ALKALMAZOTT MATEMATIKAI LAPOK

A MAGYAR TUDOMÁNYOS AKADÉMIA MATEMATIKAI TUDOMÁNYOK OSZTÁLYÁNAK KÖZLEMÉNYEI

ALAPÍTOTTÁK
KALMÁR LÁSZLÓ, TANDORI KÁROLY, PRÉKOPA ANDRÁS, ARATÓ MÁTYÁS

FŐSZERKESZTŐ
PÁLES ZSOLT

FŐSZERKESZTŐ-HELYETTESEK
BENCZÚR ANDRÁS, SZÁNTAI TAMÁS

FELELŐS SZERKESZTŐ
VIZVÁRI BÉLA

TECHNIKAI SZERKESZTŐ
KOVÁCS GERGELY

29. kötet

Szerkesztőség és kiadóhivatal: 1055 Budapest, Falk Miksa u. 12.

Az Alkalmazott Matematikai Lapok változó terjedelmű füzetekben jelenik meg, és olyan eredeti tudományos cikkeket publikál, amelyek a gyakorlatban, vagy más tudományokban közvetlenül felhasználható új matematikai eredményt tartalmaznak, illetve már ismert, de színvonalas matematikai apparátus újszerű és jelentős alkalmazását mutatják be. A folyóirat közöl cikk formájában megírt, új tudományos eredménynek számító programokat, és olyan, külföldi folyóiratban már publikált dolgozatokat, amelyek magyar nyelven történő megjelentetése elősegítheti az elért eredmények minél előbbi, széles körű hazai felhasználását. A szerkesztőbizottság bizonyos időnként lehetővé kívánja tenni, hogy a legjobb cikkek nemzetközi folyóiratok különszámaként angol nyelven is megjelenhessenek.

A folyóirat feladata a Magyar Tudományos Akadémia III. (Matematikai) Osztályának munkájára vonatkozó közlemények, könyvismertetések stb. publikálása is.

A kéziratok a főszerkesztőhöz, vagy a szerkesztőbizottság bármely tagjához beküldhetők. A főszerkesztő címe:

Páles Zsolt, főszerkesztő
1055 Budapest, Falk Miksa u. 12.
A folyóirat e-mail címe: aml@math.elte.hu

Közlésre el nem fogadott kéziratokat a szerkesztőség lehetőleg visszajuttat a szerzőhöz, de a beküldött kéziratok megőrzéséért vagy továbbításáért felelősséget nem vállal.

Az Alkalmazott Matematikai Lapok előfizetési ára évfolyamonként 1200 forint. Megrendelések a szerkesztőség címén lehetségesek.

A Magyar Tudományos Akadémia III. (Matematikai) Osztálya a következő idegen nyelvű folyóiratokat adja ki:

1. Acta Mathematica Hungarica,
2. Studia Scientiarum Mathematicarum Hungarica.

Az Alkalmazott Matematikai Lapok megjelenését támogatja
a Magyar Tudományos Akadémia Könyv- és Folyóiratkiadó Bizottsága.

A kiadásért felelős a BJMT főtitkára
Szedte és tördelte Éliás Mariann

Nyomta a Nagy és Társa Kft., Budapest
Felelős vezető: Fódi Gábor

Budapest, 2012
Megjelent 18 (A/5) ív terjedelemben
250 példányban
HU ISSN 0133-3399

ÚTMUTATÁS A SZERZŐKNEK

Az Alkalmazott Matematikai Lapok csak magyar nyelvű dolgozatokat közöl. A közlésre szánt dolgozatokat e-mailen az `aml@math.elte.hu` címre kérjük elküldeni az ábrákat tartalmazó fájlokkal együtt. Előnyben részesülnek a LaTeX-ben elkészített dolgozatok.

A kéziratok szerkezeti felépítésének a következő követelményeket kell kielégíteni:

Fejléc: A fejlécnek tartalmaznia kell a dolgozat címét és a szerző teljes nevét.

Kivonat: A fejléc után egy, képletet nem tartalmazó, legfeljebb 200 szóból álló kivonatot kell minden esetben megadni.

Fejezetek: A dolgozatot címmel ellátott szakaszokra kell bontani, és az egyes szakaszokat arab sorszámozással kell ellátni. Az esetleges bevezetésnek mindig az első szakaszt kell megnevezni.

A dolgozatban előforduló képleteket a dolgozat szakaszokra bontásától független, folytatólagos arab sorszámozással kell azonosítani. Természetesen nem szükséges minden képletet számozással ellátni, csak azokat, amelyekre a szerző a dolgozatban hivatkozni kíván.

Mind az ábrákat, mind a lábjegyzeteket szintén folytatólagos arab sorszámozással kell ellátni. Az ábrák elhelyezését a dolgozat megfelelő helyén ábraazonosító sorszámokkal kell megadni. A lábjegyzetekre a dolgozaton belül az azonosító sorszám felső indexként használatával lehet hivatkozni.

Az esetleges definíciókat és tételeket (segédtételeket és lemmákat) szakaszonként újrakezdődő, ponttal elválasztott, kettős számozással kell ellátni. Kérjük a szerzőket, hogy ezeket, valamint a tételek bizonyítását a szövegben kellő módon emeljék ki.

Irodalomjegyzék: A dolgozatok szövegében az irodalmi hivatkozás számait szögletes zárójelben kell megadni, mint például [2] vagy [1, 7–13].

Az irodalmi hivatkozások formája a következő: Minden hivatkozást fel kell sorolni a dolgozat végén található irodalomjegyzékben, a szerzők, illetve a társszerzők esetén az első szerző neve szerint alfabetikus sorrendben úgy, hogy a cirill betűs szerzők nevét a Mathematical Reviews átírási szabályai szerint latin betűsre kell átírni. A folyóiratban megjelent cikkekre [1], a könyvekre [2] a következő minta szerint kell hivatkozni:

[1] Farkas, J.: *Über die Theorie der einfachen Ungleichungen*, Journal für die reine und angewandte Mathematik **124**, (1902) 1–27.

[2] Zoutendijk, G.: *Methods of Feasible Directions*, Elsevier Publishing Company, Amsterdam and New York (1960), 120 o.

Szerző adatai: Az irodalomjegyzék után, a kézirat befejezéseképpen fel kell tüntetni a szerző teljes nevét és a munkahelye (esetleg lakása) pontos címét, illetve e-mail címét.

Idegen nyelvű kivonat: Minden dolgozathoz csatolni kell egy angol nyelvű összefoglalót.

A szerzők a dolgozatukról 20 darab ingyenes különlenyomatot kapnak. A dolgozatok után szerzői díjat az Alkalmazott Matematikai Lapok nem fizet.

TARTALOMJEGYZÉK

INDEX

Alkalmazott Matematikai Lapok **29** (2012), 1-52.

MULTIGRÁFOK FOKSOROZATAI

IVÁNYI ANTAL ÉS LUCZ LORÁND

Havel 1955-ben [28], Erdős és Gallai 1960-ban [20], Hakimi 1962-ben [27], Tripathi, Venugopalan és West 2010-ben [87], Özkan [62] 2011-ben javasoltak módszert annak eldöntésére, hogy nemnegatív egészek sorozata lehet-e egy egyszerű gráf foksorozata. Ezeknek az algoritmusoknak a legrosszabb futási ideje legalább négyzetes. Takahashi 2007-ben [84], Hell és Kirkpatrick [29] 2009-ben lineáris algoritmust javasoltak. 1974-ben Chungphaisan [18] kiterjesztette a csúcspárok között legfeljebb $b \geq 1$ élet tartalmazó multigráfokra mind a Havel–Hakimi-, mind pedig az Erdős–Gallai-tételt. Ezeknek az algoritmusoknak is legalább négyzetes a legrosszabb futási ideje. Cikkünkben bemutatjuk a Chungphaisan–Erdős–Gallai-algoritmus lineáris változatát. A Chungphaisan–Havel–Hakimi-algoritmust pedig úgy javítjuk és gyorsítjuk, hogy $b = 1,\ 2$ esetén is lineáris futási idejű legyen.

1. Bevezetés

A gyakorlatban különböző területeken szükség van objektumok rangsorolására. Ennek egyik elterjedt módszere, hogy az objektumokat páronként összehasonlítjuk, és az összehasonlítás eredményeképpen pontokat adunk az objektumoknak, végül pedig az objektumokat a kapott pontszámok alapján rangsoroljuk. Például Landau biológiai [47], Hakimi kémiai [27], Kim et al. [40], valamint Newman és Barabási [61] hálózati, Bozóki, Fülöp, Kéri, Poesz és Rónyai gazdasági [11, 12, 39], Liljeros et al. emberi kapcsolatokra vonatkozó [48], Iványi et al. pedig sportbeli [31, 32, 35, 37, 65, 67, 69] alkalmazásokra hivatkoztak.

Legyenek a, b és n egészek, $n \geq 1$ és $b \geq a \geq 0$. Az (a, b, n)-*gráfok* olyan hurokmentes – irányított vagy irányítatlan – gráfok, melyek csúcshalmaza $V = \{v_1, \ldots, v_n\}$ és a különböző v_i és v_j csúcsok legalább a és legfeljebb b éllel vannak összekötve. Eszerint az *egyszerű irányítatlan gráfok* $(0, 1, n)$-gráfok, míg a *tournamentek* $(1, 1, n)$-gráfok.

Irányított gráfok esetén, ha v_i és v_j összehasonlításakor v_i kap egy pontot, akkor annak a gráfban v_i-ből v_j-be menő irányított él felel meg. Irányítatlan gráfok esetén viszont csúcspárok kapják a pontot, és annak a két csúcsot összekötő irányítatlan él felel meg.

Ebben a cikkben elsősorban azt vizsgáljuk, hogy nemnegatív egész számok $s = (s_1, \ldots, s_n)$ nemnövekvő sorozata és adott a alsó korlát, valamint b felső

korlát esetén létezik-e olyan irányítatlan (a, b, n)-gráf, amelynek foksorozata s. Ennek megfelelően – ha mást nem mondunk – a gráf kifejezés irányítatlan gráfot jelent.

Emellett foglalkozunk a foksorozatok számával, amelyet $G(a, b, n)$-nel jelölünk.

A hasonló feladatokkal kapcsolatban megjegyezzük, hogy mind az irányítatlan, mind pedig az irányított gráfokkal kapcsolatban az utóbbi néhány évben is számos publikáció jelent meg (például [5, 7, 8, 13, 19, 21, 26, 29, 34, 50, 55, 58, 62, 65, 70, 85, 87, 88, 89], illetve [6, 9, 10, 12, 15, 22, 24, 31, 32, 37, 38, 40, 43, 46, 53, 51, 52, 57, 64, 67, 68]).

Legyenek l, m és u egész számok, továbbá $1 \leq m$ és $l \leq u$. Egész számok $s = (s_1, \ldots, s_m)$ sorozatát (l, u, m)-*korlátosnak* (röviden: korlátosnak) nevezzük, ha $l \leq s_i \leq u$ minden $1 \leq i \leq m$ indexre. Az $s = (s_1, \ldots, s_m)$ (l, u, m)-korlátos sorozatot (l, u, m)-*szabályosnak* mondjuk, ha $u \geq s_1 \geq \cdots \geq s_m \geq l$.

A vizsgálatok során kitüntetett szerepet játszanak az $(a(n-1), b(n-1), n)$-szabályos sorozatok. Ezeket a sorozatokat (a, b, n)-*grafikusnak* (vagy röviden grafikusnak) nevezzük, ha létezik olyan (a, b, n)-gráf, melynek foksorozata s.

Jelentős számú cikk (például [14, 23, 44, 56]) foglalkozik páros számok *grafikus felbontásaival*: előállítják a $2k$ páros szám pozitív egész összeadandókra való monoton csökkenő felbontásait, és az így kapott $q = (q_1, \ldots, q_m)$ sorozatok közül – amelyekre $q_1 + \cdots + q_m = 2k$ és $q_m \geq q_{m-1} \geq \cdots \geq q_1$ – szűrik ki a $(0, 2k-1, 2k)$-grafikus sorozatokat, vagy pedig rekurzióval eleve csak a grafikus sorozatokat állítják elő.

A továbbiakban főleg szabályos sorozatokkal foglalkozunk. A definíciókban az alsó és felső korlátok azért szerepelnek, hogy ellenőrző algoritmusainkat megkíméljük a nyilvánvalóan nem grafikus sorozatok ellenőrzésétől, ezért ezek a megszorítások nem jelentik az általánosság korlátozását.

A cikkben csak *teljes* gráfokkal foglalkozunk. Ezekre az jellemző, hogy ha $a \leq c \leq b$, akkor bármely két csúcs között c él is meg van engedve, és az irányított esetben azok tetszőlegesen irányíthatók (azaz eltérünk a teljes gráfok szokásos definíciójától). A *hiányos* gráfoknál bizonyos lehetőségek tiltva vannak. Például a labdarúgásnak [24, 33, 35, 45] olyan irányított $(2, 3, n)$-gráfok felelnek meg, amelyekben a csúcsokat 2 vagy 3 él köti össze, azonban 2 él esetén azok mindig ellentétesen, míg 3 él esetén azok mindig azonosan vannak irányítva.

Míg teljes gráfok esetén a sorozatok tesztelése az operációkutatás folyamos módszereivel kényelmesen megoldható (bár gyakran vannak gyorsabb algoritmusok is), hiányos gráfok esetén ezek a módszerek nem alkalmazhatók.

Cikkünk fő célkitűzése, hogy minél kisebb várható futási idejű algoritmusokat találjunk annak eldöntésére, hogy adott s szabályos sorozat grafikus-e. Eközben a minden sorozatot helyesen minősítő *pontos*, és a csak a szabályos sorozatok egy részét minősítő *közelítő* algoritmusokkal is foglalkozunk.

Érdemes megemlíteni, hogy a fokszámsorozatok számának meghatározásával kapcsolatos nehézségek miatt annak is jelentős irodalma (lásd például [8, 19, 57]) van, hogy véletlen mintavétellel becsüljük ezeket a számokat.

Melléktermékként bővítettük a *The On-Line Encyclopedia of Integer Sequences* adatbázist [36, 51, 52].

Módszerünk az összes grafikus sorozat gazdaságos előállítására is alkalmas (lásd Ruskey [71], valamint Barnes és Savage cikkeit [3, 4]).

A cikk felépítése a következő. A bevezető első rész után a $(0,1,n)$ témakör klasszikus pontos algoritmusait foglaljuk össze. A harmadik részben új pontos algoritmusokat, a negyedikben általános leszámlálási eredményeket, az ötödikben pedig új tesztelő algoritmusokat ismertetünk. A hatodik részben a közelítő algoritmusok hatékonyságát és futási idejét, míg a hetedikben a pontos algoritmusok futási idejét elemezzük. A nyolcadik rész témája a $(0,b,n)$-gráfok potenciális foksorozatainak tesztelése, míg a kilencedikben az (a,b,n)-gráfoké a főszerep. A tizedik részben a $(0,1,n)$-grafikus sorozatok párhuzamos leszámlálása a téma.

2. Klasszikus pontos algoritmusok $(0,1,n)$-gráfokhoz

Ebben a részben két, a $(0,1,n)$-gráfok potenciális foksorozatainak tesztelésére alkalmas klasszikus algoritmust ismertetünk.

2.1. Havel–Hakimi-algoritmus (HH)

A feladat megoldására az első módszert Vaclav Havel cseh matematikus javasolta 1955-ben [28, 49]. 1962-ben Louis Hakimi [27] Haveltől függetlenül publikálta ugyanezt az eredményt, ezért ma a tételt rendszerint *Havel–Hakimi-tételnek*, a módszert pedig *Havel–Hakimi-algoritmusnak* nevezik.

2.1. Tétel. (Hakimi [27], Havel [28]) *Ha $n \geq 3$, az $(s_1,\ldots,s_n)$ $(0,1,n)$-szabályos sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha az*

$$(s_2-1, s_3-1, \ldots, s_{s_1}-1, s_{s_1+1}-1, s_{s_1+2}, \ldots, s_n)$$

sorozat $(0,1,n-1)$-grafikus.

Bizonyítás. Lásd [27, 28]. □

A továbbiakban sorozatok ismétlődő elemeinek tömör jelölésére használjuk az $s=(c^d)$ típusú jelölést, ami azt jelzi, hogy a sorozat d darab c-t tartalmaz.

Ha ezen tétel alapján írunk egy rekurzív algoritmust, akkor annak futási ideje legjobb esetben – például az egy darab $n-1$ után $n-1$ nullát tartalmazó bemenetre – $\Theta(1)$, legrosszabb esetben pedig – például az n darab $(n-1)$-et tartalmazó *homogén* bemenetre – $\Theta(n^2)$. Ez ugyanis grafikus sorozat, ezért minden elemét ellenőrizni kell. Másrészt az elemek összege négyzetes, és az algoritmus az elemeket egyesével csökkenti nullára. Érdemes megjegyezni, hogy a tétel bizonyítása konstruktív, és a bizonyításon alapuló algoritmus négyzetes idő alatt nem csak ellenőriz, hanem egy megfelelő gráfot is előállít (feltéve persze, hogy létezik megfelelő egyszerű gráf).

A következő, Havel–Hakimi-típusú algoritmus csak a bemenet tesztelését végzi el, helyreállítását nem.

A cikk programjaiban a [16] tankönyvben leírt pszeudokód konvenciókat követjük.

Itt és a továbbiakban n a sorozat hosszát (a gráf csúcsainak számát) jelöli, $s = (s_1, \dots, s_n)$ a vizsgálandó szabályos sorozat, L pedig a vizsgált sorozat grafikusságát jellemzi: $L = 0$ azt jelenti, hogy a vizsgált sorozat nem grafikus; $L = 1$ esetén a sorozat grafikus, míg $L = 2$ azt jelzi, hogy az adott algoritmus *nem tud* dönteni.

2.1. Algoritmus. Havel-Hakimi(n, s)

```
1. for i = 1 to n − 1                          // 1–6. sor: s elemeinek tesztelése
2.     if s_{s_i+i} == 0                       // 2–4. sor: s nem grafikus
3.         L = 0
4.         return 0
5.     for j = i + 1 to i + s_i
6.         s_j = s_j − 1
7.     (s_{i+1}, ..., s_n) rendezése nemnövekvő sorrendbe
8. L = 1                                       // 8–9. sor: s grafikus
9. return L
```

Az algoritmust később irányított gráfokra [22, 31, 32, 41] is kiterjesztették.

2.2. Erdős–Gallai-algoritmus (EG)

Időrendben a következő eredmény Erdős Pál és Gallai Tibor alábbi szükséges és elégséges feltétele [20] volt.

Nemnegatív egészek adott $s = (s_1, \dots, s_n)$ sorozata esetén a sorozat első i elemét a sorozat s_i eleméhez tartozó *fejnek,* míg a többi elemét az s_i elemhez tartozó *faroknak* nevezzük. A fejelemek összegét H_i, míg a farokelemek összegét T_i jelöli $(i = 1, \dots, n)$. A $\sum_{k=i+1}^{n} \min(i, s_k)$ összeget pedig C_i-vel jelöljük és a farok *becsült kapacitásának* nevezzük. Ha egy s sorozatra H_n páros, akkor a sorozatot n*-párosnak,* egyébként n*-páratlannak* nevezzük.

2.2. Tétel. (Erdős, Gallai, [20]) *Ha $n \geq 1$, a $(0, 1, n)$-szabályos $(s_1, \dots, s_n)$ sorozat akkor és csak akkor $(0, 1, n)$-grafikus, ha*

$$H_n \quad \text{páros} \tag{1}$$

és

$$H_i \leq i(i-1) + C_i \quad (i = 1, \dots, n-1). \tag{2}$$

Bizonyítás. Lásd [17, 20, 73, 87]. □

A tétel alapgondolata az, hogy az első i csúcs fokait egyrészt ezen csúcsok közötti élekkel – ezekből legfeljebb $i(i-1)/2$ van – másrészt a nagyobb indexű

csúcsok fokaival lehet lekötni. A nagyobb indexű csúcsokra pedig az jellemző, hogy egyrészt legfeljebb i csúcs egy-egy fokát tudják lekötni, másrészt legfeljebb annyi fokot, mint a saját fokszámuk. A tétel szépségét az adja, hogy ezeknek a természetes szükséges feltételeknek az elégségességét is tartalmazza.

A 2.2. tételen alapul a következő Erdős–Gallai-algoritmus.

A szokásos változók mellett C az aktuális C_i-t jelöli.

2.2. Algoritmus. Erdős-Gallai(n, s)

```
1.  L = 0                                  // 1. sor: L kezdeti értékének beállítása
2.  H_1 = s_1                              // 2–4. sor: H elemeinek kiszámítása
3.  for i = 2 to n
4.      H_i = H_{i-1} + s_i
5.  if H_n páratlan                        // 5–6. sor: paritás ellenőrzése
6.      return 0
7.  for i = 1 to n − 1                     // 7–12. sor: s tesztelése
8.      C = 0                              // 7. sor: C kezdeti értékének beállítása
9.      for k = i + 1 to n                 // 8–9. sor: C frissítése
10.         C = C + min(i, s_k)
11.     if H_i − i(i − 1) > C              // 11. sor: szükséges feltétel ellenőrzése
12.         return L                       // 12. sor: s nemgrafikus
13. L = 1                                  // 13–14. sor: s grafikus
14. return L
```

Az Erdős-Gallai (röviden: EG) algoritmus memóriaigénye $\Theta(n)$. Bár ez a program csak ellenőriz, futási ideje a legjobb $\Theta(n)$ és a legrosszabb $\Theta(n^2)$ között változik. A közelmúltban Tripathi et al. [87] publikáltak a tételre konstruktív bizonyítást, amely grafikus bemenet esetén $\Theta(n^3)$ idő alatt egy megoldást is előállít.

A szabályos sorozatoknak aszimptotikusan a fele páros sorozat. Az 1. táblázathoz a $(0, 1, n)$-szabályos sorozatok számát a majd a 4. szakaszban szereplő (24) képlet alapján [1, 80], míg a $(0, 1, n)$-páros sorozatok számát az ugyancsak a 4. szakaszban következő 4.2. lemma alapján számítottuk [80]. A táblázat harmadik oszlopa a két számosság hányadosának gyors konvergenciáját szemlélteti $n = 1, \ldots, 38$ csúcs esetén.

3. Új pontos algoritmusok $(0, 1, n)$-gráfokhoz

Ebben a részben a klasszikus algoritmusok néhány gyorsított változatát mutatjuk be.

3.1. Nullamentes algoritmusok

Mivel a sorozatok végén lévő nullák izolált csúcsokat jelentenek, így azok nem befolyásolják, hogy az adott sorozat grafikus-e. Ezt a megfigyelést hasznosítja a következő állítás, amelyben p az s sorozat pozitív elemeinek a számát jelöli.

1. táblázat. A szabályos ($R(n)$) és a páros ($E(n)$) sorozatok száma, valamint ezen számok hányadosa ($E(n)/R(n)$).

n	$R(n)$	$E(n)$	$E(n)/R(n)$
1	1	1	1,0000000000000
2	3	2	0,6666666666667
3	10	6	0,6000000000000
4	35	19	0,5428571428571
5	126	66	0,5238095238095
6	462	236	0,5108225108225
7	1716	868	0,5058275058275
8	6435	3235	0,5027195027195
9	24310	12190	0,5014397367339
10	92378	46252	0,5006819805581
11	352716	176484	0,5003572279114
12	1352078	676270	0,5001708481315
13	5200300	2600612	0,5000888410284
14	20058300	10030008	0,5000427753100
15	77558760	38781096	0,5000221251603
16	300540195	150273315	0,5000107057227
17	1166803110	583407990	0,5000055150693
18	4537567650	2268795980	0,5000026787479
19	17672631900	8836340260	0,5000013755733
20	68923264410	34461678394	0,5000006701511
21	269128937220	134564560988	0,5000003432481
22	1052049481860	526024917288	0,5000001676328
23	4116715363800	2058358034616	0,5000000856790
24	16123801841550	8061901596814	0,5000000419280
25	63205303218876	31602652961516	0,5000000213918
26	247959266474052	123979635837176	0,5000000104862
27	973469712824056	486734861612328	0,5000000053420
28	3824345300380220	1912172660219260	0,5000000026224
29	15033633249770520	7516816644943560	0,5000000013342
30	59132290782430712	29566145429994736	0,5000000006558
31	232714176627630544	116357088391374032	0,5000000003333
32	916312070471295267	458156035385917731	0,5000000001640
33	3609714217008132870	1804857108804606630	0,5000000000833
34	14226520737620288370	7113260369393545740	0,5000000000410
35	56093138908331422716	28046569455332514468	0,5000000000208
36	221256270138418389602	110628135071477978626	0,5000000000103
37	873065282167813104916	436532641088444120108	0,5000000000052
38	3446310324346630677300	1723155162182151654600	0,5000000000026

3.1. Következmény. *Ha $n \geq 1$, az $(s_1, \ldots, s_n)$ $(0,1,n)$-szabályos sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha $s_1 = 0$, vagy az $(s_1, \ldots, s_p)$ sorozat $(0,1,p)$-grafikus.*

Bizonyítás. Ha a sorozatnak van pozitív eleme, akkor az állítás a Havel-Hakimi, illetve az Erdős-Gallai következménye, de közvetlenül is adódik: a nullák ugyanis nem segítenek a pozitív fokszámok párosításánál, ugyanakkor nem okoznak önálló igényt sem. □

Az ezen a tulajdonságon alapuló megvalósítást nullamentes Erdős-Gallai (EGn), illetve nullamentes Havel-Hakimi (HHn) algoritmusnak nevezzük.

3.2. Rövidített Erdős–Gallai-algoritmus (EGr)

H_i maximális értéke szabályos sorozat esetén $n(n-1)$, ezért a 2.2. tételben szereplő (2) egyenlőtlenség $i = n$ esetén biztosan teljesül, így felesleges ellenőrizni.

Ennél is hasznosabb a következő lemma. Tripathi és Vijay 2003-as cikkében [86] szerepel az az észrevétel, hogy az Erdős–Gallai-tételben a (2) egyenlőtlenséget elég csak addig ellenőrizni, amíg $H_i > i(i-1)$ teljesül.

3.1. Lemma. (Tripathi és Vijay [86]) *Ha $n \geq 1$, a $(0,1,n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha*

$$H_n \quad \textit{páros}$$

és

$$H_i - \min(H_i, i(i-1)) \leq \sum_{k=i+1}^{n} \min(i, s_k) \quad (i = 1, 2, \ldots, h),$$

ahol

$$h = \max_{1 \leq k \leq n} (k \mid k(k-1) < H_k).$$

Bizonyítás. Ha $i(i-1) \geq H_i$, akkor (2) bal oldala nempozitív, ezért az egyenlőtlenség biztosan teljesül, így felesleges ellenőrizni. □

Például a száz darab ötöst tartalmazó sorozat esetén (2) jobb oldalát az Erdős–Gallai-algoritmus szerint kilencvenkilencszer, míg a rövidített Erdős–Gallai-algoritmus szerint csak hatszor kell kiszámítani. A javításnak a várható futási időre gyakorolt hatását a 7. részben vizsgáljuk.

A 3.1. lemmán alapuló algoritmust rövidített Erdős–Gallai-algoritmusnak (EGr) nevezzük.

3.3. Ugró Erdős–Gallai-algoritmus (EGu)

Az ismétlődő elemeket összevonva egy szabályos $(s_1, \ldots, s_n)$ sorozat $(s_{i_1}^{e_1}, \ldots, s_{i_q}^{e_q})$ alakban is felírható, ahol $s_{i_1} > \cdots > s_{i_q}$, $e_1, \ldots, e_q \geq 1$, és $e_1 + \cdots + e_q = n$. Legyen $g_j = e_1 + \cdots + e_j$ $(j = 1, \ldots, q)$.

Az s_i elemet az s sorozat *ugró* elemének nevezzük, ha $i = n$, vagy $1 \le i \le n-1$, és $s_i > s_{i+1}$. Ekkor az ugró elemek az $s_{g_1}, \dots, s_{g_q}$ elemek. Az ugró (vagy ellenőrző) elemeket $c_1 = s_{g_1}, \dots, c_q = s_{g_q}$ módon jelöljük.

Tripathi és Vijai 2003-ban a [86] cikkben az Erdős–Gallai-tétel következő, lényeges gyorsítást lehetővé tevő változatát is bizonyították.

3.1. Tétel. (Tripathi, Vijay [86]) *A $(0,1,n)$-szabályos $s = (s_1, \dots, s_n)$ sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha*

$$H_n \quad \textit{páros}$$

és

$$H_{g_i} - g_i(g_i - 1) \le \sum_{k=g_i+1}^{n} \min(g_i, s_k) \quad (i = 1, \dots, q).$$

Bizonyítás. Lásd [86]. □

A következő program (EGu) az Erdős–Gallai-algoritmusnak a 3.1. lemma, valamint a 3.3. tétel alapján gyorsított változatát mutatja be.

A szokásos változók mellett itt $H = (H_1, \dots, H_n)$, ahol H_i s első i elemének az összege; p s pozitív elemeinek a száma, és s_{p+1} segédváltozó annak eldöntéséhez, hogy s_p ugró elem-e.

