

MTA DOKTORI ÉRTEKEZÉS

Tézisfüzet

Összetett Ütemezési Problémák

Kis Tamás

Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási
Kutatóintézet

2017

1. fejezet

Bevezetés

Az ütemezéselmélet az operációkutatás egyik kiemelkedően fontos területe. Ezt mutatja a témában megjelenő nagyszámú publikáció, valamint gyakorlati alkalmazás. Fontos megemlíteni, hogy a gyakorlati igények új elméleti kutatásokhoz vezettek, és vezetnek ma is, és a dolgozatban vizsgált problémák egy részét is a gyakorlat motiválta.

Röviden fogalmazva, egy ütemezési probléma megoldása során erőforrásokat, és végrehajtási időpontokat rendelünk tevékenységekhez, miközben betartunk különböző korlátozásokat, és egy (vagy több) célfüggvényt optimalizálunk.

A projektütemezésben az erőforrások alapvetően két kategóriába sorolhatóak a használatukra vonatkozó korlátozások szerint: (i) *megújulóak*, és (ii) *nem megújulóak*. A megújuló erőforrások használata minden t időpillanatban korlátos, míg a nem megújuló erőforrások esetében minden $[0, t)$ intervallumon korlátos a felhasználható mennyiség. Megemlítjük, hogyha mindkét féle korlátozás elő van írva egy erőforrásra, akkor *kétszeresen korlátozottnak* hívjuk.

Mindenekelőtt formálisan definiálom az erőforrás korlátos projekt ütemezési probléma alapesetét (resource-constrained project scheduling problem, RCPSP), amelynek különböző módosításai, variánsai alkotják a dolgozatban vizsgált problémákat.

Az RCPSP-ben adott egy véges tevékenység-halmaz, $\mathcal{J}$, egy véges erőforrás-halmaz, $\mathcal{R}$, kizárólag megújuló erőforrásokkal, és egy megelőzési reláció, $\mathcal{E} \subseteq \mathcal{J} \times \mathcal{J}$, a tevékenységek között. Minden $j \in \mathcal{J}$ tevékenységnek van egy $p_j \geq 0$ végrehajtási ideje, és minden $i \in \mathcal{R}$ erőforrásból egy $a_{ij} \geq 0$ igénye, továbbá minden erőforrásnak egy $b_i > 0$ kapacitása. A tevékenységek ha egyszer elkezdődtek, nem szakíthatók meg, tehát ha a j tevékenység elkezdődik egy S_j időpontban, akkor p_j időegységgel később fejeződik be. A végrehajtása során lefoglal a_{ij} mennyiséget minden egyes $i \in \mathcal{R}$ erőforrásból.

Az erőforrások minden pillanatban legfeljebb a kapacitásuk erejéig foglalhatóak le, tehát ha $S \in \mathbb{R}_+^{\mathcal{J}}$ a tevékenységek kezdési idejét meghatározó vektor, akkor

$$\sum_{j \in \mathcal{J} : S_j \leq t \leq S_j + p_j} a_{ij} \leq b_i, \quad \forall i \in \mathcal{R}, t \geq 0 \quad (1.1)$$

korlátnak kell teljesülnie. A megelőzési relációt pedig úgy kell figyelembe venni, hogy $(j, k) \in \mathcal{E}$ esetén j befejezése előtt k ne kezdődjön el, azaz

$$S_j + p_j \leq S_k, \quad \forall (j, k) \in \mathcal{E} \quad (1.2)$$

teljesüljön. Azokat az $S \in \mathbb{R}_+^{\mathcal{J}}$ vektorokat, amelyek teljesítik az (1.1) és (1.2) feltételeket *megengedett ütemterveknek* nevezzük. A legszélesebb körben vizsgált optimalizálási kritérium a maximális befejezési idő minimalizálása, azaz $\min C_{\max}$, ahol $C_{\max}(S) = \max_{j \in \mathcal{J}} (S_j + p_j)$. A probléma $\mathcal{NP}$ -nehéz, megoldására számos egzakt és heurisztikus megoldási módszert dolgoztak ki az évek során, lásd pl. [29, 53].

Számos eredményt ismertetünk gépütemezési problémákkal kapcsolatban. A *gépek* olyan megújuló erőforrások, melyek kapacitása 1, és minden tevékenység 0 vagy 1 egységet igényel belőlük a végrehajtásához.

A dolgozatban egyrészt olyan gépütemezési, illetve általánosabb projekt ütemezési problémákat vizsgálunk, ahol az erőforrások különösen fontos szerepet kapnak akár a korlátozásokban, akár a célfüggvényben, másrészt olyanokat, ahol egymásnak alá-fölé rendelt problémákat kell megoldani. A sokféle probléma bemutatása, és elemzése mellett áttekintést adok az összetett ütemezési problémák megoldására szolgáló módszerekről is. A dolgozat 3 fő fejezetből áll:

- A 2. fejezetben vizsgált projekt, és gépütemezési problémákban a cél megújuló erőforrások egyenletes használata.
- A 3. fejezetben olyan gépütemezési problémákkal foglalkozom, ahol a tevékenységek a gépeken túl további nem megújuló erőforrásokat igényelnek, ill. állítanak elő.
- A 4. fejezetben egymással alá-fölé rendelő kapcsolatban álló ütemezési problémákkal kapcsolatos eredményeket mutatok be.

Minden fejezetben bemutatom a vizsgált problémakört, a legfontosabb korábbi eredményeket, és új eredményeket, amelyeknek szerzője, vagy társszerzője vagyok.

2. fejezet

Erőforrások egyenletes használata

Az *egyenletes erőforrás használat* (resource leveling) probléma az RCPSP egy olyan változata, amiben adott a tevékenységek maximális befejezési ideje, T , és egy olyan $\mathcal{S}$ megengedett ütemtervet keresünk, ami minimalizálja az erőforrás használat egy függvényét, azaz

$$\min \sum_{i \in \mathcal{R}} f_i(A_i^{\mathcal{S}}) \quad (2.1)$$

feltételek: (1.1), (1.2), $0 \leq S_j \leq T - p_j$, $j \in \mathcal{J}$,

ahol az $A_i^{\mathcal{S}} : [0, T] \rightarrow \mathbb{Q}$ leképezések az erőforrás használatot adják meg, azaz $A_i^{\mathcal{S}}(t) := \sum_{j \in \mathcal{J}, S_j \leq t < S_j + p_j} a_{ij}$, és $f_i(\cdot)$ egy valós számokon értelmezett valós értékű függvény [43].

Az f_i függvények számos alakját vizsgálták eddig a szakirodalomban, ezek közül talán a legismertebb a két következő:

$$f_{\text{lin}}(A_i^{\mathcal{S}}) := \int_0^T w_{it} \max\{0, A_i^{\mathcal{S}}(t) - L_i\} dt, \quad (2.2)$$

azaz minimalizálni kell az L_i küszöb feletti használatot az $i \in \mathcal{R}$ erőforrásból, illetve

$$f_{\text{quad}}(A_i^{\mathcal{S}}) := \int_0^T w_{it} (A_i^{\mathcal{S}}(t) - L_i)^2 dt, \quad (2.3)$$

azaz minimalizálni kell az L_i küszöbtől való négyzetes eltérést az $i \in \mathcal{R}$ erőforrásból. Mindkét célfüggvényben a w_{it} súlyok nemnegatív racionális számok. Neumann és Zimmermann [44] egzakt algoritmusokat dolgoztak ki a problémára különféle célfüggvények mellett mind az erőforrás korlátos, mind pedig az (1.1) elhagyásával nyert változatra.

A dolgozat 2.1. szakaszában az egyenletes erőforrás használat problémát a (2.2) célfüggvény mellett vizsgálom, de szemben az RCPSP problémával, a tevékenységek intenzitása időegységről-időegységre változhat, és ezzel egyenes

arányban változik erőforrás igényük is a végrehajtásuk során. Az ilyen tevékenységeket *változó intenzitású tevékenységeknek* nevezzük. Megmutatom, hogy a probléma erősen $\mathcal{NP}$ -nehéz, és egy vegyes-egészértékű matematikai programmal modellezem. Elemezni fogom a megengedett egész megoldások konvex burkát, és új, érvényes egyenlőtlenségeket vezetek le, amelyekről megmutatom, hogy a probléma egy relaxációjában a konvex burok lapjait határozzák meg. Egy polinomiális idejű, egzakt szeparációs eljárás ismertetése után számítási eredményekkel igazolom, hogy az új érvényes egyenlőtlenségek segítik a probléma optimális megoldását.

Ennek a problémának egy további általánosítását nyerjük, ha a megengedjük, hogy a megelőzési relációban álló tevékenységek részben átfedjék egymást, és ezen kívül a megelőző tevékenységek "etetik" a rákövetkező tevékenységeket. Az ilyen megelőzési relációt "etető precedencia relációnak" nevezzük. Egy ilyen relációban minden $(j, k) \in \mathcal{E}$ párhoz tartozik egy $0 \leq \phi_{jk} \leq 1$ paraméter, ami azt fejezi ki, hogy a j tevékenységnek legalább milyen mértékben kell elkészülnie ahhoz, hogy k elkezdődhessen. Továbbá az "etetés" azt jelenti, hogy minden t időegység végéig j tevékenységnek legalább akkora hányada készül el, mint a k -nak. Ezekkel a korlátokkal például az írható le, hogy a j tevékenység végrehajtása során folyamatosan létrejövő félkész terméket a k tevékenység használja fel. A korábban vizsgált átfedést nem engedő megelőzési korlátok a $\phi_{jk} = 1$ paraméterrel modellezhetők. Az azzal kapcsolatos eredmények általánosíthatóak az "etető" megelőzési korlátokra, és az eredményeket a 2.2. szakasz ismerteti.

A 2.3. szakaszban egy olyan géptütemezési problémát fogok vizsgálni, ahol adott m gép, és minden feladat előre géphez van rendelve, továbbá van néhány további megújuló erőforrás, amelyeket a feladatok igényelhetnek a végrehajtásukhoz. A célfüggvény az időegységenként lekötött erőforrások mennyiségének egy $f(\cdot)$ függvényét minimalizálja, csak úgy, mint (2.1). A (2.1) célfüggvényben előforduló f_i függvényekre csak annyi megszorítást teszünk, hogy $f_i(x+z, y) = f_i(x, y-z)$ teljesüljön. Néhány jól ismert függvény: $f_i(x, y) = w_i \max\{0, x-y\}$, vagy $f_i(x, y) = w_i(x-y)^2$. Ennek a problémának a számítási bonyolultságát fogom vizsgálni, meghatározom az egyik legszűkebb $\mathcal{NP}$ -nehéz problémát, illetve a legtagabb polinomiális időben megoldható speciális esetet. Utóbbi gyakorlati haszna, hogy jól használható egy egzakt algoritmusban jó minőségű megengedett megoldások kiszámítására.

A három szakasz alapjául rendre a Kis [35], Kis [36], és a Drótos & Kis [14] cikkek szolgálnak.

2.1. Egyenletes erőforrás használat változó intenzitású tevékenységekkel

A változó intenzitású projekt ütemezési problémák egyik legkorábbi változatát Węglarz [52] definiálta. A modelljében egyetlen, kétszeresen korlátozott, folytonosan osztható erőforrást kell tevékenységekhez rendelni úgy, hogy a projekt befejezési ideje a legkisebb legyen. Az erőforrás tetszőlegesen osztható szét a tevékenységek között, a szétosztás aránya tetszőlegesen változhat, de minden időpillanatban a szétosztható mennyiség korlátos, és az összes elérhető mennyiség is az. A tevékenységek végrehajtási sebessége, intenzitása, a hozzájuk rendelt erőforrás mennyiségének nem-csökkenő függvénye. Węglarz azt vizsgálta, hogy milyen feltételek mellett létezik a feladatnak megoldása, illetve több speciális esetben (amikor a megelőzési reláció üres) analitikusan megadta az optimális megoldást.

Leachman és szerzőtársai [39] egy lineáris változatát vizsgálták Węglarz problémájának diszkrét időtengelyen, ahol a tevékenységek intenzitása időegységről-időegységre változhat, és az erőforrás igény a tevékenység intenzitásával egyenes arányos. Azaz, ha $\mathbf{x}_{jt}$ jelöli a tevékenység intenzitását a t időegységben, akkor igénye az $i \in \mathcal{R}$ erőforrásból $a_{ij}\mathbf{x}_{jt}$, ahol a_{ij} konstans. A lineáris erőforrás használat miatt feltehető, hogy $\sum_{t=0}^{C_{\max}} \mathbf{x}_{jt} = 1$ minden $j \in \mathcal{J}$ tevékenység esetén, azaz a tevékenység akkor van kész, ha az időegységenkénti intenzitások összege eléri az 1-et. A szerzők egy heurisztikus megoldási algoritmust prezentáltak a maximális befejezési idő minimalizálására. Tavares [49] ugyanazt a problémát vizsgálta, mint Leachman et al. [39], és egy nemlineáris programmal modellezte a problémát, és egy általános nemlineáris megoldóval oldott meg néhány feladatpéldányt.

