Kombinatorikus optimalizálás 11. hét

Pusztai Pál
pusztai@sze.hu

Tartalom

- Ütemezési feladatok
  - Lineáris és egészértékű programozási modellek
  - Heurisztikák
    - A súlyozott legrövidebb végrehajtási idő elve
    - A Graham-féle lista és LPT algoritmusok

Ütemezési feladatok: egy példa

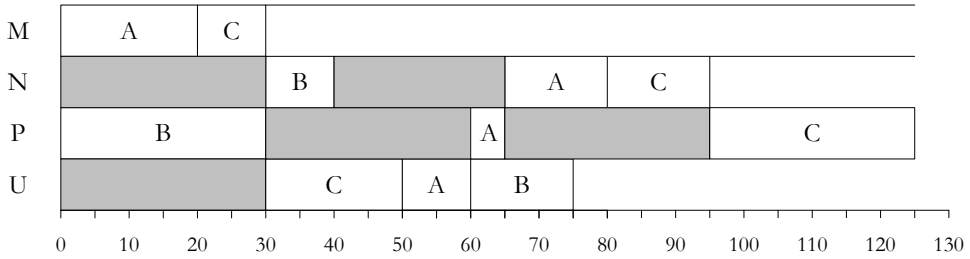
András (A), Béla (B) és Csaba (C) a következő újságokat olvassák: Magyar Nemzet (M), Népszava (N), Pesti Hírlap (P), Új Tükör (U).

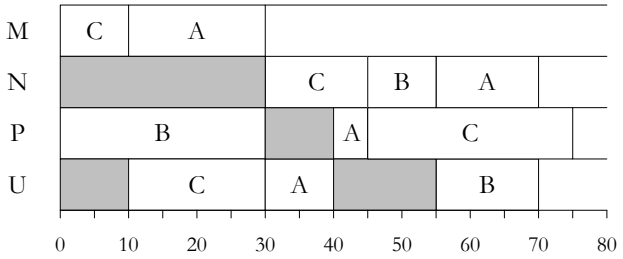
Személy	Sorrend	Percek
A	M, U, P, N	20, 10, 5, 15
B	P, N, U	30, 10, 15
C	M, U, N, P	10, 20, 15, 30

Olvasási szabályok:

- Valamennyien, ha elkezdenek egy újságot olvasni, akkor azt nem szakítják meg.
- Egynél több személy egyidejűleg nem olvassa ugyanazt az újságot.

Feladat: Az újságok olvasásának milyen időrendi beosztása (ütemezése) eredményezi a közös olvasás teljes idejének minimumát?





Ütemezések megadása Gannt diagrammal

Ütemezési modellek

Az ütemezési problémákban adottak páronként különböző gépek és m számú munka, amelyeket az $1, \dots, m$ számokkal fogunk sorszámozni.

Feladat: A munkák végrehajtásának ütemezése a gépeken úgy, hogy valamilyen cél szerint optimális ütemezést kapjunk.

A **munkák végrehajtásának ütemezésén** vagy egyszerűbben a **munkák ütemezésén** azt értjük, hogy a j-edik munkát hozzárendeljük valamely géphez egy S_j **kezdési** és C_j **befejezési idővel** minden j-re.

A munkákhoz minden modellben tartozik egy **végrehajtási idő**, amit p_j-vel szokás jelölni. Ez azt adja meg, hogy mennyi ideig tart a munkát elvégezni. Ennek megfelelően a munkához rendelt kezdési és befejezési időkre a $C_j - S_j = p_j$ feltételnek kell teljesülni.

Megjegyzés: Az előző példában a munkák az egyes személyek az egyes újságokra vonatkozó olvasási tevékenységei, míg gépeknek az újságokat tekinthetjük, hiszen a munkákat ezeken kell végrehajtani.

