

# NYILATKOZAT

Név: Domán Dániel Gergő

ELTE Természettudományi Kar, szak: Alkalmazott Matematikus MSc

NEPTUN azonosító: JCLPKS

Szakedolgozat címe:

Problémák és algoritmusok átlagos komplexitása

A **szakedolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2022.05.31.



---

*a hallgató aláírása*

EÖTVÖS LORÁND TUDOMÁNY EGYETEM  
TERMÉSZETTUDOMÁNYI KAR

Problémák, és algoritmusok átlagos komplexitása

Szakdolgozat

Domán Dániel Gergő  
Alkalmazott Matematikus MSc  
Számítástudomány specializáció



Témavezető:

Dr. Kis Tamás  
Operációkutatás tanszék  
Eötvös Loránd Tudományegyetem  
Természettudományi Kar

## Tartalomjegyzék

<b>1. Köszönetnyilvánítás</b>	<b>3</b>
<b>2. Bevezetés, motivációk</b>	<b>4</b>
<b>3. Ládapakolás</b>	<b>5</b>
3.1. Alsó korlát . . . . .	5
3.2. Best Fit . . . . .	8
<b>4. Igazságos csapatelosztás</b>	<b>16</b>
4.1. Elméleti megfontolások . . . . .	16
4.2. Szimuláció . . . . .	18
4.3. Eredmények . . . . .	19
<b>5. Random-3SAT nehézség</b>	<b>23</b>
5.1. Alapfogalmak . . . . .	23
5.2. 3 változós függvények . . . . .	24
5.3. Alkalmazás gráf problémákra . . . . .	26
5.3.1. Kiegyensúlyozott páros klikk feladat . . . . .	26
5.3.2. Minimális kettéosztás . . . . .	27
5.4. Közelítő algoritmus a minimális kettéosztás feladatra	28
5.4.1. A szimuláció eredményei . . . . .	29

## **1. Köszönetnyilvánítás**

Szeretnék köszönetet mondani Dr. Kis Tamás Tanár Úrnak, amiért elvállalta, hogy nála írhasam a szakdolgozatomat, hogy érdekes problémákat és cikkeket osztott meg velem, segített a feldolgozásukban, valamint teret adott a saját ötleteimnek és támogatta azok kivitelezését.

## 2. Bevezetés, motivációk

Egyetemi tanulmányaim során sok érdekes kérdésről, problémáról és algoritmusról tanultam. Az egyik volt közülük Christofides algoritmus a metrikus utazóügynök problémára. Erről megtanultuk, hogy  $\frac{3}{2}$ -approximáció, azonban már a bizonyítás során is többször az lehetett az érzése az embernek, hogy azok a becslések, melyekkel ez az approximációs hányadost kiszámoltuk, igen ritkán élesek. Láthattunk egy éles példát, amelyben tetszőlegesen közel mehetünk a  $\frac{3}{2}$ -hez, azonban ezzel kapcsolatban is lehetett az az érzésünk, hogy a "természetben előforduló" példák nem hasonlítanak erre. Innen származik az ötlet, hogy érdekes lenne bizonyos approximációs algoritmusok átlagos viselkedését megnézni véletlen inputon.

A szakdolgozatomban néhány ezzel kapcsolatos eredményt ismertetünk, kiegészítve néhány saját kérdésfelvetéssel, megoldási ötlettel, illetve szimulációval.

### 3. Ládapakolás

Ebben fejezetben online algoritmusokat fogunk vizsgálni a ládapakolás feladatra, azaz az algoritmus minden elemet azonnal bepakol, amint megkapja. Ezen felül feltesszük, hogy nem tudjuk pontosan, hogy hány elemet kell bepakolnunk, csak azt, hogy legfeljebb  $n$ -et. Azt vizsgáljuk, hogy a várható felesleges tár egy véletlen inputon mennyi. A fejezet P. W. Shor 1986-os cikke alapján készült [1].

A fejezetben gyakran használjuk azt a kifejezést, hogy "nagy valószínűséggel". Ez azt jelenti, hogy legalább  $1 - \frac{1}{n}$  valószínűséggel következik be az adott esemény. A ládapakolás feladatnál pedig a várható viselkedés elemzése során figyelmen kívül lehet hagyni bármit, ami legfeljebb  $\frac{1}{n}$  valószínűséggel következik be. Ez azért igaz, mert  $n$  elemet  $n$  ládába biztosan be lehet pakolni, ezért ha egy algoritmus  $O(f(n))$  ládát használ  $1 - \frac{1}{n}$  valószínűséggel,  $\frac{1}{n}$  valószínűséggel pedig akár  $O(n)$ -et, akkor a felhasznált ládák számának várható értéke

$$(1 - \frac{1}{n})O(f(n)) + \frac{1}{n}O(n) = O(f(n)).$$

Az algoritmusokat a felhasznált ládák és a felesleges tárhely alapján fogjuk elemezni, ahol a felesleges tárhely a ládáknak lévő kitöltetlen terek összege. Vagyis az összes felesleges tárhely az nem más, mint a ládák száma mínusz a súlyok összege. Mivel a súlyok várható értéke  $1/2$ , ezért

$$E(\text{ládák száma}) = \frac{n}{2} + E(\text{felesleges tárhely}).$$

#### 3.1. Alsó korlát

**3.1. Definíció.** Legyen az eloszlásunk a következő:

- i)  $k \in \{1, \dots, n\}$  random
- ii)  $U[0, 1]$ -ből  $k$  elem
- iii) a  $k$ -adik után STOP

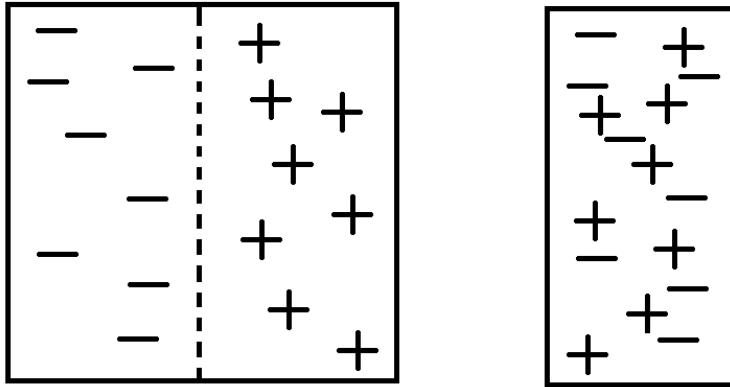
**3.1. Tétel.** Legyen  $A$  egy online algoritmus, amely  $k$  elemet kap, melyek  $U[0, 1]$ -ből származnak, illetve  $k$ -t nem ismerjük, amíg nem jutottunk el az utolsó elemig. Legyen  $k$  egy véletlen egész  $1$  és  $n$  között. Ekkor a felesleges tárhely  $\Omega(\sqrt{n} \log n)$

**Bizonyítás:** legyen  $L$  random lista, amely  $n$  elemet tartalmaz. Az mindegy, hogy egy random  $k$  hosszú listát, vagy egy  $n$  hosszú lista első  $k$  elemét pakoljuk.

Vezessük be a sík-párosítás fogalmát! Adott egy egységnyezet, és a lista minden elemnek megfeleltetünk egy pontot benne az alábbi módon: Az  $x$  koordináta legyen az elem mérete, az  $y$  pedig az, hogy hanyadiknak érkezett be: a  $j$ -ediként beérkező legyen a négyzet tetejétől  $\frac{j}{n}$ -re. Címkezzük meg őket: az  $\frac{1}{2}$ -nél kisebbek kapjanak (-) címkét, a nagyobbak (+)-t.

Tükrözzük rá az  $x = \frac{1}{2}$ -re a (+) pontokat. Az eredetileg  $\frac{1}{3}$  és  $\frac{2}{3}$  közé esők között párosítsuk a (+)-osokat (-)-osokkal a következő módon: egy (-) csak egy

tőle jobbra lévő (+) ponttal állhat párba, így az összegük 1-nél kisebb lesz. Ez azt jelenti, hogy egy ládába fognak kerülni.



1. ábra

Másképpen: Létrehozunk egy páros gráfot az egyik pontosztály az  $\frac{1}{3}$ -nál nagyobb, de  $\frac{1}{2}$ -nél kisebb (-) pontok, a másik az  $\frac{1}{2}$ -nél nagyobb, de  $\frac{2}{3}$ -nál kisebb (+) pontok. Két pont közt fusson él akkor, ha a a tükrözött félnégyszetben a (+) pont a (-)-tól jobbra esik, vagyis a hozzájuk tartozó elemek súlyának az összege 1-nél kisebb.

Az  $(\frac{1}{2}, 1)$ -ből egy elem megtölthet egy ládát, az  $(\frac{1}{3}, \frac{1}{2})$ -ből 2, illetve az  $(\frac{1}{3}, \frac{2}{3})$ -ből és a  $(\frac{2}{3}, 1)$ -ből nem kerülhet kettő 1 ládába. Így (+)-ok össz száma és a párosítatlan (-)-ok számának fele alsó korlát a felhasznált ládákra.

**3.1. Lemma.** *Adott  $n$  pont véletlenszerűen az egységnyégzetben, és minden pont azonos valószínűséggel (+) vagy (-). Legyen  $M$  egy olyan párosítás, amiben minden párosított (-) pontnak a (+) párja tőle jobbra esik, és legyen a párosításélek össz-szintkülönbsége  $\sum_e |y_{e_+} - y_{e_-}|$ . Ha  $M$  olyan, hogy az össz-szintkülönbség minimális, akkor ennek a várható értéke  $\Omega(\sqrt{n \log n})$ . A párosítatlan pontokat kössük össze a négyzet aljával.*

Legyen  $f(t) :=$  Az élek száma, amik átmetszik az  $y = t$  vízszintes egyenest. Az  $y$ -tengely az idő, és az  $y = t$ -t metsző élek száma annyi, mint a párosítatlan pontok száma a  $t$  időpillanatban. A párosításélek szintkülönbségeinek az összege megegyezik az  $f$   $[0, 1]$ -en vett  $t$ -szerinti integráljával. Tehát ez az integrál a párosítatlan pontok átlagos száma az idő szerint, vagyis  $k$  szerint. A lemmát alkalmazzuk az  $\frac{1}{3}$ -nál nagyobb  $x$ -koordinátájú pontokra, akkor kapunk egy  $\Omega(\sqrt{n \log n})$  alsó korlátot a párosítatlan pontok várható értékére. Ez pedig bizonyítja a tételt.

