Eötvös Loránd Tudományegyetem
Informatikai Kar
Komputeralgebra Tanszék

A First Fit algoritmus abszolút hibájáról

TDK dolgozat

Témavezető:
Dr. Iványi Antal Miklós
egyetemi tanár

Készítette:
Németh Zsolt
II. évfolyam
Programtervező informatikus MSc

Budapest, 2011

Tartalomjegyzék

1. Bevezető

1.1. A ládapakolási feladat

A klasszikus egydimenziós ládapakolási feladatban [3, 4, 9, 17] adott tárgyak egy $L = (a_1, a_2, \ldots, a_n)$ sorozata ($n \geq 1, n \in \mathbb{N}$), ahol minden tárgy mérete jellemezhető egy $(0,1]$-beli valós számmal. El kell helyeznünk őket minél kevesebb számú egységnyi kapacitású ládába. A továbbiakban az egyszerűség kedvéért azonosítsunk minden tárgyat a méretével, legyen tehát $a_i \in (0,1]$ minden $i = 1, 2, \ldots, n$ indexre.

Továbbá legyen $\mathcal{D}_n$ azon L listák halmaza, melyek n elemet tartalmaznak, és legyen $\mathcal{D}$ az összes valós lista halmaza, azaz

$$\mathcal{D} = \cup_{i=1}^{\infty} \mathcal{D}_i.$$

A probléma elsősorban informatikai vonatkozású. Egyrészt azonosítható a következő feladattal: egymástól független programokat (taszkokat) kell futtatnunk, és mindegyik számítógép (processzor) csak egységnyi ideig áll rendelkezésünkre. Célunk a programok olyan szétosztása a gépek között, hogy minél kevesebb gépet használjunk. Másrészt értelmezhetjük úgy is, hogy egységnyi méretű lemezeken kell adott méretű fájlokat elhelyeznünk, cél a felhasznált lemezszám minimalizálása.

A feladat NP-teljes, visszavezethető az összegzési feladatra [3, 9]. Éppen ezért a gyakorlatban közelítő algoritmusokat alkalmazunk a megoldásra [4]. Így jelentőssé válik az ilyen algoritmusok legrosszabb eseteinek vizsgálata, elemzése [2, 10].

Egy ilyen A algoritmus bemenő adatai: a méretek $L = (a_1, a_2, \ldots, a_n)$ sorozata, és esetenként az elemek n száma. A kimenő adatok pedig a szükséges $C^A(L)$ ládaszám és a ládák $H = (b_1, b_2, \ldots, b_{C^A(L)})$ telítettségei.

A vizsgálatok során H sorozat nem játszik különösebb szerepet, így tetszőleges A algoritmus tekinthető egy minden $L \in \mathcal{D}$ listához pozitív egész számot rendelő $C^A : \mathcal{D} \to \mathbb{Z}^+$ leképezés megvalósításának.

Jelölje $*$ a tetszőleges $L \in \mathcal{D}$ listához az optimális, tehát szükséges és elégséges ládaszámot rendelő algoritmust, ez a szám ekkor nyilván $C^*(L)$. Jelöléseinkben L-et néha elhagyjuk, ha az egyértelműség nem sérül.

Ekkor tetszőleges A algoritmus hibájának jellemzésére három mennyiséget definiálunk.

1.1. definíció (Hibafüggvény). *Legyen F_n azon valós listák halmaza, melyekre $C^* = n$, azaz*

$$\mathcal{F}_n := \left\{ L \in \mathcal{D} \middle| C^*(L) = n \right\}.$$

Ekkor valamely A algoritmus hibafüggvényének az

$$R_{A,n} := \sup_{L \in \mathcal{F}_n} \frac{C^A(L)}{n}$$

mennyiséget nevezzük.

1.2. definíció (Aszimptotikus hiba). *Valamely A algoritmus aszimptotikus hibájának az*

$$R_{A,\infty} := \limsup_{n\to\infty} R_{A,n}$$

mennyiséget nevezzük.

Megjegyezzük, hogy az irodalomban (lásd például [22]) az aszimptotikus hibát gyakran az utóbbival ekvivalens

$$R_{A,\infty} = \inf\left\{ r \geq 1 \middle| \exists N > 0 : \ \forall L \in \mathcal{D},\ C^*(L) \geq N : \ \frac{C^A(L)}{C^*(L)} \leq r \right\}$$

képlettel közvetlenül definiálják.

1.3. definíció (Abszolút hiba). *Valamely A algoritmus abszolút hibájának az*

$$R_A := \sup_{L\in\mathcal{D}} \frac{C^A(L)}{C^*(L)}$$

mennyiséget nevezzük.

Megjegyezzük, hogy a ládapakolási feladat azon kevés kombinatorikus optimalizálási feladatok közé tartozik, melyekre egy adott algoritmus aszimptotikus és abszolút hibája különbözhet.

1.2. Ismert közelítő algoritmusok

Ebben a részben megismerkedünk néhány nevezetes közelítő algoritmussal és bemutatjuk a hibáikra ismert eredményeket. Így látunk néhány példát a bevezetett definíciók alkalmazására is.

1.2.1. A Next Fit algoritmus

A Next Fit (továbbiakban NF) algoritmus addig rakja az elemeket a soron következő ládába, amíg lehet. Ha egy elem már nem fér az adott ládába, akkor új ládát nyit és abba folytatja az elemek pakolását. Az algoritmust magyarul egyszerű algoritmusnak is nevezik.

Lássuk az algoritmus pszeudokódját [13].

Algoritmus 1.1 NF(n, L)

```
b_1 ← a_1
C^NF ← 1
for i = 2 → n do
  if b_{C^NF} + a_i ≤ 1 then
    b_{C^NF} ← b_{C^NF} + a_i
  else
    C^NF ← C^NF + 1
    b_{C^NF} ← a_i
  end if
end for
```

Ennek az algoritmusnak a helyigénye és a futásideje egyaránt $\Theta(n)$. Ha az elemek beolvasását és az eredmények kivitelét a cikluson belül oldjuk meg, akkor a helyfoglalás $\Theta(1)$-re csökkenthető, viszont a futásidő $\Omega(n)$.

Az algoritmus hibájára vonatkozó első jellemzés 1972-ből származik. D. S. Johnson doktori értekezésében [15] bebizonyította a következő tételt.

1.1. tétel (Johnson, 1972). *Ha $L \in \mathcal{D}$, akkor*

$$C^*(L) \le C^{NF}(L) \le 2 \cdot C^*(L) - 1.$$

Továbbá, ha $k \in \mathbb{Z}^+$, akkor léteznek olyan $L_1, L_2 \in \mathcal{D}$ listák, melyekre

$$C^*(L_1) = C^{NF}(L_1) = C^*(L_2),$$

viszont

$$C^{NF}(L_2) = 2 \cdot C^*(L_2) - 2.$$

Ennél pontosabb, éles korlátot ad a tétel következő javítása.

