

RENDEZÉS-KONGRUENCIÁK ALKALMAZÁSA TÖBBGÉPES PÁRHUZAMOS ÜTEMEZÉSEK ESETÉN

AN APPLICATION OF ORDER CONGRUENCES IN PARALLEL MULTI-MACHINES SCHEDULING PROBLEMS

*Szilágyi Szilvia**

ABSTRACT

Scheduling, sorting and searching problems are among the most common instances of order. Typically an order must be transformed to another, say a partial extension or a linear extension, which itself may represent a schedule or a sort. In a scheduling problem we have to find an optimal schedule of jobs. We consider the parallel machines case, where m machines are given and we can use them to schedule the jobs. Using the greedy algorithm by Trotter (for finding linear extensions of a poset) we exhibit and implement an algorithm for determining minimal linear order congruences, solving the given parallel scheduling problem.

1. BEVEZETÉS

Ha adott elvégzendő munkák egy M halmaza, akkor ennek egy ütemezésén az M halmaz elemeinek olyan permutációját értjük, amely megadja, hogy az egyes munkákat milyen sorrendben kell végrehajtani. Általános szabály, hogy minden, a munkák elvégzéséhez felhasznált erőforrás (gép) egy időben legfeljebb egy munkán dolgozhat és minden munkát egy időben legfeljebb egy erőforráson (gépen) végezhetünk. A termelési illetve gyártási folyamatok egy ismert modelljét úgy kapjuk, hogy egy részben-rendezett halmaz segítségével ábrázoljuk a gyártási folyamat egyes fázisai közötti összefüggéseket. Az ütemezési feladat klasszikus megoldása ekkor egy olyan lineáris rendezés, amely az eredeti részben-rendezés kiterjesztése. Ha egy termék gyártása során számos technológiai fázis van, akkor a termék általában igen különböző technológiai útvonalakat járhat be. Ha az üzemben az egyes fázisokon párhuzamos berendezések is vannak, akkor a termék által bejárható útvonalak száma meghatározható. Egy lehetséges megoldás, ha a gépeket úgy csoportosítjuk (térbeli és/vagy logikai dekompozíció), hogy azok a termék előállításához szükséges műveletek egy csoportját képesek legyenek

elvégezni. Ütemezési feladatok megoldásakor tehát lineáris rendezéseket keresünk. Szpilrajn klasszikus tétele értelmében bármely részben-rendezés kiterjeszhető lineárisra [10]. Vannak azonban a gyakorlatban olyan esetek, amikor nem szükséges az elemek között teljes lineáris rendezést megadni, hanem érdemesebb nagyobb egységeket (csoportokat, blokkokat, tömböket, osztályokat) képezni. Ilyenek például bizonyos csoporttechnológiai feladatok. Nyilvánvalóan ekkor olyan osztályozás érdekel bennünket, ahol az osztályok közötti részben-rendezés lineáris. Az osztályok elkészítésekor bizonyos esetekben arra is figyelniük kell, hogy azok arányos méretűek legyenek. Adott tehát egy véges elemszámú halmaz, amelyet diszjunkt részhalmazokra (osztályokra) kell felosztani úgy, hogy az osztályok között létrejövő részben-rendezés lineáris legyen, az osztályokon belül pedig egy ekvivalenciareláció érvényesüljön. Lényegében tehát egy véges elemszámú részben-rendezett halmazt osztályozunk úgy, hogy az eredeti részben-rendezés a részhalmazok között egy újabb részben-rendezést indukáljon. Az ilyen felbontások alapján vezettük be a rendezés-kongruencia fogalmát és vizsgáltuk tulajdonságait az [7] dolgozatban.

