Multidiszciplináris tudományok, 9. kötet (2019) 1. sz. pp. 147–155 **https://doi.org/10.35925/j.multi.2019.1.18**

MEGELŐZÉSI FELTÉTELEKET TARTALMAZÓ ÜTEMEZÉSI FELADATOK PÁRHUZAMOSÍTÁSA MOHÓ ALGORITMUSOK ÁLTAL GENERÁLT RENDEZÉS-KONGRUENCIÁK SEGÍTSÉGÉVEL

Körei Attila

Egyetemi docens, Miskolci Egyetem, Matematikai Intézet, Alkalmazott Matematikai Intézeti Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: matka@uni-miskolc.hu

Szilágyi Szilvia

Egyetemi docens, Miskolci Egyetem, Matematikai Intézet, Analízis Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: matszisz@uni-miskolc.hu

Absztrakt
Az ütemezési problémák egy alternatív megközelítését jelentik azok az eljárások, amelyek a diszkrét matematika eszközeivel dolgoznak. A részbenrendezett halmazok rendezés-kongruenciáinak tanulmányozásakor világossá vált, hogy a minimális lineáris rendezés-kongruenciák jól alkalmazhatók ütemezési feladatok megoldására. Ez a metódus nemcsak akkor használható, ha egy gép dolgozik, hanem kiterjeszthető olyan esetekre is, amikor egymással párhuzamosan több egységnyi kapacitású gép dolgozhat. Cikkünkben a megelőzési feltételeket tartalmazó ütemezési feladatokra olyan optimális vagy közel optimális megoldást adó algoritmusokat mutatunk be, amelyek a megoldást minimális lineáris kongruencia formájában állítják elő.

Kulcsszavak: *részbenrendezett halmaz, lineáris kiterjesztés, párhuzamos ütemezés, rendezés-kongruencia*

Abstract
Scheduling problems can be alternately approached by methods based on discrete mathematical tools. Studying the order congruences of partially ordered sets it became clear that the minimal linear order congruences can be successfully applied in solving scheduling problems. This technique can be used in the single machine case and it can be extended to multimachine environment where machines work parallelly. In this paper we discuss scheduling problems given by precedence constraints and show algorithms yielding optimal or quasi optimal solutions in the form of minimal linear congruences.

Keywords: *poset, topological sorting, parallel scheduling, order congruency*

1. Bevezetés

Az ütemezési problémákban adott, elvégzendő tevékenységek egy olyan sorrendjét kell meghatároznunk, amely valamilyen szempont alapján optimális. Ilyen feladatokkal a mindennapi életben is gyakran találkozunk, hiszen napi időbeosztásunk megtervezése vagy egy többfogásos vacsora elkészítése is igényli az egymásra épülő tevékenységek átgondolt ütemezését. A gyártási, termelési folyamatok területén jelentkező ütemezési problémákban adott munkák gépeken való elvégzésének sorrendjét optimalizáljuk valamilyen célfüggvény szerint, bizonyos feltételek mellett. A probléma megoldása alatt egy ütemezés megadását értjük, mely meghatározza, hogy melyik munkát mikor és melyik gépen végezzük el [4]. Általános szabály, hogy a munkák elvégzéséhez felhasznált gépek mindegyike egy időben legfeljebb egy munkán dolgozhat és minden munkát egy időben legfeljebb egy gépen végezhetünk. Ha több gép is rendelkezésre áll, akkor feltesszük, hogy a gépek teljesen egyformák és további egyszerűsítésként az egyes munkákhoz tartozó megmunkálási időket is azonosnak, egységnyi idejűnek tételezzük fel. Ezek az előfeltevések ugyan kizárják a bonyolultabb modellek tárgyalását, de célunk nem is ez, hanem speciális ütemezési feladatok vizsgálata, a matematikai háttérként szolgáló részbenrendezett halmazokhoz tartozó elméleti eredmények gyakorlatba történő átültetésével. Az elvégzendő feladatok - vagy azok egy része - között megelőzési feltételek lehetnek definiálva, azaz bizonyos munkákat csak akkor lehet elkezdeni, ha a munkák egy meghatározott részét már befejeztük. Ezek a feltételek a munkák halmazán értelmezett részbenrendezési relációval adhatóak meg. Ha ismerjük a részbenrendezés lineáris kiterjesztésének a fogalmát, egy ilyen kiterjesztés meghatározásával már meg is adtuk az ütemezési feladat egy lehetséges megoldását a legegyszerűbb esetben, az egygépes környezetben. Ütemezési feladatok megoldásakor tehát lineáris rendezéseket keresünk. Szpilrajn klasszikus tétele értelmében bármely részben-rendezés kiterjeszthető lineárissá [9]. 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. Ekkor olyan osztályozás érdekel bennünket, ahol az osztályok közötti részbenrendezés lineáris. Adott tehát egy véges elemszámú halmaz, amelyet diszjunkt részhalmazokra, tehát osztályokra kell felosztani úgy, hogy az osztályok között létrejövő részbenrendezés lineáris legyen, az osztályokon belül pedig egy ekvivalenciareláció érvényesüljön. Az ütemezési feladat megoldása során tehát lényegében egy véges elemszámú részbenrendezett halmazt osztályozunk úgy, hogy az eredeti részbenrendezés a részhalmazok között egy újabb részbenrendezést indukáljon [8].

