REVISTA DE ANALIZĂ NUMERICĂ ŞI TEORIA APROXIMAŢIEI
Volumul, 2 Fascicola 1, 1973, pp. 89—104

CERCETĂRI ÎN DOMENIUL PLANIFICĂRII OPERATIVE LA INSTITUTUL DE CALCUL (II)*

de

L. NÉMETI
(Cluj)

7. Mai multe exemplare din acelaşi tip de maşini-unelte III.

Ne întoarcem la un mod de descriere a situaţiei studiate asemănător cu cel din paragraful 5 (vezi [6]). Vom nota cu $M = \{1, 2, \ldots\}$: mulţimea tipurilor de maşini de prelucrare;

$N = \bigcup_{k \in M} N_k$: mulţimea operaţiilor, partiţionată după tipurile de maşini pe care urmează să fie executate;

μ_k, $k \in M$: numărul exemplarelor existente din tipul de maşini k;

$(i, j) \in H \subset N \times N$: înseamnă că începutul operaţiei j nu poate avea loc, din motive tehnologice, decît cel puţin t_{ij} unităţi de timp după începerea operaţiei i;

t_i, $i \in N$: durata de execuţie a operaţiei i;

x_i, $i \in N$: momentul de începere a operaţiei i.

Noţiuni noi:

$I_i = [x_i, x_i + t_i[$, $i \in N$: intervalul de derulare a operaţiei i;
$\gamma(x) | L_k)$, $L_k \subset N_k$, $k \in M$: funcţia de încărcare a tipului k în raport cu submulţimea de operaţii L_k. Fiecărei operaţii din L_k (după ce operaţia a fost

* Prima parte a prezentei lucrări a fost publicată în vol. 1, fasc. 1, a acestei revista.

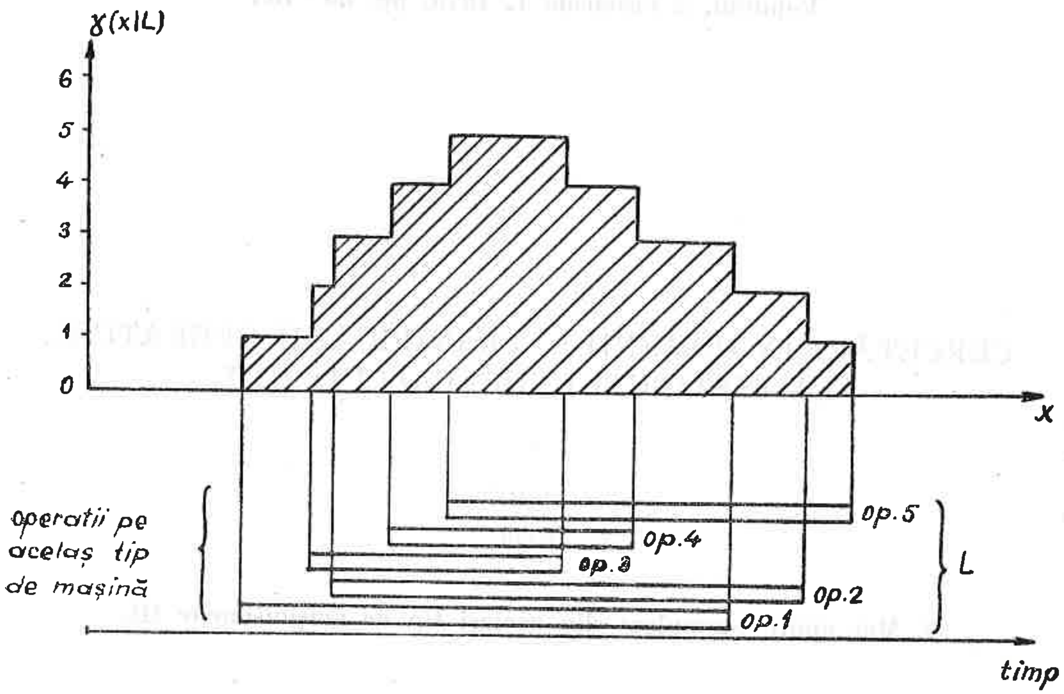

Figura 1

programată, adică i s-a stabilit momentul de începere) îi va aparține intervalul de derulare I_i. Valoarea $\gamma(x \mid L_k)$ va fi egală cu numărul acelor intervale de derulare pentru care are loc $x \in I_i$ (vezi fig. 1).

Cu aceste definiții, modelul matematic al problemei se formulează după cum urmează:

Să se determine componentele vectorului $X = (x_1, x_2, \ldots, x_n)$ care să verifice următoarele restricții: condițiile de începere

$$x_i \geqslant 0, \; i \in N; \tag{7.1}$$

condițiile de succesiune

$$x_j - x_i \geqslant t_{ij}, \; (i, j) \in H; \tag{7.2}$$

condițiile de încărcare

$$\gamma(x_i \mid N_k) \leqslant \mu_k, \; i \in N_k, \; k \in M \tag{7.3}$$

în așa fel încît funcția economică dată

$$z = F(X) \tag{7.4}$$

să devină minimă. Se presupune că funcția (7.4), cu valori reale, este crescătoare în raport cu fiecare componentă x_i și crește cu ele peste orice limită.