A 3.1-es lemma bizonyításához felhasználjuk Ajtai, Komlósi és Tusnády eredményét:

**3.2. Tétel.** (Ajtai, Komlósi, Tusnády): Adott  $n$  db (+) és ugyanennyi (-) pont az egységnégyzetben. Tekintsük azt a teljes párosítást a (+) és (-) pontok között, ahol a párosítások összhossza minimális. Ekkor ez az összhossz átlagosan  $\Theta(\sqrt{n \log n})$

A mi modellünk pár dologban különbözik ettől:

1) A mi modellünkben horizontálisan valóban véletlenszerű a pontok eloszlása (méret szerint), függőlegesen azonban egy másik eloszlásból valók (idő szerint)

2) A pontok száma: az AKT modellben megegyeznek, míg a miénkben csak azonos valószínűséggel lesznek (+) vagy (-) pontok

3) Az AKT modellben nem csak jobbra párosítjuk a (-) pontokat.

Lényegében azonban egyik se baj:

1) Az AKT modellben a pontok függőleges eloszlása egyenletes, a miénkben pedig egyenlő távolságban voltak, mielőtt az  $[\frac{1}{3}, \frac{2}{3}]$ -on kívülieket kitöröltük volna. Az egyenletes eloszlásból származó pontokat függőlegesen elmozgatjuk anélkül, hogy a sorrendjüket megváltoztatnánk addig, amíg nem kerülnek egyenlő távolságra egymástól. A mozgatás várható távolsága pontonként  $\Theta(\frac{1}{\sqrt{n}})$ , ha a mi pontjainkat is elmozgatjuk, akkor is  $\Theta(\frac{1}{\sqrt{n}})$ -t fognak várhatóan mozogni, vagyis az élek összhosszának várható változása  $O(\sqrt{n})$

2) Az általánosság megszorítása nélkül feltehető, hogy több (+) van, mint (-). Ekkor tegyünk be még véletlenszerűen (-) pontokat úgy, hogy ugyanannyian legyenek a (+)-ok és (-)-ok. Várhatóan  $\Theta(\sqrt{n})$  pontot kell hozzáadni. Egy egységnégyzetben egy él hossza legfeljebb  $\sqrt{2}$ , így az élek összhosszának változása  $O(\sqrt{n})$ .

3) Ez sem jelent problémát, azonban kicsit bonyolultabb ezt belátni. Tekintsük a duális feladatot: Az AKT bizonyítás egyik kulcslépése, hogy találunk egy  $w$  súlyfüggvényt, ami alsó korlát minden párosításra. Az ő súlyfüggvényükből fogunk konstruálni egy  $w'$  súlyfüggvényt, ami a jobboldali párosításokra vonatkozik.

Először definiáljuk a  $w$  pályáját egy  $P_1$  és egy  $P_2$  pont között:  $\frac{|w(P_1) - w(P_2)|}{d(P_1, P_2)}$ , ahol  $d(P_1, P_2)$  a pontok euklideszi távolsága.

$w : [0, 1]^2 \rightarrow [-1, 1]$

1)  $w$  pályája legfeljebb 1.

2)  $\Sigma w(P_+) - \Sigma w(P_-)$  várható értéke  $\Omega(\sqrt{n \log n})$

3) A négyzet határán  $w = 0$

Ezekből a következő módon látszik az alsó korlát:

$$\begin{aligned} \Sigma_{e \in M} d(e) &\geq \\ \Sigma_{e \in M} w(e_+) - \Sigma_{e \in M} w(e_-) &\geq \\ \Sigma w(P_+) - \Sigma w(P_-) &\geq \\ \Omega(\sqrt{n \log n}) & \end{aligned}$$



Most definiáljunk egy  $w'$  súlyfüggvényt a mi esetünkre:

$$w : [0, 1]^2 \rightarrow [0, 1]$$

- 1)  $w$  pályája legfeljebb 1.
- 2)  $\Sigma w(P_+) - \Sigma w(P_-)$  várható értéke  $\Omega(\sqrt{n \log n})$
- 3)  $w$  csökken minden vízszintes vonalon
- 4)  $w = 0$ , ha  $y = 0$

Ahhoz, hogy találjunk egy ilyen  $w'$ -t, felhasználjuk az AKT tételben szereplő  $w$ -t a középső harmadban. Ennek a tulajdonságai a következők:

$$w : [0, 1] \times [\frac{1}{3}, \frac{2}{3}] \rightarrow [-\frac{1}{6}, \frac{1}{6}]$$

- 1)  $w$  pályája legfeljebb  $\frac{1}{6}$ .
- 2)  $\Sigma w(P_+) - \Sigma w(P_-)$  várható értéke  $\Omega(\sqrt{n \log n})$
- 3) A négyzet határán  $w = 0$

Ekkor legyen  $w'$  :

$$\begin{aligned} & \frac{1}{2}(1-x)y, \text{ ha } 0 \leq y \leq \frac{1}{3} \\ & w(x, y) + \frac{1}{6}(1-x), \text{ ha } \frac{1}{3} \leq y \leq \frac{2}{3} \\ & \frac{1}{2}(1-x)(1-y), \text{ ha } \frac{2}{3} \leq y \leq 1 \end{aligned}$$

Ez a  $w'$  függvény teljesíti az 1), 2), 3), 4) pontokat.

Tegyük fel, hogy  $e = (e_+, e_-) = ((x_+, y_+), (x_-, y_-))$  él a jobboldali párosításban, vagyis  $x_- \leq x_+$ . Legyen  $c := (x_-, y_+)$

Ekkor  $w'(e_+) - w'(e_-) \leq w'(c) - w'(e_-) \leq d(c, e_-) = |y_+ - y_-|$ . Ha ezt minden élre megcsináljuk, akkor pont a 3.1-es lemma állítását kapjuk.

Megjegyzés: Ha előre ismerjük  $n$ -t, akkor a tétel nem igaz. Például az első  $\frac{n}{2}$  elemet 1-1 ládába tesszük, majd rendezzük őket. Ezután a FirstFit algorit-mussal bepakoljuk a többbit. Ekkor a felesleges tár  $\Theta(\sqrt{n})$ .

## 3.2. Best Fit

A Best Fit algoritmus során minden elem a legtelítettebb ládába kerül, amelybe belefér.

**3.3. Tétel.** *A várható felesleges tár:  $\Theta(n^{\frac{1}{2}}(\log n)^{\frac{1}{4}})$*

A MatchingBestFit legyen ugyanaz az algoritmus, mint a BestFit, kivéve, hogy ha egy ládába két elem kerül, akkor az első egy  $1/2$  vagy annál nagyobb méretű elem volt, a második pedig egy legfeljebb  $1/2$  méretű elem.

Legyen  $L$  egy lista, amit az  $A$  algoritmus bepakol egy  $A(L)$  pakolásba, ami  $\#A(L)$  ládából áll.

**3.2. Lemma.**  $L'$  legyen az a lista, amit  $L$ -ből kapunk valamelyik elem eltávolításával. Ekkor  $\#MBF(L) \geq \#MBF(L') \geq \#MBF(L) - 1$

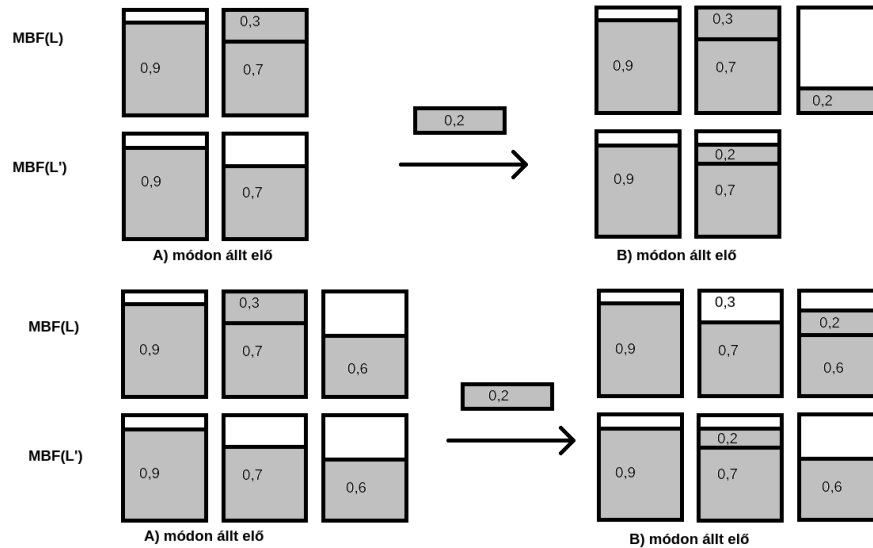
**Bizonyítás:** Azt a ládát, amely két elemet tartalmaz, nevezzük teli ládának. Azt állítjuk, hogy a következő két esemény egyike fennáll:

A)  $MBF(L)$  előáll  $MBF(L')$ -ből úgy, hogy egy egyelemű ládát egy teli ládával helyettesítünk.

B)  $MBF(L)$  előáll  $MBF(L')$ -ből úgy, hogy egy új ládát nyitunk.

Be akarjuk látni indukcióval, hogy ha egy  $p$  elemet hozzá akarunk venni a listához, amelyből az  $MBF(L)$  és  $MBF(L')$  A) vagy B) módon áll elő, akkor ezután is A) vagy B) módon fog előállni.

Tegyük fel, hogy A)-t teljesítik. Ekkor létezik egy  $b$  láda, ami  $MBF(L')$ -ben egyelemű,  $MBF(L)$ -ben teli. Ha a  $p$  elem nem a  $b$  ládába kerül, akkor A) továbbra is teljesül, hiszen  $p$  mindkét esetben ugyanabba a ládába fog kerülni. Ha pedig  $b$ -be kerül  $MBF(L')$ -ben, akkor  $MBF(L)$ -ben vagy egy üres ládába kerül (B), vagy egy másik 1-eleműbe (A). Ha eredetileg a B) állt fenn, azt hasonlóan be lehet látni.



2. ábra

Még az indukció kezdeti lépését kell megnéznünk: Egy darabig a két csomag megegyezik, majd érkezik egy  $q$  elem, ami csak az  $L$  listán szerepel. Ha ennek bepakolásához új ládát kell nyitni  $MBF(L)$ -ben, akkor a B) áll fenn, ha egy egyelemű telik meg, akkor az A).