Az ütemezési feladatok tömör leírására gyakran alkalmazzák az $(\alpha \mid \beta \mid \gamma)$ hármast, ahol α jelöli a gépekkel kapcsolatos megkötéseket, a β mező tartalmazza az egyéb feltételeket és γ adja meg a célfüggvényt [7]. Az általunk vizsgált feladatokban α értéke 1 vagy P lehet, azaz vizsgálunk egy és többgépes feladatot is; a β halmaz a $p_j = 1$ és a *prec* feltételeket tartalmazza, ami azt jelenti, hogy minden megmunkálási idő egységnyi és vannak megelőzési feltételek is. A γ mező értéke a hagyományos jelölés szerint C_{max} az általunk vizsgált feladatokban, vagyis a teljes átfutási idő szerint optimalizálunk. Célunk tehát a munkafolyamat kezdetétől a végéig tartó időtartam minimalizálása.

Cikkünkben olyan ütemezési feladatok megoldásával foglalkozunk, ahol a megoldást mohó algoritmussal generált minimális lineáris kongruenciák segítségével adjuk meg.

2. A részbenrendezett halmazok elméletének alapfogalmai

Legyen H egy tetszőleges halmaz és $\preceq$ egy reláció a H halmaz elemein. A $\preceq$ relációt részbenrendezésnek hívjuk, ha

- reflexív, azaz minden $x \in H$ esetén $x \preceq x$,
- antiszimmetrikus, azaz $x \preceq y$ és $y \preceq x$ esetén $x = y$,
- tranzitív, azaz $x \preceq y$ és $y \preceq z$ esetén $x \preceq z$.

A $(H, \preceq)$ párt részbenrendezett halmaznak nevezzük, ha a $\preceq$ reláció egy részbenrendezés. Ha $x \preceq y$ és $x \neq y$, akkor az $x \prec y$ jelölést alkalmazzuk.

1. Megjegyzés. *Gráfelméleti terminológiával élve a $(H, \preceq)$ részbenrendezett halmaz egy irányított gráf, ahol H a csúcsok halmaza, az éleket pedig a $\preceq$ relációval adjuk meg. A részbenrendezés tranzitív és antiszimmetrikus tulajdonsága segítségével könnyen belátható, hogy a gráf aciklikus, azaz nem tartalmaz kört, leszámítva a reflexivitásból adódó hurkokat.*

Legyen $(H, \preceq)$ egy részbenrendezett halmaz. Azt mondjuk, hogy az $x \in H$ és $y \in H$ elemek összehasonlíthatóak, ha vagy $x \preceq y$ vagy $y \preceq x$ (vagy mindkettő) teljesül. Ellenkező esetben az elemeket nem-összehasonlíthatóknak hívjuk és az $x \parallel y$ jelölést alkalmazzuk.

Az $a \in H$ elemet a H halmaz

- maximális elemének hívjuk, ha minden $b \in H$ esetén vagy $b \preceq a$ vagy $b \parallel a$,
- legnagyobb elemének hívjuk, ha minden $b \in H$ esetén $b \preceq a$ teljesül,
- minimális elemének hívjuk, ha minden $b \in H$ esetén vagy $a \preceq b$ vagy $b \parallel a$,
- legkisebb elemének hívjuk, ha minden $b \in H$ esetén $a \preceq b$ teljesül.

Egy részbenrendezett halmaznak legfeljebb egy legnagyobb és egy legkisebb eleme lehet, de lehet több maximális vagy minimális eleme is.

Ha a részbenrendezési relációval a H halmaz bármely két eleme összehasonlítható, akkor a relációt teljes rendezésnek hívjuk. Ha $(H, \preceq)$ részbenrendezett halmaz, akkor láncnak hívjuk az $A \subseteq H$ halmazt, amennyiben bármely $a \in A$, $b \in A$ elemek esetén $a \preceq b$ vagy $b \preceq a$ teljesül. Tehát egy részbenrendezett halmaz teljesen rendezett részhalmazai a láncok. Az A lánc hossza alatt az A halmaz elemszámát értjük. Az A láncot maximális láncnak nevezzük, ha nem valódi részhalmaza egyetlen más láncnak sem. A maximális lánc hosszát a részbenrendezett halmaz magasságának hívjuk. Vegyük észre, hogy ha egy lánc maximális, akkor tartalmaznia kell a részbenrendezett halmaz egy minimális és egy maximális elemét is. Ha $B \subseteq H$, és B bármely két eleme nem-összehasonlítható, akkor B-t antiláncnak hívjuk. A B antiláncot maximálisnak nevezzük, ha nem valódi részhalmaza egyetlen más antiláncnak sem. A maximális antilánc elemszámát a részbenrendezett halmaz szélességének hívjuk. Vegyük észre, hogy bármely A láncnak és bármely B antiláncnak legfeljebb egy közös eleme lehet, hiszen két közös pont esetén azoknak egyszerre kellene összehasonlíthatóknak és nem-összehasonlíthatóknak is lenniük, ami lehetetlen. A láncokkal és antiláncokkal kapcsolatos alapvető összefüggéseket írja le Dilworth tétele és annak duálisa [10].

1. Tétel (Dilworth tétel)**.** *Legyen w a $(H, \preceq)$ részbenrendezett halmaz szélessége. Ekkor H felbontható w darab lánc uniójára, kevesebbre viszont nem.*

2. Tétel (Duális Dilworth tétel)**.** *Legyen h a $(H, \preceq)$ részbenrendezett halmaz magassága. Ekkor H felbontható h darab antilánc uniójára, kevesebbre viszont nem.*

2. Megjegyzés. *A duális Dilworth tétel Mirsky tételként is ismert az irodalomban* [2].

Az ütemezési feladatok szempontjából nagy jelentősége van a megelőzési feltételek által kijelölt részbenrendezés lineáris kiterjesztéseinek.