Pentru rezolvare acestei probleme se va utiliza de asemenea un procedeu Branch and Bound. În acest scop trebuie să se dea:

a) Prima problemă de bază.

b) Regulile cu ajutorul cărora se generează noi probleme fundamentale din cele rezolvate.

c) Forma generală a problemelor fundamentale și metoda de rezolvare.

d) Funcția pentru compararea soluțiilor problemelor fundamentale.

Dăm aceste indicații după cum urmează:

a) Prima problemă de bază este formată din restricțiile (7.1) și (7.2) împreună cu condiția de optimizare (7.4).

b) Să presupunem că s-a rezolvat o problemă fundamentală P: soluția sa optimă $\Xi = (\xi_i \mid i \in N)$ verifică condițiile (7.1) și (7.2) și conferă funcției (7.4) valoarea m. Cu ajutorul acestei soluții se va controla, dacă funcțiile $\gamma(\xi_i \mid N_k)$ verifică restricțiile (7. 3). Dacă toate aceste condiții sînt verificate, atunci Ξ constituie o soluție admisibilă și din P nu se mai generează noi probleme fundamentale. În caz contrar vor exista tipuri $k \in M$ și submulțimi de operații $L_k \subset N_k$ cu cîte un interval $I = [a, b[$ bucurîndu-se de proprietatea că pentru orice $x \in I$ are loc

$$\nu = \gamma(x \mid L_k) - \mu_k > 0. \tag{7.5}$$

O astfel de mulțime L_k o vom numi mulțime de depășire aferent și numărul natural ν este gradul de depășire corespunzător.

Din mulțimile de depășire constituite prin soluția Ξ se va alege una. Fie această mulțime

$$L = \{i_1, i_2, \ldots, i_q\} \subset N$$

și să presupunem notarea indicilor făcută în așa fel, încît pentru momentele de terminare a operațiilor i

$$\eta_i = \xi_i + t_i$$

să aibă loc

$$\begin{cases} \eta_{i_1} \leqslant \eta_i, & i \in L \\ \eta_{i_2} \leqslant \eta_i, & i \in L \setminus \{i_1\}. \end{cases} \tag{7.6}$$

În vederea generării noilor probleme fundamentale se formează disjuncţia

$$(7.7) \qquad (x_{i_1} \geqslant \eta_{i_1}) \vee (x_{i_2} \geqslant \eta_{i_2}) \vee \ldots \vee (x_{i_q} \geqslant \eta_{i_q})$$

şi cîte unul din termenii acestei disjuncţii se va adăuga la restricţiile problemei P, obţinîndu-se în acest fel q probleme fundamentale noi.

c) Problemele fundamentale au forma generală

$$(7.8) \qquad \begin{cases} x_i \geqslant a_i, & i \in N \\ x_j - x_i \geqslant t_{ij}, & (i, j) \in H \\ z = f(X): & \text{minim}, \end{cases}$$

unde graful orientat $G(N, H)$ ataşat sistemului (7.8) este fără circuite. Este de remarcat că problemele fundamentale ale unei probleme de programare temporală se deosebesc una de alta numai prin valorile numerelor a_i, mulţimile N, H, numerele t_{ij} şi graful $G(N, H)$ fiind comune la toate. Această proprietate constituie un avantaj considerabil al acestei metode.

Se caută aşa numita soluţie minimă universală a sistemului (7.8) care în esenţă se obţine prin bine conoscuta metodă a drumului critic. Formula de rezolvare este

$$(7.9) \qquad \xi_i = \max\,[a_i, \max_{j \in P(i)} (\xi_j + t_{ij})], \quad i \in N,$$

unde $P(i)$ reprezintă mulţimea predecesorilor vîrfului i din graful $G(N, H)$ susmenţionat.

d) Funcţia folosită pentru comparare este tocmai funcţia economică $\zeta = f(\Xi)$.

Din cauza faptului că graful G rămîne neschimbat la toate problemele fundamentale, această metodă se pare că prezintă avantaje faţă de celelalte metode.

8. Fabricaţia în serie

Fabricaţia în serie se caracterizează prin faptul că operaţiile de executat se repetă în mod periodic. Atunci se va alege ca funcţia economică de minimizat durata acestei perioade, situaţie ce permite folosirea maximă a capacităţii de producţie.

Să considerăm situaţia mai simplă a cazului, cînd fiecare maşină de prelucrat este de alt tip (§ 3) şi să arătăm în ce fel condiţiile acestui caz se mai completează cu alte restricţii în cazul producţiei în serie.

Vom avea (vezi § 3)

Condiţiile de succesiune

$$(8.1) \qquad x_j - x_i \geqslant t_{ij}, \quad (i, j) \in H$$

condiţiile de neinterferenţă

$$(8.2) \qquad x_j - x_i \geqslant t_i \vee x_i - x_j \geqslant t_j, \quad (i, j) \in J.$$

La acestea se adaugă aşa numită condiţie a perioadei

$$(8.3) \qquad x_j - x_i \geqslant t_i - v, \quad (i, j) \in \bar{J}.$$

Mulţimea

$$(8.4) \qquad \bar{J} = \{(i, j) \mid i, j \in N_k;\ k \in M\} = \bigcup_{k \in M} (N_k \times N_k)$$

este înrudită cu mulţimea J definită prin (3.3). Sensul restricţiei (8.3) devine clar, dacă o scriem în forma

$$v \geqslant (x_i + t_i) - x_j, \quad (i, j) \in \bar{J}$$

care furnizează condiţia esenţială pentru durata v a perioadei.