**3.3. Lemma.**  $\#MBF(L) \geq \#BF(L)$ .

**Bizonyítás:** Legyen  $L'$  az a lista, amit  $L$ -ből úgy kapunk, hogy eltávolítjuk azokat az elemeket, amiket  $BF$  egy  $1/2$ -nél kisebb nemüres ládába pakolt, vagy egy  $MBF$  által telinek tekintett ládába kerültek. Ekkor  $MBF(L')$  ugyanúgy pakolja a ládákat, ahogyan  $BF(L)$  az  $L'$  elemeit, az  $L - L'$  elemei pedig  $MBF(L')$  által telített ládába kerültek, ezért  $\#MBF(L') = \#BF(L)$ . Alkalmazhatjuk az előző lemmát (akár többször egymás után), mivel  $L'$  úgy állt elő  $L$ -ből, hogy elemeket töröltünk belőle, ekkor pedig  $\#MBF(L) \geq \#BF(L)$ .

A tétel bizonyításához vezessük be a jobbfelső síkpárosítás problémáját [3]. Egy egységnyezetben adott ugyanannyi (-) és (+) pont egyenletes eloszlásból. Egy (-) pontnak egy olyan (+) pont lehet a párja, amely tőle jobbra és felfele helyezkedik el.

Ezt a korábbihoz hasonló módon fogjuk átalakítani a ládapakolás feladatra. Az első koordináta legyen a méret, a második pedig a beérkezés ideje. Ezután hajtsuk félbe a négyzetet az  $x = \frac{1}{2}$ -nél, vagyis az  $\frac{1}{2}$ -nél nagyobb  $x$ -koordinátájú pontokat rátükörözzük az  $x = \frac{1}{2}$ -re. Minden olyan  $MBF$  ládához, amely 2 elemet tartalmaz, behúzzunk egy élt, így kapjuk az  $M$  párosítást.

Azt állítjuk, hogy az  $M$  ténylegesen jobbfelső síkpárosítás lesz, ráadásul maximális. Ez azért igaz, mert minden (+) a (-) párja fölött lesz, mert amikor két elem egy ládába kerül  $MBF$  alapján, akkor az első  $\frac{1}{2}$ -nél nagyobb (azaz (+)), a másik pedig kisebb, és azért lesz a (-)-tól jobbra a (+), mert az összegük 1-nél kisebb. Ezen felül az  $MBF$  a (-) pontokat fentről lefele párosítja és mindig a legbaloldali elérhető (+)-szal rakja párba. Bizonyított [4], hogy ez az algoritmus megtalálja az optimális jobbfelső párosítást.

A párosítatlan pontok maximális számának várható értékére a következő korlát ismert [5]:

$$O(n^{\frac{1}{2}}(\log n)^{\frac{1}{4}}).$$

Az általunk használt ládák száma

$$\frac{n}{2}O(n^{\frac{1}{2}}(\log n)^{\frac{1}{4}}),$$

azaz a felesleges tár

$$O(n^{\frac{1}{2}}(\log n)^{\frac{1}{4}}).$$

Térjünk rá az alsó korlátra. Megmutatjuk, hogy gyakran egy elsőként bekerülő nagy elemre fog rákerülni másodikként egy kicsi. Ez ad egy jobb-felső síkpárosítást, ahol a párosítatlan pontok várható értéke  $\Omega(n^{\frac{1}{2}}(\log n)^{\frac{1}{4}})$ .

Nevezzük az első elemet, amit berakunk egy ládába a láda aljának, az utolsót a tetejének, a bepakolt elemek számát pedig a láda magasságának. Most azt nézzük, hogy az  $\frac{1}{3}$  és  $\frac{2}{3}$  közötti elemekkel mi fog történni. Az  $\frac{1}{3}$  és  $\frac{1}{2}$  közötti

elemeket nevezzük  $s$  elemeknek (small, kicsi), a  $\frac{1}{2}$  és  $\frac{2}{3}$  közötti elemeket pedig  $b$ -nek (big, nagy).

**3.4. Lemma.** *Annak a valószínűsége, hogy a következő láda  $(b, s)$  lesz, legfeljebb annyi, mint hogy  $(s, s)$ .*

**Bizonyítás:** Egy  $(b, s)$ -t akkor kaphatunk, ha létezik egy  $s$  elem egy  $B$  ládában, ami legfeljebb az  $\alpha \leq \frac{1}{2}$  része van megtöltve. (Lehet, hogy  $s$  kisebb  $\alpha$ -nál. Ha a következő elem súlya  $\alpha$  és  $1 - \alpha$  közé esik, akkor az a  $B$ -be fog kerülni. Sőt, ez az egyetlen nemüres láda, aminek a súlya legfeljebb  $1 - \alpha$ . Így a valószínűsége annak, hogy egy  $b$  elem a  $B$  ládába kerül, az ugyanannyi, mint annak a valószínűsége, hogy egy  $b$  elem mérete  $\alpha$  és  $1 - \alpha$  közé esik. Ez ugyanannyi, mint annak az esélye, hogy egy  $s$  elem  $\alpha$  és  $\frac{1}{2}$  közé esik, ami legfeljebb annyi, mint, hogy egy  $s$  elem a  $B$ -be kerül.

Az alsó korlát bizonyítása: 5 féle láda létezik:  $(s, s), (b, s), (s, b), (s, ), (b, )$ , illetve lehetnek  $\frac{1}{3}$ -nál kisebb elemek, de mivel alsó korlátot bizonyítunk, ezért ezeket nyugodtan kihagyhatjuk az elemzésből. Egy elem  $\frac{1}{6}$  eséllyel lesz akár  $s$ , akár  $b$ . A binomiális eloszlásra vonatkozó korlátok miatt legalább  $1 - \frac{1}{n}$  valószínűséggel az  $s$  és  $b$  elemek számának különbsége  $O(\sqrt{n \log n})$ . Ebből következik, hogy  $|(b, )| = 2|(s, s)| + |(s, )| \pm O(\sqrt{n \log n})$ .

Az előző lemmából következik, hogy legalább  $1 - \frac{1}{n}$  valószínűséggel  $|(b, s)| \leq |(s, s)| + O(\sqrt{n \log n})$ . Mivel egy elem  $\frac{1}{6}$  eséllyel lesz akár  $s$ , akár  $b$ , ezért legalább  $1 - \frac{1}{n}$  valószínűséggel lesz  $\frac{n}{3} \pm O(\sqrt{n \log n})$  darab  $s$  és  $b$  elem. Az  $(s, b)$  ládák megadnak egy jobb-felső párosítást az  $s$  és  $b$  ládák közt. Az  $s$  és  $b$  elemek véletlenszerűen egyenletesen vannak kiválasztva a  $[\frac{1}{3}, \frac{2}{2}]$  intervallumból, és ahogyan azt bizonyítani fogjuk a következő szakaszban, egy jobbfelső párosításnak nagy valószínűséggel  $\Omega(\sqrt{n}(\log n)^{\frac{3}{4}})$  fedetlen pontja van.

$$\text{Ebből } |(s, b)| = \frac{n}{6} - \Omega(\sqrt{n}(\log n)^{\frac{3}{4}})$$

Tudjuk, hogy a  $b$  elemek száma  $\frac{n}{6} \pm O(\sqrt{n \log n})$ , vagyis  $|(s, b)| + |(b, )| + |(b, s)| = \frac{n}{6} \pm O(\sqrt{n \log n})$ , mivel két  $b$  elem nem kerülhet egy ládába.

$$\text{Az előzőekből } |(b, )| + |(b, s)| = \frac{n}{6} \pm O(\sqrt{n \log n}).$$

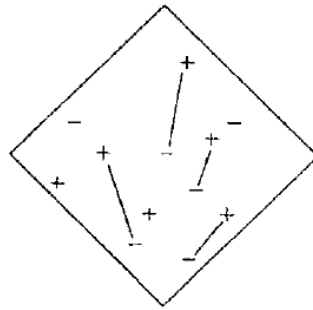
Használjuk az egyenlőtlenségeket, amik  $|(b, )|$ -re és  $|(b, s)|$ -re vonatkoznak:

$3|(s, s)| + |(s, )| = \frac{n}{6} \pm O(\sqrt{n \log n})$ . Tehát ennyi olyan ládánk lesz, ami nem tartalmaz  $\frac{1}{2}$ -nél nagyobb elemet. Mivel az  $\frac{1}{2}$ -nél nagyobb elemek száma nagy valószínűséggel  $\frac{n}{2} \pm O(\sqrt{n \log n})$ , ezért látjuk, hogy a felhasznált ládák száma  $\frac{n}{2} + \Omega(\sqrt{n}(\log n)^{\frac{3}{4}})$ .

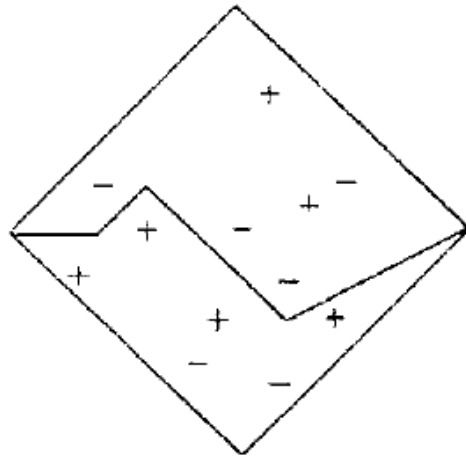
Nézzük az alsó korlátot:

**3.4. Tétel.** Adott  $n$  pont az egységnégyzetben véletlenszerűen egyenletes eloszlásból, és mindegyik azonos valószínűséggel lesz (+) vagy (-). Egy maximális jobb felső párosításban a párosítatlan pontok száma  $\Omega(\sqrt{n}(\log n)^{\frac{3}{4}})$ .

**Bizonyítás:** Forgassuk el a négyzetet 45 fokkal, ekkor egy (-) párja biztosan fölötte lesz és az él meredeksége 1-nél nagyobb, vagy -1-nél kisebb lesz, lásd: 3. ábra. Ahhoz, hogy a párosítatlan pontok számára tudjunk adni egy  $k$  alsó korlátot, vágjuk ketté a négyzetet úgy, hogy alul  $k$ -val több (+) legyen, mint (-), a határ a bal és jobb csúcs között fusson és a meredeksége essen mindig -1 és +1 közé, ekkor egy alsó (+) biztosan nem lesz párosítva egy felső (-) ponttal. Most konstruálunk egy olyan határvonalat, aminél az első részbe mindig legalább  $\Omega(\sqrt{n}(\log n)^{\frac{3}{4}})$ -del több (+) esik, mint (-), lásd: 4. ábra.

