# Gépütemezés erőforrás korlátokkal

Diplomamunka

Írta: Valki Tibor

Alkalmazott matematikus szak

Témavezető:

Kis Tamás, egyetemi adjunktus
Operációkutatási Tanszék
Eötvös Loránd Tudományegyetem, Természettudományi Kar

Eötvös Loránd Tudományegyetem
Természettudományi Kar
2012

# Tartalomjegyzék

# 1. Bevezetés

Ütemezéselméleti problémák számtalan helyen felbukkannak. Általánosan fogalmazva, a cél bizonyos tevékenységek elvégzésére olyan időbeosztást találni, amely figyelembe veszi a rendelkezésre álló erőforrásokat, és valamilyen szempont szerint optimális. Az ütemezési problémákban *munkák* elvégzését ütemezzük *gépeken*, bizonyos *feltételek* mellett egy adott *célfüggvényt* optimalizálva. A feladat tehát egy *ütemezés* meghatározása, amely megmondja, hogy melyik munkát mikor és melyik gépen végezzük. Általános szabály, hogy minden gép egy időben legfeljebb egy munkán dolgozhat, és minden munkát egy időben legfeljebb egy gépen végezhetünk, lásd [1].

A dolgozat témája az olyan ütemezési problémák vizsgálata, ahol a gépek *párhuzamosak* és *dedikáltak*, a célfüggvény a *teljes átfutási idő* vagy az *átlagos átfutási idő*, és az unáris erőforrásokon túl (ezek a gépek), további *erőforrásokat* is igényelhetnek a munkák, amelyeken osztozniuk kell. Vizsgálni fogom a probléma különböző speciális eseteit, polinomiális egzakt, és approximációs algoritmusokat mutatok be, illetve illusztrálom azok működését. Ezen kívül az 5. fejezetben megvizsgálom a problémát abban az esetben, amikor a célfüggvény nem a teljes átfutási idő, hanem az átlagos átfutási idő.

A problémák modellje (lásd [2,3]): adott az elvégzendő munkák egy $N = \{J_1, J_2, \ldots, J_n\}$ halmaza és $m$ gép: $M_1, M_2, \ldots, M_m$. A gépek *párhuzamosak* (teljesen egyformák) és *dedikáltak*: mindegyik munka egy meghatározott gépen kerül feldolgozásra, ezért az $N$ halmazt előre felosztjuk $m$ részhalmazra, $N_1, N_2, \ldots, N_m$-re, és az $N_i$ halmaz munkáit csak az $M_i$ gépen dolgozhatjuk fel, $1 \le i \le m$. A $J_j$ munka *megmunkálási ideje* $p_j \ge 0$, ennyi időt vesz igénybe a munka elvégzése.

A munkák feldolgozása lehet *megszakítható* vagy *nem-megszakítható*. Ha egy munka megszakítható, akkor a feldolgozása nem feltétlenül folyamatos. Ha nincsenek további korlátozások, akkor a gépek egymástól függetlenül működnek.

Feltesszük, hogy a munkák még *hozzáadott megújuló erőforrás(oka)t* is igényelhetnek. Adott $\lambda \ge 1$ féle erőforrás, és $\sigma_\mu \ge 1$ egység bármikor elérhető a $\mu$ erőforrásból, $1 \le \mu \le \lambda$. Egy $J_j$ munkának $\rho_{j\mu} \ge 0$ egységre van szüksége $\mu$ erőforrásból a végrehajtása során. Azok a munkák, amelyeket különböző gépekhez rendeltünk, és melyeknek a $\mu$ erőforrásra van szükségük akkor dolgozhatók fel párhuzamosan, ha a teljes erőforrás felhasználásuk nem haladja meg a $\sigma_\mu$-t: $\sum_j \rho_{j\mu} \le \sigma_\mu$.

Egy adott ütemezés esetén a $J_j$ munka *befejezési ideje* az az időpillanat, amikor $J_j$ elkészül. Egy *S* ütemtervben a $J_j$ munka befejezési idejét $C_j(S)$-el jelöljük. Az *S teljes átfutási ideje* $\max_j C_j(S)$, amit $C_{\max}(S)$-el jelölünk, míg az *átlagos átfutási idő* $n$-szerese a $\sum_j C_j(S)$. A legkisebb teljes átfutási idejű vagy a legkisebb átlagos átfutási idejű ütemezést optimálisnak nevezzük, és $S^*$-gal jelöljük, attól függően, hogy mi a problémában a célfüggvény.

A problémák jelölése: $\alpha \mid \beta \mid \gamma$, ahol „$\gamma$" helyén „$C_{\max}$" vagy „$\sum_j C_j$" szerepel. Az „$\alpha$" helyén „PD$m$" vagy „PD", szerepel, melyek közül az első azt jelenti, hogy a párhuzamos és dedikált gépek száma $m$, míg a második esetben a gépek száma része az inputnak. A „$\beta$" helyén „res$\lambda\sigma\rho$", „res$\lambda\sigma\rho$, pmtn", „res$\lambda\sigma\rho$, pmtn$^*$" és „res$\lambda\sigma\rho$, pmtn$^{**}$" valamelyike szerepel. Mindegyik esetben $\lambda$ erőforrás áll rendelkezésre, $\sigma_\mu \le \sigma \ \ \forall\mu$-re, $1 \le \mu \le \lambda$, és $\rho_{j\mu} \le \rho \ \ \forall j$-re és $\forall\mu$-re, $1 \le j \le n$, $1 \le \mu \le \lambda$. Az első esetben a munkák nem szakíthatóak meg, a második esetben megszakíthatóak, a harmadik esetben megszakíthatóak az erőforrást nem igénylő munkák, azaz azok a munkák, melyek nem igényelnek erőforrást, végül a negyedik esetben megszakíthatóak azok a munkák, melyek igényelnek (legalább egy) erőforrást.

A 2. fejezetben a legegyszerűbb problémákat mutatjuk be, ahol $\lambda = \sigma = \rho = 1$, azaz egyetlen egységnyi erőforrás áll rendelkezésre. Ezt az erőforrást a munkák egy részhalmaza igényli, a többi nem. Minden erőforrást igénylő munkának a feldolgozás során egy egységre van szüksége az erőforrásból, így egyidejűleg kettő vagy több erőforrást igénylő munkát nem lehet elvégezni.

Ennek a modellnek számos alkalmazását ismerjük olyan esetekben, amikor egy közös készüléket több felhasználó között kell megosztani. Vegyünk például egy számítógép-hálózatot, ami $m$ PC-ből és egy nyomtatóból áll. Mindegyik PC-n adott programok futnak, melyek vagy használják a hálózati nyomtatót például fájlok nyomtatására, vagy nem. Ebben az esetben a PC-ket adott párhuzamos és dedikált gépként értelmezzük, a nyomtató játssza az erőforrás szerepet, és a fájlok kinyomtatása megfelel az erőforrást igénylő munkák feldolgozásának.

Ezekre a feladatokra csak akkor létezik polinomiális algoritmus, ha a gépek száma 2 vagy ha az erőforrást nem igénylő munkák megszakíthatóak. Ha az erőforrást nem igénylő munkák nem szakíthatóak meg, akkor a feladat már 3 gép esetén is NP-nehéz.

Tehát a legtöbb feladat NP-nehéz. Ezekre a feladatokra *közelítő algoritmusok* segítségével lehet megengedett megoldást találni. Egy $H$ algoritmus közelítő algoritmus, ha mindig talál megengedett megoldást, polinomidőben fut, és ha valamilyen inputra az algoritmus outputja egy $S_H$ ütemezés, akkor $\exists \alpha \geq 1$, hogy $C_{\max}(S_H)/C_{\max}(S^*) \leq \alpha$, ha a célfüggvény a teljes átfutási idő.

Az $\alpha$-t *közelítési hányadosnak*, az algoritmust $\alpha$*-közelítő algoritmusnak* nevezzük. A közelítési hányados *éles*, ha létezik egy példánya a feladatnak, melyre $C_{\max}(S_H)/C_{\max}(S^*) = \alpha$ vagy legalább $C_{\max}(S_H)/C_{\max}(S^*) \to \alpha$, ahol a feladat mérete végtelenhez tart, ha a célfüggvény a teljes átfutási idő. Egy pozitív $\varepsilon$-ra az $(1+\varepsilon)$-közelítő algoritmusok családját polinomiális idejű közelítő sémának (PTAS) nevezzük.

A 3. fejezet 3.1. alfejezetében egyetlen tetszőleges méretű erőforrás áll rendelkezésre, a gépek száma 2, és a munkák nem szakíthatóak meg. Ezt a feladatot $\mathrm{PD2}\,|\,\mathrm{res1}\cdot\cdot\,|\,C_{\max}$-szal jelöljük, ahol tehát az erőforrás nagyságára és $\rho_j$-kre (a $J_j$ munkának ennyi egységre van szüksége az erőforrásból a végrehajtása során) nincs felső korlát, $1 \leq j \leq n$. (Ne okozzon félreértést a 2. fejezetben szereplő $\rho_i$, $1 \leq i \leq m$ jelölés!) Erre a feladatra létezik polinomiális algoritmus.

Ez a modell egy emberi erőforrás elosztás esetével illusztrálható. Vegyünk két csoportot, amelyek különböző projektekben vesznek részt. Egy projekt elvégzéséhez az illetékes csoportvezető kéréssel fordulhat a felső vezetéshez, hogy irányítson át a csoport számára segéderőt azért, hogy azok segítsenek a csoportnak a projekt teljesítésében. A vezetésnek $\sigma$ embere van, akiket erre a célra ki tud jelölni.

Itt a csoportok megfeleltethetők a gépeknek, a projektek a munkáknak, és a $\sigma$ ember egy $\sigma$ méretű erőforrásnak.

A 3. fejezet további alfejezeteiben a $\mathrm{PD2}\,|\,\mathrm{res211}\,|\,C_{\max}$, a $\mathrm{PD2}\,|\,\mathrm{res222}\,|\,C_{\max}$ és a $\mathrm{PD2}\,|\,\mathrm{res311}\,|\,C_{\max}$ feladatok szerepelnek, melyekben egy munka *egynél több erőforrást* is használhat, és melyek közül az első megoldható polinomiális idő alatt, míg a másik kettő NP-nehéz, ezért nem lehet általánosítani a fejezetben található algoritmusokat.

A 4. fejezetben $\lambda$ darab egységnyi erőforrás áll rendelkezésre ($\sigma = \rho = 1$), és a munkáknak *legfeljebb egy erőforrásra* van szükségük. Itt azt a munkát, aminek szüksége van a $\mu$ erőforrásra, $\mu$*-erőforrás munkának* nevezzük, $1 \leq \mu \leq \lambda$, és az erőforrást nem igénylő munkákat 0-*erőforrás munkáknak* is nevezhetjük. Minden $\mu$-

erőforrás munkának egy egységre van szüksége a $\mu$ erőforrásból a végrehajtása során, ezért nem hajtható végre egyszerre két olyan munka, melyeknek ugyanarra az erőforrásra van szükségük. Azt az esetet vizsgáljuk, amikor a munkák feldolgozása nem-megszakítható, valamint amikor az erőforrást nem igénylő munkák megszakíthatóak.

A modellt sok szituációra lehet alkalmazni, amelyekben különböző eszközöket vagy készülék egységeket kell megosztani több felhasználó részére.

Példának vegyünk egy $m$ munkaállomásból álló számítógép hálózatot, és két nyomtatót, amelyikből az egyik színes nyomtató. Mindegyik munkaállomáson adott programok futnak, melyek vagy használják valamelyik nyomtatót, vagy nem. Például bizonyos fájlok nyomtatását kell elvégezni, amelynek egy része színes nyomtatást igényel. Ebben az esetben a munkaállomásokat párhuzamos és dedikált gépeknek tekintjük, és a nyomtatók két hozzáadott erőforrásként szerepelnek.

Az 5. fejezetben a célfüggvény az átlagos átfutási idő, és csak a legegyszerűbb feladatokat vizsgáltam meg: $\lambda = \sigma = \rho = 1$, azaz egyetlen egységnyi erőforrás áll rendelkezésre, ráadásul minden gépen csak 1-1 erőforrást igénylő munka van.

Erre a feladatra bemutatok egy nem túl hatékony közelítő algoritmust, de az kérdés maradt, hogy ez a feladat megoldható-e polinomiális idő alatt, vagy NP-nehéz-e.

# 2. Egy egységnyi erőforrás

Ebben a fejezetben azokkal a feladatokkal foglalkozunk, melyekben egyetlen egységnyi erőforrás áll rendelkezésre, lásd [2].

Mindegyik $N_i$ halmazt ($i = 1,2,\ldots,m$) osszuk fel két részhalmazra, $Q_i$-re és $R_i$-re, ahol $Q_i$-ben csak erőforrást nem igénylő munka, és $R_i$-ben csak erőforrást igénylő munka van, $N_i = R_i \cup Q_i$, $N = \bigcup_{i=1}^{m} N_i$.

$N_i' \subseteq N_i$, $N_i' \neq \varnothing$ esetén $p\left(N_i'\right) := \sum_{J_j \in N_i'} p_j$, és $p(\varnothing) := 0$.

$J_j \in N_i$ esetén $p(J_j) := p(\{J_j\}) = p_j$.

Egy tetszőleges $S$ ütemezésre jelölje $C_i(S)$ a legkorábbi olyan időpontot, amikorra az összes munka elkészül az $M_i$ gépen. (Ne okozzon félreértést, hogy egy $J_j$ munka befejezési idejét $C_j(S)$-el jelöltük!) Ha nem okoz félreértést, $C_i(S)$ helyett egyszerűen $C_i$-t írhatunk.

Világos, hogy $C_{\max}(S) = \max\{C_i(S) \mid i = 1,\ldots,m\}$. Mivel $C_i(S) \geq p(N_i) = p(Q_i) + p(R_i)$,

(1) $C_{\max}(S) \geq \max\{p(Q_i) + p(R_i) \mid i = 1,2,\ldots,m\}$.

Ezt a relációt *gép-alapú alsó korlátnak* nevezzük. Mivel egyidejűleg két erőforrást igénylő munkát nem lehet feldolgozni, ezért

(2) $C_{\max}(S) \geq \sum_{i=1}^{m} p(R_i)$.

Ezt a relációt *erőforrás-alapú alsó korlátnak* nevezzük. Mindkét alsó korlát, (1) és (2), független attól, hogy a munkákat megszakíthatjuk-e vagy nem.

Ebben a fejezetben gyakran használjuk a *csoportosító eljárás szemléletet*. $\forall i$-re ($1 \leq i \leq m$) $Q_i$-ből egyetlen kötegelt munkát hozunk létre, amit $\xi_i$-vel jelölünk, és $R_i$-ből is létrehozunk egyetlen kötegelt munkát, amit $\rho_i$-vel jelölünk. A kötegeket összetett munkáknak tekintjük, a kötegeken belül a munkákat tetszőleges sorrendben dolgozzuk fel, szünet nélkül.

A 2.1. alfejezetben a $\mathrm{PD}m \mid \mathrm{res}111 \mid C_{\max}$, $\mathrm{PD}m \mid \mathrm{res}111, \mathrm{pmtn}^* \mid C_{\max}$ stb. feladatok komplexitását elemezzük. Megmutatjuk, hogy a feladat NP-nehéz, ha $m \geq 3$, és erősen NP-nehéz, ha $m$ változó. A 2.2. alfejezetben közelítő algoritmusokat fogunk vizsgálni a csoportosító eljárás szemlélet alapján. Mutatunk egy legjobb ilyen algoritmust. Ezen

kívül szó lesz arról, hogy ha az erőforrást nem igénylő munkákból gépenként legfeljebb két köteget készíthetünk, akkor lehet jobb algoritmust tervezni. A 2.3. alfejezetben mutatunk egy polinomiális idejű közelítő sémát a feladatra, rögzített $m$ esetén.

## 2.1. Komplexitás eredmények

Ebben az alfejezetben a $\mathrm{PD}m\,|\,\mathrm{res111}\,|\,C_{\max}$, $\mathrm{PD}m\,|\,\mathrm{res111},\mathrm{pmtn}^*\,|\,C_{\max}$ stb. problémák komplexitását elemezzük, lásd [2]. Megmutatjuk, hogy a probléma megoldható lineáris idő alatt, ha $m=2$, attól függetlenül, hogy a munkákat megszakíthatjuk-e vagy nem. Továbbá megmutatjuk, hogy megoldható lineáris idő alatt, ha $m>2$ és az erőforrást nem igénylő munkák megszakíthatóak. Végül megmutatjuk, hogy gyengén NP-nehéz, ha $m\geq 3$ rögzített, és erősen NP-nehéz, ha $m$ része az inputnak, még akkor is, ha az erőforrást igénylő munkák megszakíthatóak.

A $\mathrm{PD2}\,|\,\mathrm{res111}\,|\,C_{\max}$ és a $\mathrm{PD2}\,|\,\mathrm{res111},\mathrm{pmtn}\,|\,C_{\max}$ problémára adunk egy egyszerű a csoportosító eljárás szemléleten alapuló algoritmust.

**PD2 algoritmus** (lásd [2])

*Input:* Egy példánya a $\mathrm{PD2}\,|\,\mathrm{res111}\,|\,C_{\max}$ vagy a $\mathrm{PD2}\,|\,\mathrm{res111},\mathrm{pmtn}\,|\,C_{\max}$ problémának

*Output:* Egy optimális $S^*$ ütemezés

1. $R_1$-ből $\rho_1$, $R_2$-ből $\rho_2$, $Q_1$-ből $\xi_1$ és $Q_2$-ből $\xi_2$ köteg készítése.
2. $M_1$ gépre $\rho_1$, majd $\xi_1$ szünet nélkül a 0 időponttól kezdve.
3. $\xi_2$ az $M_2$ gépre, a 0 időponttól kezdve, majd $\rho_2$ amilyen korán csak lehetséges, azaz $\max\{p(R_1), p(Q_2)\}$-től.
4. $S^*$: az eredményül kapott ütemezés. Stop.

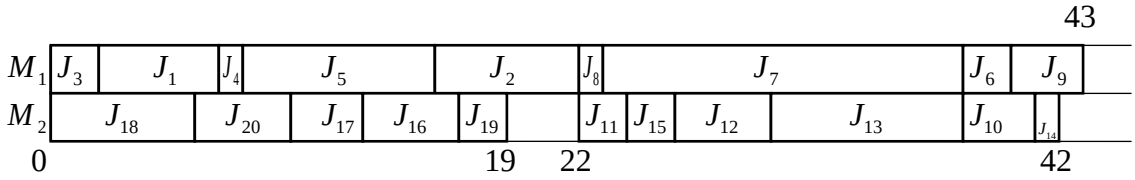
**1. Példa:** $R_1 := \{J_1,\ldots,J_5\}$, $Q_1 := \{J_6,\ldots,J_9\}$, $R_2 := \{J_{10},\ldots,J_{15}\}$, $Q_2 := \{J_{16},\ldots,J_{20}\}$, $J_1,\ldots,J_{20}$ megmunkálási idői legyenek rendre 5, 6, 2, 1, 8, 2, 15, 1, 3, 3, 2, 4, 8, 1, 2, 4, 3, 6, 2, 4.