A változó intenzitású projekt ütemezési problémák sorában a következő állomás Hans [28] dolgozata, amelyben a tevékenységeket egy adott időintervallumba kell beütemezni, közöttük megelőzési korlátok teljesülnek, és a célfüggvény az erőforrás túlzott használatát méri. Az időtengely, [39] modelljéhez hasonlóan, egységnyi hosszú periódusokra van osztva. Minden $j \in \mathcal{J}$ tevékenységhez tartozik egy maximális $h_j \leq 1$ intenzitás, és egy r_j legkorábbi kezdési, valamint d_j legkésőbbi befejezési periódus, és a $j \in \mathcal{J}$ tevékenységet teljes egészében az $[r_j, d_j]$ intervallumban kell végrehajtani. Legyen $T = [\min_{j \in \mathcal{J}} r_j, \max_{j \in \mathcal{J}} d_j]$ azoknak a periódusoknak a halmaza, ahova a tevékenységek beütemezhetőek. Ha $\mathbf{x}_{jt}$ jelenti a j tevékenység intenzitását a t

periódusban, akkor

$$\sum_{t=r_j}^{d_j} \mathbf{x}_{jt} = 1, \quad (2.4)$$

$$\mathbf{x}_{jt} \leq h_j, \quad \forall t \in [r_j, d_j] \quad (2.5)$$

feltételeknek kell teljesülni. Jelölje b_{it} az $i \in \mathcal{R}$ erőforrás kapacitását a t időegységben, és legyenek $\mathbf{y}_{it}$, $i \in \mathcal{R}$, $t \in T$, új változók, ekkor

$$\sum_{j \in J} a_{ij} \mathbf{x}_{jt} \leq b_{it} + \mathbf{y}_{it}, \quad \forall t \in T \quad (2.6)$$

A célfüggvény pedig

$$\min \sum_{i \in \mathcal{R}, t \in T} c_{it} \mathbf{y}_{it}. \quad (2.7)$$

Hans a megelőzési korlátokat úgy modellezte, hogy feltette, hogy a tevékenységek halmaza projektekre bontható, ahol egy projekt a megelőzési reláció szerint összefüggő tevékenységekből áll. Ezek után minden projekthez annyi új bináris változót vett fel, ahányféle módon a projekt tevékenységei a megelőzési, és a (2.4)-(2.6) korlátok betarthatják (egész pontosan a megengedett $\mathbf{x}_j$ értékek szupportjai halmaza adja az oszlopokat). Mivel a lehetőségek száma exponenciálisan sok lehet a feladat méretében, ezért Hans egy szétválasztás-és-árazás (branch-and-price) módszert dolgozott ki a probléma megoldására. A továbbiakban a Hans által vizsgált ütemezési problémát RCPSVP-nek fogom rövidíteni.

Új eredmények

Számítási bonyolultság. A feladat számítási bonyolultságának meghatározásához szükség van a PREEMPTIVE FLOW SHOP probléma (PFSP) definíciójára. A PFSP inputja a feladatok, és a gépek száma, m , és n , valamint további mn nemnegatív egész szám, p_{ij} , $1 \leq i \leq m$, $1 \leq j \leq n$. Az j feladatot az i gépen összesen p_{ij} ideig kell feldolgozni, de a feldolgozás megszüítható tetszőleges egész időpontban. Minden feladatot egyszerre csak egy gépen lehet feldolgozni, és minden gép egyszerre csak egy feladaton dolgozhat. A feladatoknak a gépeket az $1, \dots, m$ sorrendben kell végiglátogatni, azaz először az 1-es gépen kell őket feldolgozni, majd a 2-es gépen, stb. Cél a maximális feladat befejezési idő minimalizálása az utolsó (m -edik) gépen.

1. Tétel. *A PFSP problémáról van egy pseudo-polinomiális transzformáció az RCPSVP problémára.¹*

¹A bonyolultságelméleti fogalmakban Garey and Johnson [18] könyvét követem.

2.1. VÁLTOZÓ INTENZITÁSÚ TEVÉKENYSÉGEK

7

Mivel a PFSP erősen $\mathcal{NP}$ -nehéz, egyből kapjuk, hogy

1. Következmény. *Az RCPSVP probléma erősen $\mathcal{NP}$ -nehéz.*

Ez az eredmény kizárja azt, hogy pseudo-polinomiális időben megoldjuk az RCPSVP-t, ha $\mathcal{P} \neq \mathcal{NP}$.

Modellezés. A probléma modellezéséhez a (2.4)-(2.6) korlátokon túl szükség van további egyenlőtlenségekre a feladatok közötti megelőzési reláció miatt. Először is bevezetünk új változókat. Az $\mathbf{z}_{jt}$, $j \in \mathcal{J}, t \in [r_j, d_j]$, bináris változók azt fejezik ki, hogy a j feladat még nem fejeződött be, azaz $\mathbf{z}_{jt} = 1$, ha $\sum_{\tau=r_j}^t \mathbf{x}_{j\tau} < 1$. A definíció alapján nem veszítünk optimális megoldást azzal, ha előírjuk, hogy $\mathbf{z}_{jt} \geq \mathbf{z}_{j,t+1}$, $t \in [r_j, d_j - 1]$, minden $j \in \mathcal{J}$ tevékenység esetén. A teljes matematikai program a következő:

$$\min \sum_{i \in \mathcal{R}, t \in T} c_{it} \mathbf{y}_{it} \quad (2.8)$$

$$\sum_{t=r_j}^{d_j} \mathbf{x}_{jt} = 1, \quad \forall j \in \mathcal{J} \quad (2.9)$$

$$\mathbf{x}_{jt} - h_j \mathbf{z}_{jt} \leq 0, \quad \forall j \in \mathcal{J}, t \in [r_j, d_j] \quad (2.10)$$

$$\mathbf{x}_{kt} - h_k(1 - \mathbf{z}_{jt}) \leq 0, \quad \forall (j, k) \in \mathcal{E}, t \in [r_k, \min\{d_j, d_k\}] \quad (2.11)$$

$$\mathbf{z}_{jt} - \mathbf{z}_{j,t+1} \geq 0, \quad \forall j \in \mathcal{J}, t \in T \setminus [r_j, d_j - 1] \quad (2.12)$$

$$\sum_{j \in \mathcal{J}} a_{ij} \mathbf{x}_{jt} - \mathbf{y}_{it} \leq b_{it}, \quad \forall i \in \mathcal{R}, t \in T \quad (2.13)$$

$$\mathbf{z}_{jt} \in \{0, 1\}, \quad \forall j \in \mathcal{J}, t \in [r_j, d_j] \quad (2.14)$$

$$\mathbf{x} \geq 0, 0 \leq \mathbf{y} \leq \bar{\mathbf{b}}. \quad (2.15)$$

A megelőzési korlátokat a (2.10)-(2.12) együttesen fejezik ki, hiszen amíg a j feladat nem fejeződik be, addig a k nem kezdődhet el, minden $(j, k) \in \mathcal{E}$ feladatpárra. A fentiek alapján a (2.8)-(2.15) matematikai program megengedett megoldásai kölcsönösen egyértelműen megfeleltethetőek az RCPSVP megengedett megoldásainak. Tehát az RCPSVP-re optimális megoldást kaphatunk a fenti MIP megoldásával. Ezzel azonban nem elégszünk meg, hanem előfeldolgozással, és vágásokkal tovább erősítjük a feladat LP relaxációját. A vágásokat a $(j, k) \in \mathcal{E}$ feladatpárhoz tartozó (2.9)-(2.12) egyenlőtlenségekből vezetjük le.

Előfeldolgozás. Vegyük észre, hogy ha a $j \in \mathcal{J}$ feladat maximális intenzitása $h_j \leq 1$, akkor legalább $p_j = \lceil 1/h_j \rceil$ időegységig tart. Ebből következik, hogy $(j, k) \in \mathcal{E}$ esetén, ha $r_k < r_j + p_j$, akkor r_k növelhető $r_j + p_j$ -re, és ha $d_j > d_k - p_k$, akkor d_j csökkenthető $d_k - p_k$ -ra. Ezt addig ismételhetjük $\mathcal{E}$

elemei fölött, amíg nincs további változás az r_j , és d_j értékekben. Továbbá a modell tovább erősíthető az $\mathbf{x}_{j,z} = 1$ változó rögzítésekkel, $t \in [r_j, r_j + p_1 - 1]$. További észrevétel, hogy $(j, k) \in \mathcal{E}$ esetén a k tevékenység intenzitása a t időegységben csak akkor lehet pozitív, ha $z_{kt} = 1$ és $z_{it} = 0$. Tehát a $(j, k) \in \mathcal{E}$ párra (2.9)-(2.10) kicserélhető a következőre:

$$\mathbf{x}_{kt} \leq \begin{cases} h_k(1 - z_{jt}), & \forall t \in [r_k, \min\{r_k + p_k - 1, d_j\}] \\ h_k(z_{kt} - z_{jt}), & \forall t \in [r_k + p_k, d_j] \\ h_k z_{kt}, & \forall t \in \{\max\{d_j + 1, r_k + p_k\}, d_k\} \end{cases} \quad (2.16)$$

Vágások. Tekintsünk egy $(j, k) \in \mathcal{E}$ feladatpárt. Legyen K^{jk} a konvex burka azoknak az a megengedett intenzitás-hozzárendeléseknek a j és a k feladatokhoz, amelyekben j megelőzi k -t. Azaz, K^{jk} azoknak a $(x_j, x_k, z_j) \in \mathbb{R}^{s_j} \times \mathbb{R}^{s_k} \times \{0, 1\}^{s_j - p_j}$, ahol $s_j = d_j - r_j + 1$, és $s_k = d_k - r_k + 1$, pontoknak a konvex burka, amelyek teljesítik a következő lineáris feltételeket:

$$\sum_{t=r_j}^{d_j} x_{jt} = 1, \quad (2.17a)$$

$$0 \leq \mathbf{x}_{jt} \leq h_j, \quad t \in [r_j, r_j + p_j - 1] \quad (2.17b)$$

$$0 \leq \mathbf{x}_{jt} \leq h_j \cdot z_{jt}, \quad t \in [r_j + p_j, d_j] \quad (2.17c)$$

$$z_{jt} \geq z_{j,t+1}, \quad t \in [r_j + p_j, r_k - 1] \quad (2.17d)$$

$$z_{jt} \geq z_{j,t+1}, \quad t \in [r_k, d_j - 1] \quad (2.17e)$$

$$\sum_{t=r_k}^{d_k} \mathbf{x}_{kt} = 1, \quad (2.17f)$$

$$0 \leq \mathbf{x}_{kt} \leq h_k \cdot (1 - z_{jt}), \quad t \in [r_k, d_j] \quad (2.17g)$$

$$0 \leq \mathbf{x}_{kt} \leq h_k, \quad t \in [d_j + 1, d_k] \quad (2.17h)$$

Célunk a K^{jk} poliéder leírása lineáris egyenlőtlenségekkel. Először is bevezetünk két másik poliédert:

$$K^{j*} = \text{conv} \{ (x_j, z_j) \in \mathbb{R}^{s_j} \times \{0, 1\}^{s_j - p_j} \mid (x_j, z_j) \text{ teljesíti a (2.17a)-(2.17e)-t} \},$$

$$K^{*k} = \text{conv} \{ (x_k, \tilde{z}_k) \in \mathbb{R}^{s_k} \times \{0, 1\}^{d_j - r_k + 1} \mid (x_k, \tilde{z}_k) \text{ teljesíti a (2.17e)-(2.17h)-t} \}.$$

1. Lemma. *Legyen (x_j, x_k, z_j) tetszőleges vektor $\mathbb{R}^{s_j} \times \mathbb{R}^{s_k} \times \mathbb{R}^{s_j - p_j}$ -ben. Ekkor $(x_j, x_k, z_j) \in K^{jk}$ akkor és csak akkor, ha $(x_j, z_k) \in K^{j*}$ és $(x_k, \tilde{z}_j) \in K^{*k}$, ahol $\tilde{z}_{jt} = z_{jt}$, $\forall t \in [r_k, d_j]$.*

A lemma alapján elegendő a K^{j*} , és a K^{*k} poliéderekre egy-egy lineáris reprezentációt találni. Szerencsére mindkettő megkapható az alábbi K poliéder megfelelő paraméterezésével. Legyen K azoknak a $(x, z) \in \mathbb{R}^n \times \{0, 1\}^m$

2.1. VÁLTOZÓ INTENZITÁSÚ TEVÉKENYSÉGEK

9

pontoknak a konvex burka, melyek teljesítik a következő feltételeket:

$$\sum_{t=1}^n \mathbf{x}_t = 1 \quad (2.18a)$$

$$\mathbf{x}_t \leq h \cdot (1 - \mathbf{z}_t), \quad t \in [1, m] \quad (2.18b)$$

$$\mathbf{x}_t \leq h, \quad t \in [m+1, n] \quad (2.18c)$$

$$\mathbf{z}_t \geq \mathbf{z}_{t+1}, \quad t \in [1, m-1] \quad (2.18d)$$

$$\mathbf{x}_t \geq 0, \quad t \in [1, n]. \quad (2.18e)$$

Ebből megkapjuk a K^{j*} poliédert, ha $m = s_j - p_j$, $n = s_j$, $h = h_j$, $\mathbf{z}_t = 1 - \mathbf{z}_{j, di-t+1}$, $t \in [1, m]$, és $\mathbf{x}_t = \mathbf{x}_{d_j-t+1}^i$, $t \in [1, n]$. A K^{*k} poliédert pedig a következő helyettesítés adja: $m = d_j - r_k + 1$, $n = s_k$, $h = h_k$, $\mathbf{z}_t = \tilde{\mathbf{z}}_{k, t+r_k-1}$, $t \in [1, m]$, és $\mathbf{x}_t = \mathbf{x}_{k, t+r_k-1}$, $t \in [1, n]$. Nincs más hátra, mint a K poliéder leírása lineáris egyenlőtlenségekkel. Legyen $p = \lceil 1/h \rceil$, és $h_{\text{rem}} = 1 - (p-1)h$. Mivel $(n-m)h \geq 1$ definíció szerint, $m+p \leq n$.