3.1. Algoritmus. Erdős–Gallai-ugró(n, s, L)

```
1.  p = n                                   // 1–3. sor: nullamentesítés
2.  while s_p = 0
3.      p = p − 1
4.  H_1 = s_1                               // 4–8. sor: paritás ellenőrzése
5.  for i = 2 to p
6.      H_i = H_{i−1} + s_i
7.  if H_p páratlan
8.      return 0
9.  s_{p+1} = 0                             // 9–19. sor: fej igényének ellenőrzése
10. i = 1
11. while i ≤ p ∧ i(i − 1) < H_i
12.     while s_i == s_{i+1}
13.         i = i + 1
14.     E = 0
15.     for j = i + 1 to p
16.         E = E + min(j, s_j)
17.     if H_i > i(i − 1) + E
18.         return 0
19.     i = i + 1
20. return 1                                // 20. sor: s grafikus
```

Ennek az algoritmusnak a futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n^2)$ között változik.

Megjegyezzük, hogy az ellenőrzést elég a $(q-1)$-edik ugrópontig folytatni.

A 2. táblázat azt mutatja, hogy $n = 3, \ldots, 15$ csúcs esetén EGu hány menet alatt tudja kizárni a nem $(0,1,n)$-grafikus sorozatokat a $(0,1,n)$-szabályos sorozatok tesztelése során. $f_i(n) = f_i$ azoknak az n hosszúságú, nem $(0,1,n)$-grafikus sorozatoknak a száma, amelyek pontosan i tesztelési menetet igényeltek. A táblázat minden sorára jellemző, hogy a maximális menetszám körülbelül $\frac{n}{2}$.

2. táblázat. A $(0,1,n)$-szabályos nem $(0,1,n)$-grafikus sorozatok eloszlása $n = 3, \ldots, 15$ csúcsra aszerint, hogy az EGu algoritmus hány menet alatt tudja őket kizárni.

n/i	$R(n) - G(n)$	f_1	f_2	f_3	f_4	f_5	f_6	f_7
3	6	6						
4	24	24						
5	95	91	4					
6	360	338	22					
7	1 374	1 262	102	10				
8	5 222	4 729	409	84				
9	19 949	17 841	1 587	487	34			
10	76 362	67 645	6 025	2 294	398			
11	293 368	257 779	22 802	9 820	2 825	142		
12	1 129 961	986 274	86 292	39 745	15 554	2 096		
13	4 363 985	3 787 213	327 644	156 295	74 542	17 632	659	
14	16 891 448	14 586 597	1 248 368	605 592	327 404	111 872	11 615	
15	65 516 140	56 330 831	4 774 119	2 331 442	1 363 561	599 615	113 316	3 256

A 3. táblázat tartalmazza a $(0,1,n)$-szabályos, -grafikus és -nemgrafikus sorozatok számát, valamint az EGu algoritmus számára a nemgrafikus, grafikus és összes sorozat kiszűréséhez szükséges menetek átlagos számát $n = 3, \ldots, 15$ csúcs esetén. A táblázatban szereplő X', Y' és Z' hatékonysági jellemzők definícióját a (15), (16) and (17) képletek tartalmazzák. Figyelemre méltó, hogy n növekedtével az X' és Z' értékek csökkennek, míg az Y' értékek nőnek.

3.4. Lineáris Erdős–Gallai-algoritmus (EGl)

A következő Erdős–Gallai-Lineáris algoritmus kihasználja, hogy az s bemeneti sorozat monoton. Ennek köszönhetően a C_i kapacitásokat minden i-re konstans időben meg tudja határozni, azaz nincs szüksége arra, hogy a megfelelő farok elemeit egyenként megvizsgálja. A gyors számolás kulcsa a *súlypontokat* tartalmazó $w(s)$ sorozat.

Adott s sorozat esetén legyen $w(s) = (w_0, \ldots, w_{n-1})$, ahol $i > s_1$ esetén $w_i = 0$, egyébként pedig w_i az s sorozat legnagyobb indexű olyan elemének indexe, amelyik legalább akkora, mint i.

3. táblázat. A $(0,1,n)$-szabályos és -grafikus sorozatok száma, valamint az Erdős–Gallai-ugró algoritmus által az $n = 3,\dots,15$ hosszú sorozatok vizsgálata során végzett tesztek átlagos száma.

n	$R(n)$	$G(n)$	X'	Y'	Z'
3	10	4	0,3333333333	0,5833333333	0,4333333333
4	35	11	0,2500000000	0,5909090909	0,3571428571
5	126	31	0,2084210526	0,6064516129	0,3063492063
6	462	102	0,1768518519	0,6192810458	0,2745310245
7	1 716	342	0,1555416927	0,6219715957	0,2485014985
8	6 435	1 213	0,1388117579	0,6267518549	0,2307886558
9	24 310	4 361	0,1259433778	0,6312007949	0,2165821107
10	92 378	16 016	0,1154618789	0,6336476024	0,2053021282
11	352 716	59 348	0,1068633005	0,6357110908	0,1958472384
12	1 352 078	222 117	0,0996191461	0,6373495350	0,1879565503
13	5 200 300	836 315	0,0934514246	0,6386612700	0,1811323607
14	20 058 300	3 166 852	0,0881205642	0,6397881871	0,1752191576
15	77 558 760	120 426 20	0,0834688999	0,6407780422	0,1700028030

Az s sorozat s_i elemének ellenőrzésekor két eset van: ha $i > w_i$, akkor a C_i kapacitás egyszerűen számítható: $H_n - H_i$, mivel a farok minden s_j elemének hozzájárulása csak s_j.

Ha viszont $i \le w_i$, akkor a C_i-t definiáló szummát két részre bontjuk: az első részhez a farok azon s_j kezdő elemeinek hozzájárulása tartozik, amelyekre teljesül $s_j \ge i$, a második részhez pedig a többi elem. Legyen

$$q(s) = q = \max_{1\le i\le n}\{i \mid i(i-1) \le H_i\}.$$

3.2. Tétel. (Iványi, Lucz, Móri, Sótér [35]) *Ha $n \ge 1$, az $s = (s_1,\dots,s_n)$ $(0,1,n)$-szabályos sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha*

$$H_n \quad \textit{páros}, \tag{3}$$

továbbá

$$H_i \le i(k-1) + H_n - H_k \quad (i = 1,\dots,q), \tag{4}$$

ahol

$$k(s) = k = \begin{cases} w_i, & \text{ha } i \le w_i, \\ i, & \text{ha } i > w_i. \end{cases} \tag{5}$$

Bizonyítás. Megmutatjuk, hogy a tételben szereplő feltétel ekvivalens a 2.2. tétel feltételeivel.

A (3) feltétel pontosan megegyezik az (1) feltétellel.

Ha $i \le w_i$, akkor

$$H_i \le i(i-1) + (w_i - i + 1)i + H_n - H_{w_i} \tag{6}$$

és ha $i > w_i$, akkor

$$H_i \le i(i-1) + H_n - H_i. \tag{7}$$

Ha (6) jobb oldalán kiemeljük i-t, akkor a

$$H_i \le iw_i + H_n - H_{w_i}$$

egyenlőtlenséget kapjuk. Ha a (4) egyenlőtlenségbe (5) alapján behelyettesítjük k-t, akkor az $i \le w_i$ esetben a (6), az $i > w_i$ esetben pedig a (7) egyenlőtlenséget kapjuk. □

A következő program a 3.2. tétel alapján adott n-re tetszőleges n-szabályos sorozatról eldönti, hogy grafikus-e. A program futási ideje minden sorozatra $O(n)$. Érdemes megjegyezni, hogy akár a bemenő sorozat rendezettségétől is eltekinthetünk, mivel a sorozat elemei egész számok és mindegyik a $[0, n-1]$ intervallumba esik, így szükség esetén $O(n)$ idő alatt rendezni tudjuk a sorozatot.

A szokásos változók mellett H_i az éppen tesztelt s első i elemének az összege, w a kurrens s_i-hez tartozó súlypont; y pedig az ellenőrzés egyszerűsítéséhez használt változó (az aktuális s_i *vágópontja* (w és i maximuma)).

3.2. Algoritmus. Erdős–Gallai-lineáris(n, s, L)

```
1.  H_1 = s_1                                   // 1. sor: H_1 beállítása
2.  for i = 2 to n                              // 2–3. sor: H további elemeinek számítása
3.      H_i = H_{i-1} + s_i
4.  if H_n páratlan                             // 4–6. sor: paritás ellenőrzése
5.      L = 0
6.      return
7.  w = n                                       // 7. sor: súlypont beállítása
8.  for i = 1 to n − 1                          // 8–16. sor: s elemeinek tesztelése
9.      while w > 1 ∧ s_w < i                   // 8–10. sor: aktuális súlypont számítása
10.         w = w − 1
11.     y = max(i, w)                           // 11. sor: aktuális vágópont számítása
12.     if H_i > i(y − 1) + H_n − H_y
13.         L = 0                               // 13–14. sor: nemgrafikus s elutasítása
14.         return L
15. L = 1                                       // 15–16. sor: s grafikus
16. return L
```

3.2. Következmény. *A $(0, 1, n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozatról az* EGl *algoritmus $\Theta(n)$ idő alatt dönti el, hogy $(0, 1, n)$-grafikus-e.*

Bizonyítás. A 1–3. sorok $\Theta(n)$ időt igényelnek. Mivel a w súlypontot legfeljebb n-szer frissítjük, ezért a 4–16. sorok időigénye $O(n)$, így az algoritmus futási ideje $\Theta(n)$. □

3.5. Gyors Erdős–Gallai-algoritmus (EGgy)

Tripathi és Vijai a [86] cikkben az Erdős–Gallai-tétel következő, lényeges gyorsítást lehetővé tevő változatát is bizonyították.

Az ismétlődő elemeket gyakoriságuk segítségével tömörítve a $(0,1,n)$-szabályos $(s_1,\dots,s_n)$ sorozat felírható az $(s_{i_1}^{e_1},\dots,s_{i_q}^{e_q})$ alakban, ahol $s_{i_1}<\dots<s_{i_q}$; $e_1,\dots,e_q\geq 1$ és $e_1+\dots+e_q=n$. Legyen $g_j=e_1+\dots+e_j$ $(j=1,\ \dots,\ q)$.

Az s_i elemet az s *ugró pontjának* nevezzük, ha $i=n$, vagy $1\leq i\leq n-1$ és $s_i>s_{i+1}$. Ekkor az ugró pontok az $s_{g_1},\dots,s_{g_q}$ elemek.

3.3. Tétel. (Tripathi, Vijay [86]) *Az $s=(s_1,\dots,s_n)$ szabályos sorozat akkor és csak akkor grafikus, ha*

$$H_n \quad \textit{páros}$$

és

$$H_{g_i}-g_i(g_i-1)\leq\sum_{k=c_i+1}^{n}\min(g_i,s_k)\quad (i=1,\dots,q).$$

Bizonyítás. Lásd [86]. □

Megjegyezzük, hogy az ellenőrzést elég a $(q-1)$-edik ugró pontig folytatni.

A következő tétel – EGe és EGu előnyeit egyesítve – a tesztelési idő további csökkentését teszi lehetővé.

3.4. Tétel. *A $(0,1,n)$-szabályos $s=(s_1,\dots,s_n)$ sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha igaz az, hogy*

$$H_n \quad \textit{páros}$$

és

$$H_{g_i}\leq\begin{cases}H_n-H_{g_i}+g_i(g_i-1), & \textit{ha}\ \ w_i\leq g_i\\ H_n-H_{w_i}+g_i(w_i-1), & \textit{ha}\ \ w_i>g_i\end{cases}\quad (i=1,\dots,q-1). \tag{8}$$

Bizonyítás. A csak az ugró pontokban való tesztelés elégségességét Tripathi és Vijay [86] már bebizonyították. A tételben megadott feltétel ezeket az ellenőrzéseket végzi el, kihasználva a sorozat elemeinek monoton csökkenését, azaz a

$$\sum_{k=g_i+1}^{n}\min(g_i,s_k)$$

összeget nem számolja újra minden esetben, pontosabban nem ebben a formában végzi el a számítást, hanem explicit módon.

A kifejezés értéke a (9) formában adható meg, mégpedig azért, mert a sorozat monotonitása garantálja, hogy a $k\leq w_i$ esetén a $\min(i,s_k)$ kifejezés értéke i, míg $k>w_i$ esetén s_k. Ebből következik, hogy

$$\sum_{k=g_i+1}^{n-1}\min(g_i,s_k)=\begin{cases}H_n-H_{g_i}, & \text{ha}\ \ w_i\leq g_i\\ H_n-H_{w_i}+g_i(w_i-g_i), & \text{ha}\ \ w_i>g_i.\end{cases} \tag{9}$$

4. táblázat. Az ugró és a gyors Erdős–Gallai-algoritmusok egy sorozatra jutó átlagos műveletigénye.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EGu	4	12	16	21	26	32	37	43	49	56	63	70	77	85
$\frac{\text{EGu}}{n}$	2,0	4,0	4,0	4,2	4,3	4,6	4,6	4,8	4,9	5,1	5,3	5,4	5,5	5,7
EGgy	12	15	17	19	21	23	25	27	29	31	33	35	37	39
$\frac{\text{EGgy}}{n}$	6,0	5,0	4,3	3,8	3,5	3,3	3,1	3,0	2,9	2,8	2,8	2,7	2,6	2,6

Az eddigiek alapján az eredeti feltételt átírhatjuk a következő alakba:

$$H_{g_i} - g_i(g_i - 1) \leq \begin{cases} H_n - H_{g_i}, & \text{ha } w_i \leq g_i \\ H_n - H_{w_i} + g_i(w_i - g_i), & \text{ha } w_i > g_i. \end{cases} \tag{10}$$

A (10) egyenlőtlenséget átrendezve megkapjuk a (8) egyenlőtlenséget. □

A most megadott tétel alapján megvalósított EGgy algoritmus és az eddigi legjobb (ugró Erdős–Gallai) algoritmus sorozatonkénti átlagos műveletszámait, valamint a sorozat egyetlen elemére jutó átlagos műveletszámot tartalmazza a 4. táblázat. Itt az átlag azt jelenti, hogy a vizsgált sorozatokhoz tartozó műveletszámok összegét elosztottuk a sorozatok számával.

A táblázatból leolvasható, hogy az átlagos műveletszám a lineáris algoritmus esetében kevesebb, mint fele annyi, mint az ugró algoritmus esetében és az n érték növelésével minden lépésben ugyanannyival növekszik. Az utóbbi azért fontos, mert így az n növelésével lépésről lépésre nagyobb az új algoritmussal elért gyorsulás a korábbiakhoz képest. Az utóbbi kijelentés azonban nem meglepő, ha figyelembe vesszük, hogy a korábbi ismert algoritmusok négyzetesek, míg az új algoritmus lineáris futási idejű. Jól látható, hogy a régi módszer esetén a sorozatok egy eleméhez tartozó átlagos műveletszám az n érték növekedésével együtt nőtt, az új módszernél azonban ez a szám lépésről lépésre csökken.

A 3.4. tétel feltételeit ellenőrzi a következő algoritmus.

3.3. Algoritmus. Erdős–Gallai-gyors(n, s, L)

```
1.  H_1 = s_1                                  // 1. sor: H_1 beállítása
2.  for i = 2 to n                             // 2–4. sor: H további értékeinek számítása
3.      H_i = H_{i-1} + s_i
4.  if H_n páratlan                            // 4–7. sor: paritás ellenőrzése
5.      L = 0                                  // 5–6. sor: nemgrafikus sorozat elutasítása
6.      return
7.  w = n                                      // 7. sor: súlypont kezdeti értéke
8.  for i = 1 to n − 1                         // 8–26. sor: sorozat tesztelése
9.      if s_i == s_{i+1}                      // 9–11 sor: ugrópont tulajdonság ellenőrzése
```

```
10.          continue                                  // 10. sor: nem ugrópont átlépése
11.      while (w > 1) ∧ (s_w ≤ i)                     // 11–12. sor: súlypont frissítése
12.          w = w − 1
13.      if w < i                                  // 13–16. sor: súlypont ugrópont előtt
14.          if H_i > H_n − H_i + i(i − 1) 14–18. sor: tétel feltételének ellenőrzése
15.              L = 0                 // 15–16. sor: nemgrafikus sorozat elutasítása
16.              return
17.      else if H_i > H_n − H_w + i(w − 1) // 17–19. sor: súlypont ugrópont után
18.              L = 0                 // 18–19. sor: nemgrafikus sorozat elutasítása
19.              return
20.  L = 1                                      // 20–21. sor: grafikus sorozat elfogadása
21.  return L
```

3.5. Tétel. *Az Erdős–Gallai-gyors algoritmus műveletigénye lineáris.*

Bizonyítás. Az 1. sor időigénye $O(1)$, a 2–3. soré $\Theta(n)$, a 4–7. soré $O(1)$, a 8–20. soré $O(n)$, a 21–22. soré pedig $O(1)$. így az algoritmus teljes műveletigénye $\Theta(n)$. □

3.6. Eltoló Havel–Hakimi-algoritmus (HHe)

Havel és Hakimi eredeti tételének természetes algoritmikus megfelelőjét HHr-nek (rendező Havel–Hakimi) nevezzük, mert a tétel természetes alkalmazása minden menetben igényli a redukált bemenet rendezését.

A tétel alapján olyan megvalósítás is lehetséges, hogy a fokszámok redukálását a sorozat monotonitását megőrizve végezzük. Ekkor az eltoló Havel–Hakimi-algoritmust (HHe) kapjuk.

3.7. Paritásos Havel–Hakimi-algoritmus (HHp)

Érdekes gondolat az Erdős–Gallai- és a Havel–Hakimi-feltételek együttes alkalmazása úgy, hogy először s paritását vizsgáljuk, és csak a páros bemenetekre alkalmazzuk a rendszerint négyzetes futási idejű rekurzív ellenőrzést. Ezzel ugyan elveszítjük a nullamentes Havel–Hakimi azon jó tulajdonságát, hogy legjobb esetben konstans idő alatt lefut, viszont cserébe megkapjuk azt, hogy a várható futási idő jelentősen csökken.

3.8. Lineáris Havel–Hakimi-tesztelő algoritmus (HHl)

Az EGl algoritmusban kulcsszerepe volt az s_i elemhez tartozó w_i súlypontnak [35], amely $i > s_1$ esetén 0, egyébként a legnagyobb olyan k index, amelyre igaz, hogy $s_k \geq bi$ (természetesen ez az egyenlőtlenség a $(0, 1, n)$ -gráfokra – azaz a $b = 1$ esetben – az $s_k \geq i$ egyenlőtlenségre egyszerűsödik). Most azonban a súlypont mellett az r_i maradék is fontos: ez azt adja meg, hány felhasználatlan fok maradt az előző, s_{i-1} elem feldolgozása során.

A súlypont arra is alkalmas, hogy a Havel–Hakimi-algoritmus lineáris változatában fontos szereplő legyen. Az algoritmus alapja a következő tétel.

3.6. Tétel. *Ha $n \geq 1$, az $(s_1, \ldots, s_n)$ $(0,1,n)$-szabályos sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha*

$$s_1 < w_1, \tag{11}$$

és

$$s_i \leq w_i + r_{i-1} \quad (i = 2, \ldots, n-1), \tag{12}$$

ahol

$$w_i = \max(k \geq 0 \mid s_k \geq i) \quad (i = 1, \ldots, n), \tag{13}$$

és

$$r_i = w_i + r_{i-1} - s_i \quad (i = 1, \ldots, n). \tag{14}$$

Bizonyítás. (13) szerint w_i megadja, hogy az s sorozatban hány olyan s_k elem van, amely legalább i. Ezért a Havel–Hakimi-algoritmus első menetének végrehajtásához szükséges és elégséges (11), a további rekurzív menetekhez pedig (12), azaz az, hogy az s_i fokszám feldolgozásához elég legyen az előző menet felhasználatlan maradéka (r_i), plusz az adott menetben felhasználhatóvá váló fokok (w_i). □

A Havel–Hakimi-lineáris pszeudokódjában $r = (r_1, \ldots, r_n)$, ahol r_i az s_i-hez tartozó maradék; $w = (w_1, \ldots, w_n)$, ahol w_i az i indexhez tartozó súlypont, és $H = (H_1, \ldots, H_n)$, ahol H_i az s sorozat első i elemének összege.

3.4. Algoritmus. Havel–Hakimi-lineáris(n, s, L)

```
1.  if s_1 == 0                          // 1–3. sor: nullákból álló sorozat elfogadása
2.      L = 1
3.      return L
4.  if s_{s_1+1} == 0                    // 4–6. sor: s_1 tesztelése konstans idő alatt
5.      L = 0
6.      return L
7.  w_1 = n                              // 7-12. sor: az első súlypont és tartalék számítása
8.  j = n
9.  while s_j ≤ 1 ∧ j > 0
10.     w_1 = w_1 − 1
11.     j = j − 1
12. r_1 = w_1 − 1 + s_1
13. for i = 2 to n − 1                   // 13–21. sor: s tesztelése
14.     j = w_{i−1}                      // 14–17. sor: új súlypont kiszámítása
15.     while s_j ≤ i ∧ j > 0
16.         w_i = w_i − 1
17.         j = j − 1
18.     if w_i ≥ i                       // 18–22. sor: s grafikus?
19.         if s_i > w_i + r_{i−1}
20.             L = 0                    // 20–21. sor: s nem grafikus
```

```
21.              return L
22.          r_i = w_i − 1 + r_{i−1} − s_i          // 22. sor: r_i frissítése
23.      if w_i < i
24.          if s_i > w_i + r_{i−1}
25.              L = 0                  // 25–26. sor: s nem grafikus
26.              return L
27.          r_i = w_i + r_{i−1} − s_i              // 27. sor: r_i frissítése
28.  L = 1                                  28–29. sor: s grafikus
29.  return L
```

3.7. Tétel. *A Havel–Hakimi-lineáris algoritmus futási ideje legjobb esetben* $\Theta(1)$, *legrosszabb esetben* $\Theta(n)$.

Bizonyítás. Az 1–6. sorok időigénye $O(1)$, és például a (0^n) bemenetre a program a 3. sorban megáll, ezért a legjobb futási idő $O(1)$. A 7–11. sorok időigénye $\Theta(n)$. Mivel a súlypontok számítása legfeljebb n csökkentést igényel, a 12–29. sorok időigénye $O(n)$, ezért a legrosszabb eset $\Theta(n)$. □

3.9. Példák

3.1. Példa. Legyen az első példában $n = 4$ és $s = \left(3^3, 1\right)$. Az 1–12. sorok szerint $r_1 = 0$. Ha $i = 2$, akkor $w_i = 3$, és a 19. sor feltétele nem teljesül, ezért s *nem* $(0, 1, 4)$*-grafikus.*

3.2. Példa. A következő példában $n = 7$ és $s = \left(5, 3^2, 2, 1^3\right)$. Az 1–12. sorokban azt kapjuk, hogy $w_1 = 7$ és $r_1 = 1$. Ha $i = 2$, akkor $w_i = 4$, a 19. sor feltétele nem teljesül, és a 22. sor szerint $r_2 = 1$. Ha $i = 3$, akkor $w_i = 3$, és nem teljesül a 24. sor feltétele. Ha $i = 4$, akkor $w_i = 1$, és most sem teljesül a 24. sor feltétele. Ha $i = 5$, akkor teljesül a 09. sor $s_j \leq 1$ feltétele, és ezért s $(0, 1, 7)$*-grafikus.*

3.3. Példa. Legyen $n = 7$ és $s = \left(5, 4, 1^5\right)$. Erre a sorozatra $r_1 = 1$, és ha $i = 2$, akkor $w_i = 2$, ezért a 24. sor feltétele teljesül, így s *nem* $(0, 1, 7)$*-grafikus.*

3.4. Példa. Utolsó példánkban legyen $n = 7$ és $s = \left(5^2, 4, 3^4\right)$. Az első 12 sor szerint $r_1 = 1$. Ha $i = 2$, akkor $w_i = 7$ és $r_2 = 1$. Ha $i = 3$, akkor $w_3 = 7$ és $r_3 = 2$. Ha $i = 4$, akkor teljesül a 15. sor $s_i \leq 1$ feltétele, ezért s $(0, 1, 7)$*-grafikus.*

A következő táblázatokban bemutatjuk, hogyan oszlanak meg a kizárt grafikus és nemgrafikus sorozatok az egyes menetek között. Azt is jellemezzük, hogy átlagosan hány meneten át kell egy grafikus, illetve nemgrafikus sorozatot a kizárásáig tesztelni, és azt is, hogy a menetek hányadrészét fordítjuk átlagosan egy sorozat tesztelésére.

Az 5. táblázat a HHl által az i-edik ($i = 1, \ldots, 11$) menetben kiszűrt nem $(0, 1, n)$-grafikus sorozatok számát mutatja $n = 1, \ldots, 11$ csúcs esetén.

5. táblázat. HHl i-edik ($i = 1, \ldots, 11$) menetében a $(0, 1, n)$-szabályos sorozatok közül kiszűrt nem $(0, 1, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10	11
1	0										
2	1	0									
3	6	0	0								
4	22	2	0	0							
5	85	8	2	0	0						
6	311	35	12	2	0	0					
7	1169	128	58	17	2	0	0				
8	4369	488	239	100	24	2	0	0			
9	16524	1805	942	471	173	32	2	0	0		
10	62650	6800	3601	2021	956	289	43	2	0	0	
11	239008	25571	13677	8147	4561	1877	470	55	2	0	0

6. táblázat. HHl i-edik ($i = 1, \ldots, 11$) menetében a $(0, 1, n)$-szabályos sorozatok közül kiszűrt grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10	11
1	1										
2	2	0									
3	1	3	0								
4	1	8	2	0							
5	1	16	12	2	0						
6	1	29	48	22	2	0					
7	1	47	130	127	35	2	0				
8	1	72	306	488	290	54	2	0			
9	1	104	618	1492	1475	591	78	2	0		
10	1	145	1158	3863	5757	3868	1112	110	2	0	
11	1	195	1998	8890	18440	18662	9053	1958	149	2	0

A 6. táblázat HHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 1, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

Legyen $n_i(a, b, n, \mathrm{A}) = n_i$, illetve $m_i(a, b, n, \mathrm{A}) = m_i$ az A algoritmus által az (a, b, n)-szabályos vagy (a, b, n)-páros sorozatok vizsgálata során az i-edik ($i = 1, \ldots, n$) menetben kizárt nemgrafikus, illetve grafikus sorozatok száma, továbbá legyen

$$N = \sum_{i=1}^{n-1} n_i \quad \textit{és} \quad M = \sum_{i=1}^{n-1} m_i,$$

$$X(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i n_i}{N},$$

$$Y(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i m_i}{M},$$

$$Z(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i(m_i + n_i)}{N + M},$$

$$X'(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i n_i}{N(n-1)}, \tag{15}$$

$$Y'(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i m_i}{M(n-1)}, \tag{16}$$

$$Z'(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i(m_i + n_i)}{(N + M)(n-1)}. \tag{17}$$

A 7. táblázat a HHl algoritmus hatékonyságát jellemzi $a = 0$, $b = 1$ és $n = 1, \ldots, 11$ csúcs esetén.

7. táblázat. HHl hatékonysági jellemzői $a = 0$, $b = 1$ és $n = 2, \ldots, 11$ csúcs esetén.

n/jellemző	X	Y	Z	X'	Y'	Z'
2	1,000000000	1,000000000	1,000000000	1,000000000	1,000000000	1,000000000
3	1,000000000	1,750000000	1,300000000	0,500000000	0,875000000	0,650000000
4	1,083333333	2,454545455	1,514285714	0,361111111	0,818181818	0,504761905
5	1,126315789	3,032258065	1,595238095	0,281578947	0,758064516	0,398809524
6	1,180555556	3,588235294	1,712121212	0,236111111	0,717647059	0,342424242
7	1,220524017	4,111111111	1,796620047	0,203420670	0,685185185	0,299436674
8	1,262734584	4,629843364	1,897435897	0,180390655	0,661406195	0,271062271
9	1,299062610	5,140793396	1,988235294	0,162382826	0,642599175	0,248529412
10	1,335323852	5,650162338	2,083407305	0,148369317	0,627795815	0,231489701
11	1,368874588	6,157056683	2,174534186	0,136887459	0,615705668	0,217453419

Az 7. táblázat 11. sorában található $X'(0, 1, 11) = 0,136887459$ és $Y'(0, 1, 11) = 0,615705668$. Eszerint 11 csúcs esetén a nemgrafikus sorozatok kiszűréséhez átlagosan a menetek 14%-ára, míg a grafikus sorozatok kiszűréséhez

átlagosan 62%-ára van szükség, ahonnan az következik, hogy az összes szűréshez átlagosan a menetek 22%-át kell végrehajtani.

Érdemes megjegyezni, hogy Tripathi és Vijay ugrópontokról szóló tétele a HHl algoritmus gyorsítására is felhasználható.

4. Általános leszámlálási eredmények

Eddig például Avis és Fukuda [2], Barnes és Savage [3, 4], Burns [14], Erdős és Moser [59], Frank, Savage and Sellers [25], Kleitman és Winston [42], Rødseth, Sellers, Tverberg [70], Ruskey et al. [71], Simion [75], Stanley [83], Winston és Kleitman [90] publikáltak foksorozatok leszámlálására vonatkozó eredményeket. Az általunk vizsgált sorozatok számával kapcsolatos eredmények találhatók Sloane és Ploffe [76], valamint Stanley [82] könyvében és a *The On-Line Encyclopedia of Integer Sequences* című honlapon [78, 79, 80] is.