Kötegek: $\rho_1$, $\xi_1$, $\rho_2$ és $\xi_2$. A kötegeken belül a munkák sorrendje tetszőleges. Legyen a $\rho_1$-beli munkák sorrendje $(J_3, J_1, J_4, J_5, J_2)$, a $\xi_1$-belieké $(J_8, J_7, J_6, J_9)$, a $\rho_2$-belieké $(J_{11}, J_{15}, J_{12}, J_{13}, J_{10}, J_{14})$ és a $\xi_2$-belieké $(J_{18}, J_{20}, J_{17}, J_{16}, J_{19})$.

$p(R_1)=22$, $p(Q_1)=21$, $p(R_2)=20$ és $p(Q_2)=19$.

$\max\{p(R_1), p(Q_2)\} = 22 = p(R_1)$.

$S^*$:




**1. Tétel** (lásd [2]): A $\mathrm{PD2}\,|\,\mathrm{res111}\,|\,C_{\max}$ és a $\mathrm{PD2}\,|\,\mathrm{res111},\mathrm{pmtn}\,|\,C_{\max}$ megoldható $\mathrm{O}(n)$ idő alatt a PD2 algoritmussal.

**Biz.:** Az $S^*$ (a PD2 algoritmus outputja) nyilván megengedett és optimális, mert

$C_1 = p(Q_1) + p(R_1),\ C_2 = \max\{p(R_1), p(Q_2)\} + p(R_2),$

tehát $C_{\max}(S^*)$ eléri valamelyik alsó korlátot, (1)-et vagy (2)-t.

Nyilván $\mathrm{O}(n)$ idő elég PD2 futásához, és az eredményünk független attól, hogy megengedtük-e a megszakíthatóságot vagy sem. ■

Most a $\mathrm{PD}m\,|\,\mathrm{res111},\mathrm{pmtn}^*\,|\,C_{\max}$ problémára adunk polinomiális megoldást, ahol $m > 2$.

**PDm1 algoritmus** (lásd [2])

*Input:* Egy példánya a $\mathrm{PD}m\,|\,\mathrm{res111},\mathrm{pmtn}^*\,|\,C_{\max}$ problémának

*Output:* Egy optimális $S^*$ ütemezés

1. $\forall i$-re $i = 1$-től $i = m$-ig: $R_i$-ből $\rho_i$ és $Q_i$-ből $\xi_i$ köteg készítése.
2. $M_1$-en a 0 időponttól kezdve kezdve $\rho_1$.
   $F_1 := p(R_1)$, $\rho_1$ köteg befejezési ideje.
   $M_1$-en $F_1$-től kezdve $\xi_1$.
3. $\forall i$-re $i = 2$-től $i = m$-ig:
   3.1. $M_i$-n az $F_{i-1}$-től kezdve $\rho_i$, ahol $F_{i-1}$ $\rho_{i-1}$ köteg befejezési ideje az $M_{i-1}$-en.
   3.2. $F_i := F_{i-1} + p(R_i)$, $\rho_i$ köteg befejezési ideje az $M_i$-n.
   3.3. Kezdjük megmunkálni a $\xi_i$ köteget az $M_i$-n, a $[0, F_{i-1}]$ időintervallumban. Ha $p(Q_i) > F_{i-1}$, akkor szakítsuk meg a munkát $F_{i-1}$-nél, és folytassuk $F_i$-nél. ($F_i$-től kezdve még $p(Q_i) - F_{i-1}$ idő kell, hogy befejezzük $\xi_i$ megmunkálását.)

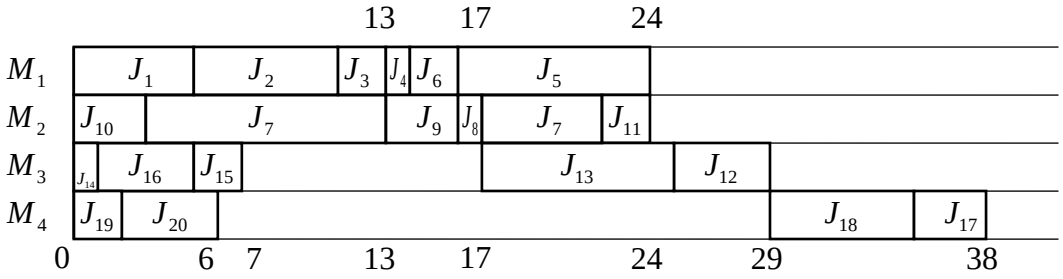
4. $S^*$: az eredményül kapott ütemezés. Stop.

**2. Példa:** $m := 4$, $N := \{J_1,\ldots,J_{20}\}$, $J_1,\ldots,J_{20}$ megmunkálási idői legyenek ugyanannyiak, mint az 1. Példában. $R_1 := \{J_1, J_2, J_3\}$, $Q_1 := \{J_4, J_5, J_6\}$, $R_2 := \{J_8, J_9\}$, $Q_2 := \{J_7, J_{10}, J_{11}\}$, $R_3 := \{J_{12}, J_{13}\}$, $Q_3 := \{J_{14}, J_{15}, J_{16}\}$, $R_4 := \{J_{17}, J_{18}\}$, $Q_4 := \{J_{19}, J_{20}\}$. Kötegek: $\rho_1$, $\xi_1$, $\rho_2$, $\xi_2$, $\rho_3$, $\xi_3$, $\rho_4$ és $\xi_4$. Legyen a $\rho_1$-beli munkák sorrendje $(J_1, J_2, J_3)$, a $\xi_1$-belieké $(J_4, J_6, J_5)$, a $\rho_2$-belieké $(J_9, J_8)$, a $\xi_2$-belieké $(J_{10}, J_7, J_{11})$, a $\rho_3$-belieké $(J_{13}, J_{12})$, a $\xi_3$-belieké $(J_{14}, J_{16}, J_{15})$, a $\rho_4$-belieké $(J_{18}, J_{17})$ és a $\xi_4$-belieké $(J_{19}, J_{20})$.

$F_1 = p(R_1) = 13$, $F_2 = 13 + p(R_2) = 17$, $F_3 = 17 + p(R_3) = 29$ és $F_4 = 29 + p(R_4) = 38$.

$p(Q_2) = 20$, $p(Q_3) = 7$ és $p(Q_4) = 6$. $p(Q_2) > F_1$, ezért a $\xi_2$ köteget megszakítjuk $F_1$-nél, és folytatjuk $F_2$-nél.

$S^*$:




**2. Tétel** (lásd [2]): A PD$m$ | res111, pmtn$^*$ | $C_{\max}$ probléma megoldható $O(n)$ idő alatt a PDm1 algoritmussal.

**Biz.:** $S^*$ nyilván megengedett, és mivel

$$C_{\max}(S^*) = \max\left\{\max_{1\le i\le m}\{p(Q_i) + p(R_i)\}, \sum_{i=1}^{m} p(R_i)\right\},$$

tehát $C_{\max}(S^*)$ eléri valamelyik alsó korlátot, (1)-et vagy (2)-t, ezért $S^*$ optimális. Az 1. lépés $O(n)$ időt igényel, míg a többi $O(m)$-et, és mivel $m \le n$, $O(n)$ idő elég PDm1 futásához. ■

Mostantól feltesszük, hogy az erőforrást nem igénylő munkák nem szakíthatóak meg. Megmutatjuk, hogy a PD3 | res111 | $C_{\max}$ és még a PD3 | res111, pmtn$^{**}$ | $C_{\max}$ probléma is NP-nehéz. A következő problémát használjuk a redukcióra.

*Partíció:* Adottak $e_1,\ldots,e_t$ pozitív egészek, ahol $\sum_{i\in T} e_i = 2E$, $T = \{1,\ldots,t\}$. Léteznek-e $T_1$ és $T_2$ részhalmazok, hogy $T_1 \cap T_2 = \varnothing$, $T_1 \cup T_2 = T$ és $\sum_{i\in T_1} e_i = \sum_{j\in T_2} e_j = E$?

Közismert, hogy a Partíció gyengén NP-teljes.

**3. Tétel** (lásd [2]): A $\mathrm{PD3}\,|\,\mathrm{res111}\,|\,C_{\max}$ és még a $\mathrm{PD3}\,|\,\mathrm{res111},\mathrm{pmtn}^{**}\,|\,C_{\max}$ feladat is gyengén NP-nehéz.

**Biz.:** Tetszőleges Partícióra definiáljuk a következő példányát a $\mathrm{PD3}\,|\,\mathrm{res111}\,|\,C_{\max}$ vagy a $\mathrm{PD3}\,|\,\mathrm{res111},\mathrm{pmtn}^{**}\,|\,C_{\max}$ feladatnak $n = t+5$ munkával.

$Q_1 = \{J_1,\ldots,J_t\},\ R_1 = \{J_{t+1}\},$

$Q_2 = \{J_{t+2}\},\ R_2 = \{J_{t+3}\},$

$Q_3 = \{J_{t+4}\},\ R_3 = \{J_{t+5}\}.$

A megmunkálási idők a következők:

$p_j = e_j,\ j = 1,\ldots,t,\ p_{t+1} = 1,$

$p_{t+2} = p_{t+4} = E+1,\ p_{t+3} = p_{t+5} = E.$

Vegyük észre, hogy pontosan egy erőforrást igénylő munka van minden egyes gépen. A tétel bizonyításához megmutatjuk, hogy a problémának ebben a konstruált esetében pontosan akkor létezik egy $S_0$ ütemezés, melyre $C_{\max}(S_0) \le y = 2E+1$, ha a Partíciónak van megoldása.

Tegyük fel, hogy a Partíciónak van megoldása, és $T_1$ és $T_2$ az elvárt részhalmazai a $T$ halmaznak. A kívánt $S_0$ ütemezés létezik, és a következőképpen írható le. Egyik gépen sincs szünet. Az $M_1$ gép kezdi a megmunkálást a $J_j$ ($j \in T_1$) munkákkal, majd a $J_{t+1}$ munka következik, végül a $J_j$ ($j \in T_2$) munkákkal fejeződik be. Az $M_2$ gép megmunkálja a $J_{t+2}$ és aztán a $J_{t+3}$ munkát. Az $M_3$ gépen a sorrend: $J_{t+5}$, $J_{t+4}$.

Könnyű leellenőrizni, hogy az imént ismertetett ütemezés megengedett, és hogy $C_{\max}(S_0) = y$.

Most tegyük fel, hogy a kívánt $S_0$ ütemezés létezik. Mivel minden gépen a teljes munkaterhelés $y$, ebből következik, hogy $C_{\max}(S_0) = y$ és egyik gépen sincs szünet. Ezen kívül az erőforrást igénylő munkák összes megmunkálási idői $2E+1$, így a $[0,2E+1]$ időintervallumban minden időpillanatban pontosan egy ilyen munka van feldolgozás alatt. Mivel a $J_{t+2}$ és a $J_{t+4}$ munkák nem megszakíthatóak, ezért az egyiket

a $[0, E+1]$ intervallumban dolgozzuk fel, míg a másikat a $[E, 2E+1]$ intervallumban. Ezért a $J_{t+3}$ és $J_{t+5}$ erőforrást igénylő munkákat a $[0, E]$ és a $[E+1, 2E+1]$ intervallumhoz rendeljük hozzá. Emiatt a $J_{t+1}$ munkát muszáj az $M_1$ gépen a $[E, E+1]$ intervallumban feldolgozni. A többi munkát az $M_1$ gépen két intervallumban, mégpedig a $[0, E]$-ben és az $[E+1, 2E+1]$-ben kell feldolgozni, amiből következik, hogy a Partíciónak van megoldása. ■

Most tekintsük a $\mathrm{PD}\,|\,\mathrm{res111}\,|\,C_{\max}$ problémát, ahol a gépek $m$ száma változó.

**4. Tétel** (lásd [2]): Ha az $m$ része az inputnak, akkor a $\mathrm{PD}\,|\,\mathrm{res111}\,|\,C_{\max}$ és még a $\mathrm{PD}\,|\,\mathrm{res111}, \mathrm{pmtn}^{**}\,|\,C_{\max}$ feladat is erősen NP-nehéz.

**Biz.:** Megtalálható [2]-ben. ■

## *2.2. Közelítő algoritmus csoportosító eljárással*

Most az NP-nehéz feladatokhoz polinomiális idejű közelítő algoritmusokat fogunk vizsgálni a csoportosító eljárás szemlélet alapján, lásd [2].

**5. Tétel** (lásd [2]): Minden olyan algoritmus esetén, mely a csoportosító eljárás szemléleten alapul, a $\mathrm{PD}m\,|\,\mathrm{res111}\,|\,C_{\max}$ probléma példányai közül létezik olyan, hogy

$$\frac{C_{\max}(S)}{C_{\max}(S^*)} \geq \begin{cases} \frac{3}{2} - \frac{1}{2m}, & \text{ha } m \text{ páratlan,} \\ \frac{3}{2} - \frac{1}{2(m-1)}, & \text{ha } m \text{ páros,} \end{cases}$$

ahol $S$ az algoritmus outputja, $S^*$ az optimális ütemezés.

**Biz.**: Legyen $m$ páratlan. Konstruálunk olyan példányát a $\mathrm{PD}m\,|\,\mathrm{res111}\,|\,C_{\max}$ feladatnak, ahol $C_{\max}(S) \geq ((3/2) - (1/2m))C_{\max}(S^*)$ teljesül.

Legyen $\forall j$-re $p_j = 1$ egység. Legyen minden gépen $m$ munka, melyből 1-nek kell az erőforrás.

Nyilvánvaló, hogy $C_{\max}(S^*) = m$ egység. Például: a $[0, m]$ intervallumban minden gép folyamatosan dolgozik, az $M_i$ gép a hozzá tartozó erőforrást igénylő munkát az $[i-1, i]$ intervallumban dolgozza fel, $i = 1, \ldots, m$.

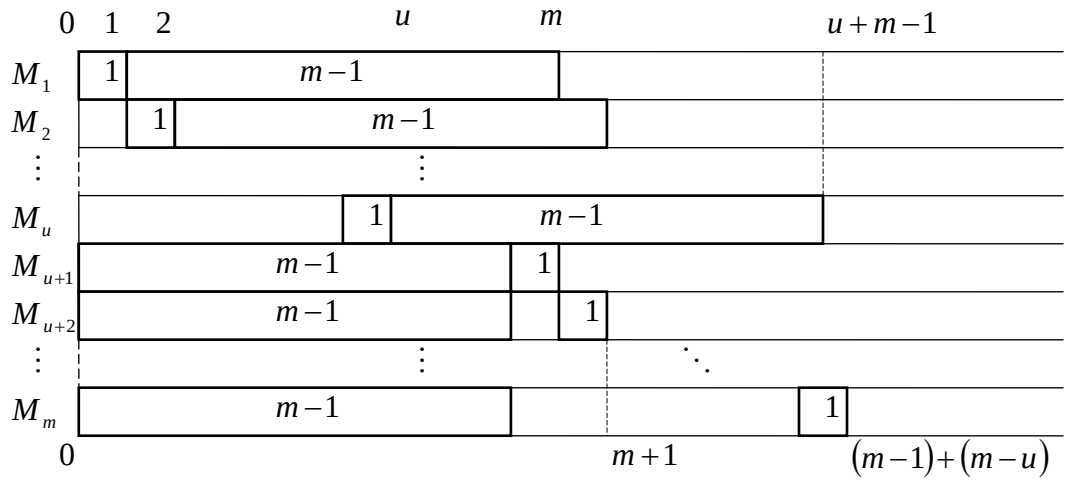
Kötegek: $\rho_i$ (megmunkálási ideje 1 egység), $\xi_i$ (megmunkálási ideje $m-1$ egység).

Kétféle gép: az egyik gépen $\rho_i$ köteg után jön $\xi_i$ köteg, a másik gépen fordítva.

A gépeket úgy rendezzük sorba, hogy

$M_1,\ldots,M_u$: első fajta gép (legkorábban $C^{(1)} = m+u-1$-nél végeznek),

$M_{u+1},\ldots,M_m$: második fajta gép (legkorábban $C^{(2)} = m+(m-u-1)$-nél végeznek).




Ebből következik, hogy $C_{\max}(S) \geq \min\{m+\max\{u-1, m-u-1\} \mid u=1,\ldots,m\}$.

Páratlan $m$-re a minimumot $u=(m-1)/2$ esetén érjük el. Így $C_{\max}(S) \geq (3m-1)/2$, és

$\frac{C_{\max}(S)}{C_{\max}(S^*)} \geq \frac{3m-1}{2m} = \frac{3}{2} - \frac{1}{2m}$, ahogy állítottuk.

Legyen most $m$ páros.

A PD$m \mid$res111$\mid C_{\max}$ feladat egy speciális példánya:

Minden gépen 1 erőforrást igénylő munka, melyeknek megmunkálási ideje $m/(m-1)$ egység az $M_1,\ldots,M_{m-1}$ gépeken, az $M_m$ gépen $\varepsilon$, ahol $\varepsilon$ egy kicsi pozitív szám.

Az $M_1, M_{m-1}$ és $M_m$ gépen 1 másik munka, melyek megmunkálási ideje $m-m/(m-1)$, $m-m/(m-1)$ és $m$ egység.

A többi $M_i$ gépen ($2 \leq i \leq m-2$) 2 erőforrást nem igénylő munka, melyek megmunkálási ideje $(i-1)(m/(m-1))$ és $m-i(m/(m-1))$ egység.

Nyilvánvaló, hogy $C_{\max}(S^*) = m+\varepsilon$ egység. Például: a $[0,m]$ intervallumban az $M_i$ gépek ($1 \leq i \leq m-1$) folyamatosan dolgoznak, és a hozzájuk tartozó erőforrást igénylő munkát az $[(i-1)(m/(m-1)), i(m/(m-1))]$ intervallumban dolgozzák fel. Az $M_m$ gép a $[0, m+\varepsilon]$ intervallumban folyamatosan dolgozik, és a hozzá tartozó erőforrást igénylő munkát az $[m, m+\varepsilon]$ intervallumban dolgozza fel.

Kötegek: $\rho_i$, $\xi_i$.

Kétféle gép: az egyik gépen $\rho_i$ köteg után jön $\xi_i$ köteg, a másik gépen fordítva.

A gépeket úgy rendezzük sorba, hogy

$M_1,\ldots,M_u$: első fajta gép (legkorábban $C^{(1)} = m + (u-1)(m/(m-1))$-nél végeznek),

$M_{u+1},\ldots,M_m$: második fajta gép (legkorábban $C^{(2)} = m + (m-u-2)(m/(m-1)) + \varepsilon$-nál végeznek).

Ellenőrizhető, hogy a legkisebb értéke $\max\{C^{(1)}, C^{(2)}\}$-nek $u = m/2$-nél van, és

$\max\{C^{(1)}, C^{(2)}\} = C^{(1)} = ((3m)/2) - (m/(2(m-1)))$.

Ezért

$$\frac{C_{\max}(S)}{C_{\max}(S^*)} \geq \frac{\frac{3m}{2} - \frac{m}{2(m-1)}}{m+\varepsilon} \xrightarrow{\varepsilon \to 0} \frac{3}{2} - \frac{1}{2(m-1)}. \blacksquare$$

Most egy algoritmust mutatunk be, ami az 5. Tétel korlátait betartó ütemtervet ad.

