetséges minimális intenzitáshoz tartozó munka erőforrás-szükségletével). A befejezési határidők betartása érdekében akkor alkalmaz emelt erőforrásszintet, ha ez a háló minden határidejének betartását biztosítja. Ellenkező esetben a többletet fenntartja a sorban később jövő olyan hálónak, melyeknél felhasználva majd minden határidő tartható lesz.

Az előütemezés, a tevékenységek egymáshoz viszonyított helyzetének merevítése miatt, bármilyen eljárás lehet. A módszer inputja, mint már jeleztük, vonalas ütemtervből is előállítható.

A szelet módszer igyekszik összhangot teremteni az adatok pontossága (pontatlansága) és a számítási idő között. Az időszakok és szeletek hosszának növelésével igen durva számításokra is alkalmas, egészen addig, amíg számításnak egyáltalán értelme van.

(Beérkezett: 1976. február 17.)

IRODALOMJEGYZÉK

1. BALLA MIHÁLYNÉ—SZOLNOKY ANTAL: Beruházások irányítása ERALL-2 módszerrel, SZÁMGÉP Kiadvány 1968.
2. BALLA MIHÁLYNÉ—SZOLNOKY ANTAL: Útmutató a VOP módszer alkalmazásához, SZÁMGÉP Kiadvány 1970.
3. Tájékoztató a KOMPLETER termelésirányítási rendszerről, ÉGSZI Kiadvány 1974.
4. HEGEDÜS GÁBOR—SZOLNOKY ANTAL: Erőforrás-allokáció SZELET módszerrel, Információ Elektronika 1976. 2.

RESOURCE ALLOCATION BY THE SLICE METHOD

The essence of the known resource allocation methods is that in order to solve the temporary insufficiency of resources, out of several, technically parallel processes that one should be postponed which is the least critical according to certain requirements.

The essence of the authors' new method is as follows:

Such a heuristic solution was sought as would be insensitive to insufficiencies of resources, which are solvable by a short term rescheduling; on the other hand, it would not interfere with the relative position of processes.

The short term rescheduling can only be the duty of operative production control. Monthly or quarterly computerized programming must, first of all, concentrate on leaving only such tasks for the operative production control.

Instead of programming the processes with full regard to resource limitations, we condense the network into a chain and schedule this latter. The results of the preliminary schedule based only on technical conditions is aggregated by time periods (say, days) and this way we get "slices", after which the method is called. By varying the length of the slices we make an attempt to eliminate overloads. As the end points of the processes, we suggest dates to the operative control that divide the slices according to the original proportions.

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ С ПОМОЩЬЮ «КУСКОВОГО» МЕТОДА

Сущность применяемых в сетевом планировании известных методов распределения: в интересах преодоления периодической недостаточности ресурсов среди параллельно совместимых в техническом отношении процессов производится отсрочка того из них, который с точки зрения определенных требований квалифицируется наименее критически.

Сущность разработанного авторами нового метода:

Поиск эвристического решения, которое

- нечувствительно к недостаточности ресурсов, которое можно преодолеть перестройкой графиков в течение короткого времени,
- не нарушает порядок процессов в отношении друг к другу.

Определение перестроек графиков, проводимых в течение короткого времени, может являться задачей только оперативного управления производством. Путем месячных и поквартальных разработок на ЭВМ следует стремиться к тому, чтобы оперативное управление сводилось только к решению этой задачи.

Вместо хронирования, принимающего во внимание ограниченность процессов со стороны ресурсов, производится сжатие сети в цепь и ее хронирование. Результаты обоснованного только в техническом отношении предварительного хронирования по единицам времени (например, по суткам) суммирования по единицам времени (например, по суткам) суммируются, что дает наименование метода — кусок. Путем изменения временного содержания отдельных кусков мы пытаемся ликвидировать перегрузки. В качестве конечных точек процессов оперативному руководству предлагаются те моменты, которые делят куски согласно исходным соотношениям.