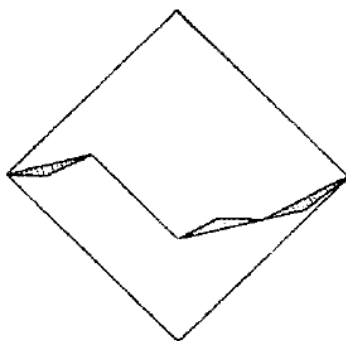


3. ábra

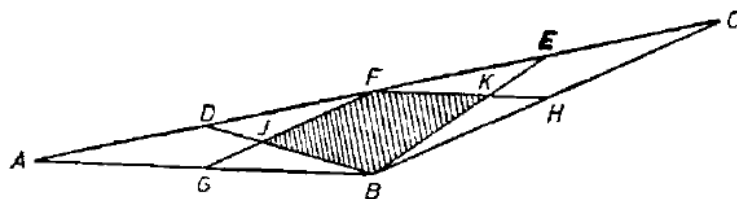


4. ábra

Szintenként konstruáljuk meg a határvonalat. Mindig lesznek benne szakaszok és háromszögek. Ha egy háromszög a határon van, akkor minden későbbi szinten a határ két csúcán át fog menni, és a határ ezen két csúc közötti része a háromszögön belül lesz. Például az 5. ábrán a határ a sötétített részben lesz. Ahhoz, hogy megkapjuk a következő szintre a határvonalat vagy egy egyenes szakasszal, vagy két háromszöggel, amiknek a területe az eredeti területének negyede lesz, mint a 6. ábrán. Ezt addig ismételjük, ameddig nem lesznek a háromszögek olyan kicsit, hogy átlagban csak 1-1 pontot tartalmazzanak. Olyan háromszöget, aminek valamelyik oldala 1 meredekségű, nem finomítunk tovább. Később látni fogjuk, hogy ez nem befolyásolja az elemzést, mert ekkor már a határvonalnak csak nagyon kicsi részén módosítana.



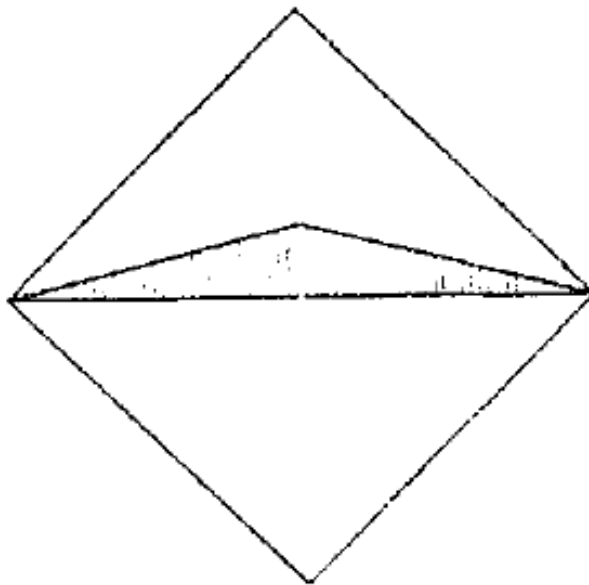
5. ábra



6. ábra

A 6. ábrán látható egy általános lépés egy háromszög finomítására. A  $G$  és  $H$  pontok felezéspontjai az  $AB$ -nek, illetve a  $BC$ -nek, a  $D$ ,  $E$ ,  $F$  negyedelik az  $AC$ -t. Egy lépésben a háromszöget helyettesíthetjük az  $ADB$  és  $BEC$  háromszögekkel vagy az  $AGF$  és  $FHC$  háromszögekkel. Hogy melyik helyettesítést válasszuk, ahhoz nézzük a középső, sötétített központi négyszöget. Ezt a határ alá rakjuk, ha több (+) van benne, mint (-), különben pedig fölé.

A kezdő lépés egy nagy háromszög lesz, az oldalainak meredeksége  $\pm(\log n)^{-\frac{1}{2}}$ , lásd: 7. ábra.



7. ábra

Tulajdonságok:

- 1)  $A, B, C$  az  $x$  koordinátákat tekintve olyanok, hogy  $B$  felezi az  $AC$ -t, vagyis  $BF$  függőleges.
- 2) Minden háromszög területe az előzőnek a negyede.
- 3) A központi négyszög területe a háromszög területének a harmada.
- 4) Ha egy háromszög oldalainak meredeksége  $(i-1)s, is, (i+1)s$ , akkor a két új háromszög vagy hasonló lesz hozzá, és ekkor az új oldalak meredekségei meg fognak egyezni az előzővel, vagy az új meredekségek  $(i-2)s, (i-1)s, is$  és  $is, (i+1)s, (i+2)s$  lesznek. Legutóbbi indukcióval következik a kezdőlépésből és abból, hogy  $m(AD) + m(DB) = 2m(AB)$ .

Ha a kezdeti lépés a 0. szint, akkor az  $i$ -edik szinten  $2^i$  háromszöget tesztlünk, mindegyiknek a területe  $\frac{1}{6} \frac{s}{4^i}$ . Egy  $A$  területű részbe a (+) és (-) pontok számának várható különbsége  $\Theta(\sqrt{nA})$ . Ez azért igaz, mert egy  $A$  területű részbe átlagosan  $nA$  pont esik, mindegyik azonos valószínűséggel lesz (+) vagy (-).

Ezért minden szinten várhatóan  $\Theta(\sqrt{ns})$  (+) pontot adunk az alsó szinthez.  $\Theta(\log n)$  lépés után várhatóan  $\Theta(\log n \sqrt{ns}) = \Theta(\sqrt{n}(\log n)^{\frac{3}{4}})$ -nel több (+) lesz

alul, mint (-). Még azt kell megmutatnunk, hogy a háromszögek meredeksége  $-1$  és  $1$  között marad. Ha minden háromszöget finomítunk, akkor ez nem lesz így. Ezért, ha bármely háromszög bármely oldala eléri az  $1$ -et vagy a  $-1$ -et, akkor nem finomítjuk tovább. Azt kell megmutatnunk, hogy érdemben ez nem befolyásolja az elemzést.

A háromszögekből készíthetünk egy bináris fát, minden háromszögnek a gyereke a következő szinten lévő finomításai. A 4. tulajdonság alapján a gyerekek meredeksége a szülőkéétől  $s$ -sel,  $0$ -val, vagy  $-s$ -sel különbözik, ráadásul ha az egyik  $-s$ -sel, akkor a másik  $+s$ -sel. Ebből adódik egy sztochasztikus folyamat egy olyan fán, aminek a gyökeréből indulunk és minden gyereket  $\frac{1}{2}$  valószínűséggel választjuk. Minden esetben a döntés után hozzáadjuk és kivonjuk ugyanazt az összeget, ami  $0$  vagy  $s$ . Ez a folyamat egy martingál, aminek a varianciája minden lépésben legfeljebb  $s^2$ . Ha a fának a mélysége legfeljebb  $c \log n$ , a leveleken lévő értékek varianciája  $s^2 c \log n$ . Ha ez az érték  $\frac{1}{4}$ , akkor egy véletlenszerűen kiválasztott levélen az érték varianciája legfeljebb  $\frac{1}{4}$ . Mivel a várható érték  $0$ , ezért a levelek legfeljebb  $\frac{1}{4}$  részének a meredeksége lesz  $1$ -nél nagyobb.

Megmutattuk, hogy  $\log n$  lépés után a háromszögeknek legfeljebb a negyed részét nem finomítjuk tovább. A maradék  $\frac{3}{4}$  része a háromszögeknek még mindig fog adni várhatóan  $O(\sqrt{n}(\log n)^{\frac{-1}{4}})$  (+) pont többletet, mivel a (+) pontok többletének az eloszlása egy  $nA$  pontot tartalmazó háromszögben binomiális, és ha ennek eltávolítjuk a felső negyedét, akkor a várható többlete a (+) pontoknak még mindig  $\Theta(\sqrt{nA})$ .



## 4. Igazságos csapatelosztás

Képzeljünk el egy testnevelés órát, ahol két focicsapatba kell beosztani a diákokat úgy, hogy a csapatok létszáma megegyezzen, és cél, hogy a két csapat nagyjából azonos képességű legyen. A következő fejezetben ezt a problémát fogjuk körbe járni, néhány elméleti megfontolás után pedig bemutatunk egy szimulációt, és az eredményeit.

### 4.1. Elméleti megfontolások

**4.1. Definíció.** Legyen  $a$  és  $b$  pozitív valós számok, ekkor az abszolút hányadosuk  $\text{absq}(a, b) := \min\{\frac{a}{b}, \frac{b}{a}\}$

**4.2. Definíció.** *Igazságos csapatelosztás feladat:*

Adott  $2n$  egész vagy valós szám az  $[a, b]$  intervallumból (általában a  $[0, 1]$ -beli valósakat fogjuk nézni), amiket egy  $A$  és egy  $B$  halmazba akarunk tenni a következő módon:  $|A| = |B| = n$ ,  $A^* := \sum_{a \in A} a$ ,  $B^* := \sum_{b \in B} b$ , és maximalizálni szeretnénk az  $\text{absq}(A^*, B^*)$ -ot.

Ezt úgy érdemes elképzelni, hogy adott páros sok ember, akiknek a képességeiket egy-egy számmal tudjuk jellemezni egy rögzített intervallumból, célunk pedig két olyan egyenlő méretű csapatba osztani, hogy a képességek összege a két csapatban nagyjából megegyezzen, konkrétan a hányadosuk legyen minél közelebb az 1-hez.

**4.1. Állítás.**  $A \max\{\text{absq}(A^*, B^*)\}$  ugyanott vétetik fel, mint a  $\min|A^* - B^*|$ .

Ez annak a következménye, hogy az  $A^* + B^*$  állandó.

**Megjegyzés:** ez a feladat NP-nehéz, hiszen visszavezethető rá az eldöntési változat, azaz be lehet-e osztani az embereket két egyenlő méretű és összképességű csapatba. Ez viszont NP-teljes, hiszen nyilván NP-ben van és visszavezethető rá a partíciós feladat (ott nem követeljük meg a két halmaztól az azonos elemszámot), ami pedig szintén NP-teljes. Ebből következően nem várható olyan algoritmus, ami polinom időben kiszámolja az optimumot, ezért közelíteni szeretnénk, illetve azt vizsgálni, hogy egy véletlen inputon egy közelítő algoritmus milyen hatékony.

A modellünkben generálunk  $2n$  véletlen számot a  $U[0, 1]$ -ből, ezeket akarjuk beosztani két csapatba.