**GT algoritmus** (lásd [2])

*Input:* Egy példánya a $\mathrm{PD}m \mid \mathrm{res}111 \mid C_{\max}$ problémának

*Output:* Egy $S_G$ heurisztikus ütemezés

1. Ha szükséges, a gépeket számozzuk át úgy, hogy $p(R_m) = \max\{p(R_i) \mid i = 1,\ldots,m\}$.
2. $R := \sum_{i=1}^{m} p(R_i)$.
3. $\forall i$-re $i = 1$-től $i = m$-ig: $R_i$-ből $\rho_i$ és $Q_i$-ből $\xi_i$ köteg készítése.
4. Ha $m$ páros és $p(R_m) \leq \frac{1}{m-1} R$, akkor $u := \frac{m}{2}$ és menjünk a 7. lépésre; különben menjünk az 5. lépésre.
5. Ha $m$ páratlan, akkor $x := m$; különben $x := m-1$.
6. A gépeket vizsgáljuk meg a számozásuk sorrendjében, és keressük meg a legkisebb $u$ indexet ($1 \leq u \leq m$), hogy $\sum_{i=1}^{u-1} p(R_i) \leq \left(\frac{1}{2} - \frac{1}{2x}\right) R$, $\sum_{i=1}^{u} p(R_i) > \left(\frac{1}{2} - \frac{1}{2x}\right) R$.
7. $M_1$ gépen $\rho_1$, majd $\xi_1$ köteg szünet nélkül a 0 időponttól kezdve. Ha $u \geq 2$, $\forall k$-ra $k = 2$-től $k = u$-ig $M_k$ gépen $\rho_k$, majd $\xi_k$ köteg szünet nélkül $\sum_{i=1}^{k-1} p(R_i)$-től kezdve.

8. $\forall k$-ra $k = u+1$-től $k = m$-ig $M_k$ gépen $\xi_k$ köteg a nulla időponttól kezdve.
9. $M_m$ gépen $\rho_m$ köteg $\max\left\{p(Q_m), \sum_{i=1}^{u} p(R_i)\right\}$-től kezdve.
10. $\forall k$-ra $k = m-1$-től (visszafelé) $k = u+1$-ig $M_k$ gépen $\rho_k$ köteg $\max\{p(Q_k), C_{k+1}\}$-től.
11. $S_G$ : az eredményül kapott ütemezés. Stop.

Nyilvánvaló, hogy az algoritmus lefut $O(mn)$ idő alatt.

**3. Példa:** Legyen ugyanaz, mint a 2. Példa.

A gépeket úgy számozzuk át, hogy az 1. és a 4. gépet felcseréljük: $R_4 := \{J_1, J_2, J_3\}$, $Q_4 := \{J_4, J_5, J_6\}$, $R_1 := \{J_{17}, J_{18}\}$, $Q_1 := \{J_{19}, J_{20}\}$.

Kötegek: $\rho_1$, $\xi_1$, $\rho_2$, $\xi_2$, $\rho_3$, $\xi_3$, $\rho_4$ és $\xi_4$. Legyen a $\rho_1$-beli munkák sorrendje $(J_{17}, J_{18})$, a $\xi_1$-belieké $(J_{19}, J_{20})$, a $\rho_2$-belieké $(J_9, J_8)$, a $\xi_2$-belieké $(J_{10}, J_7, J_{11})$, a $\rho_3$-belieké $(J_{13}, J_{12})$, a $\xi_3$-belieké $(J_{15}, J_{14}, J_{16})$, a $\rho_4$-belieké $(J_2, J_3, J_1)$ és a $\xi_4$-belieké $(J_4, J_6, J_5)$.

$p(R_1) = 9$, $p(R_2) = 4$, $p(R_3) = 12$ és $p(R_4) = 13$, ezért $R = 9+4+12+13 = 38$.

$\frac{1}{m-1} R = \frac{1}{3} 38 = 12 + \frac{2}{3} < p(R_4) = 13$.

$x = 3$.

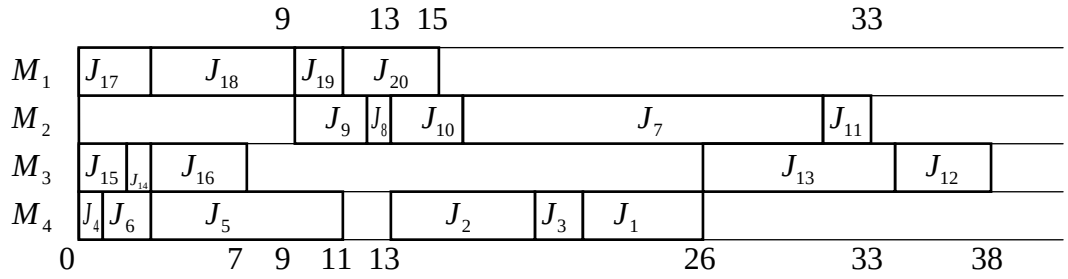
$9 = p(R_1) \le \left(\frac{1}{2} - \frac{1}{6}\right) 38 = 12 + \frac{2}{3}$, $13 = p(R_1) + p(R_2) > 12 + \frac{2}{3}$, ezért $u = 2$.

$\max\{p(Q_4), p(R_1) + p(R_2)\} = \max\{11,13\} = 13$, ezért $C_4 = 13 + p(R_4) = 26$.

$\max\{p(Q_3), C_4\} = \max\{7,26\} = 26$, ezért $C_3 = 26 + p(R_3) = 38$.

$C_1 = p(R_1) + p(Q_1) = 9 + 6 = 15$ és $C_2 = p(R_1) + p(R_2) + p(Q_2) = 13 + 20 = 33$, ezért $C_{\max}(S_G) = 38 = R$, tehát itt $S_G$ optimális.

$S_G$ :




**6. Tétel** (lásd [2]): A PD$m\,|\,\mathrm{res}111\,|\,C_{\max}$ probléma esetén, a GT algoritmus eredményeként kapott $S_G$ ütemezésre:

$$\frac{C_{\max}(S_G)}{C_{\max}(S^*)} \le \begin{cases} \dfrac{3}{2} - \dfrac{1}{2m}, & \text{ha} \quad m \quad \text{páratlan,} \\ \dfrac{3}{2} - \dfrac{1}{2(m-1)}, & \text{ha} \quad m \quad \text{páros,} \end{cases}$$

és a korlát éles, így nincs a GT algoritmusmál jobb, a csoportosító eljárás szemléleten alapuló polinomiális idejű közelítő algoritmus.

**Biz.:** $S_G$ nyilván megengedett.

Nyilván $\max\{C_i \mid i = 1,\ldots,m\} = \max\{C_1,\ldots,C_{u+1}\}$.

Ezen kívül $\max\{C_i \mid i = 1,\ldots,u\} \le \sum_{i=1}^{u-1} p(R_i) + \max\{p(Q_1),\ldots,p(Q_{u-1}),\, p(Q_u) + p(R_u)\}$,

ahol $\max\{p(Q_1),\ldots,p(Q_{u-1}),\, p(Q_u) + p(R_u)\} \le C_{\max}(S^*)$, ezért

$$\max\{C_i \mid i = 1,\ldots,u\} \le \sum_{i=1}^{u-1} p(R_i) + C_{\max}(S^*).$$

Ha a 4. lépés feltételei fennállnak, akkor az 1. lépés miatt

$$\sum_{i=1}^{u-1} p(R_i) \le \left(\frac{m}{2} - 1\right) p(R_m) \le \left(\frac{m}{2} - 1\right) \frac{1}{m-1} R \le \left(\frac{1}{2} - \frac{1}{2(m-1)}\right) C_{\max}(S^*).$$

Ha nem, akkor definíció szerint

$$\sum_{i=1}^{u-1} p(R_i) \le \left(\frac{1}{2} - \frac{1}{2x}\right) R \le \left(\frac{1}{2} - \frac{1}{2x}\right) C_{\max}(S^*).$$

Így, ha $C_{\max}(S_G) = \max\{C_1,\ldots,C_m\} = \max\{C_1,\ldots,C_{u+1}\} = \max\{C_1,\ldots,C_u\}$, akkor a tétel első állítása fennáll, még akkor is, ha $u = 1$ vagy $u = m$, feltéve, hogy a 4. lépés feltételei nem állnak fenn.

Tegyük most fel, hogy $u \ne m$ és $C_{\max}(S_G) = C_{u+1}$. Ha

$$C_{u+1} = \max\left\{ p(R_{u+1}) + p(Q_{u+1}), \sum_{i=1}^{u} p(R_i) + \sum_{i=u+1}^{m} p(R_i) \right\},$$

akkor $S_G$ optimális, és a tétel első állításával készen vagyunk. Ezért tegyük fel, hogy

$$C_{u+1} = \max_{u+2 \le v \le m} \left\{ p(Q_v) + p(R_v) + \sum_{i=u+1}^{v-1} p(R_i) \right\} \le C_{\max}(S^*) + \sum_{i=u+1}^{m-1} p(R_i).$$

Ha a 4. lépés feltételei fennállnak, akkor az 1. lépés miatt

$$\sum_{i=u+1}^{m-1} p(R_i) \le \left( \frac{m}{2} - 1 \right) p(R_m) \le \left( \frac{m}{2} - 1 \right) \frac{1}{m-1} R.$$

Ha nem, akkor páros $m$-re definíció szerint $p(R_m) > (1/(m-1))R$. Páratlan $m$ esetén, mivel $\rho_m$ a leghosszabb erőforrást felhasználó köteg, $p(R_m) \ge (1/m)R$. Ezért $u$ definíciója miatt, mindegyik esetben,

$$\sum_{i=u+1}^{m-1} p(R_i) = R - \sum_{i=1}^{u} p(R_i) - p(R_m) < \left( 1 - \left( \frac{1}{2} - \frac{1}{2x} \right) - \frac{1}{x} \right) R = \left( \frac{1}{2} - \frac{1}{2x} \right) R.$$

Tehát a tétel első állítása fennáll.

Az 5. Tételből következik, hogy a korlát éles. ■

Ha jobb teljesítményt akarunk, akkor el kell hagynunk a csoportosító eljárás szemléletet, lásd [2]. Ha az erőforrást igénylő munkákból gépenként egy köteget, az erőforrást nem igénylő munkákból legfeljebb két köteget készítünk, akkor lehetséges olyan algoritmust tervezni, ami létre hoz olyan $S_H$ ütemezést, melyre

$$\frac{C_{\max}(S_H)}{C_{\max}(S^*)} \le \begin{cases} \frac{3}{2} - \frac{1}{m+1}, & \text{ha } m \text{ páratlan,} \\ \frac{3}{2} - \frac{1}{m}, & \text{ha } m \text{ páros.} \end{cases}$$

Azonban tetszőleges $m$-re egy ilyen algoritmus leírása és elemzése túlságosan technikai, miközben a javulás jelentéktelen. Például $m = 3$-ra és $m = 4$-re $4/3$-ról $5/4$-re javul a közelítési hányados.

## *2.3. PTAS*

Ebben az alfejezetben mutatunk egy polinomiális idejű közelítő sémát (PTAS) a $\mathrm{PD}m \mid \mathrm{res}111 \mid C_{\max}$ feladatra, lásd [2]. Rögzített $m$-re és $\varepsilon \in ]0,1[$-re a séma futási ideje az input méretének polinomiális függvénye, de nem az $m$-nek és az $1/\varepsilon$-nak.

$$C := \max\left\{ \sum_{i=1}^{m} p(R_i), \max\{ p(Q_j) + p(R_j) \mid j = 1, \ldots, m \} \right\}$$

$S_G$ : A GT algoritmus outputja. Nyilvánvaló, hogy $C_{\max}(S_G) \geq C_{\max}(S^*) \geq C$.

A 6. Tétel bizonyításából következik, hogy $C_{\max}(S_G) < \frac{3}{2}C$, amiből következik, hogy

$C \leq C_{\max}(S^*) < \frac{3}{2}C$.

Ugyanis $C_{\max}(S_G) = \max\{C_1,\ldots,C_u,C_{u+1}\}$, és ha $C_{\max}(S_G) = \max\{C_1,\ldots,C_u\}$, akkor

$C_{\max}(S_G) \leq \sum_{i=1}^{u-1} p(R_i) + \max\{p(Q_1),\ldots,p(Q_{u-1}),p(Q_u)+p(R_u)\} \leq \left(\frac{1}{2}-\frac{1}{2x}\right)C + C = \left(\frac{3}{2}-\frac{1}{2x}\right)C < \frac{3}{2}C,$
ahol $x = m$, ha $m$ páratlan; $x = m-1$, ha $m$ páros.

Ha $u \neq m$ és $C_{\max}(S_G) = C_{u+1}$, akkor $C_{\max}(S_G) < \frac{3}{2}C$, mert ha

$C_{u+1} = \max\left\{p(R_{u+1})+p(Q_{u+1}),\sum_{i=1}^{u} p(R_i)+\sum_{i=u+1}^{m} p(R_i)\right\}$, akkor $C_{\max}(S_G) \leq C$, és ha

$C_{u+1} = \max_{u+2\leq v\leq m}\left\{p(Q_v)+p(R_v)+\sum_{i=u+1}^{v-1} p(R_i)\right\}$, akkor

$C_{\max}(S_G) \leq C + \sum_{i=u+1}^{m-1} p(R_i) \leq C + \left(\frac{1}{2}-\frac{1}{2x}\right)C = \left(\frac{3}{2}-\frac{1}{2x}\right)C < \frac{3}{2}C$.

$\tilde{\varepsilon} := \frac{\varepsilon}{16{,}5}$; $\delta_1,\delta_2,\ldots$ valós számsorozat, ahol $\delta_t := \left(\frac{\tilde{\varepsilon}}{m}\right)^{2^t}$ $(t = 1,2,\ldots)$.

$N^t := \left\{J_j \in N \mid \delta_t^2 C < p(J_j) \leq \delta_t C\right\}$ $(t = 1,2,\ldots)$.

Nyilvánvaló, hogy $N^1,N^2,\ldots$ páronként diszjunktak.

$\exists t_0 \in \left\{1,\ldots,\left\lceil \frac{m}{\tilde{\varepsilon}} \right\rceil\right\}$, hogy $p(N^{t_0}) \leq \frac{\tilde{\varepsilon}p(N)}{m} \leq \tilde{\varepsilon}C$, mert ha nem létezne, akkor

$p(N) > \left\lceil \frac{m}{\tilde{\varepsilon}} \right\rceil \frac{\tilde{\varepsilon}p(N)}{m}$, ami ellentmondás.

$\delta := \delta_{t_0}$.

Osszuk fel $N$-et $A_1$-re, $A_2$-re és $A_3$-ra:

$A_1 := \left\{J_j \mid p(J_j) > \delta C\right\}$ („nagy munkák”),

$A_2 := \left\{J_j \mid \delta C \geq p(J_j) > \delta^2 C\right\}$ („közepes munkák”),

$A_3 := \left\{J_j \mid \delta^2 C \geq p(J_j)\right\}$ („kis munkák”).

Nyilván $A_2 = N^{t_0}$, így $p(A_2) \leq \tilde{\varepsilon}C$.

Ezen kívül

(3) $\frac{\tilde{\varepsilon}}{m} \geq \delta \geq \left(\frac{\tilde{\varepsilon}}{m}\right)^{2^{\lceil m/(\tilde{\varepsilon})\rceil}}$.

A PTAS a feladatunkra nagy vonalakban: kezdjük az ütemezést csak a „nagy munkák” feldolgozásával, melyek kezdési idői a $\delta^2 C$ többszörösei. Aztán a hézagokat kitöltjük a erőforrást igénylő „kis munkákkal”. Végül a megmaradt munkák esetében a mohó módszert alkalmazzuk.

$\mathscr{S}_{\mathscr{R}}$: azon ütemezések halmaza, ahol a „nagy munkák” kezdési ideje a $\delta^2 C$ nem-negatív egész többszöröse.

**1. Lemma** (lásd [2]): $S_R^*$ egy ütemezés a $\mathrm{PD}m\,|\,\mathrm{res}111\,|\,C_{\max}$ feladatra, ami optimális az $\mathscr{S}_{\mathscr{R}}$-beli ütemezések között. Ekkor

$$C_{\max}\left(S_R^*\right) \le C_{\max}\left(S^*\right) + \frac{3}{2} m\delta C < \overline{C},$$

ahol $\overline{C} := \frac{3}{2} C(1+m\delta)$.

**Biz.:** A 2. Lemma bizonyításához hasonló, és megtalálható [2]-ben is. ■

Legyen $\Delta$ a $\left[0, \overline{C} - \delta C\right[$ intervallumban található, a $\delta^2 C$-nek nem-negatív egész többszöröseinek halmaza.

Legyen $\mathscr{S}_{\mathscr{B}}$ azon ütemezések osztálya, amely tartalmazza az összes lehetséges ütemezését a „nagy munkáknak”, ahol a kezdési idők $\Delta$-beliek. Az 1. Lemmából és a „nagy munkák” definíciójából következik, hogy van ilyen ütemezés.

Vegyünk egy tetszőleges $\mathscr{S}_{\mathscr{B}}$-beli $S_B$ ütemezést.

Legyen $\tau_1 < \tau_2 < \cdots < \tau_q$ a „nagy munkák” különböző kezdési <u>és</u> befejezési időinek növekvő sorozata. A befejezési időknek (természetesen) nem kell $\Delta$-belieknek lenniük.

(4) $q < 3\frac{m}{\delta} + 3m^2$,

ugyanis: $C_{\max}\left(S^*\right) \le C_{\max}\left(S^*\right) + \frac{3}{2} m\delta C < \overline{C}$, ezért $C_{\max}\left(S^*\right) < \frac{3}{2} C(1+m\delta)$.

Egy gépen maximum $b$ „nagy munka” van, így $b\delta C \le C_{\max}\left(S^*\right) < \frac{3}{2} C(1+m\delta)$.

Ebből következik, hogy $b < \frac{3}{2\delta}(1+m\delta)$, és mivel $q \le 2bm$, ezért

$$q < \frac{3}{\delta}(1+m\delta)m = \frac{3m}{\delta} + 3m^2.$$

$S_B$-re $I_0 := [0, \tau_1[, I_1 := [\tau_1, \tau_2[ \ldots, I_k := [\tau_k, \tau_{k+1}[, \ldots, I_{q-1} := [\tau_{q-1}, \tau_q[, I_q := [\tau_q, \infty[$.

Ebből következik, hogy $I_k$ ($0 \le k \le q$) belsejében a „nagy munkák” közül egyik sem kezdődik, vagy fejeződik be.

$\tilde{I}$ legyen index halmaz, ahol $k \in \tilde{I}$, pontosan akkor, ha $I_k$-ban nincs olyan „nagy munka”, melynek szüksége van az erőforrásra.

$\forall k \in \tilde{I}$-ra azok a gépek, melyeken nincs munka $I_k$-n alkotják a $\Lambda_k$ halmazt.

Mivel $\tau_q$ a befejezési ideje az utolsó „nagy munkáknak”, így $q \in \tilde{I}$.

$LP(S_B)$: lineáris program, ami hozzáveszi a „kis munkák” közül azokat, melyeknek szüksége van az erőforrásra, az $S_B$-hez nem-megszakíthatóan. Legyenek a gépek úgy beszámozva, hogy az első $m'$ gép legyen az, ahova rendeltünk ilyen „kis munkát”, azaz $R_i \cap A_3 = \varnothing$, ha $i > m'$. Legyen az $I_k$ intervallum hossza $i_k$, $k = 0,\ldots,q$.

$LP(S_B)$:

$x_{ik}$: az összmegmunkálási ideje azoknak a „kis munkáknak”, melyeknek szüksége van az erőforrásra, és az $M_i$ gépen az $I_k$ intervallumon munkáljuk meg.

Az ilyen „kis munkákat” csak olyan $I_k$ intervallumon tudjuk megmunkálni, melyre $k \in \tilde{I}$, és csak $\Lambda_k$-beli gépen.

