Bárány Imre—Fiala Tibor

# Többgépes ütemezési problémák közel optimális megoldása

## 1. Bevezetés

A következő problémával foglalkozunk. Legyen adva $m$ gép és $n$ munkadarab. Minden munkadarabot minden gépen meg kell munkálni, ismert az $i$-edik munkadarab megmunkálásához szükséges idő a $j$-edik gépen: $t_{ij}$. Egy munkadarab egyszerre csak egy gépen lehet, és egy gépen sem lehet egyszerre két munkadarabot megmunkálni. Az $i$-edik munkadarab megmunkálása a $j$-edik gépen nem szakítható meg (minden $i$-re és $j$-re). Ezután egy ütemezést úgy lehet megadni, hogy definiálunk bizonyos $\tau_{ij}$ kezdési időket az $i$-edik munkadarabra a $j$-edik gépen, a befejezési idő ekkor $\tau_{ij} + t_{ij}$ lesz. A fenti feltételek azt jelentik, hogy az $I_{ij} = (\tau_{ij}, \tau_{ij} + t_{ij})$ intervallumok bármely rögzített $i = 1, 2, \ldots, n$, illetve bármely rögzített $j = 1, 2, \ldots m$ mellett páronként diszjunktak. Az a cél, hogy az utolsó befejezési idő, $\max\limits_{i,j}(\tau_{ij} + t_{ij})$ minimális legyen. Ez az *open shop* feladat.

A *flow shop* feladatnál van még egy megszorítás: minden munkadarabot először az első, aztán a második, ... végül az $m$-edik gépen kell megmunkálni.

Cikkünket a következőképpen tagoltuk. A második rész tartalmazza a flow shop feladatra vonatkozó tételeinket és megjegyzéseket, a harmadik az open-shop problémára vonatkozókat. A negyedik részben a bizonyításokat adjuk meg, itt szerepel annak a három lemmának a kimondása és igazolása is, amelyekre a bizonyításokban szükség lesz.

## 2. A flow shop feladat

A flow shop feladat két gép esetén egyszerűen megoldható (Johnson 1954), a Johnson-féle algoritmus lépésszáma $O(n \cdot \log n)$. Három vagy több gép esetén azonban a feladat $NP$-teljes (Garey, Johnson, Sethi 1976), még abban az esetben is, ha a bemenő adatok hosszát a $[t_{ij}]$ mátrix helyett a $\sum\limits_i \sum\limits_j t_{ij}$ mennyiség segítségével mérjük. Emiatt nincs remény arra, hogy $m \geq 3$ esetén a flow shop feladatot effektív algoritmus segítségével teljes általánosságban meg lehessen oldani. Ez teszi indokolttá olyan algoritmusok keresését, amelyek valamilyen értelemben közelítőleg optimális megoldást szolgáltatnak. Ilyen algoritmusokat, vagy heurisztikus eljárásokat többen vizsgáltak már (Palmer 1965, Campbell 1970, Fejes 1974, Graham, Lawler, Lenstra, Rinnooy Kan 1979). Mi is megadunk egy közel optimális algoritmust. Eredményünk ismertetéséhez szükségünk lesz néhány fogalom és jelölés bevezetésére.

Egy ütemezést permutáció szerintinek nevezünk, ha a munkadarabok sorrendje minden gépen ugyanaz. Ez a sorrend az 1, 2, ..., $n$ számok egy $\pi$ permutációjával adható meg. Közismert (TANAJEV, SKURBA 1975), hogy két és három gép esetén az optimális ütemezések között van permutáció szerinti, négy vagy több gép esetén viszont ez általában nem igaz. Megjegyezzük, hogy egy $\pi$ permutáció még nem határoz meg egyértelműen egy ütemezést, azonban a $\pi$ permutáció szerinti ütemezések közül könnyű kiválasztani a legrövidebb átfutási idejűt. Legyen ennek átfutási ideje $T(\pi)$. Az optimális átfutási időt $T_{\text{opt}}$-tal jelöljük. Legyen továbbá

$$K = \max_{i,j} t_{ij},\ M_j = \sum_{i=1}^{n} t_{ij} \quad (j = 1, \ldots, m),$$
$$M = \max_j M_j.$$

Világos, hogy $M \leq T_{\text{opt}} \leq T(\pi)$ minden $\pi$ permutáción.

1. *tétel.* Létezik olyan $\pi$ permutáció, melyre

$$M \leq T_{\text{opt}} \leq T(\pi) \leq M + m(m-1)K.$$

Ez a permutáció megadható egy $O(n^2m^3)$ lépésszámú algoritmussal.

Ezt a tételt a következő geometriai eredményből fogjuk levezetni.

2. *tétel.* Legyen adva egy olyan $V \subset R^d$ vektorhalmaz, hogy $\|v\| \leq 1$ minden $v \in V$ esetén és $\sum_{v \in V} v = 0$. Ekkor $V$ elemei felírhatók egy $v_1, v_2, \ldots, v_n$ sorrendben úgy, hogy

$$\max_{1 \leq k \leq n} \|\sum_{i=1}^{k} v_i\| \leq d.$$

Ez a sorrend megadható egy $O(n^2m^3)$ lépésszámú algoritmussal.

Itt $\|\ \|$ valamilyen $R^d$-beli normát jelöl, az 1. tételben a maximum normára lesz szükségünk. A 2. tétel azt mondja ki, hogy tetszőleges, legfeljebb egységnyi hosszúságú $d$-dimenziós vektorokból álló zárt töröttvonal átendezhető úgy, hogy az átrendezett töröttvonal teljes egészében benne fekszik a $C(d)$ sugarú gömbben, ahol $C(d)$ a $V$-től (és annak elemszámától) független konstans.

Az 1. tétel lényege az, hogy megadható — méghozzá polinomiális algoritmussal — egy olyan permutáció, hogy $T_\pi - T_{\text{opt}} \leq C(m, K)$, ahol a $C(m, K)$ konstans nem függ a munkadarabok számától. Tehát, ha $K$ rögzített és $n \to \infty$, akkor aszimptotikusan optimális ütemezést szolgáltat.

