Szegedi Tudományegyetem

Informatikai Tanszékcsoport

On-line algoritmusok gépköltséges MSR modellben

Diplomadolgozat

Készítette:

Nagy-György Judit

programtervező matematikus szakos hallgató

Témavezető:

dr. Imreh Csanád

egyetemi adjunktus

Szeged

2005

Tartalomjegyzék

Feladatkiírás

Feladat: A gépvásárlásos és visszautasításos ütemezési feladat matematikai modelljének kidolgozása, megoldó algoritmusok fejlesztése, elemzése tesztelése

Részletesen:

- A modell definiálása, a kapcsolódó modellek (gépvásárlásos és visszautasításos) irodalmának áttekintése
- A feladat megoldására on-line (esetleg approximációs) algoritmusok kifejlesztése
- Az algoritmusok elemzése versenyképességi elemzés alapján
- Algoritmusok implementálása összehasonlítása véletlenül generált inputon

Tartalmi összefoglaló

A dolgozat célja speciális on-line ütemezési feladat megoldása, amelynek jellemzője, hogy gép vásárolható, illetve munka eldobható bizonyos költség ellenében. Az első fejezet a modell alapjául szolgáló szakirodalmat tárgyalja.

Ez a modell valójában két másik, az MSR és a gépköltséges modell ötvözete. A dolgozat első eredménye egy 2-es alsó korlát igazolása a modellbeli on-line algoritmusok versenyképességi hányadosára, majd következik annak igazolása a versenyképesség-elemzés módszer segítségével, hogy a két említett modellben különkülön hatékony algoritmusok kombinációinak az egyesített modellben nincs véges versenyképességi hányadosuk. Ezek után néhány egyszerű on-line algoritmus, és egy versenyképesség szempontjából hatékony megoldás, az optimumlopó algoritmus vizsgálata következik, amely során bizonyítást nyer, hogy versenyképességi hányadosa $1 + \varphi$, ahol φ az aranymetszés aránya.

A dolgozat vizsgálja a probléma off-line változatát is, ebben a modellben ad egy 2-közelítő és egy ε-közelítő polinom időben futó algoritmust.

A befejező részben a dolgozatban ismertetett, versenylépesség szempontjából nem hatékony on-line algoritmusok költségét veti össze statisztikailag a 2-közelítő algoritmus által generált költséggel különböző eloszlású inputadatokon.

Kulcsszavak: on-line ütemezés, versenyképességi hányados, alsó korlát, gépköltség, büntetés, off-line ütemezés, approximációs séma

Bevezetés

Ha be szeretnénk határolni az ütemezési feladatok helyét a tudományterületek között, akkor a kombinatorikus optimalizálási feladatok közé kell sorolnunk. Ez a terület (amelyről ld. részletesebben [4]) véges halmazon értelmezett függvény optimumának meghatározásával foglalkozik (de mint a későbbiekben látni fogjuk, előfordulhatnak olyan feltételek, amelyek mellett ez nem lehetséges). Az optimum meghatározásakor az is szempont, hogy ne tartson túl sokáig. Ha túl sokáig tartana, akkor feladat lehet annak gyors közelítése (általában a véges halmaz mint input méretének függvényében polinom sok elemi számítási lépés számít hatékony megoldásnak – feladatok számítási bonyolultságáról ld. [6]), vagy heurisztikus algoritmus keresése.

Ütemezési feladaton olyan feladat értendő, ahol (nem feltétlenül) adott gépekre adott munkákat kell beütemezni, azaz hozzárendelni egy kezdési és befejezési idővel, amelyeket alapvetően futási idő jellemez, amely meghatározza, mennyi ideig tart elvégezni a munkát. (különböző modellekben emellett más jellemzői is lehetnek). Egy ütemezésben általában az első munka kezdési ideje 0, a többié az előzőleg ugyanarra a gépre ütemezett feladat befejezési ideje, befejezési ideje pedig kezdési idejénél futási idejével több. A feladatnak rengeteg variációja létezik különböző megszorítások illetve engedmények bevezetésével.

Egy ütemezést megvalósító algoritmushoz valamilyen módon adott annak költsége adott munkasorozat esetén, a feladat ennek optimalizálása. A modell *off-line*, ha előre ismertek a munkák, *on-line* abban az esetben, ha egyesével jelennek meg, egy munka ütemezésekor a hátralévő munkákról semmilyen információ nincs, és az ütemezés utólag nem módosítható. (Az on-line algoritmusokat részletesen tárgyalja [2], on-line ütemezésekről pedig ld. [7].) Gyakran a költség a legtovább futó gép futási ideje (azaz a rá ütemezett munkák futási idejének összege), és az a cél, hogy ez a lehető legkisebb legyen. Ezt a költséget a szakirodalom *makepsan*-nek nevezi.

Egy ütemezési feladat off-line verziójában az optimális ütemezés megoldható, on-line esetben nem feltétlenül. Az on-line algoritmus *versenyképességi hányado-*

sa az a legkisebb c, amelyre teljesül, hogy tetszőleges input esetén az algoritmus költsége az optimális off-line költségnek legfeljebb c-szerese.

Előfordul, hogy off-line verzióban az optimális költség kiszámítása NP-nehéz feladat, ekkor érdekes lehet *approximációs sémák* vizsgálata. Approximációs algoritmustól megkövetelt, hogy polinom időben fusson (feltéve persze, hogy P $\neq$ NP) és az optimum tetszőleges közelítését eredményezze. Approximációs algoritmus is jellemezhető költsége és az optimális költség hányadosának infimumával (a versenyképességi hányadoshoz hasonlóan).

A következő fejezetben két on-line ütemezési modell összefoglalása következik, amelyek kombinációjának elemzése a dolgozat új eredménye. Csak azok a bizonyítások kerülnek itt elő, amelykre a későbbiekben szükség lesz. Az új modellben bizonyítást nyer a versenyképességi hányadosokra vonatkozó 2-es alsó korlát, és a dolgozat ismertet egy $1 + \varphi$ versenyképességi hányadosú algoritmust. Utána a modell NP-teljességének bizonyítása és approximációs séma vizsgálata következik, legvégül az on-line algoritmusok gyakorlatban történő összehasonlítása kerül sorra. (Az összehasonlításhoz használt program, input adathalmaz, output adatok és a statisztikai elemzés eredménye a mellékelt cd-n megtalálható.)

1. fejezet

Elméleti alapok

A dolgozatban használatos fontosabb jelöléseket foglalja össze az 1.1. táblázat.

1.1. Az alapprobléma

Az ütemezési feladatok alapvető változata (amelyet [4] tárgyal) az, amelyben adott fix m számú gép, a J-beli munkák egyetlen jellemzője a p_j futási idő. Cél olyan ütemezés, amelyre a makespan minimális. Ha $m \geq 2$, akkor a feladat NP-teljes.

jelölés	képlet	magyarázat
m		gépek száma (ha adott)
J	$\{1, \ldots, n\}$	ütemezendő munkák halmaza (rendezett halmaz)
J_j	$\{1, \ldots, j\}$	első j db munka
p_j		j munka futási ideje
p_H	$\sum_{j \in H} p_j$	H-beli munkák futási időinek összege
l_H	$\max_{j \in H} p_j$	leghosszabb H-beli munka
$C^A(H)$		Az A algoritmussal ütemezett H halmazra számított makespan
w_j		j munka további jellemzője (egyes modellekben)
w_H	$\sum_{j \in H} w_j$	H-beli munkák w_j jellemzőinek összege
$Z^{OPT}(J)$		optimális költség J inputon
$Z^A(J)$		A algoritmus költsége J inputon
φ	$\frac{1+\sqrt{5}}{2}$	az aranymetszés aránya

1.1. táblázat. Jelölések

R. L. Grahamtől származik az egyik legismertebb algoritmus, amely valójában on-line, és amelyet a következőkben ismertetett algoritmusok is alkalmaznak (a feladat on-line változatában a munkákat megjelenésükkor ütemezni kell valamelyik gépre, amikor a hátralévő munkákról semmilyen információ nincs, az ütemezés utólag nem módosítható).

> **Lista algoritmus**.
> *j-edik lépés:* j-t ütemezzük arra a gépre, amelyre az eddig legkevesebbet futott, vagyis arra, amelyre eddig ütemezett munkák futási idejének összege minimális (ha több ilyen van, válasszuk pl. a legkisebb sorszámút).

Az alapfeladat esetén igaz a következő tétel:

1.1. Tétel. *A lista algoritmus versenyképességi hányadosa* $2 - 1/m$.

Bizonyítás. Legyen ℓ az a munka, amely futása a legkésőbb fejeződik be a Lista algoritmus szerinti ütemezésben. A lista algoritmus szabályai miatt ℓ kezdési idejére teljesül, hogy

$$C^{Lista}(J) - p_\ell \leq \frac{1}{n} \sum_{j \in J \setminus \ell} p_j, \tag{1.1}$$

vagyis legfeljebb az ℓ-en kívüli munkák futási idejének átlaga, amiből következik, hogy

$$C^{Lista}(J) \;=\; Z^{Lista}(J) \;\leq\; \frac{1}{m} \sum_{j \in J} p_j + \frac{m-1}{m} p_\ell, \tag{1.2}$$

másrészt mivel minden munkát be kell ütemezni

$$Z^{OPT}(J) \;\geq\; \frac{1}{m} \sum_{j \in J} p_j \tag{1.3}$$

így adódik, hogy

$$Z^{Lista}(J) \;\leq\; \left(1 + \frac{m-1}{m}\right) Z^{OPT}(J) \;=\; \left(2 - \frac{1}{m}\right) Z^{OPT}(J). \tag{1.4}$$

□

A probléma off-line változatára Hochbaum és Shmoys [3]-ban leírtak egy ε-közelítő polinomiális approximációs sémát (PAS), visszavezetve a feladatot a bin-packing probléma megoldását közelítő ε-duális approximációs algoritmusra.

1.2. MSR

1.2.1. On-line modell

Az MSR (*Multiprocessor Scheduling with Rejection*) Bartal és mtsai által vizsgált on-line ütemezési modell (ld. [1]), amelyben lehetőség van a munkák elvetésére is. Adott m gép és n munka. A j munkához tartozik (p_j, w_j) pár, ahol p_j a munka elvégzéséhez szükséges futási idő, w_j pedig a j munka elvetésekor fizetendő büntetés. Minden egyes munka beütemezéséről illetve elvetéséről azonnal dönteni kell, ütemezés melletti döntés esetén be is kell rögtön ütemezni valamelyik gépre. A döntés és az ütemezés utólag már nem módosítható. Feladat: olyan ütemezés, ahol a makespan és a fizetett összbüntetés összege a lehető legkisebb.

A későbbiekben szükség lesz a munkák egy speciális részhalmazának definiálására: $B_m = \{j \mid w_j \leq p_j/m\}$. A szerzők tetszőleges α számhoz definiálták a következő algoritmust:

> $\mathbf{RTP}(\alpha)$ (*Reject-Total-Penalty*) **algoritmus**.
> *j-edik lépés:*
>
> (i) Ha $j \in B_m$, vessük el.
>
> (ii) Legyen W_{j-1} az eddigi nem B_m-beli elvetett munkák összbüntetése. Ha az érkező munka, $j \notin B_m$, akkor ha $W_{j-1} + w_j \leq \alpha p_j$, akkor vessük el, különben ütemezzük az eddig legkevesebbet futott gépre.