Ha $l,\ m$ és u egész számok, továbbá $l \le u$ és $m \ge 1$, akkor az $s = (s_1, \ldots, s_n)$ (l, u, m)-korlátos sorozatok $B(l, u, m)$ száma

$$B(l, u, m) = (u - l + 1)^m. \tag{18}$$

A (18) képlet közvetlen adódik abból, hogy az s sorozatnak mind az m eleme $u - l + 1$ lehetséges értéket vehet fel.

Az is közvetlenül belátható, hogy ha $l,\ m$ és u egész számok, továbbá $l \le u$ és $m \ge 1$, akkor az (l, u, m)-szabályos sorozatok $R(l, u, m)$ száma

$$R(l, u, m) = \binom{m + u - l}{m}. \tag{19}$$

Legyen ugyanis az $s = (s_1, \ldots, s_m)$ (l, u, m)-szabályos sorozat esetén $s' = (s'_1, \ldots, s'_m)$, ahol $s'_i = s_i + m - i$. A lehetséges s és s' sorozatok halmazai között kölcsönösen egyértelmű kapcsolat áll fenn. A különböző s' sorozatok száma pedig annyi, ahányféleképpen a különböző $l,\ l+1,\ \ldots,\ u+m-1$ számok – azaz $u + m - l$ szám – közül m számot ki tudunk választani.

Ha $l = 0,\ u = n - 1$ és $m = n$, akkor az

$$R(0, n - 0, n) = R(n) = \binom{2n - 1}{n} \tag{20}$$

alakot kapjuk.

A szimulációs vizsgálatok elemzésénél (is) hasznos a szabályos és a páros sorozatok számát megadó függvények tulajdonságainak ismerete.

4.1. Lemma. *Ha $n \ge 1$, akkor*

$$\frac{R(n+2)}{R(n+1)} > \frac{R(n+1)}{R(n)}, \tag{21}$$

$$\lim_{n\to\infty}\frac{R(n+1)}{R(n)}=4, \tag{22}$$

továbbá

$$\frac{4^n}{\sqrt{4\pi n}}\left(1-\frac{1}{2n}\right)<R(n)<\frac{4^n}{\sqrt{4\pi n}}\left(1-\frac{1}{8n+8}\right). \tag{23}$$

Bizonyítás. A (20) egyenlőség alapján

$$\frac{R(n+2)}{R(n+1)}=\frac{(2n+3)!(n+1)n!}{(n+2)!(n+1)!(2n+1)!}=\frac{4n+6}{n+2}=4-\frac{2}{n+2},$$

ahonnan (21) és (22) is közvetlenül adódik.

(23) belátásához felhasználjuk a Stirling-formula következő alakját [16]: ha $n\geq 1$, akkor

$$n!=\left(\frac{n}{e}\right)^n\sqrt{2\pi n}e^{\tau_n},$$

ahol

$$\frac{1}{12n+1}<\tau_n<\frac{1}{12n}.$$

□

1987-ben Ascher [1] a következő képletet vezette le a $(0,1,n)$-páros sorozatok $E(n)$ számára.

4.2. Lemma. (Ascher [1], Sloane and Plouffe [76]) *Ha $n\geq 1$, akkor a $(0,1,n)$-páros sorozatok $E(n)$ száma*

$$E(n)=\frac{1}{2}\left(\binom{2n-1}{n}+\binom{n-1}{\lfloor n\rfloor}\right). \tag{24}$$

Bizonyítás. Lásd [1, 76]. □

A (20) képlet és a 4.2. lemma egybevetése mutatja, hogy a páros és páratlan sorozatok számának nagyságrendje megegyezik, azonban több a páros sorozat, mint a páratlan. A 4.2. lemma alapján pontosan meg tudjuk adni $E(n)$ aszimptotikus nagyságrendjét.

4.3. Lemma. (Iványi, Lucz, Móri, Sótér [35]) *Ha $n\geq 1$, akkor*

$$\frac{E(n+2)}{E(n+1)}>\frac{E(n+1)}{E(n)},$$

$$\lim_{n\to\infty}\frac{E(n+1)}{E(n)}=4,$$

továbbá

$$\frac{4^n}{\sqrt{\pi n}}(1-\delta(n))<E(n)<\frac{4^n}{\sqrt{\pi n}}(1-\Delta(n)),$$

ahol $\delta(n)$ és $\Delta(n)$ monoton csökkenve nullához tartó sorozatok.

Bizonyítás. A bizonyítás hasonló a 4.1. lemma bizonyításához. □

Amint azt a következő állítás és az 1. táblázat is mutatja, az $E(n)/R(n)$ hányadosok sorozata monoton csökkenve $\frac{1}{2}$-hez tart.

4.1. Következmény. (Iványi, Lucz, Móri, Sótér [35]) *Ha* $n \geq 1$, *akkor*

$$\frac{E(n+1)}{R(n+1)} < \frac{E(n)}{R(n)}$$

és

$$\lim_{n\to\infty} \frac{E(n)}{R(n)} = \frac{1}{2}.$$

Bizonyítás. Lásd [35]. □

Bár az alapfeladatban nemnegatív elemekből álló sorozatok szerepelnek, algoritmusaink – a futási idő csökkentése érdekében – csak a sorozatok pozitív kezdőszeletét vizsgálják. Ennek várható hatását jellemzi a következő két állítás, amelyek a nullát tartalmazó sorozatok számát és a sorozatokban lévő nullák átlagos számát adják meg.

4.4. Lemma. *Ha* $n \geq 1$, *akkor a* $(0, 1, n)$*-szabályos sorozatok közül*

$$R_z(n) = \binom{2n-2}{n-1} = \frac{n}{2n-1} R(n).$$

tartalmaz legalább egy nullát.

Bizonyítás. A nullát tartalmazó $(0, 1, n)$-szabályos sorozatok halmaza kölcsönösen egyértelműen leképezhető a $(0, n-1, n)$-szabályos sorozatok halmazára. Az utóbbi halmaz elemszáma pedig (20) szerint

$$\binom{2n-2}{n-1} = \frac{(2n-2)!n}{n(n-1)!(2n-1)} = \frac{n}{2n-1}\binom{2n-1}{n} = \frac{n}{2n-1} R(n).$$

□

Egész számokból álló sorozat különböző elemeinek a számát az adott sorozat *szivárványszámának* nevezzük. Legyen $q_n(s)$ valószínűségi változó, amely egy véletlen $(0, 1, n)$-korlátos sorozat szivárványszámát jellemzi. $q_n(b)$ szivárványszámának várható értékét és szórását a következő állítás tartalmazza.

4.5. Lemma. (Iványi, Lucz, Móri, Sótér [35]) *Legyen* σ *egy véletlen* $(0, n-1, n)$*-korlátos sorozat és* $q_n(\sigma)$ *a szivárványszáma. Ekkor* σ $E[q_n(\sigma)]$ *várható értéke és*

$Var[q_n(\sigma)]$ szórása a következő:

$$\begin{aligned} E[q_n(\sigma)] &= n\left[1-\left(1-\frac{1}{n}\right)^n\right] = n\left(1-\frac{1}{e}\right)+O(1),\\ Var[q_n(\sigma)] &= n\left(1-\frac{1}{n}\right)^n\left[1-\left(1-\frac{1}{n}\right)^n\right]\\ &\quad + n(n-1)\left[\left(1-\frac{2}{n}\right)^n-\left(1-\frac{1}{n}\right)^{2n}\right]\\ &= \frac{n}{e}\left(1-\frac{2}{e}\right)+O(1). \end{aligned}$$

Bizonyítás. Lásd [35]. □

A következő állítás a k szivárványszámú $(0,n-1,n)$-szabályos sorozatok számát adja meg.

4.6. Lemma. (Iványi, Lucz, Móri, Sótér [35]) *Ha $1\le k\le n$ és $m\ge 1$, akkor a k szivárványszámú $(0,n-1,m)$-szabályos sorozatok $S(k,m,n)$ száma*

$$S(k,m,n)=\binom{n}{k}\binom{m-1}{k},\ k=1,\dots,n.$$

Bizonyítás. Lásd [35]. □

Eszerint a véletlen σ $(0,n-1,m)$-szabályos sorozatok $r_n(\sigma)$ szivárványszáma hipergeometriai eloszlású az $n+m-1$, n és m paraméterekkel. Legyen $\rho_n(\sigma)$ egy véletlen $(0,1,n)$-szabályos sorozat és $E[r_n(\sigma)]$, illetve $V[r_n(\sigma)]$ σ várható értéke, illetve szórása. Ekkor $\rho_n(\sigma)$ szivárványszámának várható értékét és szórását a következő állítás tartalmazza.

4.2. Következmény. (Iványi, Lucz, Móri, Sótér [35]) *Legyen ρ egy véletlen $(0,1,n)$-szabályos sorozat. Ekkor ρ $E[r_n(\rho)]$ várható értéke és $V[r_n(\rho)]$ szórása a következő:*

$$\begin{aligned} E[r_n(\rho)] &= \frac{n^2}{2n-1} = \frac{n}{2}+\frac{n}{4n-2} = \frac{n}{2}+O(1),\\ V[r_n(b)] &= \frac{n^2(n-1)}{2(2n-1)^2} = \frac{n}{8}+\frac{n}{128n^2-128n+32} = \frac{n}{8}+O(1). \end{aligned}$$

Bizonyítás. Lásd [35]. □

A pontos algoritmusokról szóló 3.1. részben beláttuk, hogy elég a $(0,1,n)$-páros sorozatok nullamentes prefixét megvizsgálni ahhoz, hogy eldöntsük, grafikus-e a vizsgált sorozat. Mivel a 4.4. lemma szerint a páros sorozatoknak aszimptotikusan csak nullmértékű hányada tartalmaz nullát (és ez a hányad a gyakorlat számára legérdekesebb n-ekre sem nagy), konkrét sorozatok vizsgálatánál nem jelentős az

időmegtakarítás. Amikor viszont az összes páros sorozatot elemezzük (az átlagos futási idő vagy $G(n)$ meghatározása érdekében), nagyon hasznos a következő lemma.

Legyen $G_z(n)$ a nullamentes grafikus n-páros sorozatok száma.

4.7. Lemma. (Iványi, Lucz, Móri, Sótér [35]) *Ha $n \geq 2$, akkor a $(0,1,n)$-grafikus sorozatok száma*

$$G(n) = G_z(n) + G(n-1).$$

Bizonyítás. A $(0,1,n)$-grafikus sorozatokban vagy $s_n = 0$, vagy $s_n > 0$. Az előbbiekben vagy $s_1 = n-1$, vagy $s_1 < n_1$. Ha $s_1 = n-1$ és $s_n = 0$, akkor az s sorozat biztosan nem grafikus, mert nincs benne elég pozitív elem. Az $s_1 < n-1$ és $s_n = 0$ tulajdonságú sorozatok $n-1$ hosszú fejei pontosan a $(0,1,n-1)$-grafikus sorozatok. □

A grafikus sorozatok $G(n)$ számának jellemzésével kapcsolatos kutatások ígéretes iránya a páros számok pozitív összeadandókra való felbontása, és annak vizsgálata, hogy az ilyen felbontások közül melyek $(0,1,n)$-grafikusak [3, 4, 14]. Ezek segítségével sikerült a grafikus sorozatok számára vonatkozó alábbi aszimptotikus korlátokat bizonyítani.

4.8. Lemma. (Burns [14]) *Léteznek olyan pozitív c és C állandók, hogy a $(0,1,n)$-grafikus sorozatok $G(n)$ száma a következő korlátok közé esik:*

$$\frac{4^n}{cn} < G(n) < \frac{4^n}{(\log n)^C \sqrt{n}}\ .$$

Bizonyítás. Lásd [14]. □

Nézzük meg, mit várhatunk a HHl algoritmus első hat sorától. Az algoritmus lehetséges bemenetei a $(0, n-1, n)$-szabályos sorozatok. Ezek $R(n)$ száma a (20) képlet szerint

$$R(n) = \binom{2n-1}{n}.$$

HHl első három sora kiszűri például azokat a sorozatokat, amelyek $(n-1)$-gyel kezdődnek, és nullával végződnek. Ezek száma (19) szerint

$$B(0, n-1, n-2) = \binom{2n-3}{n-2}.$$

Ezek közül a HHl által kiszűrt sorozatok $R_1(n)$ hányada

$$R_1(n) = \frac{\binom{2n-3}{n-2}}{\binom{2n-1}{n}} = \frac{2(2n-1)}{n} = \frac{1}{4} + \frac{1}{8n-4}\,.$$

HHl pontosan azokat a sorozatokat szűri ki, amelyek $(n-i)$-vel $(i = 1, \ldots, n-2)$ kezdődnek, és legalább i nullát tartalmaznak. Rögzített i-re az ilyen sorozatok

aszimptotikus részaránya $1/4^i$, úgy HHl aszimptotikusan a szabályos sorozatokból a

$$\sum_{i=1}^{\infty} \frac{1}{4^i} = \frac{1}{3}$$

összegnek megfelelő hányadot, azaz egy harmad részét szűri ki.

Mivel a grafikus sorozatok aszimptotikus sűrűsége nulla, ezért minden A pontos algoritmusra létezik egy $s_{1,A} + s_{2,A} + \cdots = 1$ sor (valószínűség-eloszlás), amelyben s_i az i-edik menetben kiszűrt hányad. Például $s_{1,A} = 1/3$ minden olyan pontos algoritmusra, amelyik első menetben a PT algoritmust (vagy annak valamilyen lassú változatát) használja – ilyen a HH és az EG is.

5. Tesztelő algoritmusok

Sorozatok megvalósíthatóságának vizsgálata során természetes észrevétel, hogy az s sorozat i-hez tartozó fejének H_i fokszám igényét részben belső (az adott fejen belüli), részben pedig külső (a fejnek megfelelő farokhoz tartozó) fokszámokkal elégítjük ki.

Először egy „pozitív", majd egy „paritásos", egy „binomiális" és végül egy „fejfelező" tesztelő/szűrő algoritmust mutatunk be.

5.1. Pozitív teszt

A farokban lévő nulla elemek nem növelik a farok párosítási lehetőségeit. Ez az észrevétel lehetővé teszi, hogy az i-edik elemhez tartozó farok foklekötési lehetőségeire (potenciáljára) T_i-nél pontosabb becslést adjunk. Ez a teszt a Havel–Hakimi-algoritmus első menetének megfelelő ellenőrzést végzi el. Legyen p az s sorozat pozitív elemeinek a száma.

5.1. Következmény. *Ha $n \geq 1$ és $s = (s_1, \ldots, s_n)$ $(0, 1, n)$-grafikus sorozat, akkor*

$$s_1 \leq p - 1, \quad \text{vagy} \quad s_1 = 0. \tag{25}$$

Bizonyítás. A (25) egyenlőtlenség azt a követelményt fejezi ki, amelyet a Havel–Hakimi-algoritmus az első iterációs menetben, illetve az Erdős–Gallai-algoritmus a (2) egyenlőtlenség $i = 1$ esetben való ellenőrzésével megvalósít. □

A 5.1. következményen alapuló tesztet a következő algoritmus végzi, amelyben p: a bemenetben lévő pozitív elemek száma.

5.1. Algoritmus. Pozitív teszt(n, s, L)

1. $L = 0$
2. $p = n$

3. **while** $s_p == 0$
4. $\quad p = p - 1$
5. **if** $s_1 > p - 1$
6. $\quad$ **return** L
7. $L = 2$
8. **return** L

Ennek az algoritmusnak a futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n)$ között változik.

Ennek az algoritmusnak a javított változata az alábbi Gyors teszt (Gyt) [54].

5.2. Algoritmus. Gyors teszt(n, s, L)

1. **if** $s_{s_1+1} == 0$
2. $\quad L = 0$
3. $\quad$ **return** L
4. $L = 2$
5. **return** L

A Gyors teszt ugyanazt az eredményt adja, mint Pozitív teszt, a futási ideje azonban mindig $\Theta(1)$.

5.2. paritás teszt

Első tesztünk az Erdős–Gallai-tétel első szükséges feltételén alapul. Nagyon hatékony teszt, mivel mind a korlátos, mind a szabályos sorozatoknak körülbelül fele páratlan sorozat, és a teszt ezekről lineáris idő alatt megállapítja, hogy biztosan nem grafikus sorozatok.

5.1. Lemma. *Ha $n \geq 1$ és s $(0, 1, n)$-grafikus sorozat, akkor*

$$H_n \quad \textit{páros.}$$

Bizonyítás. Egy egyszerű gráf minden éle kettővel növeli a fokszámok összegét. □

Ezt az állítást a 2.2. tétel következményeként is megkaphatjuk. A 5.1. lemmában javasolt tesztet a következő algoritmus végzi.

5.3. Algoritmus. Paritás teszt(n, s, L)

1. $L = 0$
2. $H_1 = 0$
3. **for** $i = 2$ **to** n
4. $\quad H_i = H_{i-1} + s_i$
5. **if** H_n páratlan
6. $\quad$ **return** L
7. $L = 2$
8. **return** L

Ennek az algoritmusnak a lépésszáma minden esetben $\Theta(n)$.

5.3. Binomiális teszt (Bt)

Harmadik tesztünk az Erdős–Gallai-tétel másik szükséges feltételének ötletét terjeszti ki. Lényege, hogy a fej igényének a fejen belül ki nem elégíthető részét a faroknak, a farok igényének belül ki nem elégíthető részét a fejnek kell kielégítenie, végül a teljes sorozat igényét a fej és a farok együttműködésével, valamint a fej és a farok belső éleivel kell kielégíteni. Az algoritmus nevét arról kapta, hogy a fej és a farok belső éleinek a számát egy-egy binomiális együttható segítségével becsüljük. Legyen p az s sorozat pozitív elemeinek a száma.

5.2. Lemma. *Ha $n \geq 1$ és s $(0, 1, n)$-grafikus sorozat, akkor*

$$2H_i \leq i(i-1) + T_i \quad (i = 1, \ldots, p). \tag{26}$$

Bizonyítás. A (26) egyenlőtlenség azt fejezi ki, hogy a fej H_i igényét a legfeljebb $i(i-1)$ belső lehetőség és a farok legfeljebb T_i kapacitása segítségével kell kielégíteni, ahol $T_I = H_n - H_i$. □

A 5.2. lemmában javasolt tesztet végzi el a következő program.

5.4. Algoritmus. Binomiális teszt(n, s, L)

```
1.  p = n
2.  while s_p == 0
3.      p = p − 1
4.  if p == 1
5.      L = 0
6.      return L
7.  H_1 = s_1
8.  for i = 2 to p
9.      H_i = H_{i−1} + s_i
10. for i = 1 to p
11.     if 2H_i > i(i − 1) + H_p
12.         L = 0
13.         return L
14. L = 1
15. return L
```

Az algoritmus azért kezdi s végénél p meghatározását, mert a 4.7. lemma szerint kevés nulla várható a sorozatokban.

Ennek az algoritmusnak a futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n)$ között változik.

Az eddigi szimulációs vizsgálatok szerint nagyon hatékony szűrő algoritmus. Aszimptotikus hatékonysága kulcsfontosságú az optimális tesztelő algoritmus futási ideje szempontjából.

Megjegyezzük, hogy Binomiális teszt $i = 1$ esetén elvégzi Pozitív teszt munkáját, ezért a Pozitív teszt algoritmusra nincs szükségünk. A várható futási idő szempontjából viszont a konstans idő alatt hatékony Gyors teszt hasznos lehet.

Felmerült, hogy a Binomiális teszt algoritmust is csak az ellenőrző pontokon alkalmazzuk, a szimulációs kísérletek azonban azt mutatták, hogy ezzel csökkenne az algoritmus hatékonysága.

n helyett p viszont gyengítené az algoritmust, mert például a rossz $(2, 2, 0)$ sorozatot *nem* szűrné ki. Ha azonban csak a páros nullamentes sorozatokat vizsgáljuk, a $(2, 2, 0)$ és hasonló sorozatokat egyetlen algoritmusunk sem kell tesztelnie (mert ezeket már a bemenő sorozatok előállítása során kiszűrjük).

5.4. Fej felezése (Ft)

Az s sorozat fokpárosító lehetőségeinek az eddigieknél pontosabb becslését kaphatjuk, ha a fejet két részre osztjuk. Legyen $\lfloor i/2 \rfloor = h_i$. Ekkor az $(s_1, \ldots, s_{h_i})$ sorozatot az i indexhez tartozó fej *elejének*, az $(s_{h_i+1}, \ldots, s_i)$ sorozatot pedig az i indexhez tartozó fej *végének* nevezzük.

5.3. Lemma. *Ha $n \geq 1$ és s $(0, 1, n)$-grafikus sorozat, akkor*

$$\begin{aligned} H_i \leq\ & \min(H_{h_i}, T_n - T_i, h_i(n-i)) \\ & + \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)) \\ & + \min(h_i(i-h_i), H_i) + 2\min\left(\binom{h_i}{2}, H_{h_i}\right) \\ & + 2\min\left(\binom{i-h_i}{2}, H_i - H_{h_i}\right) \quad (i = 1, \ldots, n), \end{aligned} \tag{27}$$

továbbá

$$\min(H_{h_i}, T_n - T_i, h_i(n-i)) + \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)) \leq T_i. \tag{28}$$

Bizonyítás. Legyen G az s sorozatot megvalósító G gráf. Ekkor az i indexhez tartozó fej H_i fokszámösszegét lekötő élek halmazát öt részhalmazra osztjuk: a fej eleje és a farok, a fej vége és a farok közötti, a fej két része közötti, valamint a fej részein belüli élekre. Az egyes részhalmazokba tartozó élek száma legyen rendre $X_{i,1}, \ldots, X_{i,5}$.

$X_{i,1}$ legfeljebb a fej elemeinek H_{h_i} összege, legfeljebb a farok elemeinek $T_n - T_i$ összege, és legfeljebb a fej elejéből és a farokból képezhető párok $h_{h_i}(n-i)$ szorzata lehet, azaz

$$X_{i,1} \leq \min(H_{h_i}, T_n - T_i, h_i(n-i)). \tag{29}$$

Hasonló gondolatmenettel kapjuk, hogy

$$X_{i,2} \leq \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)). \tag{30}$$

$X_{i,3}$ legfeljebb $h_i(i-h_i)$, és legfeljebb H_i, ezért

$$X_{i,3} \leq \min(h_i(i-h_i), H_i). \tag{31}$$

$X_{i,4}$ legfeljebb $\binom{h_i}{2}$, és legfeljebb H_{h_i}, így

$$X_{i,4} \le \min\left(\binom{h_i}{2}, H_{h_i}\right), \tag{32}$$

míg $X_{i,5}$ legfeljebb $\binom{i-h_i}{2}$, és legfeljebb $H_i - H_{h_i}$, ahonnan

$$X_{i,5} \le \min\left(\binom{i-h_i}{2}, H_i - H_{h_i}\right). \tag{33}$$

Az is követelmény, hogy a farok részei együtt nem léphetik túl a farok kapacitását, azaz teljesüljön

$$X_{i,1} + X_{i,2} \le T_i. \tag{34}$$

A (29), (30), (31), (32) és (33) egyenlőtlenségeket összegezve azt kapjuk, hogy

$$H_i \le X_{i,1} + X_{i,2} + X_{i,3} + 2X_{i,4} + 2X_{i,5}. \tag{35}$$

Az $X_{i,4}$ és $X_{i,5}$ előtti kettes konstansok azt veszik figyelembe, hogy a fej részein belüli hasznos élek kettővel járulnak hozzá a fej H_i igényének kielégítéséhez.

Ha a (29), (30), (31), (32) és (33) egyenlőtlenségeket a (35) egyenlőtlenségbe helyettesítjük, akkor (27) adódik, míg (34) ekvivalens a (28) egyenlőtlenséggel. □

A 5.3. lemmában javasolt tesztet a következő algoritmus végzi, melynek egyedi paraméterei egyrészt $T = (T_1, \dots, T_n)$, ahol T_i az s sorozat utolsó $n - i$ elemének összege, másrészt $X = (X_1, X_2, X_3, X_4, X_5)$: X_j a fej vége $X_{i,j}$ paraméterének aktuális értéke.

5.5. Algoritmus. Fejfelező teszt(n, s, H, T, p, L)

1. **for** $i = 2$ **to** $n - 1$
2. $\quad h = \lfloor i/2 \rfloor$
3. $\quad X_1 = \min(H_h, T_n - T_i, h(n-i))$
4. $\quad X_2 = \min(H_i - H_h, T_n - T_i, (i-h)(n-i))$
5. $\quad X_3 = \min(h(i-h), H_i)$
6. $\quad X_4 = \min\left(\binom{h_i}{2}, H_{h_i}\right)$
7. $\quad X_5 = \min\left(\binom{i-h_i}{2}, H_i - H_{h_i}\right)$
8. $\quad$ **if** $H_i > X_1 + X_2 + X_3 + 2X_4 + 2X_5$ vagy $X_1 + X_2 > T_i$
9. $\quad\quad L = 0$
10. $\quad\quad$ **return** L
11. $L = 1$
12. **return** L

Az algoritmus futási ideje legjobb esetben $\Theta(1)$, legrosszabb esetben $\Theta(n)$.

Hasonló módon a farok felezése is további sorozatok kiszűrését tenné lehetővé, de a szimulációs kísérletek szerint ez nem csökkentené a várható futási időt.

6. Közelítő algoritmusok hatékonysága és futási ideje

A tesztek elemzésénél a szabályos és páros sorozatokat vettük alapul. A páros sorozatok halmaza a legkisebb olyan halmaz, melynek elemszámát explicit képlettel meg tudjuk adni. Az $n-1 \geq b_i \geq 1$ feltételeknek eleget tevő *n-korlátos sorozatok* halmazának elemszámát is könnyű megadni, de ezen halmazok elemszáma túl gyorsan nő n növekedtével. A szabályos sorozatok elemzéséhez szerencsére nem kell *minden* korlátos sorozatot előállítani: elegendő a szabályos sorozatokat előállítani, és a rájuk vonatkozó hatékonysági jellemzőket a nekik megfelelő gyakoriságokkal súlyozni. Például egy azonos elemekből álló *homogén* szabályos sorozatnak egyetlen korlátos sorozat felel meg, míg a különböző elemekből álló $(n, n-1, \ldots, 1, 0)$ „szivárvány" sorozatnak $n!$ különböző korlátos sorozat felel meg.

Az alapvető pontos algoritmusokat kétféle módon próbáljuk gyorsítani (azaz várható futási idejüket csökkenteni). Az egyik út, hogy csökkentjük az általuk elvégzendő ellenőrzések számát. A másik út pedig az, hogy gyors (lineáris) előtesztekkel igyekszünk a rossz sorozatok jelentős részét kiszűrni, hogy csak a lehetséges bemenetek kis hányadánál legyen szükség a viszonylag lassú, de pontos alapalgoritmusokra.

Az első típusú javításra példa az Erdős–Gallai-algoritmus ugrása. A második típusra pedig példa a Havel–Hakimi-algoritmus kiegészítése előzetes paritásvizsgálattal, valamint az Erdős–Gallai-algoritmus kiegészítése nullamentesítéssel.

A futási idők csökkentése érdekében *minden* algoritmus csak a páros, nullamentes sorozatokat vizsgálta.

Adott A algoritmusnak az n hosszúságú szabályos sorozatokra vonatkozó hatékonyságát az A algoritmus által kizárt n hosszúságú sorozatok és az ugyanolyan hosszúságú szabályos sorozatok számának hányadosával jellemezzük. Ezt a hányadost $E_{\mathrm{A}}(n)$-nel jelöljük, és az A algoritmus n hosszúságú sorozatokra vonatkozó *hatékonyságának* nevezzük.

A következő közelítő algoritmusokat vizsgáljuk:

1) Nullamentesítő teszt (Nt);

2) Binomiális teszt (Bt);

3) Fejfelező teszt (Ft).

A 8. táblázat a nullamentes binomiális és a nullamentes faroktesztelt sorozatok számát, továbbá a (0,1,n)-grafikus sorozatok számát és a grafikus sorozatok száma szomszédos n helyeken felvett értékei hányadosát tartalmazza $n = 1, \ldots, 29$ csúcs esetén.

A 9. táblázat azt jellemzi, hogy a vizsgált közelítő algoritmusok a szabályos sorozatoknak milyen hányadát szűrik ki. A táblázat a nullamentes páros sorozatok száma $(E_z(n))$ mellett tartalmazza a nullamentes binomiális $(B_z(n))$, a nullamentes faroktesztelt $(F_z(n))$ és a grafikus sorozatok $(G(n))$ számának, valamint a szabályos sorozatok számának hányadosát.

