XV. FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2010. március 25-26.

KERETRENDSZER FEJLESZTÉSE ÜTEMEZÉSI FELADATOK MEGOLDÁSÁRA

MIHÁLY Krisztián

Abstract

This paper describes a short overview of the shop based scheduling models and define a general framework concept to support different scheduling algorithms.

Key words:

flow-shop, job-shop, scheduling model, framework

Összefoglaló

Jelen dolgozat rövid áttekintést ad a „shop” alapú ütemezési modellekről, illetve előrevetíti egy ütemezési algoritmusokat segítő, általános keretrendszer koncepcióját.

Kulcsszavak:

flow-shop, job-shop, ütemezési modellek, keretrendszer

Bevezetés

Ütemezési problémák az élet minden területén megtalálhatók, jelen vannak a gyártásban, szállításban és logisztikában egyaránt. Még olyan kevésbé egyértelmű területeken is felmerülhetnek, mint a kommunikáció, médiairányítás, sport, stb. Optimális ütemezésre szükség lehet egy kerékpár alkatrészeinek gyári összeszerelésénél, egy sportesemény megszervezésénél, vagy akár egy iskola tanterembeosztásának megtervezése során is.

Általánosságban véve az ütemezés célja, hogy különböző feladatok (tevékenységek, elemi lépések, munkák) optimális feldolgozási sorrendjét a feldolgozó egységek (gépek) halmazán meghatározzuk, figyelembe véve számos olyan külső kényszert, melyek kihatnak a feladatokra, a gépekre, és azok kölcsönös viszonyára.

A szakirodalomban számos ütemezési modell található. Többek között létezik a shop elrendezésű modell, egy - illetve sok gépes modellek, valós idejű modell, ciklikus ütemezési modell. Jelen kutatás célja egy olyan általános keretrendszer létrehozása, amely elrejti az alacsony szintű technológiai feladatokat, elősegíti a modell létrehozását és támogatja az ütemezési eredmények megjelenítését. A keretrendszer első verziója a shop jellegű ütemezési feladatokat támogatja és az alapvető ütemezési algoritmusok végrehajtására képes.

Általános shop ütemezési probléma

Általában az ütemtervezési problémák három halmazzal írják le: adott n db feladat halmaza $T = \{T_1, T_2, \dots, T_n\}$, m db feldolgozóegység (gép) halmaza $P = \{P_1, P_2, \dots, P_m\}$ és s féle további készlet halmaza $R = \{R_1, R_2, \dots, R_s\}$. Az ütemezés nem más, mint a T feladathoz leginkább alkalmazható P feldolgozóegység és R készlet meghatározása azzal a céllal, hogy az összes elvégzendő feladat teljesíthető legyen a külső kényszerek kielégítése mellett. A klasszikus ütemtervezési elmélet két általános kényszert határoz meg. Minden feladat egyszerre legfeljebb egy géppel végezhető el és minden gép egyszerre legfeljebb egy feladat elvégzésére képes.[1]

Az általános shop ütemtervezési probléma fontos speciális esetei a job-shop, flow-shop és open-shop problémák. Ezen modellek pontos leírásához a szakirodalomban elfogadott módszer, hogy a feladatok halmazát n db diszjunkt részhalmazra osztják, melyeket munkának (job) neveznek. J_j munka n_j feladatra oszlik $T_{1j}, T_{2j}, \dots, T_{n_{jj}}$, és két szomszédos feladatot különböző gépek végeznek el. A továbbiakban a munkák halmazát I -vel jelöljük.

Job-shop

A job-shop ütemezési probléma n_j db munkából áll, melynek végrehajtásához m db gép áll rendelkezésre. Az n_j szám tetszőleges. A feladatok között létezik egy elsőbbségi rendezés, melynek formája: $T_{1j} \rightarrow T_{2j} \rightarrow \dots \rightarrow T_{n_{jj}}$, minden $j = 1, 2, \dots, n$ -re. Ez azt jelenti, hogy a különböző munkák feladatai között nincs előzés, illetve ugyanazon munka feladatai az ugyanazon munkán belül egy feldolgozási láncot alkotnak. A T_{ij} feladatot $p_{ij} > 0$ időegység alatt kell elvégezni, előzés nélkül, az adott feladatra alkalmas géppel.

Flow-shop

A flow-shop egy speciális job-shop probléma, ahol a feladatok száma $n_j = m$ és $j = 1, \dots, n$ és $\mu_{ij} = M_i$, minden $i = 1, \dots, m, j = 1, \dots, n$. A T_{ij} feladatot M_i gépen el kell elvégezni.

A permutációs flow-shop probléma során a munkákat mindegyik gépen ugyanabban a sorrendben kell elvégezni. Az 1. ábrán a flow-shop ütemezésre láthatunk egy példát. A O_{ij} egy olyan műveletet jelöl, melyet az i géppel kell elvégezni. A második index a azt a munkát azonosítja, amelyhez a művelet tartozik.