2. EGY ÜTEMEZÉSI FELADAT

A termelési illetve gyártási folyamatok esetén az elvégzendő munkák egy véges $M = \{m_1, m_2, \dots, m_n\}$ halmazán az $m_i \leq m_j$ jelölést vezetjük be, ha az i -edik munka elvégzése időben megelőzi a j -edik munkát ($i, j \in \{1, 2, \dots, n\}$). Nyilvánvaló, hogy ekkor (M, \leq) részben-rendezett halmaz. Ha M ütemezését klasszikus értelemben keressük, akkor az nem más, mint \leq egy lineáris kiterjesztése. Több megoldást is kaphatunk, hiszen egy részben-rendezésnek több lineáris kiterjesztése is lehet. Tegyük fel, hogy egymással párhuzamosan több munka elvégzésére is lehetőségünk van. Ez azt jelenti, hogy M elemeit célszerű diszjunkt részhalmazokra felosztanunk. Alkalmazzuk a továbbiakban az $\{M_1, M_2, \dots, M_t\}, (t \leq n)$ jelölést egy ilyen osztályozásra. Ekkor $M = M_1 \cup M_2 \cup \dots \cup M_t$. Lényegében lehetséges gyártási fázisokat hozunk létre,

* egyetemi adjunktus, Miskolci Egyetem, Analízis Tanszék

amely jelenthet időbeli dekompozíciót vagy logikai dekompozíciót. A részhalmazok kialakításakor nemcsak azt tartjuk szem előtt, hogy az egyes részhalmazokba egymással szabadon felcserélhető elemek kerüljenek, hanem azt is, hogy a létrehozott tömbök egymással összehasonlíthatóak legyenek, amely lényegét tekintve egy olyan részben-rendezés, amely az eredeti \leq részben-rendezés által indukált lineáris rendezések egyike. Az eddig megtett lépésekhez könnyen találunk matematikai sémát. Mivel $\{M_1, M_2, \dots, M_t\}, (t \leq n)$ osztályozása M -nek, így tartozik hozzá egy ρ ekvivalenciareláció. Ennek megfelelően

$$M/\rho = \{M_1, M_2, \dots, M_t\}, \quad (1)$$

azaz a vizsgált partíció a ρ ekvivalenciareláció faktorhalmaza. Tekintsük továbbá azt a

$$\varphi: M \rightarrow M/\rho \quad (2)$$

függvényt, amely az m_i ($i \in \{1, 2, \dots, n\}$) munkához hozzárendeli azt az M_j ($j \in \{1, 2, \dots, t\}$) fázist, amelyben a munka elvégzésre kerül, azaz

$$\varphi(m_i) := M_j, \quad (3)$$

ha $m_i \in M_j$. A rendezés-kongruenciának a [7] dolgozatban megadott fogalmát esetünkben az alábbi megfogalmazásban célszerű használni.

2.1. Definíció: A ρ ekvivalencia relációt rendezés-kongruenciának nevezzük az M halmazon, ha az M/ρ faktorhalmazon értelmezhető olyan \leq_ρ részben-rendezés, amelyre nézve a φ függvény rendezésőrző, azaz bármely $m_i \leq m_j$ ($m_i, m_j \in M$) esetén

$$\varphi(m_i) \leq \varphi(m_j) \quad (4)$$

teljesül.

A fenti definícióban alkalmazott \leq_ρ jelölésre az indukált részben-rendezés elnevezést használjuk. A 2.1. Definícióból azonnal adódik, hogy ha ρ rendezés-kongruenciája az (M, \leq) részben-rendezett halmaznak, akkor ρ egybeesik a φ izoton függvény magjával. Nyilvánvaló az is, hogy (4) akkor és csak akkor teljesül, ha van olyan $m_i, m_j \in M$, amelyre $m_i \leq m_j$ ($i, j \in \{1, 2, \dots, n\}$).

Az (M, \leq) részben-rendezett halmaz rendezés-kongruenciáinak halmazát jelölje $\vartheta(M)$. Az $(\vartheta(M), \subseteq)$ háló relatív komplementumos, atomisztikus és duálisan atomisztikus, továbbá teljesíti a Jordan-Hölder láncfeltételt [7]. Az ütemezési feladat megoldásához olyan $\rho \in \vartheta(M)$ kongruenciára van szükségünk, amelyre a kialakított részhalmazok között létrejövő \leq_ρ indukált részben-rendezés lineáris, tehát az $(M/\rho, \leq_\rho)$ faktorhalmaz lánc.

2.2. Definíció: Egy $\rho \in \vartheta(M)$ rendezés-kongruenciát az (M, \leq) lineáris rendezés-kongruenciájának nevezzük, ha az $(M/\rho, \leq_\rho)$ faktorhalmaz lánc.