8. táblázat. A nullamentes binomiális $(B_z(n))$, nullamentes faroktesztelt $(F_z(n))$ $(0,1,-n)$-szabályos sorozatok száma, valamint a $(0,1,n)$-grafikus sorozatok száma (G_n) és a grafikus sorozatok halmazának szomszédos n helyeken felvett számosságai hányadosa $(G(n+1))/G(n)$ $n=1,\ldots,29$ csúcs esetén.

n	$B_z(n)$	$F_z(n)$	$G(n)$	$G(n+1)/G(n)$
1	**1**	**0**	**1**	2,000000
2	**2**	**2**	**2**	2,000000
3	**4**	**4**	**4**	2,750000
4	**11**	**11**	**11**	2,818182
5	**31**	**31**	**31**	3,290323
6	103	**102**	**102**	3,352941
7	349	344	**342**	3,546784
8	1256	1230	1213	3,595218
9	4577	4468	4361	3,672552
10	17040	16582	16016	3,705544
11	63944	62070	59348	3,742620
12	242218	234596	222117	3,765200
13	922369	891852	836315	3,786674
14	3530534	3409109	3166852	3,802710
15	13563764	13082900	12042620	3,817067
16	52283429	50380684	45967479	3,828918
17	202075949	194550002	176005709	3,839418
18	782879161	753107537	675759564	3,848517
19	3039168331	2921395019	2600672458	3,856630
20	11819351967	11353359464	10029832754	3,863844
21			38753710486	3,870343
22			149990133774	3,876212
23			581393603996	3,881553
24			2256710139346	3,886431
25			8770547818956	3,890907
26			34125389919850	3,895031
27			132919443189544	3,897978
28			518232001761434	3,898843
29			2022337118015338	

A 10. táblázat a Binomiális teszt és a Fejfelező teszt algoritmusok futási idejét adja meg másodpercben és műveletszámban $n = 1, \ldots, 20$ csúcsra.

Ha $n = 2$, akkor (20) szerint $R(n) = \binom{3}{2} = 3$ $(0, 1, n)$-szabályos sorozat van: $(1, 1)$, $(1, 0)$ és $(0, 0)$. Az n hosszúságú páros sorozatok számát $E(n)$-nel jelöljük. Ezzel a jelöléssel $E(2) = 2$. A Binomiális teszt által elfogadott, n hosszúságú sorozatok számát $B(n)$-nel jelölve $B(2) = 2$. Az n hosszúságú grafikus sorozatok számát jelöljük $G(n)$-nel. Ekkor $G(2) = 2$, és a Binomiális teszt hibája (hatékonysága) $R_{Bt}(2) = 2/2 = 1$.

Ha $n = 3$, akkor a szabályos sorozatok száma $R(n) = 10$. Ezek közül a (2,2,2), (2,2,0), (2,1,1), (2,0,0), (1,1,0) és (0,0,0) páros, azaz $E(3) = 6$. Ezek közül a Binomiális teszt kizárja a (2,2,0) és (2,0,0) sorozatokat, így $B(3) = 4$. A megmaradt 4 sorozat grafikus, így $F(3) = G(3) = 4$.

Ha $n = 4$, akkor a szabályos sorozatok száma $R(4) = 35$. Ezek közül 19 a páros, és a következő 11 grafikus: (3,3,3,3), (3,3,2,2), (3,2,2,1), (3,1,1,1,), (2,2,2,2), (2,2,2,0), (2,2,1,1), (2,1,1,0), (1,1,1,1), (1,1,0,0) és (0,0,0,0). A 19 páros sorozat közül a Binomiális teszt is kizárja azt a nyolc sorozatot, amelyeket az Erdős–Gallai kizárna, így $B(4) = F(4) = G(4) = 11$.

Az $R(5) = 126$ szabályos sorozat közül $E(5) = 66$ a páros, ezek között pedig $B(5) = 31$ a binomiális. Ezek a sorozatok mind grafikusak, azaz $F(5) = G(5) = 31$.

Az $R(6) = 462$ szabályos sorozat közül $E(6) = 236$ a páros, amelyek között $B(6) = 103$ binomiális sorozat van. A Binomiális teszt a 102 grafikus sorozat mellett az (5,5,3,3,3,1) rossz sorozatot is elfogadja. Ezek szerint a legfeljebb 5 hosszúságú sorozatokra nézve a Binomiális teszt hibátlanul kiszűri a nem grafikus sorozatokat, a 6 hosszú sorozatokra azonban már csak közelítő algoritmus. A Fejfelező teszt ezzel a sorozattal is megbirkózik, ezért $F(6) = G(6) = 102$.

Az $R(7) = 1716$ szabályos sorozat között $E(6) = 868$ a páros, melyek közül $B(7) = 376$ a binomiális. A binomiális sorozatok között még 34 rossz van, melyek közül a Pozitív teszt a 27 grafikus sorozat mellett a következő 7 rosszat is elfogadja: $(6, 6, 6, 4, 4, 4, 2)$, $(6, 6, 5, 4, 4, 4, 1)$, $(6, 6, 4, 4, 4, 3, 1)$, $(6, 6, 4, 3, 3, 3, 1)$, $(6, 6, 3, 3, 3, 2, 1)$, $(6, 5, 3, 3, 3, 1, 1)$, $(5, 5, 3, 3, 3, 1, 0)$. A következő Fejfelező teszt ezek közül a $(6, 6, 4, 3, 3, 3, 1)$ kivételével mindet kiszűri, így $F(7) = 343$. A cikkben nem ismertetett Farokfelező teszt $i = 4$ mellett legfeljebb $8 + 2$ fokot tud lekötni a fej eleje és a farok részei között, legfeljebb további $4 + 0$ fokot a fej vége és a farok részei között, legfeljebb további 8 fokot a fej két része között, és két fokot a fej elején belül. Ez azonban összesen csak $10 + 4 + 8 + 2 = 24$ fok, ami kevesebb a sorozat $H_7 = 26$ összes fokszámánál. Tehát a Farokfelező teszt a 7 hosszú bemenetek közül $T(7) = 342$ sorozatot fogad el, így $G(7) = 342$.

A 8. táblázatban minden sorban a pontos értékeket félkövéren írtuk. Eszerint $n \leq 4$ esetén $B(n) = G(n)$, azaz a Binomiális teszt ugyanannyi sorozatot fogad el, mint a pontos algoritmusok. $n > 4$ esetén egyre nő a Binomiális teszt hibája: $n = 5$ esetén még csak egyetlen páros sorozatról nem ismeri fel, hogy nemgrafikus, $n = 6$ esetén már hatszor hibázik.

A Pozitív teszt $n = 5$-ig hibátlan, a Fejfelező teszt $n = 6$-ig, a Farokfelező teszt pedig $n = 7$-ig.

9. táblázat. A nullamentes párossorozatok száma, továbbá a nullamentes binomiális/szabályos, nullamentes fejtesztelt/szabályos és grafikus/szabályos számarányok.

n	$E_z(n)$	$E_z(n)/R(n)$	$B_z(n)/R(n)$	$F_z(n)/R(n)$	$G(n)/R(n)$
1	0	0,000000	1,000000	1,000000	1,000000
2	1	0,333333	0,666667	0,666667	0,666667
3	2	0,300000	0,400000	0,400000	0,400000
4	9	0,257143	0,314286	0,314286	0,314286
5	28	0,230159	0,246032	0,246031	0,246032
6	110	0,238095	0,222943	0,220779	0,220779
7	396	0,231352	0,203380	0,200466	0,199301
8	1519	0,236053	0,195183	0,191142	0,188500
9	5720	0,235335	0,188276	0,183793	0,179391
10	21942	0,237524	0,184460	0,179502	0,173375
11	83980	0,238098	0,181290	0,175977	0,168260
12	323554	0,239301	0,179145	0,173508	0,164278
13	1248072	0,240000	0,177368	0,171500	0,160821
14	4829708	0,240784	0,176014	0,169960	0,157882
15	18721080	0,241379	0,174884	0,168684	0,155271
16	72714555	0,241946	0,173965	0,167634	0,152950
17	282861360	0,242424	0,173188	0,166738	0,150844
18	1101992870	0,242860	0,172533	0,165972	0,148926
19	4298748300	0,243243	0,171970	0,165306	0,147158
20	16789046494	0,243590	0,171486	0,164725	0,145521
21					0,143997
22					0,142569
23					0,141228
24					0,139961
25					0,138762
26					0,137625
27					0,136542
28					0,135509
29					0,134521

10. táblázat. A Binomiális teszt (Bt) és a Fejfelező teszt (Ht) futási ideje másodpercben és a műveletek számával megadva $n = 1, \ldots, 20$ csúcs esetén.

n	Bt, s	Bt, művelet	Ft, s	Ft, művelet
1	0	14	0	15
2	0	41	0	43
3	0	180	0	200
4	0	716	0	815
5	0	2 918	0	3 321
6	0	11 918	0	13 675
7	0	48 952	0	56 299
8	0	201 734	0	233 182
9	0	831 374	0	964 121
10	0	3 426 742	0	3 988 542
11	0	14 107 824	0	16 469 036
12	0	58 028 152	0	67 929 342
13	0	238 379 872	0	279 722 127
14	0	978 194 400	1	1 150 355 240
15	2	4 009 507 932	3	4 724 364 716
16	6	16 417 793 698	13	19 379 236 737
17	26	67 160 771 570	51	79 402 358 497
18	106	274 490 902 862	196	324 997 910 595
19	423	1 120 923 466 932	798	1 328 948 863 507
20	1 627	4 573 895 421 484	3 201	5 429 385 115 097

Az 1. táblázatban $R(n)$ értéke $n = 23$-ig az OEIS A001700 sorozata [78], $E(n)$ értéke $n = 23$-ig az OEIS A005654 sorozata [80], a 8. táblázatban $G(n)$ értéke pedig $n = 23$-ig az OEIS A0004251-es sorozata [79]. A többi értéket mi határoztuk meg: $R(24), \ldots, R(38)$, $E(24), \ldots, E(38)$, valamint $B(n)$ és $F(n)$ értékek nem szerepelnek az OEIS-ben.

Ebben a cikkben elsősorban a soros algoritmusokkal kapott eredményekről számolunk be.

A témakörben vannak párhuzamos eredmények is [60, 63, 74, 81]. Saját párhuzamos eredményeinket a 10. részben ismertetjük.

7. Pontos algoritmusok futási ideje

A következő pontos algoritmusokat vizsgáljuk:
1) HHr: Rendező Havel–Hakimi-algoritmus.
2) HHe: Eltoló Havel–Hakimi-algoritmus.

3) EG: Erdős–Gallai-algoritmus.
4) EGu: Erdős–Gallai-algoritmus ugrásokkal.
5) EGl: Erdős–Gallai-algoritmus ugrásokkal lineárisan.

A pontos algoritmusok sorozatonkénti átlagos futási idejét n függvényében mikromásodpercben a 11. táblázat tartalmazza $n = 1, \ldots, 15$ csúcsra. A sorozatok előállításához szükséges műveleteket beszámítottuk.

11. táblázat. Az elvégzett műveletek száma n függvényében a HHr, HHe, EG, EGu, és EGl algoritmusok esetén.

n	HHr	HHe	EG	EGu	EGl
1	10	15	87	-	-
2	40	61	119	12	37
3	231	236	267	116	148
4	1 170	1 052	946	551	585
5	5 969	4 477	4 000	2 677	2 339
6	31 121	20 153	18 206	12 068	9 539
7	157 345	88 548	82 154	54 184	38 984
8	784 341	393 361	372 363	238 813	160 126
9	3 628 914	1 726 484	1 666 167	1 666 167	656 575
10	17 345 700	7 564 112	7 418 447	4 552 276	2 692 240
11	80 815 538	32 895 244	32 737 155	19 680 986	11 018 710
12	385 546 527	142 460 352	143 621 072	84 608 529	45 049 862
13	1 740 003 588	613 739 913	626 050 861	362 141 061	183 917 288
14	8 066 861 973	2 633 446 908	2 715 026 827	1 543 745 902	750 029 671
15	36 630 285 216	11 254 655 388	11 717 017 238	6 557 902 712	3 055 289 271

A 11. táblázat második és harmadik oszlopának összehasonlítása azt mutatja, hogy HHe lényegesen gyorsabb, mint HHr, különösen ha n nő. A negyedik és ötödik oszlop összehasonlítása azt mutatja, hogy a futási idő lényegesen csökken, ha csak az ugró pontokban kell az elemeket tesztelni. Végül az utolsó három oszlop együtt a lineáris algoritmusnak a négyzetesekkel szembeni előnyét jelzi.

A 12. táblázat az Erdős–Gallai-lineáris futási idejét tartalmazza másodpercben és az elvégzett műveletek számával megadva, továbbá az egy páros sorozatra jutó amortizált műveletszámot.

A 12. táblázat legérdekesebb adatai az utolsó oszlopban vannak. Azt mutatják, hogy a műveletek számát osztva a vizsgált sorozatok hosszával és számával monoton csökkenő sorozatot kapunk (lásd [71]).

A 13. táblázat a $(0, 1, n)$-grafikus sorozatok első elem szerinti eloszlását mutatja $n = 1, \ldots, 12$ csúcs esetén. Ezek az adatok hasznosak az Erdős–Gallai-leszámláló algoritmus tervezéséhez (a feladat szeletekre osztásához).

A 13. táblázatban azt látjuk, hogy a gyakoriságok $n = 6$-tól nőnek $(n - 2)$-ig, és az utolsó pozitív érték kisebb, mint az utolsó előtti.

12. táblázat. Az Erdős–Gallai-lineáris algoritmus teljes és amortizált futási ideje másodpercben és a műveletek számában

n	$E(n)$	$T(n)$, s	$Op(n)$	$T(n)/E(n)/n$, s	$Op(n)/E(n)/n$
2	2	0	37	0	9.25000000000
3	6	0	148	0	8.22222222222
4	19	0	585	0	7.69736842105
5	66	0	2 339	0	7.08787878788
6	236	0	9 539	0	6.73658192090
7	868	0	38 984	0	6.41606319947
8	3 235	0	160 126	0	6.18724884080
9	12 190	0	656 575	0	5.98464132714
10	46 252	0	2 692 240	0	5.82080774885
11	176 484	0	11 018 710	0	5.67587378511
12	676 270	0	45 049 862	0	5.55126675243
13	2 600 612	0	183 917 288	0	5.44005937537
14	10 030 008	1	750 029 671	0.00000000712149	5.34132654018
15	38 781 096	5	3 055 289 271	0.00000000859525	5.25219687963
16	150 273 315	23	12 434 367 770	0.00000000956590	5.17156346504
17	583 407 990	79	50 561 399 261	0.00000000796537	5.09797604337
18	2 268 795 980	297	205 439 740 365	0.00000000727258	5.03056202928

13. táblázat. A $(0, 1, n)$-grafikus sorozatok eloszlása s_1 szerint, $n = 1, \ldots, 12$ csúcs esetén

n/s_1	0	1	2	3	4	5	6	7	8	9	10	11
1	1											
2	1	1										
3	1	1	2									
4	1	1	4	4								
5	1	2	7	10	11							
6	1	3	10	22	35	31						
7	1	3	14	34	78	110	102					
8	1	4	18	54	138	267	389	342				
9	1	4	23	74	223	503	968	1352	1213			
10	1	5	28	104	333	866	1927	3496	4895	4361		
11	1	5	34	134	479	1356	3471	7221	12892	17793	16016	
12	1	6	40	176	661	2049	5591	13270	27449	47757	65769	59348

8. $(0, b, n)$-gráfok

Ebben a részben a klasszikus tételek $(0, b, n)$-gráfokra való kiterjesztésével foglalkozunk.

8.1. Erdős–Gallai-tétel és Chungphaisan tétele

1974-ben Chungphaisan [18] mind az Erdős–Gallai-tételt, mind pedig a Havel–Hakimi-tételt kiterjesztette $(0, b, n)$-gráfokra. Az EG-tétel kiterjesztése a következő.

8.1. Tétel. (Chungphaisan [18]) *Legyen $n \geq 1$. A $(0, b(n-1), n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0, b, n)$-grafikus, ha*

$$\sum_{i=1}^{n} s_i \quad \textit{páros}$$

és

$$\sum_{i=1}^{j} s_i - bj(j-1) \leq \sum_{k=j+1}^{n} \min(bi, s_k) \quad (j = 1, \ldots, n-1).$$

Bizonyítás. Lásd [18]. □

A tételen alapuló algoritmus legrosszabb esetben négyzetes időt igényel. A következő állítás lehetővé teszi, hogy a $(0, b, n)$-szabályos sorozatokat legrosszabb esetben $\Theta(n)$ idő alatt teszteljük.

8.2. Tétel. (Iványi, [34]) *Ha $n \geq 1$, a $(0, b, n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0, b, n)$-grafikus, ha*

$$H_n \quad \textit{páros}$$

és

$$H_i > bi(y_i - 1) + H_n - H_y \quad (i = 1, \ldots, n-1),$$

ahol

$$y_i = \max(i, w_i) \quad (i = 1, \ldots, n-1).$$

Bizonyítás. Lásd [34]. □

A következő Chungphaisan–Erdős–Gallai-lineáris algoritmus (ChEGl) – amely az EGl-algoritmus természetes általánosítása – $O(n)$ idő alatt eldönti, hogy egy $(0, b, n)$-szabályos sorozat $(0, b, n)$-grafikus-e.

8.1. Algoritmus. Chungphaisan–Erdős–Gallai-lineáris(n, s, b, L)

Bemenet. n: csúcsok száma ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: $(0, b, n)$-szabályos sorozat;

b: a gráf két csúcsa között megengedett élek maximális száma.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozók. i: ciklus változó;
$w = (w_1, \dots, w_n)$: w_i az i indexhez tartozó súlypont.

```
1.  H_1 = s_1                                  // 1 sor: H_1 kezdeti értékének beállítása
2.  for i = 2 to n − 1                         // 2–3. sor: H további elemeinek számítása
3.      H_i = H_{i−1} + s_i
4.  if H_n páratlan                            // 4–6. sor: paritás ellenőrzése
5.      L = 0                                  // 5–6. sor: páratlan sorozat elutasítása
6.      return
7.  w = n                                      // 7. sor: első súlypont értékének beállítása
8.  for i = 1 to n − 1                         // 8–16. sor: s tesztelése
9.      while s_w < ib és w > 0
10.         w = w − 1
11.     y = max(i, w)
12.     if H_i > bi(y − 1) + H_n − H_y
13.         L = 0
14.         return L                           // 14. sor: s nem grafikus
15. L = 1                                      // 15–16. sor: s grafikus
16. return L
```

8.3. Tétel. (Iványi, [34]) ChEGl *futási ideje minden esetben* $\Theta(n)$.

Bizonyítás. A 1–6. sorok végrehajtása $\Theta(n)$ időt igényel. Mivel w szigorúan monoton csökken a program végrehajtása során, ezért a 7–14. sorok $O(n)$ időt igényelnek, így az algoritmus futási ideje minden esetben $\Theta(n)$. □

Legyen $b = 3$ és $s = (13, 10, 5, 5, 4, 1)$. $H_6 = 38$ páros. Ha $i = 1$, akkor $w_i = y = 5$ és a 11. sor feltétele $(13 \leq 3 \cdot 1 \cdot (5 - 1))$ nem teljesül. Ha $i = 2$, akkor viszont $w_i = y = 2$ és a feltétel teljesül $(23 > 3 \cdot 2 \cdot (2 - 1)) + 5 + 5 + 4 + 1)$, ezért s *nem* $(0, 3, 6)$*-grafikus.*

Maradjon b 3, de s-et változtassuk meg: legyen $s' = (13, 10, 5, 5, 4, 3)$. Az előző példához képest a futás során az első változás az, hogy amikor $i = 2$, akkor $23 \leq 3 \cdot 2 \cdot (2 - 1) + 5 + 5 + 4 + 3$, és így a 11. sorban lévő feltétel nem teljesül, és ugyanez az eredmény $i = 3,\ 4$ és 5 esetén is, ezért s' $(0, 3, 6)$*-grafikus.*

A 14. táblázat az (a, b, n)-szabályos és (a, b, n)-grafikus sorozatok számát tartalmazza $n = 1, \dots, 11$ csúcs, valamint $a = 0$ és $b = 1$, $a = 0$ és $b = 2$, $a = 2$ és $b = 5$ esetén. A szabályos sorozatok számát a (20) képlettel, az (a, b, n)-grafikus sorozatok számát pedig a Chungphaisan–Erdős–Gallai–lineáris algoritmussal határoztuk meg. Az utolsó oszlop elemeinek meghatározásánál hasznosítottuk a 9.1. következményt.

A következő táblázatokban bemutatjuk, hogyan oszlanak meg a kizárt grafikus és nemgrafikus sorozatok az egyes menetek között. Azt is jellemezzük, hogy átlagosan hány meneten át kell egy grafikus, illetve nemgrafikus sorozatot a kizárásáig

14. táblázat. Az (a, b, n)-szabályos és (a, b, n)-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs, valamint $a = 0$ és $b = 1$, $a = 0$ és $b = 2$, $a = 2$ és $b = 5$ esetén.

n	$R(0,1,n)$	$G(0,1,n)$	$R(0,2,n)$	$G(0,2,n)$	$R(2,3,n)$	$G(2,5,n)$
1	1	1	1	1	1	1
2	3	2	6	3	10	4
3	10	4	35	10	84	23
4	35	11	210	52	715	189
5	126	31	1287	283	6188	1582
6	462	102	8008	1706	54264	13583
7	1716	342	50388	10436	480700	122345
8	6435	1213	319770	65370	4292145	1092573
9	24310	4361	2042975	413111	38567100	9816598
10	92378	16016	13123110	2633537	348330136	88680716
11	352716	59348	84672315	16882153	3159461968	804480107

15. táblázat. ChEGL i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt nem $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	0									
2	3	0								
3	22	3	0							
4	132	26	2	0						
5	824	164	31	4	0					
6	5 084	1 026	276	75	3	0				
7	31 902	6 288	2 018	829	111	5 0				
8	201 366	39 090	13 282	7 231	1 837	203	4	0		
9	1 281 918	244 833	84 340	53 594	20 681	4 259	298	6	0	
10	8 207 232	1 548 774	529 578	365 461	183 262	59 726	8 709	470	5	0
11	52 819 163	9 866 545	3 331 910	2 385 963	1 404 590	632 058	155 070	17 213	660	7

tesztelni, és azt is, hogy a menetek hányadrészét fordítjuk átlagosan egy sorozat tesztelésére.

A 15. táblázat a ChEGl i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt nemgrafikus sorozatok számát tartalmazza $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

A 16. táblázat a ChEGl i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt $(0, 2, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

16. táblázat. ChEGl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	1									
2	2	0								
3	1	9	0							
4	1	7	42	0						
5	1	10	29	224	0					
6	1	14	49	183	1 297	0				
7	1	18	70	345	1 143	7 658	0			
8	1	23	97	559	2 326	7 262	46 489	0		
9	1	28	125	846	4 038	15 927	46 074	286 007	0	
10	1	34	159	1 191	6 520	29 629	107 724	295 609	1 779 026	0
11	1	40	193	1 624	9 668	50 663	213 399	728 610	1 900 061	11 154 877

A 17. táblázat a ChEGl algoritmus hatékonyságát jellemzi $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

17. táblázat. ChEGl hatékonysági jellemzői $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

n/jellemző	X	Y	Z	X'	Y'	Z'
2	1, 000000000	1, 000000000	1, 000000000	1, 000000000	1, 000000000	1, 000000000
3	1, 120000000	1, 900000000	1, 342857143	0, 560000000	0, 950000000	0, 671428571
4	1, 187500000	2, 820000000	1, 576190476	0, 395833333	0, 940000000	0, 525396825
5	1, 232649071	3, 803030303	1, 759906760	0, 308162268	0, 950757576	0, 439976690
6	1, 280785891	4, 788212435	1, 957042957	0, 256157178	0, 957642487	0, 391408591
7	1, 322698224	5, 770438549	2, 137870128	0, 220449704	0, 961739758	0, 356311688
8	1, 363989613	6, 751572493	2, 320248929	0, 194855659	0, 964510356	0, 331464133
9	1, 402468979	7, 733105601	2, 496464714	0, 175308622	0, 966638200	0, 312058089
10	1, 439464334	8, 714770487	2, 670148311	0, 159940482	0, 968307832	0, 296683146
11	1, 474743645	9, 697001722	2, 839981439	0, 147474365	0, 969700172	0, 283998144

8.2. Havel–Hakimi-tétel és Chungphaisan tétele

Chungphaisan [18] a következő módon terjesztette ki a Havel-Hakimi tételt.

8.4. Tétel. (Chungphaisan [18]) *Legyen $n \geq 2$ és $b \geq 1$. Az $s = (s_1, \ldots, s_n)$ $(0, b, n)$-szabályos sorozat akkor és csak akkor $(0, b, n)$-grafikus, ha a j-edik b-redukált $w_j^* = (w_1^*, \ldots, w_{n-1}^*)$ sorozat $(0, b, n)$-grafikus minden $1 \geq j \geq n$ indexre.*

Bizonyítás. Lásd [18]. □

A tételen alapuló algoritmus nagyon lassú. A tétel következő javítása azonban lehetővé teszi, hogy a tesztelést legrosszabb esetben is el tudjuk végezni $O(n)$ idő alatt.

8.5. Tétel. (Iványi, [34]) *Legyen $n \geq 1$ és $b \geq 1$. Nemnegatív egészek egy $s = (s_1, \ldots, s_n)$ $(0, b(n-1), n)$-szabályos sorozata akkor és csak akkor $(0, b, n)$-grafikus, ha*

$$\sum_{i=1}^{n} s_i \quad \textit{páros}$$

és

$$\sum_{i=1}^{j} s_i \leq bj(j-1) \leq \sum_{k=j+1}^{n} \min(jb, s_k) \quad (j = 1, \ldots, n-1).$$

Bizonyítás. Lásd [34]. □

A következő Chungphaisan–Havel–Hakimi-lineáris algoritmus (ChHHl) – amely a HH algoritmus természetes általánosítása – $O(n)$ idő alatt eldönti, hogy egy $(0, b, n)$-szabályos gráf $(0, b, n)$-grafikus-e.

8.2. Algoritmus. Chungphaisan-Havel-Hakimi-lineáris(n, s, b, L)

Bemenet. n: csúcsok száma $(n \geq 1)$;
$s = (s_1, \ldots, s_n)$: $(0, b, n)$-grafikus sorozat;
b: a gráf két csúcsa között megengedett élek maximális száma $(1 \leq b \leq 2)$.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozók. i: ciklus változó;
$w = (w_1, \ldots, w_n)$: w_i az i indexhez tartozó súlypont;
$r = (r_1, \ldots, r_n)$: r_i az i indexhez tartozó maradék.

```
1.  L = 0                                   // 1. sor: a gyakoribb érték beállítása
2.  if s_1 == 0                             // 2–4. sor: a nullákból álló sorozat grafikus
3.      L = 1
4.      return L
5.  if s_{⌈s_1/b+1⌉} == 0                   // 5–7. sor: s_1 ellenőrzése konstans idő alatt
6.      return L
7.  H_1 = s_1                               // 7. sor: H_1 kezdeti értékének beállítása
8.  for i = 2 to n − 1                      // 8–9. sor: H további elemeinek számítása
9.      H_i = H_{i−1} + s_i
10. if H_n páratlan                         // 10–11. sor: paritás tesztelése
11.     return L
12. w_1 = n                 // 12. sor: első súlypont kezdeti értékének beállítása
13. while s_{w_1} < b ∧ w_1 > 0
14.     w_1 = w_1 − 1
```

```
15. if s_1 > b(w_1 − 1) + H_n − H_{w_1}
16.     return L
17. r_1 = b(w_1 − 1) + H_n − H_{w_1} − s_1          // 17. sor: első maradék számítása
18. for i = 2 to n − 1                               // 18–34. sor: s tesztelése
19.     if H_{i−1} ≥ H_n/2 ∨ s_i ≤ 1 ∨ s_{i+1} = 0   // 19–21. sor: s elfogadása
20.         L = 1
21.         return L
22.     w_i = w_{i−1}                                // 22–24. sor: w_i frissítése
23.     while s_i < bi ∧ w_i > 0
24.         w_i = w_i − 1
25.     if w_i ≥ i                                   // 25–27. sor: esetszétválasztás
26.         if s_i > b(w_i − 1) + r_{i−1} + H_{w_{i−1}} − H_{w_i} −
                − b(w_{i−1} − w_i)(i − 1)            // 26. sor: s_i tesztelése
27.             return L
28.         r_i = b(w_i − 1) + r_{i−1} + H_{w_{i−1}} − H_{w_i} −
              − b(w_{i−1} − w_i)(i − 1) − s_i        // 28. sor: maradék frissítése
29.     else if s_i > bw_i + r_{i−1} + H_{w_{i−1}} − H_{w_i}
30.                − b(w_{i−1} − w_i)(i − 1)
31.         return L
32.         r_i = bw_i + r_{i−1} + H_{w_{i−1}} − H_{w_i}
              − b(w_{i−1} − w_i)(i − 1) − s_i        //32. sor: maradék frissítése
33. L = 1                                            // 33–34. sor: s elfogadása
34. return L
```

A következő állítás jellemzi ChHHl futási idejét.