**4.3. Definíció.** *Egy csapatelosztás várhatóan igazságos, ha  $E(A^*) = E(B^*)$ .*

Megjegyzés: Ha egy csapatelosztás várhatóan igazságos, az még nem biztos, hogy a feladat szempontjából célravezető, mivel például ha teljesen véletlenszerűen osztjuk el a csapatokat, az az elosztás is várhatóan igazságos lesz.

Ötlet: állítsuk nagyság szerinti (pl. növekvő) sorrendbe a számokat, és a rangsorban elfoglalt helyezések alapján osszuk be őket két halmazba.

A következő állításnál  $N := 2n$  az elemek száma.

**4.2. Állítás.** *Adott  $N$  véletlen szám  $U[0, 1]$ -ből. Ekkor a nagyság szerinti  $k$ -adik legkisebb elem várható értéke  $\frac{k}{N+1}$ .*

**Bizonyítás:** Ismert [6], hogy egy  $N$  elemű rendezett minta  $k$ -adik elemének az eloszlása  $\beta(k, N - k + 1)$ . Mivel egy  $\beta(a, b)$  eloszlású valószínűségi változó várható értéke  $\frac{a}{a+b}$ , ebbe helyettesítve megkapjuk a keresett állítást.

Az egyszerűség kedvéért a következő megfontolásokhoz minden értéket szorozunk fel  $N + 1$ -gyel, így a várható értékek nagyság szerint rendre  $1, 2, \dots, N = 2n$  lesznek.

**4.4. Definíció.** *Nevezzünk egy csapatelosztást helyezés szerinti csapatelosztásnak, ha az értékek szétosztásakor egyedül azt vesszük figyelembe, hogy melyik nagyság szerint hanyadik.*

**4.5. Definíció.** *Tegyük fel, hogy az értékek csökkenő sorrendben álnak. Nevezzük minden  $i$ -re 4-es blokknak a  $4i, 4i + 1, 4i + 2, 4i + 3$  indexű elemeket ( $0$ -tól indexelünk)*

**Közelítő algoritmus:** Rendezzük csökkenő sorrendbe az értékeket. A legnagyobb értéket tegyük az  $A$  csapatba, majd a következő kettőt a  $B$ -be, majd kettőt megint az  $A$ -ba, kettőt a  $B$ -be, és egészen addig kettesével váltakozunk, amíg el nem fogynak az értékek.

**4.3. Állítás.** *Ha  $n$  páros, azaz  $2n$  4-gyel osztható, akkor van helyezés szerinti várhatóan igazságos csapatelosztás.*

**Bizonyítás:** Ebben az esetben az előző algoritmus során az utolsó elem az  $A$ -ba fog kerülni. Ekkor minden 4-es blokk várhatóan igazságosan van elosztva, tehát a csapatelosztás is várhatóan igazságos lesz.

**4.4. Állítás.** *Ha  $n$  páratlan, akkor nincs helyezés szerinti várhatóan igazságos csapatelosztás.*

**Bizonyítás:** Ekkor az  $1, 2, \dots, 4k + 2$  számokat kéne két olyan csoportra osztani, amik összege azonos, ez azonban nem lehetséges, mivel az összegük páratlan.

**4.5. Állítás.** *Ha  $n$  páros, akkor exponenciálisan sok várhatóan igazságos csapatelosztás van.*

**Bizonyítás:** Egy 4-es blokkot 2 féle módon lehet várhatóan igazságosan elosztani: az  $A$ -ba kerül az első és az utolsó, a  $B$ -be a két középső, vagy pont fordítva. Ha az elemek számát megnövelem 4-gyel, akkor a 4-es blokkok száma

1-gyel nő. Minden blokkot egymástól függetlenül eloszthatunk kétféleképpen. Ekkor  $k$  blokk esetén  $2^k$  várhatóan igazságos elosztás van, amik 4-es blokkonként igazságosak, ami természetesen alsó becslés az összes várhatóan igazságos elosztásra. (Ha nem tekintünk különbözőnek két olyan elosztást, ahol az  $A$  és a  $B$  fel van cserélve, akkor az elosztások száma feleződik, de ettől még exponenciális.)

## 4.2. Szimuláció

Idézzük fel az algoritmust: Az elemeket 0-tól indexelve a  $4k$  és  $4k + 3$  indexű elemeket fogjuk az  $A$  csapatba tenni, a többi a  $B$ -be. Már láttuk, hogy ez páros  $n$ -re várhatóan igazságos. Célunk megvizsgálni ezt az algoritmust abból a szempontból, hogy mennyire van távol az optimumtól, azaz mennyi az

$$\text{absq}(\text{absq}(A_{apx}, B_{apx}), \text{absq}(A_{opt}, B_{opt}))$$

. Ennek a minimuma az approximációs hányados. Mi azonban az átlagos viselkedésre vagyunk kíváncsiak, így ezt fogjuk szimulálni egy véletlen mintán.

Megjegyzés: az előző képlet lényegében

$$\frac{\text{absq}(A_{apx}, B_{apx})}{\text{absq}(A_{opt}, B_{opt})}$$

mivel az optimális csapatelosztás abszolút hányadosa biztosan legalább akkora, mint a közelítő algoritmus által készített csapatelosztásé.

Megjegyzés: Amikor  $n \leq 2$ , azaz 2 vagy 4 emberünk van, akkor az algoritmusunk optimális.

A szimuláció elemzéséhez természetesen szükségünk van az optimum kiszámítására is. Erre az egészértékű programozást hívjuk segítségül. Legyen  $R$  az a  $2n$  elemű vektor, ami a véletlen számokat tartalmazza. A  $\mathbf{1}x$  az  $x$  vektor skaláris szorzata az azonosan 1 vektorral, vagyis az elemeinek összege.

$$\begin{aligned} \min c \\ Rx - R(\mathbf{1} - \mathbf{x}) &\geq -\mathbf{c} \\ Rx - R(\mathbf{1} - \mathbf{x}) &\leq \mathbf{c} \\ \mathbf{1}x &= n \\ x &\in \{0, 1\}^{2n} \\ c &\geq 0. \end{aligned}$$

Magyarázat: A 4.1-es állítás miatt elegendő a különbség abszolútértékét minimalizálni. Ez a minimumérték lesz  $c$ . Minden véletlen számhoz hozzárendelünk egy  $x_i$  bináris változót. Ezekből a változókból álló  $x$  vektor megmondja, hogy melyik elem melyik csapatba kerül:  $x_i = 0$  esetén az  $i$ . elem az  $A$  csapatba kerül, különben a  $B$ -be. At első három sora a programnak előírja, hogy a kifejezés

abszolútértékét minimalizáljuk, vagyis "beszorítjuk"  $c$  és  $-c$  közé, és a nemnegatív  $c$ -t minimalizáljuk. A bináris változók összege legyen  $n$ , ezzel garantáljuk, hogy mindkét csapatba  $n$  elem kerül.

A szimuláció során megvizsgáltuk a véletlenszerű elosztást is, azaz amikor az első  $n$  véletlen szám kerül az  $A$  csapatba és a többi a  $B$ -be. Az eredményeket összehasonlítottuk, az így kapott csapatokat  $A_{rd}$ -vel és  $B_{rd}$ -vel jelöljük.

### 4.3. Eredmények

Magyarázat az első táblázathoz:

*Optimum:*  $\text{absq}(A_{opt}, B_{opt})$

*Random hányados:*  $\text{absq}(\text{absq}(A_{rd}, B_{rd}), \text{absq}(A_{opt}, B_{opt}))$

*Approximációs hányados:*  $\text{absq}(\text{absq}(A_{apx}, B_{apx}), \text{absq}(A_{opt}, B_{opt}))$

$2n$	Optimum	Random hányados	Approximációs hányados
6	0.9801980198019802	0.7696260854155591	0.9801980198019802
6	0.9444444444444444	0.8788927335640139	0.8666666666666667
6	0.9807692307692307	0.4256161180068358	0.8392857142857143
6	0.9935064935064936	0.43741982774418176	0.906832298136646
6	0.9727272727272728	0.4300287093030358	0.9035087719298246
8	0.9794871794871794	0.7383601346297681	0.84688995215311
8	0.9914163090128756	0.6392902871776112	0.9094650205761317
8	0.9679144385026738	0.5778630957954864	0.9166666666666666
8	0.976027397260274	0.9073730076826052	0.9169435215946844
8	0.9850746268656716	0.8775842537524782	0.9752475247524752
10	0.9965753424657534	0.722237427647518	0.8928571428571429
10	0.9968051118210862	0.6959905496490862	0.9904458598726115
10	0.9904306220095693	0.9904301817345296	0.9439252336448598
10	1.0	0.7204301075268817	0.7582417582417582
10	0.9902912621359223	0.943118756936737	0.9248826291079812
12	0.9968051118210862	0.5186473114264376	0.9409937888198758
12	1.0	0.9506172839506173	0.9832635983263598
12	0.9951923076923077	0.8485238862050456	0.9483568075117371
12	1.0	0.8925373134328358	0.98125
12	0.9965870307167235	0.7971993444827297	0.8690095846645367
14	0.9973262032085561	0.8065042545751253	0.9302325581395349
14	0.9967320261437909	0.74815925058548	0.9583333333333334
14	1.0	0.8141025641025641	0.9929577464788732
14	1.0	0.9506172839506173	0.9586776859504132
14	1.0	0.6821192052980133	0.9291139240506329

$2n$	Optimum	Random hányados	Approximációs hányados
16	1.0	0.914572864321608	0.9291139240506329
16	1.0	0.6857142857142857	0.9666666666666667
16	1.0	0.7432432432432432	0.9496221662468514
16	1.0	0.9122401847575058	0.9391100702576113
16	1.0	0.875	0.9819819819819819
18	1.0	0.9593345656192237	0.9272727272727272
18	0.9979838709677419	0.9919597582248185	0.9780439121756487
18	1.0	0.8693284936479129	0.9656488549618321
18	0.998	0.996003984015936	0.9549902152641878
18	0.997716894977169	0.8636739860752713	0.9886363636363636
20	1.0	0.77737881508079	0.9721115537848606
20	1.0	0.8054474708171206	0.9619450317124736
20	0.9981684981684982	0.8891717723246881	0.9836363636363636
20	0.9979338842975206	0.8401388659282526	0.9897119341563786
20	1.0	0.6958393113342898	0.9865546218487395

Képzeljük el, hogy testnevelés órára egy helyettesítő tanár érkezik, aki nincs tisztában a diákok képességeivel, viszont gyorsan be akarja őket osztani két egyenlő létszámú csapatba. Ekkor jobb híján véletlenszerűen választja ki a két csapatot. Azt fogjuk vizsgálni, hogy az eddigi feltételek mellett ez mennyiben fog eltérni az általunk adott módszertől.

