

Hálótechnika alkalmazása egy kevert jellegű programozási feladatban



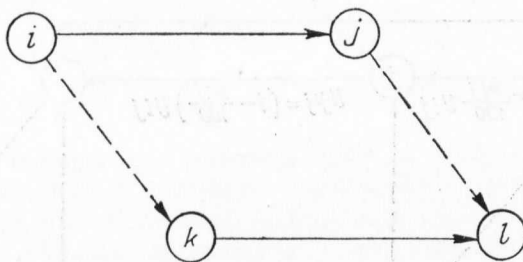
I. Bevezetés

Lapunkban már régebben ismertetésre került a hálótechnikán alapuló ERALL-2 ütemezési algoritmus [1], amelynek tárgyalásánál kitértünk a hálótechnikai alapfogalmakra. Az eljárás azóta is sűrűn alkalmazásra került, de egyrészt — számítógép adta lehetőségek miatt — kibővített formában, (ERALL-3), másrészt olyan formában is, amely a hálótechnikával való alapismeretek további tanulmányozását és az ebből kapott hálóátalakítási szabályok ismeretét kívánta. Erre térünk ki a továbbiakban részletesebben, a matematikai alapok részletes tárgyalásával, míg az alkalmazásoknál csak ismertetjük a módszer felhasználási módját, mivel az eljárás ezen részének közzététele már megtörtént¹ [2].

2. Hálótechnikai bevezető: a hálók és relációk átalakítása

2.1. Legyen a P halmaz egy kaszkádszerű háló, melyen értelmezve van egy \ll közvetlen megelőzési és egy $\ll\ll$ átlapolási részben rendező reláció [1].

Ha $(i, j) \ll (k, l)$, akkor először is vezessük be az (i, k) , valamint a (j, l) látszattevékenységeket. Ezzel az átlapolásos tevékenységek közti kapcsolat szokásos ábrázolásmódját nyerjük (1. ábra).



1. ábra

¹ A kritikus út módszerek állandó fejlesztése létrehozta az átlapolásos tevékenység-típusok technikáját, amely nemcsak a tevékenységek kezdései, hanem befejezéseik közti kapcsolat betartását is igényli, és ezzel módosította a CPM/TIME eredeti algoritmusát is (MPM-módszer). Az erőforrás-allokációs módszerek azonban mindegyik esetben az időszámítási algoritmus módosított változatait igénylik; éppen emiatt nyitva kell hagynunka kérdést, hogy CPM vagy MPM módszer alkalmazása látszik-e célszerűnek, mert az időszámítási algoritmus csak az ütemezés kezdetekor a lebegések számításánál szolgál tájékoztatással; a továbbiakban az erőforráskorlátok miatt ugyanis el kell térnünk az eredetileg kritikus tevékenységektől, és a figyelmet az erőforrások okozta „gazdasági kritikus út”-ra kell fordítani. [3]

E két látszatevékenységet az időszámítási algoritmus fel tudja használni a kritikus út megállapításában, ugyanis ezeket a megfelelő átlapolási % alapján tartammal láthatjuk el, és így a kritikus út számítás közvetlenül végezhető.

Vegyünk most fel egy j' és l' látszateseményt úgy, hogy

$$i \ll j' \ll j,$$

illetve

$$k \ll l' \ll l$$

legyen, és ezzel osszuk fel az (i, j) , illetve (k, l) tevékenységet két-két részre — (i, j') és (j', j) , illetve (k, l') és (l', l) — a következő módon:

$$(i, j') \ll (j', j)$$

$$(i, j') \ll (k, l')$$

$$(k, l') \ll (l', l).$$

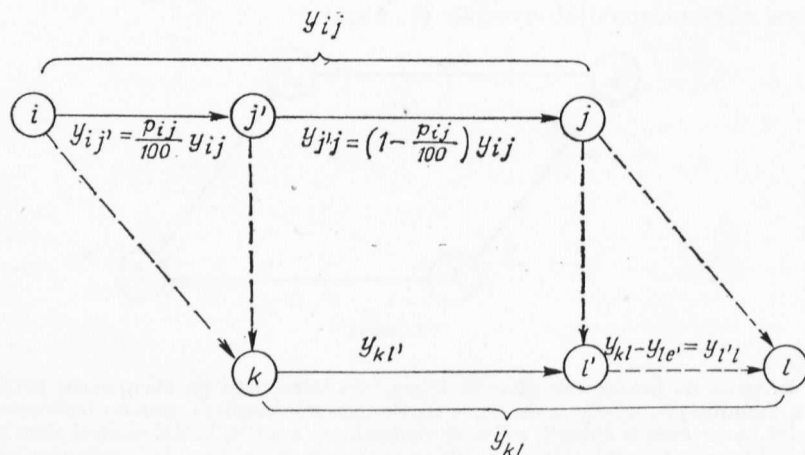
Legyen továbbá:

$$0 < p_{ij} < 100$$

és

$$y_{ij'} = \frac{p_{ij}}{100} \cdot y_{ij}$$

(ahol y -nal az egyes tevékenységek tartamát jelöltük). p_{ij} itt az (i, j) tevékenység átlapolási %-át jelöli; az ERALL-módszer esetén ugyanis a megelőző (ún. „vezér”-) tevékenység tartamának %-ában szokás megadni az átlapolás mértékét.² p_{ij} % tehát azt jelenti, hogy a (k, l) tevékenység megvalósítását az (i, j) tevékenység p_{ij} %-ának elvégzése után kezdhethetjük el.



2. ábra

² (i, j) -hez több átlapoló tevékenység is tartozhat különböző átlapolási %-kal, így p_{ij} tulajdonképpen (k, l) -től is függhet; egyszerűség kedvéért azonban ezt az esetet nem részletezzük.

A $p = 0$ szélső esetben — mellyel külön nem foglalkozunk — (i, j) és (k, l) függetlenekké válnak, a másik szélső esetben, ha $p = 100$, az $(i, j) \ll (k, l)$ érvényes, tehát e két esetben felesleges átlapolásról beszélni.

Ha mármost

$$y_{kl} \geq \left(1 - \frac{p_{ij}}{100}\right) \cdot y_{ij},$$

akkor (2. ábra)

$$(j', j) \ll (l', l),$$

ellenkező esetben pedig

$$(k, l') \ll (j, l);$$

ezek bekövetkezését egy (j', k) és ezenkívül első esetben egy (j, l') , második esetben egy (l', j) látszattevékenység beiktatása valósítja meg. (A 2. ábrán ezért nem irányítottuk a tevékenységet ábrázoló nyilat.)

Mármost, ha P kaszkádszerű háló, akkor az ily módon kiegészített P^* is az. Az [1]-ben megadott 3 tulajdonság teljesülését ugyanis könnyen beláthatjuk:

1. A 2. ábrán jól látható, hogy az i, j', j, k, l', l eseményekkel meghatározott részhálónak is egyetlen kezdőeseménye van, mégpedig i , egyetlen záróeseménye, mégpedig l , ha tehát $i = 0$ ($i \in P$) volna, ez az $i \in P^*$ esetében is egyetlen kezdőesemény lenne, hasonlóan mondhatunk ki l -re záróesemény vonatkozásában.

2. j' és l' beiktatása által 4 további részben rendezési feltételt mondtunk ki, ezek azonban egyéb viszonylatokban nem szerepelnek, mert ha

$$\text{fennáll, akkor} \quad (h, i) \ll (i, j) \quad \text{vagy} \quad (h, i) \text{---} \ll (i, j)$$

$$(h, i) \ll (i, j'), \quad \text{ill.} \quad (h, i) \text{---} \ll (i, j')$$

is, és megfelelően, ha

$$(k, l) \ll (l, m), \quad \text{ill.} \quad (k, l) \text{---} \ll (l, m),$$

akkor a

$$(l', l) \ll (l, m), \quad \text{ill.} \quad (l', l) \text{---} \ll (l, m)$$

kapcsolatok is fennállnak, ami biztosítja, hogy az eddigi elérhetőségi feltételek is teljesüljenek, s az új események mindegyike is „elérhető”.