Világos, hogy a (2.18a) egyenlet, és a (2.18b)-(2.18e) egyenlőtlenségek érvényesek K -ra, csak úgy, mint a

$$z_m \geq 0 \quad (2.18f)$$

egyenlőtlenség. A konvex burok leírásához szükség van még a következő egyenlőtlenségekre is.

2. Lemma. Legyen $\emptyset \neq S_1 \subseteq [1, m]$, és $S_2 \subseteq [m+1, n]$ olyanok, hogy $|S_1| + |S_2| = p$, és legyen t_1 a legkisebb eleme S_1 -nek. Ekkor az (S_1, S_2) egyenlőtlenség

$$h_{\text{rem}} z_{t_1} + h \sum_{t \in S_1 - \{t_1\}} z_t \leq \sum_{t \in [t_1, n] - (S_1 \cup S_2)} x_t \quad (2.18g)$$

érvényes a K poliéderre.

Legyen $P \subseteq \mathbb{R}^n \times \mathbb{R}^m$ azoknak a nemnegatív (x, z) vektoroknak a halmaza, amelyek teljesítik a (2.18a)-(2.18g) egyenlőtlenségeket.

2. Tétel. $P = K$.

Kérdés, hogy a K fenti leírásában vannak-e fölösleges egyenlőtlenségek. Erre ad választ a következő állítás.

3. Tétel. A (2.18b), (2.18d), (2.18f) egyenlőtlenségek mindig K egy lapját definiálják. A (2.18c) lapot definiál pontosan akkor, ha $h < 1$. Az $1 \leq t \leq n$ indexekre (2.18e)_t lapot definiál pontosan akkor, ha $h > 1/(n-1)$ és $t \geq m+1$, $p < n-m$. Végül, $\forall \emptyset \neq S_1 \subseteq [1, m]$, $S_2 \subseteq [m+1, n]$, ahol $|S_1| + |S_2| = p$, a megfelelő (2.18g) lapot definiál, kivéve, ha $t_1 = 1$ és $a = 1/p$.

1. Propozíció. *A (2.18g) egyenlőtlenségek $O(n \log n)$ időben szeparálhatóak.*

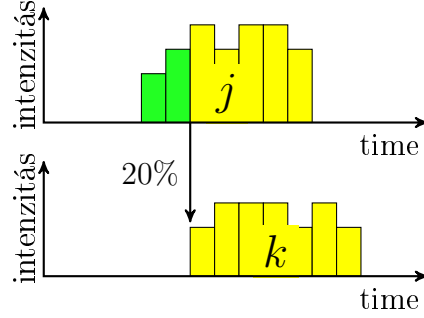
Megjegyezzük, hogy a szeparációhoz minimális $s - t$ vágást kell találni egy alkalmasan definiált hálózatban, és a hálózat struktúráját kihasználva lehet $O(n \log n)$ időben megoldani a problémát.

Számítógépes tesztelés. A fenti eredmények alapján implementáltam egy korlátozás-és-vágás típusú eljárást C++ programozási nyelven. Az implementációban felhasználtam az ILOG CPLEX 7.5 programcsomagot a probléma modellezésére, és megoldására. A szeparációs eljárást megvalósító függvényt be lehetett állítani a megoldó számára, hogy hívja meg a keresőfa bejárása során, miután a csúcsokban megtalálta az optimális LP megoldást. Az algoritmust különböző feladathalmazokon teszteltem. Az egyik feladatcsoport a Hans [28] dolgozatában szereplő, külső erőforrás használatot minimalizáló feladatok alkották. Ezen a feladatcsoporton a vágósíkos eljárás a (2.18g) egyenlőtlenségek szeparálásával az esetek döntő többségében legalább olyan jó, vagy jobb megoldást adott, mint Hans korlátozás-és-árazás algoritmus korlátozott futási időn belül, valamint az én módszerem gyakrabban talált optimális megoldást. A korlátozott futási idő azt jelenti, hogy mind Hans programja, mind a saját programom le volt állítva egy adott időkorlát túllépése után, és az addig talált legjobb megoldásokat vetettem össze feladatpéldányonként. A másik feladatcsoportot Tavares [49] problémái alkották, ahol a célfüggvény a maximális projekt befejezési idő minimalizálása, miközben nincs külső erőforrás, tehát a modellben $y = 0$. Ezeket a feladatokat sikerült mind optimálisan megoldani másodpercek alatt, és azt is kiderült, hogy az összesen 10 feladatpéldány közül 5 esetben Tavares megoldása nem volt optimális.

2.2. Etető megelőzési korlátok

A projektütemezésben széleskörűen használt megelőzési korlátok tevékenységek kezdési és befejezési idejei között állítanak fel időbeli korlátozásokat. A Kis [36] könyvfejezetben bevezetem az „etető megelőzési korlátokat”, amelyek lényege, hogy (i) a megelőző tevékenység adott százaléku elkészülése után kezdődhet a rákövetkező tevékenység, és (ii) a megelőző tevékenységnek legalább akkora hányada készül el minden t időpontig bezárólag, mint a rákövetkezőnek. A hagyományos vége-eleje típusú megelőzési korlátok annak felelnek meg, hogy a megelőző tevékenység teljesen elkészül, mielőtt a rákövetkező elkezdődik. Egy etető megelőzési korlátot illusztrál a 2.1. ábra.

Az etető megelőzési korlátos probléma az RCPSVP általánosítása, ugyanazokkal a célfüggvényekkel vizsgálható. Az etető megelőzési korlátokat a $(j, k, \phi) \in \mathcal{E}$ hármasok írják le, ahol $j, k \in \mathcal{J}$, és $0 \leq \phi \leq 1$ azt adja meg,



2.1. ábra. Változó intenzitású tevékenységek etető megelőzési korláttal összekötve. Az j tevékenységnek legalább 20%-ban el kell készülni k kezdése előtt.

hogy a j tevékenységnek legalább mekkora hányada készüljön el a k indulása előtt. A modellezéshez a $\mathbf{z}$ változók helyett szükség van a $\mathbf{z}_{jt}^\phi$ változókra. $\mathbf{z}_{jt}^\phi = 1$ pontosan akkor, ha j tevékenységnek ϕ hányada elkészült a t időegységig, különben 0. Ekkor a megelőzési korlátok modellezhetők az

$$\sum_{t=r_j}^{\ell-1} \mathbf{x}_{jt} \geq \phi(1 - \mathbf{z}_{j\ell}^\phi), \quad \forall j \in \mathcal{J}, \phi \in F_j, \ell \in [r_j + p_j^\phi, d_j], \quad (2.19a)$$

$$\mathbf{x}_{kt} \leq h_k(1 - \mathbf{z}_{jt}^\phi), \quad \forall j \in \mathcal{J}, (j, k, \phi) \in \mathcal{E}, \quad (2.19b)$$

$$\mathbf{z}_{jt}^\phi \geq \mathbf{z}_{j,t+1}^\phi, \quad \forall j \in \mathcal{J}, \phi \in F_j, t \in [r_j + p_j^\phi, d_j - 1], \quad (2.19c)$$

$$\sum_{t=r_j}^{\ell} \mathbf{x}_{jt} \geq \sum_{t=r_k}^{\ell} \mathbf{x}_{kt}, \quad \forall (j, k, \phi) \in \mathcal{E}, \ell \in [\max\{r_j, r_k\}, \min\{d_j, d_k\}], \quad (2.19d)$$

egyenlőtlenség rendszerrel, ahol $F_j = \{\phi : \exists (j, k, \phi) \in \mathcal{E}\}$, és $p_j^\phi = \lceil \phi/h_j \rceil$.

A poliéderes eredmények is általánosíthatóak. A részletes leírás helyett itt csak a legfontosabb eltérést mutatjuk be. A K poliéder helyett a K^ϕ poliéderre van szükség, ami azoknak a $(x, z) \in \mathbb{R}^n \times \{0, 1\}^{m-p^\phi}$ vektoroknak

a konvex burka, amelyek teljesítik a következő egyenlőtlenség rendszert:

$$\sum_{t=1}^n \mathbf{x}_t = 1, \quad (2.20a)$$

$$\sum_{t=1}^{\ell-1} \mathbf{x}_t \geq \phi \cdot (1 - \mathbf{z}_\ell^\phi), \quad \forall \ell \in [p^\phi + 1, m], \quad (2.20b)$$

$$\sum_{t=1}^m \mathbf{x}_t \geq \phi, \quad (2.20c)$$

$$\mathbf{z}_t^\phi \geq \mathbf{z}_{t+1}^\phi, \quad \forall t \in [p^\phi + 1, m - 1], \quad (2.20d)$$

$$0 \leq \mathbf{x}_t \leq h, \quad \forall t \in [1, n], \quad (2.20e)$$

ahol $p^\phi = \lceil \phi/h \rceil$, és $p^\phi \leq m \leq n$. A K^ϕ poliéder konvex burkának is ismeretes egy lineáris leírása. Legyen $h_{\text{rem}} = \phi - (p^\phi - 1)h$.

4. Tétel. K^ϕ azoknak az $(x, z) \in \mathbb{R}^n \times \mathbb{R}^{m-p^\phi}$ vektoroknak a konvex burka, amelyek teljesítik a (2.20a), (2.20c), (2.20d), (2.20e) feltételeket, valamint a következőket:

$$\mathbf{z}_{p^\phi+1}^\phi \leq 1, \mathbf{z}_m \geq 0, \quad (2.21a)$$

$$h_{\text{rem}} \mathbf{z}_{t_1}^\phi + h \sum_{t \in S_1 \setminus \{t_1\}} \mathbf{z}_t^\phi + \sum_{t \in [1, t_1] \setminus (S_1 \cup S_2)} \mathbf{x}_t \geq \phi - h|S_2|, \quad (2.21b)$$

minden $\emptyset \neq S_1 \subseteq [p^\phi + 1, m]$, $S_2 \subset [1, p^\phi]$ és $|S_1| + |S_2| = p^\phi$, ahol t_1 az S_1 halmaz legnagyobb eleme;

$$h \sum_{t \in U} \mathbf{z}_t^\phi + \sum_{t \in [1, n] \setminus U} \mathbf{x}_t \geq \phi, \quad (2.21c)$$

minden $U \subseteq \{p^\phi + 1, \dots, m\}$ halmazra, amelyre $h|U| \leq 1$;

$$(\phi - 1 + h|U_\ell^2|) \mathbf{z}_\ell^\phi + h \sum_{t \in U_\ell^1} \mathbf{z}_t^\phi + \sum_{t \in [1, n] \setminus (U_\ell^1 \cup U_\ell^2)} \mathbf{x}_t \geq \phi, \quad (2.21d)$$

$\forall \ell \in [p^\phi + 1, m]$, $U_\ell^1 \subseteq [p^\phi + 1, \ell]$, $U_\ell^2 \subseteq [\ell + 1, n]$, $h|U_\ell^1 \cup U_\ell^2| \leq 1$, és $h|U_\ell^2| \leq 1 - \phi$;

2. Propozíció. Az (2.21b)-(2.21d) egyenlőtlenségek polinomiális időben szeparálhatóak. Az (2.21b) egyenlőtlenségek $O(n \log n)$ időben szeparálhatóak, és (2.21c) egyenlőtlenségek $O(n)$ időben szeparálhatóak.