Cikkükben bizonyították a következőket:

1.2. Tétel. *Az $\mathbf{RTP}(\alpha)$ algoritmus versenyképességi hányadosa legfeljebb c, ha c-re teljesülnek a következő egyenlőtlenségek:*

$$c \geq 1 + (1 - \frac{1}{m})\frac{1}{\alpha} \tag{1.5}$$

$$c \geq 2 + \alpha - \frac{2}{m} \tag{1.6}$$

1.3. Következmény. *Az $\mathbf{RTP}(\varphi - 1)$ algoritmus versenyképességi hányadosa legfeljebb $(1 + \varphi)$.*

1.4. Tétel. *Nem létezik olyan online algoritmus az MSR modellben, amely versenyképességi hányadosa minden m esetén kevesebb $1 + \varphi$ -nél.*

Bizonyítás. Tegyük fel, hogy van olyan algoritmus, amely versenyképességi hányadosa, β kevesebb $1+\varphi$ -nél. Legyen m elég nagy kettő-hatvány, $a_i = (\log_2 m)^{i+1}$, k pedig a legnagyobb egész, amelyre teljesül, hogy $\log_2 m + \sum_{i=0}^{k} a_i < m$ (ezt kiszámolva k-ra $\lfloor \log_2 m / \log\log_2 m \rfloor - 1$ adódik). Tekintsük a következő, legfeljebb m munkát tartalmazó J-t: minden munka futási ideje 1, büntetésük pedig sorrendben

$$\begin{array}{rll} 1 & \text{munkáé} & 1/(1+\varphi) \\ 1 & \text{munkáé} & 1/(1+\varphi)^2 \\ & \vdots & \\ 1 & \text{munkáé} & 1/(1+\varphi)^{\log_2 m} \\ a_1 & \text{munkáé} & 1/a_1 \\ & \vdots & \\ a_k & \text{munkáé} & 1/a_k. \end{array}$$

Alkalmazzuk a „kisördög-módszert”: egy ellenfél adogatja sorban a J-beli munkákat, ha az algoritmus az első $\log_2 m$ munka valamelyikét elfogadja, akkor nem ad több munkát, és ekkor a versenyképességi hányados legalább $1+\varphi$, de ez ellentmond a feltevésnek. Tehát az algoritmus el kell, hogy fogadja az első $\log_2 m$ munkát. Jelölje b_i az $1/a_i$ büntetésű munkák közül elvetettek számát. Ezekre a munkákra a büntetés b_i/a_i, az optimális költség pedig legfeljebb 1. Mivel a teljes büntetés legfeljebb β lehet, lennie kell egy $\ell \le k$ számnak, amelyre $b_\ell/a_\ell \le \beta/k \le 3/k$. Rögzítsünk egy ilyen ℓ-t. Tekintsük a munkák módosított J' sorozatát, amely most legfeljebb $2m$ (szintén 1 futási idővel):

$$\begin{array}{rll} 1 & \text{munka büntetése} & 1/(1+\varphi) \\ 1 & \text{munka büntetése} & 1/(1+\varphi)^2 \\ & \vdots & \\ 1 & \text{munka büntetése} & 1/(1+\varphi)^{\log_2 m} \\ a_1 & \text{munka büntetése} & 1/a_1 \\ & \vdots & \\ a_\ell & \text{munka büntetése} & 1/a_\ell \\ m+1-\sum_{i=1}^{\ell}(a_i - b_i) & \text{munka büntetése} & 6. \end{array}$$

A munkasorozat elején azonos módon viselkedik az algoritmus, mint az előbb, a 6 büntetésű munkákat pedig el kell fogadnia, mert a makespan legfeljebb 2 lehet.

Tehát az elvetett munkák büntetése legalább

$$\sum_{j=1}^{\log_2 m} (1+\varphi)^{-j} \;\geq\; \varphi - 1 - \frac{1}{m}. \tag{1.7}$$

Összegezve a számokat adódik, hogy pontosan $m+1$ munka lett beütemezve, tehát a makespan legalább 2, így a költség legalább $1+\varphi-1/m$. Befejezésül elegendő J'-nek egy olyan ütemezését mutatni, amely költsége $1+o(1)$. Tekintsük azt, amelyik elvet

$$\begin{aligned}
1+\log_2 m \quad & \text{munkát, melynek büntetése} \quad 1/a_1, \\
b_1 \quad & \text{munkát, melynek büntetése} \quad 1/a_2, \\
b_2 \quad & \text{munkát, melynek büntetése} \quad 1/a_3, \\
& \vdots \\
b_{\ell-2} \quad & \text{munkát, melynek büntetése} \quad 1/a_{\ell-1}, \\
b_{\ell-1}+b_\ell \quad & \text{munkát, melynek büntetése} \quad 1/a_\ell,
\end{aligned}$$

és a többi munkát optimálisan ütemezi. Definíció szerint $b_i \leq a_i \leq a_{i+1}$, másrészt ℓ választásából adódóan $b_{\ell-1}+b_\ell \leq a_{\ell-1}+3a_\ell/k \leq a_\ell$, tehát érvényes az ütemezés. Eggyel kevesebb munka lett beütemezve, mint az on-line algoritmus esetén, tehát m, tehát a makespan 1. A büntetés

$$\frac{1+\log_2 m}{a_1} + \sum_{i=1}^{\ell} \frac{b_i}{a_{i+1}} + \frac{b_\ell}{a_\ell} \;=\; \frac{1+\log_2 m}{(\log_2 m)^2} + \frac{1}{\log_2 m} \sum_{i=1}^{\ell-1} \frac{b_i}{a_i} + \frac{b_\ell}{a_\ell}. \tag{1.8}$$

A jobb oldalon álló szumma kevesebb, mint az on-line algoritmus által fizetett büntetés, ennélfogva ez $O(1/\log_2 m)$. Az utolsó tagot korlátozza ℓ választása, így az $O(1/k) \;=\; O(\log_2 \log_2 m/\log_2 m)$. Tehát a teljes költség a fenti ütemezésnél $1+O(\log_2 \log_2 m/\log_2 m) = 1+o(1)$. □

1.2.2. Off-line modell

A off-line modellben előre ismerjük az összes munkát, így nem kell a megadott sorrendben beütemezni azokat. Ennélfogva jobb költség is elérhető megfelelő ütemezéssel. Egy gép esetén a megoldás nyilvánvalóan lineáris időben megadható: pontosan azokat a munkákat kell beütemezni, amelyek futási ideje kevesebb a büntetésénél. Viszont legalább 2 gépre a probléma már NP-nehéz. Ezért a modellben a szerzők leírtak egy $(2-1/m)$-közelítő algoritmust, egy teljesen polinomiális approximációs sémát fix gépszámra és egy polinomiális approximációs sémát tetszőleges gépszám esetén.

APPROX algoritmus.

B_m legyen a korábbiakban definiált halmaz

(i) Vessük el az összes B_m-beli munkát.

(ii) Rendezzük a $(J \setminus B_m)$-beli munkákat futási idejük szerinti nemcsökkenő sorrendbe.

(iii) Legyen S_j, ahol $0 \leq j \leq |J \setminus B_m|$, az a megoldás, amely az első j db $(J \setminus B_m)$-beli munkát a lista ütemezés szerint ütemezi, a többit pedig elveti. Válasszuk ki azt az S_i-t, amely költsége a legkisebb.

Megjegyzés: A fenti heurisztikus algoritmus (ii) lépése $O(n \log m)$, vagy ha $m > n$, akkor $O(n)$, tehát az algoritmus $O(n \log n)$ időben fut.

1.5. Tétel. *Az APPROX algoritmus költsége* $Z^H(J) \leq (2 - \frac{1}{m}) Z^{OPT}(J)$.

Bizonyítás. Tegyük fel, hogy a $J \setminus B_m$-beli munkák az (i) lépésben adott sorrendben vannak. Ha az optimális ütemezés ezek mindegyikét elveti, akkor a B_m-beli munkákat is definíciójuk miatt el kell vetni. Ekkor az S_0 halmazt választó megoldás az optimális ütemezést adja.

Egyébként legyen ℓ az utolsó $J \setminus B_m$-ből elfogadott munka. Tekintsük az S_ℓ megoldást és legyen $A = \{1, \ldots, \ell\}$ az elfogadott munkák halmaza. Ekkor a lista ütemezés viselkedése miatt a makespan legfeljebb

$$C^H(A) \;\leq\; \frac{p_A}{m} + \left(1 - \frac{1}{m}\right) p_\ell \; le \; \frac{p_A}{m} + \left(1 - \frac{1}{m}\right) Z^{OPT}(J). \tag{1.9}$$

Jelölje A^{OPT} az optimális ütemezés által elfogadott munkákat, R^{OPT} pedig az elvetetteket. Mivel az algoritmus költsége legfeljebb annyi, mint az S_ℓ megoldás költsége,

$$\begin{aligned} Z^H \;&\leq\; w_{J \setminus A} + \frac{p_A}{m} + \left(1 - \frac{1}{m}\right) Z^{OPT}(J) \\ &=\; w_{A^{OPT} \cap (J \setminus A)} + w_{R^{OPT} \cap (J \setminus A)} + \frac{p_{A^{OPT} \cap A}}{m} + \\ &\quad \frac{p_{R^{OPT} \cap A}}{m} + \left(1 - \frac{1}{m}\right) Z^{OPT}(J). \end{aligned} \tag{1.10}$$

ℓ választása miatt $A^{OPT} \cap (J \setminus A) \subseteq B_m$, ezért $w_{A^{OPT} \cap (J \setminus A)} \leq p_{A^{OPT} \cap (J \setminus A)}/m$. Másrészt mivel A-ben nincs B_m-beli munka: $p_{R^{OPT} \cap A}/m \leq w_{R^{OPT} \cap A}$. Ezeket beír-

va az (1.10) egyenlőtlenségbe kapjuk, hogy

$$Z^H(J) \leq w_{R^{OPT}} + \frac{p_{A^{OPT}}}{m} + \left(1 - \frac{1}{m}\right) Z^{OPT}(J)$$

$$\leq \left(2 - \frac{1}{m}\right) Z^{OPT}(J). \tag{1.11}$$

A korlát élességét mutatja a következő munkasorozat ütemezése:
$|J| = m$, $p_1 = \ldots = p_m = 1$, $w_1 = 1 - \varepsilon$, $w_2 = \ldots = w_m = \frac{1}{m}(1 - \varepsilon)$.
Ennek optimális ütemezése minden munkát elfogad, az APPROX algoritmus pedig mindent elvet, ezért a $Z^H(J)/Z^{OPT}(J)$ hányados tetszőlegesen közel lehet $2 - \frac{1}{m}$-hez. □

1.6. Lemma. *Az MSR probléma egész futási időkkel n-ben és $(Z^{OPT}(J))^m$-ben polinom időben megoldható.*

Megjegyzés: a megoldás dinamikus programozást használ, és fix gépszám esetén működik hatékonyan.

1.7. Tétel. *Bármely $\varepsilon \geq 0$-ra létezik ε-közelítő algoritmus, amely polinom időben fut n^m-ben és $1/\varepsilon$-ban.*

Tetszőleges m-re polinomiális approximációs séma a makespan probléma megoldására a [3]-ban leírt PAS-on alapul.