3. Az (i, k) és (j, l) tevékenységek beiktatása a tranzitivitás miatt a hurokmentességen nem változtat, ugyanígy a (j', k) -é sem, továbbá a (j, l') vagy (l', j) bármelyikét (de csak egyiket!) vezessük is be, hurkot nem okozunk, mert j és l' között más kapcsolat nincs (mint a 2. ábrából ugyancsak kitűnik).

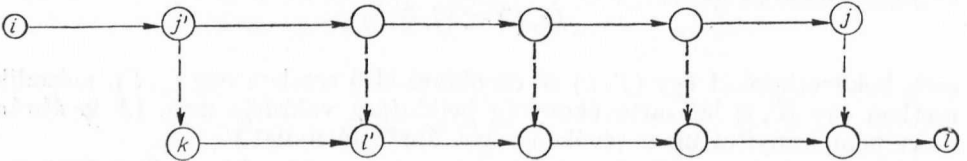
Fentiek alapján beláttuk, hogy P^* -ra is teljesülnek a kaszkádszerűség feltételei, tehát P^* is kaszkádszerű háló. Ily módon minden $(i, j) \text{---} \ll (k, l)$ átlapolásos kapcsolatot helyettesíthetünk a fenti módon adott 4 közvetlen megelőzési kapcsolattal, illetve fentieket megismételve, ennek véges számú többszörösével.

Eljárásunk gyakorlatilag azt jelenti, hogy az átlapoló, illetve átlapoló tevékenységeket az átlapolás %-ának megfelelően felbontottuk olyan kapcsolatokra, melyek már átlapolást nem tartalmaznak, s ezt egy tevékenységen belül

többször is megismételhetjük. Így például $p = 20$ (%) és $y_{ij} = y_{kl}$ mellett a 3. ábrán látható felbontás adódik.

A gyakorlatban a tevékenységek ilyen kapcsolódása legtöbbször úgy jelentkezik, hogy $y_{ij} \neq y_{kl}$, s ez esetben a láncolat egyes elemeinek tartamát megnyújtjuk vagy a tevékenységet megszakítjuk, hogy ez az eset álljon fenn.

Az átlapolás elsőrendű értelme az, hogy két (vagy több) tevékenység adott p %-os egymáshoz való eltolása által részben párhuzamosan valósul meg. Ennek alkalmazása elsősorban nagy (építőipari) létesítmények programozásánál fordul elő, például több szintes, több szakaszos (több lépcsőházas) épít-



3. ábra

mények esetén, ahol egy tevékenység 1—1 szintre vonatkozik (falazás), ugyanakkor a következő tevékenység (vakolás) megkezdésével nem várjuk meg a szint teljes elkészültét. Ez esetben elegendő, ha a tevékenységeknek a (3. ábra szerinti) felbontása helyett az (i, k) , ill. (j, l) tevékenységekhez az

$$y_{ik} = y_{jl} = \frac{p_{ij}}{100} \cdot y_{ij}$$

tartamot rendeljük, s az $\{i, j', j, k, l', l\}$ részháló átfutási ideje ebből a j' és l' kihagyása által előálló (eredeti) részháló átfutási ideje lesz. Ezen eset tárgyalása lényegesen egyszerűbb, mint az általánosé, de az általános eseten semmit sem módosít, ugyanis ha

$$y_{kl} < \left(1 - \frac{p_{ij}}{100}\right) y_{ij},$$

akkor (j, l) tartamával

$$- \left(1 - \frac{p_{ij}}{100}\right) \cdot y_{ij} + y_{kl}$$

értéket véve, a teljes átfutási időt ezen (negatív) szám segítségével ugyanígy lehet meghatározni.