1. Definíció. *Legyen $(H, \preceq)$ részbenrendezett halmaz és $\preceq_T$ teljes rendezés a H halmazon. Azt mondjuk, hogy a $\preceq_T$ teljes rendezés a $\preceq$ részbenrendezés lineáris kiterjesztése, vagy más néven a H halmaz topologikus sorrendje, ha minden*

$$x, y \in H, x \preceq y \text{ esetén } x \preceq_T y \text{ teljesül.}$$

3. Tétel (Szpilrajn [9])**.** *Tetszőleges $(H, \preceq)$ részbenrendezett halmaz esetén a $\preceq$ részbenrendezésnek van lineáris kiterjesztése.*

A lineáris kiterjesztés általában nem egyértelmű. Egy részbenrendezésnek az alaphalmaz és a reláció elemszámától függően számos lineáris kiterjesztése lehet, azonban az összes lánc generálására és megvizsgálására általában nincs elegendő kapacitás.

Az ütemezési feladatot modellező $(M, \preceq)$ részbenrendezés esetén a megoldáshoz az M halmaz elemeit célszerű diszjunkt részhalmazokra felosztani. Alkalmazzuk a továbbiakban az $\{M_1, M_2, \ldots, M_t\}$, $(t \leq n)$ jelölést egy ilyen osztályozásra. Mivel $\{M_1, M_2, \ldots, M_t\}$, $(t \leq n)$ osztályozása M-nek, így tartozik hozzá egy ρ ekvivalenciareláció. Ennek megfelelően

$$M/\rho = \{M_1, M_2, \ldots, M_t\},$$

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

$$\varphi : M \to M/\rho$$

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

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

ha $m_i \in M_j$. A rendezés-kongruenciának a [5] dolgozatban megadott fogalmát esetünkben az alábbi megfogalmazásban célszerű használni.

2. Definíció. *A ρ ekvivalencia relációt rendezés-kongruenciának nevezzük az M halmazon, ha az M/ρ faktorhalmazon értelmezhető olyan $\preceq_\rho$ részbenrendezés, amelyre nézve a φ függvény rendezésőrző, azaz bármely $m_i \preceq m_j$ $(m_i, m_j \in M)$ esetén $\varphi(m_i) \preceq \varphi(m_j)$ teljesül.*

A fenti definícióban alkalmazott $\preceq_\rho$ jelölésre az indukált részbenrendezés elnevezést használjuk. A 2. Definícióból azonnal adódik, hogy ha ρ rendezés-kongruenciája az $(M, \preceq)$ részben-rendezett halmaznak, akkor ρ egybeesik a φ izoton függvény magjával. Az $(M, \preceq)$ részbenrendezett halmaz rendezés-kongruenciáinak halmazára a $\vartheta(M)$ jelölést használjuk. Az ütemezési feladat megoldásához olyan $\rho \in \vartheta(M)$ kongruenciára van szükség, amelyre a kialakított részhalmazok között létrejövő $\preceq_\rho$ indukált részbenrendezés lineáris, tehát az $(M/\rho, \preceq_\rho)$ faktorhalmaz lánc.

3. Definíció. *Egy $\rho \in \vartheta(M)$ rendezés-kongruenciát az $(M, \preceq)$ lineáris rendezés-kongruenciájának nevezünk, ha az $(M/\rho, \preceq_\rho)$ faktorhalmaz lánc. Minimális lineáris rendezés-kongruenciáról beszélünk, ha $(M, \preceq)$-nek nincs olyan $\theta \neq \rho$ lineáris rendezés-kongruenciája, amelyre $\theta \subset \rho$.*

A nem túl nagy elemszámú részbenrendezett halmazok jól szemléltethetők Hasse-diagrammal. A $(H, \preceq)$ részbenrendezett halmazt az 1. Megjegyzés értelmében egy irányított gráfként ábrázoljuk, de elhagyjuk a reflexitivás miatti hurkokat és a tranzitivitásból adódó éleket. A reflexív és tranzitív redukció módszerével úgy csökken az irányított gráf éleinek száma, hogy a modellezett ütemezési feladatban a megelőzési feltételek nem változnak meg. Általában a $\preceq$ reláció által kijelölt irányítást sem jelöljük külön az ábrán, hanem $x \prec y$ esetén az x elemet az y elem alatt helyezzük el a diagramon.

A mohó algoritmusok gyakran sikeresen használható heurisztikák optimalizálási feladatok megoldása során. Általános jellemzőjük, hogy a probléma egy vagy több megoldását úgy alkotják meg, hogy választások sorozatát hajtják végre. A lokálisan legjobb lépés választásával generált megoldás azonban nem feltétlenül optimális, azonban bizonyos feladattípusok esetén érdemes applikálni a módszert, ugyanis számos esetben jól alkalmazható és egyszerűen implementálható. További előnye, hogy a mohó algoritmusok futási ideje nagyon jó, általában lineáris. A mohó stratégiával egy megoldandó feladatra akkor adható optimális megoldás, ha az a részproblémák esetén kapott optimális megoldásokból felépíthető. Ütemezési feladatok megoldása esetén a módszer bizonyíthatóan sok esetben nyújt optimális vagy közel optimális megoldást. Ismert, hogy az ütemezési feladatot modellező részbenrendezés lineáris kiterjesztéseinek létrehozására mohó stratégiát tartalmazó algoritmusok megfelelő kimeneti eredménnyel alkalmazhatók [8], [10].