8.6. Tétel. (Iványi, [34]) ChHHl futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n)$ között változik.

Bizonyítás. A 1–6. sorok végrehajtása $\Theta(1)$ időt igényel. Mivel ezek a sorok a nemgrafikus sorozatok jelentős részét kiszűrik, a legjobb futási idő $\Theta(1)$. A 7–11. sorok végrehajtása $\Theta(n)$ ideig tart. Mivel w szigorúan monoton csökken a program végrehajtása során, ezért a 12–24. sorok $O(n)$ időt igényelnek, így az algoritmus futási ideje minden esetben $\Theta(n)$. □

Legyen $b = 3$ és $s = (13, 10, 5, 5, 4, 1)$. Az ötödik és tizedik sorok feltételei nem teljesülnek és $r_1 = 0$. Ha $i = 2$, akkor $w_i = 5$, és teljesül a 20. sor feltétele, így s *nem* $(0, 1, 6)$*-grafikus.*

A következő példában b maradjon 3, viszont s-et változtassuk meg: legyen $s' = (13, 10, 5, 5, 4, 3)$. Az előző esethez képest annyi a változás, hogy $r_1 = 2$ az első maradék, majd $i = 2$ esetén $w_i = 2$, nem teljesül a 20. sor feltétele és $r_2 = 0$. $i = 3$ esetén teljesül a 19. sor $H_{i-1} \geq H_n/2$ feltétele, ezért s' $(0, 1, 6)$*-grafikus.*

A következő példában legyen $b = 1$ és $s = (4, 3^3, 1)$. Az 5. és 10. sorok feltételei nem teljesülnek és $r_1 = 0$. Ha $i = 2$, akkor $w_i = 4$, és nem teljesül a 20. sor

18. táblázat. ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt nem $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	0									
2	3	0								
3	22	3	0							
4	132	26	2	0						
5	824	164	31	4	0					
6	5 084	1 026	276	75	3	0				
7	31 902	6 288	2 018	829	111	5 0				
8	201 366	39 090	13 282	7 231	1 837	203	4	0		
9	1 281 918	244 833	84 340	53 594	20 681	4 259	298	6	0	
10	8 207 232	1 548 774	529 578	365 461	183 262	59 726	8 709	470	5	0
11	52 819 163	9 866 545	3 331 910	2 385 963	1 404 590	632 058	155 070	17 213	660	7

feltétele, az $i = 3$ esetben pedig a 19. sorban teljesül a $H_{i-1} \geq H_n/2$ feltétel, azaz s $(0, 1, 5)$-*grafikus*.

A 18. táblázat a ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt nem $(0, 2, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A 19. táblázat a ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 2, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

19. táblázat. ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	1									
2	2	0								
3	1	9	0							
4	1	7	42	0						
5	1	10	29	224	0					
6	1	14	49	183	1 297	0				
7	1	18	70	345	1 143	7 658	0			
8	1	23	97	559	2 326	7 262	46 489	0		
9	1	28	125	846	4 038	15 927	46 074	286 007	0	
10	1	34	159	1 191	6 520	29 629	107 724	295 609	1 779 026	0
11	1	40	193	1 624	9 668	50 663	213 399	728 610	1 900 061	11 154 877

A 20. táblázat a ChHHl algoritmus hatékonyságát jellemzi $(0, 2, n)$-szabályos sorozatok és $n = 1, \ldots, 11$ csúcs esetén.

20. táblázat. ChHHl hatékonysági jellemzői $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

$\frac{n}{\text{jellemző}}$	X	Y	Z	X'	Y'	Z'
2	1, 000000000	1, 000000000	1, 000000000	1, 000000000	1, 000000000	1, 000000000
3	1, 120000000	1, 900000000	1, 342857143	0, 560000000	0, 950000000	0, 671428571
4	1, 187500000	2, 820000000	1, 576190476	0, 395833333	0, 940000000	0, 525396825
5	1, 232649071	3, 803030303	1, 759906760	0, 308162268	0, 950757576	0, 439976690
6	1, 280785891	4, 788212435	1, 957042957	0, 256157178	0, 957642487	0, 391408591
7	1, 322698224	5, 770438549	2, 137870128	0, 220449704	0, 961739758	0, 356311688
8	1, 363989613	6, 751572493	2, 320248929	0, 194855659	0, 964510356	0, 331464133
9	1, 402468979	7, 733105601	2, 496464714	0, 175308622	0, 966638200	0, 312058089
10	1, 439464334	8, 714770487	2, 670148311	0, 159940482	0, 968307832	0, 296683146
11	1, 474743645	9, 697001722	2, 839981439	0, 147474365	0, 969700172	0, 283998144

9. (a, b, n)-gráfok

Chungphaisan tételének közvetlen következménye az alábbi állítás.

9.1. Következmény. *Legyen $n \geq 2$. Az $s = (s_1, \ldots, s_n)$ (a, b, n)-szabályos sorozat akkor és csak akkor (a, b, n)-grafikus, ha az $s' = (s_1 - a(n-1), \ldots, s_n - a(n-1))$ sorozat $(0, b-a, n)$-grafikus.*

Bizonyítás. Egy (a, b, n)-gráfban minden csúcspár elemei legalább a éllel össze vannak kötve. Ezért ha minden csúcspár esetén eltávolítunk a élet, egy $(0, b-a, n)$-*gráfot* kapunk. □

A 9.1. következmény szerint a következő három táblázat adatai megegyeznek a $(0, 3, n)$-szabályos sorozatokra vonatkozó hasonló adatokkal.

A 21. és 22. táblázatok a ChEGl i-edik – ahol $(i = 1, \ldots, 4)$, illetve $(i = 5, \ldots, 10)$ – menetében kiszűrt nem $(2, 5, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A 23. táblázat a CL i-edik $(i = 1, \ldots, 10)$ menetében kiszűrt $(2, 5, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A következő 24. táblázat a ChEGl algoritmus hatékonyságát jellemzi $a = 2$, $b = 5$ és $n = 1, \ldots, 11$ csúcs esetén.

10. $(0, 1, n)$-grafikus sorozatok párhuzamos leszámlálása

A 8. táblázat 1-től 29 csúcsig tartalmazza a grafikus sorozatok számát. A táblázat úgy készült, hogy párhuzamosítottuk az Erdős–Gallai-gyorsan algoritmust. Az eredmény az Erdős–Gallai-leszámláló (EGe) algoritmus, amely minden szóba jövő sorozatot tesztel.

21. táblázat. ChEGl i-edik ($i = 1, \ldots, 4$) menetében kiszűrt, nem $(2, 5, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4
1	0	0	0	0
2	6	0	0	0
3	57	7	0	0
4	475	83	7	0
5	4099	732	163	13
6	35500	6287	2068	441
7	312188	53601	20775	7766
8	2769457	463794	188643	97976
9	24768128	4061297	1658351	1021804
10	222858957	35952854	14508359	9681500
11	2015400842	320927140	127636563	87804078

22. táblázat. ChEGl i-edik ($i = 5, \ldots, 10$) menetében kiszűrt, nem $(2, 5, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	5	6	7	8	9	10
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	14	0	0	0	0	0
7	921	21	0	0	0	0
8	24374	1921	23	0	0	0
9	405996	71152	3572	31	0	0
10	5136605	1554803	186666	6402	34	0
11	55159143	24279000	5343051	452411	10751	43

Mivel viszonylag sok processzor vett részt a számolásban, viszont bizonytalan volt, hogy az egyes processzorok meddig vehetnek részt a számolásban, a feladatot *szeleteknek* nevezett kisebb részekre bontottuk. Célszerű volt, hogy a szeletek feldolgozása hasonló ideig tartson.

23. táblázat. ChEGl i-edik ($i = 1, \ldots, 10$) menetében kiszűrt $(2, 5, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0
3	1	19	0	0	0	0	0	0	0	0
4	1	8	141	0	0	0	0	0	0	0
5	1	11	40	1129	0	0	0	0	0	0
6	1	15	60	317	9561	0	0	0	0	0
7	1	19	81	497	2395	82435	0	0	0	0
8	1	24	108	720	3838	19074	722192	0	0	0
9	1	29	136	1016	5733	30725	153657	6385472	0	0
10	1	35	170	1366	8387	47136	247112	1259718	56880031	0
11	1	41	204	1804	11644	70961	385774	2010389	10453559	509514569

24. táblázat. ChEGl hatékonysági jellemzői $a = 2$, $b = 5$ és $n = 1, \ldots, 11$ csúcs esetén.

$\frac{n}{\text{jellemző}}$	X	Y	Z	X'	Y'	Z'
2	1, 000000000	1, 000000000	1, 000000000	1, 000000000	1, 000000000	1, 000000000
3	1, 109375000	1, 950000000	1, 309523810	0, 554687500	0, 975000000	0, 654761905
4	1, 171681416	2, 933333333	1, 541258741	0, 390560472	0, 977777778	0, 513752914
5	1, 219093269	3, 944961897	1, 739334195	0, 304773317	0, 986240474	0, 434833549
6	1, 266350711	4, 951175407	1, 942282176	0, 253270142	0, 990235081	0, 388456435
7	1, 309250339	5, 956536499	2, 135146661	0, 218208390	0, 992756083	0, 355857777
8	1, 350304891	6, 960496382	2, 325332905	0, 192900699	0, 994356626	0, 332190415
9	1, 389017669	7, 963928944	2, 510223895	0, 173627209	0, 995491118	0, 313777987
10	1, 426027860	8, 966857120	2, 691252565	0, 158447540	0, 996317458	0, 299028063
11	1, 461490194	9, 969401198	2, 868359205	0, 146149019	0, 996940120	0, 286835921

Az Erdős–Gallai-lineáris algoritmus egyik lehetséges alkalmazása, hogy meghatározzuk a grafikus sorozatok számát olyan n értékekre, amelyekre eddig a nagy számolásigény miatt nem volt ismert: Sloane *The On-Line Encyclopedia of Integer Sequences* című honlapja [77] az $n = 23$ értékig tartalmazta a grafikus sorozatok számát. Ezt kiegészítettük $n = 29$ csúcsig [79].

Az Erdős–Gallai-leszámláló (EGe) algoritmus a lineáris legrosszabb eset mellett azt is igyekszik kihasználni, hogy ha lexikografikus sorrendben ellenőrizzük a szóba jövő sorozatokat, akkor a szomszédos sorozatok bizonyos tulajdonságai nagyon ha-

sonlóak, ezért adott sorozat jellemzői az őt megelőző sorozat jellemző adataiból konstans várható idő alatt meghatározhatóak.

Igyekeztünk az ellenőrizendő sorozatok számát is csökkenteni.

Ennek egy egyszerű megoldása, hogy eleve csak a páros sorozatokat állítjuk elő. További ötlet, hogy csak a nullamentes sorozatokat vizsgáljuk. A nullát tartalmazó $(0, 1, n)$-grafikus sorozatok között ugyanis a 4.7. lemma szerint pontosan $G(n-1)$ nullamentes grafikus sorozat van. A 4.2. lemma szerint aszimptotikusan a szabályos sorozatok fele tartalmaz legalább egy nullát. Szimulációs vizsgálataink szerint ez a páros sorozatokra is igaz.

Lényeges gyorsítást jelent az is, hogy a sorozatokat csak az ugró pontokban vizsgáljuk.

Az EGe program azt is kihasználja, hogy a szomszédos sorozatok ellenőrző pontjainak a listája átlagosan konstans idő alatt származtatható a megelőző sorozat adataiból. A kiindulási értékek szintén könnyen számíthatók: az első – $q = (n-1)^n$ – sorozatra a C lista üres (azaz egyáltalán nem kell ellenőrzést végeznünk), a súlypontok listája pedig kezdetben $w = (n-1)^{n-1}$.

Az Erdős–Gallai-leszámláló algoritmus előállítja és megvizsgálja az n-páros, nullamentes sorozatokat, és kimenetként megadja a $G_z(n)$ értéket. Az algoritmus kihasználja, hogy a páros sorozatok lexikografikusan csökkenő sorozatában szomszédos sorozatok több lényeges paramétere hasonló, ezért ezek a paraméterek a vizsgált s' sorozatot megelőző s sorozat adott paraméteréből gyorsan meghatározhatóak.

Az ugrópontok $C(s')$ listája rendszerint megegyezik a $C(s)$ listával, és legfeljebb a végén változik egy vagy két elem.

Mivel a futási idő csökkentése érdekében az Erdős–Gallai-leszámláló algoritmus csak nullamentes sorozatokat állít elő és tesztel, a szeletekre bontás alapja a (20) képlet.

Feltételeztük, hogy a $(0, n-1, n)$-szabályos nullamentes sorozatok halmazának szeletekre való felbontásánál az egyes szeletek futási ideje arányos a hozzájuk tartozó $R(1, n-1, n)$-szabályos sorozatok számával.

Most tekintsünk egy példát: az $n = 29$-re írt programban az $n = 28$ esetben szerzett tapasztalatok alapján feltettük, hogy a tiszta futási idő összesen körülbelül 6000 nap lesz. Feltételezve, hogy a gépek egy részét csak éjszakára kapjuk meg, egy szelet maximális futási idejét 12 órára állítottuk. Ez pontosan 12 órás szeletek mellett 12000 szeletet jelentett volna. A tényleges adatokat a 25. táblázat tartalmazza.

11. Köszönetnyilvánítás.

A szerzők köszönik Burcsi Péter és Király Zoltán (Eötvös Loránd Tudományegyetem), Kása Zoltán (Sapientia Magyar Tudományegyetem), valamint az ismeretlen lektor jobbító észrevételeit. A kutatás az Európai Unió támogatásával, az Euró-

25. táblázat. Teljes futási idő és szeletek száma $n = 25, \ldots, 29$ csúcs esetén.

n	Futási idő (nap)	Szeletek száma
25	26	435
26	70	435
27	316	435
28	1130	2 001
29	6733	15 119

pai Szociális Alap társfinanszírozásával valósul meg (a támogatás száma TÁMOP 4.2.1/B-09/1/KMR-2010-0003).

Hivatkozások

[1] Ascher, M.: *Mu torere: an analysis of a Maori game.* Math. Mag. **60(2)**, (1987) 90–100.

[2] Avis, D., Fukuda, K.: *Reverse search for enumeration.* Discrete Appl. Math. **2**, (1993) 21–46.

[3] Barnes, T. M., Savage, C. D.: *A recurrence for counting graphical partitions.* Electron. J. Combin. **2**, (1995) R11, 10 pp.

[4] Barnes, T. M., Savage, C. D.: *Efficient generation of graphical partitions.* Discrete Appl. Math. **78(1–3)**, (1997) 17–26.

[5] Barrus, M. D.: *Havel-Hakimi residues of unigraphs*, Inf. Proc. Letters **112**, (2012) 44–48.

[6] Beasley, L. B., Brown D. E., Reid, K. B.: *Extending partial tournaments.* Math. Comput. Modelling **50(1)**, (2009) 287–291.

[7] Bereg S., Ito, H.: *Transforming graphs with the same degree sequence.* In: (ed. H. Ito et al.) The Kyoto Int. Conf. on Computational Geometry and Graph Theory, LNCS **4535.** Springer-Verlag, Berlin, Heidelberg. (2008) 25–32.

[8] Berger, A., Müller-Hannemann, M.: *Uniform sampling of digraphs with a fixed degree sequence.* In: (ed. D. M. Thilikos) WG2010, LNCS **6410**, (2010), 220–231.

[9] Berger, A.: *A note on the characterization of digraph sequences*, arXiv, arXiv:1112.1215v1 [math.CO] (6 December 2011).

[10] Berger, A., Müller-Hannemann, M.: *How to attack the NP-complete dag realization problems in practice*, arXiv, arXiv:1203.36v1, (2012).

[11] Bozóki, S., Fülöp, J., Poesz, A.: *On pairwise comparison matrices that can be made consistent by the modification of a few elements.* CEJOR Cent. Eur. J. Oper. Res. **19**, (2011) 157–175.

[12] Bozóki S., Fülöp J., Rónyai, L.: *On optimal completion of incomplete pairwise comparison matrices.* Math. Comput. Modelling **52**, (2010) 318–333.

[13] Brualdi, A. R., Kiernan K.: *Landau's and Rado's theorems and partial tournaments,* Electron. J. Combin. **16(#N2)**, (2009) (6 pp).

[14] Burns, J. M.: *The number of degree sequences.* PhD Dissertation, MIT, (2007).

[15] Busch A. N., Chen G., Jacobson M. S.: *Transitive partitions in realizations of tournament score sequences.* J. Graph Theory **64(1)**, (2010), 52–62.

[16] Cormen, T. H., Leiserson, Ch. E., Rivest, R. L., Stein, C.: *Introduction to Algorithms.* Third edition, The MIT Press/McGraw Hill, Cambridge/New York, 2009. Magyarul: *Algoritmusok.* Műszaki Könyvkiadó, Budapest, (2003).

[17] Coudum, S. A.: *A simple proof of the Erdős-Gallai theorem on graph sequences.* Bull. Austral. Math. Soc. **33**, (1986) 67–70.

[18] Chungphaisan, V.: *Conditions for sequences to be r-graphical.* Discrete Math. **7**, (1974) 31–39.

[19] Del Genio, C. I., Kim, H., Toroczkai, Z., Bassler, K. E.: *Efficient and exact sampling of simple graphs with given arbitrary degree sequence.* PLoS ONE **5(4)**, e10012 (2010).

[20] Erdős, P., Gallai, T.: *Gráfok előírt fokú pontokkal.* Mat. Lapok **11**, (1960) 264–274.

[21] Erdős, P., Király, Z., Miklós, I.: *On the swap-distances of different realizations of a graphical degree sequence*, arXiv, arXiv:1205.2842v1 [math.CO] (13 May 2012).

[22] Erdős, P. L., Miklós, I., Toroczkai, Z.: *A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs.* Electron. J. Combin. **17(1)**, (2010) R66, 10 pp.

[23] Erdős, P., Richmond L. B.: *On graphical partitions.* Combinatorica **13(1)**, (1993) 57–63.

[24] Frank, A.: *Connections in Combinatorial Optimization.* Oxford University Press, Oxford, (2011).

[25] Frank, D. A., Savage, C. D., Sellers, J. A.: *On the number of graphical forest partitions. Ars Combin.* **65**, (2002) 33–37.

[26] Garg, A., Goel, A., Tripathi, A., *Constructive extensions of two results on graph sequences.* Discrete Appl. Math. **159(17)**, (2011) 2170–2174.

[27] Hakimi, S. L.: *On the realizability of a set of integers as degrees of the vertices of a simple graph.* J. SIAM Appl. Math. **10**, (1962) 496–506.

[28] Havel, V.: A remark on the existence of finite graphs (cseh). *Časopis Pěst. Mat.* **80**, (1955), 477–480.

[29] Hell, P., Kirkpatrick, D.: *Linear-time certifying algorithms for near-graphical sequences.* Discrete Math. **309(18)**, (2009) 5703–5713.

[30] Iványi, A.: *Football sorozatok tesztelése.* In: XXV. Magyar Operációkutatási Konferencia Kivonatai (Debrecen, 2001. október 17–20.), 52–52.

[31] Iványi, A.: *Reconstruction of complete interval tournaments.* Acta Univ. Sapientiae, Inform., **1(1)**, (2009) 71–88.

[32] Iványi, A.: *Reconstruction of complete interval tournaments. II.* Acta Univ. Sapientiae, Math., **2(1)**, (2010) 47–71.

[33] Iványi, A.: *Deciding the validity of the score sequence of a soccer tournament.* In (ed. A. Frank): Open problems of the Egerváry Research Group, Budapest, (2012). http://lemon.cs.elte.hu/egres/open/.

[34] Iványi, A.: *Degree sequences of multigraphs.* Annales Univ. Budapest., Comput. **37**, (2012) 195–214.

[35] Iványi, A., Lucz, L., Móri F. T., Sótér, P.: *On the Erdős-Gallai and Havel-Hakimi algorithms.* Acta Univ. Sapientiae, Inform. **3(2)**, (2011) 230–268.

[36] Iványi, A., Lucz, L., Móri F. T., Sótér, P.: *Number of graphical partitions (degree-vectors for simple graphs with n vertices.* Elérhető: http://oeis.org/A004251.

[37] Iványi, A., Pirzada, S.: *Comparison based ranking.* In (ed. A. Iványi): Algorithms of Informatics, Vol. **3**. AnTonCom, Budapest (2011) 1262–1311.

[38] Iványi, A., Schoenfield, J. E.: *Deciding football sequences.* Acta Univ. Sapientiae, Inform., **4(1)**, (2012) 130–183.

[39] Kéri G.: *On qualitatively consistent, transitive and contradictory judgment matrices emerging from multiattribute decision procedures.* Central Eur. J. Oper. Res. **19(2)**, (2011) 215–224.

[40] Kim, H., Toroczkai, Z., Miklós, I., Erdős, P. L., Székely, L. A.: *Degree-based graph construction.* J. Physics: Math. Theor. A **42(39)**, (2009) 392–401.

[41] Kleitman, D. J., Wang, D. L.: *Algorithms for constructing graphs and digraphs with given valencies and factors.* Discrete Math. **6**, (1973) 79–88.

[42] Kleitman, D. J., Winston K. J.: *Forests and score vectors.* Combinatorica **1(1)**, (1981) 49–54.

[43] Knuth, D. E.: *The Art of Computer Programming. Volume 4A, Combinatorial Algorithms.* Addison–Wesley, Upper Saddle River, (2011).

[44] Kohnert, A.: *Dominance order and graphical partitions.* Elec. J. Comb. **11(1)**, (2004) 17 pp.

[45] Kovács, G. Zs., Pataki, N.: *Rangsorolási algoritmusok elemzése.* TDK dolgozat. ELTE TTK, Budapest, (2002) 39 oldal.

[46] LaMar, M. D.: *Algorithms for realizing degree sequences of directed graphs.* arXiv-0906:0343ve [math.CO], (7 June 2010).

[47] Landau, H. G.: *On dominance relations and the structure of animal societies. III. The condition for a score sequence.* Bull. Math. Biophys. **15**, (1953) 143–148.

[48] Liljeros, F., Edling, C. R., Amaral, L., Stanley, H., Áberg, Y.: *The web of human sexual contacts.* Nature **411**, (2001) 907–908.

[49] Lovász, L.: *Combinatorial Problems and Exercises* (corrected version of the second edition). AMS Chelsea Publishing, Boston, 2007. Magyarul: *Kombinatorikai problémák és feladatok.* Typotex, Budapest, (1999).

[50] Lucz, L.: *Párhuzamos Erdős-Gallai algoritmus.* TDK dolgozat, ELTE IK, Budapest (2011). Elérhető: http://people.inf.elte.hu/lulsaai/Holzhacker/TDK/.

[51] Lucz, L.: *Football league numbers: the possible point series for a league of n teams playing each other twice.* OEIS, A064422 számú sorozat. Elérhető: http://oeis.org/A064422.

[52] Lucz, L.: *Football league numbers with distinct point totals.* OEIS A209467 számú sorozat, Elérhető: http://oeis.org/A209467.

[53] Lucz, L.: *Gráfok foksorozatainak elemzése*, Programtervező informatikus diplomamunka, ELTE IK, Budapest, (2012). Elérhető: http://people.inf.elte.hu/lulsaai/diploma.

[54] Lucz, L., Sótér, P.: *Foksorozatokat ellenőrző algoritmusok.* TDK dolgozat. ELTE IK, Budapest, (2011). Elérhető: http://people.inf.elte.hu/lulsaai/Holzhacker/TDK/

[55] Meierling, D., Volkmann, L.: *A remark on degree sequences of multigraphs.* Math. Methods Oper. Res. **69(2)**, (2009) 369–374.

[56] Metropolis, N., Stein, P. R.: *The enumeration of graphical partitions.* European J. Comb. **1(2)**, (1980) 139–153.

[57] Miklós, I., Erdős, P. L., Soukup, L.: A remark on degree sequences of multigraphs. (2011) (benyújtva).

[58] Miller, J. W.: *Reduced criterion for degree sequences*, arXiv, arXiv:1205.2686v1 [math.CO] (11 May 2012), 18 pages.

[59] Moon, J. W.: *Topics on Tournaments.* Holt, Rinehart, and Winston, New York, (1968).

[60] Narayana, T. V., Bent, D. H.: Computation of the number of score sequences in round-robin tournaments. Canad. Math. Bull. **7(1)**, (1964) 133–136.

[61] Newman, M. E. J., Barabási, A. L.: *The Structure and Dynamics of Networks.* Princeton University Press, Princeton, NJ, (2006).

[62] Özkan, S.: *Generalization of the Erdős-Gallai inequality.* Ars Combin. **98**, (2011) 295–302.

[63] Pécsy G., Szűcs, L.: *Parallel verification and enumeration of tournaments.* Stud. Univ. Babeş-Bolyai, Inform. **45(2)**, (2000) 11–26.

[64] Pirzada, S.: *Graph Theory.* Orient Blackswan, Hydarabad (2012), to appear.

[65] Pirzada S., Iványi A.: *Minimal digraphs with given imbalance sequences.* Acta Univ. Sapientiae **4(1)**, (2012) 61–76.

[66] Pirzada, S., Iványi, A., Shah, N.: *Imbalances of bipartite multitournaments.* Annales Univ. Budapest., Comp. **37** (2012) 215–228.

[67] Pirzada, S., Iványi, A., Khan, M. A.: *Score sets and kings.* In (ed. A. Iványi): Algorithms of Informatics, Vol. **3**, ed. A. Iványi. AnTonCom, Budapest (2011) 1451–1490.

[68] Pirzada, S., Naikoo, T. A., Samee, U. T., Iványi, A.: *Imbalances in directed multigraphs.* Acta Univ. Sapientiae, Inform. **2(1)**, (2010) 47–71.

[69] Pirzada, S., Zhou G., Iványi A.: *On k-hypertournament losing scores,* Acta Univ. Sapientiae, Inform. **2(2)**, (2010) 184–193.

[70] Rødseth, Ø. J., Sellers, J. A., Tverberg, H.: *Enumeration of the degree sequences of non-separable graphs and connected graphs.* European J. Comb. **30(5)**, 1309–1319.

[71] Ruskey, F., Cohen, R., Eades, P., Scott, A.: *Alley CAT's in search of good homes.* Congr. Num., **102**, (1994) 97–110.

[72] Schoenfield, J. E.: *The number of football score sequences*, in: ed. by N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences,* (2012). http://oeis.org/A064626

[73] Sierksma, G., Hoogeveen, H.: *Seven criteria for integer sequences being graphic.* J. Graph Theory **15(2)**, (1991) 223–231.

[74] Siklósi, B.: *Soros és párhuzamos algoritmusok összehasonlítása sportversenyekkel kapcsolatos problémákban.* Programtervező matematikus diplomamunka. ELTE TTK, Budapest, (2001), 69 oldal.

[75] Simion, R.: *Convex polytopes and enumeration.* Advances in Applied Math. **18(2)**, (1996) 149–180.

[76] Sloane N. J. A., Plouffe S.: *The Encyclopedia of Integer Sequences.* Academic Press, (1995).

[77] Sloane N. J. A. (szerk.): *Encyclopedia of Integer Sequences.* (2012) http://oeis.org

[78] Sloane N. J. A.: *The number of ways to put $n+1$ indistinguishable balls into $n+1$ distinguishable boxes.* In (ed. N. J. A. Sloane): The On-line Encyclopedia of the Integer Sequences. (2012) http://oeis.org/A0017000

[79] Sloane N. J. A.: *The number of degree-vectors for simple graphs.* In (ed. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences. (2012) http://oeis.org/A004251

[80] Sloane N. J. A.: *The number of bracelets with n red, 1 pink and $n-1$ blue beads.* In (ed. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences. (2012) http://oeis.org/A0005654

[81] Soroker, D.: *Optimal parallel construction of prescribed tournaments.* Discrete Appl. Math. **29(1)**, (1990) 113–125.

[82] Stanley, R.: *Enumerative Combinatorics. Vol. 2.* Cambridge University Press, Cambridge, (1997).

[83] Stanley, R.: *A zonotope associated with graphical degree sequence.* In: Applied Geometry and Discrete Mathematics, Festschr. 65th Birthday Victor Klee. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. **4**, (1991) 555-570.

[84] Takahashi, M.: *Optimization Methods for Graphical Degree Sequence Problems and their Extensions*, PhD thesis, Graduate School of Information, Production and Systems, Waseda University, Tokyo, (2007). http://hdl.handle.net/2065/28387

[85] Tripathi, A., Tyagy, H.: *A simple criterion on degree sequences of graphs.* Discrete Appl. Math. **156(18)**, (2008) 3513–3517.

[86] Tripathi, A., Vijay, S.: *A note on a theorem of Erdős & Gallai.* Discrete Math. **265(1–3)**, (2003) 417–420.

[87] Tripathi, A., Venugopalan, S., West, D. B.: *A short constructive proof of the Erdős-Gallai characterization of graphic lists.* Discrete Math. **310(4)**, (2010) 833–834.

[88] Weisstein, E. W.: *Degree sequence.* From MathWorld—Wolfram Web Resource, (2011).

[89] Weisstein, E. W.: *Graphic sequence.* From MathWorld—Wolfram Web Resource, (2011).

[90] Winston, K. J., Kleitman, D. J.: *On the asymptotic number of tournament score sequences.* J. Combin. Theory Ser. A. **35**, (1983) 208–230.

(Beérkezett: 2011. július 17., módosítva 2012. november 19.)

IVÁNYI ANTAL
Eötvös Loránd Tudományegyetem
Informatikai Kar
1117 Budapest, Pázmány Péter sétány 1/C
e-mail: tony@inf.elte.hu

LUCZ LORÁND
Eötvös Loránd Tudományegyetem
Informatikai Kar
1117 Budapest, Pázmány Péter sétány 1/C
e-mail: lorand.lucz@gmail.com

DEGREE SEQUENCES OF MULTIGRAPHS

Antal Iványi, Loránd Lucz

Let a, b and n integers, $0 \leq a \leq b$ and $n \geq 1$. (a, b, n)-graphs are loopless multigraphs in which any two vertices are connected with an least a and at most b edges and contain n vertices. Havel in 1955 [28], Erdős and Gallai in 1960 [20], Hakimi in 1962 [27], Tripathi, Venugopalan and West in 2010 [87] proposed a method to decide, whether a sequence of nonnegative integers can be the degree sequence of a $(0, 1, n)$-graph. These methods are at least quadratic in worst case. Takahashi [84] in 2007 while Hell and Kirkpatrick [29] in 2009 proposed linear algorithm. Chungphaisan in 1974 [18] extended Havel-Hakimi and Erdős-Gallai theorem for $(0, b, n)$-graphs. We extend Erdős-Gallai-Chungphaisan theorem for (a, b, n)-graphs and propose a linear time algorithm, based on our theorem. We also propose a linear time version of the testing Havel-Hakimi algorithm and extend it for $(0, 2, n)$-graphs.