Adott J, $|J| = n$, m gép, és $\varepsilon > 0$, cél ε-közelítő megoldás keresése. Felső korlátként használható az APPROX algoritmus költsége, emiatt ha valamely j munkára $p_j > Z^H(J)$, akkor el kell vetni. Az 1.5 tétel miatt $L := Z^H(J)/2$ az optimális ütemezés alsó korlátjaként használható. Legyen $S = \{j \mid p_j \in [0, \varepsilon L/3]\}$ és $D = J \setminus S$. Az $(\varepsilon L/3, Z^H(J)]$ intervallumot osszuk fel $(l_1, l_2], \ldots, (l_s, l_{s+1}]$ részintervallumokra, ahol $s \leq 18\lceil 1/\varepsilon \rceil$, minden részintervallum hossza $\varepsilon^2 L/9$, $l_{s+1} \geq Z^H(J)$. $D_i := \{j \mid p_j \in (l_i, l_{i+1}]\}$ és legyenek a munkák minden ilyen halmazban büntetés szerinti nemnövekvő sorrendben. B_m a korábban definiált halmaz. Jelölje $D(y_1, \ldots, y_s)$ D azon részhalmazát, amely a D_i rendezett halmazból az első y_i elemet tartalmazza $(i = 1, \ldots, s)$.

H(ε) **algoritmus**.

(i) Minden $D(y_1, \ldots, y_s) \subseteq D$-re ütemezzük be a $D(y_1, \ldots, y_s)$-beli munkákat $\varepsilon/3$-közelítő makespannel a PAS segítségével, minden más D-beli munkát vessünk el, majd az $S \setminus B_m$-beli munkákat pedig tetszőleges sorrenben ütemezzük be lista ütemezés szabálya szerint.

(ii) Válasszuk ki azt az ütemezést a fentiek közül, amelyre a költség minimális.

1.8. Tétel. *Minden $\varepsilon > 0$ esetén a H(ε) algoritmus futási ideje polinomiális n-ben és m-ben és teljesül*

$$\frac{Z^{H(\varepsilon)}(J)}{Z^{OPT}(J)} \leq 1+\varepsilon. \tag{1.12}$$

Bizonyítás. Két lépésből áll a bizonyítás, az elsőben történik

$$\frac{Z^{H(\varepsilon)}(A^{OPT}\cap D)}{Z^{OPT}(J)} \leq 1+\frac{\varepsilon}{3} \tag{1.13}$$

igazolása, a másodikban pedig

$$Z^{H(\varepsilon)}(D(y_1^{OPT},\ldots,y_s^{OPT})) \leq Z^{H(\varepsilon)}(A^{OPT}\cap D)+\frac{2}{3}\varepsilon L. \tag{1.14}$$

Az (1.13) és (1.14) egyenlőtlenségekből pedig már következik, hogy

$$\frac{Z^{H(\varepsilon)}(D(y_1^{OPT},\ldots,y_s^{OPT}))}{Z^{OPT}(J)} \leq 1+\varepsilon, \tag{1.15}$$

és mivel $Z^{H(\varepsilon)}(J) \leq Z^{H(\varepsilon)}(D(y_1^{OPT},\ldots,y_s^{OPT})$, következik a tétel is.

1. lépés

A) eset. A gépeknek a heurisztikus megoldásban a $Z^{H(\varepsilon)}(A^{OPT}\cap D)$-nek megfelelő befejezési idejei nem különböznek $\varepsilon L/3$-nál jobban. Ebben az esetben a makespan legfeljebb $p_{A^{OPT}\cap D}/m + p_{S\setminus B_m}/m + \varepsilon L/3$. Az elvetett munkák büntetése legfeljebb $w_{S\cap B_m} + w_{D\setminus A^{OPT}}$. Ennélfogva

$$Z^{H(\varepsilon)}(A^{OPT}\cap D) \leq \frac{p_{A^{OPT}\cap D}}{m}+\frac{p_{S\setminus B_m}}{m}+\frac{\varepsilon L}{3}+w_{S\cap B_m}+w_{D\setminus A^{OPT}}. \tag{1.16}$$

B_m definícióját használva az optimális ütemezésre kapjuk:

$$Z^{OPT}(J) \geq \frac{p_{A^{OPT}\cap D}}{m}+\frac{p_{S\setminus B_m}}{m}+w_{S\cap B_m}+w_{D\setminus A^{OPT}}. \tag{1.17}$$

Az (1.16) és (1.17) egyenlőtlenségekből következik (1.13).

B) eset. A gépek befejezési idejei jobban különböznek, mint $\varepsilon L/3$. Mivel az S-beli munkák futási ideje kevesebb, mint $\varepsilon L/3$, ezért egyetlen $S\cap b_m$-beli munka sem lesz elvetve, és az S-beli munkák ütemezése nem növeli az $A^{OPT}\cap D$-re számított makespant. Jelölje $C^{H(\varepsilon)}(A^{OPT}\cap D)$ az $\varepsilon/3$-közelítő és $C^{OPT}(A^{OPT}\cap D)$ az optimális makespant az $A^{OPT}\cap D$-beli munkákra. Ebben az esetben

$$Z^{H(\varepsilon)}(A^{OPT}\cap D) = C^{H(\varepsilon)}(A^{OPT}\cap D)+w_{D\setminus A^{OPT}} \tag{1.18}$$

és

$$Z^{OPT}(J) \;\geq\; C^{OPT}(A^{OPT} \cap D) + w_{D \setminus A^{OPT}}. \tag{1.19}$$

Továbbá mivel $\varepsilon/3$-közelítő algoritmust használtunk az $A^{OPT} \cap D$-beli munkák ütemezésére

$$C^{H(\varepsilon)}(A^{OPT} \cap D) \;\leq\; \left(1 + \frac{\varepsilon}{3}\right) C^{OPT}(A^{OPT} \cap D). \tag{1.20}$$

A fenti három egyenlőtlenségből következik (1.13).

2. lépés

Hogy belássuk (1.14)-t, adnunk kell egy korlátot az extra hibára, amely abból ered, hogy $A^{OPT} \cap D \neq D(y_1^{OPT}, \ldots, y_s^{OPT})$. Az egyes D_i halmazokban a futási idők közti különbség legfeljebb $\varepsilon^2 L/9$, és $D(y_1^{OPT}, \ldots, y_s^{OPT})$ tartalmazza a D_i-beli munkák közül a legnagyobb büntetésűeket. Az utóbbiból következik, hogy az extra hiba csak annak köszönhető, hogy az első y_i^{OPT} munkának D_i-ben hosszabb a futási ideje, mint azoknak $A^{OPT} \cap D_i$-ben. Mivel a D-beli munkák futási ideje legalább $\varepsilon L/3$, a felső korlát pedig legfeljebb $2L$, egyetlen gépre sem ütemezhető $6/\varepsilon$-nál több D-beli munka. Ennélfogva a $A^{OPT} \cap D \neq D(y_1^{OPT}, \ldots, y_s^{OPT})$-nek köszönhető extra hozzájárulás a makespanhez legfeljebb $(6/\varepsilon)(\varepsilon^2 L/9) = 2\varepsilon L/3$, amiből pedig következik (1.14).

Az algoritmus futási idejét a heurisztikus $Z^{H(\varepsilon)}(D(y_1, \ldots, y_s))$ kiszámítása minden lehetséges $y_1, \ldots, y_s$-re dominálja. Mivel $0 \leq y_i$, $i = 1 \ldots, n$, ezért legfeljebb $n^s = O(n^{18^{\lceil 1/\varepsilon^2 \rceil}})$ lehetséges halmaz van, amelyre ki kell számítani ezt az értéket.

Az ε-közelítő ütemezés kiszámítása minden ilyen halmazra a [3]-ban leírt algoritmust használva $O((n/\varepsilon)^{\lceil 9/\varepsilon^2 \rceil})$ időben történik, az S-beli halmazok ütemezése pedig $O(n^2)$ időben megy mindgyik esetben. Tehát mindent összevetve az algoritmus futási ideje $O((n^3/\varepsilon)^{\lceil 9/\varepsilon^2 \rceil})$. Ez azt mutatja, hogy az algoritmus egy polinomiális approximációs séma a problémára tetszőleges m gépszámmal. □

1.3. Gépköltséges modell

Imreh Csanád és John Noga olyan modellt írtak le, amelyben nem fix a gépek száma, hanem adott költséggel növelhető (ld. [5]). Modelljükben gépek $+1$ költségért vásárolhatóak, és cél a vásárolt gépek száma és a makespan összegének minimalizálása. A problémának *lista-modell* (ahol érkezéskor kell dönteni a munkáról – ilyen az MSR is) és *idő-modell* (ahol a munkáknak megjelenési idejük is van, nem kell rögtön ütemezni őket, de egy adott pillanatban csak egy már megjelent munka ütemezhető) verzióját is vizsgálták; a dolgozat szempontjából az előbbi érdekes. Ebben a modellben eredményeik a következők:

> $\mathcal{A}_\rho$ **algoritmus**.
> $\rho = (0 = \rho_1, \rho_2, \dots, \rho_i, \dots)$ egy adott növekvő valós számsorozat.
> Kezdetben $i = 0$ gépünk van.
> j*-edik lépés:*
>
> (i) j megjelenésekor ha szükséges, veszünk gépeket úgy, hogy a gépek száma i legyen, amelyre teljesül $\rho_i \leq p_{J_j} < \rho_{i+1}$, majd
> (ii) j-t az (egyik) eddig legkevesebbet futott gépre ütemezzük.

1.9. Lemma. *Mind a lista, mind az idő modellben az optimális off-line költség legalább* $2\sqrt{p_J}$*. Ha* $l_J \geq \sqrt{p_J}$*, akkor az optimális költség legalább* $l_J + p_J/l_J$*.*

1.10. Tétel. *A lista modellben nincs olyan on-line algoritmus, amely versenyképességi hányadosa kevesebb lenne* $\frac{4}{3}$*-nál.*

Bizonyítás. Legyen A on-line algoritmus. Tekintsünk egy nagyon hosszú munkasorozatot, amelyben minden munka futási ideje nagyob kicsi: $p_j = \varepsilon$ minden $j \in J$-re. Olyan algoritmusnak, amely sosem vásárol második gépet, nincs véges versenyképességi hányadosa. Jelöljük ℓ-lel azt a munkát, amely megjelenésekor az algoritmus a második gépet veszi. Ha $p_{J_\ell} \leq 2$,

$$\frac{Z^A(J_\ell)}{Z^{OPT}(J_\ell)} = \frac{p_{J_\ell} - \varepsilon + 2}{p_{J_\ell} + 1} \geq \frac{4 - \varepsilon}{3}. \tag{1.21}$$

Ha $p_{J_\ell} > 2$,

$$\frac{Z^A(J_\ell)}{Z^{OPT}(J_\ell)} = \frac{p_{J_\ell} - \varepsilon + 2}{p_{J_\ell}/2 + \varepsilon + 2} \geq \frac{4}{3 + \varepsilon}. \tag{1.22}$$

ε tetszőleges választásával következik az állítás . □

1.11. Tétel. *A* $\rho = (0, 4, 9, \dots, i^2, \dots)$ *sorozat esetén az* $\mathcal{A}_\rho$ *algoritmus versenyképességi hányadosa* φ*.*

Bizonyítás. Tekintsünk egy tetszőleges J munkasorozatot, és rögzítsünk egy optimális ütemezést. Legyen m az $\mathcal{A}_\rho$ által használt gépek száma, ℓ az utolsóként befejezett munka, k pedig a gépek száma közvetlenül ℓ megjelenése után.