A második táblázatban a csapatok közti különbség négyzetének átlagát mérjük le véletlenszerűen beosztott illetve az algoritmus által beosztott csapatoknál is. Mivel az optimumot nem számítjuk ki, ez lehetőséget ad nagy számú mérésre is, esetünkben arra, hogy az elemek számát akár 100-ig is elvigyük és minden páros elemszámra 1000 mérést átlagoljunk Ezek a mérések találhatóak a második táblázatban.

$2n$	$\frac{\sum_0^{1000}(A_{rd}-B_{rd})^2}{1000}$	$\frac{\sum_0^{1000}(A_{apx}-B_{apx})^2}{1000}$
4	0.50197641	0.13385114
6	0.68422137	0.14011397
8	0.85887515	0.06635435
10	0.9888148	0.07598861
12	1.21998609	0.04537707
14	1.34525875	0.04851236
16	1.50258734	0.03313601
18	1.66179525	0.03659251
20	1.914549	0.02561581
22	2.05511143	0.02897084
24	2.18103229	0.02158441
26	2.36678783	0.02401264
28	2.57727718	0.01835024
30	2.67819141	0.02002635
32	2.82972126	0.01693121
34	3.04080119	0.01744665
36	3.2542376	0.01431702
38	3.41778684	0.0158142
40	3.58954346	0.01302296
42	3.68630532	0.01403489
44	3.86674406	0.01158731
46	4.16554782	0.01264324
48	4.26822082	0.01142258
50	4.44967978	0.01133508
52	4.59953303	0.01037729
54	4.69040645	0.01086625
56	4.87838764	0.00958407
58	5.14547867	0.0098859
60	5.2536812	0.00881419
62	5.58523742	0.00945566
64	5.53316277	0.00845897
66	5.56058083	0.00852541
68	6.02842374	0.00797614
70	6.03871535	0.00817017
72	6.461322	0.00761195
74	6.56853033	0.00765401
76	6.49123665	0.00743027
78	6.68048361	0.00753303
80	7.01211211	0.00710712

$2n$	$\frac{\sum_0^{1000} (A_{rd} - B_{rd})^2}{1000}$	$\frac{\sum_0^{1000} (A_{apx} - B_{apx})^2}{1000}$
82	7.11891642	0.00713797
84	7.2566719	0.00680103
86	7.46841203	0.00692049
88	7.69075113	0.00665397
90	7.97416713	0.00647531
92	8.17884455	0.00639388
94	8.20542642	0.00631788
96	8.32945899	0.00611597
98	8.36297797	0.00627339
100	8.64226927	0.00583695

Amit megfigyelhetünk, hogy a véletlenszerűen beosztott csapatok esetében egyre nagyobb a négyzetes különbségek átlaga, míg az algoritmusunk által beosztottaké egyre kisebb, azonban különbség van a 4-gyel osztható és nem osztható esetekben, ugyanis a  $4k + 2$  elemszámú eseteknél a négyzetes különbségek átlaga általában nagyobb, mint a  $4k$  esetekben.

## 5. Random-3SAT nehézség

A következő fejezetben a *Random-3SAT* nehézséggel foglalkozunk, mely az NP-nehézséggel állítható párhuzamba. Felállítunk egy hipotézist (párhuzamban az  $NP \neq P$  hipotézissel), és belátjuk, hogy ha ez a hipotézis igaz, akkor nincs polinom idejű algoritmus bizonyos problémákra. Sőt, bizonyos problémákat, például a minimális kettéosztás feadatát (lásd később), még approximálni se tudjuk bizonyos hányadosnál jobban. Végül pedig a minimális kettéosztás problémájára bemutatunk egy közelítő algoritmust, amit véletlen inputokon letesztelünk, hogy az eredménye mennyire van távol az optimumtól. A fejezet U. Feige 2002-es cikke alapján készült [2]. Az utolsó alfejezet saját számítási eredményeket tartalmaz.

### 5.1. Alapfogalmak

**5.1. Definíció.** *Egy 3KNF (konjunktív normálforma) klózokból áll, mindegyikben 3 literállal, azaz egy-egy bináris változóval vagy a negáltjával. Egy klózon belül a literálok között vagy kapcsolat van, a klózok közt pedig és.*

**5.2. Definíció.** *A 3SAT nyelv a kielégíthető 3KNF-ek nyelve.*

Megjegyzés: A 3SAT nyelv NP-nehéz, azaz ha  $P \neq NP$ , akkor nincs olyan algoritmus, amely polinom időben eldöntené egy 3KNF-ről, hogy kielégíthető-e.

**5.3. Definíció.** *Egy random-3KNF formulát a következő módon állítunk elő: adott  $n$  és  $m$  pozitív egész paraméter,  $n$  a változók száma,  $m$  pedig a klózok száma. A klózokat egymástól függetlenül random generáljuk.  $\Delta = \frac{m}{n}$*

Megjegyzés: Ha  $\Delta$  elég nagy ( $(\frac{7}{8})^\Delta < \frac{1}{2}$ ), akkor egy random 3KNF szinte biztosan nem kielégíthető.

**HIPOTÉZIS 1:** *Ha  $\Delta$  tetszőlegesen nagy konstans, akkor nem létezik olyan polinom idejű algoritmus, amely a legtöbb random-3KNF-t cáfolja, vagyis azt mondja rájuk, hogy nem kielégíthetőek, de sosem cáfol tévesen, azaz egy kielégíthető 3KNF esetén nem hibázik.*

**HIPOTÉZIS 2:**  $\forall \epsilon > 0$  és tetszőlegesen nagy konstans  $\Delta$  esetén nincs olyan polinom idejű algoritmus, amely a legtöbb random-3KNF-t cáfolja, de sose cáfol, ha van legalább  $(1 - \epsilon)m$  kielégíthető klóz.

**5.4. Definíció.** *Egy probléma random-3SAT (R3SAT) nehéz, ha egy rá adott polinom idejű algoritmus cáfolná a HIPOTÉZIS 2-t.*

Megjegyzés: Ha a HIPOTÉZIS 2 igaz, akkor minden polinom időben nem megoldható probléma R3SAT nehéz, ha pedig nem igaz, akkor minden számítási feladat R3SAT nehéz, így ez nem jelent új bonyolultsági osztályt.



Megjegyzés: a 3SAT problémát  $\frac{7}{8}$ -nál jobban approximálni R3SAT nehéz.

Ismert [7], hogy minden  $n$ -re létezik egy  $\Delta_n$  küszöb, amire teljesül, hogy random formulák esetén ha a klózik száma szignifikánsan kisebb  $\Delta_n n$ -nél, akkor szinte biztosan kielégíthető a formula, ha pedig lényegesen nagyobb, akkor nagy valószínűséggel nem kielégíthető. Az is ismert, hogy ez a küszöb 3 és 4,6 között van, a kutatások szerint  $\Delta_n \simeq 4,2$ ,  $n$ -től függetlenül.

## 5.2. 3 változós függvények

Adott 3 Boole változó, ezeken definiálhatunk különböző függvényeket. Legyen az *AND* az a függvény, amikor a változók között *és* kapcsolat van, az *OR*, amikor *vagy*, a *XOR* pedig akkor lesz *igaz*, ha páratlan sok változó vesz fel 1-es értéket.

Legyen  $f$  egy 3 változós Boole függvény, és jelöljük  $t$ -vel (a *true*-ból származik a jelölés) azon különböző értékadások számát, amikor a függvényérték *igaz* lesz. Ez az *AND* esetén 1, a *XOR* esetén 4, az *OR* esetén 7. Jelöljük  $b$ -vel azon különböző értékadások számát, ahol páros vagy páratlan (amelyik nagyobb) sok 1-es értéket adunk és a függvényérték *igaz* lesz. Például a *XOR* és az *OR* esetben ez 4. 13 olyan Boole függvény létezik, ahol  $2b > t$ .

**5.5. Definíció.** Legyen  $f$  egy 3 változós Boole függvény. Nevezzük *3f* formulának azokat a formulákat, melyekben minden klóz 3 literált tartalmaz, és egy értékadás egy klózt akkor elégít ki, ha az  $f$  függvény *igaz* értéket ad.

**5.1. Tétel.** Minden  $f$  3 változós függvényre és az imént definiált  $t$  és  $b$ -re R3SAT nehéz megkülönböztetni a véletlen eseteit a *3f* formuláknak, ahol alig több, mint a klózik  $t/8$ -ad része kielégíthető, és azokat, ahol majdnem  $b/4$ -ed része kielégíthető.

Megjegyzés: a tételnek következménye, hogy R3SAT-nehéz approximálni  $t/2b$ -nél jobban a *Max-3f*-et.

**Bizonyítás:** Tegyük fel, hogy van egy  $B$  algoritmusunk, amely képes megbecsülni a kielégített klózik számát egy *3f* formulában, aminek a hányadosa  $t/2b$ -nél nagyobb. Nézzük meg a speciális eseteket: *AND* ( $t = b = 1$ ), *XOR* ( $t = b = 4$ )

Legyen  $\phi$  egy 3KNF  $n$  változóval és  $m = \Delta n$  klózzal, ahol  $\Delta$  elég nagy. Megmutatjuk, hogy  $B$  hogyan tud különbséget tenni a *tipikus*, azaz  $(7/8 + \epsilon)$ -kielégíthető és a *kivételes*, azaz  $(1 - \epsilon)$ -kielégíthető esetek közt. Tekintsük a következő algoritmust, ami a 3KNF-k nagy részére *tipikust* fog mondani, de a kivételesekre sose.

Átlagban minden literál  $3\Delta/2$ -ször szerepel  $\phi$ -ben. Ha  $\Delta$  elég nagy, akkor nagy valószínűséggel egy  $\epsilon$  résztől eltekintve a literálok  $(3/2 \pm \epsilon)\Delta$  között fognak

előfordulni  $\phi$ -ben. Ha ez nem teljesül, akkor legyen a kimenet *kivételes*.

Konstruáljunk három gráfot  $\phi$ -ből. A csúcsok száma minden esetben  $2n$ , minden literálhoz egy. Az első,  $G_1$  gráfban minden klóznál behúzzunk egy élt az első és második literálhoz tartozó csúcsok közé.  $G_2$ -ben a második és harmadik,  $G_3$ -ban a harmadik és az első közé. Szemidefinit programozással [8] adunk egy felső korlátot a maximális vágás méretére. Amennyiben ez bármelyik gráf esetén nagyobb, mint  $(\frac{1}{2} + \epsilon)m$ , akkor legyen a kimenet *kivételes*. Ha az algoritmus nem állt le, akkor  $\phi$ -re tekintsünk  $3f$  formulaként, ahol feltettük, hogy  $f$  *AND* vagy *XOR*. Futtassuk rá a  $B$  algoritmust. Ha  $B$  szerint a kielégíthető klózek száma  $(\frac{b}{4} - \epsilon)m$ , legyen az output *tipikus*, különben pedig *kivételes*.

