REVISTA DE ANALIZĂ NUMERICĂ ŞI TEORIA APROXIMAŢIEI
Volumul 3, Fascicola 2, 1974, pp. 161–169

DESPRE METODELE EURISTICE PENTRU PROGRAMAREA ÎN TIMP A FABRICAŢIEI

de

L. NÉMETI
(Cluj)

1. Introducere

Despre problema programării în timp a fabricaţiei într-o secţie (atelier) de prelucrare — problema ordonanţării — s-au publicat multe lucrări. Un referat de sinteză despre rezultatele obţinute în această direcţie la Institutul de calcul din Cluj al Academiei R. S. România a apărut în această revistă [2]. Modelele matematice, propuse acolo, reprezintă nişte probleme de programare matematică cu restricţii disjunctive. Probleme de acest tip pot fi în principiu rezolvate cu ajutorul unui procedeu „branch — and — bound". De fapt, la situaţii de producţie de o anvergură industrială problemele devin de dimensiuni aşa de mari încît posibilităţile tehnicii de calcul de astăzi sînt inoperante.

De aceea se impune stabilirea unor metode de rezolvare aproximativă ; soluţiile, obţinute cu ajutorul lor verifică restricţiile impuse, însă nu sîntem siguri dacă se obţine optimul.

În lucrarea de faţă va fi discutată o clasă de metode pe care le numim *metode euristice locale.*

La aceste metode se stabileşte în prealabil, pentru fiecare operaţie $i \in N$*) un *indicator de prioritate* p_i definit pe baza unor principii care diferă de la o metodă la alta. Aceşti indicatori se utilizează în mod următor :

Să presupunem că o maşină de prelucrare a terminat executarea unei operaţii. Fie L mulţimea operaţiilor încă neexecutate şi care pot fi puse pe

*) O operaţie este executarea unui anumit fel de prelucrare la o piesă (sau la un lot de piese) pe o maşină dată de prelucrat. Operaţiile sînt numerotate de la 1 la n. Se notează $N = \{1, 2, \ldots, n\}$. Această mulţime reprezintă mulţimea operaţiilor.

maşina respectivă. Se va determina dintre acestea, operaţia $i \in L$ cu indicatorul p_i maxim şi această operaţie va fi aceea ce va fi pusă pe maşina în cauză.

În cele ce urmează se vor folosi următoarele notaţii:

$M = \{1, 2, \ldots\}$: mulţimea maşinilor de prelucrare, $m \in M$ indică maşina nr. m din parcul de maşini;

$U = \{1, 2, \ldots\}$: mulţimea tipurilor de maşini; $u \in U$ indică tipul cu cu nr. u de maşini.

$u = u(m)$: funcţia definită pe M şi cu valori în U. Ea indică tipul u al maşinii m.

$R = \{1, 2, \ldots\}$: mulţimea reperelor ce figurează în planul perioadei în cauză pentru a fi prelucrate.

O notaţie echivalentă pentru o operaţie $i \in N$ este: (r, u). Operaţia i constă din prelucrarea reperului r pe o maşină de tip u. În mod simbolic se scrie $i = (r, u)$.

t_i: Durata execuţiei operaţiei $i \in N$.

Mulţimea N admite partiţia

$N = \bigcup_{u \in U} N_u$, N_u fiind constituită prin operaţiile ce se execută pe o maşină de tip u.

$H = \{(i, j)\} \subset N \times N$: o mulţime de perechi de operaţii. Relaţia $(i, j) \in H$ exprimă faptul că operaţia i este predecesorul tehnologic al operaţiei j.

O anumită fază a programării este definită prin mulţimea $N^+ \subset N$ care conţine operaţiile programate pînă la faza în cauză. Se notează $N^- = N - N^+$ mulţimea operaţiilor încă neprogramate. La fiecare fază, mulţimii N^+ i se adaugă (cel puţin) o operaţie.