(5) $\min \sum_{i=1}^{m'} x_{iq}$

(6) $\sum_{\substack{i=1 \\ M_i \in \Lambda_k}}^{m'} x_{ik} \le i_k, k \in \tilde{I},$

(7) $\sum_{\substack{k \in \tilde{I} \\ M_i \in \Lambda_k}} x_{ik} = p(R_i \cap A_3), i = 1,\ldots,m',$

(8) $x_{ik} \ge 0, i = 1,\ldots,m', k \in \tilde{I}.$

(6): Azoknak a „kis munkáknak”, melyeknek szüksége van az erőforrásra, és az $I_k$ intervallumon munkáljuk meg, az összmegmunkálási ideje nem lehet több, mint az $I_k$ intervallum hossza ($i_k$), mert nem lehet átfedés erőforrást igénylő munkák között.

(7): Minden olyan „kis munkát”, melynek szüksége van az erőforrásra, hozzárendelünk a megfelelő géphez az $I_k$ intervallumban.

Az $LP(S_B)$-nek mindig van megengedett megoldása, mert $q \in \tilde{I}$ és $i_q = \infty$.

(5): Ha $\bar{S}_B$ jelöli a kapott ütemezést, akkor $C_{\max}(\bar{S}_B) = \tau_q + \sum_{i=1}^{m'} x_{iq}$, ahol $\tau_q$ konstans, így $\sum_{i=1}^{m'} x_{iq}$-t kell minimalizálni.

$\boldsymbol{A = A(\varepsilon, m)}$ **algoritmus** (lásd [2])

*Input:* Egy példánya a $\mathrm{PD}m \mid \mathrm{res}111 \mid C_{\max}$ feladatnak, $\varepsilon > 0$

*Output:* Egy $S_\varepsilon$ heurisztikus ütemezés

1. Határozzuk meg $\varepsilon$-ból $\tilde{\varepsilon}$-ot és $\delta$-t. Osszuk fel $N$-et $A_1$-re, $A_2$-re és $A_3$-ra („nagy munkákra, „közepes munkákra” és „kis munkákra”). Ha szükséges a gépek átszámozása úgy, hogy $R_i \cap A_3 \neq \varnothing$, ha $1 \le i \le m'$ és $R_i \cap A_3 = \varnothing$, ha $i > m'$.
2. $\mathscr{S}_{\mathscr{B}}$ megkeresése. $\mathscr{S}_{\mathscr{B}}$: azon ütemezések osztálya, amely tartalmazza az összes lehetséges ütemezését a „nagy munkáknak”, ahol a kezdési idők $\Delta$-beliek. $\forall S_B \in \mathscr{S}_{\mathscr{B}}$ ütemezéshez a többi munkát is hozzávesszük a következő módon:
   2.1. Egy $\alpha_{ik}$ optimális megoldás megkeresése az $LP(S_B)$ lineáris programnak ($i = 1, \ldots, m'$, $k \in \tilde{I}$).
   2.2. Minden $I_k$, $k \in \tilde{I}$ intervallumra, rendeljük hozzá $R_i \cap A_3$-beli legfeljebb $\alpha_{ik}$ teljes megmunkálási idejű munkákat az $M_i$ géphez az $I_k$ intervallumban ($i = 1, \ldots, m'$). A munkák bármilyen sorrendben jöhetnek és nem-megszakíthatóan rendeljük a gépekhez a gépek számozásának sorrendjében:
   Az $M_1$ gép kezd a $\tau_k$-nál, és $i = 1, \ldots, m' - 1$-re az $M_{i+1}$ gép közvetlenül azután kezd, hogy $M_i$ végzett.
   Ha egy munka nem fér az elérhető intervallumba, ideiglenesen tegyük félre.
   2.3. $\forall i$-re $i = 1$-től $i = m$-ig $\rho_i$ köteg készítése a megmaradt $R_i$-beli munkákból, azaz az $R_i \cap A_2$-beli munkákból és azokból az $R_i \cap A_3$-beli munkákból, melyeket 2.2.-nél félretettünk. Kezdjük a $\rho_1$-es köteget az $M_1$ gépen $\tau_q + \sum_{i=1}^{m'} \alpha_{iq}$ időponttól. Ezután

$\forall i = 1,\ldots,m-1$-re kezdjük a $\rho_{i+1}$-es köteget az $M_{i+1}$ gépen közvetlenül azután, hogy a $\rho_i$ köteg befejeződött az $M_i$ gépen. Ha $\rho_i$ üres, akkor hagyjuk figyelmen kívül $M_i$-t.

2.4. $i = 1,\ldots,m$-re a maradék $Q_i$-beli munkákat mohó módon rendeljük $M_i$-hez: tetszőleges sorrendben, nem-megszakíthatóan, a legkorábbi olyan időpontban kezdve, hogy a munkák beleférjenek a megmaradt hézagokba az aktuális ütemezésben.

3. A kapott ütemezésekből válasszuk ki azt, amelyiknek a legkisebb a teljes átfutási ideje. Legyen $S_\varepsilon$ a kapott ütemezés. Stop.

**7. Tétel** (lásd [2]): Az $A = A(\varepsilon, m)$ algoritmus PTAS a $PDm\,|\,res111\,|\,C_{\max}$ feladatra.

**Biz.:** Be fogjuk bizonyítani, hogy az algoritmus által kapott ütemezés teljes átfutási ideje $\leq (1+\varepsilon)C_{\max}(S^*)$, és konstans $\varepsilon$-ra és $m$-re polinomidőben fut.

Az 1. Lemmából következik, hogy $C_{\max}(S_R^*) < \overline{C}$. Mivel egy „nagy munka” megmunkálási ideje nagyobb, mint $\delta C$, ezért $S_R^*$-ban a „nagy munkák” kezdési idői $\Delta$-beliek. Ezért $\exists S_B^* \in \mathscr{S}_{\mathscr{B}}$, hogy az $S_B^*$-beli munkák (mind „nagy munkák”) kezdési idői megegyeznek az $S_R^*$-ban a „nagy munkák” kezdési időivel. Ebből következik, hogy a 2. lépés elsősorban az $S_B^*$-ra vonatkozik.

Nézzük most $LP(S_B^*)$-ot. Jelölje $\overline{S}_B^*$ azt az ütemezést, amihez a „nagy munkák” mellett a „kicsi” erőforrást igénylő munkákat is hozzárendeltük (2.1. és 2.2. lépés). Ekkor $C(S_B^*) := C_{\max}(\overline{S}_B^*) = \tau_q + \sum_{i=1}^{m'} \alpha_{iq}$, ahol $\alpha_{ik}$ optimális megoldása a lineáris programnak ($i = 1,\ldots,m'$, $k \in \widetilde{I}$). Ebből következik, hogy $C(S_B^*) \leq C_{\max}(S_R^*)$.

Mivel $\left|\widetilde{I}\right| \leq q$, az optimális megoldásban legfeljebb $m' + q$ nem-nulla változó van.

Hívjuk a pozitív $\alpha_{ik}$ értékeket „nem-rés értéknek” („non-split value”), ha $\alpha_{ik'} = 0$ $k \neq k'$-re, különben „rés értéknek” („split value”).

Ha $\alpha_{ik}$ „nem-rés érték”, akkor $\alpha_{ik} = p(R_i \cap A_3)$. Ezért minden $R_i \cap A_3$-beli munkát nem-megszakítható módon hozzárendelünk az $I_k$ intervallumhoz a 2.2. lépésben.

Ha $\alpha_{ik}$ „rés érték”, akkor az $I_k$-ban az $M_i$-n található „kicsi” erőforrást igénylő munkák(, melyeket nem-megszakíthatóan dolgozunk fel itt) összmegmunkálási ideje

nagyobb, mint $\alpha_{ik} - \delta^2 C$. Legfeljebb $m' + q$ „rés érték” van. (3)-ból és (4)-ből következik, hogy azoknak a „kicsi” erőforrást igénylő munkáknak az összmegmunkálási idejének, melyeket $C(S_B^*)$-ig nem lehet feldolgozni, egy felső korlátja

(9) $(m+q)\delta^2 C < m\delta^2 C + 3m\delta C + 3m^2\delta^2 C \le \frac{\tilde{\varepsilon}^2 C}{m} + 3\tilde{\varepsilon}^2 C < 7\tilde{\varepsilon}C$.

Jelölje $S$ azt az ütemezést, amit az $S_B^*$-ból kapunk a 2.4. lépés után. Legyen $M_r$ az a gép, ahol $S$ befejeződik. Legyen az $M_r$ gépen a $C(S_B^*)$ után az $A_2$-beli munkák összmegmunkálási ideje $r_m$, az $R_r \cap A_3$-beli munkáké $r_{sr}$, és végül a $Q_r \cap A_3$-belieké $r_{sq}$. Legyen a $C(S_B^*)$ és a $C_{\max}(S)$ közti szünet ideje $r_i$. Ekkor

$C_{\max}(S) = C(S_B^*) + r_{sq} + r_{sr} + r_m + r_i$.

$C(S_B^*)$ után az $M_r$ gépen csak akkor van szünet, ha ugyanekkor egy erőforrást igénylő munkát dolgoz fel egy másik gép. De $C(S_B^*)$ után csak „közepes” és „kicsi” erőforrást igénylő munkát dolgozunk fel. (9)-ből következik, hogy $r_{sr} + r_m + r_i \le p(A_2) + 7\tilde{\varepsilon}C$, és mivel $p(A_2) \le \tilde{\varepsilon}C$, ezért $C_{\max}(S) \le C(S_B^*) + r_{sq} + 8\tilde{\varepsilon}C$.

A 2.2. lépés után legfeljebb $q+1$ szünet van gépenként 0 és $C(S_B^*)$ között, és ez a szám a 2.3. és a 2.4. lépés után se növekszik. Amikor hozzávesszük a „kicsi” erőforrást nem igénylő munkákat ezekbe a hézagokba, két esetet kell tekintenünk. Először, ha $M_r$ gépre minden „kicsi” erőforrást nem igénylő munka belefér a hézagokba, akkor $r_{sq} = 0$. Ebben az esetben nem növekszik a teljes átfutási idő, és $C_{\max}(S) \le C(S_B^*) + r_{sq} + 8\tilde{\varepsilon}C$-ből és $C(S_B^*) \le C_{\max}(S_R^*)$-ból következik, hogy

$C_{\max}(S) \le C_{\max}(S_R^*) + 8\tilde{\varepsilon}C$.

Különben, ha „kicsi” erőforrást nem igénylő munkákat kell feldolgozni $C(S_B^*)$ után az $M_r$ gépen, a $C(S_B^*)$ előtti összes hézag hossza most kisebb, mint $\delta^2 C$. (3)-ból és (4)-ből következik, hogy a 0 és $C(S_B^*)$ közötti hézagok összhosszának egy felső korlátja

$(q+1)\delta^2 C < \left(3\frac{m}{\delta} + 3m^2 + 1\right)\delta^2 C \le 3\tilde{\varepsilon}C + 3\tilde{\varepsilon}^2 C + \frac{\tilde{\varepsilon}}{m}C < 7\tilde{\varepsilon}C$.

Ebből következik, hogy $p(N_r) \ge C(S_B^*) + r_{sq} - 7\tilde{\varepsilon}C$.

Nyilván $C_{\max}(S_R^*) \geq p(N_r)$, ezért

$C_{\max}(S) \leq (C(S_B^*) + r_{sq}) + 8\tilde{\varepsilon}C \leq (C_{\max}(S_R^*) + 7\tilde{\varepsilon}C) + 8\tilde{\varepsilon}C = C_{\max}(S_R^*) + 15\tilde{\varepsilon}C$.

Jól látszik, hogy a $C_{\max}(S) \leq C_{\max}(S_R^*) + 15\tilde{\varepsilon}C$ egyenlőtlenség érvényes az 1. esetben is.

$C \leq C_{\max}(S^*) < (3/2)C$-ből, (3)-ból és az 1. Lemmából következik, hogy

$C_{\max}(S) \leq C_{\max}(S_R^*) + 15\tilde{\varepsilon}C \leq C_{\max}(S^*) + \frac{3}{2}m\delta C + 15\tilde{\varepsilon}C \leq C_{\max}(S^*) + \frac{3}{2}\tilde{\varepsilon}C + 15\tilde{\varepsilon}C$

$= C_{\max}(S^*) + 16{,}5\tilde{\varepsilon}C \leq C_{\max}(S^*)(1 + 16{,}5\tilde{\varepsilon}) = C_{\max}(S^*)(1+\varepsilon)$.

Most már csak azt kell megmutatni, hogy az algoritmus fix $m$-re és $\varepsilon$-ra polinomidőben fut. Egyrészt az $\mathscr{S}_{\mathscr{B}}$-beli ütemezések maximális száma nem nagyobb, mint $O\left((1/\delta^2)^{m/\delta}\right)$. Másrészt a feltételek és változók száma az $LP(S_B)$ lineáris programban nem nagyobb, mint $m$-nek egy függvénye: (5)-ben a változók száma $m'$, (6)-ban a feltételek száma $|\tilde{I}|$, a változók száma nem nagyobb, mint $m'|\tilde{I}|$, (7)-ben a feltételek száma $m'$, a változók száma nem nagyobb, mint $m'|\tilde{I}|$, és végül (8)-ban a feltételek és változók száma $m'|\tilde{I}|$, ezen kívül $m' \leq m$, $|\tilde{I}| \leq q$, (4) és (3).

Tehát az $A = A(\varepsilon, m)$ algoritmus PTAS. ■

# 3. Több erőforrás

Ebben a fejezetben a $\mathrm{PD2}\,|\,\mathrm{res1}\cdot\cdot\,|\,C_{\max}$ és a $\mathrm{PD2}\,|\,\mathrm{res211}\,|\,C_{\max}$ feladattal, valamint ezek általánosításaival foglalkozunk, lásd [3]. Mivel a $\mathrm{PD3}\,|\,\mathrm{res111}\,|\,C_{\max}$ feladat NP-nehéz, ezért a fejezetben a gépek száma végig 2 lesz.

A 3.1. alfejezetben a $\mathrm{PD2}\,|\,\mathrm{res1}\cdot\cdot\,|\,C_{\max}$, a 3.2. alfejezetben a $\mathrm{PD2}\,|\,\mathrm{res211}\,|\,C_{\max}$ feladatra lineáris idejű algoritmust adunk. A 3.3. alfejezetben megmutatjuk, hogy a $\mathrm{PD2}\,|\,\mathrm{res222}\,|\,C_{\max}$ és a $\mathrm{PD2}\,|\,\mathrm{res311}\,|\,C_{\max}$ feladat NP-nehéz.

## *3.1. Egyetlen tetszőleges méretű erőforrás*

Ebben az alfejezetben a $\mathrm{PD2}\,|\,\mathrm{res1}\cdot\cdot\,|\,C_{\max}$ feladattal foglalkozunk, ahol az $N_1$-beli munkákat a $M_1$ gépen dolgozzuk fel és a $N_2 = N \setminus N_1$-beli munkákat a $M_2$ gépen, lásd [3]. Az erőforrásból $\sigma$ egység bármikor elérhető, és a $J_j$ munkának $\rho_j$ egységre van szüksége az erőforrásból a végrehajtása során. Megmutatjuk, hogy a feladat megoldható $\mathrm{O}(n\log n)$ idő alatt.

$p(Q) := \sum_{J_j \in Q} p_j$, ha $Q \subseteq N_i$ és $Q \neq \varnothing$

és $p(\varnothing) := 0$ . $J_j \in N_i$ esetén $p(J_j) := p(\{J_j\}) = p_j$ .

$N_1$-re és $N_2$-re $N_i^r = \{J_j \mid J_j \in N_i, \rho_j = r\}$ $(i \in \{1,2\}, r \in \{0,1,\ldots,\sigma\})$.

$\forall S$-re $C_{\max}(S) \geq LB_i := \sum_{r=0}^{\sigma} p(N_i^r)$ $(i \in \{1,2\})$.

$LB_m := \max\{LB_1, LB_2\}$.

$C_{\max}(S) \geq LB_{uv} := \sum_{r=u}^{\sigma} p(N_1^r) + \sum_{r=v}^{\sigma} p(N_2^r)$, ha $u$ és $v$ egész, $1 \leq u,v \leq \sigma$ és $u+v > \sigma$ .

$LB_r := \max_{\substack{1\leq u,v\leq\sigma \\ u+v>\sigma}} \{LB_{uv}\}$.

**B algoritmus** (lásd [3])

*Input:* Egy példánya a $\mathrm{PD2}\,|\,\mathrm{res1}\cdot\cdot\,|\,C_{\max}$ feladatnak

*Output:* Egy optimális $S^*$ ütemezés

1. Rendezzük mindegyik gépen a munkákat az erőforrás igényük szerinti nem-csökkenő sorrendbe. Definiáljuk $N_i^r$-eket $(i \in \{1,2\}, r \in \{0,1,\ldots,\sigma\})$.
2. Számítsuk ki $p(N_i^r)$-eket $(i \in \{1,2\}, r \in \{0,1,\ldots,\sigma\})$.

3. Számítsuk ki az $LB_m$ és $LB_r$ alsó korlátokat. Ha $LB_r = LB_{uv}$ különböző $u$ és $v$ párokra, válasszuk azt, amelyikhez a legkisebb $u$ tartozik.
4. Az $M_1$ gépen $N_1^0, N_1^1, \ldots, N_1^\sigma$ sorrendben jönnek a munkahalmazok, az $M_2$ gépen $N_2^\sigma, N_2^{\sigma-1}, \ldots, N_2^0$ sorrendben.
5. Az $M_2$ gépen 0 időpontnál kezdünk szünet nélkül. Ha $LB_1 \geq LB_r = LB_{uv}$ akkor az $M_1$ gépen is 0 időpontnál kezdünk szünet nélkül. Egyébként a 0 időponttól elkezdjük $N_1^0, N_1^1, \ldots, N_1^{u-1}$-et szünet nélkül, majd a $\sum_{r=v}^{\sigma} p\left(N_2^r\right)$-től folytatjuk az $N_1^u, N_1^{u+1}, \ldots, N_1^\sigma$-t (szünet nélkül). $S^*$: az eredményül kapott ütemezés. Stop.

Az 1. lépés $\mathrm{O}(n \log n)$ ideig tart. Mivel a nemüres $N_1^r$ és $N_2^r$ halmazok száma nem haladja meg az $n$-et, a többi lépés $\mathrm{O}(n)$ ideig tart.

**4. Példa:** $\sigma := 3$, $N := \{J_1, \ldots, J_{20}\}$, $J_1, \ldots, J_{20}$ megmunkálási idői legyenek rendre 2, 3, 6, 1, 7, 5, 3, 6, 1, 2, 3, 3, 1, 3, 3, 4, 2, 1, 2, 6. $N_1^3 := \{J_1, J_2, J_3\}$, $N_1^2 := \{J_4, J_5, J_6\}$, $N_1^1 := \{J_7, J_8\}$, $N_1^0 := \{J_9, J_{10}, J_{11}\}$, $N_2^3 := \{J_{12}, J_{13}\}$, $N_2^2 := \{J_{14}, J_{15}, J_{16}\}$, $N_2^1 := \{J_{17}, J_{18}\}$, $N_2^0 := \{J_{19}, J_{20}\}$.