**5.6. Definíció.** Egy *NAE – 3SAT* formulában egy klóz kielégíthető, ha nem azonos benne minden literál.

**5.1. Állítás.** A következők teljesülnek:

- 1) Az algoritmus elég nagy  $\Delta$  esetén majdnem minden formulára tipikus kimenetet ad.
- 2) Az algoritmus sose fog tipikust mondani egy olyan *3KNF*-ra, ahol van  $(1 - \epsilon)m$  kielégíthető klóz.

**Bizonytítás:** Mivel  $\phi$  random, ezért a gráfok lényegében véletlen gráfok  $2n$  ponttal és  $m$  éllel. Ha  $m$  kellően nagy, akkor egy egyszerű véletlent használó algoritmus megmutatja, hogy nagy valószínűséggel a maximális vágás egy véletlen gráfban legfeljebb  $(\frac{1}{2} + \epsilon)m$  élt tartalmaz. Sőt, megmutatták [9], hogy ha a maximális vágás  $(\frac{1}{2} + \epsilon)m$  élt tartalmaz, akkor a szemidefinit program kimenete a felső korlátra  $(\frac{1}{2} + \epsilon)m$  (egy másik  $\epsilon$ -ra). Ez azt jelenti, hogy ha ennél kisebb a maximális vágás, akkor nem fogjuk tévesen azt állítani, hogy nagy, és ekkor elérjük a következő fázist, vagyis futtathatjuk  $B$ -t a  $3f$  formulán.

Legyen  $\psi$  egy tetszőleges értékadása az  $n$  változónak. Azt állítjuk, hogy ha  $\phi$ -re egy *NAE – 3SAT* formulaként tekintünk, akkor legfeljebb  $(\frac{3}{4} + \epsilon)m$  klózt elégít ki  $\psi$ . Különböztünk egy random klózt és egy random literál párt, akkor annak a valószínűsége, hogy  $\psi$  a literáloknak különböző értéket ad, legalább  $\frac{2}{3}(\frac{3}{4} + \epsilon) \simeq \frac{1}{2} + \epsilon$ , mivel  $\frac{2}{3}$  annak a valószínűsége, hogy a literál pár különböző értéket vesz fel, feltéve, hogy a klóz literáljai nem mind azonos értéket vesznek fel. Utóbbi valószínűsége  $\frac{3}{4} + \epsilon$  a korábbi indirekt feltevés szerint. Így legalább az egyik gráfban a  $\psi$  által keletkezett vágásnak több, mint  $(\frac{1}{2} + \epsilon)m$  éle van (egy literál kerüljön balra, ha  $\psi$  értéke 0, és jobbra, ha 1).

Legyen  $\psi$  egy tetszőleges értékadása az  $n$  változónak, ami  $\phi$ -nek (*3SAT* formulaként) legalább  $(\frac{1}{2} - \epsilon)m$  klózt kielégíti. Ugyanez a  $\psi$  legfeljebb  $(\frac{3}{4} + \epsilon)m$  klózt elégíti ki  $\phi$ -nek *NAE – 3SAT* formulaként, így a klózek legalább  $(\frac{1}{4} - \epsilon)m$  részét elégíti ki, ha  $\phi$  *3AND* formula. Sőt, mivel minden literál nagyjából ugyanannyiszor jelenik meg  $\psi$ -ben, ezért  $\psi$  átlagosan  $\frac{3}{2}$  literált elégít ki egy klózból. Mivel mindössze  $\epsilon m$  klózból nincs kielégítve literál  $\psi$  által, ezért leg-

alább  $(\frac{3}{4} - \epsilon)m$  klózban van pontosan 1 literál kielégítve. Így  $\psi$  kielégít  $(1 - \epsilon)m$  klózt, ha  $\phi$  egy  $3XOR$  formula.

Mivel  $\phi$  random, ezért  $3AND$  formulaként  $(\frac{1}{8} + \epsilon)m$ ,  $3XOR$  formulaként  $(\frac{1}{2} + \epsilon)m$  klóz elégíthető ki. A többi függvényre hasonlóan bizonyítható bizonyos literálok negálásával.

**Következmény:** Minden  $\epsilon > 0$  és elég nagy  $\Delta$  esetén  $R3SAT$ -nehéz a legtöbb  $3AND$  formulára *tipikus* kimenetet adni, viszont sose *tipikus* értékkel visszatérni, ha a van legalább  $(\frac{1}{4} - \epsilon)$  kielégíthető klóz.

A következő szakaszban az előző tételt fogjuk alkalmazni gráf problémákra.

### 5.3. Alkalmazás gráf problémákra

**5.2. Állítás.** Minden  $\epsilon > 0$  esetén van egy  $\Delta_\epsilon > 0$  küszöb, hogy minden  $\Delta > \Delta_\epsilon$  és elég nagy  $n$  esetén nagy valószínűséggel teljesül, hogy a klózik bármely  $(\frac{1}{8} + \epsilon)m$  méretű részhalmaza egy  $3KNF$ -ben, ahol  $m = \Delta n$ , van legalább  $n + 1$  literál.

**Bizonyítás:** Rögzítsünk le egy  $n$  literálból álló  $S$  halmazt. Annak a valószínűsége, hogy egy random klóz, amiben három literál van, teljesen diszjunkt legyen  $S$ -től,  $\frac{1}{8}$ . Ha  $n$  elég nagy, akkor tudjuk, hogy  $1 - 2^{-3n}$ -nél nagyobb valószínűséggel legfeljebb  $(\frac{1}{8} + \epsilon)m$  random klóz kerül el  $S$ -t. Mivel nagyjából  $2^{2n}$  lehetőség van kiválasztani  $S$ -t, így az unióra vonatkozó korlát miatt semelyik se fog elkerülni egy  $(\frac{1}{8} + \epsilon)m$  klózból álló halmazt.

#### 5.3.1. Kiegyensúlyozott páros klikk feladat

*Input:*  $n \times n$  csúcsú páros gráf

*Output:*  $k \times k$  csúcsú teljes páros részgráf

*Célfüggvény:*  $\max k$

A kutatások korábban általában nem követelték meg, hogy a klikk kiegyensúlyozott legyen, azaz egy  $k_1 \times k_2$  csúcsú teljes páros részgráf volt a kimenet. Ekkor két érdekes célfüggvényük is lehet, egyrészt legyen minél több éle a páros klikknek, azaz maximalizáljuk a  $k_1 k_2$ -t, vagy minél több csúcsa, azaz az összegük legyen minél nagyobb, ez utóbbi egyébként polinom időben megoldható.

**5.1. Lemma.** Minden  $\epsilon > 0$  esetén  $R3SAT$  nehéz approximálni a kiegyensúlyozott klikk problémát  $1/2 + \epsilon$ -nál jobban, azaz  $R3SAT$  nehéz különbséget tenni a  $k > 1/4 - \epsilon$  és a  $k < 1/8 + \epsilon$  között.

**Bizonyítás:** Vezessük vissza rá a  $MAX3AND$  feladatot  $n'$  változóval és  $m' = \Delta n'$  klózzal. Mindkét oldalon legyen  $n = m'$  csúcs, minden klózhoz 1-1. Két különböző oldalon lévő csúcs közé húzzunk be élt, ha a hozzájuk tartozó klózik egyszerre kielégíthetőek, vagyis nem szerepel egy változó az egyikben és

a negáltja a másikban. Ha a kiegyensúlyozott páros klikk feladatot jobban tudnánk approximálni a lemma állításánál, akkor meg tudnánk különböztetni a  $MAX3AND$  formulák közül azokat, amik  $1/4 - \epsilon$ -kielégíthetők, azaz *kivételesel*, és azokat, amik  $1/8 + \epsilon$ -kielégíthetők, vagyis *tipikusak*.

Ha van  $(1/4 - \epsilon)m'$  kielégíthető klóz, akkor a nekik megfelelő csúcsok egy  $k \times k$  méretű kiegyensúlyozott páros klikknek felelnek meg, ahol  $k = (1/4 - \epsilon)n$ .

Ha a  $3AND$  formula random, akkor az 5.2-es Állítás miatt minden  $(1/8 + \epsilon)m'$  méretű részhalmaza a klózoknak tartalmaz  $n' + 1$  literált. Így minden ekkora halmazpárra igaz, hogy lesz egy változó, amely pozitív változata meg fog jelenni az egyik oldat és a negatív a másikon, ami kizárja a lehetőséget, hogy legyen egy  $k \times k$  méretű páros klikk, ahol  $k = (1/8 + \epsilon)n$ .

### 5.3.2. Minimális kettéosztás

*Input:* Egy  $2n$  csúcsú  $G$  gráf.

*Output:*  $S \subset V(G)$ ,  $|S| = n$

*Célfüggvény:* minimalizáljuk az  $S$  és  $V(G) - S$  közötti élek számát.

**5.2. Tétel.** *R3SAT-nehez  $4/3$ -nál jobban approximálni a feladatot.*

**Bizonyítás:** Visszavezetjük rá a random  $MAX3AND$  feladatot,  $n'$  változóval és  $m'$  klózzal,  $m' = \Delta n$ . Itt meg akarjuk különböztetni, azokat az eseteket, amikor legfeljebb  $(1/8 + \epsilon)m$  klóz elégíthető ki azoktól, amikor legalább  $(1/4 - \epsilon)m$ . A baloldal tartalmazzon  $2n'$  pontot, minden literálhoz egy, a jobboldal pedig  $m'$  klasztert, minden klózhhoz egy klaszter tartozik és minden klaszter egy  $4m'$  méretű klikk. Ezen felül a gráf tartalmazzon még egy extra klasztert, ami egy  $m''$  méretű klikk, ahol  $m'' = 4m'(1/2 + 2\epsilon)m'$ . Tehát a gráf csúcsainak száma  $4m'(3/2 + 2\epsilon)m' + 2n'$ .

Megjegyzés: feltesszük, hogy  $(1/4 - \epsilon)m'$  egész szám, amiből következik, hogy a ponthalmaz kettéosztható egyenlő részekre anélkül, hogy a klikkeket, vagy az extra klikket el kéne vágni: kerüljön  $S$ -be  $n'$  pont a baloldalról,  $(1/4 - \epsilon)m'$  klaszter a jobboldalról és az extra klikk.