Avem aici două mărimi de minimizat: durata perioadei v şi durata u a procesului de fabricaţie care este definită prin (3.5). Se dă prioritate lui v, adică se cer valori ale necunoscutelor x_i care să minimizeze durata v. Din mulţimea soluţiilor se va alege aceea care minimizează pe u.

Fiind vorba şi aici despre o problemă de programare cu disjuncţii care se rezolvă prin procedeul Branch and Bound, să studiem forma unei probleme fundamentale: o astfel de problemă va avea înfăţişarea

$$(8.5) \qquad \begin{cases} x_j - x_i \geqslant t_{ij} & (i, j) \in L \quad (H \subset L) \\ x_j - x_i \geqslant t_i - v, & (i, j) \in \bar{J} \end{cases}$$

şi

$$(8.6) \qquad v: \text{minim}.$$

Este evident că minimul lui v este cea mai mică valoare pentru care sistemul (8.5) este compatibil. De aici rezultă o metodă „simplistă" de rezolvare, o metodă de aproximări succesive atribuind lui v diferite valori. Are loc în orice caz

$$(8.5) \qquad v_{\min} \geqslant \max_{i \in N} t_i.$$

Această mărime v_0 poate servi ca valoare de pornire; cu mărirea treptată a acesteia, se poate aproxima $v_{\min}$.

Valoarea $v_{\min}$ odată găsită, sistemul (8.5) reprezintă o problemă de potențial a cărei soluție minimă universală minimizează variabila u.

Iterațiile numeroase indicate în aliniatul precedent pot fie evitate prin metoda prezentată în lucrarea [2] cu ajutorul căreia, din graful G atașat sistemului (8.5) rezultă alte grafe de structură mai simplă la care determinarea mărimii $v_{\max}$ se poate face deja cu metode neiterative, matriciale. Trimitem în această privință și la lucrările [1], [8], [9].

9. Stocuri tampon

În programele întocmite pe baza metodelor discutate pînă acum, apar în general „goluri", adică în desfășurarea fabricației apar din cînd în cînd timpi de așteptare ale unor mașini-unelte. O astfel de așteptare apare atunci, cînd operația ce ar urma nu poate fi abordată pe mașina considerată, întrucît reperul în cauză se găsește în curs de prelucrare pe o altă mașină.

Este posibilă și o programare mai bună, cu o folosire mai intensă a utilajelor, prin care se evită aceste goluri, cel puțin la mașinile cele mai solicitate. Acest lucru este posibil prin creierea unor stocuri tampon în fața mașinilor respective. Ca urmare, în caz de nevoie se consumă din aceste stocuri, fără a mai fi nevoie să se aștepte terminarea operațiilor tehnologic precedente. Stocul tampon consumat va fi întregit cu piesele rezultate din operațiile precedente. Aceste stocuri tampon se crează într-o perioadă inițială de pregătire, problema folosirii acestei metode se pune prin urmare numai în cazul producției în serie.

Modelul matematic este deci înrudit cu cel din paragraful precedent existînd totuși deosebiri esențiale.

a) Prima deosebire este că restricția de succesiune tehnologică se înlocuiește cu o altă restricție. În acest scop se introduce mărimea s_j a stocului tampon creiat pentru a asigura desfășurarea nestînjenită a operației j (unitatea de măsură pentru stoc va fi mărimea lotului din piesele respective). Dacă $s_j = 0$, atunci restricția de succesiune (8.1) este valabilă. Dacă $s_j \geqslant 1$, atunci între x_i și x_j nu există nici o restricție. Pentru valorile intermediare ale lui s_j, în lucrarea [3] se arată că restricția între x_i, x_j și s_j în vederea executării nestînjenite a operației j este

$$s_j t_j^0 + x_j - x_i \geqslant \tau_{ij}, \quad (i, j) \in H, \tag{9.1}$$

unde t_j^0 și τ_{ij} sînt definite în aceeași lucrare și se deduc din timpii dați de prelucrare ai operațiilor i și j.

b) A doua deosebire este că lungimea v a perioadei, definită în paragraful precedent, este dată. Ea este egală cu suma timpilor de execuție

a operațiilor ce urmează a fi executate pe mașina cea mai încărcată. Prin urmare, mărimea v în restricția (8.3) are valoarea

$$v = \bar{t} = \max_{k \in M} \sum_{i \in N_k} t_i. \tag{9.2}$$

c) Cea de a treia deosebire este că durata minimă posibilă fiind atinsă, este rațional să se aleagă ca funcție economică de minimizat valoarea totală a stocurilor tampon

$$w = \sum_{i \in N} \alpha_i s_i : \ \text{minim.} \tag{9.3}$$

Modelul matematic obține astfel următoarea înfățișare: necunoscutele x_i, s_i, $i \in N$ să verifice restricțiile:

$$\begin{cases} x_i \geqslant 0, & i \in N \\ s_i \geqslant 0, & i \in N \\ x_j - x_i \geqslant t_i - \bar{t}, & (i, j) \in \bar{J} \\ x_j - x_i \geqslant t_i \vee x_i - x_j \geqslant t_j, & (i, j) \in J \\ s_j t_j^0 + x_j - x_i \geqslant t_{ij} \vee s_j \geqslant 1, & (i, j) \in H \end{cases} \tag{9.4}$$

în așa fel încît condiția de optimizare (9.3) să fie realizată.