L_u: mulţimea operaţiilor încă neprogramate (în faza considerată) care urmează să fie executate pe tipul u şi ai căror predecesori tehnologici au fost programaţi. De fapt, L_u conţine reperele respective:

$$L_u = \{r | (r, u) \in N_u^-;\ ((r, v), (r, u)) \in H;\ (r, v) \in N^+\}. \tag{1.1}$$

x_i (respectiv x_{ru}): momentul de începere al operaţiei i programate ($i \in N_u^+$).

y_i (respectiv y_{ru}): momentul de terminare al operaţiei $i \in N_u^+$. Are loc

$$y_i = x_i + t_i. \tag{1.2}$$

η_i (respectiv η_{ru}): momentul de sosire al reperului r în faţa maşinilor de tip u (unde are loc $i = (r, u)$). Se presupune deci, că predecesorul tehnologic al operaţiei i este element al mulţimii N^+.

Are loc

$$x_i \geqslant \eta_i. \tag{1.3}$$

$\bar{\eta}_u$: momentul sosirii primului reper care urmează să fie prelucrat pe tipul u. Are loc

$$\bar{\eta}_u = \min_{(r,u) \in N^-} \eta_{ru} = \min_{r \in L_u} \eta_{ru} \tag{1.4}$$

Dacă $L_u = \emptyset$, se pune $\bar{\eta}_u = \infty$.

τ_m: momentul cînd maşina m devine liberă terminînd ultima operaţie din $N_{u(m)}^+$ ce i s-a repartizat.

2. Utilizarea indicatorilor de prioritate

În paragraful următor se va da un şir întreg de reguli, de diferite formule privind stabilirea priorităţii: Să presupunem că s-a ales una din acestea, adică ştim să calculăm *mărimea indicelui de prioritate* p_i (respectiv p_{ru}) pentru o operaţie $i = (r, u)$. Urmează să se arate cum sînt folosiţi aceşti indici, adică să se expună algoritmul de programare.

În acest scop definim noţiunea *momentului luării deciziei* pentru o maşină m. Într-o anumită fază a programării, maşina m a terminat executarea unei operaţii $(r, u(m)) \in N^+$ în momentul τ_m. Pe de altă parte, $\bar{\eta}_{u(m)}$ este momentul sosirii primului reper q spre a fi prelucrat pe această maşină: $(q, u(m)) \in N^-$. Momentul luării deciziei va fi

$$\tau'_m = \max(\tau_m, \bar{\eta}_{u(m)}). \tag{2.1}$$

Lista L_u a candidaţilor (formula (1.1)) va fi întocmită pentru acest moment.

Algoritmul nr. 1. La început, cînd $N^+ = \emptyset$, L_u conţine reperele r pentru care operaţia (r, u) nu are predecesor tehnologic. Valoarea iniţială τ'_m pentru fiecare maşină $m \in M$ se consideră dată respectiv stabilită.

Pasul 1. Se determină maşina m pentru care are loc

$$\tau_m = \min_{k \in M} \tau'_k. \tag{2.2}$$

Pasul 2. Se determină reperul r pentru care operaţia $(r, u(m))$ verifică relaţia

$$p_{ru} = \max_{q \in L_u} p_{qu}, \quad u = u(m). \tag{2.3}$$

Dacă există mai multe repere r care verifică această relaţie, se poate folosi un indicator secundar p'_{ru}, eventual unul terţiar p''_{ru} etc.

Pasul 3. Se determină $(i = (r, u(m)))$

$$\eta_i := x_l + t_l \quad (l, i) \in H \quad (l \in N^+) \tag{2.4}$$

$$x_i := \max(\eta_i, \tau'_m), \tag{2.5}$$

adică operaţia i se programează pe maşina m.