BELOV és SZTOLIN vették észre 1974-ben, hogy egy $T_\pi - T_{\text{opt}} \leq C(m, K)$ típusú becslés következik a vektor-átrendezési tételből. KADEC (1953) egy tétele kimondja, hogy $C(d) \leq 2^d$ az euklideszi normára, ezt felhasználva Belov és Sztolin megmutatta, hogy $C(m, K) \leq 2^{m-1} \cdot (m-1) \cdot K$, algoritmusuk lépésszáma $O(n^m)$. Tőlük függetlenül FIALA 1977-ben szintén megtalálta a két állítás közötti kapcsolatot. 1978-ban SZEVASZTYANOV megmutatta, hogy $C(d) \leq d$ $\big(O(d \cdot 2^d \cdot n^2)$ lépésszámú algoritmussal$\big)$, ebből igazolta, hogy $C(m, K) \leq (m-1) \cdot K$. Tőle függetlenül, ugyancsak 1978-ban BÁRÁNY $C(d) \leq \dfrac{3(d-1)}{2}$-t kapott egy $O(n^2d^3 + nd^4)$ lépésszámú algoritmussal, ebből $C(m, K) \leq \dfrac{3}{2}(m-1)^2 \cdot K$ következik. Végül Szevasztyanov és Bárány módszereit továbbfejlesztve GRINBERG és SZEVASZTYANOV igazolták a 2. tételt 1980-ban.

## 3. Az open shop feladat

Adott $S$ ütemezés esetén átfutási időnek nevezzük, és $T_S$-sel jelöljük a legelső művelet elkezdésétől a legutolsó befejezéséig eltelt időt. Első célunk az optimális átfutási idő becslése. Jelöléseink egyeznek a flow-shop feladat esetén látottakkal, tehát

$M_j = \sum_{i=1}^{n} t_{ij}$, a $j$-edik gépen elvégzendő műveletek összideje;

$L_i = \sum_{j=1}^{m} t_{ij}$, az $i$-edik munkadarab megmunkálásához szükséges összidő;

$$M = \max_j M_j$$

$$L = \max_i L_i .$$

Tetszőleges $S$ ütemezés esetén a $T_S$ átfutási időre nyilvánvalóan érvényes az alábbi becslés

$$T_S \geq \max(M, L). \tag{1}$$

Eszerint, ha olyan ütemezést tudunk megadni, melynél $T_S = M$ vagy $T_S = L$, akkor ez az ütemezés optimális. Jelöljük a $t_{ij}$ műveleti idők maximumát $K$-val! Legyen $S$ egy tetszőleges ütemezés, melynél egy gép csak akkor áll, ha nincs olyan művelet, melyhez hozzáfoghatna. Az ilyen ütemezéseket mohó ütemezésnek hívjuk a továbbiakban.

*Racsmány Annától* számazik a következő fontos megjegyzés (szóbeli közlés):

*Állítás.* Tetszőleges $S$ mohó ütemezés esetén

$$T_S \leq M + (m-1)K.$$

*Bizonyítás.* Legyen $j$ annak a gépnek a sorszáma, amelyik utolsóként fejezi be a munkát, és tekintsük a $j$-edik gép ütemezését jelző idődiagramot az 1. ábrán. Egyik állásidő alatt sem lehetett az $i_n$ munkadarabot megmunkálni. Ennek csak az lehetett az oka, hogy mindegyik állásidő alatt az $i_n$ munkadarab megmunkálás alatt áll valamelyik másik gépen.

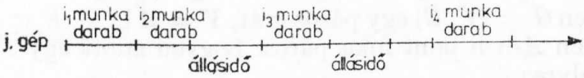
Az állásidők összege legfeljebb

$$\sum_{l \neq j} t_{i_n, l} \leq (m-1)K.$$

Ezért

$$T_S \leq \sum_{i=1}^{n} t_{ij} + (m-1)K \leq M + (m-1)K.$$

2 gép esetén mindig elérhető, hogy az átfutási idő egyezzen az (1) triviális alsó korláttal. Ezt T. Gonzalez és S. Sahni (1976) igazolta egy $O(n)$ műveletigényű algoritmussal.




*1. ábra*

Könnyű példát mutatni arra, hogy három vagy annál több gép esetén előfordulhat, hogy

$$T_{\text{opt}} > \max(M, L).$$

Azt is tudjuk, hogy három vagy több gép esetén az open shop probléma $NP$-teljes (GONZALEZ–SAHNI, 1976), ha az input méreteként az $n \cdot m$ szorzatot használjuk. Ezért nincs remény arra, hogy a problémát effektív algoritmusokkal meg lehessen oldani teljes általánosságban. Az átfutási idő alsó és felső becslése azt mutatja, hogy ha $M > L$, akkor minden mohó ütemezés közel-optimális eredményt ad. Cikkünkben azt fogjuk megmutatni, hogy ha $M$ elég nagy, akkor az optimális ütemezés is megadható effektív algoritmusokkal. Ezt állítja a következő

3. *tétel.* Minden $m$-re van olyan $c(m)$ konstans, hogy

$$M \geq c(m)K \text{ esetén } T_{\text{opt}} = M. \tag{2}$$

Természetesen az a cél, hogy minél kisebb $c(m)$-re tudjunk olyan algoritmust adni, amelyik $M$ átfutási idejű optimális ütemezést ad.

Jelöljük $c_{\min}(m)$-mel azoknak a $c(m)$ konstansoknak a minimumát (infimumát), melyekre (2) érvényes. Ekkor érvényesek a következő tételek.

4. *tétel.*

$$c_{\min}(m) \leq m^2 + 2m - 1\,,$$

és $M \geq (m^2 + 2m - 1)K$ esetén $O(n^2m^3)$ lépésben adható olyan $S$ optimális ütemezés, melyre

$$T_S = M.$$

A 4. tételt a 2. tételből fogjuk levezetni az 1. tétel bizonyításához analóg konstrukcióval.

Nehezebb konstrukció segítségével élesebb becslés is kapható $c_{\min}(m)$ nagyságrendjére.

5. *tétel.*

$$c_{\min}(m) \leq (16\, m \log_2 m + 21\, m)K\,,$$

és $M \geq (16\, m \log_2 m + 21\, m)K$ esetén $O(nm^3)$ lépésben adható olyan $S$ optimális ütemezés, melyre

$$T_S = M.$$

Világos, hogy a 3. tétel triviálisan adódik akár a 4. akár az 5. tételből.

