

Ordonanțarea în flux

În general, ordonanțarea pe mai multe utilaje este o problemă de optimizare combinatorială foarte grea.

În cazul ordonanțării în flux, reperele parcurg utilajele (în vederea procesării) în aceeași ordine.

Să considerăm m repere care se prelucrează **în flux** pe utilajele U_1, U_2, \dots, U_n : întâi pe U_1 , apoi pe U_2 ș.a.m.d., ultima procesare a fiecărui reper având loc pe utilajul U_n . Există deci $m!$ moduri diferite de repartizare a celor m repere în **pozițiile** $1, 2, \dots, m$ ale **listei de lansare în execuție**.

În continuare, **fixăm o ordine de lansare** în execuție a reperelor. Din acest moment, un reper nu va mai fi identificat prin codul său ci prin **poziția** pe care o ocupă în ordinea fixată: astfel, dacă reperul cu codul x ocupă poziția i în ordinea fixată, nu ne vom mai referi la el ca fiind „reperul x ” ci „reperul din poziția i ”. Fie d_{ij} durata procesării reperului din poziția i pe utilajul U_j .

În cele ce urmează vom da o procedură algoritmică simplă de calcul a duratei prelucrării tuturor reperelor, bineînțeles funcție de ordinea de lansare în execuție prestabilită.

În raport cu un moment de referință 0 care marchează începutul procesului de prelucrare a reperelor, vom nota cu t_{ij} momentul **terminării** procesării reperului **din poziția** i a listei de lansare în execuție, pe utilajul U_j . Este clar că termenul t_{mn} la care ultimul reper din listă își termină prelucrarea pe ultimul utilaj va semnifica și durata întregului proces de prelucrare.

În evaluarea acestor termene, două situații sunt posibile:

- Procesarea reperului din poziția „curentă” i pe utilajul U_{j-1} se termină **mai târziu** decât procesarea reperului din poziția anterioară $i - 1$ pe utilajul „curent” U_j – vezi figura 1
-

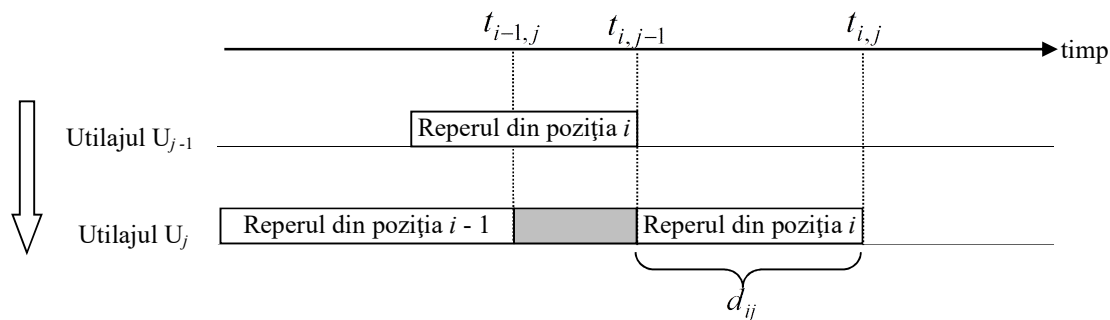


Figura 1

În această situație, utilajul U_j va sta **în așteptare** în intervalul de timp $[t_{i-1,j}, t_{i,j-1}]$ de unde:

$$t_{ij} = t_{i,j-1} + d_{ij} \quad (1)$$

- Procesarea reperului din poziția i pe utilajul U_{j-1} se termină **mai devreme** decât procesarea reperului din poziția anterioară $i - 1$ pe utilajul U_j – vezi figura 2