1.2. tétel (Iványi, 1984 [10]). *Ha $k \in \mathbb{Z}^+$, akkor léteznek olyan $L_1, L_2 \in \mathcal{D}$ listák, melyekre*

$$C^*(L_1) = C^{NF}(L_1) = C^*(L_2),$$

viszont

$$C^{NF}(L_2) = 2 \cdot C^*(L_2) - 1.$$

Így az NF algoritmus hibájának jellemző mennyiségei már felírhatók.

1.1. következmény. *Ha $n \in \mathbb{Z}^+$, akkor*

$$R_{NF,n} = 2 - \frac{1}{n},$$

továbbá

$$R_{NF} = R_{NF,\infty} = 2.$$

1.2.2. A Best Fit algoritmus

A Best Fit (továbbiakban BF) algoritmus a soron következő tárgyat mindig abba a ládába helyezi, ahol a lehető legkevesebb üres hely marad. Az algoritmust magyarul gazdaságos algoritmusnak is hívják.

Idézzük fel az algoritmus pszeudokódját is [13].

Algoritmus 1.2 BF(n, L)

$C^{BF} \leftarrow 1$
for $i = 1 \rightarrow n$ **do**
  $b_i \leftarrow 0$
end for
for $i = 1 \rightarrow n$ **do**
  $szabad \leftarrow 1.0$
  $ind \leftarrow 0$
  for $k = 1 \rightarrow C^{BF}$ **do**
    if $b_k + a_i \leq 1$ **and** $1 - b_k - a_i < szabad$ **then**
      $ind \leftarrow k$
      $szabad \leftarrow 1 - b_k - a_i$
    end if
  end for
  if $ind > 0$ **then**
    $b_{ind} \leftarrow b_{ind} + a_i$
  else
    $C^{BF} \leftarrow C^{BF} + 1$
    $b_{C^{BF}} \leftarrow a_i$
  end if
end for

Ennek az algoritmusnak a helyigénye $\Theta(n)$, időigénye pedig $O(n^2)$.

1.2.3. Az NFD algoritmus

A következő két algoritmus két részből áll. Először a lista elemeit nagyság szerinti csökkenő sorrendbe rendezzük, majd a második részben a rendezett lista elemeire alkalmazza valamelyik korábbi algoritmust. Erre utal az algoritmusok nevében szereplő D betű (**D**ecreasing).

A Next Fit Decreasing (NFD) algoritmus a rendezés után az egyszerű algoritmus (NF) szerint dolgozik. Hely- és időigénye az alkalmazott rendező algoritmus és NF megfelelő igényeiből tevődik össze.

1.2.4. A BFD algoritmus

A rendező gazdaságos algoritmus (Best Fit Decreasing) a rendezés után a gazdaságos algoritmus (BF) szerint dolgozik, így helyigénye $\Theta(n)$, időigénye pedig $O(n^2)$.

Az utolsó ismert közelítő algoritmus, amivel foglalkozunk, a First Fit nevet viseli. Vele és rendező változatával (FFD) a következő részben ismerkedünk meg.

2. A First Fit algoritmus és hibafüggvénye

Ebben a részben bemutatjuk a dolgozat fő témáját jelentő First Fit (továbbiakban FF) algoritmust és áttekintjük a hibájára ismert eredményeket. Megjegyezzük, hogy az algoritmust szokás magyarul mohó algoritmusnak is nevezni.

2.1. Az FF algoritmus

A First Fit alapgondolata, hogy a következő tárgyat mindig az első olyan ládába helyezi, amelybe belefér. Azaz, amikor a_i-t kell elhelyeznünk, akkor a legalacsonyabb indexű ládába tesszük azok közül, melyek telítettsége nem nagyobb, mint $1 - a_i$. Ha nincs ilyen, akkor új ládát nyitunk melyben a_i lesz az első elem.

Felidézzük az algoritmus pszeudokódját [13] is.

Algoritmus 2.1 FF(n, L)

```
C^FF ← 1
for i = 1 → n do
  b_i ← 0
end for
for i = 1 → n do
  k ← 1
  while b_k + a_i > 1 do
    k ← k + 1
  end while
  b_k ← b_k + a_i
  if k > C^FF then
    C^FF ← C^FF + 1
  end if
end for
```

Ennek az algoritmusnak a helyigénye $\Theta(n)$, időigénye pedig $O(n^2)$. Ha például minden fájlméret 1, akkor az algoritmus futási ideje $\Theta(n^2)$.

2.2. Az aszimptotikus hiba

A First Fit algoritmus aszimptotikus hibájának jellemzését a következő két tételre alapozzuk.

2.1. tétel (Iványi, 1984 [10, 11]). *Legyen n tetszőleges pozitív valós szám. Létezik olyan $L \in \mathcal{D}$ lista, melyre $C^*(L) = n$ és*

$$C^{FF}(L) = C^{BF}(L) = \left\lfloor \frac{17}{10} \cdot C^*(L) \right\rfloor .$$

2.2. tétel. *Létezik olyan $q > 0$ valós konstans, hogy minden $L \in \mathcal{D}$ esetén*

$$C^*(L) \leq C^{FF}(L) \leq \frac{17}{10}C^*(L) + q$$

egyenlőtlenségek teljesülnek.

Utóbbi tételt $q = 3$ érték mellett elsőként Ullman bizonyította [20]. 1974-ban D.S. Johnson, A. Demers, J.D. Ullman, M.R. Garey és R.L. Graham [16] $q = 2$ értékre is igazolták az állítást, és publikáltak olyan L listát, melyre $C^*(L) = 10$ és $C^{FF}(L) = 17$.

Érdekességképpen megjegyezzük, hogy ők akkor azt a sejtést fogalmazták meg, hogy minden $L \in \mathcal{D}$ listára ha $C^*(L) > 10$, akkor

$$C^{FF}(L) < \frac{17}{10}C^*(L).$$

Ez a sejtés később hamisnak bizonyult [10, 11], mint azt 2.1.tétel is mutatja.

Ezt követően 1976-ban Garey, Graham, Johnson és Yao [8] $q = \frac{9}{10}$-re közölt bizonyítást, mely hosszú ideig a legjobb becslésnek bizonyult.

Végül B. Xia és Z. Tan [22] 2010-ben igazolta $q = \frac{7}{10}$ mellett a tételt.

2.3. tétel (Xia, Tan, 2010). *Minden $L \in \mathcal{D}$ lista esetén az FF algoritmusra*

$$C^{FF}(L) \leq \frac{17}{10}C^*(L) + \frac{7}{10}$$

egyenlőtlenség teljesül.

2.3. Az abszolút hiba

Az abszolút hibáról D. Simchi-Levi [19] megmutatta, hogy minden $L \in \mathcal{D}$ listára

$$\frac{C^{FF}(L)}{C^*(L)} \leq \frac{7}{4}.$$

Iványi [12] erre az eredményre adott egyszerűbb bizonyítása kizárja az egyenlőség fennállásának lehetőségét is.

A legújabb eredmény az abszolút hiba esetében is Xia és Tan [22] nevéhez fűződik.