Implementáltam a vágósíkos eljárást, és a futási eredmények alapján a vágósíkok csökkentik az optimális megoldás megtalálásához szükséges keresőfa méretét, illetve erősítik az alsó/felső korlátokat számos feladatosztály esetén. Viszont az új vágósíkok nem tűnnek annyira hasznosnak, mint a 2.1. fejezetben vizsgált probléma esetén, aminek oka, hogy a megengedett átfedések miatt könnyebb jó megoldást találni.

2.3. Egyenletes erőforrás használat gépütemezési környezetben

A megújuló erőforrásokkal kiegészített gépütemezési problémák különféle speciális eseteit széleskörűen elemezték már a szakirodalomban, lásd pl. Blaze-wicz et al [4], Blazewicz et al [3]. A gépütemezési problémákkal kapcsolatos eredmények döntő többségében a kiegészítő megújuló erőforrások használatát korlátozzák, de azok egyenletes használatát nem optimalizálják, lásd pl. Kellerer & Strusevich [33, 34]. Ballestín et al [1] cikkében az erőforrások egyenletes használatát optimalizálják, de a feladatokat nem kell sorrendezni, nincsenek gépi, vagy más erőforrás korlátok. Caramia & Dell’Olmo [8] olyan ütemezési problémákat vizsgálnak, ahol minden feladat egyégnyi ideig tart, valamint a feladatok között van egy kompatibilitási reláció, és csak kompatibilis feladatokat lehet párhuzamosan végrehajtani. A cél, többek között, az erőforrások egyenletes használata, és a szerzők egy heurisztikus algoritmust ismertetnek a feladat megoldására. Végül Valls et al [51] egy több-célfüggvényű munkaerő ütemezési problémára ad egy heurisztikus algoritmust, ahol az egyik célfüggvény az átlagos munkaerő használattól való eltérés minimalizálása.

Új eredmények

A következő problémát fogjuk vizsgálni. Jelölje $\mathcal{J}$, $\mathcal{M}$, és $\mathcal{R}$ feladatok, gépek, és megújuló erőforrások véges halmazait. Minden $j \in \mathcal{J}$ feladat egy $m_j \in \mathcal{M}$ géphez van rendelve, végrehajtási ideje p_j , az $[r_j, d_j]$ időintervallumba kell beütemezni, és a_{ij} egységet foglal le az $i \in \mathcal{R}$ erőforrás típusból a végrehajtása során. A feladatok nem megszakíthatóak, és egy gép egyidejűleg csak egy feladaton tud dolgozni. Tekintsük a probléma következő speciális esetét. Minden gépen rögzített a feladatok sorrendje, kivéve az elsőt. Határozzunk meg egy olyan feladat sorrendet az első gépen, ami minimalizálja a (2.1) célfüggvényt, és az f_i függvények lineárisak (2.2) vagy kvadratikusak (2.3). Jelölje SMRLP ezt a problémát.

5. Tétel. *Az SMRLP probléma erősen $\mathcal{NP}$ -nehéz mind a (2.2), mind pedig a (2.3) célfüggvényre nézve.*

Ugyanakkor ha megkötjük az első gépen a feladatok sorrendjét, azaz csak a kezdési időket kell meghatározni betartva az előre megadott sorrendet, akkor a probléma kezelhetővé válik.

6. Tétel. *Az SMRLP probléma polinomiális időben megoldható, ha az első gépen a feladatok sorrendje rögzített.*

Ez az eredmény azért hasznos, mert segítségével hatékonyabban lehet megoldani az általános problémát is, ahol egyik gépen sem rögzített a sorrend. A Drótos & Kis [14] cikkben egy hatékony egzakt módszert mutatunk be az m -gépes egyenletes erőforrás használat problémára, ahol a rögzített sorrend melletti egygépes probléma ismételt megoldásán túl számos más algoritmikus technika is alkalmazásra kerül.

3. fejezet

Gépütemezés nem megújuló erőforrásokkal

A gépütemezés nem megújuló erőforrásokkal egy eredményekben egyre gazdagodó terület. A probléma egyik legegyszerűbb változata a következő. Egyetlen gép van, amin feladatok egy véges $\mathcal{J}$ halmazát kell beütemezni, a $j \in \mathcal{J}$ feldolgozási ideje $p_j > 0$ egész szám. Továbbá van néhány nem-megújuló erőforrás, $\mathcal{R}$, és néhány beszállítási időpont $0 = u_1 < u_2 < \dots < u_q$, a hozzánk tartozó $\tilde{b}_\ell \in \mathbb{Z}_{\geq 0}^{\mathcal{R}}$ beszállítási mennyiségekkel együtt. A $j \in \mathcal{J}$ feladat erőforrásigénye az $i \in \mathcal{R}$ erőforrás típusból $a_{ij} \geq 0$ egész szám. A feladatok indulásukkor csökkentik az erőforrások készletét, azaz a $j \in \mathcal{J}$ feladat indulásának pillanatában az $i \in \mathcal{R}$ erőforrás készlete a_{ij} mennyiséggel csökken. Az $i \in \mathcal{R}$ készlete növekszik az u_ℓ időpontokban az $\tilde{b}_{i\ell}$ mennyiségekkel. Negatív készlet nem megengedett, ami korlátozza az adott időpontban indítható feladatok körét. Egy S ütemterv minden feladatnak megadja a kezdési idejét. Akkor megengedett, ha (i) $S_j \geq 0$, $j \in \mathcal{J}$, (ii) minden $j \neq k$ feladat párra $S_j + p_j \leq S_k$ vagy $S_k + p_k \leq S_j$ (nem fedik át egymást időben), és (iii) minden $\ell \in [1, q]$ beszállításra teljesül, hogy az $u_{\ell+1}$ előtt kezdődő feladatok összisége nem haladja meg az első ℓ beszállítás össz mennyiségét, azaz $\sum_{j \in \mathcal{J}: S_j < u_{\ell+1}} a_{ij} \leq \sum_{\ell'=1}^{\ell} \tilde{b}_{i\ell'}$, $i \in \mathcal{R}$ ($u_{q+1} = \infty$). A célfüggvény lehet például a maximális befejezési idő, $C_{\max}$, vagy a súlyozott befejezési idők összege, $\sum w_j C_j$. A problémafelvetés, és az első bonyolultsági eredmények Carlier [10], és Carlier és Rinnooy Kan [9] szerzőkhöz fűződnek. Az egygépes, r erőforrást, és q beszállítást tartalmazó problémát a $C_{\max}$ célfüggvény mellett MCP_q^r -val is jelöljük a továbbiakban. Egy másik jelölés a $1|nr = r, q|C_{\max}$, lásd a 5.1 szakaszt.

Egy másik, látszólag független probléma, a következő. Ismét egyetlen gépünk van, és félkész termék típusok egy $\mathcal{R}$ halmaza, amelyeket a gépre ütemezendő $\mathcal{J}$ feladathalmaz elemei állítanak elő. Minden $j \in \mathcal{J}$ feladat-

nak van egy p_j hossza, és az i -típusú félkész termékből a_{ij} mennyiséget állít elő. Ezen túl adott néhány határidő, $0 < u_1 < u_2 < \dots < u_q$, és igényelt mennyiség $\tilde{b}_1, \dots, \tilde{b}_q$. Azt ℓ -edik határidőig az összes igény az $i \in \mathcal{R}$ félkész termékből $b_{i\ell} := \sum_{\tau=1}^{\ell} \tilde{b}_{i\tau}$. Egy S ütemterv meghatározza a feladatok kezdési idejét. Azt mondjuk, hogy S ütemterv *betartja* az ℓ -edik határidőt, ha az u_ℓ határidő előtt befejeződő munkák legalább $b_{i\ell}$ mennyiségű félkész terméket állítanak elő minden egyes $i \in \mathcal{R}$ típusból, azaz $\sum_{j \in \mathcal{J}: S_j + p_j \leq u_\ell} a_j \geq b_{i\ell}$. Ha ez nem teljesül, akkor a késés mértéke az a legkisebb $\tilde{T}_\ell > 0$, melyre $\sum_{j \in \mathcal{J}: S_j + p_j \leq u_\ell + \tilde{T}_\ell} a_j \geq b_{i\ell}$. A cél olyan ütemterv meghatározása, amely minimalizálja a maximális késést, azaz $\min_S \tilde{T}_{\max}(S) = \min_S \max_\ell \tilde{T}_\ell(S)$. Ezt a problémát, legjobb tudomásom szerint, nem vizsgálták korábban a szakirodalomban, hanem csak egy rokon változatát, ahol nem megengedett a késés, és a határidőket a lehető legkisebb maximális készlettel kell teljesíteni [5]. Az egygépes, r féle terméket, és q szállítási időpontot tartalmazó problémát a $\tilde{T}_{\max}$ célfüggvény mellett DTP_q^r -val is jelöljük a továbbiakban. Egy másik jelölés ugyanerre a $1|dm = r, q|\tilde{T}_{\max}$. Mivel a $\tilde{T}_{\max}$ érték lehet 0 is, hogy approximálhassuk az optimum értékét, ezért definiáljuk a $T_{\max}^s := \tilde{T}_{\max} + u_q - u_1$ függvényt, ami egy pozitív konstanssal tolja el a $\tilde{T}_{\max}$ függvényt.

A dolgozat 3.1. szakaszában számos eredményt mutatok be az erőforrást fogyasztó, ill. előállító feladatok ütemezésével kapcsolatban, egyetlen gépet feltételezve. Többek között megmutatom, hogy a nem megújuló erőforrást fogyasztó probléma a maximális befejezési idő célfüggvénnyel ekvivalens a félkész terméket határidőre előállító problémával a maximális késés célfüggvény mellett. De ez csak egy eredmény a sok közül, és fő haszna, hogy az egyik problémaosztályra kapott eredmény átvihető a másikkra. A fő kérdés a különböző speciális esetek *számítási bonyolultsága*, és *approximálhatósága*. Az egyik legfontosabb eszköz az *approximációt megőrző redukció* lesz, segítségével megmutatom az ütemezési problémák kapcsolatát a hátizsák feladat különböző változataival. Továbbá a bonyolultabb ütemezési problémákhoz, részben a hátizsák feladatra ismert algoritmikus technikákat használva, adok approximációs algoritmusokat, vagy bizonyítom, hogy nem approximálhatóak tetszőleges pontossággal. Nem csak az egy-gép egy-erőforrásos esetet fogom vizsgálni, hanem a több gépes, több erőforrásos problémákat is, különféle célfüggvények mellett (3.2. szakasz).

Ez a fejezet a Drótos & Kis [15], Györgyi & Kis [23], Györgyi & Kis [24], Györgyi & Kis [25], Kis [37], Györgyi & Kis [26] cikkeken alapul.

3.1. Egygépes ütemezés nem megújuló erőforrásokkal

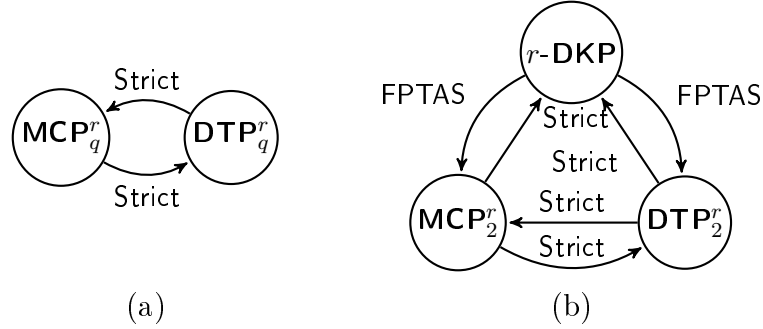
Nem megújuló erőforrásokkal kiegészített ütemezési problémákkal először Carlier [10], illetve Carlier & Rinnooy Kan [9] foglalkozott. A [10] dolgozatban a probléma definícióján túl számos bonyolultsági eredmény szerepel különféle speciális esetekre. Többek között Carlier bizonyította, hogy az $1|nr = 1, q = 2|C_{\max}$ probléma gyengén $\mathcal{NP}$ -nehéz, és a $1|nr = 1|C_{\max}$ probléma erősen $\mathcal{NP}$ -nehéz. A [9] cikkben a feladatok csak nem-megújuló erőforrásokat igényelnek, nincsenek gépek, amin a feladatokat sorrendezni kell. Slowinski [47] egy független-párhuzamos gépes modellt vizsgált, amiben a feladatok megszakíthatóak, és a nem-megújuló erőforrások felhasználása arányos a feldolgozás előrehaladtával. Ez a probléma lineáris programozással megoldható. Tokar et al [50] megmutatta, hogy az $1|nr = 1|C_{\max}$ probléma visszavezethető a kétgépes FLOW SHOP problémára, feltéve, hogy a nem-megújuló erőforrásból minden időegységben 1 egységnyi érkezik. Grigoriev et al [22] néhány további egygépes probléma komplexitását vizsgálta ($C_{\max}$ és $L_{\max}$ célfüggvények mellett), illetve 2-approximációs algoritmust adtak az $1|nr = 2, p_j = p|L_{\max}$ és az $1|nr|C_{\max}$ problémákra. Gafarov et al [17] további egygépes probléma variánsok komplexitását vizsgálta. Neumann & Schwindt [42] az RCPSP nem-megújuló erőforrásokkal való kiterjesztését vizsgálta, ahol a tevékenységek fogyaszthatnak, és elő is állíthatnak nem megújuló erőforrásokat, és egy egzakt algoritmust írtak le a $C_{\max}$ minimalizálására. Kellerer et al [31] megmutatta, hogy amennyiben csak egyetlen gép van, a probléma már akkor is $\mathcal{NP}$ -nehéz, és 3 különböző approximációs algoritmust adtak a maximális készlet minimalizálására, miközben a feladatok befejezési idejét korlátozták. Briskorn et al [5] termelő, és fogyasztó feladatokat is tartalmazó egygépes feladatok komplexitását elemezte, míg Briskorn [6] egy egzakt módszert írt le a $\sum w_j C_j$ célfüggvény minimalizálására.