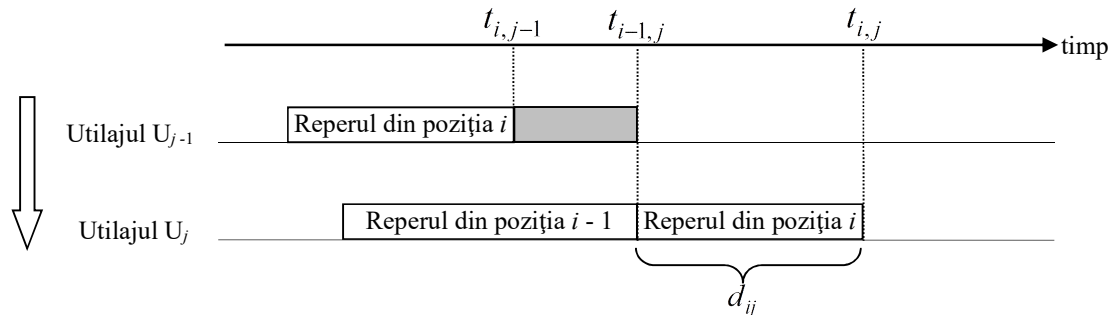


Figura 2

De această dată, reperul din poziția i va sta **în așteptare** în intervalul de timp $[t_{i,j-1}, t_{i-1,j}]$ și de aici:

$$t_{ij} = t_{i-1,j} + d_{ij} \quad (2)$$

Relațiile (1) și (2) se reunesc în formula compactă:

$$t_{ij} = \max\{t_{i-1,j}; t_{i,j-1}\} + d_{ij} \quad (3)$$

în care $t_{0j} = t_{i0} = 0 \quad i = 1, \dots, m \quad j = 1, \dots, n$.

Exemplul 1 Reperele a, b, c, d se prelucrează **în flux** pe utilajele U_1, U_2, U_3, U_4, U_5 . Duratele de prelucrare (în minute) sunt date în tabelul 1

Tabelul 1

| Reper | Durata de prelucrare (minute) | | | | |
|-------|-------------------------------|-------|-------|-------|-------|
| | U_1 | U_2 | U_3 | U_4 | U_5 |
| a | 3 | 4 | 3 | 7 | 2 |
| b | 3 | 7 | 2 | 8 | 5 |
| c | 1 | 2 | 4 | 3 | 7 |
| d | 4 | 3 | 7 | 2 | 8 |

Vom determina, pe baza relației (3), durata procesului de prelucrare a întregului lot în cazul în care reperele se lansează în execuție în ordinea:

i) a, b, c, d

ii) b, d, c, a

Calcululele sunt sistematizate în tabelele 2i) și 2ii). Cele două săgeți arată sensul de efectuare a calculului: pe prima linie, apoi pe a doua ș.a.m.d. iar pe fiecare linie de la stânga la dreapta.

Tabelul 2i)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>a</i> | 1 | 3 | 7 | 10 | 17 | 19 |
| <i>b</i> | 2 | 6 | 14 | 16 | 25 | 30 |
| <i>c</i> | 3 | 7 | 16 | 20 | 28 | 37 |
| <i>d</i> | 4 | 11 | 19 | 27 | 30 | 45 |

Tabelul 2ii)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>b</i> | 1 | 3 | 10 | 12 | 20 | 25 |
| <i>d</i> | 2 | 7 | 13 | 20 | 22 | 33 |
| <i>c</i> | 3 | 8 | 15 | 24 | 27 | 40 |
| <i>a</i> | 4 | 11 | 19 | 27 | 34 | 42 |

În concluzie, dacă reperele sunt lansate în ordinea *a*, *b*, *c*, *d*, întregul proces durează $t_{45} = 45$ minute. Procesarea reperelor în ordinea *b*, *d*, *c*, *a* durează numai $t_{45} = 42$ minute.

În ceea ce privește rezolvarea problemei de ordonanțare în flux deosebim două cazuri importante.

- Ordonanțarea în flux pe două utilaje;
- Ordonanțarea în flux pe $m \geq 3$ utilaje.

Separarea celor două cazuri este urmarea complexității lor diferite: în primul caz avem de a face cu o problemă de optimizare ușoară; al doilea caz este tot atât de “dificil” cât este o problemă a comisvoiajorului!

1. Ordonanțarea în flux pe două utilaje

Reamintim problema:

Un număr de repere sunt prelucrate pe două utilaje U_1 și U_2 . Fiecare reper se prelucrează **mai întâi** pe U_1 și **apoi** pe U_2 , duratele operațiilor fiind cunoscute. În ce ordine vor fi lansate reperele pentru ca timpul necesar terminării lor să fie **minim**?