2.4. tétel (Xia, Tan, 2010). *Minden $L \in \mathcal{D}$ listára teljesül a*

$$C^{FF}(L) \leq \frac{12}{7}C^*(L)$$

egyenlőtlenség.

Cikkük végén a kínai szerzőpáros az alábbi problémát fogalmazza meg.

Probléma. *Azt sejtjük, hogy az FF algoritmus abszolút hibája pontosan* $\frac{17}{10}$*, ami azt jelenti, hogy FF abszolút és aszimptotikus hibája megegyezik. Ez nem gyakori a ládapakolási algoritmusoknál. A sejtés bizonyításához elsőként azt kellene eldönteni, hogy létezik-e olyan lista, melyre* $C^{FF} = 12$ *és* $C^* = 7$*, vagy nem.*

A dolgozat harmadik részét e probléma vizsgálatának szenteljük.

2.4. Az FFD algoritmus és hibafüggvénye

A korábbiakhoz hasonlóan az FF algoritmus esetén is értelmezhető a rendező változat. Az FFD (First Fit Decreasing) algoritmus először csökkenő sorrendbe rendezi a bemenő $L \in \mathcal{D}$ lista elemeit, majd FF szerint működik tovább. Helyigénye $\Theta(n)$, időigénye pedig $O(n^2)$.

Disszertációjában Johnson [15] megmutatta, hogy minden $L \in \mathcal{D}$ listára

$$C^{FFD}(L) \leq \frac{11}{9} C^*(L) + 4.$$

Megjegyezzük, hogy az FFD algoritmus aszimptotikus hibája nem lehet nagyobb, mint $\frac{11}{9}$ [16].

Az additív konstans értékét a fenti egyenlőtlenségben többeknek is sikerült csökkenteni. Ezek közül a legfrissebb eredmény Dósa György [5] nevéhez fűződik.

2.5. tétel (Dósa, 2007). *Minden* $L \in \mathcal{D}$ *listára*

$$C^{FFD}(L) \leq \frac{11}{9} C^*(L) + \frac{6}{9}$$

egyenlőtlenség teljesül.

Az abszolút hibát Simchi-Levi [19] határozta meg.

2.6. tétel (Simchi-Levi, 1994). *Minden* $L \in \mathcal{D}$ *listára*

$$\frac{C^{FFD}(L)}{C^*(L)} \leq \frac{3}{2}$$

egyenlőtlenség teljesül.

Ez a becslés egyben éles is, mivel a probléma megoldására nem létezik polinom idejű algoritmus, aminek az abszolút hibája $\frac{3}{2}$-nél kisebb [9] (kivéve persze akkor, ha $P = NP$ igaz).

Megjegyezzük, hogy bár a rendező FF algoritmus abszolút és aszimptotikus hibái kisebbek, mint a sima FF algoritmus esetén, ennek ellenére utóbbit is széles körben használják. Ennek oka, hogy a First Fit alkalmazható online algoritmusként [6], vagyis olyan esetekben, amikor az elemek valamilyen sorrendben érkeznek, és azonnal ládába kell helyeznünk őket, nincs információnk a későbbi elemekről.

2.5. Hibajellemzők összefoglalása

Az 2.1. ábrán összefoglaljuk a 6 alapvető algoritmus hibajellemzőivel kapcsolatos eddigi eredményeket. Az ábrán ϵ tetszőlegesen kis pozitív számot jelent, valamint

$$\gamma = \sum_{i=1}^{\infty} \frac{1}{a_i},$$

ahol $a_1 = 1$ és $i \in \mathbb{N}^+$ esetén

$$a_{i+1} = a_i(a_i + 1).$$

Ekkor $\gamma \sim 1.691$.

A	$R_{A,n}$	R_A	$R_{A,\infty}$
NF	$2 - \frac{1}{n}$[10]	2 [15]	2 [15]
FF	$\frac{\lfloor 1.7n \rfloor}{n} \le R_{FF,n} \le 1.7 + \frac{0.7}{n}$ [10, 22]	$\frac{17}{10} \le R_{FF} \le \frac{12}{7}$ [10, 22]	$\frac{17}{10}$ [20]
BF	$\frac{\lfloor 1.7n \rfloor}{n} \le R_{BF,n} \le 1.7 + \frac{0.7}{n}$[10, 16]	$\frac{\lfloor 1.7n \rfloor}{n} \le R_{BF} \le \frac{17}{10} + \frac{2}{n}$ [10, 22]	$\frac{17}{10}$ [20]
NFD	$\gamma \le R_{NFD,n} \le \gamma + \frac{3}{n}$[1]	$\gamma \le R_{NFD} \le 2 - \frac{1}{n}$[1]	γ [1]
FFD	$\frac{11}{9} - \frac{2}{n} \le R_{FFD,n} \le \frac{11}{9} + \frac{6}{9n}$ [16, 5]	$\frac{3}{2}$ [19]	$\frac{11}{9}$ [16]
BFD	$\frac{11}{9} - \frac{2}{n} \le R_{BFD,n} \le \frac{11}{9} + \frac{1}{n}$ [23]	$\frac{3}{2}$ [19]	$\frac{11}{9}$ [16]

2.1. ábra. *Alapvető ládapakoló algoritmusok hibajellemzői.*

Mint látjuk, az abszolút hiba értékére FF, BF és NFD esetén csak becslések állnak rendelkezésünkre. Ebben a dolgozatban FF abszolút hibájának felső becslését javítjuk: a 3. részben bebizonyítjuk

$$R_{FF} \le \frac{101}{59}$$

egyenlőtlenséget.

3. Az FF algoritmus abszolút hibájáról

Ebben a részben a korábban megfogalmazott problémával foglalkozunk. Célunk, hogy bebizonyítsuk a következő állítást.

3.1. tétel. *Minden $L \in \mathcal{D}$ listára teljesül a*

$$C^{FF}(L) \leq \frac{101}{59} C^*(L)$$

egyenlőtlenség.

Ezt több lépésen keresztül fogjuk megtenni. Elsőként bevezetjük a szükséges jelöléseket és megfogalmazzuk az alapvető segédtételeket. Ezután megmutatjuk, hogy Xia és Tan [22] már ismertetett felső korlátja nem éles. Végül eljutunk a 3.1. tétel felső becsléséhez.

3.1. A telítettségi lemmák

Először is bevezetjük a szükséges jelöléseket. Ezek nagy része az egyszerűség kedvéért megegyezik Xia és Tan terminológiájával.

Az $\frac{1}{2}$-nél nagyobb elemeket *nagynak*, az $\frac{1}{4}$-nél nagyobbakat *közepesnek* nevezzük. Figyeljünk oda, hogy közepes elem is lehet $\frac{1}{2}$-nél nagyobb.