Új eredmények

Az egygépes környezetben elért új eredmények három fő csoportra oszthatók: (i) approximációt megőrző redukciók a csak termelő feladatokat, vagy csak fogyasztó feladatokat tartalmazó ütemezési problémák, és az r -dimenziós hátizsák probléma között, (ii) approximációs algoritmusok, amelyek nem állnak elő redukciók segítségével, (iii) az approximáció nehézségével kapcsolatos eredmények. Az approximációt megőrző redukciókkal kapcsolatos releváns fogalmakat és eredményeket a 5.2. szakasz foglalja össze.

Emlékeztetőül, az r -dimenziós hátizsák probléma (r -DKP) inputjában adott n elem, mindegyiket egy r -dimenziós, nemnegatív egész számokból

álló a_j vektor, és egy $c_j \geq 0$ profit jellemez. Adott továbbá egy r -dimenzós kapacitás vektor, b . Egy olyan K részhalmazát keressük az elemeknek, hogy $\sum_{j \in K} a_{ij} \leq b_i$, minden $i = 1, \dots, r$ esetén, és $\sum_{j \in K} c_j$ a lehető legnagyobb értéket veszi fel ezen feltételek mellett. A hátizsák feladattal kapcsolatos számos eredményről ad kitűnő áttekintést Kellerer et al [32] könyve.



3.1. ábra. Approximációt megőrző redukciónak ütemezési problémák, és az r -dimenziós hátizsák probléma között.

Az (i) eredményeket a 3.1. ábra foglalja össze. Egy irányított él egy Π_1 problémáról egy Π_2 problémára azt jelenti, hogy a Π_1 redukálható Π_2 -re adott típusú redukciónal. A redukciónkat a 5.2. szakasz foglalja össze. Amint láthatjuk, a DTP_q^r és az MCP_q^r approximálhatósági szempontból egyformák, mivel az ábra (a) része szerint mindkét probléma visszavezethető a másikra egy szigorú ("Strict") redukció segítségével. Ez azt jelenti, hogy ha pl. az MCP_q^r problémára adunk egy approximációs algoritmust / bizonyítunk egy nem approximálhatósági eredményt, akkor a DTP_q^r problémára is lényegében ugyanazt az eredményt kapjuk. A szigorú redukciónk a DTP_q^r és az MCP_q^r problémák között a következő két állítás következményei:

3. Lemma. *Adott a DTP_q^r probléma egy példánya $I_D = \{n, q, (p_j, a_j)_{j=1}^n, (u_\ell, \tilde{b}_\ell)_{\ell=1}^q\}$. Defináljuk az MCP_q^r egy példányát $I_M = \{n, q, (p_j, a_j)_{j=1}^n, (u'_\ell, b'_\ell)_{\ell=1}^q\}$ a következő módon:*

$$\begin{aligned} u'_\ell &= u_q - u_{q+1-\ell} \\ b'_\ell &= \tilde{b}_{q+1-\ell} \end{aligned} \quad \ell = 1, \dots, q.$$

Ekkor, ha σ egy olyan sorrendje a munkáknak, amelynek maximális késése $T_{\max}^\sigma$ az I_D feladatpéldányon, akkor a σ sorrend megfordításával kapott ütemtervben az utolsó munka befejezési ideje $u_q + T_{\max}^\sigma$ az I_M feladatpéldányon.

4. Lemma. *Adott az MCP_q^r problémának egy $I_M = \{n, q, (p_j, a_j)_{j=1}^n, (u_\ell, \tilde{b}_\ell)_{\ell=1}^q\}$ példánya. Definíáljuk a DTP_q^r egy példányát $I_D = \{n, q, (p_j, a_j)_{j=1}^n, (u'_\ell, b'_\ell)_{\ell=1}^q\}$ a következő módon:*

$$\begin{aligned} u'_\ell &= u_q - u_{q+1-\ell} \\ b'_\ell &= b_{q+1-\ell} \end{aligned} \quad \ell = 1, \dots, q.$$

Ekkor, ha S egy olyan ütemterv az I_M feladatpéldánya, amelyben a munkák $C_{\max}^S$ időpontban fejeződnek be, akkor fordított sorrendben ütemezve a munkákat (várakozás nélkül) olyan ütemtervet kapunk az I_D feladatpéldányra, amiben a maximális késés legfeljebb $C_{\max}^S - u_q$.

7. Tétel. *Az MCP_q^r problémáról van szigorú approximációt megőrző redukció a DTP_q^r problémára, és fordítva, a DTP_q^r problémáról van szigorú approximációt megőrző redukció a MCP_q^r problémára.*

Az ábra (b) része szerint mind az MCP_2^r , mind pedig a DTP_2^r szigorú redukcióval visszavezethető az r -dimenziós hátizsák feladatra, azaz

8. Tétel. *Van egy szigorú approximációt megőrző redukció az MCP_2^r problémáról az r -DKP problémára.*

Mivel a szigorú approximációt megőrző redukciók kompozíciója is szigorú approximációt megőrző redukció, azonnal kapjuk, hogy

2. Következmény. *Van egy szigorú approximációt megőrző redukció a DTP_2^r problémáról az r -DKP problémára.*

Ennek következtében az r -dimenziós hátizsák feladatra ismert összes approximációs algoritmus bevezethető a két ütemezési probléma 2 periódusos változatának megoldására. Másrészt, mindkét ütemezési problémáról van FPTAS redukció az r -dimenziós hátizsák feladatra, azaz

9. Tétel. *Van egy FPTAS approximációt megőrző redukció az r -DKP problémáról a MCP_2^r problémára.*

Mivel az FPTAS approximációt megőrző és szigorú approximációt megőrző redukciók kompozíciója is FPTAS approximációt megőrző redukció, azonnal kapjuk, hogy

3. Következmény. *Van egy FPTAS approximációt megőrző redukció az r -DKP problémáról a DTP_2^r problémára.*

	Probléma		Eredmény	Forrás
1.	MCP_2^1	$1 nr = 1, q = 2 C_{\max}$	FPTAS	[23], [24]
2.	MCP_2^{const}	$1 nr = \text{const}, q = 2 C_{\max}$	$\nexists$ FPTAS ^a	[24]
3.	MCP_2^*	$1 nr, q = 2 C_{\max}$	$\nexists$ PTAS	[25]
4.	$MCP_{\text{const}}^{\text{const}}$	$1 nr = \text{const}, q = \text{const}, r_j C_{\max}$	PTAS ^b	[25]
5.		$1 nr = 1 \sum w_j C_j$	$\nexists$ FPTAS	[37]
6.		$1 nr = 1, q = 2 \sum w_j C_j$	FPTAS	[37]
7.	DTP_2^1	$1 dm = 1, q = 2 \tilde{T}_{\max}^s$	FPTAS	[15], [24]
8.	DTP_2^{const}	$1 dm = \text{const}, q = 2 \tilde{T}_{\max}^s$	$\nexists$ FPTAS ^c	[24]
9.	DTP_2^*	$1 dm, q = 2 \tilde{T}_{\max}^s$	$\nexists$ PTAS	[24], [25]
10.	$DTP_{\text{const}}^{\text{const}}$	$1 dm = \text{const}, q = \text{const} \tilde{T}_{\max}^s$	PTAS	[24], [25]

^aHa nr konstans és legalább 2

^bA feladatoknak lehet pozitív legkorábbi kezdési idejük

^cHa dm konstans, és legalább 2

3.1. táblázat. Approximációs sémák egygépes, nem-megújuló erőforrásos ütemezési problémákra.

Azonnal kapjuk, hogy konstans $r \geq 2$ esetén sem az MCP_2^r , sem pedig az DTP_2^r probléma nem rendelkezik teljesen polinomiális idejű approximációs sémával (FPTAS), mivel jól ismert tény, hogy a 2-dimenziós hátizsák problémára nincs FPTAS, ha $\mathcal{P} \neq \mathcal{NP}$, [19].

Az (ii) és (iii) eredményeket a 3.1. táblázat foglalja össze. Elegendő csak a nem-megújuló erőforrást fogyasztó feladatokat tartalmazó ütemezési problémákat tárgyalnunk ($C_{\max}$ célfüggvény esetén), mivel fentebb már megmutattuk, hogy approximálhatóság szempontjával ekvivalens a termelő feladatos problémával. A táblázat első két sorát már fentebb tárgyaltuk, a redukciók következményei. A harmadik sorban azt állítjuk, hogy az $1|nr, q = 2|C_{\max}$ problémára nincs PTAS, ha $\mathcal{P} \neq \mathcal{NP}$. Vegyük észre, hogy ebben a variánsban a nem-megújuló erőforrások száma része az inputnak, és csak 2 beszállítási időpont van. Ez a negatív eredmény a következő állítás következménye:

10. Tétel. *Van olyan $\varepsilon > 0$ konstans, hogy $\mathcal{NP}$ -nehéz a $1|nr, q = 2|C_{\max}$ probléma optimumát $(1 + \varepsilon)$ -nél kisebb hibával közelíteni.*

A táblázat negyedik sora szerint abban az esetben, ha az erőforrások

száma, és a beszállítási időpontok száma is konstans, akkor a $C_{\max}$ célfüggvényhez van PTAS, azaz

11. Tétel. *Az $1|nr = \text{const}, q = \text{const}|C_{\max}$ problémára van PTAS.*

Megjegyezzük, hogy a PTAS enumeráción, és egy lineáris program megoldásának kerekítésén alapul. Ugyanakkor konstruáltunk egy tisztán enumeráción alapuló PTAS-t is az $1|nr = 1, q = \text{const}|C_{\max}$ problémára.

A súlyozott befejezési idő minimalizálása egyetlen gépen, $1||\sum w_j C_j$, egy klasszikus, alaposan elemzett probléma a szakirodalomban. Jól ismert tény, hogy optimális megoldást kapunk, ha a munkákat w_j/p_j hányados szerint nem növekvő sorrendben ütemezzük [2, 7]. Ugyanakkor ha nem-megújuló erőforrásokkal egészítjük ki a problémát, akkor csak komplexitási eredmények voltak ismeretesek. Gafarov et al [17] megmutatta, hogy az $1|nr = 1||\sum w_j C_j$ probléma gyengén $\mathcal{NP}$ -nehéz. Ezzel szemben a következő, erősebb állítás is teljesül:

12. Tétel. *Az $1|nr = 1||\sum w_j C_j$ probléma erősen $\mathcal{NP}$ -nehéz.*

Ebből egyből következik a táblázat 5. sorának eredménye, azaz ha $\mathcal{P} \neq \mathcal{NP}$, akkor a problémára nincs FPTAS. Ugyanakkor az is belátható, hogy ha csak két beszállítási időpont van ($1|nr = 1, q = 2||\sum w_j C_j$), akkor a probléma gyengén $\mathcal{NP}$ -nehéz. Ebben az esetben konstruálható egy FPTAS, ahogy azt a táblázat 6. sora jelzi.

13. Tétel. *Az $1|nr = 1, q = 2||\sum w_j C_j$ problémára van FPTAS.*

3.2. Párhuzamos gépes ütemezés nem megújuló erőforrásokkal

A párhuzamos gépes ütemezési probléma a maximális befejezési idő célfüggvénnyel, de nem megújuló erőforrások nélkül, $P||C_{\max}$, igen alaposan vizsgált probléma a szakirodalomban. Mivel a probléma erősen NP-nehéz [18], számos approximációs algoritmus ismert a megoldására. Graham [20] megmutatta, hogy egy egyszerű ütemezési stratégia, ami a munkákat tetszőleges sorrendben, egymás után ütemezi, és a következő munkát arra a gépre teszi, ahol a legkorábban elkezdődhet, $(2 - 2/(m + 1))$ -approximációs algoritmus. Továbbá, ha a munkákat feldolgozási idő szerint nem növekvő sorrendben ütemezzük a fenti stratégiával, akkor $(4/3 - 3/m)$ -approximációs algoritmust kapunk. A következő fontos előrelépés Hall & Shmoys [27] polinómális idejű approximációs sémája (PTAS) az általánosabb $P|r_j|L_{\max}$ problémára. Megjegyezzük, hogy ha $\mathcal{P} \neq \mathcal{NP}$, akkor ennél jobb nem is várható, mivel a

$P||C_{\max}$ probléma erősen $\mathcal{NP}$ -nehéz. Amennyiben a gépek száma konstans, $Pm||C_{\max}$, akkor a probléma gyengén $\mathcal{NP}$ -nehéz, és ismeretes is rá teljesen polinomiális idejű approximációs séma [46].