Alkalmazott Matematikai Lapok **29** (2012), 53-66.

KÖZÖSSÉGEK ÉS SZEREPÜK A KISVILÁG GRÁFOKBAN

BARTALOS ISTVÁN ÉS PLUHÁR ANDRÁS

Összefoglaló írásunkban kísérletet teszünk a gráfokra kifejlesztett közösségkereső algoritmusok áttekintésére, egységesítésére és kiértékelésére. Bemutatjuk az eredményként előálló közösségi információ felhasználását a gráfos adatbányászatban és a gráfok segítségével végrehajtható modellezésben, melyeknek sikeres gyakorlati alkalmazásai vannak.

1. Bevezetés

A *kisvilág gráfok* felfedezése jelentősen megváltoztatta, kibővítette a gráfelméleti kutatások irányát, lásd Barabási és Albert [2, 3]. Nemcsak ezek a gráfok különböznek a korábban vizsgált gráfoktól, hanem a velük kapcsolatban megfogalmazott kérdések és problémák is.

Nem könnyű feladat egy kisvilág gráf felépítéséhez szükséges információk összegyűjtése, vagy éppen annak eldöntése, *hogyan* készítsünk a rendelkezésre álló adatokból gráfot, lásd Csernenszky és társai, illetve Hidalgo és társai [13, 23]. Ugyanígy, bár számos próbálkozás történt, nincs minden igénynek eleget tevő modell véletlen kisvilág gráfok generálására sem, lásd Cami és Deo [11].

A valós alkalmazásokban fellépő méretek miatt időigényes algoritmusok nemigen használhatók, így jobbára meg kell elégedni egyszerűbb heurisztikákkal, melyek sokszor a fizikából kölcsönzött intuícióból erednek, lásd Barabási, Bollobás, Newman cikkei [3, 5, 28]. A szokásos jelölést követve egy G gráf ponthalmazát $V(G)$-vel, élhalmazát pedig $E(G)$-vel jelöljük. Ha az utóbbi *rendezett* párokat tartalmaz, akkor G *irányított*, és az élek *súlyozottak* is lehetnek.

A legtöbb további vizsgálat egyik alapvető feltétele a gráf pontjainak klasszifikációja, csoportokba rendezése. Ez történhet osztályozással, azaz $V(G)$-t felbontjuk $\{\mathcal{C}_i\}_{i=1}^m$ halmazok, ún. *klaszterek* diszjunkt uniójára. A másik megközelítésben nem kívánjuk meg sem a csoportjaink diszjunktságát, sem azt, hogy együtt kiadják $V(G)$-t. Ezeket az entitásokat szokás közösségeknek hívni; mi itt *közösség* alatt mindig ezeket értjük, míg az osztályozás elemeit klasztereknek hívjuk. Rengeteg erőfeszítés történt a klaszterek előállítására, vizsgálatára, illetve alkalmazására, részletesen lásd pl. Newman [28]. Annyit megjegyeznénk, hogy a klaszterek előállítására mind ún. *top down* (felülről lefelé) és *bottom up* (alulról felfele) építkező

algoritmusokat javasoltak. Ezzel szemben a közösségek keresésére szolgáló algoritmusok jobbára az alulról építkezést használják, azaz kisebb közösségek növelésével próbálnak megfelelő eredményhez jutni.

A klaszterezés (és így a közösségkeresés is) elméletileg megalapozhatatlan Kleinberg [25] eredménye szerint, ezért a sokszor követett pragmatikus megoldás marad: veszünk egy ésszerűnek tűnő algoritmust, az eredményét definiáljuk klasztereknek/közösségeknek, és megnézzük használhatóságát.

2. Néhány algoritmus

Három tipikus közösségkereső algoritmust tekintünk, melyek hasonló elven alapulnak. Az egyik első, ténylegesen használt algoritmus az N^{++}, Csizmadia és társai, ill. Pluhár [8, 15, 31, 32]. A k-klikk perkolációs algoritmus, a CPM, az első széles körben ismert módszer, melyet Palla és társai [29] szintén valós feladatokra alkalmaztak. Az élek klaszterezése a harmadik – főként elméleti érdekességű Pluhár, Evans és Lambiote [31, 17].

2.1. Az $\mathbf{N^{++}}$ algoritmus

[32, 15] Ez egy generikus algoritmus egy tetszőleges

$$f : 2^{V(G)} \times V(G) \to \mathbb{R}$$

és $c : \mathbb{N} \to \mathbb{R}$ függvénnyel, ahol $f(A, x)$ jelenti az A közösség és az x csúcs kapcsolatának erősségét. Csatoljuk x-et A-hoz, ha $f(A, x) \geq c(|A|)$.

A **Build** szubrutin lentről felfelé építkezve megadja a közösségek $\mathcal{K}$ halmazának első közelítését.

Algorithm 2.1 A Build pszeudó kódja

1. **begin**(Build)
2. **input** G, k, c //max k-elemű c-közösségeket keresünk
3. **let** $\mathcal{K} := V(G)$ //kezdetben a csúcsok a közösségek $L = 0$
4. **for** $i = 1$ to k
5. $\forall A \in \mathcal{K}$, $x \in V(G)$ ha $f(A, x) \geq c(|A|)$, akkor tegyük $A \cup \{x\}$-t $\mathcal{K}$-ba.
6. Töröljük az összes olyan $A \in \mathcal{K}$-t, amelyre $A \subset B \in \mathcal{K}$ és $A \neq B$.
7. **print** $\mathcal{K}$, „G legfeljebb k-elemű c-közösségei”.
8. **end**(Build)

A Build végrehajtása után a **Merge**-t használjuk a majdnem azonos közösségek összeolvasztására. Legyen C olyan gráf, amelyben $V(C) = \mathcal{K}$ és $(A, B) \in E(C)$, ha $A \cap B$ „elég nagy". Cseréljük ilyenkor $\mathcal{K}$-t $(\mathcal{K} \setminus \{A, B\}) \cup \{A \cup B\}$-ra. Ezután a C elemei legyenek a közösségek. A tapasztalat az alábbi értékeket javasolja. Jelentse a nagy a 60%-át a kisebb halmaz elemszámának. Az $f(A, x)$ értéke az x és A közötti egy és kettő hosszúságú utak számától függ. Tehát ahhoz, hogy megkapjuk az x-et tartalmazó közösségeket, elegendő keresni az $N^{++}(x) := N(N(x))$ halmazban, azaz legfeljebb a második szomszédok között.

Néhány hasonló módszert sorol Fortunato [18].

2.2. k-klikkek perkolációja

Röviden CPM módszer, [29]. Itt $k \in \mathbb{N}$ adott, mint az algoritmus paramétere. Miután megtaláltuk az összes k-klikket G-ben, tekintjük azt a Q_k gráfot, melynek csúcsai ezen klikkek és $(A, B) \in E(Q_k)$ pontosan akkor, ha $|A \cap B| = k - 1$. A közösségek Q_k összefüggő komponensei klikkjeinek egyesítései lesznek.

2.3. Élek klaszterezése

[31, 17] Klaszterezzük valamilyen módon az élek halmazát. Az egyes klaszterek éleinek végpontjai lesznek a közösségek.

Ezek a módszerek különböznek a talált közösségek típusaiban és a számítási költségeikben is. Jóllehet az élek klaszterezését könnyű végrehajtani, használata mégis jelentős hátrányokkal jár (pl. a kapott közösségek átfedése legfeljebb egy csúcspont mélységű).

Az N^{++} és a CPM a legígéretesebb algoritmusok; persze az implementációk minősége lényeges szempont. Kisvilág gráfokon mindkettő majdnem lineáris időben fut, ami természetes követelmény, ha valódi feladatokkal foglalkozunk.[1]

2.4. Egységes szemlélet

Vegyük észre, hogy a három felsorolt algoritmus család végrehajtása két lépésből áll. Először egy $\mathcal{F} = (V, \mathcal{H})$ hipergráfot határoznak meg, ahol $V = V(G)$ és $\mathcal{H} \subset 2^V$. A $\mathcal{H}$ elemei lesznek a közösségek *építőkövei*. A második lépésben $\mathcal{H}$-t alkalmas d távolságfüggvénnyel ellátva $\mathcal{M} = (\mathcal{H}, d)$ metrikus teret készítünk. Ezután valamilyen klaszterező algoritmussal $\mathcal{M}$ klasztereinek egy $\mathcal{C}$ halmazát kapjuk. Végül a keletkezett klasztereket V részhalmazaival azonosítjuk úgy, hogy egy $C_i \in \mathcal{C}$-re $K_i := \cup_{H \in C_i} H$, ahol K_i közösség megfelel C_i klaszternek.

A fenti algoritmusoknál $\mathcal{H}$ elemei (az építőkövek) rendre kis sűrűségű részgráfok, k-klikkek, illetve élhalmazok. A köztük levő kapcsolatot leíró $\mathcal{D}$ gráfban pontosan akkor van él, ha a kapcsolat szoros. Az első esetben $(K_i, K_j) \in \mathcal{D}$, ha

[1] Ez csúcsok millióit jelenti. Az N^{++} elérhető a Sixtep szoftverrel, míg a klikk-perkolációt a CFinderrel próbáltuk ki. Ezennel megköszönjük a programok készítőinek, hogy tudományos célokra elérhetővé tették a szoftverüket.

$|K_i \cap K_j|$ elég nagy, a másodikban, ha $|K_i \cap K_j| = k - 1$, míg a harmadik esetben ez paraméter.

2.5. Központiság alapú közösségkeresések

Az előző alfejezet paradigmájába bele nem illő megoldások is lehetségesek. Costa [12] a nagy rangú pontok közül választ egy független halmazt; ezek lesznek a közösségek közepei, majd ρ sugarú gömböket képez körülöttük. Távolságfüggvénynek a G természetes metrikáját használja, amely a ρ paraméter értékétől függően átfedésekhez vezet(het). Egy másik megközelítésben Kovács és társai [26] először egy kifinomult hatásfüggvényt számolnak ki, amely a pontok központiságának mértéke. Ennek alapján nívófelületet képeznek, és a felület kiemelkedéseit azonosítják mint közösségeket.

3. Kiértékelés

Mivel a közösségek (vagy klaszterek) definíciói többé-kevésbé tetszőlegesek, Kleinberg [25], hasznosságuk mérésére is sokféle elgondolás született. Jóllehet ez alapvető kérdés, a kutatók nézőpontjai természetesen eltérőek. Az alábbiakban vázoljuk, hogyan lehet egy-egy közösség fogalom használhatóságát megállapítani. Egy *direkt módszer* közvetlenül hasonlítja össze az adódó közösségeket és a gráfról meglevő egyéb információnkat, míg az *indirekt módszerek* egy modell változójaként kezelik a közösségi információt, és az előrejelzés pontosításának mértékén mérik ennek hasznosságát.

3.1. Tapasztalatok és paraméterezés

Először futtatni kell az algoritmusokat, meg kell kapni az eredményeket és esetleg matematikai következtetéseket levonni bizonyos gráfosztályokról. Nagyon fontos az algoritmusok sebessége. Valódi sebességüket nem könnyű összehasonlítani, mivel ez erősen függ az implementációjuktól és a tesztgráfoktól (gyakorlati gráf avagy elméleti konstrukció).

Mindhárom algoritmus gyors, és általában is a alfejezetben leírt család algoritmusai hatalmas méretű problémák megoldására képesek. A pontban még visszatérünk erre a kérdésre, és közlünk néhány eredményt a futási időkről és a megoldások jóságáról, részletesen lásd Griechisch és Pluhár [22].

A klikk-perkolációs módszer figyelemre méltó mind elméleti, mind gyakorlati szemszögből nézve. Az Erdős-Rényi random gráfok kapcsán alaposan megvizsgálták, Bollobás és Riordan [6], és a gyakorlatban is használhatónak bizonyult, Adamcsek és társai [1]. Mindazonáltal a CPM néha túl nagy közösségeket ad, és a paraméterezése is rejtélyes, hiszen hogyan döntjük el, milyen értéke legyen k-nak?

Az N^{++} algoritmus meglehetősen heurisztikus, elméleti vizsgálata nem kivitelezhető. Fő előnye a sebesség, a közösségek kis átmérője és a megbízhatóság.

Az élklaszterező módszereket még kevéssé vizsgálták. Nyilvánvaló hátrányuk,

hogy az általuk kapott közösségeknek legfeljebb egy közös elemük lehet. Valódi gráfoknál ez túl szoros feltétel.

Néhány benchmark gráfon kipróbáltunk a CPM és az N^{++} algoritmusokat, a tapasztalatokat Zachary híres gráfján illusztráljuk, lásd Zachary [35]. Ez a gráf a baráti kapcsolatokat írja le egy karate klubban, amely éppen a vizsgált időszakban vált ketté. Az egyik rész (A) a japán mesterrel maradt, míg a másik (B) az amerikai helyettesével tartott. A CPM $k = 3$ esetén három közösséget ad, rendre 3, 6 és 24 mérettel, míg $k = 4$-re szintén három közösség keletkezik, melyek mérete 4, 4 és 7. $k = 5$ esetén egyetlen 6 pontú közösség lesz. Itt a $k = 3$ és $k = 4$ esetek közösségeinek kombinálása tűnik jó megoldásnak, és a közösségek ekkor az A és B halmazok *belsejében* húzódnak. Az N^{++} algoritmus 12 közösséget ad, rendre a darabszámok/méretek: 4/3, 5/4, 1/6 és 2/7. Egyet kivéve a közösségek A, vagy B belsejében vannak. A szakadás egy lehetséges magyarázata így éppen az A-t és B-t összekötő közösség felbomlása lehet.

3.2. Grafikus

A korai publikációk általában a gráf valamilyen vizuális formája alapján határozzák meg a közösségeket. A szem által végzett klaszterezések jónak bizonyultak. Az átlapolódó közösségek meghatározása már nehezebb, mert a vizualizáció már nem annyira kézenfekvő.

Egy lehetőség a különböző klaszterezések, közösségek összehasonlítására a gráf lerajzolása és a tetszés szerinti értékelése. A tapasztalat szerint a jó klaszterezések a szem számára is kellemesek, az egy klaszterbe kerülő pontok többnyire közel vannak egymáshoz. A közösségek vizsgálatára már nem olyan egyszerű ilyen módon. Néhány ötlet segíthet, pl. a közösségek metszetgráfjának a megjelenítése. Az $I(G)$ metszetgráfban G közösségei a pontok, és két pont akkor összekötött, ha a közösségek metszete nem üres, azaz $I(G) = (V(H), E(H))$, ahol $V(H) = \mathcal{K}$ és $(C_i, C_j) \in E(H)$, ha $|C_i \cap C_j| > 0$. Hátránya ennek a megközelítésnek, hogy csak kis gráfokon használható, és a klaszterek meghatározása mindig szubjektív.[2]

Ismét a Zachary-gráfot tekintve, lásd Griechisch és Pluhár [22], a CPM egy nem összefüggő H gráfot ad. Az N^{++} által adott H metszetgráf informatívabb. Két sűrű részgráfból áll, melyeknek egy közös x pontja van, amely vágópont H-ban. Az x-nek megfelel egy négy pontból álló C_9 közösség, amely a japán mestert (1), a helyettesét (33) és a 3, illetve 9 számokkal címkézett embereket tartalmazza. (Ez a közösség különben az egyetlen, amelynek nem üres a metszete A-val és B-vel is.) $C_9 \cong K_4 \setminus e$, az egyetlen hiányzó él éppen az $(1, 33)$, ami érthető. Amikor a klub szakadása megtörtént, az elszakította a 3 és a 9 pontot, és ezzel megszűnt a C_9 közösség, amely addig kapocs lehetett a klubban. Kis fantáziával feltételezhető, hogy eleve a 3-as és a 9-es barátsága volt a klub kohéziójának az alapja, és mikor ez már nem viselte el a feszültséget, és megszakadt, akkor az a klub végét is jelentette egyben.

[2]Gráfok vizualizálására a *force directed* algoritmus bizonyult a legjobbnak. Azonban ez $O(n^2)$ időt igényel, ami megakadályozza használatát, ha n milliós nagyságú.

3.3. Véletlen kisvilág gráfok

Sokféle módon lehet véletlen gráfokat generálni, melyek megragadják a kisvilág gráfok egy-egy lényeges tulajdonságát, lásd Barabási és Albert, Cami és Deo [2, 11]. Ezek közül a *Preferential Attachment* (PA) és a *Vertex Copy* (VC) modellekről szólunk részletesebben. Megjegyezzük, hogy másfajta megközelítések is vannak, pl. a véletlen metszetgráf modellt vizsgálja Stark [34].[3]

Mindkét modell rekurzívan definiált; egy már meglévő részgráfhoz vesz hozzá egy új x pontot, de az x szomszédságát másképp generálják. A PA-modellben az x pont k új élt hoz, ezeket egymástól függetlenül és véletlenül kötjük a régi pontokhoz, egy y-hoz a $d(y)$ fokszámmal arányos valószínűséggel. A VC-modellben egy régi s pontot választunk egyenletes eloszlással, és az új x ponttal az $N(s)$ pontjait p valószínűséggel, egymástól függetlenül összekötjük.

A tapasztalatok vegyesek, és többet mondanak a modellekről, mint a CPM, vagy az N^{++} algoritmusokról. Az alábbiakban illusztráljuk a futási eredményeket két, nagyjából egy kategóriába tartozó gráfhalmazon, részletesen [22]. A gráfok 100 pontúak, a G_1 és H_1 gráfokat a PA-modell adja, $|E(G_1)| = 192$, $|E(H_1)| = 358$, míg a G_2 és H_2 gráfokat, amelyekre $|E(G_2)| = 151$ és $|E(H_2)| = 378$, a VC-modell szerint állítottuk elő. A #C és #CO a klaszterek, illetve a közösségek számát jelenti, míg a k fejlécű oszlop a k méretű közösségek száma. A CPM esetében a k fejlécű oszlop viszont az algoritmus k paraméterére utal, amely szerint a futás történt. A klaszterekt Newman modularitás maximalizáló heurisztikája állította elő, lásd a következő alfejezetben.

gráf / algoritmus	#C	#CO	3	4	5	6	7	> 7
G_1 / CPM	10	7	7					
G_1 / N^{++}	10	9	5	0	0	2	1	1
G_2 / CPM	9	17	13	4				
G_2 / N^{++}	9	22	8	7	2	4	1	0
H_1 / CPM	6	10	7	3				
H_1 / N^{++}	6	37	5	2	3	9	7	12
H_2 / CPM	6	24	4	8	6	6		
H_2 / N^{++}	6	26	8	3	2	5	1	7

3.4. Modularitás

A Newman-modularitás [28] a G gráf és komponenseinek alábbi függvénye:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

[3] A metszetgráfokra a CPM hajlamos túl nagy közösségeket adni. A lehetséges javítás erre maximálni a közösségek átmérőjét az N^{++}-hoz hasonlóan.

ahol $m = |E(G)|$, A_{ij} a G adjacencia mátrixa, k_i az i-edik csúcs fokszáma, c_i a komponense és $\delta(c_i, c_j)$ a Kronecker-szimbólum. A klaszterező algoritmusok alapulhatnak valamilyen matematikai vagy fizikai heurisztikán, mint pl. edge-betweenness (EB), eigenvectors (EV), label propagation (LP), spin glass (SG), walk trap (WT), vagy megpróbálják maximalizálni a modularitási függvényt az összes komponensek halmazán valamilyen mohó algoritmussal.

A modularitásra adott formula általánosítható közösségekre, Népusz és társai [27], ha s_{ij}-t írunk $\delta(c_i, c_j)$ helyett, ahol s_{ij} valamilyen i és j közötti hasonlósági mérték. (Jelen esetben u_i az i-edik pont valószínűségi eloszlása a közösségek fölött, és $s_{ij} = \langle u_i, u_j \rangle$, de lehetne bármely $\|u_i - u_j\|$ norma is.)

Másrészt a közösségek *közvetlenül* is megkaphatók a modularitási függvény értékének maximalizálásával is, lásd [22]. Mivel egy kvadratikus célfüggvény maximalizálását kell elvégezni, ez a megközelítés csak kis gráfok esetén lehetséges, bár így is hasznos benchmarkokat ad. Egy másik út az optimum heurisztikákkal való megközelítése, csakúgy, mint a klaszterezés esetén. Egy másik tanulság, hogy a klaszterek és a közösségek szerkezete nem mérhető ugyanazzal a mértékkel, ezért további súlyozást kell használni. Az algoritmusok tesztelésének eredményeit a már jól ismert Zachary-gráfon mutatjuk be. A klaszterezést klikk-perkoláció (CPM) követi, a klikkek mérete $k = 3$ és $k = 4$, az algoritmus N^{++}. A futási idők másodpercben adottak, $\#C$ mutatja a klaszterek, vagy közösségek számát (amelyik adott esetben értelmezett).

algoritmus	modularitás	futásidő	#C
EB	0.4013	0.0100	5
EV	0.3727	0.0000	3
Gr	0.3807	0.0000	3
LP	0.4020	0.0000	3
SP	0.4063	1.1500	6
WT	0.4198	0.0000	4
CPM 3	0.2438	0.012	3
CPM 4	0.2557		3
N^{++}	0.1947	0.6690	12

Algoritmusaink használhatóságát olyan hálózatokon ellenőrizhetjük, amelyek közösségei ismertek. Megfigyelhetők a különféle közösségi hálózatok (telekommunikációs, ismeretségi, Erasmus-kapcsolatok gráfja stb.) működése közötti hasonlóságok, és majdnem minden algoritmus hasznos észrevételeket eredményez. Megállapítható, hogy a közösségeket használó algoritmusok sokkal jobbak, mint a csak klasztereket használók.

3.5. Finomítások, idő és rendezések

Végezhetünk a grafikus módszerhez hasonló tanulmányokat is, ha van valami-

lyen, az éleken vagy a csúcsokon értelmezett függvényünk. Látunk néhány nagyon szubjektív, de mégis említésre méltó jelenséget.

i. Mindenekelőtt a klaszterek rendszerint jóval nagyobbak, mint a közösségek, és a számuk is kevesebb.

ii. A közösségek száma akár a hatványtörvényt is követheti/követi, bár ezt ellenőrizni nem lehetséges.

iii. A közösségek rendszerint a klasztereken belül vannak, és ezeknek egy finom szerkezetét mutatják. A fordított irány is előfordul, ilyenkor a klaszterek adnak információt a közösségekről. Azaz a legérdekesebb közösségek azok, amelyek elemei több klaszterhez tartoznak.

iv. A szociális gráfokban meggyőződtünk a gyenge kapcsolatok szerepéről, Granovetter [20], és vizsgáltunk is néhány algoritmust. Az N^{++} által kapott közösségeken belül szinte kizárólag csak erős élek vannak, míg a gyenge élek a közösségek között vannak. A kisvilág gráfok másik típusánál az ún. technikai gráfoknál[4] ilyet nem tapasztaltunk. Adatainkat Hidalgó és társai [23] cikkéből vettük. (A CPM nem adott jó eredményt semmilyen k-ra, talán azért, mert túl érzékeny a mérési hibákra és a hiányzó adatokra.)

v. Szociális gráfokban a csúcsoknak természetes attribútuma lehet az az időpont, amikor a csúcs csatlakozott a hálózathoz. Ez a sorrend nem mutatható ki, ha az egész hálózat klasztereit nézzük, de figyelemre méltó az egybeesés, ha csak egy kiválasztott csúcs szomszédságát tekintjük. Ebben az esetben a klaszterek néha jellemezhetők valamilyen időintervallummal, vagy térbeli korláttal. Megjegyzendő, hogy a közösségek átnyúlhatnak a klaszterek határain.

3.6. Dinamikus gráfok

Az alkalmazásokban fellépő gráfok függhetnek az időtől, így esetleg eldöntendő kérdés, melyik formájukat használjuk.[5] Az egyik alapvető feladat a közösségek nyomonkövetése, a változásának a leírása. Ezt Palla és társai [30], illetve Bóta és társai [9] kísérelték meg. A megállapítások hasonló és eltérő elemeket egyaránt tartalmaznak; az utóbbinak sok forrása lehet. Az egyik, hogy míg a [30] kísérletei a CPM, a [9] szerzői az N^{++} algoritmust használták. Különböztek az adatbázisok, a [30] az ún. co-authorship gráfot és egy (amerikai) telefonhívási gráfot, míg a [9] egy banki tranzakciós gráfot és egy (magyar) telefonhívási gráfot elemzett. Végül a metodika is különbözött, a [30] szerzői egyszerű axiomatikus feltételekkel éltek a közösségekkel történhető elemi eseményekre (változatlan marad, eltűnik, kettéválik,

[4] A szociális gráfoknál az (x, y) és (x, z) élek megléte megnöveli az (y, z) él létezésének feltételes valószínűségét, míg a technikai gráfokban ilyenkor ez a valószínűség csökken.

[5] Például a két egymás utáni hónapban a telefonhívásokból előállított gráfok élhalmaza csak kb. 30%-ban egyezik meg.

egyesül, nő, zsugorodik), addig a [9] kísérletei megmutatták, hogy az esetek egy jelentős része nem fér bele ebbe a keretbe. Nyitott kérdés, hogy az élek erőssége összefügg-e azzal, mennyire változó közösségekben húzódnak az élek, lásd még az előző alfejezet **iv.** pontját.

3.7. Súlyozás

Súlyozott gráfokkal nehéz foglalkozni. Jóllehet az indirekt módszerek numerikus eredményei megbízhatóbbak, de ha ezeket kiterjesztjük súlyozott gráfokra, az eredmények még kevéssé ismertek, Bóta [7].

Az alábbiakban az indirekt kiértékelés egy modelljét vázoljuk.[6] Az infekciós modellek a valódi gráfok alkalmazásának középpontjában állnak, Boguña és Pastor-Satorras [4], de alkalmasat konstruálni nehéz. Fő szempontjai: (i) melyik modellt válasszuk, (ii) mik a lényeges változók, és (iii) hogyan határozzuk meg a paraméterek értékét. Vizsgálataink a banki szféra két problémájára koncentrálódtak: 90 napot meghaladó nem fizetés, az ún. *hitel default*, és általában a késedelmes fizetés, Csernenszky és társai [13, 14]. Hangsúlyozzuk, hogy bár a két probléma hasonló, mégis vannak köztük lényeges különbségek.

A fő hasonlóság a fenti két folyamatban, hogy mindkettő ragályos, azaz az üzleti partnereket is megfertőzheti. Mindazonáltal nagy gondossággal kell vizsgálni a jelenségeket, hiszen az üzleti nehézségek nem pusztán a környezetből adódhatnak, belső okai is vannak.[7] Tehát a feladatunk az, ha egy problémára, pl. a hitel default esetén, adottak egy-egy cég *apriori* valószínűségei, akkor becsüljük meg az *a posteriori* default valószínűségeket, amelyek egy fertőzési folyamat után értelmezettek. A valószínűségek különbségét tekinthetjük az adott problémában fellépő *hálózati hatásnak*. A probléma jellege miatt (azaz nincs felépülés, a fertőzés valószínűsége nem konstans az éleken) kizárjuk az epidemiológiában amúgy sikeres SIR vagy SIS modellek használatát. A célunknak legjobban a független kaszkád modell felel meg.

3.8. Független kaszkád modell (IC)

A független kaszkádról, vagy megalkotói alapján a Domingos–Richardson-modellről lásd bővebben Domingos és Richardson, Kempe és társai [16, 24]. Megjegyezzük, hogy a modell egy ekvivalens változatát vizsgálta korábban Granovetter [21].

Adott egy G élsúlyozott gráf, ahol a (v, w) élhez a $p_{v,w}$ valószínűséget társítjuk. Az infekció az alábbi módon történik.

Az első lépésben a fertőzött csúcsok F_1 halmazát tekintjük aktívnak, azaz $F_1 = A_1$.

[6] Más megközelítéssel egy esettanulmányt vizsgálunk, amely bizonyította a hálózati modellek és a közösségek használhatóságát.

[7] A gazdaság általános állapota figyelembe vehető egy fiktív ponttal, amely mindenkivel össze van kötve.

Általánosan a $w \in V(G) \setminus F_{i-1}$ csúcs $p = \prod_{v \in A_{i-1}} p_{v,w}$ valószínűséggel fertőződik meg az i-edik lépésben, és ekkor $w \in F_i$. A frissen fertőzött pontok a *rákövetkező lépésben* fertőzhetnek csupán, azaz $A_i = F_i \setminus F_{i-1}$. Ha valamely i-re $F_i = F_{i-1}$, akkor leáll a folyamat.

Megjegyezzük, hogy a pontok fertőzési valószínűségének kiszámítása nehéz probléma, jobbára szimulációkon alapul, lásd Kempe és társai, Csernenszky és társai [24, 13].