A eset: $m = 1$

Ez csak akkor lehetséges, ha $p_J < 4$. A költség ekkor $1 + p_J$. Ha az optimális ütemezés szintén 1 gépet vesz, akkor a költségek hányadosa 1. Különben az optimális költség legalább $2 + p_J/2$, így adódik, hogy a költségek hányadosa nem lehet nagyobb $5/4$-nél.

A többi eset felhasznál néhány egyszerű egyenlőtlenséget:

$$m \;\le\; \sqrt{p_J} \;<\; m+1 \tag{1.23}$$

$$k \;\le\; \sqrt{p_{J_\ell}} \;<\; k+1 \tag{1.24}$$

$$\mathcal{A}_\rho \;\le\; m + \frac{p_{J_{\ell-1}}}{k} + p_\ell \;=\; m + \frac{p_{J_\ell}}{k} + \frac{k-1}{k} p_\ell \tag{1.25}$$

B eset: $m > k$ és $p_\ell \le \sqrt{p_J}$

Felhasználva a fenti egyenlőtlenségeket:

$$\begin{aligned}
\frac{Z^{\mathcal{A}_\rho}(J)}{Z^{OPT}(J)} &\le \frac{m + (k+1)^2/k + (k-1)\sqrt{p_J}/k}{2\sqrt{p_J}} \\
&\le \frac{m + k + 2 + 1/k + \sqrt{p_J} - \sqrt{p_J}/k}{2\sqrt{p_J}} \\
&\le \frac{3\sqrt{p_J} + (m - \sqrt{p_J})/(m-1)}{2\sqrt{p_J}} \\
&\le \frac{3}{2} \;\le\; \varphi.
\end{aligned} \tag{1.26}$$

C eset: $m = k > 1$ és $p_\ell \le \sqrt{p_J}$

$$\frac{Z^{\mathcal{A}_\rho}(J)}{Z^{OPT}(J)} \;\le\; \frac{m + p_J/m + \sqrt{p_J}}{2\sqrt{p_J}} \;\le\; \frac{3m + 3 + 1/m}{2(m+1)} \;\le\; \frac{19}{12} \;\le\; \varphi. \tag{1.27}$$

D eset: $m > k$ és $p_\ell > \sqrt{p_J}$

Felhasználva a fenti egyenlőtlenségeket:

$$\begin{aligned}\frac{Z^{\mathcal{A}_\rho}(J)}{Z^{OPT}(J)} &\leq \frac{m+(k+1)^2/k+p_\ell(k-1)/k}{p_\ell+p_J/p_\ell} \leq \frac{2\sqrt{p_J}+p_\ell}{p_\ell+p_J/p_\ell} \\ &= \frac{2+p_\ell/\sqrt{p_J}}{p_\ell/\sqrt{p_J}+\sqrt{p_J}/p_\ell} \leq \varphi. \end{aligned} \tag{1.28}$$

E eset: $m = k > 1$ és $p_\ell > \sqrt{p_J}$

$$\begin{aligned}\frac{m+p_J/m-p_\ell/m}{2\sqrt{p_J}} &\leq \frac{m+(m+1)^2/m-p_\ell/m}{2(m+1)} \\ &= 1+\frac{1-p_\ell}{2m(m+1)} \leq 1, \end{aligned} \tag{1.29}$$

mivel az egyenlőtlenség bal oldala növekvő p_J-ben. Ennélfogva

$$m+\frac{p_J}{m}-\frac{p_\ell}{m} \leq 2\sqrt{p_J} \tag{1.30}$$

és

$$\frac{Z^{\mathcal{A}_\rho}(J)}{Z^{OPT}(J)} \leq \frac{2\sqrt{p_J}+p_\ell}{p_\ell+p_J/p_\ell} = \frac{2+p_\ell/\sqrt{p_J}}{p_\ell/\sqrt{p_J}+\sqrt{p_J}/p_\ell} \leq \varphi, \tag{1.31}$$

ahol a végső egyenlőtlenség abból következik, hogy az $f(x) = (2+x)/(x+1/x)$ függvény maximuma éppen φ.

Már csak azt kell megmutatni, hogy a versenyképességi hányados nem lehet φ-nél kevesebb. Legyen $|J| = N^3+1$, $p_j = 1/N$, ha $j < N^3+1$ és $p_j = \varphi N$, ha $j = N^3+1$. $\mathcal{A}_\rho$ az első N^3 munkát úgy ütemezi, hogy N gépet vásárol, és ezek mindegyikére N^2 munkát ütemez, az utolsót pedig egy teszőleges gépre ütemezi. Tehát $Z^{\mathcal{A}_\rho}(J) = N+N+\varphi N$, és így

$$\frac{Z^{\mathcal{A}_\rho}(J)}{Z^{OPT}(J)} = \frac{(2+\varphi)N}{\varphi N+\lceil (N+\varphi)/\varphi \rceil} \to \frac{2+\varphi}{\varphi+1/\varphi} = \varphi, \text{ ha } N \to \infty. \tag{1.32}$$

□

2. fejezet

MSR gépköltséggel

2.1. On-line algoritmusok

2.1.1. A modell

A modell az előzőekben ismertetett két modell kombinációja. Adott munkák J halmaza. A $j \in J$ munkát jellemzi a (p_j, w_j) pár, ahol p_j a munka elvégzéséhez szükséges futási idő, w_j pedig a j munka elvetésekor fizetendő büntetés. Ezekre csak annyi megkötés van, hogy nemnegatív valósak (egyéb megkötésekkel a feladat variációihoz juthatunk). A J-beli munkák egyesével jelennek meg, munka megjelenésekor azonnal dönteni kell elvetés vagy beütemezés mellett, utóbbi esetén azonnal be is kell ütemezni valamelyik gépre. A döntés és az ütemezés utólag nem módosítható. Gép bármikor vásárolható +1 költségért, és utólag nem adható el. Egy algoritmus költsége adott input esetén a vásárolt gépek száma, elvetett munkák büntetése és a beütemezett munkákra számított makespan összege. Feladat ennek minimalizálása.

2.1.2. Alsó korlát

Versenyképességi hányadosokra vonatkozó alsó korlát vizsgálatakor felmerül a kérdés, hogy a kiindulási modellek korlátai alkalmazhatóak-e. Az MSR modell alsó korlátjának bizonyítása kihasználja, hogy fix számú gép van, és nem számít a költségbe, ezért nem adaptálható az új modellre. Más a helyzet a gépköltséges modell korlátjával. Ha csupa olyan munka érkezik, amely büntetése legalább eggyel nagyobb futási idejénél, akkor minden olyan ütemezésnél, amely valamely munkát elveti, található egy nem rosszabb ütemezés, amely minden munkát elfogad: az el-

fogadott munkákat az eredeti módon ütemezzük, az elvetett munkák mindegyike számára veszünk egy gépet, és arra ütemezzük. Ekkor viszont alkalmazható az 1.10 tétel. Ennél nagyobb alsó korlát is bizonyítható on-line algoritmus versenyképességi hányadosára:

2.1. Tétel. *Nem létezik olyan on-line algoritmus, amely versenyképességi hányadosa 2-nél kevesebb.*

Bizonyítás. Tegyük fel, hogy van olyan on-line A algoritmus, amely versenyképességi hányadosa c valamely $c = 2 - \varepsilon$ esetén. Az 1.10 tétel alapján ez nem lehet kisebb, mint $\frac{4}{3}$, ezért feltehetjük, hogy $0 < \varepsilon \leq \frac{2}{3}$. Legyen n olyan, amelyre teljesülnek a következők:

$$n > 2c \quad \text{és} \tag{2.1}$$

$$n > \frac{4+\varepsilon}{2\varepsilon - \varepsilon^2} \tag{2.2}$$

Legyen $|J| = n^2$, minden $j \in J$-re pedig $p_j = n^2$, $w_j = n$. Ha minden munkát külön-külön gépre beütemezünk, akkor annak költsége $n^2 + n^2$. Könnyen látható, hogy ez az optimális ütemezése J-nek. Ha olyan ütemezést veszünk, amely valamely gépre legalább két munkát ütemez, annak költsége legalább $2n^2 + 1$. Tehát az optimális ütemezés minden gépre legfeljebb egy munkát ütemez. Ha m munkát ütemez be, akkor annak költsége $m+n^2+(n^2-m)n$. Mivel az $f(x) = x+n^2+(n^2-x)n$ lineáris és csökkenő függvény, minimumhelye a $[0, n^2]$ intervallumon n^2, ezért J optimális ütemezése minden munkát elfogad és n^2 gépet vásárol, tehát J optimális ütemezésének költsége $Z^{OPT}(J) = 2n^2$.

A nem vetheti el az összes munkát, mivel az összes munka elvetésének költsége (2.1) miatt nagyobb, mintha mindent elfogadna: $n^2 n > c2n^2$, viszont $Z^{A(J)} \leq c2n^2$. Legyen k $(1 \leq k \leq n)$ az első olyan munka, amelyet A elfogad. Legyen most $J = J_k$. Az algoritmus költsége ezen az inputon $Z^{A(J)} = (k-1)n + n^2 + 1$, mivel az első $k-1$ munkát elveti, a k-adik beütemezéséhez pedig vásárol egy gépet. Mivel az algoritmus versenyképességi hányadosa c,

$$(k-1)n + n^2 + 1 \leq cZ^{OPT}(J). \tag{2.3}$$

Másrészt akár elfogadjuk az összes munkát, akár elvetjük, a költség nem lehet kisebb, mint az optimum:

$$Z^{OPT}(J) \leq k + n^2 \tag{2.4}$$

$$Z^{OPT}(J) \leq kn \tag{2.5}$$

(2.3) és (2.4) egyenlőtlenségekből következik:

$$\begin{aligned}(k-1)n+n^2+1 &\leq c(k^2+n)\\ k(n-c) &\leq n+(c-1)n^2-1,\end{aligned}$$

ebből pedig

$$k \leq \frac{n+(c-1)n^2-1}{n-c} < (c-1)\frac{n^2}{n-c}+\frac{n-1}{n-c} \leq (1-\varepsilon)\frac{n^2}{n-2}+\frac{n-1}{n-2}. \tag{2.6}$$

Másrészt (2.3) és (2.5) egyenlőtlenségekből következik:

$$\begin{aligned}(k-1)n+n^2+1 &\leq ckn\\ n^2+1-n &\leq (c-1)kn,\end{aligned}$$

amiből

$$k \geq \frac{n^2-n+1}{(c-1)n} = \frac{n-1}{c-1}+\frac{1}{n(c-1)} > \frac{n-1}{c-1} = \frac{n-1}{1-\varepsilon}. \tag{2.7}$$

Viszont k a (2.6) és (2.7) egyenlőtlenségek jobb oldalain szereplő felső és alsó korlátaira teljesül az is, hogy

$$\begin{aligned}(1-\varepsilon)\frac{n^2}{n-2}+\frac{n-1}{n-2} &< \frac{n-1}{1-\varepsilon} \qquad (2.8)\\ (1-\varepsilon)^2n^2+(1-\varepsilon)(n-1) &< (n-1)(n-2)\\ 0 &< (2\varepsilon-\varepsilon^2)n-(4+\varepsilon)n+(1+\varepsilon)\\ 0 &< n-\frac{4+\varepsilon}{2\varepsilon-\varepsilon^2}n+\frac{1+\varepsilon}{2\varepsilon-\varepsilon^2},\end{aligned}$$

ez utóbbi pedig (2.2) miatt igaz. Viszont ekkor (2.6), (2.7) és (2.8) egyenlőtlenségekből következik az ellentmondás, abból pedig az állítás. □

2.1.3. $\mathrm{RTP}(\alpha)$ és $\mathcal{A}_\rho$ algoritmusok kombinációi

Legkézenfekvőbb gondolat az MSR és a gépköltséges modellek algoritmusainak ötvözése. Ennek több módja is lehetséges. A korábbiakban leírt eredmények ismeretében azonban meglepő eredményeket kaphatunk velük kapcsolatban. Vegyük sorra:

(Mindegyik kombinált algoritmus esetén α adott szám, $\rho = (0, \rho_2, \ldots, \rho_i, \ldots)$ növekvő sorozat, $B_i = \{j \mid w_j \leq p_j/i\}$ ha $i \neq 0$ és $B_0 = \{j \mid w_j \leq p_j\}$. A_j az algoritmus által a j-edik lépésben már elfogadott munkák halmaza, R_j pedig az addig elvetetteké. Kezdetben mindig 0 gép van.)