3. Megelőzési feltételeket tartalmazó ütemezési feladatok

Részbenrendezett halmazok a gyakorlatban ott lépnek fel, ahol a vizsgált objektumok között természetes módon hierarchikus viszonyok létesíthetők. A megelőzési feltételeket is tartalmazó ütemezési problémák tehát részbenrendezett halmazokkal jól modellezhetőek, akár gyártási-termelési folyamatok keretében, akár általánosabb értelemben tekintjük őket. Tegyük fel, hogy adott az elvégzendő munkák egy $M = \{m_1, m_2, \ldots, m_n\}$ véges halmaza, továbbá adott a munkák sorrendjére vonatkozóan egy precedencia-reláció, azaz $m_i \preceq m_j$ azt jelöli, hogy az i-edik munka elvégzése időben megelőzi a j-edik munkát ($i, j \in \{1, 2, \ldots, n\}$). Nyilvánvaló, hogy az így definiált relációval az $(M, \preceq)$ pár egy részbenrendezett halmaz.

3.1. Egygépes ütemezés

Az $(1 \mid P_j = 1, prec \mid C_{max})$ hármassal definiált ütemezési feladatban egy gép végzi az összes munkafeladatot, mindegyiket egységnyi idő alatt, a munkák között megelőzési feltételek is adottak és célunk a teljes átfutási idő minimalizálása. A 3. Tétel alapján minden részben rendezés kiterjeszthető lineáris rendezéssé. Könnyű látni, hogy egy ilyen, lineáris rendezéssel definiált ütemezés során a gép folyamatosan dolgozik, így az n munkát n idő alatt végzi el, optimálisan megoldva így az ütemezési feladatot. A kérdés tehát csak az, hogy milyen algoritmussal keressük az $(M, \preceq)$ részbenrendezett halmaz esetén az M lehetséges topologikus sorrendjeit.

A Trotter-féle mohó algoritmus az $(M, \preceq)$ részbenrendezett halmaz esetén a $\preceq$ reláció lineáris kiterjesztését úgy állítja elő, hogy az első lépésben az M minimális elemei közül választ egyet. Ha a lánc első $x_1, ..., x_i$ tagja már megvan, akkor az $i + 1$-edik elemet az

$$(M \setminus \{x_1, ..., x_i\}, \preceq)$$

részbenrendezett 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 [10]. Könnyen látható, hogy ez az algoritmus egy részbenrendezett halmaz esetén számos lineáris kiterjesztés generálását lehetővé teszi, viszont nem alkalmas az összes lineáris kiterjesztés létrehozására. Egygépes ütemezések esetén, ahol maga a lineáris kiterjesztés a feladat megoldása, a mohó algoritmussal gyors megoldást adhatunk.

1. Példa. *Legyen*

$$M = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}, m_{16}\}$$

és az $(M, \preceq)$ részbenrendezett halmazzal reprezentált ütemezési feladatot az 1. ábrán látható Hasse-diagrammal szemléltetjük. A Trotter-féle mohó algoritmussal ekkor például az alábbi láncokat kaphatjuk meg:

$$L_1 : m_{16}, m_{15}, m_{14}, m_{13}, m_{12}, m_{11}, m_{10}, m_8, m_6, m_9, m_7, m_3, m_5, m_4, m_2, m_1$$

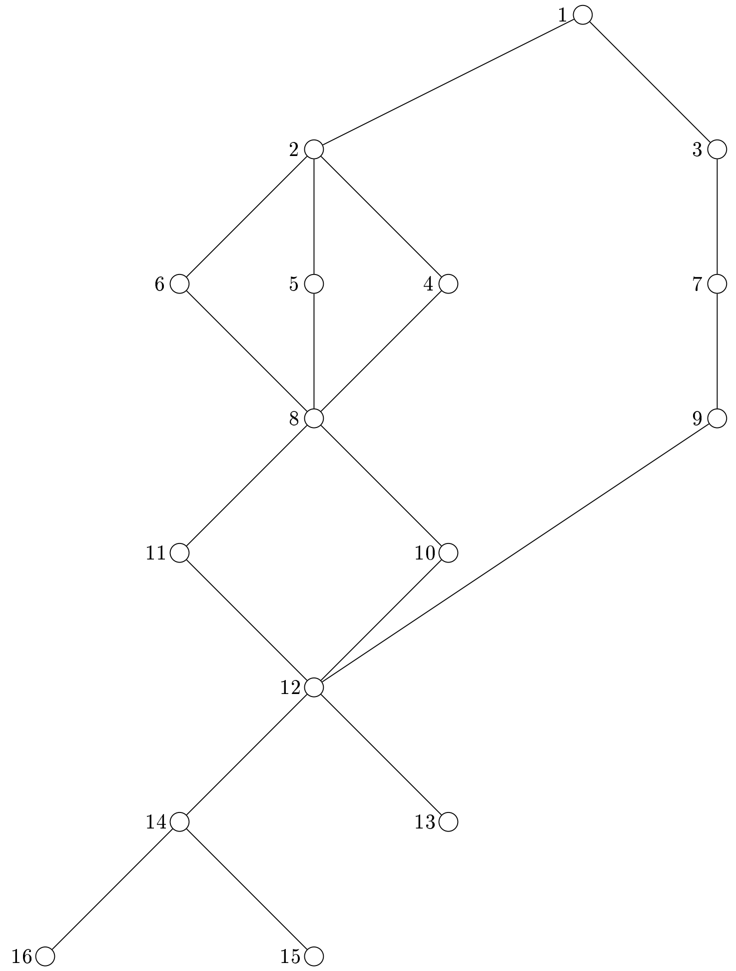
és

$$L_2 : m_{16}, m_{15}, m_{14}, m_{13}, m_{12}, m_9, m_7, m_3, m_{11}, m_{10}, m_8, m_6, m_5, m_4, m_2, m_1.$$

Az algoritmussal még további láncok is készíthetők. Könnyen látható az is, hogy például az

$$L_3 : m_{16}, m_{15}, m_{14}, m_{13}, m_{12}, m_9, m_{10}, m_{11}, m_7, m_8, m_6, m_5, m_4, m_2, m_3, m_1$$

lánc szintén lineáris kiterjesztése az eredeti részbenrendezésnek, azonban ezt a láncot a Trotter-féle mohó algoritmus nem állítja elő.