Minden klaszterben legyen egy megjelölt *összekötő* pont. Húzzunk be élt egy összekötő pont és egy literálhoz tartozó pont közé, ha a literál benne van az összekötő pont klaszteréhez tartozó klózban. Ezeket nevezzük *páros* éleknek. Tudjuk, hogy  $3m'$  van belőlük. Mivel létezik olyan kettéosztás, amelyik csak páros éleket vág el, a minimális kettéosztás nem vág el egyetlen klasztert se (már egy klaszter elvágása  $4m' - 1$  élt jelentene), és az extra klikket sem. Ebből következik, hogy a minimális kettéosztás egyik halmaza tartalmaz a baloldalról pontosan  $n'$  pontot,  $(1/4 - \epsilon)m'$  klasztert és az extra klikket.

Vegyük észre, hogy minden összekötő pontból 3 él megy balra, tehát a baloldali pontok átlagfokszáma  $3\Delta/2$ . Nagy valószínűséggel (a random  $3AND$  formulától függően), csak egy  $\epsilon$  része a páros éleknek csatlakozik olyan baloldali pontokhoz, melyek foka  $3\Delta/2$ -től  $\epsilon$ -nál jobban eltér (\*). Ha ez nem teljesül, akkor nem tudjuk megcsinálni a redukciót egy kettéosztásra és ekkor az algoritmus kimenete legyen *kivételes* az eredeti  $3AND$  formulára.

$O(\epsilon)$  hibától eltekintve feltehetjük, hogy minden baloldali pont foka azonosan  $3\Delta/2$ . Ha a  $3AND$  formulának van  $(1/4 - \epsilon)m'$  kielégíthető klóza, akkor az  $S$  halmazba ezekhez a klózokhoz tartozó klaszterek pontjai kerüljenek, valamint az extra klikk, és az az  $n'$  literál, amely ezeket kielégíti. Egyedül azokat az ékeket vágja el ez a kettéosztás, amelyek ezt az  $n'$  literált kötik össze a kielégítetlen klózokkal. A páros élek  $S$ -en belül  $3(1/4 - \epsilon)m'$ . Az összfokszám az ezen literáloknak  $3n'\Delta/2 = 3m'/2$ . Így a kettéosztás  $3(1/4 - \epsilon)m'$  élt tartalmaz.

Egy random  $3AND$  formulában szükségünk van ennek a kettévágásnak az egyik oldalára, hogy tartalmazzon  $n'$  literált,  $(1/4 - \epsilon)m'$  klózt és az extra klikket. Ez az  $n'$  literál legfejebb  $(1/8 + \epsilon)m'$  klózzal vannak 3-szorosan összekötve az 5.2-es Állítás miatt, és további  $(1/8 - 2\epsilon)m'$ -vel kétszeresen. Így ezen a kettévágás éleinek száma legalább

$$3m'/2 - 3(1/8 + \epsilon)m' - 2(1/8 - 2\epsilon)m' = (1 - \epsilon)m'$$

A két esetben az élszámok hányadosa tetszőlegesen közel tud lenni a  $4/3$ -hoz.

Összegezve, ha lehetséges lenne a minimális kettéosztás feladatot  $4/3$ -nál jobban approximálni, akkor fel tudnánk ismerni a *tipikus*  $3AND$  formulákat a következő módon:

Állítsuk elő a gráfot és a kettéosztást a fenn említett módon.

Ha a baloldalon lévő pontok fokszámaira nem teljesül a (\*), akkor azt mondjuk, hogy a formula *kivételes*.

Futtassuk az approximációs algoritmust a kettéosztásra és legyen a végeredmény *tipikus*, ha az output kizárja, hogy legyen egy  $3(1/4 - \epsilon)m'$  élű kettéosztás.

#### 5.4. Közelítő algoritmus a minimális kettéosztás feladatra

Az előző fejezetben bebizonyítottuk, hogy a minimális kettéosztás feladatot  $\frac{4}{3}$ -nál jobban approximálni R3SAT-nehéz. Vagyis ha a *HIPOTÉZIS* 2 igaz, akkor nincs olyan polinom idejű algoritmus, ami minden gráf esetén  $\frac{4}{3}$ -nál jobban meg tudja közelíteni az optimumot. A következő fejezetben bemutatunk egy algoritmust, ami megpróbálja megközelíteni az optimumot. A hatékonyságát szimuláció segítségével mérjük meg egy-egy véletlen gráfon.

Az algoritmusunk lényege, hogy az összes lehetséges kiindulási állapotból számolunk lényegében mohó módon egy kettéosztást, és a legjobbat választjuk.

Legyen egy fázis a következő: A gráf két pontját berakjuk két külön halmazba,  $A$ -ba és  $B$ -be. Egy lépés álljon abból, hogy megkeressük a gráfnak azon pontját, amelyet ha  $A$ -hoz hozzáveszünk, akkor az  $A$  és  $B$  közt a lehető legkevesebb él fusson. Ezután megkeressük ugyanezt  $B$ -re. A lépések száma a gráfok pontszámának fele, mínusz kettő, vagyis amíg be nem osztunk minden pontot  $A$ -ba vagy  $B$ -be.

Az iménti fázist a gráf összes lehetséges pontpárjával végigcsináljuk, és azzal a kettéosztással térünk vissza, amelyben a legkevesebb él futott  $A$  és  $B$  közt.

Az optimumot ezúttal is egészértékű programozás segítségével fogjuk kiszámolni. Az élekhez hozzárendelünk egy bináris  $x$  vektort, a pontokhoz pedig egy  $y$ -t, és azt akarjuk elérni, hogy  $x(e)$  akkor legyen 1, ha az él bekerül a vágásba, különben 0, az  $y(v)$  pedig akkor, amikor a pont az egyik halmazba kerül, és 0, ha a másikba. Az  $y$  értékei közül pontosan a pontszám fele legyen 1 és ugyanennyi 0. Minimalizáljuk az élek számát a vágásban:

$$\begin{aligned} \min \mathbf{1}x \\ x(u, v) &\geq y(u) - y(v), \quad \forall (u, v) \in E \\ x(u, v) &\geq y(v) - y(u), \quad \forall (u, v) \in E \\ \mathbf{1}y &= \frac{n}{2} \\ x &\in [0, 1]^{|E|} \\ y &\in [0, 1]^{|V|} \end{aligned}$$

( $n$ -ről feltesszük, hogy páros).0

#### 5.4.1. A szimuláció eredményei

Először a következő véletlen gráfon szimuláltunk: legyen  $n$  páros, adott  $n$  pont, és bármely kettő közé kerüljön él  $\frac{1}{2}$  valószínűséggel. Az első néhány  $n$  esetén (konkrétan  $n \leq 38$ ) a közelítő algoritmus megtalálta az optimumot minden futtatáskor.

Kicsivel változatosabb eredmények születtek, ha a gráf a Barabási-Albert modelltől származott [10]. Az általunk vizsgált modellben  $n$  pont esetén egy  $m+1$  csúcsú csillagból kiindulva minden további csúcs  $m$  szomszédot kap, és nálunk  $m = n/3$ . Az alábbi táblázatban található a szimuláció eredménye, minden páros  $n$ -re 20 és 40 között 6-szor lefuttattuk az algoritmust, és megmutatjuk, hogy a mi algoritmusunk kimenetének hányszorosa az optimum átlagosan, illetve legrosszabb esetben a 6-ból. Az 5.2-es tételben szereplő  $4/3$ -os korláthoz képest mi ennek a hányadosnak a reciprokát számoljuk ki a méréseinkben, azaz az optimális költséget osztjuk a közelítő algoritmus által kiszámolt megoldás költségével.

$n$	6 mérés átlaga	6 mérés minimuma
20	0.99479	0.96875
22	1.0	1.0
24	1.0	1.0
26	1.0	1.0
28	0.99462	0.98387
30	0.98438	0.97260
32	0.99593	0.98780
34	0.98283	0.96938
36	0.99192	0.98095
38	0.99039	0.98347
40	0.99632	0.99253

Ahhoz, hogy statisztikai szempontból érdekes következtetéseket vonjunk le, vagy legyen bármilyen elképzelésünk az asszimptotikus viselkedésről, szükségünk lenne további mérésekre, nagyobb pontszámok esetén azonban az optimumnak a kiszámítása időigényes. Azt viszont így is észrevehetjük, hogy bizonyítottan nem várható (csak a hipotézisek megdőlése esetén) olyan approximációs algoritmus, amely minden gráfra  $4/3$ -nál jobban megközelíti az optimumot, az nem jelenti azt, hogy gyakorlatban, bizonyos korlátozások mellett is reménytelen lenne az optimális megoldást ennél a hányadosnál lényegesen jobban közelíteni.

## Hivatkozások

- [1] P. W. Shor, "The average-case analysis of some on-line algorithms for bin packing", *Combinatorica* 6, pp. 179-200 (1986).
- [2] U. Feige. "Relations between average case complexity and approximation complexity." In *Proc. on 34th Annual ACM Symposium on Theory of Computing*, 534-543, 2002.
- [3] R.M. Karp, M. Luby, and A. Marchetti-Spaccamela, "Probabilistic analysis of multidimensional bin packing problems," *Proceedings of the 16th ACM Symp. on Theory of Computing*, pp. 289-298, 1984
- [4] R. M. Karp, M. Luby and A. Marchetti-Spaccamela, "Probabilistic analysis of multi-dimensional bin packing problems", *Proc. 16th ACM Symp. on Theory of Computing*, (1984), 289-298.
- [5] F. T. Leighton and P. W. Shor, "Tight bounds for minimax grid matching, with applications to the average case analysis of algorithms". *Proc. 18th ACM Symp. on Theory of Computing*, (1986), 91-103.
- [6] Bolla M., Krámlı A., Nagy-György J. "Többváltozós statisztikai módszerek" 23. oldal, 2013
- [7] Ehud Friedgut. "Necessary and sufficient conditions for sharp thresholds of graph properties, and the k-SAT problem". *J. Amer. Math. Soc.*, 12:1017-1054, 1999.
- [8] M. Goemans, D. Williamson "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming". *JACM*, 42:1115-1145, 1995
- [9] Uri Zwick "Outward rotations: a tool for rounding solutions for semidefinite programming relaxations, with applications to MAX CUT and other problems". In *Proc. 31st STOC*, 1999, 679-687.
- [10] A. L. Barabási and R. Albert "Emergence of scaling in random networks", *Science* 286, pp 509-512, 1999.