Ütemezési modellek

- A munkák (modellektől függő) egyéb paraméterei
  - r_j **érkezési idő**. A munka végrehajtása csak ezután kezdődhet, azaz $S_j \geq r_j$.
  - d_j **határidő**.
    - Csak olyan ütemezéseket fogadunk el, amelyekre $C_j \leq d_j$.
    - Megszeghető a határidő, de a célfüggvényben a határidő is szerepel.
  - w_j **súly**, vagy $f_j(t)$ **súlyfüggvény**. Mennyire fontos a munka, ill. annak t időpontra való befejezése.
- Az optimalizálandó célfüggvények
  - Az utolsónak befejezett munka befejezési ideje, azaz a $\max\{C_j : 1 \leq j \leq m\}$ függvény.
  - A befejezési idők összegfüggvénye. Általános esetben, mikor a munkáknak van súlya vagy súlyfüggvénye $\sum_{j=1}^{m} w_j C_j$, ill. $\sum_{j=1}^{m} f_j(C_j)$ függvényeket minimalizáljuk.
  - Ha a munkákhoz határidő is tartozik, akkor a két figyelembe vehető érték (amelyek maximumát, vagy súlyozott összegét minimalizáljuk):
    - **Késési idő** (lateness): $L_j = C_j - d_j$.
    - **Csúszási idő** (tardiness): $T_j = \max\{0, L_j\}$.
    - Cél lehet még az elkésett munkák ($T_j > 0$) számának minimalizálása.
  - Ha érkezési idők is vannak, akkor szokásos a befejezési idők helyett a **folyási időt** (flow time) vizsgálni: $F_j = C_j - r_j$. Ezekben a modellekben az F_j értékek maximuma vagy súlyozott összege a célfüggvény.

Ütemezési modellek

- Egyéb megkötések (extra feltételek)
  - Bizonyos munkákat csak más munkák után lehet elvégezni. (Az újságolvasási példa is ilyen, ahol az egyes személyek egy adott sorrendben olvassák az újságokat.)
  - A munkák végrehajtási ideje függ attól hogy melyik gépen kerül végrehajtásra. Minden munkához egy **végrehajtási vektor** tartozik, ahol az i-edik komponens azt adja meg, hogy az M_i gépen mennyi ideig tart a munkát végrehajtani.
    - A j-edik munka végrehajtási vektorának i-edik komponense p_j/v_i, ahol v_i az M_i gép sebessége.
    - Korlátozott hozzárendelés esetén néhány komponens végtelen a többi megegyezik, azaz a gépek azonosak, csak a munka néhány gépen nem hajtható végre.
  - Megengedjük, hogy a munkák végrehajtása megszakítható legyen. Ekkor a j-edik munkához nem egy darab legalább p_j hosszú időintervallumot kell hozzárendelnünk, valamely gépen, hanem több, egymást nem átfedő intervallumot (akár különböző gépeken), amelyek összhossza legalább p_j.
- Jelölés
  - Az egyes modellek (problémák) megadására egy három részből álló jelölést fogunk használni (lásd jegyzet), ahol:
    - az első mező tartalmazza a gépek számára, típusára vonatkozó információkat,
    - a második mező a modellre vonatkozó egyéb feltételeket,
    - a harmadik a célfüggvényt adja meg.

  pl. A $Pn|prmp|C_{max}$ probléma esetén n párhuzamos azonos gépen ütemezzük a munkákat, a maximális befejezési időt minimalizálva, a munkák megszakítását engedélyezve.

Lineáris programozási modell

Tekintsük a $Pn|prmp|C_{max}$ problémát.

Jelölje $x_{ij} \geq 0$ változó azt az időmennyiséget, amelyet az i-edik gép összesen a j-edik munkával tölt.

$$\sum_{i=1}^{n} x_{ij} = p_j \quad (j = 1, \dots, m)$$

$$\sum_{i=1}^{n} x_{ij} \leq C_{max} \quad (j = 1, \dots, m)$$

$$\sum_{j=1}^{m} x_{ij} \leq C_{max} \quad (i = 1, \dots, n)$$

$$x_{ij} \geq 0, \quad (i = 1, \dots, n;\ j = 1, \dots, m)$$

$$C_{max} \rightarrow \min$$

Egészértékű programozási modellek

Tekintsük az $1||\sum w_j C_j$ problémát.

Egy gép van, a munkákat két paraméter határozza meg (a végrehajtási idő és a munka súlya), cél a teljes súlyozott befejezési idő minimalizálása.