Az 5. tétel algoritmikus bizonyításának magvát az alábbi eredmény képezi. Jelöljünk $G = (V, E)$-vel egy páros gráfot, ahol $V = A \cup B$ a pontok halmaza és $E \subset A \times B$ az élek halmaza. Tetszőleges $S \subset E$, $x \in V$ esetén az $x$ pontra illeszkedő élek halmaza legyen $H(x, S)$. Ha $\lambda : E \to R$ egy adott súlyfüggvény, akkor legyen

$$\lambda(x, S) = \sum_{e \in H(X,s)} \lambda(e).$$

6. *tétel.* Legyen $G = (V, E)$ egy páros gráf, $V = A \cup B$, $E \subset A \times B$ és tegyük fel, hogy minden $A$-beli pont foka páros. Legyen adott egy $\lambda : E \to R^+$ súlyfüggvény, amelyre

$$\lambda(e) \leq K \quad (e \in E).$$

Ekkor $O(|E| \cdot \min(|A|, |B|))$ lépésben megadható az éleknek egy $E = E_1 \cup E_2$ particiója úgy, hogy

$$x \in A \text{ esetén} \qquad |H(x, E_1)| = |H(x, E_2)| = \frac{1}{2} |H(x, E)|,$$

$$x \in B \text{ esetén} \qquad \left|\lambda(x, E_1) - \frac{1}{2}\lambda(x, E)\right| \leq K,$$

$$\text{és} \qquad \left|\lambda(x, E_2) - \frac{1}{2}\lambda(x, E)\right| \leq K.$$

Ezt a tételt nem fogjuk bizonyítani, mert a bizonyítás elég hosszadalmas (lásd FIALA 1979).

## 4. Bizonyítások

Szükségünk lesz három lemmára.

1. *lemma.* Legyen $x^0$ az

$$\begin{aligned} \langle a_i, x\rangle &= b_i \quad (i \in I), \\ \langle a_j, x\rangle &\leq b_j \quad (j \in J) \end{aligned} \tag{3}$$

relációkkal adott $H \subseteq R^n$ konvex halmaznak egy extremális pontja, itt $I$ és $J$ véges indexhalmazok. Ekkor $x^0$-ban legalább $n - |I|$ számú $\langle a_j, x\rangle \leq b_j$ alakú egyenlőtlenség egyenlőségként teljesül.

*Bizonyítás.* Legyen $B = \{j \in J \mid \langle a_j, x\rangle = b_j\}$. Ha most $|I \cup B| < n$, akkor tekintsük a következő homogén egyenletrendszert ($y \in R^n$):

$$\begin{aligned} \langle a_i, y\rangle &= 0 \quad i \in I, \\ \langle a_j, y\rangle &= 0 \quad j \in B. \end{aligned}$$

Itt $n$ változóra $n$-nél kevesebb egyenletünk van, tehát létezik nullától különböző $y^0 \in R^n$ megoldás. Ekkor azonban minden elég kis abszolút értékű $t$-re

$$x^0 + ty^0 \in H,$$

ami ellentmond annak, hogy $x^0$ $H$-nak extremális pontja.

2. *lemma.* Legyenek $v_1, \ldots, v_n$ $m$-dimenziós vektorok, amelyekre $\|v_i\| \leq K$ és $\sum_{i=1}^{n} v_i = v$. Ekkor $O(nm^3)$ lépésben megadható $\{1, \ldots, n\}$-nek egy olyan $S \cup R$ particiója, hogy

$$\left\|\sum_{i\in S} v_i - \frac{1}{2} v\right\| \leq \frac{n}{2} K \text{ és } \left\|\sum_{i\in R} v_i - \frac{1}{2} v\right\| \leq \frac{n}{2} K.$$

*Bizonyítás.* Tekintsük a

$$P = \left\{x \in R^n \,\middle|\, \sum_{i=1}^{n} x_i v_i = \frac{v}{2},\ 0 \leq x \leq 1\right\}$$

poliédert. Ez a poliéder nemüres, mert $\left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right) \in P$, legyen $x^*$ a $P$-nek egy extremális pontja. Az 1. lemma szerint $x_i^* = 0$ vagy $x_i^* = 1$ legfeljebb $m$ koordináta kivételével. Legyen

$$A = \{i \mid x_i^* = 1\},$$

$$B = \left\{i \mid 1 > x_i^* > \frac{1}{2}\right\},$$

$$C = \left\{i \mid \frac{1}{2} \geq x_i^* > 0\right\}.$$

Legyen továbbá $S = A \cup B$ és $R = \{1, \ldots, n\} \setminus S$. Ekkor

$$\sum_{i \in S} v_i - \frac{v}{2} = \sum_{i \in S} v_i - \sum_{i=1}^{n} x_i^* v_i = \sum_{i \in B} (1 - x_i^*) v_i + \sum_{i \in C} (-x_i^*) v_i .$$

$B$ és $C$ együttes elemszáma legfeljebb $m$, az $1 - x_i^*$ $(i \in B)$ és $-x_i^*$ $(i \in C)$ együtthatók abszolú értéke legfeljebb $\frac{1}{2}$, tehát

$$\left\| \sum_{i \in S} v_i - \frac{1}{2} v \right\| \leq \frac{m}{2} K .$$

Másrészt

$$\sum_{i \in R} v_i - \frac{1}{2} v = - \left( \sum_{i \in S} v_i - \frac{1}{2} v \right),$$

tehát a lemma másik állítása is teljesül. Az $x^*$ extremális pont előállításához legfeljebb $n$-szer kell egy (3) alakú egyenletrendszert megoldani. Egy megoldás pl. *Gauss*-algoritmussal $O(m^3)$ lépést igényel, így a teljes lépésszám $O(nm^3)$.

3. *lemma.* Legyen $G = (V, E)$ páros gráf, $V = A \cup B$, $E \subseteq A \times B$, $|B| = m$, $|A| = s$. Tegyük fel, hogy a $\lambda : E \to R^+$ súlyfüggvényre $\lambda(e) \leq K$ (minden $e \in E$-re). Legyen továbbá $2^{k-1} < m \leq 2^k = m'$. Ekkor $O(s \cdot m \cdot \log_2 m \times \times \min(s, m))$ lépésben megadható az éleknek egy $E = E_1 \cup E_2 \cup \ldots \cup E_{m'}$ particiója úgy, hogy

(A) $x \in A$, $1 \leq j \leq m'$ esetén $x$ foka $E_j$-ben legfeljebb 1,

(B) $x \in B$, $1 \leq j \leq m'$ esetén

$$\lambda(x, E_j) \geq \frac{1}{m'} \lambda(x, E) - 2K ,$$

(C) $x \in B$, $1 \leq i \leq m'$ esetén

$$\left| \lambda(x, \bigcup_{s=1}^{i} E_s) - \frac{i}{m'} \lambda(x, E) \right| \leq K \log m' .$$

*Bizonyítás.* Először azzal az esettel foglalkozunk, mikor $m = m' = 2^k$, ebből az általános eset egyszerűen következik majd. Először az $E$ élhalmazt a 6. tétel szerint felbontjuk $E^0 \cup E^1$ alakban, ekkor

$x \in A$, $i = 0, 1$ esetén $x$ foka $E^i$-ben $2^{m-1}$, és

$x \in B$, $i = 0, 1$ esetén $\left| \lambda(x, E^i) - \frac{1}{2} \lambda(x, E) \right| \leq K$.