1. kombinált algoritmus (KA1).

j-edik lépés:

(i) j megjelenésekor ha szükséges, veszünk gépeket úgy, hogy a gépek száma i legyen, amelyre teljesül $\rho_i \leq p_{A_{j-1}\cup\{j\}} < \rho_{i+1}$

(ii) ha $j \in B_i$, akkor elvetjük

(iii) ha $j \notin B_i$, de $w_{R_{j-1}\setminus B_i} + w_j \leq \alpha p_j$, akkor is elvetjük

(iv) különben j-t az (egyik) eddig legkevesebbet futott gépre ütemezzük a lista algoritmus szerint.

2.2. Állítás. *Az 1. kombinált algoritmusnak nem létezik véges versenyképességi hányadosa.*

Bizonyítás. Tegyük fel, hogy $c > 0$ valós szám az algoritmus versenyképességi hányadosa. Legyen $n > c$, $|J| = n$ és minden $j \in J$-re $P_j = \rho_{j+1}$ és $w_j = 1/n$. Ekkor az optimális ütemezés minden munkát elvet és $Z^{OPT}(J) = 1$. A kombinált algoritmus is elveti a munkákat, de előtte mindig vásárol egy-egy gépet, tehát $Z^{\mathbf{KA1}}(J) = n+2 > n > cZ^{OPT}(J)$ az n-re vonatkozó feltétel miatt. Ebből viszont következik, hogy c nem lehet az algoritmus versenyképességi hányadosa. □

2. kombinált algoritmus (KA2).

j-edik lépés:

(i) j munka megjelenésekor kiszámítjuk azt az i-t, amelyre teljesül $\rho_i \leq p_{A_{j-1}\cup\{j\}} < \rho_{i+1}$

(ii) ha $j \in B_i$, akkor elvetjük

(iii) ha $j \notin B_i$, de $w_{R_{j-1}\setminus B_i} + w_j \leq \alpha p_j$, akkor is elvetjük

(iv) különben ha szükséges, veszünk gépeket úgy, hogy a gépek száma i legyen, és j-t az (egyik) eddig legkevesebbet futott gépre ütemezzük a lista algoritmus szerint.

2.3. Állítás. *A 2. kombinált algoritmusnak nem létezik véges versenyképességi hányadosa.*

Bizonyítás. Tegyük fel, hogy $c > 0$ valós szám az algoritmus versenyképességi hányadosa. Legyen $n > 2c$ és k olyan, amelyre $\rho_2/2 \leq n/k < \rho_2$. Legyen

továbbá $|J| = kn$, minden $j \in J$-re $p_j = w_j = n/k$. Ha veszünk n gépet, és mindegyikre beütemezünk k munkát, akkor a költség $n + k(n/k) = 2n$. Ebből következik, hogy $Z^{OPT}(J) \leq 2n$. Az algoritmus viszont minden munkát elvet, tehát $Z^{\mathbf{KA2}}(J) = kn(n/k) = n^2$. Viszont $n^2 > c2n$ az n-re vonatkozó feltétel miatt, tehát az algoritmus versenyképességi hányadosa nem lehet c. □

3. kombinált algoritmus (KA3).

j-edik lépés:

(i) i a gépek aktuális száma, ha $j \in B_i$, akkor elvetjük

(ii) ha $j \notin B_i$, de $w_{R_{j-1} \setminus B_i} + w_j \leq \alpha p_j$, akkor is elvetjük

(iii) különben ha szükséges, veszünk gépeket úgy, hogy a gépek száma i legyen, amelyre $\rho_i \leq p_{A_{j-1} \cup \{j\}} < \rho_{i+1}$, és j-t az (egyik) eddig legkevesebbet futott gépre ütemezzük a lista algoritmus szerint.

2.4. Állítás. *A 3. kombinált algoritmusnak nem létezik véges versenyképességi hányadosa.*

A 2.3 állítás bizonyítása alkalmazható itt is.

2.1.4. Az optimum alsó közelítésének tulajdonságai

Ha megnézzük egy ütemezés költségét, az két részből áll: egyrészt az elvetett munkák miatt keletkezett költség (összbüntetés), másrészt az elfogadott gépek miatt keletkezett költség (ez a vásárolt gépek számából és a makespanből tevődik össze). Ez utóbbit érdemes közelebbről is megvizsgálni: legyen az elfogadott munkák halmaza A. Ha vizuálisan képzeljük el az ütemezést, akkor a j munka futási idejét egy $p_j \times 1$-es téglalap szimbolizálja (legyen az 1 hosszú oldal vízszintes). Egy gépre történő ütemezés így tulajdonképpen a téglalapok egymásra pakolása. Ezzel a reprezentációval könnyen belátható, hogy az elfogadott gépek miatt keletkezett költség egy legalább p_A területű téglalap félkerülete. Vegyük észre, hogy a téglalapra érvényesek bizonyos megkötések: hosszabbik oldala legalább l_A és legalább 1. Legyen T_A az ezen megkötéseket teljesítő p_A területű téglalapok halmaza.

Vegyük most az optimális ütemezést (OPT). Jelölje m' a vásárolt gépek számát, M a makespant, A' a beütemezett munkák halmazát, R' az elvetetteket. Mivel (a munkákat reprezentáló téglalapokat egy Mm' területű téglalapba pakoltuk be):

$$\begin{aligned} Mm' &\geq p_{A'} \\ m' &\geq \frac{p_{A'}}{M} \end{aligned}$$

ezért

$$M + m' \ \geq \ M + \frac{p_{A'}}{M}, \tag{2.9}$$

a jobb oldal viszont egy éppen $p_{A'}$ területű téglalap félkerülete, amely $T_{A'}$-beli. Vezessük be a következő jelölést:

$$M_H := \begin{cases} \max\{\sqrt{p_H}, l_H\} & \text{ha } p_H > 1 \\ 1 & \text{különben} \end{cases} \tag{2.10}$$

2.5. Állítás. *A $T_{A'}$-beli téglalapok közül az $M_{A'} \times p_{A'}/M_{A'}$ oldalú minimális kerületű. Máshogy:*

$$\frac{K(t)}{2} \ \geq \ M_{A'} + \frac{p_{A'}}{M_{A'}} \quad \textit{minden } t \in T_{A'} \textit{ esetén.} \tag{2.11}$$

Bizonyítás. Vegyük az

$$f(x) \ = \ x + \frac{p_{A'}}{x} \tag{2.12}$$

függvényt a $[M_{A'}, p_{A'})$ intervallumon. (Ez egy $T_{A'}$ téglalap félkerülete, ahol a nem rövidebb oldal hossza x.) A függvény deriváltja,

$$f'(x) \ = \ 1 + \frac{-p_{A'}}{x^2} \tag{2.13}$$

az értelmezési tartományon pozitív, vagyis $f(x)$ növekvő, tehát minimumhelye $M_{A'}$. Ebből pedig következik az állítás. □

2.6. Következmény. $Z^{OPT}(J) \ \geq \ w_{R'} + M_{A'} + p_{A'}/M_{A'}$.

Bizonyítás. Gyorsan adódik a (2.9) és (2.11) egyenlőtlenségekből. □

Az eddigiekből adódik az optimumnak a későbbiekben használatos alsó közelítése:

Jelölés: legyen $Z^{OPT^*}(J) \ := \ \min_{A \subseteq J}\{w_R + M_A + p_A/M_A\}$, ahol $R = J \setminus A$, és legyen A^* az OPT*(J)-hez tartozó halmaz, ha teljesül rá, hogy felveszi rajta a minimumot (ilyen létezik, mivel J véges halmaz) és $R^* = J \setminus A^*$.

2.7. Következmény. $Z^{OPT}(J) \ \geq \ Z^{OPT^*}(J)$.

Vizsgáljuk meg OPT^* tulajdonságait!

1. tulajdonság

$$Z^{OPT^*}(J) \ \leq \ M_J + \frac{p_J}{M_J} \quad \Rightarrow \quad w_{R^*} \ \leq \ M_J + \frac{p_J}{M_J} - M_{A^*} - \frac{p_{A^*}}{M_{A^*}}$$

2. tulajdonság

$$Z^{OPT^*}(J) \;\leq\; w_J \quad \Rightarrow \quad M_{A^*} + \frac{p_{A^*}}{M_{A^*}} \;\leq\; w_{A^*}$$

3. tulajdonság Ha $j \in A^*$,

$$Z^{OPT^*}(J) \;\leq\; w_{R^*} + w_j + M_{A^*\setminus\{j\}} + \frac{p_{A^*\setminus\{j\}}}{M_{A^*\setminus\{j\}}} \quad \Rightarrow$$

$$w_j \;\geq\; M_{A^*} + \frac{p_{A^*}}{M_{A^*}} - M_{A^*\setminus\{j\}} - \frac{p_{A^*\setminus\{j\}}}{M_{A^*\setminus\{j\}}}$$

4. tulajdonság Ha $j \in R^*$,

$$Z^{OPT^*}(J) \;\leq\; w_{R^*\setminus\{j\}} + M_{A^*\cup\{j\}} + \frac{p_{A^*\cup\{j\}}}{M_{A^*\cup\{j\}}} \quad \Rightarrow$$

$$w_j \;\leq\; M_{A^*\cup\{j\}} + \frac{p_{A^*\cup\{j\}}}{M_{A^*\cup\{j\}}} - M_{A^*} + \frac{p_{A^*}}{M_{A^*}}$$

Ezen tulajdonságok sugallják a következő algoritmusokat:
(A_j az algoritmus által a j-edik lépésben már elfogadott munkák halmaza R_j pedig az addig elvetetteké. Kezdetben mindig 0 gép van.)

1. tulajdonságot kihasználó algoritmus (T1).
$\rho \;=\; (0 \;=\; \rho_1, \rho_2, \ldots, \rho_i, \ldots)$ egy adott növekvő valós számsorozat.
Kezdetben $i = 0$ gépünk van.
j-edik lépés:

(i) Ha

$$w_{R_{j-1}} + w_j \;<\; M_{J_j} - M_{A_{j-1}} + \frac{p_{J_j}}{M_{J_j}} - \frac{p_{A_{j-1}}}{M_{A_{j-1}}},$$

akkor elvetjük,

(ii) különben $A_j := A_{j-1} \cup \{j\}$, és az A_j halmazbeli munkák beütemezéséhez használjuk az $\mathcal{A}_\rho$ algoritmust.