A nem megújuló erőforrásokkal kiegészített $P|nr|C_{\max}$ problémára nem találtam a szakirodalomban approximálhatósági eredményeket.

Új eredmények

Kezdjük egy negatív eredménnyel. Amennyiben a gépek száma nem konstans, azaz része az inputnak, és legalább két nem megújuló erőforrás van, akkor belátható a következő:

14. Tétel. *Annak eldöntése, hogy van-e 2 hosszúságú ütemterv a tetszőlegesen sok gépet, 2 nem megújuló erőforrást, és egységnyi hosszú munkákat tartalmazó ütemezési problémára, $(P|nr = 2, q = 2, p_j = 1|C_{\max})$, $\mathcal{NP}$ -nehéz probléma.*

4. Következmény. *Tetszőlegesen kicsi $\varepsilon > 0$ esetén, a $P|nr = 2, q = 2, p_j = 1|C_{\max}$ problémát $(3/2 - \varepsilon)$ hibával approximálni $\mathcal{NP}$ -nehéz.*

5. Következmény. *Ha $\mathcal{P} \neq \mathcal{NP}$, akkor nincs polinomiális idejű approximációs séma a $P|nr = \text{const}|C_{\max}$ problémára.*

Jelölje $C_{\max}^{nr}$ a $P|nr|C_{\max}$ optimum értékét, és $C_{\max}^P$ a $P||C_{\max}$ optimum értékét ugyanazon a feladatpéldányon. Mivel u_q után minden munka bármikor ütemezhető, és emiatt $\max\{u_q, C_{\max}^P\} \leq C_{\max}^{nr} \leq u_q + C_{\max}^P$, ezért Hall & Shmoys [27] polinomiális idejű approximációs sémája $(2 + \varepsilon)$ -approximációs algoritmust ad a $P|nr|C_{\max}$ problémára tetszőleges rögzített $\varepsilon > 0$ esetén.

Amennyiben a gépek száma rögzített, akkor két pozitív eredmény is bizonyítható.

15. Tétel. *A $Pm|nr = \text{const}|C_{\max}$ problémához van polinomiális idejű approximációs séma.*

Az algoritmus, és a bizonyítás alapötlete hasonló az $1|nr = \text{const}, q = \text{const}|C_{\max}$ problémára adott PTAS-éhoz (11. Tétel). Valójában a párhuzamos gépes problémára adott PTAS speciális esetként megoldja az egygépes problémát is, sőt, a $q = \text{const}$ feltétel el is hagyható.

A következő eredmény már az $L_{\max}$ célfüggvénnyel kapcsolatos. Ha fel tesszük, hogy csak egy nem megújuló erőforrás van, és a feladatok időigénye arányos az erőforrás igényükkel (vagy fordítva), akkor a maximális késés célfüggvény is jól approximálható (persze $L_{\max}$ -ot meg kell növelni egy konstanssal, jelen esetben $u_q + \max d_j$ -vel, hogy approximálni lehessen).

16. Tétel. *A $Pm|nr = 1, p_j = a_j|L_{\max}$ problémához van polinomiális idejű approximációs séma.*

4. fejezet

Kétszintű ütemezési problémák

Az eddig fejezetekben vizsgált ütemezési problémák közös vonása, hogy egyetlen döntési szintet érintettek. Azonban a gyakorlatban szükség lehet egymással alá-fölé rendelt viszonyban álló, különböző, akár egymásnak ellentmondó célfüggvényekkel rendelkező döntéshozók ütemezési problémáinak megoldására is. Ehhez ad megfelelő elméleti háttérrel a kétszintű optimalizálás. A kétszintű optimalizálási problémákban két, egymásnak alá-fölé rendelt döntéshozó van, a "vezető", és a "követő". Mindkét döntéshozónak van egy optimalizálási problémája, saját döntési változókkal, korlátokkal, és célfüggvénnyel. A két probléma között a következő kapcsolatok állnak fenn: a "követő" korlátozásai, és/vagy célfüggvénye a vezető döntési változóinak függvénye, másrészt a vezető célfüggvényében a követő döntési változói is szerepelnek. Mindkét döntéshozó optimális megoldást keres. Először a vezető dönt, lerögzíti saját döntési változóit, majd a "követő" dönt, azonban az ő optimalizálási problémáját a "vezető" döntése is befolyásolja, lásd fentebb. Meghatároz egy optimális megoldást, és tudtára adja a vezetőnek, aki végül kiértékeli a saját célfüggvényét, aminek értékét a saját, és a követő döntési változói együttesen határoznak meg. Egy kétszintű optimalizálási probléma megoldása során a vezetőnek keresünk olyan optimalizálási algoritmust, ami megfelelő módon veszi figyelembe a követő lehetséges optimális megoldásait a vezető döntésének függvényében [13].

Fontos még megemlíteni, hogy a kétszintű optimalizálási problémáknak mindig két változata van a "követő" stratégiája szerint. Az *optimista* esetben a követő mindig olyan optimális megoldást választ (ha több is létezik adott paraméterezés esetén), ami a legkedvezőbb a "vezető" számára, míg a *pesszimista* esetben olyat, ami a legrosszabb kimenetelt adja a vezetőnek.

Általánosan elmondható, hogy a többszintű optimalizálási problémák nem ekvivalensek többcélú optimalizálási modellekkel [13], ugyanakkor Fülöp János [16] megmutatta, hogy a lineáris kétszintű optimalizálási probléma ek-

vivalens egy alkalmasan megválasztott többcélú optimalizálási problémával abban az esetben, ha a felső szint feltételhalmaza üres.

Az ütemezéselmélet fenti játékelméleti megközelítése nagyon kevés eredménnyel rendelkezik, a publikációk egy-egy esettanulmányhoz köthetők [30, 40]. Legjobb tudomásom szerint szisztematikus vizsgálatokra elemi ütemezési problémákon csak a jelen dolgozat alapjául szolgáló publikációkban került sor. Tekintsük például a következő párhuzamos gépes, kétszintű ütemezési problémát. A felső szint döntéshozója gépekhez akarja rendelni egy $\mathcal{J}$ feladathalmaz elemeit, míg az alsó szinten gépenként kell sorrendezni. A felső szinten a célfüggvény $\sum_j w_j^1 C_j$, míg az alsó szinten $\sum_j w_j^2 C_j$, tehát mindkét szinten a súlyozott befejezési időt kell minimalizálni, de eltérő súlyokkal. A felső szintnek úgy kell gépekhez rendelni a feladatokat, hogy az alsó szint (gépenkénti) optimális megoldása alapján kapott C_j feladat befejezési idők a lehető legjobb $\sum_j w_j^1 C_j$ értéket adják. Ezzel a problémával a dolgozat 4.1. fejezetében foglalkozom. Vizsgálni fogom a probléma komplexitását, kapcsolatát a többcélú optimalizálással, valamint a MAX k -CUT problémával (egy élsúlyozott, irányítatlan gráf csúcshalmazát kell k részre bontani úgy, hogy a részek közötti élek összsúlya minimális legyen). Több speciális esetre polinomiális idejű algoritmust vezetek le, ezek közül az egyik a MAX k -CUT probléma korábban nem vizsgált speciális esetére is alkalmazható. Egy NP-nehéz variánsra pedig, amelyben csak annyi megszorítás van, hogy a gépek száma konstans, egy teljesen polinomiális idejű approximációs sémát (FPTAS) adok a $Pm||\sum w_j C_j$ problémára ismert FPTAS általánosításával.

A 4.2. fejezet témája a kétszintű rendelés elfogadás probléma, ahol is egyetlen gép van, valamint egy feladat halmaz, és mindegyik feladatnak van egy határideje. A felső szint döntéshozója arról dönt, hogy mely feladatokat fogadja el, célja egy maximális összsúlyú feladathalmaz elfogadása. Az elfogadott feladatokat határidőket időre el kell végezni. Az alsó szint döntéshozója a gépen sorrendezi az elfogadott feladatokat a súlyozott befejezési idő célfüggvény mellett, saját súlyozása szerint. A felső szint döntéshozójának figyelembe kell venni az alsó szint optimális sorrendjét a feladatok elfogadásakor, mert a sorrend befolyásolja azt, hogy az elfogadott feladatok elkészülnek-e időre. Ezzel a problémával kapcsolatban vizsgálni fogom a számítási bonyolultságot, kapcsolatát a többcélú optimalizálással, valamint egy speciális esetben megmutatom, hogy az $1||\sum U_j$ problémára jól ismert Moore-Hodgson [41] algoritmus általánosítható a kétszintű rendelés-elfogadási problémára.

A fejezet eredményei a Kis & Kovács [38] cikken alapulnak.

4.1. A kétszintű súlyozott befejezési idő probléma

Az (egyszintű) egygépes súlyozott befejezési idő probléma, $1||\sum w_j C_j$, jól ismert a szakirodalomban, polinomiális időben megoldható a Smith hányados szabály segítségével [48]. A párhuzamos gépes változata, $P||\sum w_j C_j$, már erősen NP-nehéz (Jan Karel Lenstra nem publikál eredménye (Brucker [7])). A kétszintű változattal kapcsolatban nem volt ismeretes sem komplexitási, sem algoritmikus eredmény.

Új eredmények

Kezdjük egy bonyolultsági eredménnyel.

3. Propozíció. *A kétszintű, párhuzamos gépes, súlyozott befejezési idő probléma döntési változata erősen NP-nehéz optimista, és pesszimista esetben is.*

Most térjünk át a megoldások létezésére.

4. Propozíció. *Optimista esetben mindig van optimális megoldása a kétszintű súlyozott befejezési idő problémának. Pesszimista esetben csak akkor, ha nincs olyan j feladat, amelyre $w_j^2 = 0$ és $w_j^1 > 0$.*

A "követő" a gépekhez rendelt feladatokat optimista esetben a következő sorrend szerint dolgozza fel:

$$j \text{ megelőzi } k \text{ feladatot, ha } w_j^2/p_j > w_k^2/p_k \text{ vagy } (w_j^2/p_j^2 = w_k^2/p_k^2 \text{ és } w_j^1/p_j > w_k^1/p_k). \quad (4.1)$$

Pesszimista esetben a "követő" sorrendje

$$j \text{ megelőzi } k \text{ feladatot, ha } w_j^2/p_j > w_k^2/p_k \text{ vagy } (w_j^2/p_j^2 = w_k^2/p_k^2 \text{ és } w_j^1/p_j < w_k^1/p_k). \quad (4.2)$$

5. Lemma. *Optimista esetben mindig van egy állásidő mentes optimális ütemezés, amiben a gépenkénti optimális sorrendet a (4.1) határozza meg.*

6. Lemma. *Pesszimista esetben, ha a "követő" nem hozhat létre felesleges állásidőt, akkor a gépenkénti optimális sorrendet a (4.2) határozza meg.*

Szoros kapcsolat áll fenn a kétszintű optimalizálási probléma, és MAX k -CUT probléma között. Tekintsünk egy n pontú teljes gráfot, aminek a csúcsai az ütemezendő feladatokat reprezentálják. Optimista esetben sorrendezzük

a feladatokat (4.1)-szerint, pesszimista esetben (4.2)-szerint. Definiáljuk a következő élsúlyozást:

$$c(j, k) := p_j w_k^1, \text{ ha } j \text{ megelőzi } k \text{ feladatot a választott sorrendben.} \quad (4.3)$$

Ha $n < m$, akkor kevesebb, mint m gép is elegendő a feladatok ütemezésére, ezért a továbbiakban feltesszük, hogy $n \geq m$.

17. Tétel. *A MAX m -CUT probléma optimális megoldása a (4.3) élsúlyozás mellett a kétszintű, súlyozott befejezési idő probléma optimális megoldását adja, és fordítva.*

Ez az eredmény nem csak a probléma megoldására ad egy általános módszert, de lehetővé teszi egy speciális eset megoldását, illetve az ütemezési problémára kapott eredmények átvitelét a MAX m -CUT problémára.

Azt mondjuk, hogy w^1 és w^2 ugyanazt a feladatsorrendet határozza meg, ha $w_j^1/p_j \leq w_k^1/p_k$ pontosan akkor, ha $w_j^2/p_j \leq w_k^2/p_k$.