Dezavantajul acestui model constă în faptul că problemele fundamentale ce se nasc din el, nu mai sînt probleme de potențial. Motivul rezidă în ultima condiție din (9.4) unde apar trei necunoscute (în loc de diferența a două necunoscute). Introducînd însă substituția

$$z_i = x_i + s_i t_i^0, \quad i \in N, \tag{9.5}$$

sistemul (9.4) se transformă într-un sistem (cu necunoscutele x_i și z_i) ale cărui probleme fundamentale sînt probleme de potențial. Pentru detalii, trimitem la lucrarea citată, unde se tratează și situația existenței mai multor exemplare din același tip de mașini.

10. Lucrul în mai multe schimburi

Acest model de organizare a fabricației a fost tratat — din punctul de vedere al ordonanțării — în lucrarea [5]. O situație esențial nouă față de cazurile analizate în paragrafele anterioare apare în unul din următoarele două cazuri (care pot apărea și simultan):

a) Diferite mașini-unelte lucrează într-un număr diferit de schimburi respectiv în diferite schimburi.

b) O operaţie începută într-un schimb σ (= 1, 2, sau 3) şi neterminată în acelaşi schimb se întrerupe. Următorul schimb începe cu o altă operaţie, iar prima operaţie se continuă în acelaşi schimb σ din ziua următoare.

Abaterile de la cazurile tratate anterior îşi au originea în următorul fapt: dacă o operaţie, avînd durata de execuţie t, începe în momentul x, atunci ea se termină în momentul $y = x + t$. La lucrul în mai multe schimburi, funcţia $y = f(x)$ este mult mai complicată. Ea nu este liniară, în general nu este continuă (posedă salturi) şi nu este definită pe întregul interval $]-\infty, +\infty[$.

Pentru a deduce noua funcţie $f(x)$ se introduc mai multe noţiuni noi. În cele ce urmează se studiază situaţia mai simplă cînd există cîte un exemplar din fiecare tip de maşină de prelucrare.

Se definesc două perechi de funcţii pe toată axa reală prin relaţiile:

$$(10.1)\qquad \begin{cases} x = [x] + \{x\}, & [x]: \text{ număr întreg, } 0 \leqslant \{x\} < 1, \\ x = [x]' + \{x\}', & [x]': \text{ număr întreg, } 0 < \{x\}' \leqslant 1. \end{cases}$$

Se consideră o operaţie oarecare şi maşina pe care ea se execută. Maşina lucrează într-o mulţime dată de schimburi $S \subset \{1, 2, 3\}$. Durata unui schimb este unitatea de timp. Schimburile din S formează un interval de timp cu momentul de începere ε. Are loc

$$(10.2)\qquad \left|\begin{array}{l} \left.\begin{array}{l} S = \{1\} \\ S = \{1, 2,\} \\ S = \{1, 2, 3\} \end{array}\right\} \Rightarrow \varepsilon = 3c \\ \left.\begin{array}{l} S = \{2\} \\ S = \{2, 3\} \end{array}\right\} \Rightarrow \varepsilon = 3c + 1 \\ \left.\begin{array}{l} S = \{3\} \\ S = \{3, 1\} \end{array}\right\} \Rightarrow \varepsilon = 3c + 2 \end{array}\right.$$

unde c este un număr întreg oarecare. Mulţimile S sînt ordonate.

Se notează

$$(10.3)\qquad l = |S| \qquad (l = 1, 2, \text{ sau } 3).$$

Există două *moduri de uzinaj*.

Modul de *uzinaj* A: O operaţie începută într-un schimb $\sigma \in S$ şi neterminată în acest schimb, se continuă în schimbul următor.

Modul de *uzinaj* B: este descris la aliniatul b) de la începutul acestui paragraf.

Se mai determină schimbul în care se găseşte un moment dat x de timp:

$$(10.4)\qquad \sigma(x) = 1 + \left[3\left\{\frac{x}{3}\right\}\right].$$

Trecînd acum la determinarea funcţiei $y = f(x)$ se arată că are loc pentru modul de uzinaj A:

$$(10.5)\qquad y = f(x) = A(x) = x + t + (3 - l)\left[\frac{3\left\{\frac{x-\varepsilon}{3}\right\} + t}{l}\right]' \quad \text{(vezi fig. 2)},$$

iar pentru modul de uzinaj B

$$(10.6)\qquad y = f(x) = x + t + 2\,[t + \{x\}]' \quad \text{(vezi fig. 3)}.$$

Intervalele lui x în care are loc

$$(10.7)\qquad \{x\} > 1 - \{t\}',$$

le numim intervale interzise. Motivarea acestei denumiri iese la iveală mai jos.