Jelölje $\mathfrak{B}^*$ az optimális pakoláshoz használt ládák halmazát és $\mathfrak{B}^{FF}$ az FF algoritmus által használtakét. Ha egy B_1 ládát az FF algoritmus egy másik B_2 ládánál korábban nyitott meg, akkor azt mondjuk, hogy B_1 *megelőzi* B_2-t. Az FF algoritmus terminálása után azokat a ládákat, melyek pontosan egy (kettő) elemet tartalmaznak, i-*ládának* (ii-*ládának*) nevezzük.

A legalább kettő (három, négy) elemet tartalmazó ládákat II-*ládának* (III-*ládának*, IV-*ládának*) nevezzük. Egy i-láda (ii-láda, II-láda, III-láda, IV-láda) egy elemét i-*elemnek* (ii-*elemnek*, II-*elemnek*, III-*elemnek*, IV-*elemnek*) nevezzük.

Legyen $\mathfrak{B}_i$ ($\mathfrak{B}_{ii}, \mathfrak{B}_{II}, \mathfrak{B}_{III}, \mathfrak{B}_{IV}$) az i-ládák (ii-, II-, III-, IV-ládák) halmaza, és N_i ($N_{ii}, N_{II}, N_{III}, N_{IV}$) az i-ládák (ii-, II-, III-, IV-ládák) száma.

Ekkor nyilván

$$\mathfrak{B}^{FF} = \mathfrak{B}_i \cup \mathfrak{B}_{II} = \mathfrak{B}_i \cup \mathfrak{B}_{ii} \cup \mathfrak{B}_{III},$$

valamint

$$C^{FF} = N_i + N_{II} = N_i + N_{ii} + N_{III}.$$

Egyszerűen adódik a következő lemma.

3.1. lemma. $C^* \geq N_i$.

Bizonyítás. Nyilvánvaló, hogy bármely két i-elem összmérete 1-nél nagyobb, különben FF nem nyitott volna új ládát a később érkezőnek (hanem egymásra rakta volna őket). De emiatt bármely két ilyen elem nem kerülhet egy ládába az optimális pakolás esetén sem. Tehát $C^* \geq N_i$. □

A következő két lemmát telítettségi lemmáknak is szokás nevezni. Kimondásukon túl bizonyításaikat is felidézzük most, mivel az alkalmazott technikákat a későbbiekben is használjuk.

3.2. lemma (Iványi [12], 2004). *Legyen $k \geq 1$ egész és legyenek $B_1, B_2, \ldots, B_{k+1}$ ládák $\mathfrak{B}^{FF}$-ben. Tegyük fel, hogy minden $i = 1, 2, \ldots, k$ indexre B_i megelőzi B_{k+1}-et és B_{k+1} legalább k elemet tartalmaz. Ekkor*

$$\sum_{i=1}^{k+1} B_i > k.$$

Bizonyítás. Legyen a B_{k+1} ládában elhelyezett legkisebb elem mérete x. Ekkor minden $i = 1, 2, \ldots, k$ indexre $B_i > 1 - x$, mivel x csak úgy kerülhetett B_{k+1}-be, hogy egyik őt megelőző ládába sem fért bele. Továbbá $B_{k+1} > kx$, mivel B_{k+1}-ben legalább k elem van és x a legkisebb méretű. Így

$$\sum_{i=1}^{k+1} B_i > k(1-x) + kx = k,$$

ezt kellett bizonyítanunk. □

Most kimondjuk és bebizonyítjuk a második telítettségi lemmát.

3.3. lemma (Xia, Tan [22], 2010). *Legyenek $k \geq 1$ és $M \geq k+1$ egészek. Ha $B_1, B_2, \ldots, B_M$ ládák $\mathfrak{B}^{FF}$-beliek és mindegyikük legalább k darab elemet tartalmaz, akkor*

$$\sum_{i=1}^{M} B_i > \frac{kM}{k+1}.$$

Bizonyítás. M szerinti indukcióval bizonyítunk. Az állítás igaz $M = k+1$-re, mivel ekkor az állítás 3.2. lemma speciális esete. Tegyük fel, hogy az állítás igaz $M = j \geq k+1$-re.

Alkalmazzuk 3.2. lemmát $B_i, B_i, \ldots, B_i, B_{j+1}$ ládákra. Ekkor

$$kB_i + B_{j+1} > k$$

minden $i = 1, 2, \ldots, j$ indexre. Összeadva ezt a j darab egyenlőtlenséget

$$k\sum_{i=1}^{j} B_i + jB_{j+1} > jk$$

adódik.

Ebből és az indukciós feltevésből kapjuk

$$\sum_{i=1}^{j+1} B_i = \frac{j\sum_{i=1}^{j} B_i + jB_{j+1}}{j} = \frac{k\sum_{i=1}^{j} B_i + jB_{j+1} + (j-k)\sum_{i=1}^{j} B_i}{j} >$$

$$> \frac{jk + (j-k)\frac{kj}{k+1}}{j} = \frac{k(j+1)}{k+1}$$

egyenlőtlenséget, ami épp a bizonyítandó állítás $M = j+1$-re. Tehát igazoltuk a lemmát. □

Vegyük észre, hogy a két lemma nem ekvivalens, és egyik sem következménye a másiknak: 3.2. lemma hátránya, hogy csak $k+1$ darab ládára alkalmazható, míg 3.3. lemma megköveteli minden ládára a legalább k-s elemszámot.

Utóbbira adott bizonyításunkban viszont nem használjuk ki teljesen a feltételeket: valójában egyúttal a következő lemmát is igazoltuk, mely 3.2. és 3.3. lemmák általánosítása.

3.4. lemma (Általános telítettségi lemma). *Legyenek $k \geq 1$ és $M \geq k+1$ egészek. Továbbá legyenek $B_1, B_2, \ldots, B_M$ ládák $\mathfrak{B}^{FF}$-beliek. Tegyük fel, hogy minden $j = k+1, k+2, \ldots, M$ indexre B_j legalább k elemet tartalmaz és minden $i = 1, 2, \ldots, k$ indexre B_i megelőzi B_j-t. Ekkor*

$$\sum_{i=1}^{M} B_i > \frac{kM}{k+1}.$$

3.1. megjegyzés. A telítettségi lemmák láthatóan arra használhatók, hogy bizonyos feltételek teljesülése esetén alulról tudjuk becsülni néhány ládára a bennük lévő tárgyak összméretét. Ez különösen hasznos lehet például akkor, ha ismerjük az FF által használt C^{FF} ládaszámot (mely gyakran egy előre rögzített érték a hibafüggvény elemzésekor): ha ugyanis az összes ládába elhelyezett tárgyak

$$\sum_{B \in \mathfrak{B}^{FF}} B$$

összméretét alulról tudjuk becsülni, akkor az egyben a C^* ládaszámnak is alsó becslése lesz. Fennáll ugyanis a következő egyenlőtlenség.

3.1. állítás. *Tetszőleges $L \in \mathcal{D}$ listára az FF algoritmus által használt ládák halmazát $\mathfrak{B}^{FF}(L)$-el jelölve teljesül a*

$$\sum_{B \in \mathfrak{B}^{FF}(L)} B \leq C^*(L)$$

egyenlőtlenség.