5. Propozíció. *Ha w^1 és w^2 ugyanazt a feladatsorrendet határozza meg, akkor a vezető optimális megoldását a $P || \sum_j w_j^1 C_j$ probléma optimális megoldása adja.*

A 17. Tétel alapján kapjuk, hogy

6. Következmény. *A MAX k -CUT probléma polinomiális időben megoldható egy élsúlyozott teljes gráfban, ha az élsúlyok $c(i, j) := \min\{p_i, p_j\}$, ahol p_i a csúcsokra írt nem-negatív súlyozás.*

Azt mondjuk, hogy w^2 a p_j szerint nem-csökkenő (nem-növekvő) sorrendet indukál, ha $w_j^2/p_j > w_k^2/p_k$ pontosan akkor, ha $p_j < p_k$ ($p_j > p_k$).

6. Propozíció. *Ha $w^1 \equiv 1$ és w^2 a p_j szerint nem-csökkenő sorrendet indukál, akkor a kétszintű ütemezési probléma ekvivalens a $P || \sum C_j$ párhuzamos gépes ütemezési problémával, és polinomiális időben megoldható.*

18. Tétel. *Ha $w^1 \equiv 1$ és w^2 a p_j szerint nem-növekvő sorrendet indukál, akkor a kétszintű ütemezési probléma polinomiális időben megoldható.*

Vegyük észre, hogy az előző, és a 17. Tétel alapján kapjuk, hogy

7. Következmény. *A MAX k -CUT probléma polinomiális időben megoldható egy élsúlyozott teljes gráfban, ha az élsúlyok $c(i, j) := \max\{p_i, p_j\}$, ahol p_i a csúcsokra írt nem-negatív súlyozás.*

Végül, ha nem tesszük fel, hogy $w^1 \equiv 1$, de a gépek száma nem része az inputnak, akkor bebizonyítható a következő:

19. Tétel. *Ha a gépek száma konstans, akkor a kétszintű súlyozott befejezési idő problémára van FPTAS.*

Megjegyezzük, hogy az FPTAS-unk általánosítása Schuurman & Wöginer [46] FPTAS algoritmusának a $Pm||\sum w_j C_j$ problémára.

4.2. A kétszintű rendelés elfogadási probléma

Az egyszintű $1||\sum w_j U_j$ probléma NP-nehéz, ugyanakkor, ha minden feladat súlya egyforma, akkor polinomiális időben megoldható a Moore-Hogdson algoritmussal [41]. Mivel a feladatokat határidő szerint nem-csökkenő sorrendbe rendezve minimalizálhatjuk a maximális késést [3, 7], ezért az időben elkészülő feladatokról feltehetjük, hogy ilyen sorrendben vannak ütemezve. A kétszintű ütemezési problémában, ahol a "vezető" az időben elkészülő feladatok összsúlyát maximalizálja (ami azzal ekvivalens, hogy a késők összsúlyát minimalizálja), miközben a követő a géphez rendelt feladatok súlyozott befejezési idejét minimalizálja, a "vezető" problémáját nehezíti, hogy a követő optimális megoldását az adja, hogy a feladatokat w_j^2/p_j szerint rendezi nem-növekvő sorrendbe.

Új eredmények

Több eredmény is analóg a 4.1. alfejezetben látottakkal, így azokat csak vázoljuk.

7. Propozíció. *A kétszintű, rendelés elfogadási probléma NP-nehéz mind az optimista, mind a pesszimista esetben.*

A megoldások létezésének feltételei azonosak a 4. Propozícióval. A "követő" számára a feladatok optimális sorrendje optimista esetben

$$j \text{ megelőzi } k \text{ feladatot, ha } w_j^2/p_j > w_k^2/p_k \text{ vagy } (w_j^2/p_j^2 = w_k^2/p_k^2 \text{ és } d_j < d_k), \quad (4.4)$$

míg pesszimista esetben

$$j \text{ megelőzi } k \text{ feladatot, ha } w_j^2/p_j > w_k^2/p_k \text{ vagy } (w_j^2/p_j^2 = w_k^2/p_k^2 \text{ és } d_j > d_k). \quad (4.5)$$

A (4.4) és (4.5) sorrendekkel a 5. és 6. lemmákkal analóg állítások bizonyíthatóak. Fentebb láttuk, hogy a vezető problémája NP-nehéz, ugyanakkor igazolható a következő:

8. Propozíció. *A kétszintű rendelés elfogadási probléma pseudo-polinomiális időben megoldható.*

Megjegyezzük, hogy az algoritmus az $1||\sum w_j U_j$ problémára jól ismert dinamikus program általánosítása. Egy speciális eset viszont polinomiális időben megoldható:

20. Tétel. *Ha $w^1 \equiv 1$, akkor a kétszintű rendelés elfogadási probléma polinomiális időben megoldható.*

Megjegyezzük, hogy az algoritmus a Moore-Hogdson algoritmus általánosítása.

5. fejezet

Appendix

5.1. Az $\alpha|\beta|\gamma$ jelölés rendszer

Gépütemezési problémák osztályozására Graham és szerzőtársai [21] dolgozták ki az $\alpha|\beta|\gamma$ jelölési/osztályozási rendszert, ahol α jelöli a feldolgozási környezetet, β az egyéb megszorításokat, γ pedig a célfüggvényt. Az α mező néhány lehetséges értéke:

- 1 – egyetlen gép van,
- P – az inputban adott számú párhuzamos gép van, és a feladatok feldolgozási ideje nem függ a hozzájuk rendelt géptől,
- Pm – konstans m darab párhuzamos gép van.

A β mező lehet üres, illetve néhány lehetséges értéke:

- r_j – minden feladatnak van egy legkorábbi kezdési ideje, ami előtt nem lehet beütemezni,
- $p_j = 1$ – minden feladat egységnyi ideig tart,
- ddc – a feladatok gépekhez vannak rendelve az inputban,
- $pmtn$ – a műveletek megszakíthatóak, és később folytathatóak, akár másik gépen is,
- res – további kiegészítő megújuló erőforrások vannak,
- nr – további kiegészítő nem-megújuló erőforrásokat jelöl,
- $nr = \text{const}$ – a nem megújuló erőforrások száma konstans,

- $q = \text{const}$ – a nem megújuló erőforrásból konstans sok beszállítás van.

A γ mező néhány lehetséges értéke:

- $C_{\max} := \max_j C_j$ – maximális feladat befejezési idő, ahol C_j a j feladat befejezési ideje egy ütemtervben,
- $\sum_j w_j C_j$ – befejezési idők súlyozott összege,
- $L_{\max} := \max_j C_j - d_j$ – maximális késés, adott d_j határidőkhöz képest.

Néhány példa: $1||\sum C_j$ azt a problémát jelöli, ahol egyetlen gép van, és a munkák olyan sorrendjét kell meghatározni, ami minimalizálja a befejezési idők összegét; $P||C_{\max}$ azt a problémát jelenti, ahol az inputban adott a feladatok, és a gépek száma, és minden $j \in \mathcal{J}$ munkának van egy p_j hossza, és minden feladathoz pontosan egy gépet kell rendelni (és persze a munkákat a gépeken sorrendezni) úgy, hogy a munkák befejezési idejeinek a maximuma ($C_{\max}$) a lehető legkisebb legyen. A $P|nr|C_{\max}$ feladat annyival több az előbbinél, hogy tartalmaz nem megújuló erőforrásokat is, amelyeket a feladatok fogyasztanak, és a készletüket külső forrásból adott időpontban és mennyiségben pótolják. Végül a $Pm|nr = 1, q = \text{const}|C_{\max}$ feladatban konstans sok gép van, egy nem megújuló erőforrás konstans sok beszállítással, és a célfüggvény a maximális feladat befejezési idő.

5.2. Approximációs algoritmusok alapfogalmai

$\mathcal{NP}$ -nehéz feladatokra polinomiális időben, garantáltan jó megoldást kaphatunk approximációs algoritmusok segítségével. Formálisan, adott egy optimalizálási probléma $\Pi = (D, S, c)$, ahol D a feladat példányainak a halmaza, minden $I \in D$ feladatpéldányra $S(I)$ a megengedett megoldások halmaza (lehet üres is), és minden $\sigma \in S(I)$ megoldásra $c(I, \sigma) \in \mathbb{R}$ a σ megoldás értéke, lásd [18] 6.1. fejezetét. Amennyiben $S(I)$ nem üres, jelölje $\sigma^*(I)$ az optimális megoldást, azaz minimalizálási probléma esetén minden $\sigma \in S(I)$ megoldásra teljesül, hogy $c(I, \sigma) \geq c(I, \sigma^*(I))$, míg maximalizálási problémánál $c(I, \sigma) \leq c(I, \sigma^*(I))$. Az optimum értékét $OPT(I) := c(I, \sigma^*)$ jelöli. Egy A approximációs algoritmus a Π problémára minden $I \in D$ feladatpéldányhoz meghatároz egy $\sigma \in S(I)$ megengedett megoldást, amennyiben $S(I)$ nem üres; vagy "kijelenti", hogy $S(I)$ üres. Elvárjuk, hogy A polinomiális idejű algoritmus legyen Π minden példányán. Jelölje $A(I)$ az algoritmus által visszaadott megoldás értékét. Az A algoritmus *relatív hibája* minimalizálási [maximalizálási] probléma esetén

$$R_A(I) := \frac{A(I)}{OPT(I)} \quad \left[R_A(I) := \frac{OPT(I)}{A(I)} \right].$$

A definíció szerint $1 \leq R_A(I) \leq \infty$ teljesül minimalizálási, és maximalizálási problémák esetén is.

Az A algoritmus *abszolút teljesítmény hányadosa* a következő mennyiség:

$$R_A := \inf\{r \geq 1 \mid R_A(I) \leq r \text{ minden } I \in D \text{ feladatpéldányra}\}.$$

Az A algoritmus *aszimptotikus hibája* pedig

$$R_A^\infty := \inf\left\{r \geq 1 \mid \begin{array}{l} \text{van olyan } N \in \mathbb{N}_+, \text{ hogy } R_A(I) \leq r \text{ minden olyan} \\ I \in D \text{ feladatpéldányra, amelyre } OPT(I) \geq N \end{array} \right\}.$$

Az aszimptotikus hiba azt méri, hogyan teljesít az algoritmus ha az optimum a végtelenbe tart. Az általunk vizsgált algoritmusoknál az abszolút teljesítmény hányados, és az aszimptotikus hiba egybeesik.

Egy $(\mathcal{NP}$ -nehéz) optimalizálási probléma *approximálhatóságát* méri a legkisebb (akár a probléma paramétereitől függő) r hányados, amire van olyan A approximációs algoritmus, hogy $R_A \leq r$ vagy $R_A^\infty \leq r$. Ebből a szempontból az approximációs sémák kitüntetett szerepet játszanak. Egy *polinomiális idejű approximációs séma* (PTAS) egy olyan $\{A_\varepsilon\}_{\varepsilon>0}$ algoritmus sereg, ami tetszőlegesen kicsi $\varepsilon > 0$ értékhez tartalmaz egy $1 + \varepsilon$ abszolút teljesítmény hányadosú algoritmust, ami polinomiális idejű a feladatpéldányok méretében. Egy *teljesen polinomiális idejű approximációs séma* (FPTAS) annnyival több ennél, hogy A_ε futási ideje polinomiális nem csak a feladatpéldányok méretében, de $1/\varepsilon$ -ban is.

Optimalizálási problémák approximálhatóságára léteznek negatív eredmények is, ilyen például, ha megmutatjuk, hogy adott optimalizálási problémára nincs PTAS, azaz nem lehet tetszőlegesen kicsi hibával közelíteni az optimumot polinomiális időben, ha $\mathcal{P} \neq \mathcal{NP}$.

Mind pozitív, mind pedig negatív approximálhatósági eredmények bizonyításához igen hasznos eszköztár az approximációt megőrző redukciók. Ezek közül az alábbiakban a szigorú (Strict), PTAS, és FPTAS redukciókat definiáljuk. A tárgyalás alapját a [12, 11] cikkek képezik.