2.8. Állítás. *Az 1. tulajdonságot kihasználó algoritmusnak nem létezik véges versenyképességi hányadosa.*

Bizonyítás. Tegyük fel, hogy $c > 0$ valós szám az algoritmus versenyképességi hányadosa. Legyen $n > 2c,\ |J| = n+1,\ p_1 = n^2,\ w_1 = 0,\ p_j = w_j = n$, ha $j > 1$. Könnyen látható, hogy az optimális ütemezés az első munkát elveti, a többit elfogadja, vásárol n gépet és mindegyikre 1 munkát ütemez, míg az algoritmus minden munkát elvet, hiszen $(j-1)n < n^2 + 2$, így az algoritmus költségére:

$$n^2 > c2n = \; Z^{OPT^*}(J),$$

ez viszont ellentmondás. □

2. tulajdonságot kihasználó algoritmus (T2).

$\rho = (0 = \rho_1, \rho_2, \ldots, \rho_i, \ldots)$ egy adott növekvő valós számsorozat. Kezdetben $i = 0$ gépünk van.

j-edik lépés:

(i) Ha

$$w_{A_{j-1}\cup\{j\}} < M_{A_{j-1}\cup\{j\}} + \frac{p_{A_{j-1}\cup\{j\}}}{M_{A_{j-1}\cup\{j\}}},$$

akkor elvetjük,

(ii) különben $A_j := A_{j-1} \cup \{j\}$, és az A_j halmazbeli munkák beütemezéséhez használjuk az $\mathcal{A}_\rho$ algoritmust.

2.9. Állítás. *A 2. tulajdonságot kihasználó algoritmusnak nem létezik véges versenyképességi hányadosa.*

Bizonyítás. Tegyük fel, hogy $c > 0$ valós szám az algoritmus versenyképességi hányadosa. Legyen $n > 2c,\ |J| = n,\ p_j = w_j = n$, minden $j \in J$-re. Könnyen látható, hogy az optimális ütemezés minden munkát elfogad, vásárol n gépet és mindegyikre 1 munkát ütemez, míg az algoritmus minden munkát elvet (hiszen $n < n + 1$), így az algoritmus költségére:

$$n^2 > c2n = Z^{OPT^*}(J),$$

ez viszont ellentmondás. □

3. és 4. tulajdonságot kihasználó algoritmus (T3).

$\rho = (0 = \rho_1, \rho_2, \ldots, \rho_i, \ldots)$ egy adott növekvő valós számsorozat. Kezdetben $i = 0$ gépünk van.

j-edik lépés:

(i) Ha

$$w_j < M_{A_{j-1}\cup\{j\}} - M_{A_{j-1}} + \frac{p_{A_{j-1}\cup\{j\}}}{M_{A_{j-1}\cup\{j\}}} - \frac{p_{A_{j-1}}}{M_{A_{j-1}}},$$

akkor elvetjük,

(ii) különben $A_j := A_{j-1} \cup \{j\}$, és az A_j halmazbeli munkák beütemezéséhez használjuk az $\mathcal{A}_\rho$ algoritmust.

2.10. Állítás. *A 3. és 4. tulajdonságot kihasználó algoritmusnak nem létezik véges versenyképességi hányadosa.*

A 2.9 állítás bizonyítása alkalmazható itt is.

2.1.5. Optimumlopó algoritmus

Úgy tűnik, az optimum alsó közelítésének vizsgálata önmagában nem vezet eredményre, ha csak teljes inputra nézzük. Érdemes megvizsgálni, hogy hogyan viselkedik az optimum alsó közelítéseinek sorozata a J_j $(j = 1, 2, \ldots, n)$ inputsorozat esetén.

2.11. Lemma. *Legyen $|J| = n$. Létezik olyan az $\{OPT^*(J_j)\}$ sorozat elemeihez tartozó $\{A_j^*\}$ sorozat, amelyre teljesül: $A_j^* \subseteq A_{j+1}^* \quad j = 1, \ldots, n-1$.*

Bizonyítás. Tegyük fel, hogy ilyen sorozat nincs. Mivel J véges halmaz, ezért véges sok halmazsorozat létezik, amely az állítás feltételét teljesíti. Legyen $\{A_j^*\}$ közülük az, amelynek leghosszabb kezdőszeletére teljesül, hogy $A_{j-1}^* \subseteq A_j^*$ ha $j = 1, \ldots,$ k-1 és $A_{k-1}^* \not\subseteq A_k^*$ $(k \geq 1)$.

1. eset: $k \in R_k$
A 2.5 állítás miatt

$$Z^{OPT^*}(J_{k-1}) \leq w_{R_k^* \setminus \{k\}} + M_{A_k^*} + \frac{p_{A_k^*}}{M_{A_k^*}} \tag{2.14}$$

$$w_{R_{k-1}^*} + w_k + M_{A_{k-1}^*} + \frac{p_{A_{k-1}^*}}{M_{A_{k-1}^*}} \leq w_{R_k^*} + M_{A_k^*} + \frac{p_{A_k^*}}{M_{A_k^*}} \tag{2.15}$$

$$w_{R_{k-1}^* \cup \{k\}} + M_{A_{k-1}^*} + \frac{p_{A_{k-1}^*}}{M_{A_{k-1}^*}} \leq Z^{OPT^*}(J_k), \tag{2.16}$$

vagyis A_{k-1}^* egy J_k-hoz tartozó halmaz. Legyen $A_k^{**} := A_{k-1}^*$ és $A_j^{**} := A_j^*$, ha $j \neq k$. Erre a sorozatra teljesül, hogy $A_{j-1}^{**} \subseteq A_j^{**}$ ha $j = 1, \ldots, k$, ami pedig ellentmondás.

2. eset: $k \in A_k$

a) $M_{A_{k-1}^*} > M_{A_k^*}$
Mivel $M_{A_k^*} + p_{A_k^* \setminus \{k\}}/M_{A_k^*}$ egy $T_{A_k^* \setminus \{k\}}$-beli téglalap félkerülete, ezért a 2.5 állítás és az *a)* feltétel alapján

$$w_{R_{k-1}^*} + M_{A_{k-1}^*} + \frac{p_{A_{k-1}^* \cup \{k\}}}{M_{A_{k-1}^*}} = Z^{OPT^*}(J_{k-1}) + \frac{p_k}{M_{A_{k-1}^*}} <$$

$$w_{R_k^*} + M_{A_k^* \setminus \{k\}} + \frac{p_{A_k^* \setminus \{k\}}}{M_{A_k^*}} + \frac{p_k}{M_{A_k^*}} = Z^{OPT^*}(J_k) \tag{2.17}$$

ami nem lehetséges.

b) $M_{A^*_{k-1}} \leq M_{A^*_k}$

$$Z^{OPT^*}(J_{k-1}) \leq w_{R^*_{k-1}} + w_{R^*_k \cap A^*_{k-1}} + M_{A^*_k \cap A^*_{k-1}} + \frac{p_{A^*_k \cap A^*_{k-1}}}{M_{A^*_k \cap A^*_{k-1}}} \leq$$

$$w_{R^*_{k-1}} + w_{R^*_k \cap A^*_{k-1}} + M_{A^*_{k-1}} + \frac{p_{A^*_k \cap A^*_{k-1}}}{M_{A^*_{k-1}}} \tag{2.18}$$

teljesül a a 2.5 állítás miatt, mivel a jobb oldalon szereplő $M_{A^*_{k-1}} + p_{A^*_k \cap A^*_{k-1}} / M_{A^*_{k-1}}$ egy $T_{A^*_k \cap A^*_{k-1}}$-beli téglalap félkerülete, hiszen

$$A^*_k \cap A^*_{k-1} \subseteq A^*_k \quad \Rightarrow \quad M_{A^*_k \cap A^*_{k-1}} \geq M_{A^*_k}. \tag{2.19}$$

A *b)* feltétel és a (2.18) alapján következik, hogy

$$\frac{p_{A^*_k \cap A^*_{k-1}}}{M_{A^*_k}} \leq \frac{p_{A^*_k \cap A^*_{k-1}}}{M_{A^*_{k-1}}} \leq w_{R^*_k \cap A^*_{k-1}} \tag{2.20}$$

másrészt

$$w_{R^*_k \cap R^*_{k-1}} + M_{A^*_k \cup A^*_{k-1}} + \frac{p_{A^*_k \cup A^*_{k-1}}}{M_{A^*_k \cup A^*_{k-1}}} \leq$$

$$w_{R^*_k \cap R^*_{k-1}} + M_{A^*_k} + \frac{p_{A^*_k \cup A^*_{k-1}}}{M_{A^*_k}} \tag{2.21}$$

mivel $M_{A^*_k} \geq l_{A^*_k}$, $M_{A^*_{k-1}} \geq l_{A^*_{k-1}}$, ezért a *b)* feltétel miatt $M_{A^*_k} \geq l_{A^*_k \cup A^*_{k-1}}$ (és legalább 1), tehát a jobb oldalon szereplő

$$M_{A^*_k} + \frac{p_{A^*_k \cup A^*_{k-1}}}{M_{A^*_k}}$$

kifejezés egy $T_{A^*_k \cup A^*_{k-1}}$-beli téglalap félkerülete. A (2.20) egyenlőtlenségből adódik, hogy

$$w_{R^*_k \cap R^*_{k-1}} + M_{A^*_k} + \frac{p_{A^*_k \cup A^*_{k-1}}}{M_{A^*_k}} =$$

$$w_{R^*_k \cap R^*_{k-1}} + M_{A^*_k} + \frac{p_{A^*_k \cap A^*_{k-1}} + p_{A^*_k \cap R^*_{k-1}} + p_{R^*_k \cap A^*_{k-1}} + p_j}{M_{A^*_k}} \leq$$

$$w_{R^*_k \cap R^*_{k-1}} + w_{R^*_k \cap A^*_{k-1}} + M_{A^*_k} + \frac{p_{A^*_k \cap A^*_{k-1}} + p_{A^*_k \cap R^*_{k-1}} + p_j}{M_{A^*_k}} =$$

$$w_{R^*_k} + M_{A^*_k} + \frac{p_{A^*_k}}{M_{A^*_k}} = Z^{OPT^*}(J_k) \tag{2.22}$$

A (2.21) és (2.22) egyenlőtlenségekből pedig következik, hogy $A^*_k \cup A^*_{k-1}$ is J_k-hoz tartozó halmaz. Legyen $A^{**}_k := A^*_k \cup A^*_{k-1}$ és $A^{**}_j := A^*_j$, ha $j \neq k$. Erre a sorozatra teljesül, hogy $A^{**}_{j-1} \subseteq A^{**}_j$ ha $j = 1, \ldots, k$, ami pedig ellentmondás. □

2.12. Lemma. *Legyen $|J| = n$, $\{A_j^*\}$ olyan sorozat, amelyre teljesül a 2.11 lemma, és $R_j^* = J_j \setminus A_j^*$. Ekkor teljesül:*

$$\sum_{j=1}^{n} w_{R_{j-1}^* \cup A_j^*} \;\leq\; M_{A_n^*} + \frac{p_{A_n^*}}{M_{A_n^*}} \tag{2.23}$$