1. ábra. *Hasse-diagram a vizsgált ütemezési feladathoz*

3.2. Többgépes ütemezés

A korábban tárgyalt egygépes ütemezési feladatot a munkák között adott részbenrendezési reláció egy megfelelő lineáris kiterjesztésének megadásával oldottuk meg, ami úgy rendezi sorrendbe az egyes munkákat, hogy az azokra megszabott sorrendi kötöttségeket betartja. Ezzel a megoldással a teljes munka elvégzéséhez szükséges összidő megegyezik a részmunkák elvégzéséhez szükséges időtartamok összegével, hiszen csak akkor kezdhetünk bele a következő munkába, ha az előzőt már befejeztük. Gyakran van azonban lehetőségünk arra, hogy csökkentsük a teljes átfutási időt a munkafolyamatok párhuzamosításával, hiszen lehetséges, hogy ugyanannak a munkafázisnak az elvégzésére egyszerre több gép is rendelkezésünkre áll és általában a munkák között vannak nem összehasonlíthatóak, azaz olyanok, melyek esetén nem kell megvárnunk az egyik befejezését ahhoz, hogy elkezdhessük a másikat. Valójában egyetlen olyan eset van, amikor csak szekvenciálisan képzelhető el a munkafolyamat, amikor az $m_1, m_2, \ldots, m_n$ munkák esetén teljesülnie kell az $m_i \prec m_{i+1}$ relációnak minden $i \in \{1, 2, \ldots, n-1\}$-re. Ekkor az ütemezési feladatnak egyetlen megoldása van, méghozzá az n hosszúságú $\{m_1, m_2, \ldots, m_n\}$ lánc. A párhuzamosan működő berendezések esetén a problémák megközelíthetők két részfeladat összességeként is. Egyrészt meg kell találni a munkák gépek közötti optimális elosztását, másrészt meg kell határozni az egyes gépeken az elvégzendő feladatok leghatékonyabb sorrendjét.

A továbbiakban a többgépes ütemezések legegyszerűbb változatával, a párhuzamos gépek esetével foglalkozunk, azaz feltételezzük, hogy korlátlan számú, teljesen egyforma gép áll rendelkezésünkre. Ha a gépek száma előre rögzített, akkor további speciális ütemezési feladatosztályokba tartozó problémákkal állunk szemben. Az $(P \mid P_j =$

$1, prec \mid C_{max}$) típusú feladatok általánosságban NP-nehéz problémák, de bizonyos megkötésekkel polinomiális időben megoldható feladatokhoz juthatunk.

Coffman és Graham algoritmusa például a kétgépes ($P2 \mid P_j = 1, prec \mid C_{max}$) feladatot helyesen oldja meg, tehát optimális megoldást eredményez [1], [4]. Az algoritmus első lépésében az ütemezési feladatot modellező irányított gráf tranzitív redukcióját kell elvégezni. Esetünkben erre nincs szükség, hiszen a Hasse-diagram, amellyel az ütemezési feladatot modellezzük, nem tartalmaz háromszöget. A második lépésben osztályozzuk, majd címkézzük természetes számokkal a munkákat az alábbi lépésekben:

1. Kiválasztjuk a részben-rendezett halmaz maximális elemeit. Ezeket 1-től kezdődően beszámozzuk.
2. Megkeressük azokat az elemeket, amelyek maximális elemekké válnak, ha az előző lépésben címkével ellátott elemeket töröljük. Ezen elemek címkézéséhez a korábban címkével ellátott csúcsokhoz rendelt értékeket használjuk fel úgy, hogy minden pont mellé odaírjuk a rákövetkező csúcsok címkéjét csökkenő sorrendben, majd a kapott sorozatokat lexikografikus sorrend szerint rendezzük és ennek megfelelően címkézzük.
3. A fenti lépést addig folytatjuk, míg valamennyi munka címkézetté válik.
4. Amikor minden munka címkézett, akkor csökkenő címke szerinti listás ütemezést végzünk, azaz amikor egy gép szabaddá válik, akkor ráütemezzük a soron következő feladatot.

Megjegyezzük, hogy a Coffman-Graham algoritmus címkéző lépése lényegében a vizsgált részbenrendezés egy lineáris kiterjesztését állítja elő, míg utolsó lépése mohó stratégiát követ, hiszen minden döntési pontban az adott helyzetben optimálisnak látszó, azaz szabad gépet választja.

2. Példa. *A 1. ábrán látható Hasse-diagramon a csúcsok számozása a Coffman-Graham algoritmus címkéző lépésének megfelelően történt. Két gépre az alábbi optimális ütemezés adható az algoritmussal:*

1. táblázat

G_1 :	m_{16}	m_{14}	m_{12}	m_{11}	m_8	m_6	m_4	m_2	m_1
G_2 :	m_{15}	m_{13}	m_9	m_{10}	m_7	m_5	m_3	$-$	$-$

Kétgépes környezetben tehát 9 osztályra bontható az elvégzendő munkák halmaza.

Igazolható, hogy a Coffman-Graham algoritmus három, illetve több gép esetén már nem feltétlenül ad optimális megoldást.