Următorul algoritm, datorat lui JOHNSON (1954), rezolvă problema într-un număr de pași egal cu numărul reperelor. Pentru simplitate vom presupune că toate duratele de prelucrare sunt diferite între ele. În cazul în care mai multe operații – fie pe U_1 fie pe U_2 fie pe amândouă – au durate egale vom modifica ușor unele din ele pentru ca toate să fie distincte!

Start: Toate reperele sunt declarate **neprogramate**.

Pasul 1 Se identifică operația cu cea mai mică durată, aferentă unui reper **neprogramat**.

Pasul 2 Dacă operația selectată se execută pe **primul** utilaj U_1 , reperul respectiv se programează **la început**, dar **după** reperatele anterior programate la început.

Dacă operația selectată se execută pe **al doilea** utilaj U_2 , reperul în cauză se va programa **la sfârșit** dar **înaintea** reperelor anterior programate la sfârșit.

Se șterge reperul programat din lista reperelor neprogramate și se revine la pasul 1 în cadrul unei noi iterații.

Exemplul 2 Reperatele R_1, \dots, R_8 se procesează în flux pe utilajele U_1 și U_2 . În tabelul 3 sunt date duratele operațiilor (ore).

Tabelul 3

| Reper | R_1 | R_2 | R_3 | R_4 | R_5 | R_6 | R_7 | R_8 |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Durata operației pe U_1 | 8 | 4 | 9 | 6 | 2 | 5 | 8 | 3 |
| Durata operației pe U_2 | 7 | 2 | 5 | 10 | 3 | 5 | 4 | 6 |

Să se determine ordinea de lansare în execuție a reperelor care minimizează durata întregului proces de prelucrare.

Soluție. Perturbăm „ușor” unele durate pentru ca toate să fie diferite – vezi tabelul 4. După cum se va vedea în final, unele modificări sunt de prisos în timp ce altele conduc la mai multe ordonări optime – adică cu aceeași durată totală minimă! Important este faptul că, prin perturbare, la fiecare iterație a algoritmului, un singur reper este programat! Pentru calculul termenelor de terminare t_{ij} ale diferitelor operații se revine la valorile inițiale ale duratelor.

Tabelul 4

| Reper | R_1 | R_2 | R_3 | R_4 | R_5 | R_6 | R_7 | R_8 |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Durata operației pe U_1 | 8 | 4 | 9 | 6 | 2 | 5,01 | 8,01 | 3,01 |
| Durata operației pe U_2 | 7 | 2,01 | 5 | 10 | 3 | 5,02 | 4,01 | 6,01 |

(modificările s-au făcut „la întâmplare!”)

Start: declarăm toate reperatele neprogramate.

Iterația 1 Cea mai scurtă durată - 2 ore – o are prelucrarea reperului R_5 pe **primul** utilaj U_1 , drept care reperul R_5 este programat **la începutul** listei de lansare:

| | | | | | | | |
|-------|--|--|--|--|--|--|--|
| R_5 | | | | | | | |
|-------|--|--|--|--|--|--|--|

Ștergem reperul R₅ din lista reperelor neprogramate.

Iterația 2 Minimul duratelor operațiilor neprogramate este 2,01; operația respectivă este aferentă reperului R₂ și se execută pe **al doilea utilaj** U₂. Programăm reperul R₂ **la sfârșitul** listei de lansare:

| | | | | | | | | |
|----------------|--|--|--|--|--|--|--|----------------|
| R ₅ | | | | | | | | R ₂ |
|----------------|--|--|--|--|--|--|--|----------------|

Ștergem reperul R₇ din lista reperelor neprogramate.

Continuând, se găsește în final următoarea ordine de lansare în execuție:

| | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| R ₅ | R ₈ | R ₆ | R ₄ | R ₁ | R ₃ | R ₇ | R ₂ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|

Pentru ordinea rezultată calculăm termenele de terminare t_{ij} ale diferitelor operații, aplicând relația (3) – vezi tabelul 5

Tabelul 5

| Reper | R ₅ | R ₈ | R ₆ | R ₄ | R ₁ | R ₃ | R ₇ | R ₂ |
|--|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Momentele de terminare ale operațiilor pe U ₁ | 2 | 5 | 10 | 16 | 24 | 33 | 41 | 45 |
| Momentele de terminare ale operațiilor pe U ₂ | 5 | 11 | 16 | 26 | 33 | 38 | 45 | 47 |

Durata minimă de procesare a tuturor reperelor este de 47 ore.