Az ütemezési feladat megoldása tehát az (M, \leq) részben-rendezett halmaz egy lineáris rendezés-kongruenciája lesz. Meg kell azonban jegyeznünk, hogy a megoldást adó lineáris rendezés-kongruenciának még egy további tulajdonsággal is rendelkeznie kell. Ugyanis az (M, \leq) részben-rendezett halmaz lineáris rendezés-kongruenciái közül csak olyanra van szükségünk, amely tovább már nem finomítható $\{M_1, M_2, \dots, M_t\}, (t \leq n)$ partíciót eredményez.

2.3. Definíció: Egy $\rho \in \vartheta(M)$ rendezés-kongruenciát az (M, \leq) minimális lineáris rendezés-kongruenciájának nevezzük, ha (M, \leq) -nek nincs olyan $\theta \neq \rho$ lineáris rendezés-kongruenciája, amelyre $\theta \subseteq \rho$.

Az ütemezési feladat megoldásához célunk az (M, \leq) részben-rendezett halmaz minimális lineáris rendezés-kongruenciáinak meghatározása. A [7] dolgozatban igazolásra került, hogy ha a ρ ekvivalencia-reláció az (M, R) részben-rendezett halmaz intervallum-kongruenciája, akkor ρ rendezés-kongruenciája is (M, R) -nek. Ha R a \leq részben-rendezés lineáris kiterjesztése, akkor könnyen igazolható, hogy a ρ reláció (M, \leq) -nek rendezés-kongruenciája, azaz (M, \leq) rendezés-kongruenciáinak keresése visszavezethető az eredeti \leq részben-rendezés valamely lineáris kiterjesztésének intervallumokra történő feldarabolására. Az ütemezési feladat klasszikus megoldásához tehát az első lépést az (M, \leq) részben-rendezett halmaz esetén a \leq reláció lineáris kiterjesztéseinek meghatározása jelentheti. Erre az irodalomban több kész megoldás is olvasható. Ezek egyike Trotter-től származik [11].

3. A TROTTER-FÉLE MOHÓ ALGORITMUS

Megoldásunk első lépésével kapcsolatos legfőbb probléma az, hogy egy részben-rendezett halmaz lineáris kiterjesztéseinek a száma az alaphalmaz elemszámának függvényében exponenciálisan nő. Ennek a problémának a leküzdésére használjuk a Trotter-féle mohó algoritmust [11].

Jelöljük n -nel az X halmaz elemszámát. A mohó algoritmus az (X, \leq) részben-rendezett halmaz esetén a \leq reláció lineáris kiterjesztését úgy állítja elő, hogy először X minimális elemei közül választ egyet. Ha a lánc első x_1, \dots, x_i tagja már megvan, akkor az $(i + 1)$ -edik elemet az

$$(X \setminus \{x_1, \dots, x_i\}, \leq) \quad (5)$$

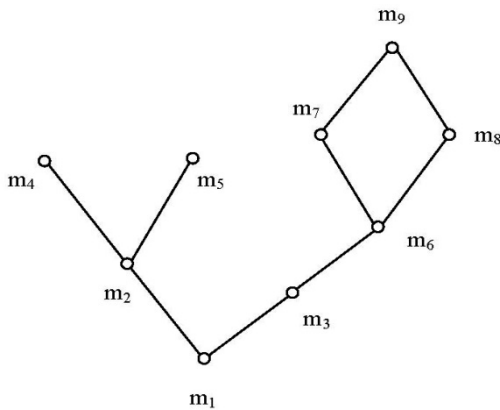
részben-rendezett halmaz minimális elemeinek halmazából választja a mohó feltétel szerint. Ez azt jelenti, hogy $i > 0$ esetén nem tetszőlegesen választ a minimális elemek közül, hanem azokra a minimális elemekre szűkít, amelyek az előző lépésben kiválasztott elemmel összehasonlíthatóak, amennyiben van ilyen elem.