Ha most $i_1 = 0, 1$; $i_2 = 0, 1$; $\ldots$ $i_s = 0, 1$ $(s < K)$ $E^{i_1 i_2 \ldots i_s}$ már értelmezve van úgy, hogy

$x \in A$ esetén $x$ foka $E^{i_1 \ldots i_s}$-ben $2^{m-s}$, és

$$x \in B \text{ esetén } \left| \lambda(x, E^{i_1 \ldots i_s}) - \frac{1}{2} \lambda(x, E^{i_1 \ldots i_{s-1}}) \right| \leq K,$$

akkor megint a 6. tétel szerint $E^{i_1 \ldots i_s}$-et fel lehet bontani két diszjunkt $E^{i_1 \ldots i_s 0}$; és $E^{i_1 \ldots i_s 1}$ részre úgy, hogy

$x \in A$, $i = 0, 1$ esetén $x$ foka $E^{i_1 \ldots i_s i}$-ben $2^{m-s-1}$ és

$$x \in B,\ i = 0, 1 \text{ esetén } \left| \lambda(x, E^{i_1 \ldots i_s i}) - \frac{1}{2} \lambda(x, E^{i_1 \ldots i_s}) \right| \leq K.$$

Ezt a konstrukciót elvégezve $s = 1, 2, \ldots, k - 1$-re $E^{i_1 \ldots i_k}$ $(i_1 = 0, 1; \ldots i_k = 0, 1)$ értelmezve van.

Legyen $j \in \{1, 2, \ldots, m\}$, ekkor $j - 1$ diadikus alakja:

$$j - 1 = 2^k \left( \frac{i_1}{2} + \frac{i_2}{4} + \ldots + \frac{i_k}{2^k} \right), \text{ ahol } i_s = 0 \text{ vagy } 1.$$

Ezzel megadjuk az $E_j$ halmazt, legyen

$$E_j = E^{i_1 \ldots i_k}.$$

A konstrukcióból azonnal adódik, hogy az $E_j$ halmazok páronként diszjunktak és

$$x \in A \text{ esetén } x \text{ foka } E_j\text{-ben } 1.$$

$t$ szerinti indukcióval könnyen igazolható, hogy $x \in B$ és $t = 1, 2, \ldots, k$ esetén

$$\left| \lambda(x, E^{i_1 \ldots i_t}) - \frac{1}{2^t} \lambda(x, E) \right| \leq 2K.$$

Speciálisan $t = k$ mellett

$$x \in B,\ 1 \leq j \leq m \text{ esetén } \lambda(x, E_j) \geq \frac{1}{m} \lambda(x, E) - 2K.$$

Végül legyen $l \in \{1, \ldots, m\}$ és $l - 1$ diadikus alakja

$$l - 1 = 2^k \left( \frac{i_1}{2} + \frac{i_2}{4} + \ldots + \frac{i_n}{2^k} \right).$$

Ha most $x \in B$, akkor

$$\left| \lambda(x, \bigcup_{j=1}^{l} E_j) - \frac{l}{m} \lambda(x, E) \right| = \left| \sum_{t=1}^{k} i_t(\lambda(x, E^{i_1 \ldots i_t}) - \frac{1}{2^t} \lambda(x, E)) \right| \leq$$
$$\leq kK = K \cdot \log_2 m.$$

Ezzel a $(C)$ tulajdonságot is ellenőriztük, tehát kész vagyunk az $m = m'$ esettel.

Legyen tehát $m < m'$. Egészítsük ki a $B$ halmazt $m' - m$ új ponttal, és ezzel a bővebb halmazzal és az eredeti $A$-val képezzünk egy teljes páros gráfot! Az új éleken $\lambda$-t nullának definiálva azonnal adódik a lemma állítása az eddig bizonyítottak alapján. A lépésszám becslését a 6. tétel lépésszámából kapjuk, hiszen a 6. tétel algoritmusát $\log_2 m'$-ször kell alkalmaznunk.

*A 2. tétel bizonyítása* (*Grinberg–Szevasztyanov*). Indukció segítségével megkonstruáljuk a $V_d \subset V_{d+1} \subset \ldots \subset V_n = V$ halmazokat és a $\lambda_i : V_i \to R^1$ számokat úgy, hogy

$$| V_i | = i, \quad 0 \leq \lambda_i(x) \leq 1 \ (\forall i \text{ és } \ \forall x \in V_i\text{-re}),$$

$$\sum_{x \in V_i} \lambda_i(x) = i - d \ (\forall i),$$

$$\sum_{x \in V_i} \lambda_i(x) x = 0 \qquad (\forall i).$$

Az indukciót $i = n$-nel kezdjük és visszafelé megyünk. $\underline{i = n}$: Legyen $V_n = V$ és $\lambda_n(x) = \dfrac{n-d}{n}$ $(\forall x \in V_n\text{-re})$. $\underline{i + 1 \to i}$: Tekintsük azokat a $(\mu(x) \mid x \in V_{i+1})$ vektorokat, amelyek kielégítik a következő lineáris rendszert:

$$0 \leq \mu(x) \leq 1 \quad (x \in V_{i+1}),$$

$$\sum_{x \in V_{i+1}} \mu(x) = i - d,$$

$$\sum_{x \in V_{i+1}} \mu(x) x = 0.$$

A megoldáshalmaz konvex és kompakt, továbbá nemüres, mert például

$$\mu(x) = \frac{i-d}{i+1-d} \lambda_{i+1}(x)$$

hozzátartozik. Legyen $\mu^0(x)$ a megoldáshalmaz egy csúcsa. Állítjuk, hogy $\mu^0(x) = 0$ valamely $x \in V_{i+1}$ esetén. Ellenkező esetben ugyanis $\mu^0(x) > 0$ minden $x \in V_{i+1}$-re, a lemma miatt $\mu^0(x) = 1$ legalább $(i + 1) - (d + 1) = i - d$ darab $x \in V_{i+1}$-re. Ebből viszont $\sum_{x \in V_{i+1}} \mu^0(x) > i - d$ következne. Tehát valamely $x^0 \in V_{i+1}$-re $\mu^0(x^0) = 0$. Legyen most $V_i = V_{i+1} \backslash \{x^0\}$ és $\lambda_i(x) = \mu^0(x)$ $(x \in V_i)$.