2.2. Ezekután értelmezzük a P' hálót a következőképpen (ahol P egy kaszkádszerű háló):

$$\text{ha } i \in P, \text{ akkor } i \in P',$$

$$\text{ha } i \notin P, \text{ akkor } i \notin P',$$

$$\text{ha } (i, j) \in P, \text{ akkor } (j, i) \in P'.$$

P' -höz tehát ugyanazon események tartoznak, mint P -hez, és ugyanazon eseménypárok között vannak értelmezve a tevékenységek is, de ellentétes irányítással.

Világos, hogy ekkor 0-nak n , n -nek 0 felel meg, és az egyes események épp-úgy elérhetőek maradnak, mint előzetesen, valamint érvényes a hurokmentesség is, P' tehát szintén kaszkádszerű háló.

2.3. Legyen P_1 és P_2 mindegyike kaszkádszerű háló, legyenek P_1 és P_2 eseményei

$$i'_x \in P_1 \quad (x = 1, \dots, X),$$

illetve

$$i''_y \in P_2 \quad (y = 1, \dots, Y),$$

és legyen

$$P_1 \cap P_2 = 0;$$

ekkor alkossuk meg a következő P^{**} halmazt: először is legyen

$$P_1 \cup P_2 \subset P^{**},$$

valamint tetszőleges számú

$$(i'_x, i''_y)$$

tevékenységet felvéve, köztük az

$$(i'_1, i''_1)$$

és

$$(i'_X, i''_Y)$$

tevékenységeket (utóbbi esetben a kezdő és végpontok — események — összekötéséről van szó), alkossuk meg P^{**} -ot. (P^{**} eseményei tehát P_1 és P_2 összes eseményeivel azonosak, de a tevékenységek száma több lett az újabb kapcsolatok bevezetésével, melyek azonban mind $P_1 \rightarrow P_2$ irányításúak.)

Mármost P^{**} -nak egyetlen kezdőeseménye van, i'_1 , egyetlen végeseménye i''_Y , a két legutóbb említett tevékenység hozzávétele által az elérhetőséget is biztosítottuk (hiszen P_1 és P_2 -re külön-külön teljesen az elérhetőségi feltétel). Nem nehéz belátni továbbá azt sem, hogy P^{**} hurokmentes, mivel egyrészt P_1 és P_2 is az, másrészt csak $P_1 \rightarrow P_2$ irányítással létesítettünk újabb kapcsolatot, így P_2 -ből P_1 -be vissza nem juthatunk. Így tehát beláttuk, hogy P^{**} is kaszkádszerű háló, s ily módon a P_1 és P_2 -t egyesíthetjük („összevarrhatjuk”) kaszkádszerű hálóvá.

2.4. Legyen mármost a P halmaz a következő struktúrájú:

a) P kaszkádszerű háló,

b) létezik egy

$$i_1^*, i_2^*, \dots, i_M^*$$

lánc, és ennek elemeihez tartoznak olyan

$$P_{i_1}^*, P_{i_2}^*, \dots, P_{i_M}^*$$

halmazok, hogy mindegyik

$$P_{i_m}^* \quad (m = 1, \dots, M) \subset P$$

halmaz is kaszkádszerű háló, továbbá

$$P_{i_m}^* \text{ első elemére } (i_m^1)$$

$$0 \ll i_m^1,$$

utolsó (i_m^n) elemére

$$i_m^n \ll i_m^*,$$

vagyis $P_{i_m}^*$ a $(0, i_m^1)$, ill. (i_m^n, i_m^*) látszattevékenységekkel csatlakozik a fenti lánchoz. Pontosabban P -ről csak annyit kell kikötnünk, hogy megfelelő látszattevékenységek és események beiktatásával ilyené tehető, tehát a $P_{i_m}^*$ halmazok diszjunktak, ami azt jelenti, hogy az i_m^* -ot megelőző munkák különböző m -ek mellett egymástól nem függenek.