3.9. Súlyozás és optimalizálás

A megfelelő modellhez az IC-modellt módosítanunk kell. Mivel az a posteriori fertőzési valószínűségeket úgyis szimulációkkal becsüljük, kézenfekvő a szimuláció részévé tenni az a priori fertőzési valószínűségeket [14]. Ezzel a kezdeti fertőzés 0-1 értékei helyett teszőleges eloszlást használhatunk. Nagyobb problémát okoz a $p_{v,w}$ élfertőzési valószínűségek becslése, ezt az irányt a fenti cikk mellett az alábbi publikációkban kísérelték meg: Goyal és társai, Saito és társai [19, 33]; sajnos alapvetően különböző feltevésekkel dolgozva.

A megoldás a következőképpen történhet. A szokásos módon *tanuló* és *teszt* adatbázist veszünk fel. A $p_{v,w}$ valószínűségeket a tanulóhalmaz segítségével becsüljük, majd a teszthalmazzal mérjük vissza. A másik probléma, hogy a $p_{v,w}$ valószínűségek becslése alulhatározott problémához vezet; itt azt feltételezzük, $p_{v,w}$ a v, w pontok és a (v, w) élhez tartozó attribútumoknak valamilyen (számunkra ismeretlen) függvénye. Ezt néhány paraméter segítségével fejezzük ki, majd a paramétereket optimalizáljuk, hogy minél jobban közelítse a tanulóhalmazban megadott tényleges fertőzési folyamatot. Végül meg kell választanunk a célfüggvényt, amely a becsléseink jóságát méri. A Bóta és társai [10] kutatásaiban ez a szokásos normákat jelenti, míg az alkalmazás jellege miatt a [14] az ún. *gain curve* megközelítést használta. Ebben a gráf pontjait a modell által (a teszthalmazon) számított fertőzési valószínűség szerinti *fordított sorrendbe* állítjuk. Legyenek ezek a valószínűségek $w_1 \geq, \ldots, \geq w_n$. Definiáljuk a *nyereség* (gain) függvényt a

$$\mathrm{gain}(x) = \frac{\sum_{i \leq x} w_i}{\sum_{i=1}^{n} w_i}$$

formulával, és maximalizáljuk a

$$\int_{x=1}^{n} \mathrm{gain}(x) dx$$

értéket.

A $p_{v,w}$ élfertőzési valószínűségek az alább részletezendő attribútumokból lettek felépítve. Szisztematikus kereséssel lettek kipróbálva a függvények[8], illetve a paraméterezésük. A végső aggregálása a traszformált értékeknek hasonlóan történt, míg a legjobb paraméter értékek keresése *grid search* által történt.[9]

[8] Az alapfüggvények: lineáris, kvadratikus, logaritmus, exponenciális és szigmoid.

[9] A tapasztalat szerint nagyobb feladatok megoldását adhatja a numerikus deriválás és a gradiens módszer megfelelő kombinációja, lásd [10].

3.10. Eredmények

Itt egyetlen kísérletet emelnénk ki a sok lehetséges modell közül. A részletes tanulmányt, amely az OTP KKV szektor adatbázisán alapult, lásd [14]. A tranzakciós adatbázis 2008 augusztus és 2009 április (6 hónapos) időintervallumában rögzített adatin alapult a tranzakciós gráf, míg a fertőzési folyamat 2009 február és április (3 havi) adatait használta. A default események felvétele az alábbi két intervallumban történt: egy hosszabb 2009 május és 2010 április között (12 hónap), egy rövidebb pedig 2009 május és 2009 július között (3 hónap).

A következő tapasztalatok adódtak:

1. A rövidebb (3 hónapos) default monitoron alapuló modellek jobban teljesítenek, mint a hosszabbon.
2. Az élek irányítása lényegés vevő-eladó formában kell felvenni, azaz ha x utal pénzt y-nak, akkor $(x, y) \in E(G)$.[10]
3. Indirekt élek. Ha van $x - z$ és $z - y$ tranzakció, de z nem ismert (pl. nem kliense az OTP-nek), a fertőzési modellben szerepet kaphat (x, y) élként elszámolva, ahol az attributumokra a IV/ii használandó.
4. A lényegesnek bizonyult változók, illetve a rájuk vonatkozó tapasztalatok:

   (i) A közösségi információ. (Adott él tartozik-e közösségbe?)

   (ii) Az (x, y) él örökli az x változóit (de y-ét nem).

   (iii) A relatív forgalom számít, azaz az élen küldött transzfer és a traszfer összegének hányadosa.

   (iv) A kliens életkora. (Milyen öreg egy vállalat?)

   (v) Viselkedés típusú változók (queuing, overdraft stb.).

   Mindazonáltal a legerősebb változók az (i) és (iii) pontban említettek.

A modellek által adott javítás az ún. *lift* segítségével értelmezhetők. A [14] szerint a defaultba eső kliensek megtalálásában a szektortól függően 3-4, egyes szektorokban (a legkockázatosabb ügyfelek esetén) 10-12-szeres lift adódik. A közösségi hatás erős, ha (x, y) egy közösségen belül futó él, akkor kb. háromszoros fertőzési valószínűséggel számolandó, a hasonló, de közösségen kívül futó élhez képest. Hasonló eredményekről számol be a [13] dolgozat.

4. Köszönetnyilvánítás

A kutatásokat az OTKA és a Magyar kormány és az Európai Unió "Social Renewal Operational Programme" keretében működő TÁMOP pályázat támogatta.

[10] A modell irányítatlan élekkel is javítást hoz a hálózatot nem használó modellekhez képest; ezt egyfajta hálózati hatás okozza, hisz a gazdaság szereplői kölcsönös függésben vannak, illetve a hálózat a szektort is megragadja.

Az első szerzőt a TÁMOP-4.2.1/B-09/1/KONV-2010-0005, míg a második szerzőt az OTKA K76099 és futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

[1] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, T. Vicsek CFinder: *Locating cliques and overlapping modules in biological networks.* Bioinformatics **22**, (2006) 1021–1023.

[2] R. Albert and A. L. Barabási: *Emergence of scaling in random networks.* Science **286**, (1999) No. 5439, 509–512.

[3] R. Albert, A. L. Barabási: *Statistical mechanics of complex networks.* Reviews of Modern Physics **74**, (2002).

[4] M. Boguñá, R. Pastor-Satorras, A. Vespignani: *Absence of epidemic threshold in scale-free networks with connectivity correlations.* Preprint cond-mat/0208163, (2002).

[5] B. Bollobás: *Modern Graph Theory.* Springer, New York (1998).

[6] B. Bollobás and O. Riordan: *Clique percolation.* Random Structures Algorithms **35**, (2009) No. **3**, 294–322.

[7] A. Bóta: *Applications of Overlapping Community Detection.* $(CS)^2$ - Conference of PhD Students in Computer Science, Szeged (2010).

[8] A. Bóta, L. Csizmadia and A. Pluhár: *Community detection and its use in Real Graphs.* Proceedings of the 13th International Multiconference INFORMATION SOCIETY - IS (2010) Volume A, 393–396.

[9] A. Bóta, M. Krész and A. Pluhár: *Dynamic Communities and their Detection.* Acta Cybernetica **20**, (2011) 35–52.

[10] A. Bóta, M. Krész and A. Pluhár: *Systematic learning of edge probabilities in the Domingos-Richardson model.* Int. J. Complex Systems in Science, Volume **1(2)**, (2011) 115–118.

[11] A. Cami, N. Deo: *Techniques for analyzing dynamic random graph models of web-like networks: An overview.* Networks **51**, (2008) No. 4, 211–255.

[12] Luciano da Fontoura Costa: *Hub-Based Community Finding.* arXiv:cond-mat/0405022 v1 3 May 2004.

[13] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár and T. Tóth: *The use of infection models in accounting and crediting.* Challenges for Analysis of the Economy, the Businesses, and Social Progress, Szeged (2009).

[14] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár and T. Tóth: *Parameter Optimization of Infection Models.* $(CS)^2$ - Conference of PhD Students in Computer Science, Szeged (2010).

[15] L. Csizmadia: *Recognizing communities in social graphs.* MSc thesis, University of Szeged, (2003).

[16] P. Domingos, M. Richardson: *Mining the Network Value of Costumers.* 7th Intl. Conf. on Knowledge Discovery and Data Mining, (2001).

[17] T. S. Evans and R. Lambiote: *Edge Partitions and Overlapping Communities in Complex Networks.* arXiv:0912.4389v1, (2009).

[18] S. Fortunato: *Community Detection in graphs.* arXiv:0906.0612

[19] A. Goyal, F. Bonchi and L. V. S. Lakshmanan: *Learning influence probabilities in social networks.* WSDM '10 Proceedings of the third ACM international conference on Web search and data mining ACM New York, NY, USA (2010) doi: 10.1145/1718487.1718518

[20] M. Granovetter: *The Strength of Weak Ties.* American Journal of Sociology **78(6)**, (1973) 1360–1380.

[21] M. Granovetter: *Threshold models of collective behavior.* American Journal of Sociology **83(6)**, (1978) 1420–1443.

[22] E. Griechisch: *Clustering and community finding methods in graphs.* MSc thesis, University of Szeged, (2010).

[23] C. A. Hidalgo, B. Klinger, A. L. Barabási and R. Hausmann: *The Product Space Conditions the Development of Nations.* Science (2007) **317**: 482–487.

[24] D. Kempe, J. Kleinberg and E. Tardos: *Maximizing the Spread of Influence though a Social Network.* Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, (2003).

[25] J. Kleinberg: *An Impossibility Theorem for Clustering.* Advances in Neural Information Processing Systems (NIPS) 15, (2002).

[26] I. A. Kovács, R. Palotai, M. S. Szalay and P. Csermely: *Community Landscapes: An Integrative Approach to Determine Overlapping Network Module Hierarchy, Identify Key Nodes and Predict Network Dynamics.* (2010) PLoS ONE **5(9)**: e12528. doi:10.1371/journal.pone.0012528

[27] T. Népusz, A. Petróczi, L. Négyessy and F. Bazsó: *Fuzzy communities and the concept of bridgeness in complex networks.* arXiv:0707.1646v3, (2007).

[28] M. E. J. Newman: *The structure and function of complex networks.* Preprint cond-mat/0303516 (2003).

[29] G. Palla, I. Derényi, I. Farkas and T. Vicsek: *Uncovering the overlapping community structure of complex networks in nature and society.* Nature **435**, (2005) 814.

[30] G. Palla, A.-L. Barabási and T. Vicsek: *Quantifying social group evolution.* Nature **446**, (2007) 664–667.

[31] A. Pluhár: *A telefonos logfile-on alapuló ismeretségi gráfok klasztereiről.* Research Report (2001).

[32] A. Pluhár: *Ismeretségi gráfok közösségeinek meghatározása gyors algoritmusokkal.* Research Report (2002).

[33] K. Saito, R. Nakano and M. Kimura: *Prediction of Information Diffusion Probabilities for Independent Cascade Model.* Knowledge-Based Intelligent Information and Engineering Systems Lecture Notes in Computer Science, (2008) Volume 5179/2008, 67–75, DOI: 10.1007/978-3-540-85567-5_9

[34] D. Stark: *The vertex degree distribution of random intersection graphs.* Random Structures and Algorithms **24(3)**, (2004) 249–258.

[35] W. W. Zachary: *An information flow model for conflict and fission in small groups.* Journal of Anthropological Research **33**, (1977) 452–473.

(Beérkezett: 2011. 10. 18.)

BARTALOS ISTVÁN
Szegedi Tudományegyetem
Természettudományi és Informatikai Kar
Informatikai Tanszékcsoport (Kalmár László Intézet)
6720 Szeged, Árpád tér 2.
Levelezési cím: 6701 Szeged, Postafiók 652.
bartalos@inf.u-szeged.hu,

PLUHÁR ANDRÁS
Szegedi Tudományegyetem
Természettudományi és Informatikai Kar
Informatikai Tanszékcsoport (Kalmár László Intézet)
6720 Szeged, Árpád tér 2.
Levelezési cím: 6701 Szeged, Postafiók 652.
pluhar@inf.u-szeged.hu

COMMUNITIES AND THEIR ROLE IN SMALL WORLD GRAPHS

István Bartalos and András Pluhár

We survey and unify the methods developed for finding overlapping communities in Small World graphs and make some attempt to evaluate those. We also demonstrate how these community information help in graph mining or in the investigation of complex graph models that have succesful applications.

Alkalmazott Matematikai Lapok **29** (2012), 67-100.

ÉRZÉKENYSÉGVIZSGÁLATOK A STATISZTIKAI ELJÁRÁSOKBAN

TAKÁCS SZABOLCS

Bizonyos matematikai eljárások fontos, kihagyhatatlan része az úgynevezett érzékenységvizsgálat. E vizsgálat során arra vagyunk elsősorban kíváncsiak, hogy a különböző inputadatok megváltozása következtében feladatunk megoldása (eredménye) milyen mértékben változik – illetve milyen viselkedést mutat. Érdekes kérdés lehet az is, hogy milyen input változások esetén nem módosul a megoldás, ahogyan az is, hogy mely input adatok lesznek nagyobb, mely input adatok pedig kisebb hatással a kimeneti adatok változásaira.

A statisztikai kérdésfelvetések során más és más területeken eltérő fogalmi háttérrel vizsgálhatjuk ezt a jelenséget. Ahogy majd látni fogjuk: mást jelent az érzékenység a becsléselméletben, mást egyes hipotézisvizsgálati módszereknél és megint mást jelent az elsősorban modellezésre használt eljárások esetében.

Cikkünkben nem kívánunk teljes betekintést nyújtani e vizsgálati módszerek széles tárházába és alkalmazásába – pusztán arra vállalkozunk, hogy felvázoljuk e terület széles alkalmazási spektrumát. Szeretnénk továbbá felhívni a figyelmet ezen – általában kiegészítő – eljárások fontosságára.

A cikkben nem célunk új matematikai állítások megfogalmazása – sokkal inkább bizonyos kérdések felvetése, melyekre a cikk megírása során tett kutatómunkánk kapcsán nem találtunk megnyugtató válaszokat.

1. Bevezető

A statisztika az egyik leginkább alkalmazott területe a matematikának: számtalan területen jelen van kutatási eszközként, alkalmazói pedig nem feltétlenül matematikusok. Például Prékopa [37] műszaki alkalmazásokat tartalmazó könyve is segédanyagként szolgálhat azok számára, akik nem matematikusként, de műszaki területeken kívánják a statisztikát alkalmazni. Azonban a könyv nem tartalmazza (mert nem is tartalmazhatja) a tudományterület néhány olyan sajátosságát, melyek az utóbbi évtizedekben kezdtek teret nyerni, hiszen jellemzően mind számításigényes eljárások.

Számos tudományterület foglalkozik azzal a kérdéssel, hogy egyes kísérletek végeredménye milyen mértékben, illetve milyen módon függ a bemeneti adatoktól.

Mely bemeneti adatok azok, melyekre nézve a kísérlet stabilitást mutat és melyek azok, amelyek esetleg az egész kísérlet érvényességét veszélyeztetni tudják?

A kísérletek érvényessége, eredményessége – ha úgy tetszik, a kimeneti adatok bemeneti adatoktól való érzékenysége – fontos kutatási sarokpont, melyre nem minden kutatási folyamat során jut elég figyelem, vagy ha úgy tetszik, nem is feltétlenül vizsgálat tárgya egyes kísérletekben.

Egyre gyakrabban olvasni olyan tudományos, vagy tudományt népszerűsítő cikkeket, ahol a bemeneti adatokkal való, nem eléggé körültekintő bánásmód téves, vagy legalábbis nem igazolható következtetések levonására adott okot. Erre lehet példa LeVay, a Science folyóiratban megjelent tanulmánya [31] – melyet azóta többen is megkérdőjeleztek, illetve eredményeit cáfolták. A szerző e cikkében HIV-fertőzött homoszexuális és nem HIV-fertőzött, heteroszexuális férfiakat vizsgált haláluk után, és agyi struktúrájukban markáns eltérésekre bukkant. Azonban a halál közvetlen okaként szolgáló betegséget „elfelejtette" vizsgálata tárgyává tenni – később kiderült, hogy az eltérésekért nem a szexuális beállítottság, hanem maga a HIV-vírus a felelős (lásd pl. Bayne és társai tanulmányát, melyben kifejtik, hogy többek között a HIV-vírus okozta elváltozások kiszűrése után semmifajta hatását nem tudták kimutatni a szexuális orientációnak).

A kérdés persze úgy is felvethető, hogy ebben az esetben a figyelmetlenség okozta-e az adatokban való különbségek hibás értelmezését – vagy egy olyan szokásjog esetleges megléte, mely a bemeneti adatok különbségeiben való alaposabb vizsgálódás hiányát eredményezhette?

Ugyanis statisztikai oldalról persze úgy értelmezhető a kérdés, hogy a HIV-státusz figyelmen kívül hagyása, vagy ha úgy tetszik, nem megfelelő kezelése olyan különbségeket eredményezett a kimeneti adatokban, melyekből az azóta megjelent tanulmányok szerint, téves következtetés sikerült levonni.

Így persze felvetődik a kérdés: a statisztikai eljárásoknál az érzékenység (a bemeneti adatok változékonyságának, vagy változásának a kimeneti adatok vizsgálatának fényében) maguknak a módszereknek sajátja, vagy külön is érdemes rájuk kitérni?

Cikkünkben megpróbáljuk néhány statisztikai terület esetén az „érzékenységvizsgálat" analóg fogalmait bemutatni, illetve kitérni a fenti kérdésre: a statisztikai eljárásoknak e vizsgálat sajátja kellene, hogy legyen? Vagy netán a különböző eljárásoknál – a bemeneti adatok bizonyos anomáliái vagy tulajdonságai esetén – kiegészítő vizsgálatokra lenne szükség?

A cikkben három nagyobb egységet különíthetünk el. Az első nagyobb fejezetben az egész cikk során használt statisztikai módszerek rövid, áttekintő bemutatását olvashatjuk. Külön kitérünk a becsléselmélet és a hipotézisvizsgálatok főbb pontjaira. A második rész az érzékenységvizsgálatokról szól a statisztikai módszerek alkalmazása esetében. 3 nagyobb részfejezetre bontottuk a kérdést: érzékenységvizsgálatok a becsléselméletben, ahol a módszereket részint a mintanagyság, részint pedig a vizsgált paraméterek esetére osztályoztuk. A második részfejezetben a hipotézisvizsgálatok esetét tárgyaljuk, külön kitérve bizonyos speciális módszerekre, nem hagyományos statisztikai eljárásokra. A harmadik részfejezetben

egy biostatisztikai módszert mutatunk be – egy konkrét példán is végigvezetve az olvasót.

2. Statisztikai bevezető

E fejezetben bemutatjuk azokat a statisztikában használt definíciókat, illetve fogalmakat, melyekre a cikk olvasása során szükségünk lehet. Alapvetően három területre koncentrálva gyűjtöttük össze ezeket a formulákat: egyik oldalról a becsléselmélethez kapcsolódó eljárásokra és elnevezésekre koncentrálunk, másik oldalról pedig az ezzel erősen összekapcsolható hipotézisvizsgálati fogalmakat is szeretnénk bemutatni.

A harmadik terület valójában algoritmusok gyűjteménye: szimulációs technikák, melyeket statisztikai eljárások során alkalmazhatunk. Egy szimulációs módszert mi is bemutatunk e fejezet végén.

2.1. Becsléselmélet

Az alább található bevezető definíciók lényegében bármely, bevezető statisztikai könyvben, jegyzetben megtalálhatók. Angol nyelven Lehmann pontbecslésekről szóló könyve [29], magyarul akár Borovkov [9], akár Bolla és Krámli [6] frissebb kiadású könyvei említhetők, illetve egyetemi jegyzetek formájában szintén magyar nyelven Prékopa [37] vagy Mogyoródi [33] munkái lelhetők fel.

A becsléselmélet alkalmazása során az alábbi statisztikai kérdésekre keressük a választ.

Legyen adott egy X véletlen változó és egy $(\Omega, \mathcal{A}, P)$ valószínűségi mező.

Az $X : \Omega \to \mathbb{R}$ valószínűségi változónk adott θ paraméterét szeretnénk megbecsülni. E kérdésfelvetésre azért is szükség lehet, mert a becslési eljárások számos módon függhetnek vizsgálatunk tárgyát képező paramétereinktől.

Amiben minden becslési eljárás megegyezik: veszünk egy $X_1, \ldots, X_n$, n elemű mintát, mely minta segítségével:

$$T(X_1, \ldots, X_n) : \mathbb{R}^n \to \Theta$$

statisztika alapján becslést készítünk $\theta \in \Theta$ paraméterre.

Becslésünk jóságát általánosságban a

$$d(T(X_1, \ldots, X_n); \theta),$$

megfelelő d metrikában mért eltéréssel mérhetjük.

Megjegyzés. Gyakori a $d(a, b) = (a - b)^2$ négyzetes eltérés használata, alkalmazása.

Legyen $E(T(X_1, \ldots, X_n)) = \theta^*$ és jelölje $u = \theta^* - \theta$ a statisztikai eljárásunk torzításának mértékét.

2.1. Definíció. Amennyiben $u = 0$, úgy a $T(X_1, \ldots, X_n)$ becslést torzítatlan becslésnek szokás hívni.

Megjegyzés. Általában nem ad félreértésre okot, de érdemes megjegyezni, hogy θ^* elméleti paraméter (pl. elméleti átlag, elméleti szórás, elméleti ferdeség, elméleti csúcsosság).

A $T(X_1, \ldots, X_n)$ statisztika konkrét értékére a tapasztalati paraméter (tapasztalati átlag, tapasztalati szórás stb.) elnevezéssel szokás élni.

Azaz, a véletlen változó eloszlásának elméleti jellemzőjét szeretnénk a tapasztalati, mintából számított paraméterek segítségével megbecsülni.

Legyen $\delta(T(X_1, \ldots, X_n))$ a $T(X_1, \ldots, X_n)$ becslés valamely szóródási mutatója.

Többnyire a szórást[1] választjuk szóródási mutatónak, de érdemes azt is figyelembe venni, hogy a δ szóródási mutatót a d metrikával összhangba hozzuk, illetve akár vizsgálat tárgya is lehet a metrika és a szóródási mutató egymáshoz való viszonya. Például ha $d(a, b) = |a - b|$ választással élünk, akkor δ-ra az átlagos abszolút eltérés bizonyos szempontból jobb (indokoltabb) választásnak látszik az átlagos négyzetes eltérés (szórás) helyett.

A standard hiba így például az alábbi

$$H(X_1, \ldots, X_n) = u + \delta(X_1, \ldots, X_n)$$

összegként definiálható. Ez felfogható úgy is, hogy az eljárás hibája nem más, mint a becslés torzításának és – pusztán mert véletlen jelenségeket vizsgálunk – az eredendő eltéréseknek az együttese.

2.2. Definíció. Amennyiben a becslés torzítatlan (tehát $u = 0$), úgy ha teljesül, hogy

$$\lim_{n\to\infty} H(X_1, \ldots, X_n) = 0,$$

a becslést konzisztens becslésnek nevezzük. Tehát a konzisztens becslés egy olyan torzítatlan becslés, melynek standard hibája a mintaelemszám növelésével tetszőlegesen csökkenthető.

Megjegyzés. A két metrika, d és δ szerepe igen eltérő. Vegyük azt a példát, hogy attól függetlenül hogy mit is szeretnénk becsülni, mi mindenképpen egy konstans értéket mondunk: legyen ez 42. Így a $\delta = 0$ esettel állunk szemben – azaz, hacsak nem 42 a valódi paraméter, amit becsülni szeretnénk, úgy az eljárásunk „véletlen vizsgálatából fakadó" hibáját kiiktattuk, csak a torzítás marad.

[1] A szórás a variancia négyzetgyöke, azaz az átlagtól való átlagos négyzetes eltérés négyzetgyöke. Azonban ennek viselkedése és így megbízhatósága erősen függ a vizsgált változónk eloszlásától, ahogy ezt Lee és munkatársai dolgozatukban [28] megállapítják – erről a későbbiekben, részminták szórásának tesztelésekor részletesebben szót ejtünk.

Így a statisztikánk jóságát mérő d metrikában a véletlen szerepét kiiktattuk – de az eljárásunk valódi paramétertől való eltérését ettől még mérni fogjuk.

Amennyiben egyes paraméterekre több becslési eljárás is létezik (és általában létezik), akkor a lehetséges becslések közül az alábbi módon szokás választani:

2.3. Definíció. Két becslés közül azt nevezzük hatékonyabbnak, melynek kisebb a hibája adott mintanagyság mellett.

A fenti definíciók értelmében egy adott paraméterre az elérhető leghatékonyabb becslést érdemes választanunk (amennyiben az létezik).

Léteznek más megközelítések is egy-egy becslés elkészítésének vizsgálatakor. Világos, hogy az eddigiekben azt tekintettük alapnak, hogy a becslésünkből számított tapasztalati paraméter és elméleti paraméter várhatóan milyen távol lesznek egymástól.

Becslést alkothatunk úgy is, ha mintában rejlő információnk vizsgálatából indulunk ki:

2.4. Definíció. Legyen az X $(X_1, \ldots, X_n)$ független azonos eloszlású minta az X háttérváltozó eloszlásából, amely tehát a θ paramétertől függ, $\theta \in \Theta$. Feltesszük azt is, hogy $\dim(\theta) = 1$, és hogy Θ konvex. Ekkor a minta úgynevezett Fisher-féle információja:

$$I_n(\theta) = E\left[\left(\frac{\partial}{\partial\theta} l_\theta(x)\right)^2\right] > 0,$$

ahol az $l_\theta(x)$ az úgynevezett loglikelihood függvény, azaz a tapasztalati sűrűségfüggvény[2] logaritmusa.

Ez vezet az úgynevezett maximum-likelihood becslésekhez, amikor is lényegében arról van szó, hogy a minta alapján leginkább valószínű θ paramétert (eloszlást) választjuk a Θ paramétertérből.

Megjegyzés. Megjegyezzük, hogy másfajta becslési eljárásokat találhatunk, ha a

$$H(X_1, \ldots, X_n) = u + \delta(X_1, \ldots, X_n)$$

hibából elindulva úgy gondolkodunk, hogy az eltéréseket – annak mértékétől függően – más és más módokon büntetjük. Ezt a veszteséget nevezhetjük akár rizikónak is (bizonyos határig nem érdekel minket az eltérés vagy a torzítás, míg egy

[2]A tapasztalati sűrűségfüggvény lényegében egy oszlopdiagramként fogható fel (vagy annak simításaként). Technikailag úgy kell elképzelni, hogy a valószínűségi változó értékkészletét ekvidisztáns módon felosztjuk (a változót diszkretizáljuk) – majd az adott intervallumok relatív gyakoriságait ábrázoljuk. Az értékkészletet felosztó intervallumok számára általában $\sqrt{n}$ értéket választanak, ha $n < 100$, míg $1 + \log_2(n)$ értéket, amennyiben $n \geq 100$.

adott határt átlépve az eltérésekért például exponenciális módon fizetnünk kell). Ilyenkor értelemszerűen azt a $\theta \in \Theta$ paramétert fogjuk választani, ahol a veszteségünk (vagy rizikónk) minimális.

A becsléseinket sokszor az alábbi megközelítésben érdemes tárgyalni: tegyük fel, hogy most rendelkezünk két, $T_1(X_1, \ldots, X_n)$ és $T_2(X_1, \ldots, X_n)$ statisztikával (becsléssel) a $\theta^* \in \Theta$ paraméterre.

2.5. Definíció. Ekkor a $(T_1(\underline{X}), T_2(\underline{X}))$ intervallum legalább $1-\varepsilon$ szintű konfidenciaintervallum a θ^* paraméterre, ha

$$P(T_1(\underline{X}) < \theta^* < T_2(\underline{X})) \geq 1 - \varepsilon,$$

ahol $\varepsilon > 0$. A $1-\varepsilon$ az úgynevezett konfidenciaszint.

Megjegyezzük, hogy általánosítható bármely $f(\theta^*)$ függvényére a paraméternek e fenti felírása, ilyen esetben a

$$P(T_1(\underline{X}) < f(\theta^*) < T_2(\underline{X})) \geq 1 - \varepsilon$$

egyenlőtlenségnek kell fennállnia.

2.2. Hipotézisvizsgálat

Az előző fejezetben megalkottuk a konfidenciaintervallumokat, melyek azzal a tulajdonsággal bírtak, hogy vagy a $\theta^* \in \Theta$ paramétert, vagy annak valamely függvényét tartalmazták adott valószínűséggel. Ilyenkor azonban döntéseket is tudunk hozni – mely döntések átvezetnek minket a hipotézisvizsgálatok területére.

A hipotézisvizsgálatok során – igazodva most a becsléselméletben alkalmazott jelöléseinkhez – az alábbi módon járunk el általában: legyen $H_0 : \theta \in \Theta_0$ és $H_1 : \theta \in \Theta_1$, ahol $\Theta_0 \bigcap \Theta_1 = \emptyset$ és $\Theta_0 \bigcup \Theta_1 = \Theta$.

Fontos feltétele a hipotézisvizsgálatoknak, hogy a $T(\underline{X})$ statisztikánk eloszlását H_0 esetén ismernünk kell. A döntéshozatal felfogható olymódon, hogy e H_0 feltételezés mellett megalkotunk egy – a korábbi fejezetben már ismertetett, ε szintű konfidenciaintervallumot.