Pasul 4. Dacă există o operaţie $j = (q, u) \in L_u$ pentru care are loc

$$\max\,(\eta_j, \tau'_m) + t_j \leqslant x_i \quad (p_j \leqslant p_i) \tag{2.6}$$

atunci operaţia j se programează şi ea, punîndu-se

$$x_j := \max\,(\eta_j, \tau'_m). \tag{2.7}$$

Astfel de cazuri pot interveni numai la o parte din indicatorii de prioritate definiţi în paragraful 3.

Dacă există mai multe operaţii j ce verifică relaţia (2.6), atunci pasul 4 se înlocuieşte cu algoritmul nr. 2 (a se vedea mai jos).

Pasul 5.

$$\tau_m := x_i + t_i \tag{2.8}$$

$$L_u := L_u - \{r, q, q', \ldots\}, \tag{2.9}$$

unde $q, q', \ldots$ sînt reperele la care s-a aplicat pasul 4 respectiv algoritmul nr. 2.

$$L_v := L_v \cup \{s\}, \tag{2.10}$$

unde s parcurge valorile $r, q, q', \ldots$ şi tipul v este definit prin relaţia

$$((s, u), (s, v)) \in H, \tag{2.11}$$

adică tipul v este succesorul tehnologic al tipului u în raport cu reperul s (vom avea aşadar atîtea tipuri v cîte valori ia parametrul s).

$$N^+ := N^+ \cup \{r, q, q', \ldots\} \tag{2.12}$$

$$N^- := N^- - \{r, q, q', \ldots\}. \tag{2.13}$$

Dacă $N^- = \emptyset$, calculul s-a terminat

Pasul 6. Se recalculează mărimile τ'_k, $k \in M$ cu ajutorul formulelor (1.1), (1,4) şi (2.1). Menţionăm că o schimbare a valorilor τ'_k are loc numai la maşina m precum şi eventual la maşinile din tipurile v.

Pasul 7. Se recalculează indicii de prioritate p_i pentru operaţiile $i \in N^-$, dacă este cazul. Se reia pasul 1.

Algoritmul nr. 2. Inlocuieşte pasul 4 din algoritmul nr. 1, dacă există o submulţime $J_u \subset L_u$ (cu $|J_u| \geqslant 2$), fiecare element $j \in J_u$ verificînd condiţia (2.6). Scopul este de a umple cît mai complet cu alte operaţii intervalul de staţionare $[\tau'_m, x_i[$, apărut în urma programării operaţiei i.

Pasul A. Se stabileşte

$$y := \tau'_m \text{ şi} \tag{2.14}$$

se determină mulţimea J_u (vezi pasul 4 din algoritmul nr. 1).

Pasul B. Din J_u se alege operaţia $j = (q, u)$ respectiv reperul q cu cel mai mic η_j. Dacă există mai multe operaţii cu η_j minim, se poate folosi un indicator de prioritate secundar p'_j etc.

Pasul C. Se determină

$$\begin{cases} x_j := \max\,(y, \eta_j) \\ J_u := J_u - \{q\} \\ y := x_j + t_j. \end{cases} \tag{2.15}$$

Pasul D. Se determină mulţimea $K_u \subset J_u$ conţinînd operaţiile k pentru care are loc

$$\max\,(y, \eta_k) + t_k \leqslant x_i. \tag{2.16}$$

Pasul E. Dacă $K_u = \emptyset$, se sare la algoritmul nr. 1 pasul 5, altfel urmează

Pasul F. $J_u := K_u$, se sare la pasul B.

Menţionăm că algoritmul nr. 2 reprezintă doar un procedeu euristic pentru atingerea scopului (umplerea intervalului $[\tau'_m, x_i]$). S-ar putea concepe eventual şi alte metode.