Ebben az esetben kaszkádszerű háló lesz az a halmaz (Q) is, amely P -ből a következőképpen állítható elő:

$$Q' = P - \bigcup_{m=1}^M P_{i_m}^*,$$

majd képezzük Q^* -ot:

$$Q^* = Q' + \bigcup_{m=1}^M P_{i_m}^{**},$$

és vegyük még hozzá a

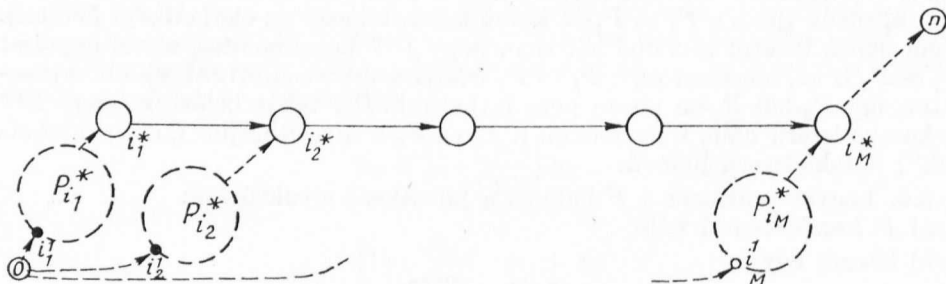
$$(0, i_1^*), (i_1^1, i_2^1), \dots, (i_{M-1}^1, i_M^1), \text{ végül } (i_M^1, n)$$

látszattevékenységet.

Az így képzett Q halmazról könnyű belátni, hogy szintén kaszkádszerű háló, mert

Q első eleme is 0, utolsó eleme n ,

a hálók egyesítése a 2.2. és 2.3. alapján történt, ami az ott bizonyítottak szerint azt jelenti, hogy Q valóban kaszkádszerű háló. A közölt eljárás a 4. ábrán látható vázlat szerint valósította meg a háló átalakítását.



4. ábra

Szóban kifejezve eljárásunk azt jelenti, hogy egy kaszkádszerű hálót adott lánchoz csatlakoztatva felbontunk, majd a lánchoz előlről kapcsolódó rész-hálókat „megfordítjuk”, és így csatlakoztatjuk vissza a lánchoz (a kaszkádszerűséghez szükséges megfelelő látszattevékenységekkel kiegészítve).

3. Az ütemezési feladat megfogalmazása

Az ütemezési feladat az [1]-ben szereplővel rokon, és a következőket jelenti: az

$$i_1^*, \dots, i_M^*$$

láncc minden eleméhez rendeljünk hozzá egy

$$t(i_1^*), \dots, t(i_M^*)$$

időpontot (ahol $\{i_1^*, \dots, i_M^*\} \subset P$), majd rendeljünk a P halmaz minden olyan tevékenységéhez, melyek kezdő- és végeseménye nem tartozik az $\{i_1^*, \dots, i_M^*\}$ halmazhoz, az [1]-ben leírt módon korlátozott erőforrásokat, illetve ezek segítségével tartamokat. Ezekután a tevékenységeket oly módon ütemezzük, hogy egyrészt

$$t_n \rightarrow \min !,$$

másrészt

$$t_0 \rightarrow \max !$$

legyen, figyelemmel az erőforráskorlátokra, és emellett az

$$i_1^*, \dots, i_M^*$$

elemekhez a

$$t(i_1^*), \dots, t(i_M^*)$$

időpont tartozzék.

A két optimálási követelményt a P halmazból kialakított Q halmazra vonatkoztatva láthatjuk, hogy egymásnak nem mondanak ellent, mert a 2.4.-ben definiált Q halmazon a két követelmény azonosan (minimumként) értelmezhető.