Formálisan a munkák párhuzamosítása többgépes ütemezés esetén azt jelenti, hogy az $M = \{m_1, m_2, \ldots, m_n\}$ halmazt páronként diszjunkt részhalmazokra osztjuk fel (particionáljuk), azaz létrehozzuk az $M_1, M_2, \cdots, M_t$, $(t \leq n)$ blokkokat úgy, hogy

$$M = M_1 \cup M_2 \cup \cdots \cup M_t,$$

és az egy blokkba került munkák ne legyenek egymással összehasonlíthatóak. Ha megadjuk az így létrehozható partíciók elemszámának minimumát, akkor megválaszoljuk azt a kérdést, hogy mennyi az a legkevesebb idő, ami alatt a teljes munkafolyamat elvégezhető az optimális párhuzamosítás alkalmazásával.

Ha a Trotter-féle mohó algoritmust kiegészítjük a [8] cikkben szereplő algoritmussal, akkor megkaphatjuk adott részbenrendezés esetén az ütemezési feladat megoldását minimális lineáris kongruencia formájában. Ha tekintjük az

$$L : x_1, ..., x_n \qquad (x_i = m_j, \quad i, j \in \{1, 2, ..., n\})$$

láncot, akkor elő kell állítanunk az

$$M_1 = [x_1, x_{i_1}], \quad M_2 = [x_{i_1+1}, x_{i_2}], \quad ... \quad M_t = [x_{i_{t-1}+1}, x_n]$$

intervallumokat. Az így kialakított partíció az M halmaz kongruenciája. Lineáris kongruenciát a 3. Definíció értelmében akkor kapunk, ha az

$$\{M_1, M_2, ..., M_t\}$$

halmazon indukált részbenrendezés lineáris. Ez akkor következik be, ha bármely két egymást követő M_j, M_{j+1} $(j \in \{1, ..., t-1\})$ intervallum esetén léteznek olyan $x_k \in M_j$ és $x_l \in M_{j+1}$ $(1 \leq k < l \leq n)$ elemek, amelyekre fennáll, hogy $x_k \preceq x_l$. Ahhoz, hogy minimális lineáris kongruenciát állítsunk elő az szükséges, hogy a kialakított $M_1, M_2, ..., M_t$ intervallumok $(M, \preceq)$-ben antiláncok legyenek és bármely két szomszédos M_j, M_{j+1} $(j \in \{1, ..., t-1\})$ intervallum esetén találjunk olyan $x_k \in M_j$ és $x_l \in M_{j+1}$ $(1 \leq k < l \leq n)$ elemeket, amelyekre fennáll a rákövetkezőségi tulajdonság, azaz $x_k \prec x_l$. Ezt a [8] cikk algoritmusa azzal éri el, hogy a Trotter-féle mohó algoritmusból bemenetként kapott láncot úgy darabolja, hogy rákövetkezőségi kapcsolat esetén új blokkot alakít ki, ezzel biztosítva azt, hogy az azonos osztályba tartozó elemek antiláncot adjanak. Ezzel a technikával azonban általában nem kapunk optimális megoldást az ütemezési feladatra.

3. Példa. *Ha az 1. ábrával szemléltetett részbenrendezéshez tartozó ütemezési feladat esetén az 1. példában megadott L_1 lineáris kiterjesztést tekintjük, akkor a [8] cikkben található algoritmus az alábbi minimális lineáris kongruenciát eredményezi:*

$$\{m_{16}, m_{15}\}, \{m_{14}, m_{13}\}, \{m_{12}\}, \{m_{11}, m_{10}\}, \{m_8\}, \{m_6, m_9\},$$

$$\{m_7\}, \{m_3, m_5\}, \{m_4\}, \{m_2\}, \{m_1\}.$$

Az L_2 lánc esetén pedig a

$$\{m_{16}, m_{15}\}, \{m_{14}, m_{13}\}, \{m_{12}\}, \{m_9\}, \{m_7\}, \{m_3, m_{11}\},$$

$$\{m_{10}\}, \{m_8\}, \{m_6, m_5, m_4\}, \{m_2\}, \{m_1\}$$

megoldást kapjuk. Látható, hogy az elvégzendő munkákat 11 blokkra bontottuk mindkét esetben. Megállapíthatjuk, hogy az L_1 lánc darabolásával kétgépes környezetben nem kapunk optimális megoldást, hiszen a Coffman-Graham algoritmussal 9 munkafázist alakítottunk ki. Az is könnyen látható, hogy az L_2 láncból szintén nem kapunk optimális megoldást.

Az optimális megoldás előállításához a [8] cikk algoritmusának bemeneteként a Coffman-Graham algoritmus címkéző lépését is használhatjuk, hiszen az láncot eredményez. Könnyen látható, hogy a kimenetként adódó blokkok nem adnak minimális lineáris kongruenciát, azaz nem kapjuk meg az ütemezési feladat megoldását. A blokkok antilánc tulajdonságát kell helyreállítani, így az alábbi lépésekkel generáljuk az optimális megoldást:

1. Címkézzük az elemeket a Coffman-Graham algoritmus címkéző lépésének megfelelően.
2. A kapott láncot a rákövetkezőségi tulajdonság felhasználásával blokkokra bontjuk.
3. A blokkokon antilánc vizsgálatot végzünk. Ha a blokk antilánc, akkor osztályként kezeljük, ha nem, akkor feldaraboljuk csökkenő címkék szerint létrehozott antiláncok uniójára, így alakítva ki az ekvivalenciaosztályokat.