Az algoritmus csak a munkahalmazok sorrendjét határozza meg (a 4. lépésben), legyen a munkahalmazokon belül a munkák sorrendje tetszőleges. Legyen az $N_1^3$-beli munkák sorrendje $(J_3, J_1, J_2)$, az $N_1^2$-belieké $(J_4, J_5, J_6)$, az $N_1^1$-belieké $(J_8, J_7)$, az $N_1^0$-belieké $(J_9, J_{10}, J_{11})$, az $N_2^3$-belieké $(J_{12}, J_{13})$, az $N_2^2$-belieké $(J_{16}, J_{15}, J_{14})$, az $N_2^1$-belieké $(J_{17}, J_{18})$ és az $N_2^0$-belieké $(J_{19}, J_{20})$.

$p\left(N_1^3\right) = 11$, $p\left(N_1^2\right) = 13$, $p\left(N_1^1\right) = 9$, $p\left(N_1^0\right) = 6$,

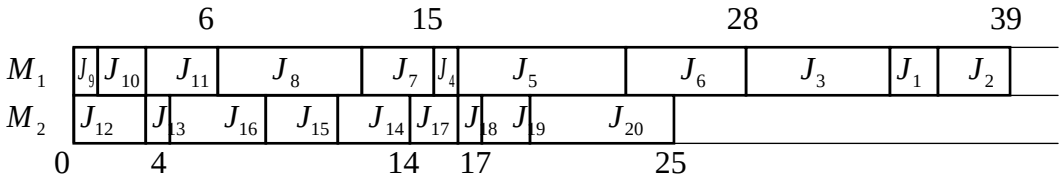
$p\left(N_2^3\right) = 4$, $p\left(N_2^2\right) = 10$, $p\left(N_2^1\right) = 3$ és $p\left(N_2^0\right) = 8$.

$LB_1 = 39$ és $LB_2 = 25$, ezért $LB_m = 39$.

$LB_r = LB_{2,2} = 38$.

Mivel $39 = LB_1 \geq LB_r = LB_{2,2} = 38$, ezért mindkét gépen 0 időpontnál kezdünk szünet nélkül.

$S^*$:




**8. Tétel** (lásd [3]): Az $S^*$ valóban optimális.

**Biz.:** Először megmutatjuk, hogy $S^*$ megengedett. Ha $LB_1 \geq LB_r$, akkor egyik gépen sincs szünet, így akkor lenne nem-megengedett, ha létezne olyan $u$ és $v$, amelyre $u+v>\sigma$ és

$\sum_{r=0}^{u-1} p\left(N_1^r\right) < \sum_{r=v}^{\sigma} p\left(N_2^r\right)$, és átfedés van $N_1^u$ és $N_2^v$ között. Ekkor ellentmondásra jutunk:

$$LB_r \geq LB_{uv} = \sum_{r=v}^{\sigma} p\left(N_2^r\right) + \sum_{r=u}^{\sigma} p\left(N_1^r\right) > \sum_{r=0}^{u-1} p\left(N_1^r\right) + \sum_{r=u}^{\sigma} p\left(N_1^r\right) = LB_1 .$$

Ezért, ha $LB_1 \geq LB_r$, akkor $S^*$ megengedett és $C_{\max}\left(S^*\right) = LB_m$.

Ha $LB_1 < LB_r = LB_{uv}$, akkor az $u$ és $v$ választása garantálja a megengedettséget, és $C_{\max}\left(S^*\right) = \max\left\{LB_2, LB_r\right\}$.

Tehát $S^*$ valamelyik alsó korlátot eléri, így $S^*$ optimális. ■

## *3.2. Két egységnyi méretű erőforrás*

Most a $\mathrm{PD2}\,|\,\mathrm{res211}\,|\,C_{\max}$ feladattal foglalkozunk, ahol az $N_1$-beli munkákat a $M_1$ gépen dolgozzuk fel és a $N_2 = N \setminus N_1$-beli munkákat a $M_2$ gépen, lásd [3]. A 2 erőforrásból 1 egység bármikor elérhető. Megmutatjuk hogy a feladat megoldható $\mathrm{O}(n)$ idő alatt.

$N_i = N_i^0 \cup N_i^1 \cup N_i^2 \cup N_i^{1,2}$ ($i \in \{1,2\}$), ahol az $N_i^0$-beli munkáknak nincs szükségük egyik erőforrásra sem, az $N_i^1$-beli munkáknak az 1-es számú erőforrásból, az $N_i^2$-beli munkáknak a 2-es számú erőforrásból 1 egységre van szükségük, valamint az $N_i^{1,2}$-beli munkáknak mindkét erőforrásból szükségük van 1 egységre.

$\forall S$-re $C_{\max}(S) \geq LB_i := p(N_i)$ ($i \in \{1,2\}$).

$LB_m := \max\left\{LB_1, LB_2\right\}$.

$C_{\max}(S) \geq LB_r$, ahol

$$LB_r := p\left(N_1^{1,2}\right) + p\left(N_2^{1,2}\right) + \max\left\{p\left(N_1^1\right) + p\left(N_1^2\right), p\left(N_2^1\right) + p\left(N_2^2\right), p\left(N_1^1\right) + p\left(N_2^1\right), p\left(N_1^2\right) + p\left(N_2^2\right)\right\}.$$

**D algoritmus** (lásd [3])

*Input:* Egy példánya a $\mathrm{PD2}\,|\,\mathrm{res211}\,|\,C_{\max}$ feladatnak

*Output:* Egy optimális $S^*$ ütemezés

1. Ha szükséges, számozzuk át a gépeket és/vagy az erőforrásokat úgy, hogy $p\left(N_2^2\right)=\max\left\{p\left(N_1^1\right), p\left(N_1^2\right), p\left(N_2^1\right), p\left(N_2^2\right)\right\}$ legyen.
2. Az $M_1$-es gépen a 0 időponttól $N_1^1$, majd $N_1^0$ szünet nélkül. Az $M_2$-es gépen a 0 időponttól $N_2^2$, majd $N_2^{1,2}$, $N_2^1$, és végül $N_2^0$ szünet nélkül.
3. Az $M_1$-es gépen a $\max\left\{p\left(N_1^1\right)+p\left(N_1^0\right), p\left(N_2^2\right)+p\left(N_2^{1,2}\right)\right\}$ időponttól $N_1^2$, majd a
   $\max\left\{p\left(N_1^1\right)+p\left(N_1^0\right)+p\left(N_1^2\right), p\left(N_2^2\right)+p\left(N_2^{1,2}\right)+p\left(N_1^2\right), p\left(N_2^2\right)+p\left(N_2^{1,2}\right)+p\left(N_2^1\right)\right\}$
   időponttól $N_1^{1,2}$. $S^*$: az eredményül kapott ütemezés. Stop.

Könnyű ellenőrizni, hogy az algoritmus $\mathrm{O}(n)$ ideig tart.

**5. Példa:** $N:=\{J_1,\ldots,J_{20}\}$, $J_1,\ldots,J_{20}$ megmunkálási idői legyenek ugyanannyiak, mint a 4. Példában. Átszámozás után: $N_2^{1,2}:=\{J_1,J_2,J_3\}$, $N_2^2:=\{J_4,J_5,J_6\}$, $N_2^1:=\{J_{17},J_{18}\}$, $N_2^0:=\{J_{19},J_{20}\}$, $N_1^{1,2}:=\{J_{12},J_{13}\}$, $N_1^2:=\{J_{14},J_{15},J_{16}\}$, $N_1^1:=\{J_7,J_8\}$, $N_1^0:=\{J_9,J_{10},J_{11}\}$.

Legyen az $N_1^{1,2}$-beli munkák sorrendje $(J_{12},J_{13})$, az $N_1^2$-belieké $(J_{15},J_{14},J_{16})$, az $N_1^1$-belieké $(J_8,J_7)$, az $N_1^0$-belieké $(J_9,J_{11},J_{10})$, az $N_2^{1,2}$-belieké $(J_2,J_1,J_3)$, az $N_2^2$-belieké $(J_5,J_6,J_4)$, az $N_2^1$-belieké $(J_{18},J_{17})$ és az $N_2^0$-belieké $(J_{19},J_{20})$.

$p\left(N_1^{1,2}\right)=4$, $p\left(N_1^2\right)=10$, $p\left(N_1^1\right)=9$, $p\left(N_1^0\right)=6$,

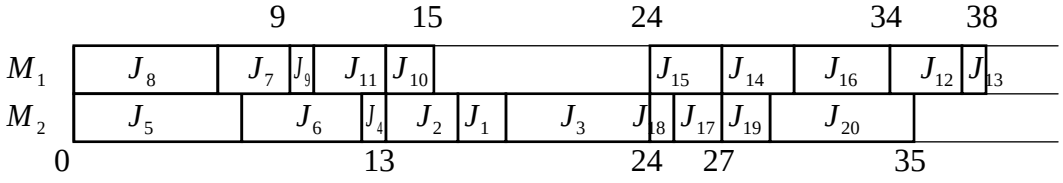
$p\left(N_2^{1,2}\right)=11$, $p\left(N_2^2\right)=13$, $p\left(N_2^1\right)=3$ és $p\left(N_2^0\right)=8$.

$\max\left\{p\left(N_1^1\right)+p\left(N_1^0\right), p\left(N_2^2\right)+p\left(N_2^{1,2}\right)\right\}=\max\{15,24\}=24$.

$\max\left\{p\left(N_1^1\right)+p\left(N_1^0\right)+p\left(N_1^2\right), p\left(N_2^2\right)+p\left(N_2^{1,2}\right)+p\left(N_1^2\right), p\left(N_2^2\right)+p\left(N_2^{1,2}\right)+p\left(N_2^1\right)\right\}$

$=\max\{25,34,27\}=34$.

$S^*$:



**9. Tétel** (lásd [3]): Az $S^*$ valóban optimális.

**Biz.:** $S^*$ nyilván megengedett. Az $M_2$ $p(N_2) \le LB_m$ idő alatt végez. Az $M_1$

$$\max\left\{p(N_1^1)+p(N_1^0)+p(N_1^2),\ p(N_2^2)+p(N_2^{1,2})+p(N_1^2),\ p(N_2^2)+p(N_2^{1,2})+p(N_2^1)\right\}+p(N_1^{1,2})$$

$$=\max\left\{p(N_1),\ p(N_2^{1,2})+p(N_1^{1,2})+\max\left\{p(N_2^2)+p(N_1^2),\ p(N_2^2)+p(N_2^1)\right\}\right\}$$

$\le \max\{LB_m, LB_r\}$ idő alatt végez.

Tehát $S^*$ valamelyik alsó korlátot eléri, így $S^*$ optimális. ■

## *3.3. Két és három erőforrás: NP-nehéz*

A B és a D algoritmus nem általánosítható több erőforrás vagy nem-egység méretű 2 erőforrás esetére. Megmutatjuk, hogy 2 erőforrás esetén, melyek mérete 2 a $\mathrm{PD2}\,|\,\mathrm{res222}\,|\,C_{\max}$ NP-nehéz. Ezen kívül a D algoritmus nem általánosítható a több, mint 2 egység méretű erőforrás esetére. Megmutatjuk, hogy 3 egységnyi méretű erőforrás esetén a $\mathrm{PD2}\,|\,\mathrm{res311}\,|\,C_{\max}$ NP-nehéz, lásd [3]. Mindkét feladat esetén egy munka egynél több erőforrást is használhat.

**10. Tétel** (lásd [3]): A $\mathrm{PD2}\,|\,\mathrm{res222}\,|\,C_{\max}$ feladat gyengén NP-nehéz.

**Biz.:** Tetszőleges Partícióra definiáljuk a következő példányát a $\mathrm{PD2}\,|\,\mathrm{res222}\,|\,C_{\max}$ feladatnak $n = t+8$ munkával.

Az $M_1$-en a munkáknak 2 csoportja van:

$U$-munkák ($U_i, i = 1,2,\ldots,t$) és $X$-munkák ($X_1$, $X_2$ és $X_3$).

Az $M_2$-n is a munkáknak 2 csoportja van:

$Y$-munkák ($Y_1$, $Y_2$ és $Y_3$) és $Z$-munkák ($Z_1$ és $Z_2$).

A megmunkálási idők:

$p(U_i) := 5e_i$, $i = 1,\ldots,t$,

$p(X_1) := p(X_2) := p(X_3) := 1$,

$p(Y_1) := p(Y_2) := p(Y_3) := 3$,

$p(Z_1) := p(Z_2) := 5E-3$.

Az $U$-munkáknak nincs szükségük erőforrásra, a $Z$-munkákra mind szükségük van mindkét erőforrásból 2 egységre.

Az $X$ - és $Y$ -munkáknak a következő szükségletük van az erőforrásokból:

| Munka | 1. erőforrás | 2. erőforrás |
|---|---|---|
| $X_1$ | 0 | 2 |
| $X_2$ | 1 | 1 |
| $X_3$ | 2 | 0 |
| $Y_1$ | 2 | 0 |
| $Y_2$ | 1 | 1 |
| $Y_3$ | 0 | 2 |

Bizonyítandó: létezik $S_0$, melyre $C_{\max}(S_0) \le y = 10E + 3$ pontosan akkor, ha a Partíciónak van megoldása.

Tegyük fel, hogy a Partíciónak van megoldása, és $T_1$ és $T_2$ az elvárt indexhalmazok. Olyan $S_0$ ütemezés, melyre $C_{\max}(S_0) = y$, a következőképpen írható le. Egyik gépen sincs szünet a $[0, y]$ intervallumban. Az $M_2$ gép az $(Y_1, Z_1, Y_2, Z_2, Y_3)$ sorrendben, az $M_1$ gép az $(X_1, \pi_1(U), X_2, \pi_2(U), X_3)$ sorrendben dolgozza fel a munkákat, ahol $\pi_1(U)$ a $T_1$-ből vett indexű, $\pi_2(U)$ a $T_2$-ből vett indexű $U$ -munkák tetszőleges sorozata.

Tegyük most fel, hogy $S_0$ létezik. Ebből következik, hogy $C_{\max}(S_0) = y$, és egyik gépen sincs szünet a $[0, y]$ intervallumban. Az erőforrás igények miatt az $M_1$ gép csak azalatt dolgozhatja fel az $X_k$ munkát, amíg az $M_2$ gép az $Y_k$ munkát dolgozza fel $k \in \{1,2,3\}$-ra. Mivel az összes megmunkálási idő egész, ezért $S_0$-ban az összes kezdési és befejezési idő egész.

Először megmutatjuk, hogy $S_0$-ban az $M_2$ gépen nem lehet két $Y$ -munkát közvetlenül egymás után feldolgozni. Tegyük fel, hogy $Y_p$-t és $Y_q$-t közvetlenül egymás után dolgozzuk fel (ebben a sorrendben), ahol $p, q \in \{1,2,3\}$ és $p \neq q$. Az $M_1$ azalatt dolgozza fel az $X_p$-t, amíg az $M_2$ az $Y_p$-t dolgozza fel, és az $M_1$ azalatt dolgozza fel az $X_q$-t, amíg az $M_2$ az $Y_q$-t dolgozza fel. Nincs hézag $X_p$ és $X_q$ között, mert egy lehetséges hézag hossza egy egész szám lehetne 1-től 4-ig, de nincs olyan $U$ -munka, ami beleillene egy ilyen hézagba. Legyen az $X_p$-t megelőző $U$ -munkák indexeinek halmaza $Q_1$, és az $X_q$ után következő $U$ -munkák indexeinek halmaza $Q_2$. Az $M_2$ gépen a munkák lehetséges sorrendje $(Z_1, Y_p, Y_q, Z_2, Y_r)$, ahol $Y_r$ egy $Y_p$-től és $Y_q$-tól különböző $Y$ -munka. Ekkor az $M_1$ gépen a munkák lehetséges sorrendje

$(\varphi_1(U), X_p, X_q, \varphi_2(U), X_r)$, ahol $\varphi_1(U)$ a $Q_1$-ből vett indexű, $\varphi_2(U)$ a $Q_2$-ből vett indexű $U$-munkák tetszőleges sorozata. Ekkor az összmegmunkálási ideje $\varphi_1(U)$-nak $5E-1$, míg $\varphi_2(U)$-nak $5E+1$. Ez ellentmondás, mert mindegyik $U$-munkának a hossza az 5 többszöröse.

Tehát az $M_2$ gépen a munkák sorrendje $(Y_p, Z_1, Y_q, Z_2, Y_r)$ valamilyen $p$, $q$ és $r$ hármasra. Ebből következik, hogy két egymást követő $X$-munkát $U$-munkák választanak szét. Legyen az $X_p$-t és $X_q$-t szétválasztó $U$-munkák indexeinek halmaza $Q_1$, és az $X_q$-t és $X_r$-t szétválasztó $U$-munkák indexeinek halmaza $Q_2$. Ekkor az $M_1$ gépen a munkák sorrendje $(X_p, \varphi_1(U), X_q, \varphi_2(U), X_r)$, ahol $\varphi_1(U)$ a $Q_1$-ből vett indexű, $\varphi_2(U)$ a $Q_2$-ből vett indexű $U$-munkák tetszőleges sorozata.

Tegyük fel, hogy $\varphi_1(U)$ hossza kevesebb, mint $5E$. Ekkor az $M_1$ gépen $\varphi_1(U)$ $1+(5E-5)$-nél nem később fejeződik be, míg az $X_q$ legkorábban $5E$-nél kezdődhet.

Tegyük fel, hogy $\varphi_1(U)$ hossza több, mint $5E$. Ekkor az $M_1$ gépen $\varphi_1(U)$ $1+(5E+5)$-nél nem korábban fejeződik be, míg az $X_q$ legkésőbb $5E+3$-nál kezdődhet.

Tehát $\varphi_1(U)$ és $\varphi_2(U)$ hossza $5E$, ezért a Partíciónak van megoldása: $T_1 := Q_1$ és $T_2 := Q_2$. ■

**11. Tétel** (lásd [3]): A $PD2|res311|C_{\max}$ feladat is gyengén NP-nehéz.

**Biz.:** Tetszőleges Partícióra definiáljuk a következő példányát a $PD2|res311|C_{\max}$ feladatnak $n = t+8$ munkával.

A 10. Tétel bizonyításában használt 4 csoportot vesszük ugyanolyan megmunkálási időkkel.

Az $U$-munkáknak nincs szükségük erőforrásra, a $Z$-munkáknak mind szükségük van mindhárom erőforrásra.

Az $X$ - és $Y$ -munkáknak a következő szükségletük van az erőforrásokból:

| Munka | 1. erőforrás | 2. erőforrás | 3. erőforrás |
|---|---|---|---|
| $X_1$ | 1 | 1 | 0 |
| $X_2$ | 1 | 0 | 1 |
| $X_3$ | 0 | 1 | 1 |
| $Y_1$ | 0 | 0 | 1 |
| $Y_2$ | 0 | 1 | 0 |
| $Y_3$ | 1 | 0 | 0 |

Ezután hasonló a bizonyítás, mint a 10. Tétel bizonyítása. ■

# 4. Különböző erőforrások

Ebben a fejezetben a $\mathrm{PD}m\,|\,\mathrm{res}\lambda 11\,|\,C_{\max}$ és a $\mathrm{PD}m\,|\,\mathrm{res}\lambda 11,\mathrm{pmtn}^{*}\,|\,C_{\max}$ feladattal foglalkozunk, ahol a munkáknak legfeljebb 1 erőforrásra van szükségük, lásd [3]. $N = N_1 \cup \cdots \cup N_m$ és $N_i = N_i^0 \cup N_i^1 \cup \cdots \cup N_i^\lambda$, $i = 1,\ldots,m$, ahol $N_i^\mu$-ben az $M_i$ géphez tartozó $\mu$-erőforrás munkák vannak, és $N_i^\mu$-k között lehetnek üresek is, $i = 1,\ldots,m$, $\mu = 0,1,\ldots,\lambda$. Egy tetszőleges $S$ ütemezésre jelölje $C_i(S)$ a legkorábbi olyan időpontot, amikorra az összes munka elkészül az $M_i$ gépen. Ha nem okoz félreértést, $C_i(S)$ helyett egyszerűen $C_i$-t írhatunk.