Ezzel a konstrukciót befejeztük. Világos, hogy az algoritmus lépésszáma $O(n^2 d^3)$.

Legyen most $v_i = V_i \backslash V_{i-1}$, ha $i > d$, $v_1, \ldots, v_d$ pedig a maradék vektorok valamilyen sorrendje. Ekkor ha $k \leq d$

$$\| \sum_{i=1}^{k} v_i \| \leq \sum_{i=1}^{k} \| v_i \| \leq k \leq d .$$

Másrészt, ha $k > d$, akkor

$$\| \sum_{i=1}^{k} v_i \| = \| \sum_{i=1}^{k} v_i - \sum_{x \in V_k} \lambda_k(x) x \| = \| \sum_{x \in V_k} x - \sum_{x \in V_k} \lambda_k(x) x \| =$$

$$= \| \sum_{x \in V_k} (1 - \lambda_k(x)) x \| \leq \sum_{x \in V_k} (1 - \lambda_k(x)) = k - (k - d) = d .$$

Az 1., 4. és 5.-tétel bizonyításához úgynevezett *redukált feladatot* értelmezünk, melynél mindegyik gépnek ugyanannyit kell dolgoznia. Pontosan fogalmazva az eredeti $t_{ij}$ idők helyett olyan $t'_{ij}$ fiktív időket vezetünk be, melyekre

$$t_{ij} \leq t'_{ij} \leq K, \text{ és}$$

$$\sum_{i=1}^{n} t'_{ij} = M \quad j = 1, 2, \ldots, m\text{-re.}$$

Ilyen $t'_{ij}$ számok $O(mn)$ lépésben megadhatók, hiszen $n \cdot K \geq M$, tehát $M_j < M$ esetén az $M - M_j$ különbséget szét lehet darabolni $n$ részre úgy, hogy az $i$-edik rész ne haladja meg a $K - t_{ij}$ értéket. Ha az $i$-edik részt hozzáadjuk a $t_{ij}$-hez, megkapjuk a kívánt $t'_{ij}$ időket. Ha most a $t'_{ij}$ idők által definiált problémára egy $S'$ ütemezés adott, akkor ebből úgy kaphatunk egy $S$ ütemezést az eredeti problémára, hogy az $S'$ szerinti $t'_{ij}$ hosszúságú időintervallumon belül tetszőleges módon kijelölünk egy $t_{ij}$ hosszúságú intervallumot, és az $i$-edik munkadarab $j$-edik műveletét ebben az időintervallumban végezzük el. Ez nyilván megengedett ütemezés, és $S$ átfutási ideje nem hosszabb $S'$ átfutási idejénél. A redukált és az eredeti feladat $n$, $m$, $M$ és $K$ paraméterei azonosak, tehát az átfutási időt ezekkel becsülve a kapott becslés az eredeti feladatra is érvényes marad. Ezért az 1., 4. és 5. tétel bizonyítását eleve a redukált feladatra végezzük. A műveleti időket $t_{ij}$-vel fogjuk jelölni $t'_{ij}$ helyett, és feltesszük, hogy $M_1 = M_2 = \ldots = M_m$.

*Az 1. tétel bizonyítása.* Az $i$-edik munkához rendeljük hozzá a

$$v_i = \begin{bmatrix} t_{i1} - t_{i2} \\ t_{i2} - t_{i3} \\ \vdots \\ t_{im-1} - t_{im} \end{bmatrix}$$

$(m - 1)$-dimenziós vektort. Világos, hogy $\| v_i \| = \| v_i \|_{\max} \leq K$ és hogy $\sum_{i=1}^{n} v_i = 0$. Tehát a $v_1, \ldots, v_n$ vektorhalmazra alkalmazható a 2. tétel $d = m - 1$-gyel. Az áttekinthető jelölésrendszer kedvéért tegyük föl, hogy a 2. tétel algoritmusa éppen a $v_1, v_2, \ldots, v_n$ sorrendet adja, tehát

$$\max_{1 \leq k \leq n} \| \sum_{1}^{k} v_i \| \leq (m - 1)K.$$

Másképp fogalmazva ez azt jelenti, hogy minden $j = 1, \ldots, m - 1$ és minden $k = 1, 2, \ldots, n$ esetén

$$-(m - 1)K \leq \sum_{i=1}^{k} t_{ij} - \sum_{i=1}^{k} t_{ij+1} \leq (m - 1)K.$$

Definiáljuk most a $\tau_{ij}$ kezdési időket:

$$\begin{aligned}
\tau_{11} &= 0, & \tau_{i1} &= \tau_{i-1\,1} + t_{i-1\,1}, \\
\tau_{12} &= mK, & \tau_{i2} &= \tau_{i-1\,2} + t_{i-1\,2}, \\
&\vdots \\
\tau_{1i} &= (j - 1)mK, & \tau_{ij} &= \tau_{i-1\,j} + t_{i-1\,j}, \\
&\vdots \\
\tau_{1m} &= (m - 1)mK, & \tau_{im} &= \tau_{i-1\,m} + t_{i-1\,m}.
\end{aligned}$$

Világos, hogy tetszőleges rögzített $j$ esetén az $I_{ij} = (\tau_{ij}, \tau_{ij} + t_{ij})$ intervallumok páronként diszjunktak. Megmutatjuk, hogy ez érvényes tetszőleges rögzített $i$ esetén is. Tegyük fel tehát, hogy $j < j'$, igazolni akarjuk, hogy $I_{ij}$ és $I_{ij'}$ diszjunkt. Ehhez elég megmutatni, hogy $\tau_{ij} + t_{ij} \leq \tau_{ij'}$, ezt viszont elég $j' = j + 1$ esetén igazolni. De akkor

$$\tau_{i\,j+1} - (\tau_{ij} + t_{ij}) = j \cdot mK + \sum_{p=1}^{i-1} t_{p\,j+1} - \left[(j-1)mK + \sum_{p=1}^{i-1} t_{pj} + t_{ij}\right] =$$

$$= mK + \left(\sum_{p=1}^{i-1} t_{p\,j+1} - \sum_{p=1}^{i-1} t_{pj}\right) - t_{ij} \geq mK - (m-1)K - K = 0\,.$$

A $\tau_{ij}$-kkel definiált ütemezés tehát megengedett. Egyszerű számolás mutatja, hogy a befejezési idő éppen $M + m(m-1)K$, tehát $\tau(\pi) \leq M + m(m-1)K$.