Formálisan, egy *redukció* egy (f, g) függvénpár, ahol f a Π_1 optimalizálási probléma példányait a Π_2 optimalizálási probléma példányaira képezi le, és g a Π_1 probléma minden egyes I_1 példányához megad egy megengedett megoldást, a Π_2 probléma $f(I_1)$ példányához tartozó tetszőleges megoldásból kiindulva. A következő ábra szemlélteti a redukciót:

$$\begin{array}{ccccc} \text{Problémák:} & & \Pi_1 & & \Pi_2 \\ \text{Példányok:} & & I_1 & \xrightarrow{f} & f(I_1) \\ & & & & \downarrow \\ \text{Megoldások:} & g(I_1, y) & & g \longleftarrow & y \end{array}$$

(f, g) *szigorú redukció* (Strict reduction) a Π_1 problémáról a Π_2 problémára ($\Pi_1 \leq_{\text{strict}} \Pi_2$) ha f és g ha polinomiális időben számíthatóak a paramétereik méretében, és Π_1 minden I_1 példányára, valamint $f(I_1)$ minden megengedett y megoldására teljesül, hogy

$$R_{\Pi_1}(I_1, g(I_1, y)) \leq R_{\Pi_2}(f(I_1), y).$$

(f, g) *PTAS redukció* (PTAS reduction) a Π_1 problémáról a Π_2 problémára ($\Pi_1 \leq_{\text{PTAS}} \Pi_2$) ha vannak olyan $\alpha(\cdot)$, $t_f(\cdot, \cdot)$ és $t_g(\cdot, \cdot, \cdot)$ függvények, amelyek teljesítik a következőket:

- i) Π_1 tetszőleges I_1 példányára, és minden $\varepsilon > 0$ számra, $f(I_1, \varepsilon)$ a Π_2 probléma egy példánya, és f $t_f(|I_1|, \varepsilon)$ időben számítható,
- ii) Az $f(I_1, \varepsilon)$ tetszőleges y megoldására, $g(I_1, y, \varepsilon)$ megoldása I_1 -nek, és g $t_g(|I_1|, |y|, \varepsilon)$ időben kiszámítható,
- iii) minden $\varepsilon > 0$ -ra mind $t_f(\cdot, \varepsilon)$, mind pedig $t_g(\cdot, \cdot, \varepsilon)$ polinomokkal korlátozhatók,
- iv) α a Π_1 probléma hiba paraméterét képezi le a Π_2 probléma hiba paraméterére úgy, hogy $f(I_1, \varepsilon)$ minden y megoldására teljesül a következő:

$$R_{\Pi_2}(f(I_1, \varepsilon), y) \leq 1 + \alpha(\varepsilon) \text{ implies } R_{\Pi_1}(I_1, g(I_1, y, \varepsilon)) \leq 1 + \varepsilon. \quad (5.1)$$

7. Lemma. (Crescenzi [12]) Legyenek Π_1 és Π_2 optimalizációs problémák, és $\Pi_1 \leq_{\text{PTAS}} \Pi_2$. Ekkor, ha Π_2 -re van egy polinomiális idejű approximációs séma (PTAS), akkor Π_1 -re is van ilyen séma.

A következő állítás mutatja a kapcsolatot a szigorú és a PTAS redukciók között.

8. Lemma. (Crescenzi et al. [11]) Minden szigorú redukció egyben PTAS-redukció is.

Tehát Lemma 7 érvényes marad, ha a PTAS redukciót szigorúra cseréljük az állításban. Végül, egy *FPTAS redukció* (FPTAS reduction) olyan mint egy PTAS redukció a következő módosításokkal:

- iii') $t_f(\cdot, \varepsilon)$ és $t_g(\cdot, \cdot, \varepsilon)$ polinommal korlátozhatóak $1/\varepsilon$ -ban is.
- iv') α olyan függvény, ami a Π_1 probléma minden I_1 példányához, és tetszőleges $\varepsilon > 0$ hiba paraméterhez rendel egy hiba paramétert a Π_2

problémához úgy, hogy az $f(I_1, \varepsilon)$ tetszőleges y megoldására teljesül a következő:

$$\text{Ha } R_{\Pi_2}(f(I_1, \varepsilon), y) \leq 1 + \alpha(I_1, \varepsilon), \text{ akkor } R_{\Pi_1}(I_1, g(I_1, y, \varepsilon)) \leq 1 + \varepsilon. \quad (5.2)$$

Azaz, (5.2) leváltja (5.1) -et az FPTAS redukcióban.

v') α polinomiális időben számítható $|I_1|$ -ben és $1/\varepsilon$ -ban is.

vi') Van olyan kétváltozós polinom $\text{poly}(\cdot, \cdot)$, hogy $1/\alpha(I_1, \varepsilon) \leq \text{poly}(|I_1|, 1/\varepsilon)$ minden $\varepsilon > 0$ esetén.

9. Lemma. (Crescenzi [12]) Ha (f, g) egy FPTAS redukció a Π_1 problémáról Π_2 problémára, és ha Π_2 problémához van teljesen polinomiális idejű approximációs séma (FPTAS), akkor Π_1 problémára is van FPTAS.

Megmutatható, hogy egy FPTAS redukció nem feltétlenül PTAS redukció is egyben, és fordítva, egy PTAS redukció általában nem FPTAS redukció. Ugyanakkor teljesül a következő:

10. Lemma. (Crescenzi et al. [11]) Minden szigorú redukció egyben FPTAS redukció is.

11. Lemma. (Crescenzi [12], Orponen et al [45]) A fenti redukciók tranzitívek.

Irodalomjegyzék

- [1] Francisco Ballestín, Christoph Schwindt, and Jürgen Zimmermann. Resource leveling in make-to-order production: modeling and heuristic solution method. *International Journal of Operations Research*, 4(1):50–62, 2007.
- [2] Jacek Błażewicz, Klaus H Ecker, Erwin Pesch, Günter Schmidt, and Jan Weglarz. *Handbook on scheduling: from theory to applications*. Springer Science & Business Media, 2007.
- [3] Jacek Błażewicz, Klaus H Ecker, Erwin Pesch, Günter Schmidt, and Jan Weglarz. *Scheduling computer and manufacturing processes*. Springer Science & Business Media, 2013.
- [4] Jacek Blazewicz, Jan Karel Lenstra, and AHG Rinnooy Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.
- [5] Dirk Briskorn, Byung-Cheon Choi, Kangbok Lee, Joseph Leung, and Michael Pinedo. Complexity of single machine scheduling subject to nonnegative inventory constraints. *European Journal of Operational Research*, 207(2):605–619, 2010.
- [6] Dirk Briskorn, Florian Jaehn, and Erwin Pesch. Exact algorithms for inventory constrained scheduling on a single machine. *Journal of Scheduling*, 16:105–115, 2013.
- [7] Peter Brucker. *Scheduling Algorithms*. Springer-Verlag Berlin Heidelberg, 2007.
- [8] Massimiliano Caramia and Paolo Dell’Olmo. Assessing the resource usage in scheduling with incompatibilities. *OR Spectrum*, 25(4):521–547, 2003.
- [9] J Carlier and AHG Rinnooy Kan. Scheduling subject to nonrenewable-resource constraints. *Operations Research Letters*, 1(2):52–55, 1982.

- [10] Jacques Carlier. *Problèmes d'ordonnancement à contraintes de ressources: algorithmes et complexité*. Université Paris VI-Pierre et Marie Curie, Institut de programmation, 1984.
- [11] Pierluigi Crescenzi. A short guide to approximation preserving reductions. In *Computational Complexity, 1997. Proceedings., Twelfth Annual IEEE Conference on (Formerly: Structure in Complexity Theory Conference)*, pages 262–273. IEEE, 1997.
- [12] Pierluigi Crescenzi and Alessandro Panconesi. Completeness in approximation classes. *Information and Computation*, 93(2):241–262, 1991.
- [13] Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.
- [14] Márton Drótos and Tamás Kis. Resource leveling in a machine environment. *European Journal of Operational Research*, 212(1):12–21, 2011.
- [15] Márton Drótos and Tamás Kis. Scheduling of inventory releasing jobs to minimize a regular objective function of delivery times. *Journal of Scheduling*, 16(3):337–346, 2013.
- [16] János Fülöp. On the equivalence between a linear bilevel programming problem and linear optimization over the efficient set. Technical report, Laboratory of Operational Research and Decision Systems, Computer and Automation Research Institute, Hungarian Academy of Science, Budapest, 1993.
- [17] Evgeny R Gafarov, Alexander A Lazarev, and Frank Werner. Single machine scheduling problems with financial resource constraints: Some complexity results and properties. *Mathematical Social Sciences*, 62(1):7–13, 2011.
- [18] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [19] Georgii V Gens and Eugenii V Levner. Computational complexity of approximation algorithms for combinatorial problems. In *International Symposium on Mathematical Foundations of Computer Science*, pages 292–300. Springer, 1979.
- [20] Ronald L. Graham. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2):416–429, 1969.

- [21] Ronald L Graham, Eugene L Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326, 1979.
- [22] Alexander Grigoriev, Martijn Holthuijsen, and Joris van de Klundert. Basic scheduling problems with raw material constraints. *Naval Research Logistics (NRL)*, 52(6):527–535, 2005.
- [23] Péter Györgyi and Tamás Kis. Approximation schemes for single machine scheduling with non-renewable resource constraints. *Journal of Scheduling*, 17:135–144, 2014.
- [24] Péter Györgyi and Tamás Kis. Reductions between scheduling problems with non-renewable resources and knapsack problems. *Theoretical Computer Science*, 565:63–76, 2015a.
- [25] Péter Györgyi and Tamás Kis. Approximability of scheduling problems with resource consuming jobs. *Annals of Operations Research*, 235(1):319–336, 2015b.
- [26] Péter Györgyi and Tamás Kis. Approximation schemes for parallel machine scheduling with non-renewable resources. *European Journal of Operational Research*, 258(1):113–123, 2017.
- [27] Leslie A Hall and David B Shmoys. Approximation schemes for constrained scheduling problems. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 134–139. IEEE, 1989.
- [28] Elias Willem Hans. *Resource loading by branch-and-price techniques*. PhD thesis, 2001.
- [29] Sönke Hartmann and Dirk Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research*, 207(1):1–14, 2010.
- [30] John K Karlof and Wei Wang. Bilevel programming applied to the flow shop scheduling problem. *Computers & operations research*, 23(5):443–451, 1996.
- [31] Hans Kellerer, Vladimir Kotov, Franz Rendl, and Gerhard J Wöginger. The stock size problem. *Operations Research*, 46(3-supplement-3):S1–S12, 1998.

- [32] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer, 2004.
- [33] Hans Kellerer and Vitaly A Strusevich. Scheduling parallel dedicated machines under a single non-shared resource. *European Journal of Operational Research*, 147(2):345–364, 2003.
- [34] Hans Kellerer and Vitaly A Strusevich. Scheduling problems for parallel dedicated machines under multiple resource constraints. *Discrete Applied Mathematics*, 133(1):45–68, 2003.
- [35] Tamás Kis. A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Mathematical programming*, 103(3):515–539, 2005.
- [36] Tamás Kis. Rcps with variable intensity activities and feeding precedence constraints. In *Perspectives in modern project scheduling*, pages 105–129. Springer, 2006.
- [37] Tamás Kis. Approximability of total weighted completion time with resource consuming jobs. *Operations Research Letters*, 43(6):595–598, 2015.
- [38] Tamás Kis and András Kovács. On bilevel machine scheduling problems. *OR spectrum*, 34(1):43–68, 2012.
- [39] Robert C Leachman, Abdurrezak Dincerler, and Sooyoung Kim. Resource-constrained scheduling of projects with variable-intensity activities. *IIE transactions*, 22(1):31–40, 1990.
- [40] Zrinka Lukač, Kristina Šorić, and Višnja Vojvodić Rosenzweig. Production planning problem with sequence dependent setups as a bilevel programming problem. *European Journal of Operational Research*, 187(3):1504–1512, 2008.
- [41] J Michael Moore. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management science*, 15(1):102–109, 1968.
- [42] Klaus Neumann and Christoph Schwindt. Project scheduling with inventory constraints. *Mathematical Methods of Operations Research*, 56(3):513–533, 2002.

- [43] Klaus Neumann and Jürgen Zimmermann. Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research*, 117(3):591–605, 1999.
- [44] Klaus Neumann and Jürgen Zimmermann. Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, 127(2):425–443, 2000.
- [45] Pekka Orponen and Heikki Mannila. On approximation preserving reductions: Complete problems and robust measures (revised version). *Department of Computer Science, University of Helsinki*, 1990.
- [46] Petra Schuurman and Gerhard Wöginger. Approximation schemes—a tutorial. In RH Möhring, CN Potts, AS Schulz, GJ Wöginger, and LA Wolsey, editors, *Lectures on Scheduling*. 2009.
- [47] Roman Slowinski. Preemptive scheduling of independent jobs on parallel machines subject to financial constraints. *European Journal of Operational Research*, 15:366–373, 1984.
- [48] Wayne E Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.
- [49] Luís Valadares Tavares. A review of the contribution of operational research to project management. *European Journal of Operational Research*, 136(1):1–18, 2002.
- [50] Aysegül Toker, Suna Kondakci, and Nesim Erkip. Scheduling under a non-renewable resource constraint. *Journal of the Operational Research Society*, 42(9):811–814, 1991.
- [51] Vicente Valls, Ángeles Pérez, and Sacramento Quintanilla. Skilled workforce scheduling in service centres. *European Journal of Operational Research*, 193(3):791–804, 2009.
- [52] Jan Węglarz. Project scheduling with continuously-divisible, doubly constrained resources. *Management Science*, 27(9):1040–1053, 1981.
- [53] Jan Węglarz, Joanna Józefowska, Marek Mika, and Grzegorz Waligóra. Project scheduling with finite or infinite number of activity processing modes—a survey. *European Journal of Operational Research*, 208(3):177–205, 2011.