FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2002. március 22-23.

PERMUTÁCIÓ FLOW SHOP TERMELÉSÜTEMEZÉSI FELADAT MEGOLDÁSI VÁLTOZATAINAK KIÉRTÉKELÉSE

Oláh Béla

1. Abstract

The goal of this scientific work is to describe a quick and efficient algorithm so as to evaluate the solutions of the production-scheduling problem.

In the first part of the paper the production-scheduling problem is defined and the scheduling criteria are showed. In the central chapter of my paper a very simple evaluating algorithm is demonstrated which evaluates the sequences with the help of dynamic programming according to the minimum make-span, maximum machines utilization and the minimum work in process inventories.

2. A termelésütemezési feladat általános megfogalmazása és értékelési kritériumai

Adott n számú termék, melyeken m számú különböző munkafolyamatot kell elvégezni. Meg kell határozni a termékeknek azt a sorrendjét a gépeken, mely bizonyos előre megadott szempontok szerint optimális.

A feladatok elkészített ütemezésének talán legfontosabb elemét az értékelési kritériumok jelentik, amelyek alapján a kialakított alternatív sorrendek értékelhetők. A leggyakrabban használt kritériumok időalapúak. A gyártásütemezés célfüggvényei a következők lehetnek:

- minimális átfutási idő;
- várakozási idők minimalizálása (akár a gépek, akár a feladatok esetében);
- technológiai berendezések maximális kihasználása;
- minimális gyártásközi készletek;
- minimális gyártási idő- és költségfordítás;
- a munkák előírt határidőre való elkészülése.

3. A megoldási változatok kiértékelésére szolgáló algoritmus kidolgozása

Feltehetjük a következő kérdést: mennyi idő alatt lehet egy tetszőleges sorrendet megvalósítani (mennyi lesz az átfutási ideje a termékeknek a gépeken). Az, hogy a munkát t idő alatt be tudjuk

fejezni, azt jelenti, hogy X_1 -től X_n -be t idő alatt el tudok jutni a gráfban úgy, hogy minden tevékenységnek megfelelő élet bejárunk. E szerint az elvégzés minimális idejét az X_1 és X_n közötti maximális értékű út határozza meg, ugyanis az éleinek megfelelő tevékenységeket csak egymás után hajthatjuk végre, s ezzel párhuzamosan más tevékenységeket is végezhetünk.

3.1. Maximális út keresése dinamikus programozással

Határozzuk meg az X_1 és X_n közötti maximális értékű utat. Használhatnánk a legrövidebb utat kereső algoritmusok valamelyikét, de a gráf speciális tulajdonságai lehetővé teszik, hogy a Bellman-Pontrjagin-féle optimalizálási elvet alkalmazzuk.

TÉTEL. Egy, az X_1 csúcsból az X_n csúcsba vezető maximális (minimális) út csak maximális (minimális) részutakat tartalmazhat, azaz ha X_i a részút kezdőpontja és X_j a végpontja ($1 < i < j < n$), akkor a részút X_i és X_j között maximális (minimális). Röviden úgy is mondhatjuk, hogy optimális út csak optimális részutakból állhat. A tétel bizonyítása [2]-ben megtalálható.

Ezen tétel felhasználásával a mátrix reprezentációban egy rendkívül szellemes megoldást kapunk. Ez a módszer végtelenül hatékony, és köszönhetően a csak két egymásba ágyazott for ciklusnak, valamint a kis adatbázis reprezentációnak ($n \times m$) meglepően gyors is, szemben a Dijkstra algoritmussal, amely $O(V^2 + E)$ lépésben találja meg a leghosszabb utat.

Adott egy $(n \times m)$ -es T mátrix, melyben a t_{ij} az i -dik termék műveleti ideje a j -dik gépen. Az időegység a feladattól függően szabadon választható (pl.: óra, perc, nap). A t_{ij} természetesen csak akkor kezdődhet el, ha a $t_{i-1,j}$ (az i -edik termék megmunkálása a $j-1$ -edik gépen) már befejeződött és a $t_{i-1,j}$ (az $i-1$ -edik termék j -edik művelete) is véget ért.

$$\text{Időmátrix: } T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{1m} \\ t_{21} & t_{22} & t_{23} & t_{2m} \\ t_{31} & t_{32} & t_{33} & t_{3m} \\ \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & t_{nm} \end{bmatrix}$$

Tekintsük az időmátrixot egyben optimális sorrendnek. Szeretnénk kiértékelni a következő célfüggvényeket:

- minimális átfutási idő
- technológiai berendezések maximális kihasználása,
- minimális gyártásközi készletek,

Az átfutási idő nem más, mint az első munkadarab gépre kerülésétől az utolsó termék befejezéséig terjedő időintervallum, azaz a megfelelő mátrix első elemétől az utolsóig a maximális út.,