3. Regulile de prioritate

Se mai introduc notaţiile:

ξ^r: termenul de predare prescris pentru reperul r;

$T_i^0 = T'_{ru}$: suma timpilor de execuţie de la prima operaţie a reperului r pînă la începerea operaţiei (r, u);

$T_i^* = T^*_{ru}$: suma timpilor de execuţie de la terminarea operaţiei (r, u) pînă la terminarea tuturor prelucrărilor reperului r;

$T_r = T_i^0 + t_i + T_i^*$: timpul total de prelucrare a reperului r;

v_r: valoarea (totalul cheltuielilor de fabricaţie) reperului r finisat;

V_r: valoarea (importanţa economică) întregii comenzi din care reperul r face parte.

Există un mare număr de astfel de reguli, respectiv formule, în vederea stabilirii indicatorilor de prioritate. Ele sînt parţial contradictorii. Pentru unele din ele există şi încercări de justificare teoretică, pentru restul se poate doar afirma că regula în cauză este „plauzibilă", „acceptabilă". În general este aşa că o anumită regulă care a dat rezultate bune la o problemă dată, concretă, poate da rezultate slabe la alte probleme concrete.

Astfel de formule sînt (K este o constantă pozitivă arbitrară):

(3.1)

a) $p_i = K - \eta_i$ sau
$p_i = \tau'_m - \eta_i$: ordinea sosirii sau timpul minim de aşteptare (primul sosit, primul servit)

b) $p_i = K - \xi^r$: cel mai scurt termen de predare

c) $p_i = K - (\xi^r - T^r)$: cea mai mică rezervă totală de timp

d) $p_i = K - [\xi^r - \max(\tau'_m, \eta_i)]$: termenul de predare cel mai apropiat

e) $p_i = K - [\xi^r - \max(\tau'_m, \eta_i) - t_i - T_i^*]$: cea mai mică rezervă de timp

f) $p_i = v_r$: cel mai costisitor reper

g) $p_i = V_r$: cea mai importantă comandă

h) $p_i = K - \max(\eta_i - \tau'_m, 0) + t_i$: ordinea terminării operaţiei (primul plecat, primul servit)*

i) $p_i = t_i$: cel mai mare timp de execuţie a operaţiei

j) $p_i = K - t_i$: cel mai mic timp de execuţie a operaţiei

k) $p_i = T_i^0$: cel mai mare timp de prelucrare pentru operaţiile premergătoare

l) $p_i = K - T_i^0$: cel mai mic timp de prelucrare pentru operaţiile premergătoare

m) $p_i = T_i^*$: cel mai mare timp de prelucrare restantă

n) $p_i = K - T_i^*$: cel mai mic timp de prelucrare restantă

o) $p_i = T_r$: cel mai mare timp total de prelucrare a reperului

p) $p_i = K - T_r$: cel mai mic timp total de prelucrare a reperului.

Menţionăm că pasul 4 din algoritmul nr. 1 (respectiv algoritmul nr 2) nu intervine la folosirea formulelor a) şi h).

Se pare că formulele cele mai semnificative sînt: a), e), h), j) şi m). Se mai pot concepe şi alte asemenea formule.

S-au întreprins cîteva cercetări ([1], [3]) privind eficacitatea unora dintre formulele a) ... p); în lucrarea [3] sînt rezumate concluziile unor cercetări stabilite prin simulare, la o scară destul de mare (cca. 10 mii de repere cu cîte 10 operaţii). Rezultatul acestor cercetări pare a fi următorul: nici una din regulile studiate nu arată o superioritate netă faţă de altele. Mai avantajos par a fi regulile combinate de prioritate, mai ales cele disjunctive cu care ne vom ocupa acum. Combinarea indicatorilor de prioritate poate fi făcută în trei feluri:

1) Aditiv: $p_i = p_i^\alpha + p_i^\beta$ (α şi β sînt alese din literele $a \ldots p$ din formulele (3.1)).

2) Multiplicativ: $p_i = p_i^\alpha \cdot p_i^\beta$.

* Aşa numita regulă FOFO (first off, first on).