3.1. Algoritmus:

1. **Proc** Trotter
2. **Input:** X halmaz
3. P halmaz (részben-rendezés)
4. **Output:** L vektor (lánc)
5. $X_0 \leftarrow X$
6. $P_0 \leftarrow P$
7. $\mathbf{P}_0 \leftarrow (X_0, P_0)$
8. $G_0 \leftarrow (S_0 \leftarrow \min(\mathbf{P}_0)$ tetszőleges eleme)
9. **for** $i \leftarrow 0$ **to** $n - 1$ **do**
10. **if** $i > 0$ **then**
 $S'_i = \{x \in S_i \mid x_i < x \text{ } P \text{ -ben}\}$
11. **if** $S'_i = \emptyset$ **then** $G_i = S_i$
12. **if** $S'_i \neq \emptyset$ **then** $G_i = S'_i$
13. $x_{i+1} \leftarrow G_i$ egy tetszőleges eleme
14. $L[i + 1] \leftarrow x_{i+1}$
15. $X_{i+1} \leftarrow X_i \setminus \{x_{i+1}\}$
16. $P_{i+1} \leftarrow P(X_{i+1})$
17. $\mathbf{P}_{i+1} \leftarrow (X_{i+1}, P_{i+1})$
18. $S_{i+1} \leftarrow \min(\mathbf{P}_{i+1})$
19. **RETURN**

Az algoritmus működésének szemléltetésére tekintünk az alábbi példát.

3.2. Példa: Az (M, \leq) részben-rendezett halmazt Hasse-diagramjával szemléltetjük.



1. ábra

Ekkor például $L_1: [m_1, m_3, m_6, m_7, m_2, m_4, m_5, m_8, m_9]$ és $L_2: [m_1, m_2, m_4, m_5, m_3, m_6, m_7, m_8, m_9]$ olyan lineáris kiterjesztései \leq -nek, amelyeket a 3.1. Algoritmus eredményezhet.

4. AZ ÜTEMEZÉSI FELADAT MEGOLDÁSA

Jól látható, hogy klasszikus értelemben, azaz amikor egy időben csak egy gép dolgozik (tehát nincs lehetőség párhuzamos munkák végzésére) az ütemezési feladatnak a Trotter-féle mohó algoritmus eredményeként kapott lineáris kiterjesztés minden esetben egy megoldását adja. A minimális lineáris rendezés-kongruenciák

meghatározásával azt az alternatív ütemezési feladatot fogjuk megoldani, amikor megengedjük, hogy egy időben több gép is dolgozzon. Lényegében logikai dekompozíciót végzünk, amelynek eredményeként azt várjuk, hogy a gyártási (futási) idő csökkenjen. Az ütemezési feladatot reprezentáló (M, \leq) részben-rendezett halmaz esetén \leq lineáris kiterjesztését a Trotter-féle mohó algoritmus szolgáltatja. Ha tekintjük az $L: x_1, \dots, x_n$ ($x_i = m_j$, $i, j \in \{1, 2, \dots, n\}$) láncot (ahol L a \leq lineáris kiterjesztése), akkor elő kell állítanunk az $M_1 = [x_1, x_{i^1}]$, $M_2 = [x_{i^1+1}, x_{i^2}]$, \dots , $M_t = [x_{i^{t-1}}, x_n]$ intervallumokat. Az így kialakított partició az M halmaz rendezés-kongruenciája. Lineáris rendezés-kongruenciát a 2.2. Definíció értelmében akkor kapunk, ha az

$$\{M_1, M_2, \dots, M_t\}, \quad (t \leq n) \quad (6)$$

halmazon indukált részben-rendezés lineáris. Ez akkor következik be, ha bármely két egymást követő intervallum esetén léteznek olyan $x_k \in M_j$ és $x_l \in M_{j+1}$ ($1 \leq k < l \leq n$) elemek, amelyekre $x_k \leq x_l$ teljesül. Ahhoz, hogy minimális lineáris rendezés-kongruenciát állítsunk elő elégséges, ha a kialakított M_1, M_2, \dots, M_t intervallumok (M, \leq) -ben antiláncok és bármely két szomszédos M_j, M_{j+1} ($j \in \{1, \dots, t-1\}$) intervallum esetén legyenek olyan $x_k \in M_j$ és $x_l \in M_{j+1}$ elemek ($1 \leq k < l \leq n$), amelyekre $x_k < x_l$, azaz x_k rákövetkezője az x_l elem, tehát nincs olyan $x_s \in M$ elem, amelyre $x_k < x_s < x_l$ teljesül. Ha a Trotter-féle mohó algoritmust kiegészítjük a fenti észrevételeinkkel, akkor az ütemezési feladat megoldását minimális lineáris rendezés-kongruencia formájában kapjuk meg. Jelöljük n -nel az L -vektor hosszát.