3.2. megjegyzés. A telítettségi lemmák más szemléletes tartalommal is bírnak. 3.4. lemma egyenlőtlenségében M-el osztva

$$\frac{1}{M}\sum_{i=1}^{M} B_i > \frac{k}{k+1}$$

adódik. Ebben az értelemben a lemma állítása az, hogy ha FF algoritmus M darab ládájára bizonyos feltételek teljesülnek, akkor ezen ládák átlagos telítettsége alulról becsülhető.

3.2. A $C^* = 7$ és $C^{FF} = 12$ probléma

Mint már említettük, elsőként azt szeretnénk megmutatni, hogy 2.4. tétel felső becslése nem éles. Ehhez szükségünk lesz a következő állításra.

3.2. tétel. *Nem létezik olyan $L \in \mathcal{D}$ lista, melyre*

$$C^{FF}(L) = 12 \qquad \textit{és} \qquad C^*(L) = 7.$$

Bizonyítás. A bizonyítást több lépésen keresztül végezzük.

A) Tekintsük azoknak a listáknak a halmazát, melyeket az FF algoritmus 12 ládába pakol:

$$\mathcal{D}_{12}^{FF} := \left\{ L \in \mathcal{D} \middle| C^{FF}(L) = 12 \right\}.$$

A továbbiakban csak $L \in \mathcal{D}_{12}^{FF}$ listákkal foglalkozunk. Ekkor $\mathfrak{B}^{FF}(L)$ halmaz tetszőleges L listára nyilván 12 elemű, jelöljük ezeket rendre $B_1(L), B_2(L), \ldots, B_{12}(L)$-el úgy, hogy minden $1 \leq i < j \leq 12$ indexekre B_i megelőzi B_j-t.

Azt kell belátnunk, hogy minden $L \in \mathcal{D}_{12}^{FF}$ listára $C^*(L) > 7$. L kiírását továbbra is elhagyjuk olyan esetekben, amikor az egyértelműség és érthetőség nem sérül.

B) Ha a pontosan egy elemet tartalmazó i-ládák N_i számára

$$N_i \geq 8,$$

akkor 3.1. lemma alapján $C^* \geq 8$.

$N_i = 0$ esetén az összes láda $\mathfrak{B}_{II}$-beli, így 3.3. lemmát alkalmazhatjuk $M = 12$ és $k = 2$ mellett. Ekkor

$$\sum_{i=1}^{12} B_i > 12 \cdot \frac{2}{3} = 8,$$

így 3.1. állítás alapján kész vagyunk.

Ha $N_i = 1$, akkor $\mathfrak{B}_{II}$ halmaz 11 elemére alkalmazható 3.3. lemma, ahol $M = 11$ és $k = 2$. Így

$$\sum_{B \in \mathfrak{B}_{II}} B > 11 \cdot \frac{2}{3} > 7,$$

és 3.1. állítás alapján kész vagyunk.

Végül legyen

$$2 \leq N_i \leq 6.$$

Ekkor $\mathfrak{B}_i$ elemeire 3.3. lemmát alkalmazzuk, ahol $k = 1$ és $M = N_i$. A lemma $\mathfrak{B}_{II}$ elemeire is alkalmazható $k = 2$ és $M = 12 - N_i$ értékekkel.

Ezek alapján

$$\sum_{i=1}^{12} B_i = \sum_{B \in \mathfrak{B}_i} B + \sum_{B \in \mathfrak{B}_{II}} B > N_i \cdot \frac{1}{2} + (1 - N_i) \cdot \frac{2}{3} \geq 6 \cdot \frac{1}{2} + 6 \cdot \frac{2}{3} = 7,$$

és 3.1. állítás alapján kész vagyunk.

C) Tehát legyen a továbbiakban

$$N_i = 7.$$

Tegyük fel, hogy van olyan II-láda, melyet megelőz i-láda, vagyis léteznek olyan $1 \leq p < q \leq 12$ indexek, hogy $B_p \in \mathfrak{B}_i$, $B_q \in \mathfrak{B}_{II}$ és B_p megelőzi B_q-t.

Ekkor legyen $r \neq q$ tetszőleges olyan index, hogy $B_r \in \mathfrak{B}_{II}$. Mivel B_p megelőzi B_q-t és utóbbi legalább 2 elemet tartalmaz, így 3.2. lemma biztosan alkalmazható r értékétől függetlenül, és

$$B_p + B_q + B_r > 2$$

adódik.

Ugyanezen lemmát $\mathfrak{B}_{II}$ eddig kimaradó 3 elemére alkalmazva

$$\sum_{B \in \mathfrak{B}_{II} \setminus \{B_q, B_r\}} B > 2.$$

Végül alkalmazzuk 3.3. lemmát $\mathfrak{B}_i$ elemeire B_p kivételével. Ekkor $k = 1$ és mivel 6 ilyen láda van így $M = 6$, tehát

$$\sum_{B \in \mathfrak{B}_i \setminus \{B_p\}} B > 6 \cdot \frac{1}{2} = 3$$

egyenlőtlenség adódik.

A három egyenlőtlenséget összeadva kapjuk, hogy ekkor

$$\sum_{i=1}^{12} B_i > 2 + 2 + 3 = 7,$$

és 3.1. állítás alapján kész vagyunk.

D) Az az eset marad, amikor $B_6, B_7, \ldots, B_{12}$ i-ládák, és $B_1, B_2, \ldots, B_5$ II-ládák. Indirekt tegyük fel, hogy $C^* = 7$ lehetséges ($N_i = 7$ miatt kevesebb biztos nem lehet).

$\mathfrak{B}_i$ elemeire 3.3. lemmát alkalmazva

$$\sum_{i=6}^{12} B_i > 7 \cdot \frac{1}{2} = 3.5 \tag{3.1}$$

adódik.

Alkalmazzuk 3.2. lemmát B_3, B_4, B_5 ládákra, ekkor

$$\sum_{i=3}^{5} B_i > 2. \tag{3.2}$$

Az indirekt feltevés miatt 3.1. állítás alapján

$$\sum_{i=1}^{12} B_i \leq 7$$

egyenlőtlenségnek is fenn kell állnia, amit átrendezve és (3.1), (3.2) egyenlőtlenségeket alkalmazva

$$\sum_{i=1}^{5} B_i \leq 7 - \sum_{i=6}^{12} B_i < 7 - 3.5 = 3.5, \tag{3.3}$$

valamint

$$B_1 + B_2 \leq 7 - \sum_{i=3}^{12} B_i < 7 - (3.5 + 2) = 1.5 \tag{3.4}$$

egyenlőtlenségek adódnak.

Utóbbi egyúttal azt is jelenti, hogy B_1 és B_2 közül legalább az egyik 0.75-nél kisebb.