4. Példa. *Tekintsük ismét az 1. ábrával szemléltetett ütemezési feladatot! A láncban ekkor csökkenő címkével találjuk az elemeket:*

$$L_4 : m_{16}, m_{15}, m_{14}, m_{13}, m_{12}, m_{11}, m_{10}, m_9, m_8, m_7, m_6, m_5, m_4, m_3, m_2, m_1$$

Az eljárás eredményeként adódó blokkok:

$$M_1 = \{m_{16}, m_{15}\}, \quad M_2 = \{m_{14}, m_{13}\}, \quad M_3 = \{m_{12}\},$$

$$M_4 = \{m_{11}, m_{10}, m_9, m_8, m_7, m_6, m_5, m_4, m_3, m_2\}, \quad M_5 = \{m_1\}$$

Vizsgáljuk a kialakított blokkok esetén az antilánc tulajdonságot! M_1, M_2, M_3 és M_5 antiláncok, viszont M_4 nem az, így szükség van az antiláncokra tördelésre:

$$M_4 = \{m_{11}, m_{10}, m_9\} \cup \{m_8, m_7\} \cup \{m_6, m_5, m_4\} \cup \{m_3, m_2\}$$

Három gépre az alábbi, 8 osztályt tartalmazó optimális megoldást kaptuk:

2. táblázat

G_1 :	m_{16}	m_{14}	m_{12}	m_{11}	m_8	m_6	m_3	m_1
G_2 :	m_{15}	m_{13}	–	m_{10}	m_7	m_5	m_2	–
G_3 :	–	–	–	m_9	–	m_4	–	–

A duális Dilworth tételből következik (2. Tétel), hogy egy tetszőleges részbenrendezett halmaz esetén a maximális lánc hossza egyenlő az antiláncokkal történő fedés esetén a szükséges antiláncok minimális számával. Ennek megfelelően azonnal látható, hogy legalább annyi antiláncra van szükség a fedéshez, mint ahány eleme a maximális hosszúságú láncnak van. Tehát a párhuzamos berendezéseket tartalmazó ütemezési feladatok megoldásának előállításához olyan antilánc fedést és láncot kell keresnünk, amelyek esetén az antiláncok száma megegyezik a lánc hosszával. Ezt a kérdést járjuk körül a 2. Tétel segítségével. Először átfogalmazzuk a tételt [6] alapján, hogy céljainknak jobban megfeleljen.

4. Tétel. *Legyen $(M, \preceq)$ egy véges, t magasságú részbenrendezett halmaz. Ekkor M felbontható az $M_1, M_2, \cdots, M_t$ részhalmazok uniójára úgy, hogy minden $i \in \{1, 2, \ldots, t\}$ és $m \in M_i$ esetén az összes olyan p elemre, melyre $p \prec m$, $p \in M_1 \cup M_2 \cup \cdots \cup M_{i-1}$ is teljesül.*

Bizonyítás. A bizonyítást részletesen közöljük, mivel egy lehetséges algoritmust ad az egymással párhuzamosan végezhető feladatok csoportjainak kialakítására. Legyen $m \in M$ egy tetszőleges elem az $(M, \preceq)$ részbenrendezett halmazban. Az m elemet soroljuk be az M_i részhalmazba, ha az m-ben végződő leghosszabb lánc hossza i. Ezzel létrehoztuk az $M_1, M_2, \cdots, M_t$ részhalmazokat, melyek páronként diszjunktak, hiszen nincs olyan elem, amit több osztályba is be lehetne sorolni, és egyik részhalmaz sem üres, hiszen ha a leghosszabb lánc elemei $m_1 \prec m_2 \prec \cdots \prec m_t$, akkor $m_i \in M_i$, $i = \{1, 2, \ldots, t\}$.

Tegyük fel az állítással ellentétben, hogy van olyan i index és $m \in M_i$ elem, amelyhez van olyan $p \prec m$ elem, hogy $p \notin M_1 \cup M_2 \cup \cdots \cup M_{i-1}$. Ekkor kell lennie olyan p-ben végződő L láncnak, amelynek a hossza legalább i. Mivel $p \preceq m$ és $p \neq m$ az L lánc kiterjeszthető az m elemig. Így találtunk olyan láncot, amely m-ben végződik és a hossza legalább $i + 1$, ami ellentmondás. □

1. Következmény. *Az $(M, \preceq)$ ütemezési feladat elvégzéséhez szükséges minimális idő megegyezik az $(M, \preceq)$ részbenrendezett halmaz magasságával, azaz maximális láncának hosszával.*

Bizonyítás. Vegyük észre, hogy a 4. Tétel alapján az M_i blokk elemei egyszerre, az i-edik lépésben ütemezhetőek, hiszen az összes M_i-beli munkánál korábban elvégzendő munkák már valamelyik alacsonyabb sorszámú blokknak az elemei. □

3. Megjegyzés. *Az ütemezéselméletben a feladatot modellező részbenrendezett halmaz leghosszabb láncát gyakran kritikus útnak nevezik, hiszen a lánc hosszánál kevesebb elemszámú lépésben a kérdéses munka nem végezhető el.*