1. ábra: A permutáció flow shop Gantt-diagrammja

A technológiai berendezések maximális kihasználása nem más, mint a gépek holtidejének minimalizálása. Minimális gyártásközi készleteket pedig, a termékek állásidejének minimalizálásával érhetünk el. Ezzel a módszerrel az átfutási idő, a gépek holtideje illetve a termékek állásideje is egyszerűen meghatározható. Miután megvan a kiértékelendő sorrendünk az első feladatunk a maximális útmátrix elkészítése, amit jelöljünk B mátrixszal, ahol b_{ij} jelenti az i -edik termék j -edik gépen történő megmunkálásának befejezéséig eltelt időt. Következésképpen b_{nm} nem lesz más, mint a keresett átfutási idő!

$$\text{Maximális útmátrix: } B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{1m} \\ b_{21} & b_{22} & b_{23} & b_{2m} \\ b_{31} & b_{32} & b_{33} & b_{3m} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & b_{nm} \end{bmatrix}$$

$$b_{ij} = t_{ij} \quad \text{ha } i = 1, j = 1 \quad (1)$$

$$b_{ij} = \sum_{\lambda=1}^j t_{i\lambda} \quad \text{ha } i = 1, j = 2, \dots, m \quad (2)$$

$$b_{ij} = \sum_{\lambda=1}^i t_{\lambda j} \quad \text{ha } j = 1, i = 2, \dots, n \quad (3)$$

$$b_{ij} = t_{ij} + \max(t_{i-1,j}, t_{i,j-1}) \quad \begin{array}{l} \text{a } i = 2, \dots, n \\ \text{és } j = 2, \dots, m \end{array} \quad (4)$$

Ehhez hasonlóan a j -edik gép állásideje úgy számolható, hogy a j -edik gép főidejéből levonva a gépen történő konkrét megmunkálások idejét valamint az első termék a j -edik gépig történő átfutását.

$$h_j = b_{nj} - \sum_{i=1}^n t_{ij} - \sum_{k=1}^{j-1} t_{1k} \quad \text{ha } j = 2, \dots, m \quad (5)$$

Észrevehetjük, hogy a $\sum_{k=1}^{j-1} t_{ik}$ kifejezés $b_{i,j-1}$ -gyel lesz egyenlő, amit már kiszámoltunk a B mátrix

képzésénél. A képlet tehát a következő alakra egyszerűsödik:

$$h_j = b_{nj} - \sum_{i=1}^n t_{ij} - b_{1,j-1} \quad \text{ha } j = 2, \dots, m \quad \text{ami megegyezik} \quad (6)$$

$$h_j = b_{nj} - \sum_{i=2}^n t_{ij} - b_{1j} \quad \text{összefüggéssel, ha } j = 2, \dots, m \quad (7)$$

Az összes gép holtideje a $\sum_{j=2}^m h_j$ képlettel számítható. Sok szakirodalom a gépeknek az első termékre való várakozási idejét is a holtidők közé számítja, ilyenkor a képlet tovább egyszerűsödik:

$$h_j = b_{nj} - \sum_{i=1}^n t_{ij} \quad \text{ha } j = 2, \dots, m \quad (8)$$

Nagy gépszám esetén természetesen ezzel az eljárással, nem lesz valami hatékony a berendezések kihasználása, ugyanis a technológiai sorrend végén található gépek akár órákat is várhatnak, amíg megérkezik az első megmunkálandó termék!

Az i -edik termék állásidejének kiszámítása a j -edik gép holtidejének előállítására szolgáló képlettel (7) teljesen azonos módon:

$$r_i = b_{im} - \sum_{j=2}^m t_{ij} - b_{i1} \quad \text{ha } i = 2, \dots, n \quad (9)$$

Az összes termék állásideje pedig a $\sum_{i=2}^n r_i$ képlettel számítható.

Természetesen, itt is figyelembe lehet venni a termékeknek az első gépre való várakozási idejét:

$$r_i = b_{im} - \sum_{j=1}^m t_{ij} \quad \text{ha } i = 2, \dots, n \quad (10)$$

4. Irodalomjegyzék

- [1] Tóth T.: Tervezési elvek, modellek és módszerek a számítógéppel integrált gyártásban, Miskolci Egyetemi Kiadó, 1998. pp. 132-144.
- [2] T. Bányai: Auftragsreihenfolge-Optimierung mit genetischen Algorithmus, Proceedings of the Conversations at Miskolc International Conference., Miskolc, 2001, pp.111-117.
- [3] T. Bányai: Mobilrobotos egységakománypépző és osztályozó rendszerek tervezési módszerei, Ph.D. értekezés, Miskolc, 1999, p.114
- [4] T. Bányai, J. Cselényi: Planung von Mobilrobotssystemen, Publications of the University of Miskolc, Series C: Mechanical Engineering, Volume 49: Production systems, Miskolc, 1999, pp.: 19-28, Társszerző: Dr. Cselényi József

Oláh Béla egyetemi hallgató

Miskolci Egyetem Anyagmozgatási és Logisztikai Tanszék

Miskolc-Egyetemváros H-3515

E-mail: qgebulls@gold.uni-miskolc.hu