E) Az i-ládákban lévő összesen 7 darab elemről tudjuk, hogy bármely kettő összege 1-nél nagyobb (különben FF nem rakta volna őket külön ládákba). Ráadásul ha lenne köztük kettő, melyek mérete legfeljebb $\frac{1}{2}$, akkor őket közös ládába rakta volna FF, tehát ez sem fordulhat elő.

Következésképpen a $\mathfrak{B}_i$ ládáiban található elemek között legalább 6 nagy elem van.

Ha a hetedik elem mérete legfeljebb $\frac{1}{4}$, akkor a nagy elemek mindegyike biztosan $\frac{3}{4}$-nél is nagyobb, különben a hetedik elem nem kerülne külön ládába FF alkalmazásakor. Így a nagy elemek összmérete legalább

$$6 \cdot \frac{3}{4} > 4.$$

A $\mathfrak{B}_{II}$-beli ládák tartalmának összege 3.3. lemmával becsülhető ($M = 5$ és $k = 2$), ez alapján

$$\sum_{i=1}^{5} B_i > 5 \cdot \frac{2}{3} > 3.$$

Mivel a nagy elemek mindegyike $\mathfrak{B}_i$-beli ládában van, így

$$\sum_{i=1}^{12} B_i = \sum_{B \in \mathfrak{B}_i} B + \sum_{i=1}^{5} B_i > 4 + 3 = 7,$$

ami ellentmondás.

Tehát a $\mathfrak{B}_i$ halmaz ládáiban található 7 darab elem között 6 nagy és 1 közepes van (utóbbi mérete is lehet $\frac{1}{2}$-nél nagyobb).

Ez egyben azt is jelenti, hogy ezen a 7 elemen kívül legfeljebb 8 darab közepes elem lehet a listában. Ugyanis a $\mathfrak{B}_i$-beli ládák elemei a $\mathfrak{B}^{OPT}$ optimális pakolás esetén is külön ládákba kerülnek, mert bármelyik kettő összege nagyobb, mint 1. Nyilvánvaló, hogy nagy elem mellé legfeljebb egy darab közepes, és közepes elem mellé legfeljebb két darab közepes elem fér egy ládába. Mivel feltevésünk szerint $C^* = 7$, így maximum

$$6 \cdot 1 + 1 \cdot 2 = 8$$

további közepes elem lehetséges.

Ezek a közepes elemek nyilván $\mathfrak{B}_{II}$-beli ládába kerülnek.

F) Tegyük fel, hogy

$$B_1 \leq 0.75.$$

Ekkor FF algoritmus végrehajtása során a legelső B_1 ládán csak közepes elemek juthattak túl (a kisebb elemeket FF ebbe a ládába helyezte volna).

Mivel

$$\{B_2, B_3, B_4, B_5\} \subset \mathfrak{B}_{II},$$

így ezen ládák mindegyikébe csak úgy kerülhetett legalább 2 elem, ha az i-ládák elemein kívül pontosan 8 közepes elem van. Ezek közül a B_2, B_3, B_4, B_5 ládák mindegyikébe pontosan 2 darab került.

Mivel $C^* = 7$, így e 8 közepes elem között van két olyan, melyek $\mathfrak{B}^{OPT}$ optimális pakolás esetén azonos ládába kerülnek. Jelöljük ezeket az elemeket x_2, x_3-mal és legyen x_1 az a $\mathfrak{B}^{OPT}$ pakolásban velük egy ládába kerülő elem, melyet FF $\mathfrak{B}_i$ halmaz egy ládájába helyez (az előző pont szerint minden $\mathfrak{B}^{OPT}$-beli ládában van ilyen).

Ekkor

$$x_1 + x_2 + x_3 \leq 1. \tag{3.5}$$

Épp ezért $\mathfrak{B}^{FF}$ pakolásban x_2 és x_3 nem kerülhet egy ládába, mert akkor x_1 elemet FF ebbe (vagy ezt megelőző) ládába rakná, pedig x_1 egy $\mathfrak{B}_i$-beli láda eleme.

Tegyük fel tehát, hogy x_2-t FF algoritmus x_3 ládáját megelőző ládába helyezte, és a mellé kerülő közepes elemet jelölje y_2. Legyen y_1 a $\mathfrak{B}^{OPT}$ optimális pakolásban y_2-vel egy ládába kerülő azon elem, melyet FF i-ládába tesz. Ekkor

$$y_1 + y_2 \leq 1 \tag{3.6}$$

és

$$x_1 + y_1 > 1 \tag{3.7}$$

egyenlőtlenségek nyilván teljesülnek.

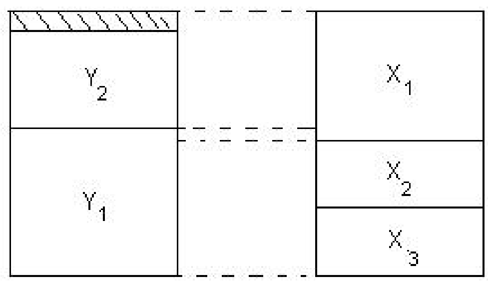
Összeadva (3.5) és (3.6) egyenlőtlenségeket

$$(x_1 + y_1) + (x_2 + x_3 + y_2) \leq 2$$

adódik, így (3.7) felhasználásával

$$x_2 + x_3 + y_2 < 1.$$

Ez azt jelenti, hogy FF algoritmus során x_3 nem juthat túl azon a ládán, melyben csak x_2 és y_2 elemek vannak. Az itt tárgyalt elemek közti egyenlőtlenségeket szemlélteti az 3.1. ábra.



3.1. ábra

Ellentmondásra jutottunk, és mivel más eset nem lehetséges, így

$$B_1 > 0.75$$

kell teljesüljön.

3.3. lemma alkalmazásával

$$\sum_{i=2}^{5} B_i > 4 \cdot \frac{2}{3} = \frac{8}{3}$$

egyenlőtlenség adódik, így (3.3) felhasználásával

$$\frac{3}{4} < B_1 < \frac{5}{6}. \tag{3.8}$$

G) Ekkor (3.4) miatt

$$B_2 < \frac{3}{4}$$

is teljesül.

Ez jelenti, hogy B_3, B_4, B_5 ládákban csak közepes elemek lehetnek. Ha közülük egy B_i, $i \in \{3, 4, 5\}$ ládában 3 elem van, akkor nyilván

$$B_i > 3 \cdot \frac{1}{4},$$

valamint (3.8) miatt

$$B_1 > \frac{3}{4},$$

a többi II-ládára pedig 3.3. lemma alapján

$$\sum_{B \in \mathfrak{B}_{II} \setminus \{B_1, B_i\}} B > 3 \cdot \frac{2}{3} = 2.$$

Így

$$\sum_{j=1}^{5} B_j > 3 \cdot \frac{1}{4} + \frac{3}{4} + 2 = 3.5,$$

ami ellentmond (3.1) egyenlőtlenségnek.