M_1	O_{11}	O_{12}	O_{13}	O_{14}					
M_2		O_{21}	O_{22}	O_{23}	O_{24}				
M_3			O_{31}		O_{32}	O_{33}	O_{34}		

1. ábra Operációs folyamat flow-shop modellben

Open-shop

Az open-shop probléma annyiban különbözik a flow-shop-tól, hogy műveletek között nincs elsőbbségi

kapcsolat. Formailag a feladatok száma mindegyik munka esetén ugyanynyi és m -mel egyenlő, $n_j=m$, $j = 1,2,\dots,n$. Valamint a T_{1j} -t a P_1 -n kell elvégezni, a T_{2j} -t a P_2 -n, és így tovább.

2. ábra Operációs folyamat open-shop modellben

Ütemezési technológiák

Egy tipikus ütemezési probléma számos egyidejű és gyakran egymásnak többszörösen ellentmondó célokat tartalmaz. A lehetséges ütemezések száma egy kombinatorikus jelleg miatt egy óriási keresési teret hoz létre, mely gyorsan nő a munkák, gépek, stb. számának függvényében. Egy m gép és n munka problémának akár $(m!)^n$ megoldása is létezhet, mely jelenséget kombinatorikus robbanásnak neveznek. Továbbá a követelmények közt is találhatunk ellentmondásokat: határidők, szükséges gépek száma, költségek, stb., melyek együttes kielégítése nem lehetséges. Néhány speciális esetben a létezik a probléma megoldására alkalmazható analitikus módszer, mint például a Johnson algoritmus. Legtöbbször azonban heurisztikus algoritmusok alkalmazása szükséges. Az algoritmusokat bonyolultságuk alapján minden probléma P vagy NP problémakörbe osztályozhatjuk. A P osztályba azon problémák tartoznak, melyek megvalósítási költsége a méretük polinomiális függvénye. Az NP problémák esetében ez a költség a mérettel exponenciálisan változik.[3]

Az ütemezési problémák elméleti leírása

Ronald Graham, Eugene Lawler, Jan Karel Lenstra & Alexander Rinnooy Kan egy egyszerűen alkalmazható leírási módszert vezetett be az elméleti ütemezési problémák megoldására [4][5]. A leírás három komponensből áll: α , β és γ . Mindegyik komponenshez egy vesszővel tagolt felsorolás tartozik. Az α rész a feldolgozási környezetet, a β a munka jellemzőit, míg a γ a célkritériumot írja le. Jelen dolgozat keretein belül a mélyebb ismertetésre nincs lehetőség.

Az ütemező keretrendszer architektúrális koncepciója

A keretrendszer objektum orientált programozási paradigmát követ. Az alapobjektumokat osztályok, illetve interfészek írják le. A magas szintű osztálydiagram a 3. ábrán látható. Az ábrán látható „Framework” osztály írja le az általános keretrendszert. Ez az osztály tárolja a létrehozott modellre mutató referenciákat, eléri az ütemező osztályt és képes megjeleníteni az ütemezési eredményeket. Egy ütemezési modell különböző gépeket és feladatokat tárol. A feladatokat a megrendelésekből származtatjuk. Maga a megrendelés származhat egy csatlakoztatott ERP, vagy termelésütemezési modulból. Ebben a modellben az ütemezőt egy interfész reprezentálja. A későbbiekben az ütemező algorit-

must megvalósító osztálynak ezt az interfészt kell implementálnia ahhoz, hogy a keretrendszerrel kommunikálni tudjon. A munkákon értelmezett kényszerek és az ütemezést kiértékelő célfüggvények az ábrán nincsenek feltüntetve.

Az ütemezési eredmények megjelenítéséhez jelen verzióban szöveges, táblázatos formát használunk. Továbbfejlesztési lépésként a Siemens Plant Simulation termékét integráljuk az elkészült keretrendszerhez.

3. ábra A keretrendszer magas szintű osztálydiagrammja

Irodalomjegyzék

- [1] J. BLAZEVICZ, K.H. ECKER, E. PESCH, G. SCHMIDT, J.WEGLARZ, *Scheduling Computer and Manufacturing Processes*, Springer
- [2] P. BRUCKER, S. KNUST, *Complex Scheduling*, Springer
- [3] Tapan P. BAGCHI, *Multiobjective scheduling by genetic algorithms*.
- [4] B. CHEN, C.N. POTTS and G.J. WOEGINGER, *A review of machine scheduling: Complexity, algorithms and approximability*. Handbook of Combinatorial Optimization (Volume 3) (Editors: D.-Z. Du and P. Pardalos), 1998, Kluwer Academic Publishers. 21-169. ISBN 0-7923-5285-8 (HB) 0-7923-5019-7
- [5] P. BRUCKER, S. KNUST, *Complexity results for scheduling problems*.

Mihály Krisztián, okleveles mérnök-informatikus, levelező doktorandusz
 Munkahely: SAP Labs Hungary Kft, illetve Miskolci Egyetem, Alkalmazott Informatika Tanszék
 Levelezési cím: Magyarország, 3645 Mályinka, Lillafüredi út 18.
 Tel (mobil): +36 70 702 48 47
 E-mail: mihalykrisztian@gmail.com