4.1. Algoritmus:

1. **Proc** Minimális lineáris rendezés-kongruencia
2. **Input:** X halmaz
3. P halmaz (részben-rendezés)
4. L vektor (Trotter-féle algoritmusból)
5. **Output:** \mathcal{A} (minimális lineáris kongruencia)
6. $X_1 \leftarrow X$
7. $P_1 \leftarrow P$
8. $\mathbf{P}_1 \leftarrow (X_1, P_1)$
9. $A_1 \leftarrow \{L[1]\}$
10. $\mathcal{A} \leftarrow A_1$
11. **for** $i \leftarrow 1$ **to** $n - 1$ **do**
12. **if** $(L[i], L[i + 1]) \in \mathbf{P}_i$
13. **then** $A_{i+1} \leftarrow \{L[i + 1]\}$
14. **if** $(L[i], L[i + 1]) \notin \mathbf{P}_i$
15. **then** $A_{i+1} \leftarrow A_i \cup \{L[i + 1]\}$
16. $\mathcal{A} \leftarrow \mathcal{A} \cup \{A_{i+1}\}$
17. **if** $A_i \subseteq A_{i+1}$ **and** $A_i \neq A_{i+1}$
18. **then** az \mathcal{A} halmazból $\{A_i\}$ törlése
19. $X_{i+1} \leftarrow X_i \setminus \{L[i]\}$
20. $P_{i+1} \leftarrow P(X_{i+1})$
21. $\mathbf{P}_{i+1} \leftarrow (X_{i+1}, P_{i+1})$
22. **RETURN**

A 4.1. Algoritmus az (X, \leq) részben-rendezett halmaz esetén egy ekvivalencia osztályaival megadott minimális lineáris rendezés-kongruenciát úgy állít elő, hogy a bemenetként kapott L lánc első eleméből kialakít egy osztályt. Most $L[i]$ jelöli az L lánc i -edik elemét. Ha $i \geq 1$, akkor az algoritmus megvizsgálja, hogy a lánc i -edik és $i + 1$ -edik eleme között fennáll-e a \leq reláció. Ha $L[i] \leq L[i + 1]$, akkor új osztályt hoz létre, amely az $L[i + 1]$ elemet tartalmazza. Ha $L[i] \not\leq L[i + 1]$, akkor az új osztályt úgy alakítja ki, hogy az előző lépésben létrehozott osztályt bővíti az $L[i + 1]$ elemmel. Ha az előző lépésben létrehozott osztály valódi részhalmaza az i -edik lépésben megalkotott halmaznak, akkor azt töröljük, hiszen annak elemeit az új osztály tartalmazza. Könnyen látható, hogy az egy osztályba kerülő elemek antiláncot képeznek az (X, \leq) részben-rendezett halmazban, továbbá bármely két egymás után kialakított (és nem törölt) osztály (intervallum) esetén találunk olyan elemeket, amelyekre a megkövetelt rákövetkezési tulajdonság teljesül.

Sikerült tehát a kitűzött ütemezési feladatunkhoz olyan megoldást találni, amelynek segítségével az egyes elvégzendő munkák olyan logikai dekompozíciója végezhető el, amely csökkenti a teljes munka elvégzéséhez szükséges időt, azáltal, hogy lehetővé válik bizonyos munkafázisok során a párhuzamos munkavégzés.