Se mai folosesc noţiuni deja definite în paragrafele anterioare:

M: mulţimea maşinilor

$N = \bigcup_{k \in M} N_k$: mulţimea operaţiilor

$H \subset N \times N$: mulţimea perechilor de operaţii legate între ele printr-o relaţie de succesiune tehnologică.

$J \subset N \times N$: mulţimea perechilor de operaţii ce se execută pe aceeaşi maşină.

Pentru fiecare maşină $k \in M$ se dă mulţimea S^k a schimburilor în care lucrează. Pentru orice operaţie $i \in N_k$ are loc $S_i = S^k$, de asemenea

$$S_i = S_j, \qquad (i, j) \in J.$$

După aceste pregătiri putem trece la construirea modelului matematic al problemei. Sistemul de restricţii va conţine:

condiţiile de începere

$$(10.8)\qquad x_i \geqslant 0, \qquad i \in N;$$

condiţiile schimburilor

$$(10.9)\qquad \sigma(x_i) \in S_i, \qquad i \in N;$$

condiţiile de succesiune tehnologică

$$(10.10)\qquad x_j \geqslant f(x_i), \qquad (i, j) \in H;$$

unde $f(x) = A(x)$ respectiv $B(x)$ conform cazului de uzinare a operaţiei i;

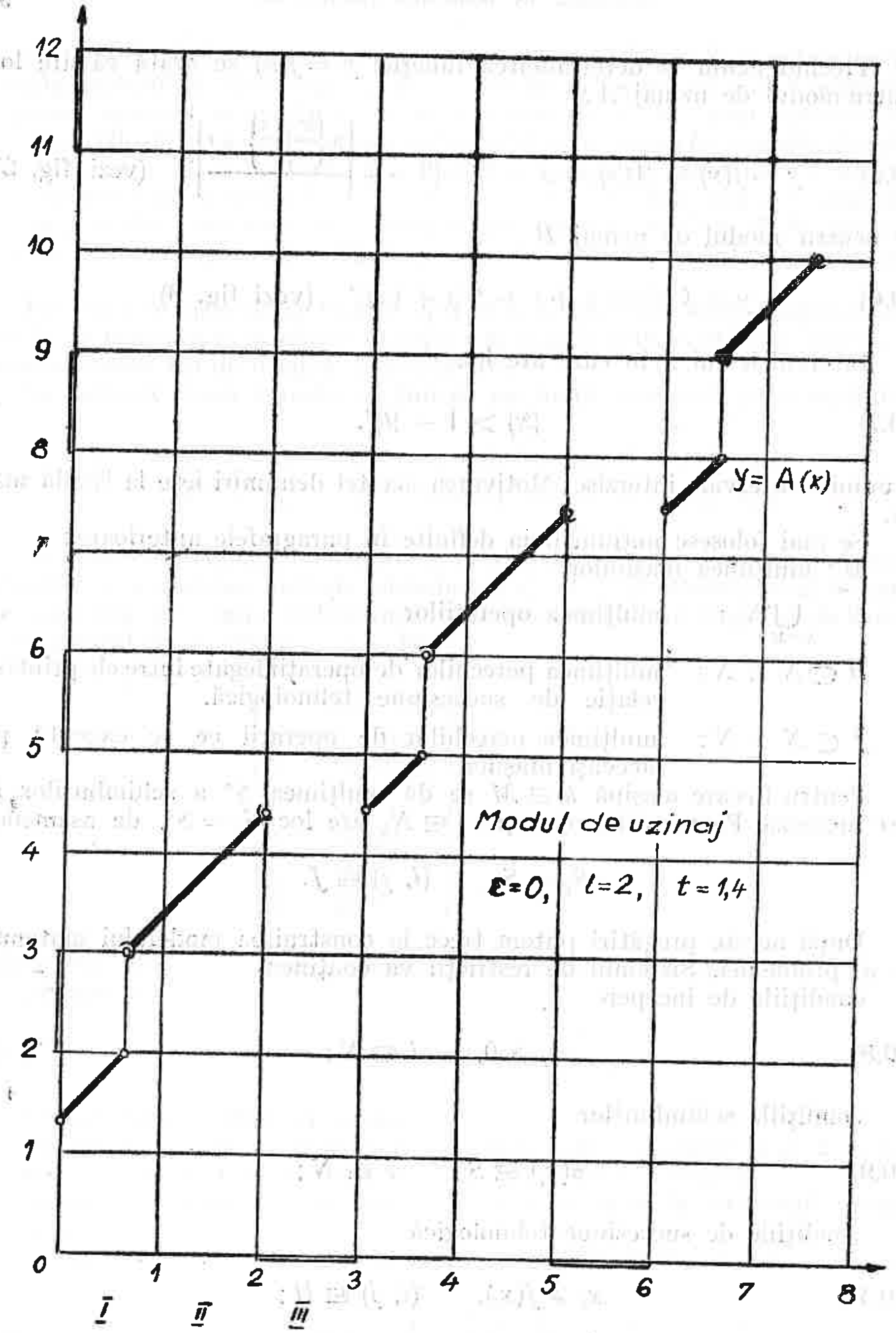

Figura 2

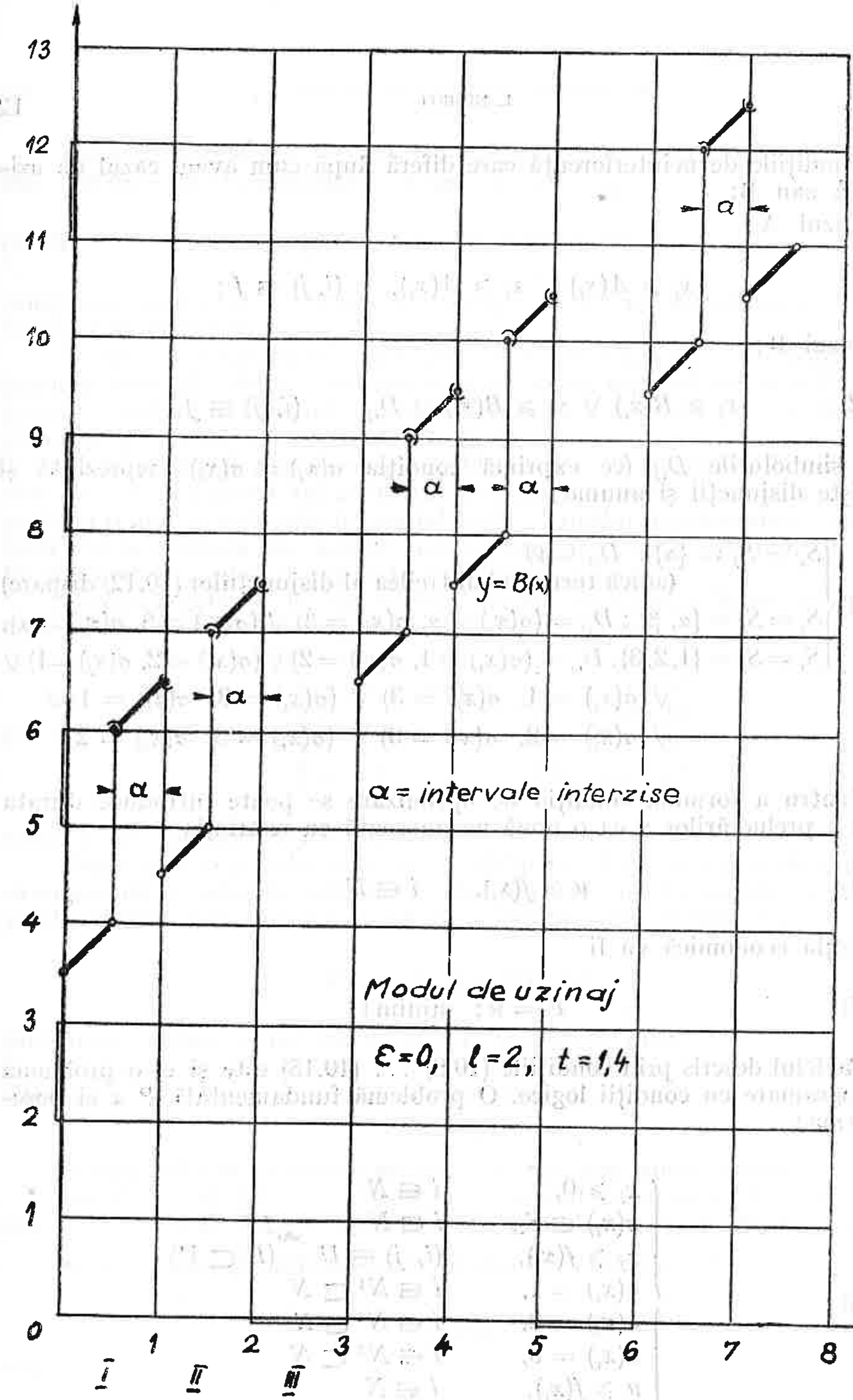