Bizonyítás. A 2.5 állítás miatt

$$Z^{OPT^*}(J_{j-1}) \;=\; w_{R_j^* \setminus \{j\}} + w_{R_{j-1}^* \cap A_j^*} + M_{A_{j-1}^*} + \frac{p_{A_{j-1}^*}}{M_{A_{j-1}^*}} \;\leq$$

$$a_{R_j^*} + M_{A_j^* \setminus \{j\}} + \frac{p_{A_j^* \setminus \{j\}}}{M_{A_j^* \setminus \{j\}}} \quad \Rightarrow \tag{2.24}$$

$$w_{R_{j-1}^* \cap A_j^*} \;\leq\; M_{A_j^* \setminus \{j\}} + \frac{p_{A_j^* \setminus \{j\}}}{M_{A_j^* \setminus \{j\}}} - \left(M_{A_{j-1}^*} + \frac{p_{A_{j-1}^*}}{M_{A_{j-1}^*}} \right) \tag{2.25}$$

$$\sum_{j=1}^{n} w_{R_{j-1}^* \cap A_j^*} \;\leq\; \sum_{j=1}^{n} M_{A_j^* \setminus \{j\}} + \frac{p_{A_j^* \setminus \{j\}}}{M_{A_j^* \setminus \{j\}}} - \left(M_{A_{j-1}^*} + \frac{p_{A_{j-1}^*}}{M_{A_{j-1}^*}} \right) \;\leq$$

$$\sum_{j=1}^{n} M_{A_j^*} + \frac{p_{A_j^*}}{M_{A_j^*}} - \left(M_{A_{j-1}^*} + \frac{p_{A_{j-1}^*}}{M_{A_{j-1}^*}} \right) \;= \tag{2.26}$$

$$M_{A_n^*} + \frac{p_{A_n^*}}{M_{A_n^*}} \tag{2.27}$$

mivelhogy (2.26) teleszkopikus összeg. Ezzel igazoltuk az állítást. □

A fenti lemmák szerint megfelelő optimum-alsó-közelítés sorozatban a költség változását eredményező értékek összege korlátos, tehát hatékony lehet egy olyan algoritmus, amely folyamatosan figyeli egy ilyen sorozat viselkedését és lemásolja azt (természetesen az eddigi munkák ütemezésén nem változtathat, csak az új munkáról való döntés függ a sorozat viselkedésétől).

Optimumlopó algoritmus (OL).
$\rho = (0 = \rho_1, \rho_2, \ldots, \rho_i, \ldots)$ egy adott növekvő valós számsorozat, $\{A_j^*\}$ olyan halmazsorozat, amelyre igaz a 2.11 lemma, $R_j^* = J_j \setminus A_j^*$. Kezdetben $i = 0$ gépünk van.
j-edik lépés:

(i) Ha $j \in R_j^*$, akkor elvetjük akkor elvetjük,
(ii) különben $A_j := A_{j-1} \cup \{j\}$, és az A_j halmazbeli munkák beütemezéséhez használjuk az $\mathcal{A}_\rho$ algoritmust.

2.13. Tétel. *Az optimumlopó algoritmus versenyképességi hányadosa nem lehet kevesebb* $1+\varphi$*-nél.*

Bizonyítás. Mivel az algoritmus az A_n halmaz ütemezésére az $\mathcal{A}_\rho$ algoritmust használja, az 1.11 állítás alapján

$$Z^{OL(J)} \;\leq\; w_{R_n} + \varphi\left(M_{A_n} + \frac{p_{A_n}}{M_{A_n}}\right) \;le\; w_{R_n} + \varphi\left(M_{A_n^*} + \frac{p_{A_n^*}}{M_{A_n^*}}\right) \tag{2.28}$$

másrészt a 2.11 lemma teljesülése miatt

$$R_n \;=\; \bigcup_{j=1}^{n} R_j^* \;=\; \bigcup_{j=1}^{n-1} \left(R_j^* \setminus R_{j+1}^*\right) \cup R_n^* \;=\; \bigcup_{j=1}^{n-1} \left(R_j^* \cap A_{j+1}^*\right) \cup R_n^* \tag{2.29}$$

ezért a 2.12 lemma alapján

$$w_{R_n} \;=\; w_{R_n^*} + \sum_{j=2}^{n} w_{R_j^* \cap A_{j+1}^*} \;\leq\; w_{R_n^*} + M_{A_n^*} + \frac{p_{A_n^*}}{M_{A_n^*}} \tag{2.30}$$

A 2.28 és 2.30 egyenlőtlenségekből pedig adódik

$$\begin{aligned} Z^{OL(J)} \;&\leq\; w_{R_n^*} + (1+\varphi)\left(M_{A_n^*} + \frac{p_{A_n^*}}{M_{A_n^*}}\right) \;\leq \\ &(1+\varphi)\left(w_{R_n^*} + M_{A_n^*} + \frac{p_{A_n^*}}{M_{A_n^*}}\right) \;\leq \\ &(1+\varphi) Z^{OPT}(J) \end{aligned} \tag{2.31}$$

és éppen ezt kellett bizonyítani. □

Megjegyzés: Az $\{A_j^*\}$ halmazsorozat meghatározásának módjáról nincs szó a fentiekben. Mivel 2^n lehetséges halmaz van, hasznos lenne egy olyan algoritmus, amely az összes eset végigvizsgálásánál gyorsabban meghatározza a sorozatot.

2.2. Approximációs séma

A séma vizsgálatának létjogosultságát igazolja a következő állítás:

2.14. Állítás. *A gépköltséges MSR probléma off-line változata NP-nehéz.*

Bizonyítás. Ehhez elegendő visszavezetni rá az alapproblémát. Legyen adott az alapfeladatban $J = \{1, \ldots, n\}$, $j \in J$ munka futási ideje p_j, ennek optimális ütemezését keressük. Ehhez kell konstruálni egy feladatot az új modellben, amely optimális megoldásából adódik az alapfeladat megoldása. Legyen $N = \min\{n, m\}$. Legyen J' az ütemezendő munkák halmaza: $J' = \{1', \ldots, (N+n)'\}$, $p_{j'} = N$, ha $0 \leq j \leq N$ és a többi munka legyen a J-beli munkák megfelelője a következő módon: $p_{(N+j)'} = p_j/(p_J + 1)$. Másrészt legyen $w_j = p_j + 2$ minden $j \in J'$-re.

Könnyen látható, hogy J' optimális ütemezése egyetlen munkát sem vet el, az első N munkát külön-külön gépre ütemezi, az utolsó n munkát pedig ezen gépekre ütemezi. Ezt az N gépet egyesével feleltessük meg az alapproblémában adott gépeknek (vagy ha az több, akkor az első N gépnek), és az egyes J-beli munkákat ütemezzük be arra a gépre, amelynek megfelelőjére lett ütemezve a munka J'-beli párja. Ez az ütemezés szükségképp optimális, mert ha lenne jobb, akkor annak megfelelőjét használva J' ütemezésekor jobb ütemezést kapnánk. □

Indokolt tehát, hogy hatékony közelítést adjunk az optimális ütemezésre. Először egy 2-közelítő algoritmus következik, majd egy ε-közelítő polinomiális algoritmus, amely az MSR probléma H(ε) algoritmusán alapul.

2-KÖZELÍTŐ algoritmus (K(2)).

(i) Minden m-re ($m = 0, \ldots, n$) futtassuk J-re az APPROX algoritmust, és jelölje $Z^{H_m}(J)$ annak MSR modellbeli költségét.
(ii) Ezek közül válasszuk a minimális költségű ütemezést: $Z^{K(2)}(J) = \min\{Z^{H_m}(J) + m\}$.

2.15. Állítás. *A K(2) algoritmus költsége* $Z^{K(2)}(J) \leq 2Z^{OPT}(J)$ *és* $O(n^2 \log n)$ *időben fut.*

Bizonyítás. Legyen m' az optimális ütemezésben vásárolt gépek száma és $Z^{OPT}_{MSR(m)}(J)$ az optimális ütemezés az MSR modellben m gép esetén. Mivel APPROX $2 - 1/m$ közelítő m gép esetén, tudjuk, hogy teljesül rá (1.12).

$$\begin{aligned} Z^{K(2)}(J) &= \min\{Z^{H_m}(J) + m\} \leq Z^{H_{m'}}(J) + m' \\ &\leq \left(2 - \frac{1}{m'}\right) Z^{OPT}_{MSR(m')}(J) + m' \leq 2Z^{OPT}(J). \end{aligned} \tag{2.32}$$

Mivel az APPROX algoritmus futási ideje $O(n \log n)$, amelyet a K(2) algoritmus azt $n+1$-szer hajtja végre, így igaz az állítás második része is. □

ε-KÖZELÍTŐ algoritmus (K(ε)).

(i) Minden m-re ($m = 0, \ldots, n$) futtassuk J-re a H(ε) algoritmust, és jelölje $Z^{H_m(\varepsilon)}(J)$ annak MSR modellbeli költségét.

(ii) Ezek közül válasszuk a minimális költségű ütemezést: $Z^{K(\varepsilon)}(J) = \min\{Z^{H_m(\varepsilon)}(J) + m\}$.

2.16. Állítás. *Minden $\varepsilon > 0$ esetén a K(ε) algoritmus futási ideje polinomiális n-ben és teljesül*

$$\frac{Z^{K(\varepsilon)}(J)}{Z^{OPT}(J)} \leq 1 + \varepsilon. \tag{2.33}$$

Bizonyítás. Az (i) lépés $n+1$-szer futtatja a H(ε) algoritmust, amely futási ideje $O((n^3/\varepsilon)^{\lceil 9/\varepsilon^2 \rceil})$, a második lépés pedig $O(n)$ idejű, tehát összesítve az algoritmus futási ideje $O(n(n^3/\varepsilon)^{\lceil 9/\varepsilon^2 \rceil})$.

Még azt kell igazolni, hogy az algoritmus ε-közelítő. Legyen m' az optimális ütemezésben vásárolt gépek száma, és $Z^{OPT}_{MSR(m)}(J)$ az optimális ütemezés az MSR modellben m gép esetén. Mivel H(ε) ε-közelítő algoritmus,

$$\begin{aligned}
\frac{Z^{K(\varepsilon)}(J)}{Z^{OPT}(J)} &= \frac{\min\{Z^{H_m(\varepsilon)}(J) + m\}}{Z^{OPT}(J)} \\
&\leq \frac{Z^{H_{m'}(\varepsilon)}(J) + m'}{Z^{OPT}(J)} \\
&\leq \frac{(1+\varepsilon) Z^{OPT}_{MSR(m')}(J) + m'}{Z^{OPT}(J)} \\
&\leq \frac{(1+\varepsilon) Z^{OPT}(J)}{Z^{OPT}(J)} \leq 1 + \varepsilon
\end{aligned} \tag{2.34}$$

□

3. fejezet

Algoritmusok összehasonlítása

3.1. Adatok és algoritmusok

Az ismertetett on-line algoritmusok közül hatnak nem volt véges versenyképességi hányadosa, de a versenyképességi hányadoson kívül más mérőszámokat is vizsgálhatunk. A versenyképességi hányados a legrosszabb esetet nézi, de elképzelhető, hogy ez ritkán jön elő.