A célfüggvényt szavakban röviden így fogalmazhatjuk meg: adott (időben meghatározott) alapprogramhoz határozzuk meg az annak részeit megelőző és követő műveletek célszerű ütemezését. A célszerűség itt a megelőző tevékenységek vonatkozásában azt jelenti, hogy ezeket úgy állítsuk programba, hogy az alapprogram tevékenységeitől mintegy „visszafelé” ütemezzünk, ugyanakkor azonban a korlátos erőforrásokra is legyünk figyelemmel. Ezzel egyúttal arra is választ kapunk, hogy az előkészítő munkákat mikor és mennyivel kell az alapprogram előtt megkezdeni, hogy a munkák végzése egyrészt folyamatos legyen, másrészt az alapprogramban szereplő munkák elkezdéséig azok mind befejeződjenek (korlátos erőforrásokat véve figyelembe):

Gyakorlatban ez úgy jelentkezett, hogy egy nagy gép (toronydaru) előre betervezett működéséhez határozzuk meg az alapépítmények szükséges legkésőbbi kezdését, ill. a befejező munkák legkorábbi befejezését, figyelemmel az erőforráskorlátokra.

4. A feladat megoldása az ERALL-eljárás segítségével

Az ütemezési algoritmussal a feladatot két lépésben oldottuk meg:

4.1. A feladat átalakítása.

A 2. fejezetben részletesen tárgyaltuk a feladat megoldására szolgáló hálóttechnikai eljárásokat. Itt ezért csak vázlatosan foglaljuk össze annak lépéseit.

4.1.1. A tevékenységeket 3 halmazba soroljuk:

E , elemei azon tevékenységek és események, amelyek megvalósulása (bekövetkezése) az adott alapprogram tevékenységeit megelőzi;

I , az alapprogramhoz tartozó tevékenységek és események;

$U = Q - (E \cup I)$, azaz az alapprogramot követő tevékenységek és események.

4.1.2. Az E halmazhoz tartozó tevékenységek irányítását megfordítjuk, és az eseményeket átkódoljuk.

4.1.3. A kezdési és befejezési időpontokat átalakítjuk sorszámos alakúra, s az E halmaz ütemezését ezen „fordított módszer”-rel elvégezzük.

4.2. A megoldás 2. része:

4.2.1. Az E halmazhoz tartozó ütemezés után visszaállítjuk az eredeti helyzetet, s azokban az időszakokban, ahol mind előzmény, mind utólagos munkák folynak, az erőforráskorlátokat a már beütemezett tevékenységek szükségletei szerint módosítjuk (csökkentjük).

4.2.2. Elvégezzük az U halmazra nézve az allokációt.

4.2.3. A teljes programot átkonvertáljuk naptári dátumokra (ha a feladat megadása így történt).

4.3. Megjegyzések:

A 2.1.-ben tárgyalt eljárással az átlapolások kezelése is megoldódik; az allokációs módszer pedig az [1]-ben már tárgyalt ERALL-eljárás, így azt nem részletezzük.

Ha a feladat véghatárideje adott, és a kezdési határidőre nézve tehetünk engedményt, akkor a 4.2.2. és 4.1.3. lépések sorrendjét megcseréljük.

Ha az alprogram is ugyanazon erőforrásokból használ fel, amelyek az E és U halmazhoz tartoznak, akkor természetesen ezeket először kigyűjtjük, és a korlátokat a megfelelő időszakban ezekkel módosítjuk.

A megoldhatóság feltételei általában azonosak az [1]-ben leírtakkal; természetes követelmény még az is, hogy az I elemeihez tartozó tevékenységek összes erőforrásigénye a korlátos erőforrásokból az összkorlát alatt maradjon (illetve azt ne lépje túl), valamint az is, hogy a feladat kezdési és befejezési határideje (ha ilyen van) olyan legyen, hogy az erőforráskorlátok túllépése nélkül tartható legyen. (Ez utóbbiak is általános követelmények hasonló feladatokban, ezért ezek vizsgálatát mellőzzük.)