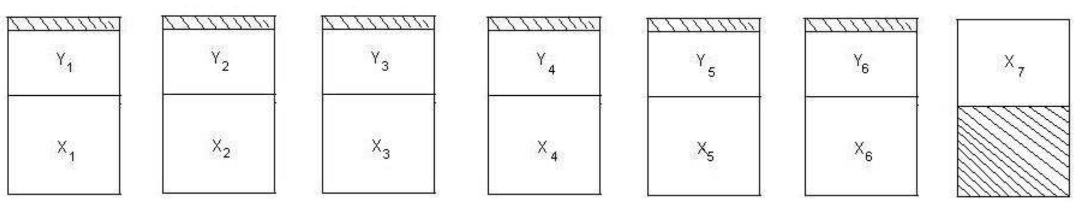
Tehát B_3, B_4, B_5 mindegyikében 2 darab közepes elem található.

H) Ezen 6 darab közepes elem közül $\mathfrak{B}^{OPT}$ optimális pakolásban semelyik kettő sem kerül ugyanabba a ládába. Ha ugyanis lennének ilyen elemek, akkor az **F)** pontban látott módon megmutatható, hogy ezek nem kerülhetnek az FF algoritmus $\mathfrak{B}^{FF}$ pakolásában egy ládába, de egyikük se kerülhet a másikét megelőző ládába sem.

Tehát $\mathfrak{B}^{OPT}$ ládái között 6 olyan van, melyben található egy FF által i-ládába tett és egy B_3, B_4, B_5 valamelyikébe tett elem is. Vezessük be ezekre a ládákra

$$B_1^{OPT}, B_2^{OPT}, \dots, B_6^{OPT}$$

jelöléseket. Az utolsó láda legyen B_7^{OPT}. Ezt a helyzetet szemlélteti a 3.2. ábra.



3.2. ábra

Az ábrának megfelelően jelöljük a továbbiakban B_i^{OPT} azon elemét, melyet FF i-ládába pakol x_i-vel ($i \leq 7$), a B_3, B_4, B_5 valamelyikébe helyezett elemét pedig y_i-vel.

I) Vizsgáljuk most a B_2 ládában lévő elemek számát. (3.8) egyenlőtlenség miatt az első ládán túljutó elemek mérete biztosan nagyobb, mint $\frac{1}{6}$.

Először tegyük fel, hogy B_2 ládában legalább négy elem van. Ekkor legyen ezek közül a legkisebb mérete

$$\frac{1}{6} + x,$$

ahol $x > 0$ valós szám. Mivel ezt az elemet FF nem B_1-be rakta, így B_1 nyilván

$$B_1 > \frac{5}{6} - x.$$

Viszont most az első két ládában lévő elemek összméretére

$$B_1 + B_2 > \frac{5}{6} - x + 4 \cdot \left(\frac{1}{6} + x\right) = 1.5 + 3x$$

egyenlőtlenség adódik, ami ellentmond (3.4)-nek.

Most tegyük fel, hogy B_2 elemeinek száma 2. Ez a két elem $\mathfrak{B}^{OPT}$ optimális pakolásban nyilván nem ugyanabban a B_i^{OPT} ládában van, különben az ott mellettük lévő x_i elemet FF B_2 ládába tenné.

A két elem közül legalább az egyik így valamely B_i^{OPT}, $i \leq 6$ ládában van $\mathfrak{B}^{OPT}$-ban. A másik elem ládája legyen ekkor B_j^{OPT}, $j \neq i$. Így biztosan fennáll

$$x_i + x_j + y_i + B_2 \leq 2$$

egyenlőtlenség, hisz minden bal oldali elem B_i^{OPT} és B_j^{OPT} valamelyikében van.

De teljesül

$$x_i + x_j > 1$$

is, mert mindkét elemet FF i-ládába teszi. Emiatt viszont

$$y_i + B_2 < 1,$$

tehát FF y_i-t B_2-be kellene, hogy tegye, nem B_3, B_4, B_5 valamelyikébe, ellentmondás.

Végezetül nézzük azt a lehetőséget, hogy B_2 elemeinek száma 3. $\mathfrak{B}^{OPT}$ optimális pakolásban nem lehet mindhárom elem ugyanabban a ládában, különben az ott mellettük lévő x_i elemet FF B_2 ládába tenné. Sőt, semelyik kettő sem lehet azonos B_i^{OPT} ládában, különben tekintsük őket egyetlen elemnek, mely mérete a kettő összege, és alkalmazzuk az előbbi, kételemű B_2-re látott gondolatmenetet.

Tehát különböző $\mathfrak{B}^{OPT}$-beli ládákban vannak B_2 elemei. Ez azt jelenti, hogy legalább kettő a

$$\left\{B_i^{OPT} :\ i \leq 6\right\}$$

ládák valamelyikében van, és mivel $x_1, x_2, \ldots, x_7$ elemek közül legfeljebb egy nem nagy (lásd **E)**), így van olyan $j \leq 6$ index, hogy B_j^{OPT} ládában x_j nagy és található benne B_2-beli elem. Jelölje ezt az elemet z.

Ekkor nyilván

$$x_j + y_j + z \leq 1,$$

és mivel y_j-t FF nem B_2-be, hanem nála későbbi ládába helyezi, így

$$B_2 + y_j > 1.$$

Ezekből az egyenlőtlenségből B_2 z-n kívüli két elemére

$$B_2 - z > x_j > \frac{1}{2}$$

adódik (mivel x_j nagy).

Mivel z elemet FF nem B_1 ládába rakja, így fenn kell álljon

$$B_1 + z > 1$$

egyenlőtlenség. Ekkor viszont

$$B_1 + B_2 = (B_1 + z) + (B_2 - z) > 1 + \frac{1}{2} = 1.5,$$

ami ellentmond (3.4) egyenlőtlenséggel. Más eset nem lehetséges, így ezzel a bizonyítást befejeztük. □

3.3. Szigorú egyenlőtlenség

Mielőtt továbblépünk, felidézünk egy diofantikus egyenletekre vonatkozó lemmát.

3.5. lemma (Diofantikus egyenletek, [18]). *Legyenek a és b relatív prímek, u tetszőleges egész szám. Az*

$$ax + by = u$$

lineáris diofantikus egyenletnek végtelen sok megoldása van. Ha (x_0, y_0) pár egész megoldás, akkor az összes többi előállítható

$$x = x_0 + bv \qquad y = y_0 - av$$

alakban, ahol v egész.

Most már minden eszköz a rendelkezésünkre áll ahhoz, hogy megmutassuk: a 2.4. tételben szereplő egyenlőtlenségben az egyenlőség esete nem állhat fenn. Lássuk be tehát a következő tételt.

3.3. tétel. *Minden $L \in \mathcal{D}$ listára teljesül a*

$$C^{FF}(L) < \frac{12}{7} C^*(L)$$

egyenlőtlenség.

Bizonyítás. 2.4. tétel alapján nyilván elegendő megmutatnunk, hogy nem létezik olyan $L \in \mathcal{D}$ lista, melyre

$$7 \cdot C^{FF}(L) - 12 \cdot C^*(L) = 0.$$

Mivel ennek a diofantikus egyenletnek $C^{FF} = 12$ és $C^* = 7$ megoldása, így 3.5. lemma alapján kapjuk az összes pozitív egész megoldást:

$$C^{FF} = 12k \qquad C^* = 7k,$$

ahol $k \in \mathbb{Z}^+$.