Világos, hogy $C_{\max}(S) = \max\{C_i(S)\,|\,i = 1,\ldots,m\}$. Mivel $C_i(S) \geq p(N_i) = \sum_{\mu=0}^{\lambda} p(N_i^\mu)$,

$$C_{\max}(S) \geq \max\left\{\sum_{\mu=0}^{\lambda} p(N_i^\mu)\,|\,i = 1,2,\ldots,m\right\}.$$

Ezt a relációt *gép-alapú alsó korlátnak* nevezzük. Mivel két olyan munkát, melyek ugyanazt az erőforrást igénylik, egyidejűleg nem lehet feldolgozni, ezért

$$C_{\max}(S) \geq \max\left\{\sum_{i=1}^{m} p(N_i^\mu)\,|\,\mu = 1,2,\ldots,\lambda\right\}.$$

Ezt a relációt *erőforrás-alapú alsó korlátnak* nevezzük. Ezek az alsó korlátok függetlenek attól, hogy megengedünk-e megszakíthatóságot vagy sem.
Ez a modell kibővíthető egy általánosabb feladattá, ahol a munkákhoz átállítási és eltávolítási idők tartoznak. Az átállítás megelőzi a tényleges munkafolyamatot, és a feldolgozáshoz alkalmassá teszi a munkát, de nem mindig van rá szükség. Átállításra minden gépen szükség van az első munka feldolgozása előtt, és akkor, amikor két egymást követő munka közül az egyik $N_i^{\mu_1}$-beli, a másik $N_i^{\mu_2}$-beli, $\mu_1 \neq \mu_2$, $\mu_1, \mu_2 = 0,1,\ldots,\lambda$, $i = 1,\ldots,m$. Nincs szükség átállításra két $N_i^\mu$-beli munka között, $\mu = 0,1,\ldots,\lambda$, $i = 1,\ldots,m$. Vegyük figyelembe, hogy ha egy $N_i^\mu$-beli munkát feldolgozunk, akkor nem szükséges az összes maradék $N_i^\mu$-beli munkát közvetlenül utána feldolgozni. Tehát valamelyik $N_i^\mu$-beli munkákat fel lehet dolgozni egyetlen kötegként, vagy fel lehet osztani több kötegre, ahol minden köteg első munkájának szüksége van átállításra. Minden $N_i^\mu$-höz hozzárendelünk egy $s_i^\mu$ átállítási időt. Amikor az egyik gépen történik átállítás, akkor a többi gépen bármilyen tevékenység történhet.

Hasonlóan az eltávolítás a tényleges munkafolyamat után történik, amit akkor kell végrehajtani, amikor egy köteg feldolgozása befejeződik, még az ezt követő köteghez tartozó átállítás előtt. Minden $N_i^\mu$-höz hozzárendelünk egy $t_i^\mu$ eltávolítási időt. Amikor az egyik gépen történik eltávolítás, akkor a többi gépen bármilyen tevékenység történhet. Erre a modellre a teljes átfutási idő az utolsó eltávolítás befejezési ideje. Ebben a fejezetben is gyakran fogjuk használni a csoportosító eljárás szemléletet. Itt $\forall i$-re ($1 \le i \le m$) és $\forall \mu$-re ($0 \le \mu \le \lambda$) $N_i^\mu$-ből $H_i^\mu$ köteget készítünk, ha nem üres. A kötegeket összetett munkáknak tekintjük, a kötegeken belül a munkákat tetszőleges sorrendben dolgozzuk fel, szünet nélkül. Vegyük észre, hogy egy $H_i^0$ ($1 \le i \le m$) köteget bármikor fel lehet dolgozni, de egy $H_i^\mu$ köteget nem lehet a $H_k^\mu$ kötegekkel egyszerre feldolgozni, ha $k = 1,2,\ldots,m$, $k \ne i$, $1 \le \mu \le \lambda$.

Több alkalommal egy megfelelő *open shop* feladatból nyerünk algoritmikus ötleteket. Az open shop modellben minden munka elvégzése *műveletekre* bomlik, melyek elvégzése egy-egy meghatározott gépen történik. A műveletek elvégzésének sorrendje minden munka esetén tetszőleges. Tehát minden munkához gépek tartoznak, ahol a műveleteket fel kell dolgozni. A gépek sorrendje nem adott, és a feladat megoldása során választani kell egyet, különböző munkákhoz, különböző sorrend tartozhat. Egy munka egyszerre most is legfeljebb egy gépen lehet, azaz ugyanannak a munkának két művelete egyszerre nem futhat két gépen. Az open shop feladat szabványos jelölése, ahol a gépek száma $m$, és a cél a teljes átfutási idő minimalizálása: $\mathrm{O}m \,||\, C_{\max}$ (ha a munkák nem szakíthatóak meg) és $\mathrm{O}m \mid \mathrm{pmtn} \mid C_{\max}$ (ha a munkák megszakíthatóak). Ha a gépek száma változó, akkor a feladatok jelölése: $\mathrm{O} \,||\, C_{\max}$ és $\mathrm{O} \mid \mathrm{pmtn} \mid C_{\max}$. Gonzalez és Sahni megoldotta az $\mathrm{O}2 \,||\, C_{\max}$ feladatot $\mathrm{O}(n)$ idő alatt (lásd [4,7]), míg az $\mathrm{O}3 \,||\, C_{\max}$ gyengén NP-nehéz. Az $\mathrm{O} \,||\, C_{\max}$ erősen NP-nehéz; tulajdonképpen NP-nehéz eldönteni, hogy létezik-e olyan ütemezés, melynek teljes átfutási ideje legfeljebb 4. Másrészt az $\mathrm{O} \mid \mathrm{pmtn} \mid C_{\max}$ feladat megoldható polinomiális időben.

A 4.1. alfejezetben lineáris idejű algoritmust adunk a $\mathrm{PD}2 \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatra és annak általánosabb változatára is átállítási és eltávolítási időkkel. A 4.2. alfejezetben a $\mathrm{PD}m \mid \mathrm{res}\lambda 11, \mathrm{pmtn}^* \mid C_{\max}$ feladat komplexitását elemezzük. A 4.3. alfejezetben mutatunk egy egyszerű mohó közelítő algoritmust a $\mathrm{PD} \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatra, melyre

a közelítési hányados legfeljebb 2. A 4.4. alfejezetben mutatunk egy polinomiális idejű közelítő sémát (PTAS) a $\mathrm{PD}m \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatra.

## 4.1. Két gép: nem-megszakítható, erőforrást nem igénylő munkák

Ebben az alfejezetben megmutatjuk, hogy a $\mathrm{PD}2 \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladat megoldható lineáris idő alatt, még akkor is, ha a munkákhoz átállítási és eltávolítási idők tartoznak, lásd [3].

Először nézzük a feladatot átállítási és eltávolítási idők nélkül. Vegyük észre, hogy a $\lambda = 1$ esetet már megvizsgáltuk. Alkalmazzuk a csoportosító eljárás szemléletet, és vezessük vissza a feladatot a 2 gépes open shop feladatra ($\mathrm{O}2 \mid\mid C_{\max}$). Készítsünk legfeljebb $\lambda + 2$ összetett munkát: $K_0^A, K_0^B, K_1, \ldots, K_\lambda$. A $K_1, \ldots, K_\lambda$ munkák mindegyike két tevékenységből áll: valamilyen sorrendben végrehajtja az $M_1$ és az $M_2$ gép. A $K_\mu$ munkák feldolgozási idői az $M_1$ gépen $a(K_\mu) := p(N_1^\mu)$, és az $M_2$ gépen $b(K_\mu) := p(N_2^\mu)$ ($\mu = 1, \ldots, \lambda$).

A $K_0^A$-t csak az $M_1$ dolgozza fel, és $a(K_0^A) := p(N_1^0)$, $b(K_0^A) := 0$. Hasonlóan a $K_0^B$-t csak az $M_2$ dolgozza fel, és $a(K_0^B) := 0$, $b(K_0^B) := p(N_2^0)$.

Ezeket $\mathrm{O}(n \log n)$ idő alatt találhatjuk meg: rendezzük sorba mindkét gépen a munkákat az igényelt erőforrás sorszáma szerinti növekvő sorrendben, kezdve az erőforrást nem igénylő munkákkal, folytatva az 1-es számú erőforrást igénylő munkákkal stb.

$\forall \mu \geq 1$-re a $K_\mu$ munka két tevékenysége nem végezhető el egyszerre. Másrészt $K_0^A$ vagy $K_0^B$ bármi mással egy időben feldolgozható: a megfelelő gépen feldolgozható akkor is, ha a másik gép feldolgoz valamit, bármi legyen is az. Így ez olyan, mint egy kétgépes open shop. Egy optimális $S_0^*$ open shop ütemezés kapható pl. a közismert Gonzalez-Sahni algoritmussal, ami a munkák számát figyelembe véve lineáris idejű, lásd [4,7].

Vegyük észre, hogy

$$C_{\max}(S_0^*) = \max\left\{ a(K_0^A) + \sum_{\mu=1}^{\lambda} a(K_\mu), b(K_0^B) + \sum_{\mu=1}^{\lambda} b(K_\mu), \max\left\{ a(K_\mu) + b(K_\mu) \mid 1 \leq \mu \leq \lambda \right\} \right\},$$

ami megfelel a teljes átfutási idő optimális értékének az eredeti feladatban. $S_0^*$-ból az

eredeti feladatra megoldás: $K_\mu$ helyett $N_1^\mu$ vagy $N_2^\mu$, melyekben az eredeti munkák tetszőleges sorrendben jöhetnek.

Most nézzük a feladatot átállítási és eltávolítási időkkel: $s_i^\mu$, $t_i^\mu$.

$K_0^A$, $K_0^B$ és $K_\mu$ $(1 \le \mu \le \lambda)$, $a(K_\mu)$, $b(K_\mu)$ stb., mint fent. $K_\mu$-höz átállítási és eltávolítási idők: $s_i(K_\mu) := s_i^\mu$, $t_i(K_\mu) := t_i^\mu$, $i \in \{1,2\}$, $\mu = 1,\ldots,\lambda$.

$K_0^A$-hoz és $K_0^B$-hez:

$s_1(K_0^A) := s_1^0$, $t_1(K_0^A) := t_1^0$, $s_2(K_0^A) := t_2(K_0^A) := 0$,

$s_2(K_0^B) := s_2^0$, $t_2(K_0^B) := t_2^0$, $s_1(K_0^B) := t_1(K_0^B) := 0$.

Az így kapott open shop-ra létezik lineáris idejű algoritmus, lásd [5]. Az kapjuk, hogy

$$C_{\max}(S_0^*) = \max\Bigg\{ s_1(K_0^A) + a(K_0^A) + t_1(K_0^A) + \sum_{\mu=1}^{\lambda} (s_1(K_\mu) + a(K_\mu) + t_1(K_\mu)),$$

$$s_2(K_0^B) + b(K_0^B) + t_2(K_0^B) + \sum_{\mu=1}^{\lambda} (s_2(K_\mu) + b(K_\mu) + t_2(K_\mu)),$$

$$\max\{\min\{s_1(K_\mu) + t_2(K_\mu), s_2(K_\mu) + t_1(K_\mu)\} + a(K_\mu) + b(K_\mu) \mid 1 \le \mu \le \lambda\} \Bigg\},$$

ami megfelel a teljes átfutási idő optimális értékének az eredeti feladatban. Vegyük észre, hogy $\min\{s_1(K_\mu) + t_2(K_\mu), s_2(K_\mu) + t_1(K_\mu)\} + a(K_\mu) + b(K_\mu)$ a lehető legkisebb idő, ami alatt $K_\mu$ munka feldolgozható.

Már korábban megállapítottuk, hogy az összetett munkákat $O(n \log n)$ idő alatt alakíthatjuk ki. Azoknak az összetett munkáknak, melyeknek legalább az egyik tevékenysége nem-nulla, az összes száma legfeljebb $n$. Eredményünket a következő tétel foglalja össze.

**12. Tétel** (lásd [3]): A $PD2 \mid res\lambda 11 \mid C_{\max}$ feladat megoldható $O(n \log n)$ idő alatt, még akkor is, ha vannak átállítási és eltávolítási idők.

Ez a polinomiális idejű algoritmus nem általánosítható több mint két gép esetére. A 3. és 4. Tétel szerint, *ha az erőforrást nem igénylő munkák nem szakíthatóak meg* és nincsenek átállítási és eltávolítási idők, a $PD3 \mid res111 \mid C_{\max}$ gyengén NP-nehéz, és a $PD \mid res111 \mid C_{\max}$ erősen NP-nehéz. Ezért a következő alfejezetben az erőforrást nem igénylő munkák *megszakíthatóak lesznek*, lásd [3].

## 4.2. Megszakítható, erőforrást nem igénylő munkák

Először a $\mathrm{PD}m \mid \mathrm{res}211, \mathrm{pmtn}^* \mid C_{\max}$ feladatot nézzük átállítási és eltávolítási idők nélkül, ahol az erőforrást igénylő munkáknak a 2 erőforrásból 1-re van szükségük. Ezt visszavezetjük 2 gépes open shop feladatra ($\mathrm{O}2 \mid\mid C_{\max}$). Az erőforrást nem igénylő munkákat ideiglenesen hagyjuk figyelmen kívül. Az open shop feladat legyen a következő: a 2 gép legyen $A$ és $B$, és legyen $m$ munka: $K_1, \ldots, K_m$. Minden $K_i$ munka két tevékenységből áll: valamilyen sorrendben végre hajtja az $A$ és a $B$ megszakítás nélkül. A $K_i$ munkára a tevékenységeinek megmunkálási idői:

$a(K_i) := p(N_i^1),\ b(K_i) := p(N_i^2),\ i = 1, \ldots, m.$

Egy munka tevékenységei nem végezhetők el egyszerre, és legfeljebb egy munka végezhető el egy időben egy gépen. Ezért a $K_i$ két tevékenysége megfelel $N_i^1$-nek és $N_i^2$-nek, és az összes megmunkálási idő $\mathrm{O}(n)$ idő alatt kiszámolható. Egy optimális $S_0^*$ ütemezés erre az open shop feladatra megtalálható a Gonzalez-Sahni algoritmussal, ami $\mathrm{O}(m)$ ideig tart. Az $S_0^*$ ütemezés átalakítható egy optimális $S_1$ ütemezéssé a $\mathrm{PD}m \mid \mathrm{res}211 \mid C_{\max}$ feladatra erőforrást nem igénylő munkák nélkül. Ha az $S_0^*$ ütemezésben a $K_i$ munkát ($1 \le i \le m$) $A$ munkálja meg $[\tau^1, \tau^2]$-ben, akkor az $S_1$ ütemezésben az $N_i^1$-beli munkákat $M_i$ munkálja meg ugyanakkor egy blokként (a blokkban tetszőleges sorrendben). Hasonlóan ha az $S_0^*$-ban a $K_i$ munkát ($1 \le i \le m$) $B$ munkálja meg $[\tau^3, \tau^4]$-ben, akkor az $S_1$ ütemezésben az $N_i^2$-beli munkákat $M_i$ munkálja meg ugyanakkor egy blokként (a blokkban tetszőleges sorrendben). Az eredményül kapott ütemezés nyilvánvalóan megengedett, és a teljes átfutási ideje megegyezik az $S_0^*$ ütemezés teljes átfutási idejével. Ebből következik, hogy

$$C_{\max}(S_0^*) = \max\left\{\sum_{i=1}^{m} a(K_i), \sum_{i=1}^{m} b(K_i), \max\{a(K_i) + b(K_i) \mid 1 \le i \le m\}\right\},$$

ami megfelel az optimumnak az eredeti feladatban, az erőforrást nem igénylő munkák nélkül. Ezután az erőforrást nem igénylő munkákat hozzá vesszük megszakítható módon, tetszőleges sorrendben a megfelelő gépeken, ha lehet a $[0, C_{\max}(S_0^*)]$-ban található hézagokba, de ha az összes hézagot kitöltöttük és marad még hátra munka (vagy egy része), akkor $C_{\max}(S_0^*)$-tól jöhet a maradék szünet nélkül.

Ezzel bebizonyítottuk a következő tételt.

**13. Tétel** (lásd [3]): A $\mathrm{PD}m\,|\,\mathrm{res}211,\mathrm{pmtn}^{*}\,|\,C_{\max}$ feladat átállítási és eltávolítási idők nélkül $\mathrm{O}(m+n)$ idő alatt megoldható.

Ez a polinomiális idejű algoritmus nem általánosítható több mint két erőforrás esetére.

A $\mathrm{PD}3\,|\,\mathrm{res}311,\mathrm{pmtn}^{*}\,|\,C_{\max}$ feladat átállítási és már eltávolítási idők nélkül is NP-nehéz:

**14. Tétel** (lásd [3]): A $\mathrm{PD}3\,|\,\mathrm{res}311\,|\,C_{\max}$ feladat átállítási és eltávolítási idők nélkül, ahol minden munkának szüksége van egy erőforrásra gyengén NP-nehéz.

**Biz.:** Megtalálható [3]-ban, ahol a Partíciót használják a visszavezetésre. ■

**15. Tétel** (lásd [3]): A $\mathrm{PD}3\,|\,\mathrm{res}211\,|\,C_{\max}$ feladat átállítási időkkel, ahol minden munkának szüksége van egy erőforrásra gyengén NP-nehéz.

**Biz.:** Megtalálható [3]-ban, ahol a Partíciót használják a visszavezetésre. ■

## *4.3. Mohó közelítés*

Ebben az alfejezetben a $\mathrm{PD}\,|\,\mathrm{res}\lambda 11\,|\,C_{\max}$ feladatot vizsgáljuk, ahol a gépek száma ($m$) része az inputnak, lásd [3]. Mutatunk egy egyszerű mohó algoritmust, ami létrehoz egy nem-megszakítható ütemezést, melynek teljes átfutási ideje legfeljebb kétszer nagyobb, mint az optimális érték.

A $\mathrm{PD}\,|\,\mathrm{res}\lambda 11\,|\,C_{\max}$ feladatra, egy nem-megszakítható $S$ ütemezés *sűrű* (dense), ha csak akkor van szünet, ha nem áll a feldolgozás elkezdéséhez készen munka. Egy ilyen ütemezést kaphatunk a következő mohó algoritmussal.

**G algoritmus** (lásd [3])

*Input:* $\mathrm{PD}\,|\,\mathrm{res}\lambda 11\,|\,C_{\max}$ feladat

*Output:* Egy heurisztikus $S$ ütemezés

1. Bármikor, amikor valamelyik $M_i$ gép elérhető, vizsgáljuk meg az összes munkát, amit még hozzá kell rendelni $M_i$-hez tetszőleges sorrendben, és keressük meg azt, amelyik legkorábban kezdhető az $M_i$-n: $J_k$. Rendeljük hozzá $J_k$-t $M_i$-hez.
2. Addig ismételjük az 1. lépést, amíg az összes munkát nem ütemeztük. $S$: az eredményül kapott ütemezés. Stop.