Legyen x_{jk} egy döntési változó, amely 1 ha a j munka az ütemezésben hamarabb kerül végrehajtásra, mint a k, és 0 egyébként.

$$x_{kj} + x_{jk} = 1 \quad (j,k = 1,\dots,m; j \neq k)$$

$$x_{kj} + x_{jl} + x_{lk} \leq 2 \quad (j,k,l = 1,\dots,m; j \neq k; k \neq l; l \neq j)$$

$$x_{jk} \in \{0,1\} \quad (j,k = 1,\dots,m)$$

$$x_{jj} = 0 \quad (j = 1,\dots,m)$$

$$\sum_{j=1}^{m}\sum_{k=1}^{m} w_j p_k x_{kj} + \sum_{j=1}^{m} w_j p_j \rightarrow \min$$

- Megjegyzés
  - A j munka befejezési ideje $\sum_{k=1}^{m} p_k x_{kj} + p_j$, ezért a célfüggvény valóban $\sum w_j C_j$.
  - A munkákra vonatkozó precedencia feltételek könnyen beilleszthetők.
  - A modell nem terjeszthető ki több gépre.

Egészértékű programozási modellek

Az $1||\sum w_j C_j$ probléma alábbi modellje kiegészíthető több gép esetére is.

Feltétel: A végrehajtási idők egészek.

Következmény: Elegendő azokat a lehetséges ütemezéseket vizsgálni, amelyekben a kezdési idők is egészek.

Legyen az x_{jt} változó 1 ha a j munka a t időpontban kezdődik, 0 különben. Legyen $l = \sum_{j=1}^{m} p_j - 1$.

$$\sum_{t=0}^{l} x_{jt} = 1 \quad (j = 1, \dots, m)$$

$$\sum_{j=1}^{m} \sum_{s=\max\{t-p_j+1,0\}}^{t} x_{js} = 1 \quad (t = 0,1, \dots, l)$$

$$x_{jt} \in \{0,1\} \quad (j = 1, \dots, m;\ t = 0,1, \dots, l)$$

$$\sum_{j=1}^{m} \sum_{t=0}^{l} w_j (t + p_j) x_{jt} \to \min$$

Megjegyzés: A változók száma rendkívül nagy lehet.

A legrövidebb végrehajtási idő elve

Egy célfüggvényt **regulárisnak** nevezünk, ha monoton növő a befejezési időkben.

Észrevétel: Reguláris célfüggvények esetében elegendő azon ütemezéseket vizsgálni, amelyekben nincs üres idő, azaz a gépek megszakítás nélkül dolgoznak.

Tekintsük az $1||\sum w_j C_j$ problémát.

Ezen célfüggvény esetén a w_j/p_j hányados tekinthető a j munka egységenkénti fontosságának, így természetes gondolat azon munkákat ütemezni először, amelyekre ez a hányados nagyobb. Ezt az elvet „súlyozott legrövidebb végrehajtási idő" elvének nevezzük, és a WSPT-vel („weighted shortest processing time") rövidítjük.

Tétel: A WSPT elv optimális ütemezést ad az $1||\sum w_j C_j$ problémára.

Megjegyzés: Amennyiben a problémában az egyes munkáknak nincsenek súlyai, akkor a WSPT elv az SPT elvre redukálódik, amely szerint mindig a legkisebb végrehajtási idővel rendelkező munkát ütemezzük először.

Tétel: Az SPT elv optimális ütemezést ad az $1||\sum C_j$ problémára.

Tekintsük a $Pn||\sum C_j$ problémát.

Tegyük fel, hogy m/n egész. Ez elérhető, ha a munkák sorozatát kiegészítjük fiktív 0 végrehajtási idejű munkákkal. Az MSPT (módosított SPT) elv sorbarendezi a munkákat növekvő végrehajtási idő szerint, ezt követően az első n munka lesz az n darab gépen elsőként végrehajtva, a második n munka másodikként, és így tovább, az utolsó n munka m/n-edikként.

Tétel: Az MSPT elv optimális ütemezést ad a $Pn||\sum C_j$ problémára.

Megjegyzés: Ha a munkák paraméterei között határidők is szerepelnek, akkor az EDD („earliest due date") elv azon munkákat ütemezi elsőként, amelyeknek legkorábbi a határideje.

Feladatok

- Milyen ütemezést ad a WSPT elv az $1||\sum w_j C_j$ problémára és mekkora lesz a megoldás célfüggvényértéke az alábbi bemenő adatok esetén? A megoldást Gannt diagramon ábrázoljuk! (A munkákat a (w_j, p_j) értékpárral adtuk meg.)