Az algoritmus lépésszáma a 2. tételből adódik.

*A 4. tétel bizonyítása.* Olyan $Q$ ütemezést konstruálunk, melynél $T_Q = M$, tehát minden gép megállás nélkül dolgozik és $M$ idő elteltével egyszerre fejezik be a munkát.

Az 1. Tétel bizonyításához hasonlóan alkalmazzuk a

$$v_i = \begin{bmatrix} t_{i1} - t_{i2} \\ \vdots \\ t_{im-1} - t_{im} \\ t_{im} - t_{i1} \end{bmatrix} \in R^m$$

$(i = 1, \ldots, n)$ vektorokra a 2. tétel algoritmusát! A kapott új sorrendet $1, 2, \ldots, n$-nel jelölve $1 \leq k \leq m$ és $j = 1, \ldots, m-1$, illetve $j = m$ esetén

$$-mK \leq \sum_{i=1}^{k} t_{ij} - \sum_{i=1}^{k} t_{ij+1} \leq mK\,,$$

illetve

$$-mK \leq \sum_{i=1}^{k} t_{im} - \sum_{i=1}^{k} t_{i1} \leq mK.$$

A munkadarabok sorrendje az egyes gépeken ilyen lesz:

| | | | | |
|---|---|---|---|---|
| az első gépen: | $1, 2, 3, \ldots$ | $\ldots$ | $\ldots\, n-1, n$ | |
| a másodikon: | $i_2 + 1, i_2 + 2, \ldots$ | $n, 1, \ldots$ | $\ldots$ | $, i_2$ |
| a harmadikon: | $i_3 + 1, i_3 + 2, \ldots$ | $\ldots$ | $n, 1, \ldots$ | $, i_3$ |
| $\vdots$ | | | | |
| az utolsón: | $i_m + 1, i_m + 2, \ldots$ | $\ldots$ | $n, 1, \ldots$ | $, i_m$. |

Rekurzió segítségével megadjuk az $i_2, i_3, \ldots, i_m$ indexeket. Ez rögtön meg fogja adni az első munkadarab kezdési idejét, az $s$-edik gépen, $\tau_s$-et. Ugyanis nyilván ($s \geq 2$-re)

$$\tau_s = \tau_{1s} = \sum_{k=i_s+1}^{n} t_{ks}, \tag{4}$$

a további kezdési idők pedig

$$\tau_{is} = \begin{cases} \tau_s + \sum\limits_{k=1}^{i-1} t_{ks}, & \text{ha } i \leq i_s, \\ \sum\limits_{k=i_s+1}^{i} t_{sk}, & \text{ha } i > i_s. \end{cases}$$

Legyen tehát $i_2$ az a legnagyobb index, melyre

$$\sum_{i=i_2+1}^{n} t_{i2} \geq (m+1)K.$$

Ha már $i_s$ $(s \geq 2)$ meg van határozva, akkor (4) alapján $\tau_s$ is, és legyen $i_{s+1}$ az a legnagyobb index, melyre

$$\sum_{i=i_{s+1}+1}^{n} t_{i\, s+1} \geq \tau_s + (m+1)K.$$

Könnyen látható, hogy $\tau_s \leq \tau_{s-1} + (m+2)K$ $(s = 2, 3, \ldots, m$ esetén), tehát ha $M \geq (m^2 + 2m - 1)K$, akkor $i_s$ és $\tau_s$ jól van definiálva $s = 2, 3, \ldots, m$-re, sőt

$$M \geq \tau_m + (m+1)K.$$

Ezzel $s = 2, 3, \ldots, m$-re definiálva van $i_s$, tehát az egész ütemezés is.

A $\tau_2, \ldots, \tau_m$ „késleltetések'' azt biztosítják, hogy egyik munkadarab se kerüljön egy időben két gépen megmunkálásra. Valóban, legyen $i$ egy tetszőleges munkadarab indexe. Jelöljük $s$-sel azt a legnagyobb indexet, melyre $\tau_k + \sum\limits_{s=1}^{k} t_{sk} \leq M$. (Ez azt jelenti, hogy $j = 2, \ldots, k$-ra a $j$-edik gépen az $i$-edik munkadarab megmunkálása a $\tau_j$ időpont után kezdődik, $j = k+1, \ldots, n$-re pedig $\tau_j$ előtt.) Azt fogjuk belátni, hogy ekkor

$$\begin{cases} \tau_{i2} \geq \tau_{i1} + t_{i1}, \\ \vdots \\ \tau_{ik} \geq \tau_{i\,k-1} + t_{i\,k-1} \end{cases} \tag{5}$$

és $k < m$ esetén

$$\tau_{i1} \geq \tau_{im} + t_{im}, \tag{6}$$

$$\begin{cases} \tau_{im} \geq \tau_{i\,m-1} + t_{i\,m-1}, \\ \vdots \\ \tau_{i\,k+2} \geq \tau_{i\,k+1} + t_{i\,k+1}. \end{cases} \tag{7}$$

(5) igazolása egyszerű: $2 \leq j \leq k$ esetén

$$\tau_{ij} - \tau_{ij-1} = \tau_j - \tau_{j-1} + \sum_{l=1}^{i-1} t_j - \sum_{l=1}^{i-1} t_{j-1} \geq (m+1)K - mK = K \geq t_{ij-1}.$$

Ugyanígy adódik (7) is. Végül (6) is teljesül, mert

$$\tau_{im} = \sum_{l=i_m+1}^{i-1} t_{lm} = \tau_m - M + \sum_{l=1}^{i-1} t_{lm} \leq$$

$$\leq \tau_m - M + \sum_{l=1}^{i-1} t_{l1} + mK \geq \sum_{l=1}^{i-1} t_{l1} - K = \tau_{il} - K.$$

Beláttuk tehát, hogy a megadott ütemezés ütközésmentes és így optimális. Az algoritmus műveletigénye automatikusan adódik a 2. tételből.