3) Disjunctiv: Se stabileşte un prag c^α pentru indicatorul p^α. Dacă cel mai mare indicator p al operaţiilor din L_u este mai mare (sau egal) decît c^α, atunci se va programa operaţia posedînd acest indicator p^α maxim. Dacă toate operaţiile din L_u posedă indicatori p^α sub pragul c^α, atunci se va programa operaţia cu indicator p^β maxim.

După [3], metoda 3) poate furniza rezultate simţitor mai bune, decît regulile elementare (3. 1) sau combinările lor conform metodelor 1) sau 2).

Pragul c^α poate varia de la o operaţie la alta. De exemplu, la rezerva minimă (p^e) se poate pune

$$c^e = K - \lambda(t_i + T^*) \tag{3.2}$$

cu $0 < \lambda < 1$ (de ex. $\lambda = 0,2$); adică, dacă rezerva de timp $\rho = K - p^e$ scade sub 100λ % din timpul minim necesar pentru executarea operaţiilor încă neprogramate, atunci se va folosi p^e, în caz contrar alt indicator (de exemplu p^h).

Se pot indica foarte multe astfel de combinaţii (α, β); eficacitatea lor poate depinde în bună măsură şi de alegerea corespunzătoare a pragului c^α.

Un alt exemplu, în afară de cel din aliniatul precedent ($\alpha = e$, $\beta = h$), este $\alpha = a$, $\beta = j$. Adică: Dacă primul reper sosit a aşteptat pînă la momentul τ'_m peste durata de prag c^a, atunci acesta urmează să fie prelucrat. Dacă timpul de aşteptare al fiecărui reper din L_u este sub c^a, atunci reperul cu cel mai mic timp de prelucrare (p^j) va fi ales pentru a fi programat. Pentru pragul c^a se poate pune de exemplu $c^a = \lambda t_i$ $(0 < \lambda)$.

4. Eficienţa şi modalităţile de utilizare ale metodelor euristice locale

Pentru evaluarea eficacităţii ordonanţării obţinute se cere în primul rînd respectarea termenelor de predare impuse. Dacă unele termene vor fi depăşite, atunci pot fi aplicate eventual penalizări.

În al doilea rînd, se cere ca funcţia aleasă de optimizat să obţină valori cît mai bune (penalizările susmenţionate pot fi eventual incluse în valoarea funcţiei de optimizat).

Subliniem faptul că metodele euristice locale lucrează fără funcţii explicite de optimizat. Este dealtfel destul de greu de a evalua „calitatea" unui rezultat obţinut prin folosirea uneia dintre metodele sus indicate. Două metode se oferă practicii pentru o astfel de evaluare.

a) Programul elaborat, respectiv valoarea unei funcţii scop $z = f(X)$ — dacă vectorul X reprezintă ordonanţarea obţinută — se compară cu rezultatele obţinute în secţie înaintea aplicării ordonanţării elaborate.

b) Se elaborează mai multe ordonanţări, folosind diferite variante ale formulei (3.1) respectiv combinări ale acestora, se calculează valoarea funcţiei scop la fiecare ordonanţare şi se alege varianta cea mai favorabilă.

Metoda b) este posibilă datorită faptului că volumul calculelor la metoda indicatorilor de prioritate nu este mare şi un calculator electronic poate determina în timp util mai multe variante. Se înţelege că ambele metode a) şi b) pot fi folosite concomitent.

Observăm că pentru a putea întocmi un program de fabricaţie pentru un interval de timp $\vartheta = [\theta_0, \theta[$ dat, avem nevoie — în afara datelor obişnuite: parcul de maşini, repere de prelucrat şi lotizarea lor, tehnologia de prelucrare etc. — de anumite date *iniţiale*:

— pentru fiecare maşină, momentul cînd ea devine liberă în intervalul ϑ pentru prelucrările cerute;

— pentru fiecare reper, momentul cînd acesta devine disponibil pentru a fi prelucrat în secţie, respectiv în ce stare de prelucrare se găseşte fiecare reper găsit în secţie în momentul θ_0.

