Kombinatorikus optimalizálás 12. hét

Pusztai Pál

pusztai@sze.hu

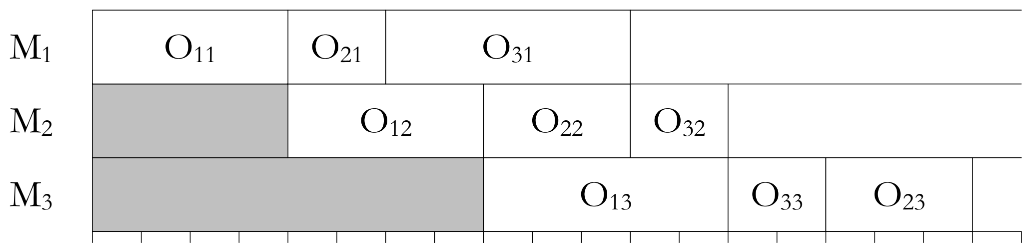
Tartalom

- Ütemezési feladatok
  - Shop ütemezések
  - A permutációs flow shop probléma (PFSP)
    - Johnson algoritmusa két gépre
    - A Nawaz-Enscore-Ham (NEH) algoritmus

Shop ütemezés

Shop ütemezés: Minden végrehajtandó munka több műveletből áll. Minden egyes művelet egy, a művelethez rendelt gépen hajtható végre, és adott a művelet végrehajtási ideje.

- **Open shop ütemezés**: Ha bármely munkára a hozzátartozó műveletek tetszőleges sorrendben végrehajthatók.
- Ha az egyes munkák műveletei kötött sorrendben hajtandók végre.
  - **Job shop ütemezés**: Ha a munkák műveleteihez tartozó gépek sorrendje nem azonos (az újságolvasási példa egy ilyen feladat).
  - **Flow shop ütemezés**: Ha a gépek sorrendje azonos minden munkára.



Egy flow shop ütemezés

- **Permutációs flow shop** probléma: Ha a munkák sorrendje (is) azonos minden gépre. (A munkáknak azt a sorrendjét kell meghatározni, amelyre az ütemezés optimális.)

Megjegyzés: A probléma általában még ennyi kikötés (PFSP) mellett is NP-nehéz marad, de pl. két gépre már ismert hatékony eljárás, ezt be is mutatjuk.

PFSP ütemezés két gépre

Jelölje a gépeket M_1 és M_2 a munkákat pedig rendre $J_1, \dots, J_m$. A J_i $(i = 1, \dots, m)$ munka két műveletből áll, először az O_{i1} műveletet kell az M_1 gépen végrehajtani, ami τ_{i1} ideig tart, majd az O_{i2} műveletet az M_2 gépen ami τ_{i2} ideig tart.

Johnson algoritmusa

1. Legyen $k = 1$ és $l = m$, és tekintsük a nem ütemezett munkák $J = \{J_1, \dots, J_m\}$ halmazát.
2. Keressük meg a $\{\tau_{i1}, \tau_{i2} | J_i\}$ halmaz egy minimális elemét. Jelölje a hozzá tartozó indexet i. Ha i nem egyértelműen meghatározott, akkor a lehetséges indexek közül válasszuk a legkisebbet. Amennyiben $\tau_{i1} = \tau_{i2}$, akkor válasszuk a τ_{i1} elemet.
3. Ha a 2. lépésben választott elem τ_{i1}, akkor helyezzük a J_i munkát a listánk k-adik helyére és töröljük a J halmazból.

   Legyen $k = k + 1$, majd lépjünk az 5. lépésre.
4. Ha a 2. lépésben választott elem τ_{i2}, akkor helyezzük a J_i munkát a listánk l-edik helyére és töröljük a J halmazból.

   Legyen $l = l - 1$, majd lépjünk az 5. lépésre.
5. Ha J üres, akkor vége az eljárásnak. Az **optimális ütemezést** kapjuk meg, ha az eljárás során meghatározott lista sorrendje szerint hajtjuk végre a munkákat késlekedés nélkül.

   Ellenkező esetben $(J \neq \emptyset)$ folytassuk az eljárást a 2. lépéssel.

Tétel: Az $F2||C_{max}$ probléma esetén van olyan optimális ütemezés, amelyben a munkák sorrendje mindkét gépen ugyanaz.

Példa

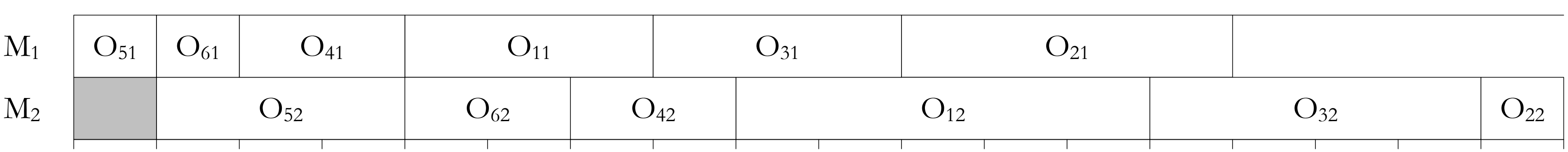
Legyenek az ütemezendő munkák az alábbiak:

Munkák	Műveletek	Végrehajtási idők
J_1	O_{11}, O_{12}	3, 5
J_2	O_{21}, O_{22}	4, 1
J_3	O_{31}, O_{32}	3, 4
J_4	O_{41}, O_{42}	2, 2
J_5	O_{51}, O_{52}	1, 3
J_6	O_{61}, O_{62}	1, 2

Johnson algoritmusa az alábbi listát eredményezi:

$$J_5 \prec J_6 \prec J_4 \prec J_1 \prec J_3 \prec J_2$$

Tehát egy optimális ütemezés a következő (az optimum értéke 18):



Feladatok

- Milyen ütemezést ad Johnson algoritmusa és mekkora lesz a megoldás célfüggvényértéke az alábbi bemenő adatok esetén? A megoldást Gannt diagramon ábrázoljuk!