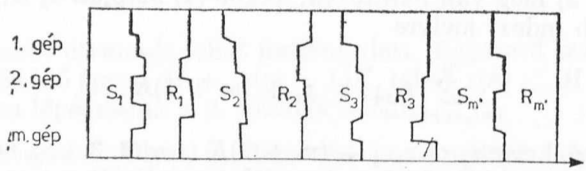
*Az 5. tétel bizonyítása.* Feltesszük, hogy a gépek számára

$$2^{k-1} < m \leq 2^k$$

teljesül ($k$ természetes szám).

A továbbiakban $2^k$-t jelöljük $m'$-vel. Megjegyezzük, hogy ez az $m'$ az eredeti $m$-nek kevesebb, mint kétszerese.

Az optimális ütemezéshez tartozó idődiagram jellegét a 2. ábra mutatja. Bevezetve az $S_1 \cup S_2 \cup \ldots \cup S_{m'} = S$ és $R_1 \cup R_2 \cup \ldots \cup R_{m'} = R$ jelöléseket, az ábrán látható $2m'$ blokkról a következőket tesszük fel:




*2. ábra*

(a) $S \cup R$ az összes elvégzendő művelet halmazának egy partíciója, azzal a kiegészítéssel, hogy egy munkadarabnak vagy minden művelete $S$-ben vagy minden művelete $R$-ben van.

(b) Ha egy munkadarabnak minden művelete $S$-ben van, akkor minden $i$-re $S_i$-ben legfeljebb egy művelete van ($i = 1, 2, \ldots, m'$). Ha egy munkadarab minden művelete $R$-ben van, akkor mindegyik $R_i$-hez legfeljebb egy művelete tartozik.

(a) miatt egy $S_i$-beli és egy $R_j$-beli művelet különböző munkadarabhoz tartozik, tehát szomszédos blokkok közötti ütközés nem fordulhat elő.

Ha el tudjuk érni, hogy $S_j/R_j$ valamennyi művelete hamarabb fejeződjék be, mint $S_{j+1}/R_{j+1}$ bármelyik művelete elkezdődik ($j = 1, 2, \ldots, m' - 1$), akkor az ütemezés ütközésmentes, és (mivel minden gép megállás nélkül dolgozik) optimális lesz.

Ezt a célt úgy fogjuk elérni, hogy a műveletek halmazát (gépenként) egyenletes rétegekre szedjük szét. $M$-et pedig olyan nagyra választjuk, hogy egy réteg (blokk) vastagsága révén képes legyen a két szomszédos réteget (időben) elszigetelni.

A konstrukció a következő lépésekből áll:

1. A $v_i = (t_{i1}, t_{i2}, \ldots, t_{im})$ vektorokra alkalmazva a 2. lemma algoritmusát, elkészítjük a munkadarabok halmazának egy $\{1, 2, \ldots, n\} = \bar{S} \cup \bar{R}$ partícióját, úgy, hogy minden $j$-re ($j = 1, 2, \ldots, m$)

$$\left| \sum_{i \in \bar{S}} t_{ij} - \frac{M}{2} \right| \leq \frac{m}{2} K$$

és

$$\left| \sum_{i \in \bar{R}} t_{ij} - \frac{M}{2} \right| \leq \frac{m}{2} K$$

teljesüljön. (Itt kihasználtuk, hogy $\sum_{i=1}^{n} t_{ij} = M$.)

$S$-sel jelöljük az $\bar{S}$-beli munkadarabok műveleteinek halmazát, $R$-rel pedig az $\bar{R}$-beli munkadarabok műveleteinek halmazát.

2. Definiálunk egy $G_S = (V, E)$, $U = A \cup B$, $E = A \times B$ páros gráfot, ahol az $A$ halmaz $|\bar{S}|$ elemű, pontjai az $\bar{S}$-beli munkadaraboknak felelnek meg; a $B$ halmaz $m$ elemű, pontjai a gépeknek felelnek meg. Az $i$-edik munkadarabnak megfelelő pontból a $j$-edik gépnek megfelelő pontba vezető élen legyen a $\lambda$ súlyfüggvény értékre $t_{ij}$. Analóg módon bevezetjük a $G_R$ páros gráfot, ahol az $A$ halmaz $|\bar{R}|$ elemű, pontjai az $\bar{R}$-beli munkadaraboknak felelnek meg, a konstrukció többi része pedig változatlan.

Ezekre a gráfokra alkalmazva a 3. lemma algoritmusát az $S$, illetve $R$ halmazt egyaránt $m'$ részre szedjük, azaz

$$S = S_1 \cup S_2 \cup \ldots \cup S_{m'}, \quad \text{és}$$
$$R = R_1 \cup R_2 \cup \ldots \cup R_{m'},$$

partíciókat készítünk úgy, hogy

(A) Egy $\bar{S}$-beli munkadarab minden művelete különböző $S_j$-be tartozzon. $\bar{R}$-beli munkadarab minden művelete másik $R_j$-be tartozzék.

(B) $j = 1, 2, \ldots, m$ és $l = 1, 2, \ldots, m'$ esetén

$$\sum_{\substack{i \\ (i,j)\in S_l}} t_{ij} \geq \frac{M}{2m'} - \frac{5}{2} K$$

$$\sum_{\substack{i \\ (i,j)\in R_l}} t_{ij} \geq \frac{M}{2m'} - \frac{5}{2} K$$

(C) $j = 1, 2, \ldots, m$ és $l = 1, 2, \ldots, m'$ esetén

$$\left| \sum_{\substack{i \\ (i,j)\in S_1\cup S_2\cup\ldots\cup S_l}} t_{ij} - \frac{l}{m'} \sum_{i\in\bar{S}} t_{ij} \right| \leq K \cdot \log_2 m'$$

$$\left| \sum_{\substack{i \\ (i,j)\in R_1\cup R_2\cup\ldots\cup R_l}} t_{ij} - \frac{l}{m'} \sum_{i\in\bar{R}} t_{ij} \right| \leq K \cdot \log_2 m'.$$

A (C)-beli egyenlőtlenségek összeadásával

$$\left| \sum_{\substack{i \\ (i,j)\in S_1\cup R_1\cup\ldots\cup S_l\cup R_l}} t_{ij} - \frac{l}{m} \cdot M \right| \leq 2K \cdot \log_2 m'$$

adódik. Eszerint az ábrán szaggatott vonallal jelölt választóvonalak ingadozásának mértéke legfeljebb $4K \log_2 m'$. (B) miatt mindegyik réteg vastagsága legalább $\frac{M}{2m'} - \frac{5}{2} K$. Emiatt az ütközések elkerülésének elégséges feltétele az $\frac{M}{2m'} - \frac{5}{2} K \geq 4K \log_2 m'$, tehát

$$M \geq (8m' \log_2 m' + 5m')K$$

egyenlőtlenség.