Cu alte cuvinte, avem nevoie de un *inventar complet şi detaliat al producţiei neterminate* pentru momentul θ_0, lucrare care în general nu poate fi întocmită în condiţii bune decît atunci, cînd există o *urmărire operativă mecanizată* a stadiului producţiei.

După ce ordonanţarea elaborată s-a definitivat şi a fost pusă în aplicare, ea nu va rămîne, în general, nemodificată în decursul intervalului ϑ de plan. Dimpotrivă, fiecare modificare mai importantă în derularea programului — depăşiri de termeni de sosire, abateri de norme prevăzute de timp, un rebut mai important, defectarea unei maşini, etc. — compromite ordonanţarea elaborată. În acest caz, programul trebuie recalculat pentru restul perioadei, ţinînd cont de noile condiţii. Pentru aceasta însă se cere din nou un inventar al producţiei neterminate.

Menţionăm faptul important că metoda indicatorilor de prioritate poate fi folosită şi în mod decentralizat. În acest caz nu se întocmeşte un program în prealabil pentru perioada de planificare, nu se cere inventarul producţiei neterminate, ci programarea operaţiei se face de la caz la caz în momentele luării deciziei definite la începutul paragrafului 2. Deciziile pot fi luate cu puţine calcule simple, folosind eventual unele tabele auxiliare, şi de către tehnicianul, maistrul care coordonează programarea activităţii maşinii în cauză.

Utilizarea decentralizată are aşadar multe avantaje, ea prezintă însă şi neajunsuri serioase. Intrucît nu se întocmeşte un program prealabil, nu se ştie în ce măsură termenele sînt respectate; nefiind elaborate mai multe variante, nu există posibilitatea alegerii variantei celei mai avantajoase.

Putem să tragem următoarele concluzii cu privire la alternativa: folosire centralizată sau decentralizată a metodei.

În uzinele, unde urmărirea operativă a stadiului producţiei nu este mecanizată, singura posibilitate este utilizarea decentralizată a metodei.

În celelalte cazuri se recomandă a se proceda în felul următor: Se întocmeşte un program de fabricaţie centralizat, elaborat pe o perioadă uzuală de programare (săptămînă, chenzină, lună, etc.). Acest program se obţine prin elaborarea mai multor variante şi prin alegerea celei mai favorabile. Producţia se lansează conform acestui program. În cazul modificărilor neprevăzute mai sus menţionate, modificările cuvenite ale programului se fac în mod decentralizat (fără inventarieri noi.)

UEBER HEURISTISCHE METHODEN DER ABLAUFPLANUNG DER FERTIGUNG

ZUSAMMENFASSUNG

In der Arbeit werden sogenannte lokale heuristische Lösungsmethoden des Problems der Ablaufplanung (Maschinenbelegungsproblem) dargestellt. Bei diesen Methoden wird für jeden Arbeitsgang eine Zahl, der Prioritatsindikator angegeben, auf Grund dessen der jeweils durchzuführende Arbeitsgang ausgewählt wird. Es wird ein Algorithmus zur Verwendung dieser Prioritätsindikatoren angegeben, sowie mehrere Definitionsformeln für dieselben vorgeschlagen. Es werden schliesslich die Anwendungsmöglichkeiten dieser Methoden diskutiert.

BIBLIOGRAFIE

[1] Cervera S., Sussamann B., *Expériences sur des heuristiques de planification à court terme*, SEMA, D. S. Note de travail **102**, 1969.

[2] Németi L., *Cercetări în domeniul planificării operative la Institutul de calcul I, II*, Rev. anal. num. teor. aprox. **1**, 53—63 (1972) şi **2**, 89—104 (1973).

[3] Opitz H., *Moderne Produktionstechnik*, Girardet, Essen, 1970.

Institutul de calcul din Cluj al Academiei Republicii Socialiste România.

Primit la 22. III. 1974.