Figura 3

Condiţiile de neinterferenţă care diferă după cum avem cazul de uzinare A sau B:

Cazul A:

$$x_j \geqslant A(x_i) \lor x_i \geqslant A(x_j), \qquad (i, j) \in J; \tag{10.11}$$

cazul B:

$$x_j \geqslant B(x_i) \lor x_i \geqslant B(x_j) \lor D_{ij}, \qquad (i, j) \in J, \tag{10.12}$$

unde simbolurile D_{ij} (ce exprimă condiţia $\sigma(x_i) \neq \sigma(x_j)$) reprezintă şi ele nişte disjuncţii şi anume:

$$\begin{cases} S_i = S_j = \{\alpha\}: \ D_{ij} = \emptyset \\ \qquad\qquad \text{(adică termenul al treilea al disjuncţiilor (10.12) dispare)} \\ S_i = S_j = \{\alpha, \beta\}: D_{ij} = (\sigma(x_i) = \alpha, \ \sigma(x_j) = \beta) \lor (\sigma(x_i) = \beta, \ \sigma(x_j) = \alpha), \\ S_i = S_j = \{1, 2, 3\}: D_{ij} = (\sigma(x_i) = 1, \ \sigma(x_j) = 2) \lor (\sigma(x_i) = 2, \ \sigma(x_j) = 1) \lor \\ \qquad \lor (\sigma(x_i) = 1, \ \sigma(x_j) = 3) \lor (\sigma(x_i) = 3, \ \sigma(x_j) = 1) \lor \\ \qquad \lor (\sigma(x_i) = 2, \ \sigma(x_j) = 3) \lor (\sigma(x_i) = 3, \ \sigma(x_j) = 2). \end{cases} \tag{10.13}$$