5. Gyakorlati alkalmazások

A fentiekben leírt, matematikailag megalapozott módszert sikerrel alkalmaztuk egy lakásépítési programban, ahol is több, az építkezés fő tevékenységeit meghatározó daru előzetes számítások alapján programozva lett, és ehhez kellett a többi erőforrás ütemtervét elkészíteni az elmondott szempontok szerint. Felmerült a módszer olyan alkalmazása is, amikor az alprogram nem előre adott, hanem azt is a módszernek kell elkészítenie. Ilyen esetre példa a lineáris programozásból ismert gépterhelési probléma (1. pl. [4]), ahol az egyes gépek leterhelését lineáris programozás segítségével optimalizálhatjuk, így kiválaszthatjuk az egyes létesítményekhez a megfelelő gépet (darut), s ezek leterheléséhez igazítjuk a kivitelezés többi fázisát a hálótéchnika segítségével. Mivel az I halmaz ütemezése vagy előre adott, vagy pedig valamilyen egzakt (pl. lineáris) programozási módszerrel történt, a feladat hatékonysága és egzaktysága azonos az ERALL-módszer már közölt hatékonyságával és egzaktyságával.

A hálótéchnikán alapuló ütemezési eljárások alkalmazási területe erősen kibővült azzal, hogy olyan esetekben is alkalmazásra kerülnek, amikor a programozási feladat lényege egy sorrendprogramozási vagy lineáris programozási

feladat, de számolni kell a rendszerben fellépő kapacitáskorlátokkal. A módszer kisebb-nagyobb mértékben több ízben alkalmazásra került, s a gyakorlat azt bizonyítja, hogy a hálótechnika alkalmazása az ilyen „kevert” jellegű feladatokban is hasznos.

(Beérkezett: 1973. október 5.)

IRODALOM

1. SZABÓ, I.—SZOLNOKY, A.: Az ERALL-2 hálótechnikai algoritmus. SZIGMA, 1970. 3. sz. 195—211. o.
2. CZÉKMÁNY, I.—SZABÓ, I.: Vezértevékenységre épülő hálós szervezési módszerek. Informatió Elektronika, 1971. 4. sz. 258—262. o.
3. „Hálótervezési módszerek”. Budapest, 1968. Országos Ügyvitelgépésítési Felügyelet.
4. KREKÓ BÉLA: Lineáris programozás. Budapest, 1962. Közgazdasági és Jogi Könyvkiadó.

APPLICATION OF NETWORKS IN A PROGRAMMING PROBLEM OF MIXED CHARACTER

The article introduces the application of network techniques — i.e. one of its special methods, the procedure ERALL which our journal has already described — in a new connection combined with another problem. It determines the chronological distribution of the preceding and following jobs, their latest starts and their earliest possible ends or a so called basic program, that was made independent by the method, in general not by way of net technics. In the procedure the continuation of activities of the basic program, and the constraints on resources are taken into consideration. The detailed mathematical description is followed by a brief account of application and the conditions of solvability.

ИСПОЛЬЗОВАНИЕ СЕТЕВОЙ ТЕХНИКИ В ЗАДАЧЕ ПРОГРАММИРОВАНИЯ СМЕШЕННОГО ХАРАКТЕРА

Статья показывает использование сетевой техники — и ее специальной методики, написанной в нашем журнале, метод ЭРАЛЛ, — в новом аспекте, комбинированно с другой задачей. К так называемой основной программе, которую, независимо от методики, изготовили по методу сетевой техники, определяется распределение по времени предыдущих, а также следующих работ, позднейшие начала и возможные самые ранние окончания, следя, чтобы предыдущие работы могли непрерывно следовать действия основной программы и везде предусматривать возможности ограничения ресурсов. После исчерпывающего освещения математических основ, статья коротко рассказывает о ее применениях и занимается условиями ее решения.