Világos hogy a G algoritmus futási ideje legfeljebb $\mathrm{O}(nm\min\{n,m\})$.

**16. Tétel** (lásd [3]): A $PD|res\lambda 11|C_{\max}$ feladatra legyen $S^*$ egy optimális ütemezés, és $S$ egy sűrű ütemezés, amit a G algoritmussal kaptunk. Ekkor

$\frac{C_{\max}(S)}{C_{\max}(S^*)} < 2$.

**Biz.:** Az általánosság megszorítása nélkül tegyük fel, hogy $M_m$-nél fejeződik be az $S$.

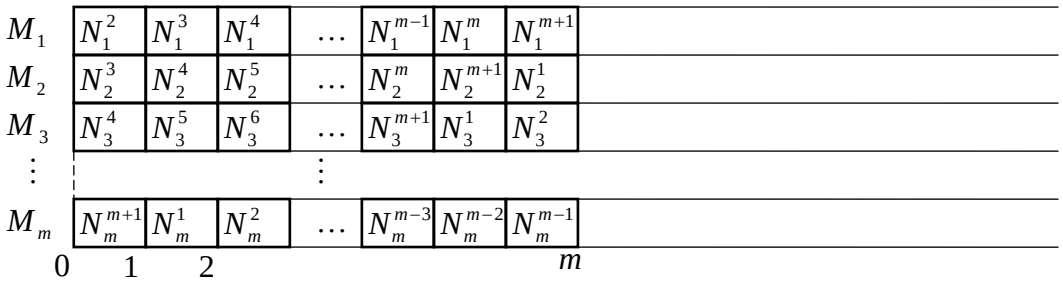
$Q$: az utolsó szünet utáni munkák halmaza az $M_m$-en. Tegyük fel, hogy $Q$ nem üres, mert különben $S$ optimális ütemezés, és a tétel nyilvánvalóan igaz.

$Q$-ban minden munka igényel erőforrást, különben egy erőforrást nem igénylő munka hamarabb is kezdődhetett volna. Legyen $J_k \in Q$, és $J_k$-nak a $\nu$ erőforrás kell $(1 \le \nu \le \lambda)$. Ebből következik, hogy $M_m$-en az összes szünet nem haladhatja meg a többi gépen $\nu$ erőforrást igénylő munkák összmegmunkálási idejét. Ezért

$C_{\max}(S) \le \sum_{\mu=0}^{\lambda} p(N_m^{\mu}) + \sum_{i=1}^{m-1} p(N_i^{\nu}) < 2C_{\max}(S^*)$, az alsó korlátok miatt. ■

Nyitott kérdés, hogy a közelítési hányados éles-e. Azonban $C_{\max}(S)/C_{\max}(S^*)$ lehet egyenlő $2-1/m$-mel. Elegendő a következő esetét venni a $PDm|res\lambda 11|C_{\max}$ feladatnak. Minden gépen $m$ egységméretű munka van, az erőforrások száma $\lambda = m+1$. Mindegyik $N_i^{\mu}$ üres, ha $\mu = i$, és pontosan 1 munkát tartalmaz, ha $\mu \ne i$ $(1 \le i \le m,\ 1 \le \mu \le m+1)$.

$C_{\max}(S^*) = m$, mert lehet úgy ütemezni a munkákat, hogy ne legyen hézag $[0,m]$-ben:




Ugyanakkor a G algoritmus eredménye a legrosszabb esetben pl. a következő.

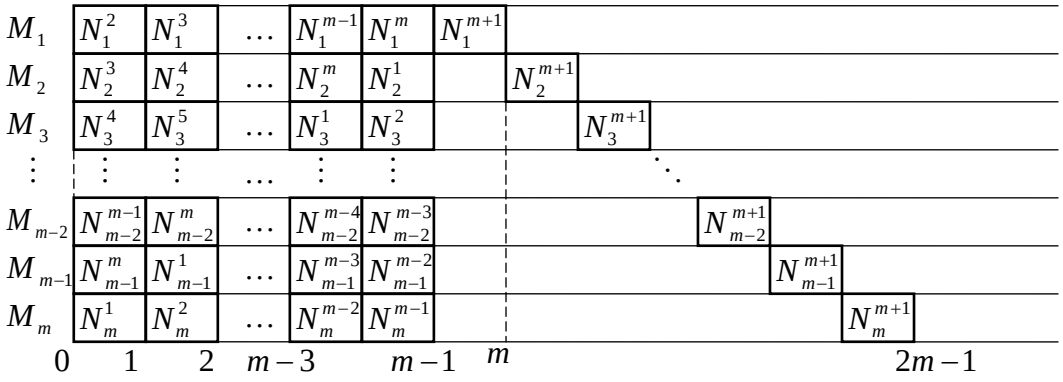
A $[0,1]$-ben az $N_1^2$-beli, az $N_2^3$-beli,…, az $N_{m-1}^m$-beli és az $N_m^1$-beli, majd az $[1,2]$-ben az $N_1^3$-beli, az $N_2^4$-beli,…, az $N_{m-2}^m$-beli, az $N_{m-1}^1$-beli és az $N_m^2$-beli munkákat dolgozzuk fel. És így tovább.

Az $[m-3,m-2]$-ben az $N_1^{m-1}$-beli, az $N_2^m$-beli, az $N_3^1$-beli,… és az $N_m^{m-2}$-beli, majd

az $[m-2, m-1]$-ben az $N_1^m$-beli, az $N_2^1$-beli, az $N_3^2$-beli,… és az $N_m^{m-1}$-beli munkákat dolgozzuk fel.

Ezután az $[m-1, m]$-ben az $N_1^{m+1}$-beli, az $[m, m+1]$-ben az $N_2^{m+1}$-beli, az $[m+1, m+2]$-ben az $N_3^{m+1}$-beli,… és végül az $[2m-2, 2m-1]$-ben az $N_m^{m+1}$-beli munkákat dolgozzuk fel.

Az így kapott $S$-re $C_{\max}(S) = 2m-1$.




## 4.4. PTAS rögzített számú gépek esetén

Ebben az alfejezetben mutatunk egy polinomiális idejű közelítő sémát (PTAS) a $\mathrm{PD}m \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatra, ahol a gépek $m$ száma rögzített, de az erőforrások $\lambda$ száma az input része. Az $\varepsilon$ pozitív és rögzített, lásd [3].

Ha a gépek $m$ száma része az inputnak, akkor PTAS nem lehetséges $\mathrm{O} \mid\mid C_{\max}$-ra, így $\mathrm{PD} \mid \mathrm{res}\lambda 11 \mid C_{\max}$-ra sem, kivéve ha $P = NP$.

Ez az alfejezet a 2.3. alfejezet általánosítása, ahol a $\mathrm{PD}m \mid \mathrm{res}111 \mid C_{\max}$ feladatra mutattunk PTAS-t.

$C := \max\left\{\max\left\{\sum_{\mu=0}^{\lambda} p\left(N_i^\mu\right) \mid i = 1,2,\ldots,m\right\}, \max\left\{\sum_{i=1}^{m} p\left(N_i^\mu\right) \mid \mu = 1,2,\ldots,\lambda\right\}\right\}$, így $C_{\max}\left(S^*\right) \geq C$. A 16. Tétel bizonyításából következik, hogy

$C_{\max}(S) \leq \sum_{\mu=0}^{\lambda} p\left(N_m^\mu\right) + \sum_{i=1}^{m-1} p\left(N_i^\nu\right) < 2C$, ahol $S$ egy sűrű ütemezés, amit a G algoritmussal kaptunk, $\nu$-t pedig a bizonyítás során definiáltuk. Tehát $C \leq C_{\max}\left(S^*\right) < 2C$.

$\tilde{\varepsilon} := \varepsilon/111$. Hasonlóan a 2.3. alfejezethez, osszuk fel $N$ -et $A_1$ -re, $A_2$ -re és $A_3$ -ra:

$A_1 := \{J_j \mid p(J_j) > \delta C\}$ („nagy munkák”),

$A_2 := \{J_j \mid \delta C \geq p(J_j) > \delta^2 C\}$ („közepes munkák”),

$A_3 := \{J_j \mid \delta^2 C \geq p(J_j)\}$ („kis munkák”),

ahol $\delta$ -t úgy választjuk meg, hogy teljesüljenek a következők:

$p(A_2) \leq \tilde{\varepsilon} C$ és $\frac{\tilde{\varepsilon}}{m^3} \geq \delta \geq \left(\frac{\tilde{\varepsilon}}{m^3}\right)^{2^{\lceil m/(\tilde{\varepsilon}) \rceil}}$ .

$\mathscr{S}_{\mathscr{R}}$ : azon ütemezések halmaza, ahol a „nagy munkák” kezdési ideje a $\delta^2 C$ nem-negatív egész többszöröse.

**2. Lemma** (lásd [3]): $S_R^*$ egy ütemezés a $\mathrm{PD}m \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatra, ami optimális az $\mathscr{S}_{\mathscr{R}}$ -beli ütemezések között. Ekkor

$C_{\max}(S_R^*) \leq C_{\max}(S^*) + 2m\delta C < \overline{C}$ ,

ahol $\overline{C} := 2C(1 + m\delta)$.

**Biz.:** Egy $S^*$ optimális ütemezésre legyen $\tau_1 < \tau_2 < \ldots < \tau_u$ a „nagy munkák” különböző kezdési időinek növekvő sorozata. Legyen egy gépen maximum $b$ „nagy munka”. Mivel $C_{\max}(S^*) < 2C$, ezért $b\delta C \leq C_{\max}(S^*) < 2C$, tehát $b < 2/\delta$. Mivel $u \leq bm$, ezért

$u < 2\frac{m}{\delta}$.

Alakítsuk $S^*$-ot egy $\mathscr{S}_{\mathscr{R}}$ -beli $S_R$ ütemezéssé úgy, hogy $u$ iterációval a munkákat jobbra toljuk a következő módon. $S_0 := S^*$. Az $i$ -edik iterációban ($1 \leq i \leq u$) kiindulunk $S_{i-1}$ ütemezésből. $\sigma_i$ a legkisebb többszöröse $\delta^2 C$ -nek, ami nagyobb vagy egyenlő, mint $\tau_i$. Azoknak a munkáknak a kezdési idejét megnöveljük $\sigma_i - \tau_i$ -vel, melyek $S_{i-1}$ -beli kezdési ideje nem korábbi, mint $\tau_i$. Kapjuk az $S_i$ ütemezést, ahol a „nagy munkák” kezdési ideje

$$\tau_j := \begin{cases} \tau_j & \text{, ha} \quad 1 \leq j \leq i, \\ \tau_j + \sigma_i - \tau_i & \text{, ha} \quad i \leq j \leq u. \end{cases}$$

Nyilván $\tau_1, \ldots, \tau_i$ többszöröse $\delta^2 C$ -nek és $S_i$ továbbra is megengedett. Végül megkapjuk $S_u$ -t, és $S_R := S_u$. Minden iterációnál a munkák befejezési ideje, és így a

teljes átfutási idő legfeljebb $\delta^2 C$ -vel nő. Így a teljes növekedés legfeljebb $u\delta^2 C < 2\frac{m}{\delta}\delta^2 C = 2m\delta C$ , ezzel a bizonyítás kész. ■

Hasonlóan a 2.3. alfejezethez legyen $\Delta$ a $\left[0, \overline{C} - \delta C\right[$ intervallumban található, a $\delta^2 C$ -nek nem-negatív egész többszöröseinek halmaza.

Legyen $\mathscr{S}_{\mathscr{B}}$ azon ütemezések osztálya, amely tartalmazza az összes lehetséges ütemezését a „nagy munkáknak", ahol a kezdési idők $\Delta$-beliek. A 2. Lemmából és a „nagy munkák" definíciójából következik, hogy van ilyen ütemezés.

Vegyünk egy tetszőleges $\mathscr{S}_{\mathscr{B}}$-beli $S_B$ ütemezést.

Legyen $\tau_1 < \tau_2 < \cdots < \tau_q$ a „nagy munkák" különböző kezdési *és* befejezési időinek növekvő sorozata.

$q < 4\frac{m}{\delta} + 4m^2$ ,

ugyanis: $C_{\max}\left(S^*\right) \le C_{\max}\left(S^*\right) + 2m\delta C < \overline{C}$ , ezért $C_{\max}\left(S^*\right) < 2C\left(1 + m\delta\right)$.

Egy gépen maximum $b$ „nagy munka" van, így $b\delta C \le C_{\max}\left(S^*\right) < 2C\left(1 + m\delta\right)$.

Ebből következik, hogy $b < \frac{2}{\delta}\left(1 + m\delta\right)$, és mivel $q \le 2bm$ , ezért

$q < \frac{4}{\delta}\left(1 + m\delta\right)m = \frac{4m}{\delta} + 4m^2$ .

$S_B$-re $I_0 := [0, \tau_1], I_1 := [\tau_1, \tau_2], \ldots, I_k := [\tau_k, \tau_{k+1}], \ldots, I_{q-1} := \left[\tau_{q-1}, \tau_q\right], I_q := \left[\tau_q, \infty\right[$.

Ebből következik, hogy $I_k$ ( $0 \le k \le q$ ) belsejében a „nagy munkák" közül egyik sem kezdődik, vagy fejeződik be. Legyen az $I_k$ intervallum hossza $i_k$ , $k = 0, \ldots, q$ . $\forall k$ -ra $\tilde{M}_k$ az összes olyan gép halmaza, ahol szünet van $I_k$ -ban, és $\tilde{R}_k$ azoknak az erőforrásoknak a halmaza, melyekre nincs szükség $I_k$ -ban. A „kis munkákat" vesszük most hozzá $S_B$ -hez *megszakíthatóan* az $LP\left(S_B\right)$ lineáris program segítségével:

$x_{i\mu k}$ : az összmegmunkálási ideje azoknak a „kicsi" $\mu$ -erőforrás munkáknak, melyeket az $M_i$ gépen az $I_k$ intervallumon munkáljuk meg.

Az ilyen munkákat csak akkor tudjuk $I_k$ -n megmunkálni az $M_i$ gépen, ha $M_i \in \tilde{M}_k$ , és $\mu \in \tilde{R}_k$ .

(10) min *z*

(11) $z \geq \sum_{\mu=0}^{\lambda} x_{i\mu q}, i = 1, \ldots, m,$

(12) $z \geq \sum_{i=1}^{m} x_{i\mu q}, \mu = 1, \ldots, \lambda,$

(13) $i_k \geq \sum_{\mu=0}^{\lambda} x_{i\mu k}, i = 1, \ldots, m, k = 0, \ldots, q-1,$

(14) $i_k \geq \sum_{i=1}^{m} x_{i\mu k}, \mu = 1, \ldots, \lambda, k = 0, \ldots, q-1,$

(15) $\sum_{\substack{k=0 \\ M_i \in \tilde{M}_k \\ \mu \in \tilde{R}_k}}^{q} x_{i\mu k} = p\left(N_i^{\mu} \cap A_3\right), i = 1, \ldots, m, \mu = 0, \ldots, \lambda,$

(16) $x_{i\mu k} \geq 0, i = 1, \ldots, m, \mu = 0, \ldots, \lambda, k = 0, \ldots, q.$

(13): A gép-alapú alsó korlát a „kis munkák” összmegmunkálási idejére az $M_i$ gépen az $I_k$ intervallumon nem lehet több, mint az intervallum hossza, $i_k$.

(14): Az erőforrás-alapú alsó korlát a „kicsi” $\mu$-erőforrás munkák összmegmunkálási idejére az $I_k$ intervallumon nem lehet több, mint $i_k$.

(10)-(12): A célfüggvény optimális értéke ($z$) egyenlő az $I_q$-hoz rendelt „kis munkákra” a gép-alapú alsó korlát és az erőforrás-alapú alsó korlát maximuma.

(15): Minden „kis munkát” hozzárendelünk a megfelelő géphez az $I_k$ intervallumokban.

**3. Lemma** (lásd [3]): Legyen $S_B$ egy $\mathscr{S}_{\mathscr{B}}$-beli ütemezés, melynek teljes átfutási ideje $t_q$. Tegyük fel, hogy az $LP(S_B)$ lineáris program optimális megoldása: $(x_{i\mu k})$ és $z$. Ekkor $S_B$-t ki lehet egészíteni $S_1$ ütemezéssé ($S_B$-hez hozzávesszük a „kis munkákat”), hogy:

i. a „nagy munkák” kezdési ideje még mindig a $\delta^2 C$ többszöröse;

ii. azoknak a „kicsi” $\mu$-erőforrás munkáknak, melyeket az $M_i$ gépen az $I_k$-n munkáljuk meg, az összmegmunkálási ideje $x_{i\mu k}$ ($0 \leq \mu \leq \lambda$);

iii. a „kis munkák” megszakíthatóak és a megszakítások száma nem több, mint $4m^2(q+1)$;

iv. $C_{\max}(S_1) = t_q + z$.

**Biz.:** Leírunk egy eljárást $S_B$ kiegészítésére. Minden $I_k$ -ra ($k = 0,\ldots,q-1$) és $\left[t_q, t_q + z\right]$-re tegyük a következőket. Formáljunk összetett munkákat: $K_0, K_1, \ldots, K_\lambda$, ahol $K_\mu$ megmunkálási ideje az $M_i$ gépen az $I_k$ intervallumon $x_{i\mu k}$. Hagyjuk figyelmen kívül, hogy a gépek foglaltak a „nagy munkák” feldolgozásával, és találjunk egy optimális megszakítható open shop ütemezést a $K_1, \ldots, K_\lambda$ összetett munkákhoz. Egy lehetséges algoritmus ilyen ütemezés megtalálásához minden intervallumra megtalálható [6]-ban. Vegyük észre, hogy az algoritmus implementálható olyan módon, hogy minden intervallumban maximum $4m^2$ legyen a megszakítások száma. Ha szükséges a megmaradt üres intervallumokban dolgozzuk fel $K_0$-t, amit különböző gépeken egyszerre lehet feldolgozni. ■

$\boldsymbol{A = A(\varepsilon, m)}$ **algoritmus** (lásd [3])

*Input:* Egy példánya a $\mathrm{PD}m \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatnak, $\varepsilon > 0$

*Output:* Egy $S_\varepsilon$ heurisztikus ütemezés

1. Határozzuk meg $\varepsilon$-ból $\tilde{\varepsilon}$-ot és $\delta$-t. Osszuk fel $N$-et $A_1$-re, $A_2$-re és $A_3$-ra („nagy munkákra, „közepes munkákra” és „kis munkákra”).
2. Minden $\mathscr{S}_{\mathscr{B}}$-beli $S_B$ ütemezésre oldjuk meg a megfelelő $LP(S_B)$ lineáris programot, és keressük meg a célfüggvény optimális értékét: $z^*$. Legyen $S_B^*$ olyan $\mathscr{S}_{\mathscr{B}}$-beli ütemezés, melyre $z^* + \tau_q$ minimális.
3. A 3. Lemma bizonyításában leírt technikával egészítsük ki $S_B^*$-ot a „kis munkákkal” (megszakíthatóan) $S_1$ ütemezéssé. $I_{q+1} := \left[\tau_q, \tau_q + z^*\right]$.
4. Vizsgáljuk át $S_1$ ütemezést elejétől a végéig. $\forall \mu = 0,\ldots,\lambda$ -ra és $\forall i = 1,\ldots,m$ -re legyenek $T_{i\mu}^l$-ek ($l = 1,2,\ldots$) időintervallumok , ahol „kicsi” $\mu$-erőforrás munkákat dolgozunk fel (megszakíthatóan) az $M_i$ gépen az $S_1$-ben. A $T_{i\mu}^l$-ek hossza legyen $d_{i\mu}^l$. Ideiglenesen távolítsuk el azokat a munkákat, melyeket valamelyik $T_{i\mu}^l$-ben dolgozunk fel. Azért hogy $S_1$-et nem-megszakítható ütemezéssé alakítsuk, $\forall T_{i\mu}^l$ intervallumhoz rendeljük hozzá azokat a „kis munkákat” $N_i^\mu$-ből, melyek összmegmunkálási ideje legfeljebb $d_{i\mu}^l$. A munkákat tetszőleges

sorrendben, nem-megszakíthatóan rendeljük a gépekhez. Ha egy munka nem fér az elérhető intervallumba, ideiglenesen tegyük félre.