Munkák	Műveletek	Végrehajtási idők
J_1	O_{11}, O_{12}	2, 4
J_2	O_{21}, O_{22}	4, 1
J_3	O_{31}, O_{32}	3, 2
J_4	O_{41}, O_{42}	2, 3
J_5	O_{51}, O_{52}	4, 3
J_6	O_{61}, O_{62}	1, 2

Egy PFSP heurisztika

Jelölje a gépeket $M_1, \dots M_n$, és a munkákat $J_1, \dots, J_m$. Legyen a végrehajtási idők mátrixa (p_{ij}), ahol p_{ij} azt adja meg, hogy az M_i gépen mennyi a J_j munka végrehajtási ideje.

Legyen $\pi_1, \dots, \pi_k$ egy tetszőleges sorrendje k számú munkának $(1 \le k \le m, \pi_l \in J_1, \dots, J_m, l = 1, \dots, k)$. A $\pi_1, \dots, \pi_k$ munkák teljes feldolgozási ideje kiszámolható az alábbi f függvény segítségével.

$$f(i,j) = 0, \text{ ha } i = 0, \text{ vagy } j = 0,$$
$$f(i,j) = \max\{f(i-1,j), f(i,j-1)\} + p_{i\pi_j}, \text{ ha } i = 1, \dots, n; j = 1, \dots, k.$$

A (rész)permutációhoz tartozó teljes feldolgozási idő: $C_{max}(\pi) = f(n,k)$.

- A NEH (Nawaz, Enscore, Ham) algoritmus

1. Rendezzük a munkákat a feldolgozási összidejük $(\sum_{i=1}^{n} p_{ij}, j = 1, \dots, m)$ szerint csökkenő sorrendbe. Jelölje a kapott permutációt π.

   Legyen $r_1 = \pi_1$, és $k = 1$.
2. Legyen $k = k + 1$.

   Ha $k \le m$, akkor szúrjuk be a π_k munkát r-be úgy, hogy az eredményül kapott k számú munkához tartozó C_{max} érték minimális legyen. (Összesen k darab helyre szúrható be π_k, hiszen r éppen $k-1$ darab munkából áll. Ha több ilyen hely is lenne, akkor válasszuk a legkisebb indexűt.)

   Ha $k = m$ akkor vége az eljárásnak, r az eredmény permutáció, egyébként folytassuk a 2. lépéssel.

Hatékonyság: $\Theta(nm)+O(m \lg m)+O(m^3n)$, azaz $O(m^3n)$.

Példa

Legyen $n = 3$, $m = 4$, és a végrehajtási időket tartalmazó mátrix az alábbi:

$$P = \begin{pmatrix} 2 & 1 & 1 & 3 \\ 1 & 4 & 3 & 1 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

A munkákhoz tartozó teljes végrehajtási idők: $(7, 8, 6, 5)$, így $\pi = (2, 1, 3, 4)$.

$$k = 2 \quad \begin{array}{c} \begin{matrix} 1 & 2 \end{matrix} \\ \begin{pmatrix} 2 & 3 \\ 3 & 7 \\ 7 & 10 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{matrix} 2 & 1 \end{matrix} \\ \begin{pmatrix} 1 & 3 \\ 5 & 6 \\ 8 & 12 \end{pmatrix} \end{array} \quad r = (1, 2)$$

$$k = 3 \quad \begin{array}{c} \begin{matrix} 3 & 1 & 2 \end{matrix} \\ \begin{pmatrix} 1 & 3 & 4 \\ 4 & 5 & 9 \\ 6 & 10 & 13 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{matrix} 1 & 3 & 2 \end{matrix} \\ \begin{pmatrix} 2 & 3 & 4 \\ 3 & 6 & 10 \\ 7 & 9 & 13 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 2 & 3 & 4 \\ 3 & 7 & 10 \\ 7 & 10 & 12 \end{pmatrix} \end{array} \quad r = (1, 2, 3)$$

$$k = 4 \quad \begin{array}{c} \begin{matrix} 4 & 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 3 & 5 & 6 & 7 \\ 4 & 6 & 10 & 13 \\ 5 & 10 & 13 & 15 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{matrix} 1 & 4 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 2 & 5 & 6 & 7 \\ 3 & 6 & 10 & 13 \\ 7 & 8 & 13 & 15 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{matrix} 1 & 2 & 4 & 3 \end{matrix} \\ \begin{pmatrix} 2 & 3 & 6 & 7 \\ 3 & 7 & 8 & 11 \\ 7 & 10 & 11 & 13 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} 2 & 3 & 4 & 7 \\ 3 & 7 & 10 & 11 \\ 7 & 10 & 12 & 13 \end{pmatrix} \end{array}$$

Az eredmény permutáció $r = (1, 2, 4, 3)$, a C_{max} érték 13, ami most az optimumérték is egyben.

Feladatok

- Milyen ütemezést ad a NEH algoritmus és mekkora lesz a megoldás célfüggvényértéke az alábbi bemenő adatok esetén? A megoldást Gannt diagramon ábrázoljuk!

$$n = 3, m = 4, P = \begin{pmatrix} 1 & 2 & 1 & 3 \\ 3 & 3 & 4 & 1 \\ 4 & 5 & 2 & 2 \end{pmatrix}$$