  $J_1 = (2,4), J_2 = (3,5), J_3 = (4,4), J_4 = (3,2), J_5 = (2,3)$

- Milyen ütemezést ad a MSPT elv a $Pn||\sum C_j$ problémára és mekkora lesz a megoldás célfüggvényértéke az alábbi bemenő adatok esetén, ha $n = 3$? A megoldást Gannt diagramon ábrázoljuk!

  $J_1 = 4, J_2 = 3, J_3 = 5, J_4 = 2, J_5 = 1$

Heurisztikák

Az egyik legegyszerűbb változatot vizsgáljuk. A modellben n darab azonos $M_1, \dots, M_n$ gépünk van, a $J_1, \dots, J_m$ munkák mindegyikéhez egy paraméter tartozik, a p_j $(j = 1, \dots, m)$ végrehajtási idő.

Feladat: Egy olyan ütemezés megkonstruálása, amelyre a befejezési idők maximuma minimális.

A probléma egy gép esetén triviális. Bármely olyan ütemezés optimális, amelyben a gép folyamatosan dolgozik, (nem várunk a munkák elvégzése közben) a $\max\{C_j : 1 \leq j \leq m\}$ érték megegyezik a végrehajtási idők összegével. Továbbá az is nyilvánvaló, hogy amennyiben minden munkát elvégzünk, akkor nem fejezhetjük be a munkákat ezen idő előtt.

Megjegyzés: A probléma $n \geq 2$ esetén NP-nehéz.

Az $n \geq 2$ esetben egy lehetséges ütemezést meghatározhatunk azáltal, hogy az egyes munkákat mely gépekhez rendeljük hozzá.

Amennyiben minden gépre megkapjuk, hogy mi a géphez rendelt munkák halmaza, akkor ezen munkák tetszőleges ütemezése optimális lesz az adott gépen, ha a gép folyamatosan dolgozik.

Minden gépen a maximális befejezési idő, a géphez rendelt munkák végrehajtási idejeinek összege lesz.

Egy adott géphez rendelt munkák végrehajtási idejeinek összegét a gép **töltésének** nevezzük.

A munkák gépekhez rendelését úgy kell tehát elvégeznünk, hogy a maximális géptöltés a lehető legkisebb legyen. A következőkben két heurisztikát ismertetünk a feladat megoldására.

Heurisztikák

- Lista algoritmus (Graham)
  - Előkészítő rész
    - A J_1 munkát rendeljük az M_1 géphez, továbbá legyen $r = 1$.
  - Iterációs rész (r. iteráció)
    - Ha $r = m$, akkor vége az eljárásnak.
    - Ellenkező esetben a J_{r+1}munkát rendeljük ahhoz a géphez, amely gépen minimális a töltés. Ha több ilyen gép is van válasszuk a legkisebb indexűt.
    - Legyen $r = r + 1$, majd folytassuk az eljárást a következő iterációs lépéssel.

Tétel: A lista algoritmus approximációs hányadosa $2 - 1/n$, ahol n a gépek száma.

Megjegyzés: Az algoritmus „gyengéje”: sok rövid munka után jön egy hosszú.

- LPT algoritmus (Graham)

1. Rendezzük sorba a munkákat csökkenő végrehajtási idő szerint.
2. A munkák kapott listáján hajtsuk végre a lista algoritmust.

Tétel: Az LPT algoritmus approximációs hányadosa $4/3 - 1/(3n)$, ahol n a gépek száma.

Megjegyzés: A lista algoritmus akkor is használható, ha nem ismert az ütemezés elkezdésekor az összes munka, hanem a munkákat azonnal, a többi munkára vonatkozó ismeretek nélkül kell ütemeznünk. Az ilyen problémákat **on-line ütemezési problémáknak** nevezzük.

Feladatok

- Milyen ütemezést ad a lista algoritmus és mekkora lesz a megoldás célfüggvényértéke az alábbi bemenő adatok esetén, ha $n = 2$? A megoldást Gannt diagramon ábrázoljuk!

$$J_1 = 4, J_2 = 3, J_3 = 5, J_4 = 2, J_5 = 1, J_6 = 3, J_7 = 2, J_8 = 4$$

- Milyen ütemezést ad az LPT algoritmus a fenti adatokra? Mekkora lesz a megoldás célfüggvényértéke $n = 2$, ill. $n = 3$ esetben?