5. $\forall i$-re $i=1$-től $i=m$-ig $\rho_i$ köteg készítése a megmaradt $N_i$-beli munkákból. Kezdjük a $\rho_1$-es köteget az $M_1$ gépen a $\tau_q + z^*$ időponttól. Ezután $\forall i = 1,\ldots,m-1$-re kezdjük a $\rho_{i+1}$-es köteget az $M_{i+1}$ gépen közvetlenül azután, hogy a $\rho_i$ köteg befejeződött az $M_i$ gépen. Ha $\rho_i$ üres, akkor hagyjuk figyelmen kívül $M_i$-t. Legyen $S_\varepsilon$ az eredményül kapott ütemezés. Stop.

**17. Tétel** (lásd [3]): Az $A = A(\varepsilon, m)$ algoritmus PTAS a $\mathrm{PD}m \mid \mathrm{res}\lambda 11 \mid C_{\max}$ feladatra.

**Biz.:** Hasonló a 7. Tétel bizonyításához, és megtalálható [3]-ban is. ■

# 5. Átlagos átfutási idő: legegyszerűbb problémák

Ebben a fejezetben a cél az *átlagos átfutási idő* minimalizálása, ami ekvivalens a befejezési idők összegének minimalizálásával.

Ha nincs erőforrás, akkor az egygépes esetben ($1 \| \sum_j C_j$) az optimális megoldás: a munkákat a megmunkálási idők szerint nem-csökkenő sorrendbe rendezzük (*SPT sorrend*), majd a 0 időponttól kezdve szünet nélkül rakjuk a gépre a munkákat ebben a sorrendben, lásd [1]. Hasonló a helyzet, ha több párhuzamos és dedikált gép van, és nincs erőforrás ($\mathrm{PD}m \| \sum_j C_j$ vagy $\mathrm{PD} \| \sum_j C_j$): minden gépen a hozzátartozó munkákat SPT sorrendbe rendezzük, majd a 0 időponttól kezdve szünet nélkül rakjuk a gépre a munkákat ebben a sorrendben. Az így kapott ütemezést jelöljük $S_{SPT}$-vel.

Világos, hogy a $\mathrm{PD}m \mid \mathrm{res111} \mid \sum_j C_j$ vagy a $\mathrm{PD} \mid \mathrm{res111} \mid \sum_j C_j$ probléma esetén

$$\sum_j C_j(S^*) \geq \sum_j C_j(S_{SPT}),$$

tehát ha $S_{SPT}$ megengedett, akkor optimális. ($S_{SPT}$ biztosan megengedett, ha nincs erőforrást igénylő munka, vagy csak 1 gépen vannak erőforrást igénylő munkák.)

Ha nincsenek erőforrást nem igénylő munkák, akkor az optimális megoldás: a munkákat SPT sorrendbe rendezzük, majd ebben a sorrendben rakjuk a munkákat a 0 időponttól kezdve a megfelelő gépre közvetlenül azután, hogy az előző munka feldolgozása befejeződött.

Nézzük most a $\mathrm{PD2} \mid \mathrm{res111} \mid \sum_j C_j$ problémát, ahol mindkét gépen 1-1 darab erőforrást igénylő munka van.

**A2 algoritmus**

*Input:* Egy példánya a $\mathrm{PD2} \mid \mathrm{res111} \mid \sum_j C_j$ problémának, ahol mindkét gépen 1-1 darab erőforrást igénylő munka van

*Output:* Egy $S_{A2}$ heurisztikus ütemezés

1. $S_{SPT}$ ütemezés meghatározása.
2. Ha így az erőforrást igénylő munkák között van átfedés, akkor azt az erőforrást igénylő munkát, amelyik később fejeződik be, és az utána következő erőforrást nem igénylő munkákat jobbra toljuk el úgy, hogy az erőforrást igénylő munka kezdési ideje a másik erőforrást igénylő munka

befejezési ideje legyen. Ha az erőforrást igénylő munkák ugyanakkor fejeződnek be, akkor mindegy, hogy melyiket választjuk.

3. $S_{A2}$: az eredményül kapott ütemezés. Stop.

**18. Tétel:** Az A2 algoritmus 2-közelítő algoritmus.

**Biz.:** Azt kell belátni, hogy $\sum_j C_j(S_{A2}) \le 2\sum_j C_j(S^*)$.

A 2. lépés során az egyik gépen nincs változás, a másik gépen az erőforrást igénylő munka előtti munkák (ha vannak ilyenek) befejezési idői nem változnak, a többi munka befejezési idői legfeljebb az erőforrást igénylő munka megmunkálási idejével nő (legyen ez $p^1$). Azoknak az erőforrást nem igénylő munkáknak, melyeket eltoltunk, a megmunkálási idői legyenek rendre $p^2,\ldots,p^k$. Az SPT sorrend miatt $p^1 \le p^2 \le \ldots \le p^k$. Tehát

$$\sum_j C_j(S_{A2}) \le \sum_j C_j(S_{SPT}) + kp^1 \le \sum_j C_j(S_{SPT}) + p^1 + p^2 + \cdots + p^k \le 2\sum_j C_j(S_{SPT}),$$

és mivel $\sum_j C_j(S^*) \ge \sum_j C_j(S_{SPT})$, ezért $\sum_j C_j(S_{A2}) \le 2\sum_j C_j(S^*)$. ■

Nézzük most a probléma általánosítását, a $\mathrm{PD}m \mid \mathrm{res}111 \mid \sum_j C_j$ problémát, ahol mindegyik gépen 1-1 darab erőforrást igénylő munka van.

**Am algoritmus**

*Input:* Egy példánya a $\mathrm{PD}m \mid \mathrm{res}111 \mid \sum_j C_j$ problémának, ahol mindegyik gépen 1-1 darab erőforrást igénylő munka van

*Output:* Egy $S_{Am}$ heurisztikus ütemezés

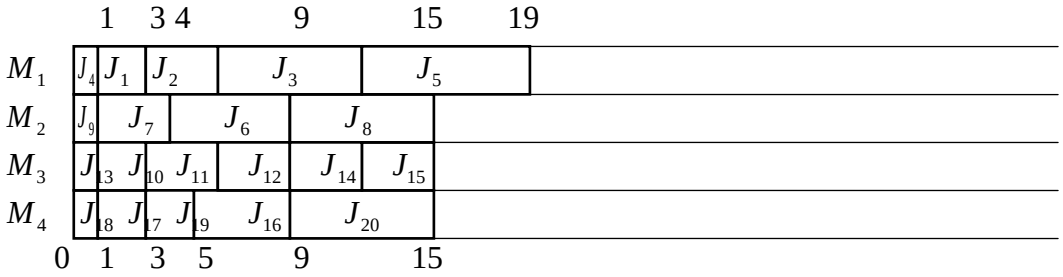
1. $S_{SPT}$ ütemezés meghatározása.
2. Ha most az erőforrást igénylő munkák között van átfedés, akkor menjünk a 3. lépésre; különben menjünk az 5. lépésre.
3. Jelölje átmenetileg $J_0$ azt az erőforrást igénylő munkát, amelyre létezik másik erőforrást igénylő munka, hogy köztük van átfedés, és az ilyen tulajdonságú munkák között a legkorábban fejeződik be. Ha több ilyen is van, akkor mindegy, hogy melyiket választjuk.
4. Válasszuk ki azokat az erőforrást igénylő munkákat, amelyekre van átfedés köztük és $J_0$ között. Ezeket és az utánuk következő munkákat toljuk jobbra úgy, hogy az erőforrást igénylő munkák kezdési idői a $J_0$ befejezési ideje legyen. Menjünk a 2. lépésre.

5. $S_{Am}$: az eredményül kapott ütemezés. Stop.

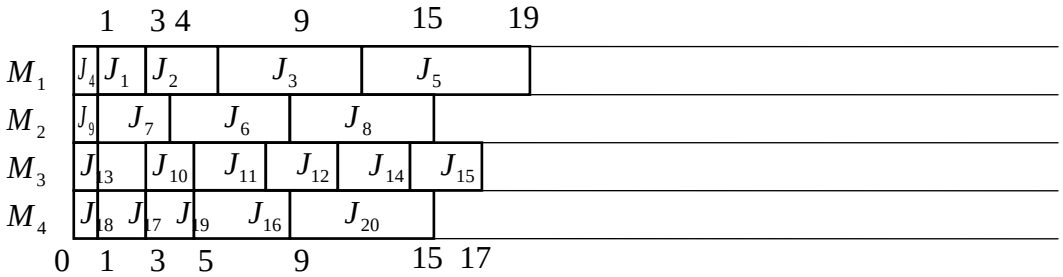
**6. Példa:** $m := 4$, $N := \{J_1,\ldots,J_{20}\}$, $J_1,\ldots,J_{20}$ megmunkálási idői legyenek ugyanannyiak, mint a 4. Példában. $N_1 := \{J_1,\ldots,J_5\}$, $N_2 := \{J_6,\ldots,J_9\}$, $N_3 := \{J_{10},\ldots,J_{15}\}$, $N_4 := \{J_{16},\ldots,J_{20}\}$.

Legyenek az erőforrást igénylő munkák: $J_1$, $J_6$, $J_{10}$, $J_{16}$.

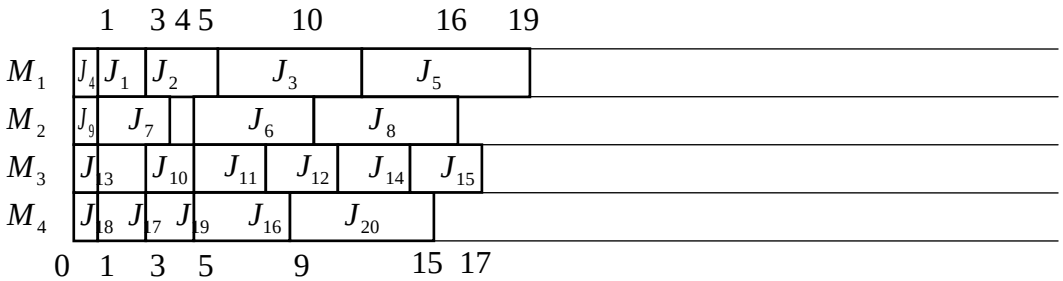
$S_{SPT}$:




$J_0 := J_1$

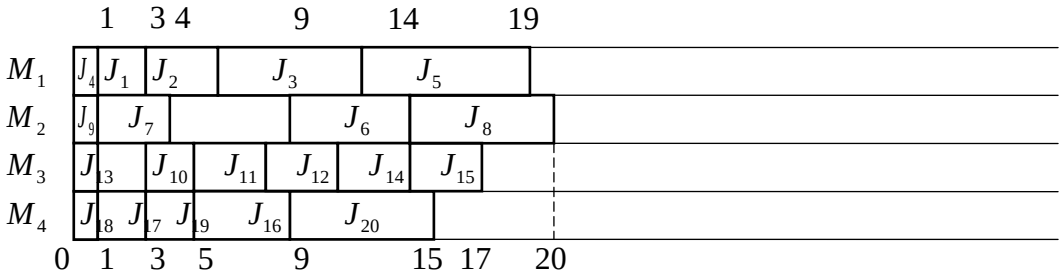



$J_0 = J_{10}$



$J_0 = J_{16}$

$S_{A4}$:




**19. Tétel:** Az Am algoritmus $2^{m-1}$-közelítő algoritmus.

**Biz.:** Azt kell belátni, hogy $\sum_j C_j(S_{Am}) \le 2^{m-1} \sum_j C_j(S^*)$.

$S_0 := S_{SPT}$. Jelölje $S_i$ azt az ütemezést, melyet $S_{i-1}$-ből kapunk a 4. lépés végrehajtása során, $i = 1,2,\ldots,k$, $S_k = S_{Am}$.

Világos, hogy a 4. lépés után az átmenetileg $J_0$-lal jelölt erőforrást igénylő munka és a többi erőforrást igénylő munka között nincs átfedés, és a későbbiek során sem lesz, így a 4. lépést legfeljebb $(m-1)$-szer hajtjuk végre, tehát $k \le m-1$.

A 4. lépés során, amikor az $S_{i-1}$-ből kapjuk $S_i$-t, a gépeken a befejezési idők vagy nem változnak, vagy legfeljebb a géphez tartozó erőforrást igénylő munka megmunkálási idejével nőnek. Azoknak az erőforrást igénylő munkáknak, melyeket eltoltunk, a megmunkálási idői legyenek rendre $p^{1,1}, p^{1,2},\ldots, p^{1,l}$, ahol $l < m$, mert $J_0$-t nem toltuk el. Azoknak az erőforrást nem igénylő munkáknak, melyeket eltoltunk, a megmunkálási idői legyenek rendre $p^{2,1}, p^{3,1},\ldots, p^{k_1,1}, p^{2,2}, p^{3,2},\ldots, p^{k_2,2},\ldots, p^{2,l}, p^{3,l},\ldots, p^{k_l,l}$.

Az SPT sorrend miatt $p^{1,x} \le p^{2,x} \le \ldots \le p^{k_x,x}$, $x = 1,2,\ldots,l$.

$\sum_j C_j(S_i) \le \sum_j C_j(S_{i-1}) + k_1 p^{1,1} + \cdots + k_l p^{1,l}$,

$k_x p^{1,x} \le p^{1,x} + p^{2,x} + \cdots + p^{k_x,x}$, $x = 1,2,\ldots,l$,

$p^{1,1} + p^{2,1} + \cdots + p^{k_1,1} + p^{1,2} + p^{2,2} + \cdots + p^{k_2,2} + \cdots + p^{1,l} + p^{2,l} + \cdots + p^{k_l,l} \le \sum_j C_j(S_{i-1})$,

tehát $\sum_j C_j(S_i) \le 2\sum_j C_j(S_{i-1})$, $i = 1,2,\ldots,k$.

Ebből következik, hogy

$\sum_j C_j(S_{Am}) = \sum_j C_j(S_k) \le 2\sum_j C_j(S_{k-1}) \le 2^2 \sum_j C_j(S_{k-2}) \le \ldots \le 2^k \sum_j C_j(S_0)$

$= 2^k \sum_j C_j(S_{SPT}) \le 2^k \sum_j C_j(S^*) \le 2^{m-1} \sum_j C_j(S^*)$. ■

# 6. Összegzés

A dolgozatban olyan ütemezési problémák megoldása volt a cél, ahol a munkák feldolgozásához a gépeken túl, további erőforrásokra is szükség van, és ahol a gépek párhuzamosak és dedikáltak.

A célfüggvény először a teljes átfutási idő volt, lásd [2,3]. Megtudtuk, hogy melyik probléma oldható meg polinomiális idő alatt, és melyik nem:

Azoknak a problémáknak a komplexitása, melyekben egy munka egynél több erőforrást is használhat:

| Gépek száma | Erőforrások száma | Erőforrás mérete | Felhasználás felső korlátja | Komplexitás | Hivatkozás |
|---|---|---|---|---|---|
| 2 | 1 | $\sigma$ | $\sigma$ | $O(n \log n)$ | 8. Tétel |
| 2 | 2 | 1 | 1 | $O(n)$ | 9. Tétel |
| 2 | 2 | 2 | 2 | o.NP | 10. Tétel |
| 2 | 3 | 1 | 1 | o.NP | 11. Tétel |
| 3 | 1 | 1 | 1 | o.NP | 3. Tétel |

Azoknak a problémáknak a komplexitása, melyekben az erőforrások egységnyi méretűek, és egy munkának legfeljebb egy erőforrásra van szüksége:

| Gépek száma | Erőforrások száma | Átállítási idők | Eltávolítási idők | Erőforrást nem igénylő munkák megszakíthatóak | Komplexitás | Hivatkozás |
|---|---|---|---|---|---|---|
| 2 | $\lambda$ | Igen | Igen | Nem | $O(n \log n)$ | 12. Tétel |
| 3 | 1 | Nem | Nem | Nem | o.NP | 3. Tétel |
| $m$ | 1 | Nem | Nem | Nem | s.NP | 4. Tétel |
| 3 | 2 | Igen | Nem | Igen | o.NP | 15. Tétel |
| 3 | 3 | Nem | Nem | Igen | o.NP | 14. Tétel |
| $m$ | 1 | Nem | Nem | Igen | $O(n)$ | 2. Tétel |
| $m$ | 2 | Nem | Nem | Igen | $O(m+n)$ | 13. Tétel |

A táblázatokban „o.NP” azt jelenti, hogy gyengén NP-nehéz, „s.NP” pedig azt, hogy erősen NP-nehéz.

Az NP-nehéz problémákat közelítő algoritmusokkal (pl. PTAS) lehet megoldani, lásd 6., 7., 16. és 17. Tétel.

Ezután az 5. fejezetben a célfüggvény az átlagos átfutási idő volt. Amíg a $\mathrm{PD2}\,|\,\mathrm{res111}\,|\,C_{\max}$ probléma megoldása nagyon egyszerű volt (lásd 1. Tétel), addig a $\mathrm{PD2}\,|\,\mathrm{res111}\,|\,\sum_j C_j$ probléma komplexitása még abban az esetben is kérdés maradt, amikor mindkét gépen 1-1 darab erőforrást igénylő munka van.

A problémára és általánosítására ( $\mathrm{PD}m\,|\,\mathrm{res111}\,|\,\sum_j C_j$ , ahol mindegyik gépen 1-1 darab erőforrást igénylő munka van) megengedett megoldást egy nem túl hatékony közelítő algoritmussal sikerült találni, lásd 18. és 19. Tétel.

## Hivatkozások

[1] Jordán Tibor, Ütemezés, egyetemi jegyzet (2007).

[2] H. Kellerer, V.A. Strusevich, Scheduling parallel dedicated machines under a single non-shared resource, European J. Oper. Res. 147 (2003) 345-364.

[3] H. Kellerer, V.A. Strusevich, Scheduling problems for parallel dedicated machines under multiple resource constraints, Discrete Appl. Math. 133 (2004) 45-68.

[4] T. Gonzalez, S. Sahni, Open shop scheduling to minimize finish time, J. Assoc. Comput. Mach. 23 (1976) 665-679.

[5] V.A. Strusevich, Two-machine open shop sheduling problem with setup, processing and removal times separated, Comput. Oper. Res. 20 (1993) 597-611.

[6] E.L. Lawler, J. Labetoulle, On preemptive scheduling of unrelated parallel processors by linear programming, J. Assoc. Comput. Mach. 25 (1978) 612-619.

[7] Peter Brucker, Scheduling Algorithms, Fifth Edition, Springer (2007): 158-160.