Pentru a formula condiţia de optimizare se poate introduce durata totală a prelucrărilor u ca o nouă necunoscută cu restricţia

$$u \geqslant f(x_i), \qquad i \in N \tag{10.14}$$

şi funcţia economică va fi

$$F = u: \ \text{minim!} \tag{10.15}$$

Modelul descris prin condiţiile (10.8) ... (10.15) este şi el o problemă de programare cu condiţii logice. O problemă fundamentală P a ei posedă forma:

$$\begin{cases} x_i \geqslant 0, & i \in N \\ \sigma(x_i) \in S_i, & i \in N \\ x_j \geqslant f(x_i), & (i, j) \in U \quad (H \subset U) \\ \sigma(x_i) = 1, & i \in N^1 \subset N \\ \sigma(x_i) = 2, & i \in N^2 \subset N \\ \sigma(x_i) = 3, & i \in N^2 \subset N \\ u \geqslant f(x_i), & i \in N \\ u: \ \text{minim!} \end{cases} \tag{10.16}$$

Dacă mulţimile N^1, N^2, N^3 nu sînt disjuncte două cîte două, problema (10.16) este incompatibilă. Vom nota cu

$$N^0 = N \setminus (N^1 \cup N^2 \cup N^3) \tag{10.17}$$

mulţimea operaţiilor ce nu sînt înrolate apriori într-un schimb prin problema fundamentală studiată.

În cele ce urmează descriem procedeul de rezolvare a unei probleme fundamentale P (10.16). Acesteia i se va ataşa un graf orientat $G(N, U)$ parţial ponderat (la vîrfuri), anume fiecare vîrf $i \in N^\lambda$ ($\lambda = 1$, 2 sau 3) primeşte ponderea „de schimb" λ ($\lambda \in S_i$).

Dacă graful G are circuite, problema P este incompatibilă. În caz contrar este cunoscut (vezi de ex. metoda drumului critic) că vîrfurile grafului G pot fi ordonate în aşa fel încît vîrfurilor predecesorare ale vîrfului i li se atribuie un număr ordinal mai mic decît cel atribuit lui i. Necunoscutele x_i vor fi calculate în această ordine.

Necunoscutelor x_i la care vîrful i nu are predecesori în graful G (acest fapt se va nota cu $i \in C_0$) li se atribuie valoarea

$$x_i = \begin{cases} 0 & \text{pentru} \quad i \in C_0 \cap N^1 \\ 1 & \quad ,, \qquad i \in C_0 \cap N^2 \\ 2 & \quad ,, \qquad i \in C_0 \cap N^3 \\ s_i - 1 & \quad ,, \qquad i \in C_0 \cap N^0 \end{cases} \tag{10.18}$$

unde s_i este primul element din S_i.

După susmenţionata ordonare a vîrfurilor, dacă ajungem să calculăm necunoscuta x_i ataşată vîrfului i, atunci toate necunoscutele x_j ataşate predecesorilor j ai vîrfului i sînt deja determinate. Notînd cu

$$P(i) = \{j \mid (j, i) \in U\}$$

mulţimea predecesorilor lui i, se va determina întîi

$$x_i' = \max_{j \in P(i)} f(x_j). \tag{10.19}$$

Pentru cele ce urmează se vor distinge mai multe cazuri:

Cazul 1 $\quad i \in N^0, \quad \sigma(x_i') \in S_i, \quad$ modul de uzinare A

sau $\quad i \in N^0, \quad \sigma(x_i') \in S_i, \quad$ modul de uzinare B şi

x_i nu se găseşte într-un interval interzis, adică are loc

$$\{x_i'\} \leqslant 1 - \{t_i\}' \text{ (vezi 10.7)}$$

sau

$$i \notin N^\circ, \ \sigma(x_i') = \lambda_i.$$

Se va pune

$$x_i = x'_i. \tag{10.20}$$

Cazul 2 $i \in N^0$, $\sigma(x'_i) \in S_i$, modul de uzinare B şi x'_i se găseşte într-un interval interzis

sau

$$i \in N^\circ, \quad \sigma(x'_i) \notin S_i.$$

Se va pune

$$x_i = [x'_i] + \alpha_i, \tag{10.21}$$

unde α_i este cel mai mic număr natural pentru care are loc

$$\sigma(x_i) \in S_i.$$

Cazul 3 $i \notin N^0$, $\sigma(x'_i) \neq \lambda_i$.
Se va pune

$$x_i = [x'] + \beta_i, \tag{10.22}$$

unde β_i este cel mai mic număr natural pentru care are loc

$$\sigma(x_i) = \lambda_i.$$

Se demonstrează în lucrarea citată că prin acest procedeu se obţine soluţia minimă universală cu o durată totală de desfăşurare a programului

$$u = \max_{i \in N} f(x_i). \tag{10.23}$$

11. Metode euristice. Concluzii.

Dacă se consideră modelele expuse în paragrafele precedente sub punctul de vedere al volumului calculelor necesare, se constată următoarele: dacă numărul operaţiilor nu este prea mare, atunci modelul simplu, „clasic", expus în paragraful 3 este abordabil prin calculatoare moderne de talia medie (50...200 mii de operaţii pe secundă, 10 ... 50 mii de cuvinte în memoria operativă), Cu cît creşte însă numărul operaţiilor, timpii de calcul cresc într-un mod inadmisibil. Situaţia este şi mai pregnantă la modelele arătate în paragrafele următoare, modelele ce aproximează mai bine realitatea tehnologică şi organizatorică complexă a intreprinderilor. În acest fel cercetările descrise vizează mai puţin aplicaţiile practice imediate, decît obiectivul de a servi ca rezervă a ştiinţei în vederea aplicării lor la calculatoare viitoare mult mai puternice.