Figyelembe véve, hogy az itt szereplő $m'$ az $m$-nek legfeljebb 2-szerese, az eredeti $m$-mel

$$M \geq (16m \log_2 m + 21m)K$$

elegendő.

Az $\bar{S} \cup \bar{R}$ felbontás a 2. lemma szerint $O(nm^3)$ műveletet igényel. A 3. lemma algoritmusát 2-szer alkalmazzuk. Mindkét esetben $|A| = S \leq n$, tehát min $(S, m) \leq m$ miatt a további műveletek lépésszáma $O(nm^2 \log_2 n)$. Ezzel az 5. tételt teljes egészében beláttuk.

*(Beérkezett: 1982. január 13-án.)*

## IRODALOM


BÁRÁNY I.: A vector-sum theorem and its application to improve flow-shop guarantees, *Mathematics of Operations Research*, 1981.

CAMPBELL, H. G., DUDEK, R. A. and SMITH, L. M.: A heuristic algorithm for the n job, m machine sequencing problem, *Management Science* 16 (1970) 630—637.

GAREY, M. R., JOHNSON, D. S. and SETHI, R.: The complexity of flow-shop and job-shop scheduling, *Mathematics of Operations Research* 1 (1976) 117—129.

GONZALES, T., SAHNI, S.: Open shop scheduling to minimize finish time, *J. Assoc. Comp. Mach.* 23 (1976) 665—679.

GRAHAM, R. L., LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G.: Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of Discrete Math.* 5 (1979) 287—326.

FEJES K.: Flow-shop ütemezések közelítő megoldási módszereinek összehasonlítása, *Számológép* 2 (1974) 5—15.

FIALA T.: Közelítő algoritmus a három gép problémára, *Alkalmazott Matematikai Lapok* 3 (1977) 389—398.

FIALA T.: *Ütemezési algoritmusok* (doktori disszertáció) 1979.

JOHNSON, S. M.: Optimal two- and three-stage production schedules with setup times included, *Naval Res. Log. Quaterly* 1 (1954) 61—68.

PALMER, D. S.: Sequencing jobs through a multi-stage process in the minimum total time — A quick method of obtaining a near optimum, *Operational Research Quarterly* 16 (1965) 101—107.

Белов, И. С.–Столин, Я. Н.: Алгоритм в одномаршрутной задаче календарного планирования, в книге: Мат. экономика и функциональный анализ, Наука, 1974.

Гринберг, В. С.–Севастьянов С. В.: О величине константы Штейница, Функциональный анализ и его приложения 14 (1980), 56–57.

Кадец, М. И.: Об одном свойстве векторных ломаных в $n$-мерном пространстве, Успехи Мат. Наук 8 (1953), 139–143.

Севастьянов, С. В.: О приближенном решении некоторих задач теории расписаний Дискретный анализ 32 (1978), 66—75.

Танаев, В. С.–Шкурба, В. В.: Введение в теорию расписаний, Наука, 1975.


## NEARLY OPTIMUM SOLUTION OF MULTI-MACHINE SCHEDULING PROBLEMS


We deal with flow shop and open shop problems. Since in case of three or more machines both problems are NP complete, only obtaining of a nearly optimum polynomial algorithm may be hoped.

Let the number of work-pieces be $n$, that of machines $m$. In case of a flow shop problem Theorem 1 corrects a previous result of *Belov* and *Stolin*. According to this theorem by means of an algorithm of $O(n^2m^3)$ steps such a schedule may be given whose finish time $T$ yields an asymptotically optimum estimation. Proof is based on a theorem of vector rearrangement (Theorem 2) where it is stated that any closed broken line consisting of $d$-dimensional vectors with unit length at most may be rearranged in such a way that it is placed in a sphere with radius $d$. Theorem 2 is proved by the method of *Grinberg* and *Sewastianoff*.

In case of an open shop problem not only nearly optimum, but also optimum solution may be obtained. This is discussed more precisely in Theorem 3. In Theorem 4 an algorithm consisting of $O(n^2m^3)$ steps is given which yields an optimum scheduling under certain conditions. By means of a more complicated procedure a more exact estimation can also be given. According to Theorem 5 optimum scheduling may be constructed in $O(nm^3)$ steps under other conditions.

## БЛИЗКОЕ К ОПТИМАЛЬНОМУ РЕШЕНИЕ ПРОБЛЕМЫ СОСТАВЛЕНИЯ МНОГОМАШИННОГО ГРАФИКА

Статья посвящена проблемам *flow-shop* и *open-shop.* Поскольку в случае трех и больше машин эти задачи NP-полные, то можно надеяться на получение лишь приближенного оптимального полиноминального алгоритма.

Пусть количество обрабатываемых деталей равно $n$, машин $m$. В случае задачи *flow-shop* теорема 1 улучшает более ранний результат Белова и Столина. В соответствии с этим, с помощью алгоритма с последовательностью действий $O(n^2\, m^2)$ можно дать такой график, который дает асимптотически оптимальную оценку на время прохождения T. Доказательство основано на формуле векторного преобразования (теорема 2), которая говорит, что замкнутая ломаная линия, состоящая из d-мерных векторов произвольной, но не больших одной единицы длины, может быть преобразована так, чтобы помещалась в шаре радиусом $d$. Доказательство теоремы 2 ведется методом Гринберга и Севастьянова.

В случае задачи *open-shop* можно получить не только близкое к оптимальному, но и оптимальное решение. Это решение дает теорема 3.

В теореме 4 мы имеем алгоритм с такой последовательностью действий, который при определенных условиях может дать оптимальный график. С помощью более сложного метода может быть дана еще более точная оценка. В соответствии с теоремой 5 при других условиях при последовательности действий $O(nm^3)$ может быть составлен оптимальный график.