Érdemes ezért megvizsgálni, hogy az említett algoritmusok hogyan viselkednek különböző eloszlású inputadatokon. Ebben a fejezetben a kombinált algoritmusok (paraméterekként $\alpha = \varphi - 1$-et és $\rho = 0, 4, \ldots, i^2, \ldots$ sorozatot használva), az optimum alsó közelítésének tulajdonságait használó algoritmusok (szintén az előbbi ρ sorozattal) és a 2-KÖZELÍTŐ algoritmus kerül összehasonlításra ilyen szempontból. A többi algoritmus lassúsága miatt maradt ki. A 2-KÖZELÍTŐ referenciaként szerepel, mivel az off-line probléma NP-nehéz, és az ε-közelítő algoritmus is jóval lassabb (ε-tól függően).

Az összehasonlítás a következőképpen történt:

Öt különböző eloszlású adathalmazon futottak az algoritmusok, ezek között egy kis várható értékű normális eloszlású (E=0.5, D=0.2), egy nagy várható értékű normális eloszlású (E=10000, D=5000), egy 2 paraméterű exponenciális eloszlású, egy 0 és 1 közti (kicsi) egyenletes eloszlású és egy 0 és 10000 közti (nagy) egyenletes eloszlású volt. Az inputok hossza is véletlenszerűen változott 1 és 3000 között. Az algoritmusokat összehasonlító program (forráskód C nyelvű, fordítása gcc-vel történt, mindkettő megtalálható a mellékelt cd-n algoteszt.c illetve algoteszt.exe néven) minden egyes inputra az output táblázatba (eredmeny.txt) írta az input hosszát (inputban szereplő munkák száma) és az egyes algoritmusok költségét az inputon.

Az így kapott táblázat adatai az SPSS programmal kerültek feldolgozásra.

3.2. Eredmények

Az elemzés minden input esetén minden on-line algoritmus költsége és a 2-KÖZELÍTŐ algoritmus költsége hányadosának kiszámításával kezdődött, majd ezen hányados következő statisztikai jellemzőinek kiszámításával:

- átlag
- szórás
- minimum
- maximum

minden eloszlástípus és különböző inputméretek esetén.

Ha megvizsgáljuk, hogy mekkora az on-line algoritmusok és a 2-KÖZELÍTŐ algoritmus költségének hányadosa az egyes inputadatokon, akkor azt találjuk, hogy változatos, melyik algoritmus mikor jobb. Megvizsgálva a 3.1 táblázatot látható, hogy a KA1 és KA2 algoritmusokra vonatkozó hányadosok szinte minden inputon kicsik (sosem mennek 1.7 fölé), ugyanígy a T2 algoritmusé is: az ő hányadosa is 2 alatt marad mindig (a teljes statisztikai táblázatot ld. a függelékben). Ezzel szemben a T2 és T3 algoritmusok nagy költséget adnak a kis várható értékű eloszlások esetén. A KA3 algoritmus mindenfajta inputra nagy eredményt ad, ezek között az kis normális eloszlású adatokon még viszonylag kis hányadost találunk, míg a többi mind 10 feletti.

eloszlás típusa	KA1	KA2	KA3	T1	T2	T3
kis normál	1.10	1.10	2.52	1.08	11.76	11.77
nagy normál	1.53	1.53	12.86	1.33	1.35	1.52
exponenciális	1.13	1.13	13.66	1.09	1.14	1.18
kis egyenletes	1.12	1.12	12.18	1.09	12.18	12.18
nagy egyenletes	1.60	1.60	658.23	1.38	1.34	1.49
Total	1.29	1.29	134.58	1.19	5.83	5.90

3.1. táblázat. On-line algoritmusok költségének aránya a 2-KÖZELÍTŐ algoritmus költségéhez különböző eloszlású inputokon

Ha az input eloszlása helyett méretét (munkák száma) nézzük (3.2 táblázat), akkor azt láthatjuk, hogy a KA3 algoritmusra vonatkozó hányados itt is mindenütt

nagy. A KA1, KA2 és T1 algoritmusok hányadosa itt is egészen kicsi minden típusra, míg a T2 és T3 algoritmusok csak csak kevés munka esetén adnak kis (1.6 alatti) eredményt (a teljes statisztikai táblázat szintén megtalálható a függelékben).

munkák száma	KA1	KA2	KA3	T1	T2	T3
<=3000	1.29	1.29	176.01	1.18	6.55	6.64
<1000	1.29	1.29	40.55	1.21	4.31	4.35
<60	1.33	1.33	9.78	1.29	1.57	1.60
Total	1.29	1.29	134.58	1.19	5.83	5.90

3.2. táblázat. On-line algoritmusok költségének aránya a 2-KÖZELÍTŐ algoritmus költségéhez különböző méretű inputokon

A KA3 algoritmus sikertelenségének oka valószínűleg abban keresendő, hogy még gépvásárlás előtt számítja ki, hogy az aktuális B_i halmazban van-e a következő munka, és mivel a munkák futási ideje és büntetése független változó az adathalmazban, tehát 1/2 eséllyel nagyobb a futási idő magánál a büntetésnél, ezért nagyobb eséllyel vet el egy munkát B_i-beliként, mint a többi algoritmus.

Kitekintés

A dolgozat célja volt egy új modell vizsgálata, de mint ilyen, sok kérdést is felvet. Egy részük a dolgozat során nyilvánvalóvá vált: az on-line algoritmusok versenyképességi hányadosainak alsó korlátja és az optimumlopó algoritmus versenyképességi hányadosa mint felső korlát közti szakadék áthidalásának kérdése (ti. mi lehet a versenyképességi hányadosok éles alsó korlátja), másrészt probléma az optimum alsó közelítésének gyors kiszámítása az optimumlopó algoritmus számára. Az sem tisztázott, hogy az optimumlopó algoritmus alsó korlátja éles-e.

A témakör indukál egyéb kérdéseket is: a modellben található-e az ismertetetteknél hatékonyabb ütemezési algoritmus? További érdekes kérdés, hogy vajon az idő modellben hasonló eredményeket lehet-e elérni, és az is, hogy korlátok bevezetése a futási időre illetve a büntetésre mennyiben módosítja a modellt.

Függelék

Report							
eloszlás típusa		KA1	KA2	KA3	T1	T2	T3
kis normál	Mean	1.10	1.10	2.52	1.08	11.76	11.77
	Std. Deviation	0.06	0.06	1.35	0.05	4.79	4.79
	Minimum	1.02	1.02	1.08	1.02	2.52	2.52
	Maximum	1.33	1.33	7.25	1.29	18.80	18.80
	N	78	78	78	78	78	78
nagy normál	Mean	1.53	1.53	12.86	1.33	1.35	1.52
	Std. Deviation	0.07	0.07	12.02	0.22	0.37	0.52
	Minimum	1.21	1.21	1.46	1.03	1.02	1.03
	Maximum	1.65	1.65	49.54	1.82	2.93	3.45
	N	81	81	81	81	81	81
exponenciális	Mean	1.13	1.13	13.66	1.09	1.14	1.18
	Std. Deviation	0.04	0.04	4.33	0.07	0.17	0.16
	Minimum	1.00	1.00	2.70	1.00	1.01	1.00
	Maximum	1.23	1.23	18.90	1.47	2.12	2.18
	N	69	69	69	69	69	69
kis egyenletes	Mean	1.12	1.12	12.18	1.09	12.18	12.18
	Std. Deviation	0.06	0.06	4.26	0.05	4.26	4.26
	Minimum	1.04	1.04	1.63	1.02	1.63	1.63
	Minimum	1.28	1.28	19.11	1.31	19.11	19.11
	N	88	88	88	88	88	88
nagy egyenletes	Mean	1.60	1.60	658.23	1.38	1.34	1.49
	Std. Deviation	0.07	0.07	315.68	0.26	0.33	0.45
	Minimum	1.18	1.18	7.33	1.07	1.03	1.07
	Minimum	1.68	1.68	1141.32	1.90	2.42	2.89
	N	75	75	75	75	75	75
Total	Mean	1.29	1.29	134.58	1.19	5.83	5.90
	Std. Deviation	0.23	0.23	290.19	0.21	6.06	6.01
	Minimum	1.00	1.00	1.08	1.00	1.01	1.00
	Minimum	1.68	1.68	1141.32	1.90	19.11	19.11
	N	391	391	391	391	391	391

3.3. táblázat. On-line algoritmusok költségének aránya a 2-KÖZELÍTŐ algoritmus költségéhez különböző eloszlású inputokon

Report							
munkák száma		KA1	KA2	KA3	T1	T2	T3
<=3000	Mean	1.29	1.29	176.01	1.18	6.55	6.64
	Std. Deviation	0.24	0.24	333.57	0.21	6.77	6.70
	Minimum	1.02	1.02	1.08	1	1.01	1.06
	Minimum	1.65	1.65	1141.32	1.9	19.11	19.11
	N	273	273	273	273	273	273
<1000	Mean	1.29	1.29	40.55	1.21	4.31	4.35
	Std. Deviation	0.21	0.21	96.20	0.19	3.50	3.46
	Minimum	1.04	1.04	1.12	1.02	1.03	1.03
	Minimum	1.68	1.68	439.66	1.82	11.29	11.29
	N	111	111	111	111	111	111
<60	Mean	1.33	1.33	9.78	1.29	1.57	1.60
	Std. Deviation	0.22	0.22	9.80	0.26	0.62	0.65
	Minimum	1	1	1.63	1.08	1.02	1
	Minimum	1.62	1.62	29.46	1.85	2.52	2.52
	N	7	7	7	7	7	7
Total	Mean	1.29	1.29	134.58	1.19	5.83	5.90
	Std. Deviation	0.23	0.23	290.19	0.21	6.06	6.01
	Minimum	1	1	1.08	1	1.01	1
	Minimum	1.68	1.68	1141.32	1.9	19.11	19.11
	N	391	391	391	391	391	391

3.4. táblázat. On-line algoritmusok költségének aránya a 2-KÖZELÍTŐ algoritmus költségéhez különböző méretű inputokon

Nyilatkozat

Alulírott Nagy-György Judit, programtervező matematikus szakos hallgató, kijelentem, hogy dolgozatomat a Szegedi Tudományegyetem Informatikai Tanszékcsoport Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszékén készítettem, programtervező matematikus diploma megszerzése érdekében.

Kijelentem, hogy a dolgozat saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem könyvtárában, a kölcsönözhető könyvek között helyezik el.

Szeged, 2005. május 4. . . . . .

aláírás

Irodalomjegyzék

[1] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, L. Stougie. Multiprocessor Scheduling with Rejection. *SIAM Journal on Discrete Mathematics*, 13 (2000), 64–78.

[2] A. Borodin, R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998

[3] D. S. Hochbaum, D. B. Shmoys. Using Dual Approximation Algorithms for Scheduling Problems: Theoretical and Practical Results. *Journal of the ACM*, 34 (1987), 144–162.

[4] B. Imreh. *Kombinatorikus optimalizálás*. Novadat Bt, Győr, 1999

[5] Cs. Imreh, J. Noga. Scheduling with Machine Cost. *In Randomization Approxiamtion and Combinatorical Optimization Algorithms and Techniques*, ed. D. Hochbaum and K. Jansen. 1999, 168–176.

[6] C. H. Papadimitriou. *Számítási bonyolultság*. Novadat Bt, Győr, 1999

[7] J. Sgall. On-line scheduling. *LNCS 1442*, ed. A. Fiat, G. J. Woeginger. Springer, Berlin, 1998, 196–231