A 4. Tétel bizonyításában leírt módszerrel létrehozhatjuk az $M = M_1 \cup M_2 \cup \cdots \cup M_t$ partíciót, ahol a partícióban szereplő M_i részhalmazok az egymással párhuzamosan végezhető munkákat tartalmazzák. A módszer algoritmizálásakor minden egyes m elemről el kell dönteni, hogy melyik az a leghosszabb lánc, mely m-ben végződik. Az ilyen láncok feltérképezése és a lánchosszok maximumának meghatározása nagy elemszámú és relációszámú részbenrendezett halmaz esetén nem egyszerű feladat. Az egymással párhuzamosan végezhető munkák csoportjainak meghatározása mohó algoritmusok segítségével is megvalósítható. Ezt az eljárást *minimális elem módszernek* nevezzük, mert lényegi lépése a részbenrendezett halmaz minimális elemeinek meghatározása. Egy adott a elem esetén megvizsgáljuk, van-e olyan $b \neq a$ elem, amelyre $b \preceq a$. Ha egyetlen ilyen elem sincs, akkor a minimális, egyébként nem. Első lépésben az A_1 halmazba összegyűjtjük az $(M, \preceq)$ részbenrendezett halmaz összes minimális elemét, majd M-ből töröljük az A_1 elemeit és a relációk listájából az összes relációt, melyben A_1-beli elem szerepelt. Az eljárást folytatjuk az $(M \setminus A_1, \preceq)$ részbenrendezett halmazra, ennek minimális elemei alkotják az A_2 halmazt és így tovább. Világos, hogy minden egyes A_i halmaz M-nek egy antilánca, hiszen egy részbenrendezett halmaz minimális elemei nem lehetnek egymással összehasonlíthatók. Ha M legnagyobb láncának hossza t, akkor a fenti módszerrel éppen t antilánc uniójára bontjuk M-et, hiszen a maximális lánc elemei mind különböző antiláncokba kerülnek. Vegyük észre, hogy a minimális elem módszerrel kapott partíció $A_1, A_2, \ldots, A_t$ blokkjai megegyeznek a 4. Tétel által előállított felbontás $M_1, M_2, \ldots, M_t$ blokkjaival, továbbá minimális lineáris kongruenciát eredményeznek. Tegyük fel, hogy $a \in A_i$ valamely $i = \{1, 2, \ldots, t\}$ esetén, azaz a az i-dik lépésben lett minimális elem. Ez azt jelenti, hogy előtte már volt $i - 1$ nála kisebb elem, tehát a egy i hosszú lánc befejező eleme, és ez az a-ban véget érő leghosszabb lánc volt, mert különben a már hamarabb minimális elemmé vált volna. Ezért $a \in M_i$, azaz $A_i \subset M_i$. Megfordítva, tegyük fel, hogy $m \in M_i$ valamely $i = \{1, 2, \ldots, t\}$ esetén. Ekkor van legalább egy i hosszú lánc, melynek m a maximális eleme. Ennek a láncnak a j-edik elemét A_j-be soroljuk a minimális elem módszerrel $j = \{1, 2, \ldots, i - 1\}$), a lánc i-edik elemét, m-et pedig az A_i halmazba, tehát $M_i \subset A_i$ is teljesül, azaz $M_i = A_i$ minden $i = \{1, 2, \ldots, t\}$) esetén.

5. Példa. *Tekintsük újra az 1. ábrával szemléltetett ütemezési feladatot! Alkalmazzuk a minimális elem módszert! Ekkor négy gépes környezetre az alábbi optimális megoldást kapjuk:*

3. táblázat

G_1 :	m_{16}	m_{14}	m_{12}	m_{11}	m_8	m_6	m_2	m_1
G_2 :	m_{15}	–	–	m_{10}	m_7	m_5	–	–
G_3 :	m_{13}	–	–	m_9	–	m_4	–	–
G_4 :	–	–	–	–	–	m_3	–	–

4. Összefoglalás

Ütemezési problémák gyakran fordulnak elő úgy az operációkutatás, mind az elméleti számítástudomány területén. Számos modell az NP-nehéz feladatok osztályába tartozik, ezért az optimális megoldást adó exponenciális időigényű algoritmusok idő vagy memória gondok miatt nem használhatók a gyakorlatban jelentkező feladatok esetén. Fontos tehát olyan algoritmusok kidolgozása, amelyek gyorsan és hatékonyan optimális vagy közel optimális megoldást eredményeznek redukált környezetben. Cikkünkben megelőzési feltételeket tartalmazó ütemezési problémákat modellezünk részbenrendezett halmazokkal. Az optimális megoldást mohó lépést tartalmazó algoritmussal állítjuk elő minimális lineáris kongruencia formájában.

5. Köszönetnyilvánítás

A cikkben ismertetett kutató munka az EFOP-3.6.1-16-2016-00011 jelű „Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése” projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

6. Irodalom

[1] Coffman, E.G., Graham, R.L.: *Optimal scheduling for two-processor systems*, Acta Informatica Vol. 1 (1972) pp. 200-213. https://doi.org/10.1007/BF00288685

[2] Frank, A., Jordán, T.: *Diszkrét optimalizálás*, ELTE, Typotex, Budapest, 2014. ISBN: 9789632792316

[3] Iványi, A. (szerk.): *Informatikai algoritmusok*, ELTE, Eötvös Kiadó, Budapest, 2005. ISBN: 9634637752

[4] Jordán, T., Recski, A., Szeszlér, D.: *Rendszeroptimalizálás*, Typotex, Budapest, 2004. ISBN: 9639548391

[5] Körtesi, P., Radeleczki, S., Szilágyi, Sz.: *Congruences and isotone maps on partially ordered sets*, Mathematica Pannonica 16/1 (2005) pp.39-55.

[6] Lehman, E., Leighton, F.T., Meyer, A.R.: *Mathematics for Computer Science* (2010) http://courses.csail.mit.edu/6.042/fall10/mcs-ftl.pdf

[7] Pinedo, M.: *Scheduling – Theory, Algorithms and Systems)*, Prentice Hall, 2002.

[8] Szilágyi, Sz.: *Rendezés-kongruenciák alkalmazása többgépes párhuzamos ütemezések esetén*, GÉP 63. évf. 5. sz. (2012), pp. 67-70.

[9] Szpilrajn, E.: *Sur l'extension de l'ordre partiel*, Fund. Math. 16 (1930) pp.386-389. https://doi.org/10.4064/fm-16-1-386-389

[10] Trotter, W. T.: *Combinatorics and Partially Ordered Sets, Dimension Theory*, The John Hoplkins University Press, Baltimore-London, 1992.