Alkalmazzuk 2.3. tételt a megoldáspárokra. Ez alapján csak olyan k értékekre létezhet $L \in \mathcal{D}$ lista $C^{FF}(L) = 12k$ és $C^*(L) = 7k$ paraméterekkel, melyekre

$$12k \leq \frac{17}{10} \cdot 7k + \frac{7}{10}$$

teljesül, ami csak $k = 1$ pozitív egészre igaz.

$k = 1$ esetén viszont 3.2. tétel alapján nem létezik lista a kívánt paraméterekkel.

□

3.4. A 3.1. tétel bizonyítása

Most bebizonyítjuk a fejezet elején kimondott tételt.

Tekintsük azokat az $a, b \in \mathbb{Z}^+$ számpárokat, melyekre

$$\frac{17}{10} < \frac{a}{b} < \frac{12}{7},$$

valamint

$$a \leq \frac{17}{10} b + \frac{7}{10}$$

egyenlőtlenségek teljesülnek, és $b \leq 59$.

Az ilyen tulajdonságú számpárokból összesen 20 darab van, ezek hányadosai:

$$\begin{gathered} S = \{29/17, 41/24, 46/27, 53/31, 58/34, 63/37, 65/38, \\ 70/41, 75/44, 77/45, 80/47, 82/48, 87/51, \\ 89/52, 92/54, 94/55, 97/57, 99/58, 101/59\}. \end{gathered}$$

A halmaz legnagyobb eleme
$$\max S = \frac{101}{59}.$$
Vegyük észre, hogy $C^* = 59$ és $C^{FF} = 101$ választással a 2.3. tételben egyenlőség áll fenn, azaz
$$C^{FF} = \frac{17}{10} \cdot C^* + \frac{7}{10}.$$

Ebből viszont következik, hogy minden $p, q \in \mathbb{Z}^+$ számokra ha
$$\frac{p}{q} \geq \frac{101}{59},$$
és $q > 59$, akkor
$$p > \frac{17}{10} \cdot q + \frac{7}{10},$$
tehát nem létezik olyan $L \in \mathcal{D}$ lista, melyre $C^*(L) = q$ és $C^{FF}(L) = p$.

Vagyis minden L listára
$$C^{FF}(L) \leq \frac{101}{59} C^*(L),$$
és ezt kellett bizonyítanunk. $\square$

Köszönetnyilvánítás

A kutatáshoz és a dolgozat elkészítéséhez az Eötvös Loránd Tudományegyetem TAMOP 4.2.1/B-09/1/KMR-2010-0003 számú projektje nyújtott támogatást.

Hivatkozások

[1] B. S. Baker, E. G. Coffman, Jr., A tight asymptotic bound for next-fit-decreasing bin-packing. *SIAM Journal on Algebraic and Discrete Methods*, **2** (1981) 147–152.

[2] E. G. Coffman Jr., M. R. Garey, D. S. Johnson, Approximation algorithms for bin-packing: An updated survey. In G. Ausiello, M. Lucertini, P. Serani, editors, *Algorithm Design for Computer SystemDesign*. Springer-Verlag, Wien, 1984. CISM Courses and Lectures Number 284, 49–106.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, The MIT Press/McGraw-Hill, 2009 (Harmadik kiadás). Magyarul: *Algoritmusok*, Műszaki Kiadó, Budapest, 1999.

[4] Csirik János, *Ládapakolási algoritmusok*, Szeged, 1989, 123 oldal.

[5] G. Dósa, The tight bound of first fit decreasing bin-packing algorithm is $FFD(L) \leq \frac{11}{9}OPT(L) + \frac{6}{9}$, *Lecture Notes in Computer Science* **4614** (2007) 1–11.

[6] Dósa György, Imreh Csanád, *Online algoritmusok.* Elektronikus jegyzet. SZTE TTIK, Szeged, 2011 (kézirat).

[7] M. R. Garey, R. L. Graham, J. D. Ullman, An analysis of some packing algorithms. In: *Combinatorial Algorithms*, ed. R. Rustin. Algorithmic Press, 1973, 39–48.

[8] M.R. Garey, R.L. Graham, D.S. Johnson, A.C.C. Yao, Resource constrained scheduling as generalized bin packing, *Journal of Combinatorial Theory*, Series A **21** (1976) 257–298.

[9] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1978.

[10] A. Iványi, Performance bounds for simple bin packing algorithms. *Ann. Univ. Sci. Budapest. Sect. Comput.* **5** (1984) 77–82.

[11] A. Iványi, Estimation of the efficiency of bin-packing algorithms. (Russian) *Problemy Kibernet.* **41** (1984) 253–256.

[12] Iványi A, *Processzorütemezés.* Kézirat. ELTE, Numerikus és Gépi Matematika Tanszék, Budapest, 2004.

[13] Iványi Antal, *Informatikai algoritmusok.* ELTE Eötvös Kiadó, Budapest, 2004.

[14] Iványi Antal, Hámori Árpád, Madarász János, Németh Zsolt, *Algoritmusok hatékonyságának oktatása.* Az Informatika a Felsőoktatásban 2011 konferenciára benyújtott előadáskivonat. Budapest, 2011.

[15] D.S. Johnson, *Near-optimal bin packing algorithms*, Doctoral Thesis, MIT, Cambridge, MA, 1973.

[16] D.S. Johnson, A. Demers, J.D. Ullman, M.R. Garey, R.L. Graham, Worst-case performance bounds for simple one-dimensional packing algorithms, *SIAM Journal on Computing* **3** (1974) 299–325.

[17] Lovász László, Gács Péter, *Algoritmusok*, Tankönyvkiadó, Budapest, 1989.

[18] I. Niven, H.S. Zuckerman, H.L. Montgomery, *An Introduction to the Theory of Numbers*, 5th ed., John Wiley & Sons, 1991.

[19] D. Simchi-Levi, New worst-case results for the bin-packing problem, *Naval Research Logistics* **41** (1994) 579–585.

[20] J. D. Ullman, *The performance of a memory allocation algorithm*, Technical Report 100, Princeton University, Princeton, NJ, 1971.

[21] W. Fernandez de la Vega, G. S. Lueker, Bin packing can be solved within $1+\varepsilon$ in linear time. *Combinatorica* **1** (4) (1981) 349–355.

[22] B. Xia, Z. Tan, Tighter bounds of the First Fit algorithm for the bin-packing problem. *Discrete Applied Mathematics* **158** (2010) 1668–1675.

[23] M. Yue, A simple proof of the inequality $FFD(L) \leq \frac{11}{9}OPT(L) + 1\ \forall L$, for the FFD bin-packing algorithm, *Acta Mathematicae Applicatae Sinica*, **7** (1991) 321–331.