Notă: Se observa că singura perturbare care a contat a vizat valoarea 5 prezentă de trei ori în tabelul 4. Prin „permutarea” perturbărilor acestei valori (vezi tabelul 4) - restul modificărilor rămânând aceleași sau chiar schimbându-se!! - se obțin alte două ordonări

| | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| R ₅ | R ₈ | R ₄ | R ₁ | R ₃ | R ₆ | R ₇ | R ₂ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|

și

| | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| R ₅ | R ₈ | R ₄ | R ₁ | R ₆ | R ₃ | R ₇ | R ₂ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|

cu aceeași durată totală de execuție, conform tabelelor 6 și 7.

Tabelul 6

| Reper | R ₅ | R ₈ | R ₄ | R ₁ | R ₃ | R ₆ | R ₇ | R ₂ |
|--|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Momentele de terminare ale operațiilor pe U ₁ | 2 | 5 | 11 | 19 | 28 | 33 | 41 | 45 |
| Momentele de terminare ale operațiilor pe U ₂ | 5 | 11 | 21 | 28 | 33 | 38 | 45 | 47 |

Tabelul 7

| Reper | R ₅ | R ₈ | R ₄ | R ₁ | R ₆ | R ₃ | R ₇ | R ₂ |
|--|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Momentele de terminare ale operațiilor pe U ₁ | 2 | 5 | 11 | 19 | 24 | 33 | 41 | 45 |
| Momentele de terminare ale operațiilor pe U ₂ | 5 | 11 | 21 | 28 | 33 | 38 | 45 | 47 |

2. Ordonanțarea în flux pe $m \geq 3$ utilaje

Pentru ordonarea în flux pe cel puțin trei utilaje prezentăm o procedură simplă dar foarte performantă, **euristica NEH** (după inițialele autorilor **NAVAZ**, **ENSCORE** și **HAM**)

Start: Se aranjează reperate într-o listă \mathcal{L} în ordinea **descrescătoare** a **sumelor** duratelor de procesare pe **toate** utilajele.

Se iau primele două reperate și se ordonează așa încât să se minimizeze timpul necesar terminării lor.

Celelalte reperate se vor programa în ordinea în care apar în lista \mathcal{L} după schema din:

Pasul iterativ: Presupunem programate primele $k - 1$ reperate din lista \mathcal{L} . În raport cu aceste reperate, deja programate, pentru reperul din poziția k a listei \mathcal{L} există k locuri posibile de **inserție** – vezi figura 3.

Pentru fiecare loc de inserție se calculează durata procesării celor k reperate, în ordinea rezultată din inserție. Se inserează reperul considerat în locul corespunzător duratei **minime**.

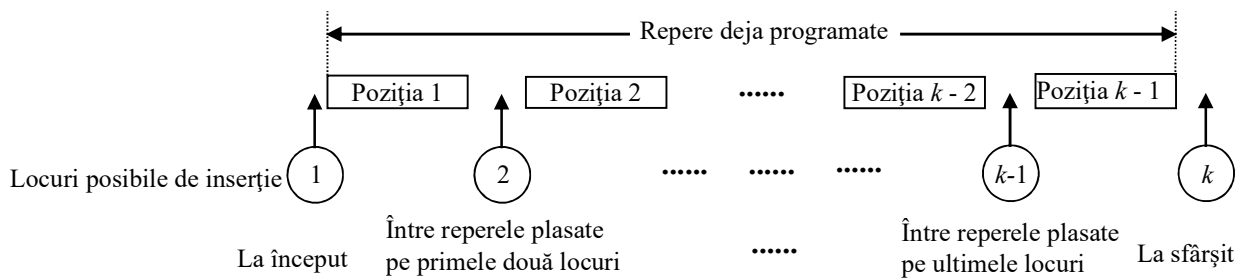


Figura 3

Exemplul 5. Aplicăm algoritmul NEH la programarea lansării în execuție a reperelor din exemplul 1. Sumele duratelor de prelucrare pe cele cinci utilaje sunt date în tabelul 8:

| Reper | a | b | c | d |
|---|-----|-----|-----|-----|
| Durata totală a operațiilor de prelucrare | 19 | 25 | 17 | 24 |

Reperatele vor fi analizate în ordinea b, d, a, c .