Pentru necesităţile actuale, o altă direcţie de cercetare promite a fi mai eficace: aceea de a construi metode de suboptimizare, adică metode euristice, de aproximare, unde restricţiile problemei sînt verificate — se elaborează programe admisibile — însă atingerea optimului nu este garantată. Totuşi, nişte principii practice de selecţionare fac plauzibilă ipoteza că soluţia găsită nu se va îndepărta prea mult de cea optimă. Se cere în schimb ca procedeul de calcul să nu fie prea voluminos, să fie accesibil calculatoarelor actuale.

La Institutul de calcul au fost elaborate mai multe din aceste metode din care o parte a şi fost aplicată cu succes.

În cele ce urmează vom schiţa pe scurt unul din aceste metode. Procedeul conceput se aplică la fabricaţia în serie, fără stocuri tampon, existînd mai multe maşini din acelaşi tip. Programul elaborat constă din programele pentru fiecare maşină, iar un astfel de program nu este altceva decît o listă conţinînd operaţiile ce urmează a fi executate la această maşină precum şi momentele lor de începere. Într-o fază intermediară, programele se numesc programe parţiale. Procedeul constă din alegeri succesive de maşini şi de operaţii ce sînt la rînd pentru a fi programate, şi din înserarea operaţiei alese în programul parţial al maşinii alese. Intrebarea este cum se fac aceste alegeri şi aceste înserări.

Ele se execută pe baza următoarelor principii:

a) Se va alege pentru a fi programată maşina cea mai încărcată.

b) Dintre piesele ce urmează a fi programate pe una din maşinile din tipul ales, se va programa piesa ce necesită durata mai mare pînă la prelucrarea sa completă.

c) Inscrierea operaţiilor în programul parţial se va face la aceea maşină şi în aşa fel, încît perioada de funcţionare a tipului respectiv să rămînă cît mai scurtă.

Bineînţeles, există multe detalii şi reguli speciale în vederea aplicării acestor principii, ne lipseşte aici spaţiul pentru a intra în detalii (vezi şi [4]).

Obiectivele viitoare de cercetare în Institutul de calcul la problema atelierului sînt clare: perfecţionarea modelelor exacte şi amelioarera metodelor euristice de planificare operativă.

FORSCHUNGEN IM RECHENINSTITUT AUF DEM GEBIET DER OPERATIVEN PLANUNG (II)

ZUSAMMENFASSUNG

In diesem zweiten Teil des Berichts über die erhaltenen Forschungsergebnisse über Ablaufplanung werden noch folgende Probleme dargelegt.

7. Das dritte Modell des Falles, wenn mehrere Exemplare aus demselben Typ von Bearbeitungsmaschinen vorhanden sind.

8. Serienherstellung.
9. Pufferbestände.
10. Arbeit in mehreren Schichten.
11. Heuristische Methoden. Schlussfolgerungen.

BIBLIOGAFIE

[1] R. Cruon, Ph. Hervé, *Quelques résultats relatifs à une structure algébrique et son application au problème central de l'ordonnancement.* Rev. Fr. Rech. Op. **9**, *34*, 3–20 (1965).

[2] L. Németi, *Sur le problème de l'ordonnancement dans la fabrication en série.* Rev. Fr. Inf. Rech. Op. **2**, *11*, 47–68 (1968).

[3] — *Programarea în timp a fabricației cu stocuri tampon.* Stud. Cerc. Calcul Econ. Cibern. Econ. **2**, 59–66, 1969.

[4] — *Teză de doctorat,* (Bibl. Inst. Polit. București) 1969.

[5] — *Ablaufplanung bei Arbeit in mehreren Schichten.* Anal. St. Univ. Iași Secția I-a, **21**, *1*, 463–478, (1970).

[6] — *O aplicare specială a metodei Branch and Bound.* Stud. Cercet. Mat. **22**, 8, 1213–1222 (1970).

[7] — *Cercetări în domeniul planificării operative la Institutul de calcul* I. Rev. anal. num. teoria aprox. **1**, *1*. (197).

[8] V. Peteanu, *Asupra compatibilității problemei centrale a ordonanțării.* Stud. Cerc. Mat. **19**, *1*, 47–52 (1967).

[9] — *Optimal Paths in Networks and Generalizations.* Mathematica Cluj. **11**, *2*, 311–328 (1969) și **12**, 1, 169–188 (1970).

Primit la 2. IX. 1972.

Institutul de Calcul din Cluj
al Academiei Republicii Socialiste România