4.2. Példa: Ha a 3.2. Példában megadott L_1 lineáris kiterjesztést tekintjük, akkor a 4.1. Algoritmus az alábbi minimális lineáris rendezés-kongruenciát eredményezi:

$$\{\{m_1\}, \{m_3\}, \{m_6\}, \{m_7, m_2\}, \{m_4, m_5, m_8\}, \{m_9\}\} \quad (7)$$

Az L_2 lánc esetén pedig a

$$\{\{m_1\}, \{m_2\}, \{m_4, m_5, m_3\}, \{m_6\}, \{m_7, m_8\}, \{m_9\}\} \quad (8)$$

megoldást kapjuk. Látható, hogy az elvégzendő munkát hat fázisra bontottuk mindkét esetben és legfeljebb három gép dolgozhat párhuzamosan. Az is jól látható, hogy ebben az esetben a minimális lineáris rendezés-kongruencia alapján elvégzett logikai dekompozíció kilenc időegységről hat időegységre csökkenti az előállítási (futási) időt, amennyiben feltételezzük, hogy minden elvégzendő munka ugyanakkora időigénnyel bír.

Az is könnyen észrevehető, hogy ha a 4.1. Algoritmus bemenetként megadott láncot nem a Trotter-féle mohó algoritmus eredményezi, akkor a minimális lineáris rendezés-kongruenciára megadott feltételek általában nem teljesülnek a kialakított ekvivalencia-osztályokra, tehát nem szolgáltatnak optimális megoldást az ütemezési feladatra.

4.3. Példa: Tekintsük a 3.2. Példában az (M, \leq) részben-rendezett halmaz esetén az

$$L_3: [m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9] \quad (9)$$

lineáris kiterjesztést. Könnyen látható, hogy ezt a láncot a 3.1. Algoritmus nem állítja elő. Ekkor ugyanis $x_1 = m_1$ és $x_2 = m_2$, így $S_2 = \{m_4, m_5, m_3\}$ és $G_2 = \{m_4, m_5\}$. Emiatt x_3 -nak a mohó algoritmus nem választhatja m_3 -at, mert $m_3 \notin G_2$. Ha az L_3 lineáris kiterjesztés a 4.1. Algoritmus bemenete, akkor az alábbi osztályozást kapjuk:

$$\{\{m_1\}, \{m_2, m_3, m_4, m_5, m_6\}, \{m_7, m_8\}, \{m_9\}\}. \quad (10)$$

Ekkor azonban az $\{m_2, m_3, m_4, m_5, m_6\}$ osztály elemei nem képeznek antiláncot, mert

$$m_2 \leq m_4, \quad m_2 \leq m_5, \quad m_3 \leq m_6. \quad (11)$$

Az ütemezési feladatnak az így elkészített osztályozás nyilvánvalóan nem megoldása.

KÖSZÖNETNYÍLVÁNÍTÁS

A bemutatott kutató munka a TÁMOP-4.2.1.B-10/2/KONV-2010-0001 jelű projekt részeként az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

IRODALOM

- [1] CHAJDA I., SNÁŠEL V. (1998) *Congruences in ordered sets*, Czech Math. Journal **123**, 95-100.
- [2] DAEVY B. A., PRIESTLEY H. A. (1990) *Introduction to Lattices and Order*, Cambridge University Press, Cambridge.
- [3] FISHBURN P.C. (1985) *Interval Orders and Interval Graphs*, Willey, New York.
- [4] FOLDES S., RADELECZKI S. (2004) *On interval decomposition lattices*, *Discussiones Mathematicae, General Algebra and Applications* **24**, 95-114.
- [5] HALAŠ R. (1995) *Congruences on pos ets*, *Contributions to General Algebra* **12**, Velad Johannes Heyn, Klagenfurt, 195-210.
- [6] KOLIBIAR M. (1987) *Congruence relations and direct decompositions of ordered sets*, *Acta Sci. Math. (Szeged)* **51**, 129-135.
- [7] KÖRTESEI P., RADELECZKI S., SZILAGYI Sz. (2005) *Congruences and isotone maps on partially ordered sets*, *Mathematica Pannonica* **16/1**, 39-55.
- [8] ORE O. (1943) *Chains in partially ordered sets*, *Bull. Amer. Math. Soc.* **49**, 558-566.
- [9] PINEDO M. (2002) *Scheduling (Theory, algorithms and systems)*, Prentice Hall.
- [10] SZPILRAJN E. (1930) *Sur l'extension de l'ordre partiel*, *Fund. Math.* **16**, 386-389.
- [11] TROTTER W. T. (1992) *Combinatorics and Partially ordered sets, Dimension Theory*, The Johns Hopkins University Press, Baltimore-London.