Primele două pot fi lansate în execuție fie în ordinea b urmat de d fie invers. Calculele termenelor corespunzătoare t_{ij} de terminare a operațiilor – conform relației (3) – sunt date în tabelele 9i) și 9ii).

Tabelul 9i)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>b</i> | 1 | 3 | 10 | 12 | 20 | 25 |
| <i>d</i> | 2 | 7 | 13 | 20 | 22 | 33 |

Tabelul 9ii)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>d</i> | 1 | 4 | 7 | 14 | 16 | 24 |
| <i>b</i> | 2 | 7 | 14 | 16 | 24 | 29 |

Se constată că lansarea reperului *d* înaintea lui *b* duce la o durată totală de execuție (a celor două repere!) mai mică: numai 29 minute față de 33 minute cât ar dura procesul corespunzător ordinii inverse. **Algoritmul cere ca această precedență să fie respectată în continuare!**

Iterația 1 Inserarea reperului următor *a* conduce la examinarea următoarelor succesiuni:

$$(a, d, b); (d, a, b); (d, b, a)$$

Termenele t_{ij} corespunzătoare sunt sistematizate în tabelele 10 i),ii) și iii)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>a</i> | 1 | 3 | 7 | 10 | 17 | 19 |
| <i>d</i> | 2 | 7 | 10 | 17 | 19 | 27 |
| <i>b</i> | 3 | 10 | 17 | 19 | 27 | 32 |

Tabelul 10i)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>d</i> | 1 | 4 | 7 | 14 | 16 | 24 |
| <i>a</i> | 2 | 7 | 11 | 17 | 24 | 26 |
| <i>b</i> | 3 | 10 | 18 | 20 | 32 | 37 |

Tabelul 10ii)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>d</i> | 1 | 4 | 7 | 14 | 16 | 24 |
| <i>b</i> | 2 | 7 | 14 | 16 | 24 | 29 |
| <i>a</i> | 3 | 10 | 18 | 21 | 31 | 33 |

Tabelul 10iii)

Cea mai mică durată corespunde ordinii (*a, d, b*) și aceasta va fi menținută în continuare.

Iterația 2 Pentru ultimul reper *c* din listă există patru posibilități de inserție:

$$(c, a, d, b); (a, c, d, b); (a, d, c, b); (a, d, b, c)$$

Ele sunt analizate în următoarele patru tabele:

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>c</i> | 1 | 1 | 3 | 7 | 10 | 17 |
| <i>a</i> | 2 | 4 | 8 | 11 | 18 | 20 |
| <i>d</i> | 3 | 8 | 11 | 18 | 20 | 28 |
| <i>b</i> | 4 | 11 | 18 | 20 | 28 | 33 |

Tabelul 11i)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>a</i> | 1 | 3 | 7 | 10 | 17 | 19 |
| <i>c</i> | 2 | 4 | 9 | 14 | 20 | 27 |
| <i>d</i> | 3 | 8 | 12 | 21 | 23 | 35 |
| <i>b</i> | 4 | 11 | 19 | 23 | 31 | 40 |

Tabelul 11ii)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>a</i> | 1 | 3 | 7 | 10 | 17 | 19 |
| <i>d</i> | 2 | 7 | 10 | 17 | 19 | 27 |
| <i>c</i> | 3 | 8 | 12 | 21 | 24 | 34 |
| <i>b</i> | 4 | 11 | 19 | 23 | 32 | 39 |

Tabelul 11iii)

| Reper | Poziția în lista de lansare | Termene t_{ij} | | | | |
|----------|-----------------------------|------------------|----------------|----------------|----------------|----------------|
| | | U ₁ | U ₂ | U ₃ | U ₄ | U ₅ |
| <i>a</i> | 1 | 3 | 7 | 10 | 17 | 19 |
| <i>d</i> | 2 | 7 | 10 | 17 | 19 | 27 |
| <i>b</i> | 3 | 10 | 17 | 19 | 27 | 32 |
| <i>c</i> | 4 | 11 | 19 | 23 | 30 | 39 |

Tabelul 11iv)

În concluzie, algoritmul NEH propune ca ordinea de lansare în execuție a reperelor să fie *c*, *a*, *d*, *b* pentru care durata totală este de 33 minute.