Ez az intervallum az alábbi módon interpretálható: amennyiben H_0 feltételezés igaz, úgy bármely, adott eloszlásból származó minta esetén számított $T(\underline{X})$ statisztika értékének legalább $1-\varepsilon$ valószínűséggel az adott intervallumba kell esnie.

A konfidenciaintervallumot elfogadási tartománynak nevezzük, míg annak komplementer halmazát kritikus tartománynak. Amennyiben a mintánkból számított $T(\underline{X})$ az elfogadási tartományba esik, úgy a H_0 nullhipotézis mellett döntünk és azt mondhatjuk, hogy a minta nem mond ellent e feltételezésnek (adott ε szinten). Míg ha a $T(\underline{X})$ a kritikus tartományból vesz fel értéket, úgy a H_1 ellenhipotézist választjuk és azt mondhatjuk, hogy az adott minta alapján a H_0 nullhipotézis teljesülése valószínűtlen (adott ε szint mellett), így e hipotézist elvetjük.

Világos, hogy ilyen esetekben két hibát[3] követhetünk el: ha a nullhipotézist nem tartjuk meg, pedig igaz, akkor az úgynevezett elsőfajú hibát követjük el – ennek valószínűsége legfeljebb ε, így elmondható, hogy a konfidenciaszint segítségével az elsőfajú hiba becsülhető. Szokás mind ε-t, mind $1-\varepsilon$ mennyiséget szignifikanciának, vagy szignifikanciaszintnek nevezni – általában nem okoz félreértést egyik vagy másik használata. Jelölésben hagyományosan α használatos a szignifikanciaszintre (nem a becsléselméletben használt ε).

A másikfajta hibát akkor követjük el, ha a nullhipotézist elfogadjuk, holott az nem teljesül. Ezt a hibát másodfajú hibának nevezzük és a statisztikai eljárás erejével becsülhető. E hiba mértékét β-val szokás jelölni, és a próba erejét β vagy $1-\beta$ jelöli (és a szignifikanciához hasonlóan itt sem szokott félreértést eredményezni egyik vagy másik mennyiség használata).

Megjegyzés. Fontos kiemelnünk: míg az elsőfajú hiba felülről becsülhető a szignifikanciaszinttel, addig a másodfajú hibát nem tudjuk becsülni. A hipotézisvizsgálati eljárások (próbák) ereje így általában csak adott helyzetben, tapasztalati úton az adott problémára vonatkoztatva kimérhető mennyiségek.[4]

2.3. Egy szimulációs módszer

A szimulációs technikák általában nem találhatók meg bármely bevezető statisztikai könyvben, azonban széles körben használtak, így a statisztikai bevezető fejezetben ezeket az eljárásokat is ismertetjük vázlatosan. A bevezetőben – miután a továbbiakban is csak erre koncentrálunk – az úgynevezett bootstrap eljárást ismertetjük, mely részletesen megtalálható például Efron e témában klasszikusnak számító cikkében [15].

A becsléselméletben már definiált hibát explicit formában a legritkább esetben lehet megadni, így például Monte-Carlo-módszer segítségével, szimulációval becsülhetjük.

– *(1.)* $\widehat{\theta}(x_1,\ldots,x_n)$ a statisztika értéke. (Egy adott $X_1=x_1,\ldots,X_n=x_n$ realizáció mellett.)

  Ekkor $\sigma(\theta)=\sqrt{Var_\theta(X_1,\ldots,X_n)}$ a statisztika valódi hibája.

  Ezt többnyire lehetetlen zárt formában felírni.

– *(2.)* Miután F eloszlást nem ismerjük, ezért $\widehat{F}$-pal, a tapasztali eloszlásfügg-

[3] Gondoljunk a farkast kiáltó pásztorfiú esetére. A *farkaskiáltás* tekinthető az úgynevezett elsőfajú hibának: nincsen gond a vizsgált rendszerben, mégis hibáról, problémáról teszünk jelentést. A másodfajú hiba ennek ellentéte, nevezhetjük *struccpolitikának* – a mesében a harmadik farkaskiáltás után a falusiak viselkedése: gond van a rendszerben, és mégsem veszünk róla tudomást.

[4] Az elsőfajú hiba mindig azt jelenti, hogy az adott, fix eloszlás mellett sikerült egy valószínűtlen mintát vennünk, melyből elutasítottuk a nullhipotézisben feltett eloszlásunkat. Azonban a másodfajú hiba azt jelenti, hogy a nullhipotézis nem az, aminek gondoljuk – viszont ez számtalan módon bekövetkezhet, ezért nem tudjuk egzakt módon megmondani e hiba valószínűségét, csak például szimulációkat készíteni az adott, konkrét minta ismeretében. Úgy is fogalmazhatunk, hogy a döntéshozatalunkhoz minden esetben az adott szignifikancia-szinten döntő, legerősebb próbára van szükségünk.

vénnyel becsüljük. Ekkor $\widehat{\sigma}_B = \sigma\left(\widehat{F}\right)$[5] becsüli $\sigma(F)$-et.

Itt csak approximációról van szó, hiszen ezt sem tudjuk zárt alakban felírni.

Megjegyzés. A tapasztalati eloszlásfüggvény nem más, mint hogy a lehetséges realizációkból megmondjuk, hogy a véletlen változónak adott értékei milyen valószínűséggel vétetnek fel (folytonos változó esetén adott értéknél nem nagyobb értékeket, vagy milyen valószínűséggel vesz fel a véletlen változó). E technikával tehát egy lépcsős függvényt nyerünk, mely a mintaelemek értékei esetén $\frac{1}{n}$ függvényértéket emelkedik.

A bootstrap eljárás ezek után egy független, azonos eloszlású, egyszerű, visszatevéses mintavételezés a tapasztalati eloszlásfüggvény alapján.

Ez tehát nem más, mint egy $U(X_1, \ldots, X_n)$, $X_1, \ldots, X_n$ pontokra koncentrált diszkrét egyenletes eloszlás szerint vett újabb és újabb véletlen mintavételezés.

Így tehát egy approximációs eljárást kell végrehajtanunk, mely a következő lépésekből áll.

- **(i)** $\widehat{F}$ meghatározása.
- **(ii)** $\widehat{F}$-ból független mintavétel segítségével $X_1^*, \ldots, X_k^*$ úgynevezett bootstrap minta létrehozása. Itt be kell tartanunk, hogy $\forall i: \; P(X_i^* = x_j) = \frac{1}{n}$. (Minden mintaelem ugyanolyan valószínűséggel veheti fel a realizációban szereplő különböző értékeket). *Azaz: a mintából független módon választunk, visszatevéses mintavételezéssel k darabot.*
- **(iii)** $\widehat{\theta}^* = \theta(X_1^*, \ldots, X_k^*)$ bootstrap másolatból származó statisztika kiszámítása.
- **(iv)** az **(ii)** és **(iii)** lépések B számú ismétlése. Így előállítunk egy $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_B^*$ független boostrap másolatból származó statisztika-becslés mintát.
- **(v)** $\widehat{\sigma}_B$ approximáció kiszámítása az alábbi formula segítségével:

$$\widehat{\sigma}_B = \sqrt{\sum_{b=1}^{B} \frac{\left(\widehat{\theta}_b^* - \widehat{\theta}_\bullet^*\right)^2}{B-1}},$$

ahol

$$\widehat{\theta}_\bullet^* = \frac{1}{B} \sum_{b=1}^{B} \left(\widehat{\theta}_b^*\right).$$

[5] $\widehat{\sigma}_B$ az approximációs eljárás utolsó lépésében formalizálásra kerül, mely tehát nem más, mint a tapasztalati eloszlás szórása.

Megjegyzés. Ekkor, ha $B \to \infty$, úgy a $\widehat{\sigma}_B$ közelíti $\sigma(F)$-et. B optimális megválasztásáról nincsenek különösebb viták: általában elegendő 100 és 500 közötti bootstrap minta kiszámítása.

Más filozófia alapján folytatható addig az **(ii)**-**(v)** lépések egymásutánja, ameddig a lépésenként kiszámított és korrigált $\widehat{\theta}_\bullet$ valamilyen, előre meghatározott, minőséget előíró korlátnál kevesebbet változik 1 lépés alatt.

Fontos megjegyezni, hogy az approximáció konvergenciájához elégséges feltétel a véges szórásnégyzet (közös eloszlást feltételezzünk fel itt is az X_i változókra nézve), mely – mint azt már láttuk, a centrális határeloszlás tétel teljesülése miatt szükséges.

A bootstrap algoritmus egyik előnye az, hogy a tapasztalati eloszlásfüggvényből táplálkozva lehetőséget biztosít számunkra, hogy pl. a tapasztalati kvantilisek becslésével tapasztalati konfidenciaintervallumokat is meghatározzunk.

3. Érzékenységvizsgálatok

A bevezető fejezetek után rátérhetünk az érzékenységvizsgálatok kérdésére. Már a két bevezető fejezetből is érzékelhető, hogy a statisztikában igen fontos – de gyakran nem elég hangsúlyos – terület az adatok érzékenységének vizsgálata.

Más megközelítésben: a hagyományos eljárások sok helyen, sok formában elérhetők, megtalálhatók – ezek alkalmazása azonban feltételekhez kötött. Annak ismerete, vizsgálata, hogy e feltételek sérülése esetén mi történik a vizsgálatunk kimeneti adataival, nem teljesen kidolgozott. Értjük ezalatt azt, hogy bár a módszerek gondosan felsorolják az alkalmazhatóság feltételeit, nem szólnak arról, hogy mit kellene tenni, ha egyes feltételek sérülnek. A könnyen elérhető programcsomagok nem feltétlenül tartalmazzák a feltételek vizsgálatait, ennek következtében az alternatív eljárások végképp nem kerülnek bemutatásra.

3.1. Becslések érzékenységvizsgálata

Becsléseink elkészítésekor három olyan pont is megemlíthető, mely garantáltan befolyásolja a becslésünk minőségét, jóságát.

1. A d metrika: különböző metrikákban a becslésünk jóságát más és más eltérések fogják befolyásolni – így azt is megállapíthatjuk, hogy attól függően, hogy mely eltérésekre vagyunk érzékenyebbek, esetleg eltérő becsléseket kell majd alkalmaznunk.

2. A n mintanagyság: általánosan megfogalmazható az az elvárás, hogy egy véletlen jelenséget vizsgálva a mintanagyság növelésével egyre jobb becsléseket nyerjünk – de legalábbis ne romoljon a becslésünk minősége.

3. X véletlen változó eloszlása: e harmadik tulajdonság nem biztos, hogy elsőre szembetűnő, de viszonylag könnyen elfogadható, ha arra gondolunk, hogy egy olyan véletlen változó, mely pl. sűrűbben vesz fel extrém nagy, vagy éppen extrém kicsi értékeket, ugyanazon T statisztikára nézve merőben más viselkedést tud mutatni, mint pl. egy dichotóm véletlen változó.

Ezek után felmerül a kérdés: a becsléselméletben, egyes becslések alkalmazása során e három kritérium közül melyekre rendelkezik a statisztika érzékenységvizsgálatra vonatkozó válaszokkal, illetve mely területekre kell még esetleg válaszokat keresni?

E kérdéskört első megközelítésben az úgynevezett standard hibák meghatározása jelenti. A standard hibát általában négyzetes módon határozzák meg – mi ennél általánosabban, a becslési eljárás hibájáról fogunk szólni.

3.1.1. A metrikák

Jól felfogott érdekünkben használunk többesszámot e részfejezet címében: nem mindegy ugyanis, hogy a becslési eljárás véletlentől való függését mérő δ-t szeretnénk vizsgálni – vagy pedig a valódi paraméter és a becsült paraméter d-vel jelölt várható eltérését.

Megjegyzés. Általánosságban az úgynevezett standard hibát szokás a becslések esetén meghatározni, mely az elméleti és a tapasztalati paraméter eltéréséből származtatott átlagos eltérés.

A soron következő példákhoz tartozó vizsgálatokat megtalálhatjuk például Jones és Gill 1998-as cikkében [24].

Megjegyzés. Többször fogunk élni az alábbi jelöléssel: $f\,(\alpha; df)$. Ez azt jelenti, hogy az adott f típusú eloszlás, df szabadsági fokhoz tartozó, α szignifikanciaszintjének úgynevezett kvantilise.

Például $1,89 = t(0,05; 7)$ azt jelenti, hogy a 7 szabadsági fokhoz, $\alpha = 0,05$ szignifikancia-szinthez tartozó kvantilise[6] az úgynevezett t-eloszlásnak (vagy Student-féle t-eloszlásnak).

3.1. Példa. Az első négy tapasztalati momentum konfidenciaintervallumát az alábbi módokon határozhatjuk meg:

– Átlag:

$$\overline{X} \pm t_{(\frac{\alpha}{2}, n-1)} \frac{s^2}{\sqrt{n}},$$

azaz az átlag esetén kis mintáknál (például $n \leq 100$) a megfelelő szabadságfokú és megbízhatósági szintet használó t-eloszlás kvantilisével dolgozunk,

[6]Ez a kvantilis az eloszlásnak az a pontja, melyre igaz, hogy a 7 szabadságfokú t-eloszlásból származó véletlen változó 1,89-nél kisebb értéket 95, tehát ennél nagyobb értéket 5%-os valószínűséggel vesz fel.

nagy mintáknál a standard normális eloszlás is használható a t-eloszlás helyett.

Megjegyzés. Megfigyelhető, hogy az átlag becslése így konzisztens: a standard hibája a minta végtelenbe tartása mellett 0-hoz konvergál – amennyiben véges a szórása a vizsgált véletlen változónknak.

– Szórás:

$$\sqrt{\frac{(n-1)s^2}{\chi^2_{\left(\frac{\alpha}{2},n-1\right)}}} \leq \sigma \leq \sqrt{\frac{(n-1)s^2}{\chi^2_{\left(1-\frac{\alpha}{2},n-1\right)}}}.$$

– Ferdeség:

$$g_1 = \frac{\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^3}{n}}{\left(\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2}{n}\right)^{\frac{3}{2}}}, \qquad G_1 = \frac{\sqrt{n(n-1)}}{n-2}g_1,$$

$$SES = \sqrt{\frac{6n(n-1)}{(n-2)(n+1)(n+3)}}.$$

Innen a ferdeség konfidenciaintervalluma:

$$G_1 \pm z_{\left(\frac{\alpha}{2}\right)} SES,$$

ahol $z_{\left(\frac{\alpha}{2}\right)}$ nem más, mint a standard normális eloszlás eloszlásfüggvénye inverzének értéke az $\frac{\alpha}{2}$ helyen. Ez utóbbi az alábbi módon is írható:

$$\frac{G_1}{SES} \sim Z,$$

azaz $\frac{G_1}{SES}$ eloszlása standard normális[7].

– Csúcsosság:

$$a_4 = \frac{\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^4}{n}}{\left(\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2}{n}\right)^2}, \qquad g_2 = a_4 - 3,$$

$$G_2 = \frac{n-1}{(n-2)(n-3)}\left((n+1)g_2 + 6\right).$$

[7] A standard normális eloszlást szokás Z-vel jelölni, az eloszlásban való viselkedést pedig $\sim$ segítségével.

A csúcsosság standard hibája:

$$SEK = 2SES\sqrt{\frac{n^2-1}{(n-3)(n+5)}}.$$

Így a csúcsosság konfidenciaintervalluma meghatározható, hiszen

$$\frac{G_2}{SEK} \sim Z.$$

E fenti konfidenciaintervallumok meghatározásakor felmerülhet a kérdés, hogy az átlagra vonatkozó konfidenciaintervallum leggyakoribb alkalmazása, nevezetesen az egymintás t-próba miként viselkedik abban az esetben, ha a normalitás feltételét nem tudjuk garantálni.

Megjegyzés. Egy fontos megjegyzést kell itt tennünk. Majd a későbbiekben még látni fogjuk, hogy a normalitás esetén nem feltétlenül az a legnagyobb problémánk, hogy az átlagot miként tesztelhetjük, hanem már azon is el kell gondolkodnunk, hogy az átlagot teszteljük-e egyáltalán?

Gondoljunk itt arra, hogy az átlagnak van egy olyan, szükségszerű háttérjelentése, melyet az elméleti paraméter okán hordoz: nevezetesen a várható érték miatt az átlag interpretációjához hozzá tartozik, hogy „ezt az értéket várjuk". Azonban ha például társasjátékot játszunk egy hatoldalú dobókockával, akkor egészen biztosan lehetünk abban, hogy − bár a várható értéke a dobásainknak $3,5$ − a játékot játszók közül senki sem várja, hogy $3,5$-et dobjon. Azt azonban mindenki elfogadja, hogy a dobások fele $3,5$ alatt, míg másik fele $3,5$ felett lesz. Ez azonban a medián, tehát ilyen esetben indokoltabbnak látszik ezt tesztelni − még ha meg is egyezik az értéke szimmetrikus eloszlások esetén az átlaggal.

Ebben a témában számos publikáció látott napvilágot, a teljesség igénye nélkül: a közelmúltban jelent meg magyar nyelven Vargha összefoglaló cikke [45] a Statisztikai Szemlében, illetve idézhető két klasszikusnak számító, t-próba próbastatisztikáján módosítást javasló cikk: Johnson 1978-as cikke [23], illetve egy korábbi, 1949-es cikk Gayentől [17]. E két utóbbi cikkben az alábbi módosításokat javasolják a t-próba[8] próbastatisztikáján:

$$t_{JOHNSON} = t + G_1\sqrt{n}\left(\frac{1}{6n} + \frac{\left(\overline{X}-\mu_0\right)^2}{3s^2}\right),$$

[8]A t-próba (vagy student-próba) egy ismert, klasszikus statisztikai próba. Ennek során a vizsgált nullhipotézisünk: $H_0: E(X) = \mu_0$, próbafüggvénye $t = \frac{\overline{X}-\mu_0}{\frac{s}{\sqrt{n}}}$, ahol $\overline{X}$ és s megegyezik a korábbi jelölésekkel. A t próbastatisztika tehát a mintából számított próbastatisztika (így maga is véletlen), melynek eloszlása az úgynevezett t-eloszlás − amennyiben X véletlen változó eloszlása normális, illetve teljesül a nullhipotézis.

míg Gayen azt mondja, hogy a szokásos $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ helyett az alábbi függvényt[9] használjuk:

$$f(x) = \phi(x) - \frac{G_1}{3!}\phi^{(3)}(x) + \frac{G_2}{4!}\phi^{(4)} + \frac{G_1^2}{72}\phi^{(6)}(x),$$

ahol $\phi^{(r)}$ az r-edik deriváltat jelenti, míg G_1 és G_2 a fent már definiált tapasztalati ferdeség és csúcsosság.

Innen azt is láthatjuk, hogy Johnson módosítása a ferde eloszlások esetén nyújt segítséget számunkra, míg Gayen mind a ferdeséget, mind a csúcsosságot korrigálja módosításában.

E fenti paraméterek viselkedéséről és tulajdonságairól, illetve a standard hibák viselkedéséről széles körben lehet még további szakirodalmat találni, többek között:

- A különböző változók, véletlen jelenségek bizonyos paramétereinek (általában átlag) standard hibáinak összefoglaló táblázata több helyen is megtalálható, erre példa lehet [49]. E táblázatokból arra vonatkozóan kaphatunk információkat, hogy jól specifikált véletlen jelenségek esetén, azok elméleti paraméterét milyen pontossággal lehetett megbecsülni – adott mintanagyság mellett.

- Efron és Tibshirani Statistical Science folyóiratban megjelent cikkükben [15] empirikus és elméleti eredményeket foglalnak össze a bootstrap metódus kapcsán. Ezt az eljárást alkalmazhatjuk különböző paraméterekre vonatkozó standard hibák és konfidenciaintervallumok meghatározására, illetve vizsgálják e módszer általános statisztikai tulajdonságait is (például különböző becslési eljárásokban való viselkedését).

- Belia és munkatársai cikkükben [2] felhívják a figyelmet az általunk is feltett egyik kérdésre, illetve tapasztalatra. E témakörben ugyanis számos anomália van jelen: rosszul interpretált adatokkal és következtetésekkel találkozhatunk e szerzők szerint (tételesen megneveznek idézett cikkükben tanulmányokat), és az általuk idézett tanulmányokban a tanulmányt jegyzők konfidenciaintervallumok és/vagy standard hibák helytelen meghatározása, ábrázolása vagy értelmezése után vonnak le hibás vagy megkérdőjelezhető következtetéseket.

- Végül – egyáltalán nem utolsó sorban, átvezetendő a mintanagyság problémájához e kérdéskört – a Judkins által vizsgált, Fay-féle eljárásban [25] arról van szó, hogy becslésünk megbízhatósága drasztikus mértékben romlik, ha a mintavételezési eljárásunk során nem tudtuk a mintaelemeink függetlenségét

[9] A hagyományos t-próbába kisebb elemszámok esetén a t-eloszlást használjuk, míg nagyobb elemszám esetén (gyakorlatban például 150-nél nagyobb mintáknál) a standard normális eloszlást. Gayen azt javasolja, hogy a normalitás sérülése esetén e két, általánosan használt eloszlás helyett e módosítottat alkalmazzuk inkább. Hangsúlyozzuk, hogy Johnson és Gayen módosításait akkor használjuk, ha szakmailag még mindig indokolt az átlag bárminemű tesztelése a normalitás sérülése esetén. Ellenkező esetben – ahogy már említettük – más középértékek tesztelése indokolt.

garantálni (ez könnyedén előfordulhat többek között szociológiai vizsgálatoknál, hiszen például az egy munkahelyen dolgozók, vagy az egy iskolában tanulók semmiképpen sem tekinthetők függetlennek). Ennek hatásvizsgálatát egy korábbi cikkünkben [44] mutatjuk be esettanulmányként, ahol az OECD által szervezett oktatáspolitikai felmérés adatainak elemzésén a különböző módszerek hatásmechanizmusát elemezzük. A Fay-féle eljárás egy másik aspektusát – a már említett, Efronék [15] által is vizsgált szimulációs eljárással való kapcsolatát – taglalja Saavedra egy előadásában [40].

Megjegyzés. Ez utóbbi tanulmánnyal rá is világíthatunk e kérdéskör egy újabb problémájára: ha úgy találjuk, hogy valamely eljárás biztonságát szimulációs technikák segítségével szeretnénk vagy tudjuk vizsgálni, még akkor sem egyértelmű, hogy mely szimulációs eljárást válasszuk.

Felmerülhet e felsorolás után a kérdés: a hibás döntések e kérdéskör (az érzékenységvizsgálat) elhanyagoltsága, nem kellően fontosnak tartott mivolta miatt keletkeznek – vagy valójában az alkalmazott eljárásoknak kellene olyan biztonsági hálót tartalmazniuk, melyek a hibás döntéseket is kellően megszűrik?

Ez alatt érthetjük például azt, hogy az alkalmazók számára könnyen elérhető statisztikai programcsomagokban az eljárások nem feltétlenül tartalmazzák az adott eljárások feltételeinek teljes vizsgálatát – és ha bizonyosakat tartalmaznak, úgy nem feltétlenül azokat, melyek miatt a tapasztalatok szerint leginkább instabillá válhatnak az eljárások. Egészen pontosan: a programcsomagok általában képesek a feltételek ellenőrzésére – csak azok nem feltétlenül képezik egy-egy eljárás szerves részét. Ne felejtsük el megemlíteni, hogy akár így is előfordulhat a már idézett LeVay féle fiaskó [31].

3.1.2. A mintanagyság

Bizton állíthatjuk, hogy e kérdés szakirodalma és e kérdésben elvégzett vizsgálatok kellő támpontot tudnak nyújtani bárki számára azon kérdés eldöntésében, hogy egyes paraméterek vizsgálata során az adott paraméter és a kiválasztott minta esetszáma között milyen jellegű összefüggések adódnak.

Első feltételezésünk az lehet, hogy a populációnk, melyet vizsgálunk végtelen. (Egészen más a helyzet ugyanis, ha véges populációkkal dolgozunk, erről is lesz még szó.)

A végtelen populációk esetén az elmélet a konzisztens becslések biztonságára hívja fel a figyelmet, illetve azokat a becsléseket részesíthetjük előnyben, melyekről összefoglalóan azt mondhatjuk el: a mintaelemszám növelésével csökken a korábban már definiált hibájuk.

3.2. Példa. A mintanagyság döntően befolyásolja a becsléseink pontosságát és így a belőlük levonható következtetéseket is. Tegyük fel, hogy az általunk vizsgált populációban a két nem magasságát szeretnénk összehasonlítani. A férfiak (1) és nők (2) testmagasságának átlagára és korrigált tapasztalati szórására az alábbi eredményeket kapjuk:

$$\overline{X}_1 = 180,001\,cm \quad s_1 = 10\,cm,$$
$$\overline{X}_2 = 180\,cm \quad s_2 = 10\,cm.$$

A fenti adatok természetesen kitaláltak a probléma érzékeltetése érdekében.

Tegyük fel, hogy első esetben a két minta nagysága $n_1 = n_2 = 100$. Ebben az esetben a kétmintás t-próba próbastatisztikája:

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{(n_1 - 1)\, s_1^2 + (n_2 - 1)\, s_2^2}} \sqrt{\frac{n_1 n_2\,(n_1 + n_2 - 2)}{n_1 + n_2}} = 0,0007,$$

azaz nincsen szignifikáns különbség a két változó között, hiszen a szokásos 5%-os szignifikancia-szint melletti kritikus érték $1,96$ lenne.

Azonban ha a mintanagyságot drasztikusan megnöveljük, $n_1 = n_2 = 10^9$ értékekre, úgy $t = 2,236$ adódik, ami már szignifikáns eltérést jelez. Azaz: a mintanagyság növekedése – minden más paraméter fixen tartása mellett – automatikusan csökkenti az elsőfajú hiba valószínűségét, ennek következtében viszont anomáliák adódhatnak. Egy ilyen anomália a fenti: $0,001$ cm-es eltérés tehát szignifikáns különbségként jelentkezik e próbában – amit igen nehéz komoly eltérésként értelmezni.

Cohen azt javasolja [12] könyvében, hogy az ilyen helyzetekre alkalmazzuk kiegészítő mutatóként az átlagok standardizált különbségét, mely nem más, mint

$$\Delta_{Cohen} = \frac{\overline{X}_1 - \overline{X}_2}{s} = 0,0001,$$

ahol $s = 10$, a teljes minta korrigált tapasztalati szórása.

Amennyiben ez az érték 0,3 alatti, úgy azt mondhatjuk, hogy (bár lehet szignifikáns az eltérés), az szakmailag gyenge hatást mutat. Amennyiben 0,7 feletti Δ értéket tapasztalunk, úgy szakmailag jelentős eltérésre bukkantunk – a köztes értékek szakmailag közepes hatást jeleznek. Azaz: a becslésünk pontosságának javulása automatikusan eredményezi a stabilabb, pontosabb döntéshozatalt – ám ez nem feltétlenül jelent szakmailag is releváns eltéréseket.[10]

Megjegyzés. Fontos kiemelni: a testmagasságokat ilyen módon összehasonlító példánkban a statisztikai döntéshozatal addig terjed, hogy megállapítsuk a szignifikáns eltérések jelenlétét. A döntésünk szakmai utóélete már nem a statisztika, hanem az adott, statisztikát alkalmazó tudományterület feladata és felelőssége.

[10] A fenti példával élve: azért, mert van egy teljes földkerekséget felölelő becslésünk a férfiak és nők testmagasságáról, melyből azt tapasztaljuk, hogy a férfiak magassága szignifikánsan nagyobb $0,001$ cm-rel, nem fogjuk minden építészeti főiskolán és egyetemen azt tanítani, hogy az új tudományos eredményeinknek köszönhetően minden újonnan építendő sportlétesítmény férfiaknak szánt öltözőjébe tegyenek egy kicsivel keskenyebb linóleumot, hogy a magasságbéli különbségeket mostantól korrigáljuk.

Lehmann egyik, becsléselmélettel foglalkozó könyvében [29] számos tételt találhatunk arra vonatkozóan, hogy a véletlen változó bizonyos tulajdonságai mellett milyen hibahatárok érhetők el[11]. E könyv második fejezetében egzisztencia állításokat találhatunk, továbbá olyan feladatokat, problémákat tárgyal, melyben konkrét becslésekre (pl. átlag, szórás, kovariancia) hol az úgynevezett rizikó, hol pedig a Fisher-információ segítségével vizsgálja a becslések jóságát, illetve elemzi a kívánt mintanagyságot.

Hasonlóan ide köthetők elméleti megközelítések alapján a különböző „nagy számok törvényei", illetve a különböző, becslésekre vonatkozó egyenlőtlenségek (Markov, Csebisev).

Annak megválaszolására, hogy adott bizonytalanság eléréséhez milyen mintanagyságra van szükségünk többféle módon is választ kaphatunk, többek között:

– Amennyiben ismerjük a becslésünk eloszlását, úgy meghatározható segítségével a becslésünk úgynevezett konfidenciaintervalluma. Erre közismert példa a mintaátlag és annak standard hibája [29], de ismert a szórás (mely Cochran tétele értelmében az átlagtól független módon becsülhető) konfidenciaintervalluma is (pl. Cochran cikkében [11] megtalálható). Ezeket a formulákat már korábban bemutattuk.

  Fletcher és Webster cikkükben [16] a ferdeség hatását vizsgálták különböző becslésekben, míg szintén a ferdeséggel, illetve az eloszlás csúcsosságával összefüggésben, ezen két paraméter becslésének jóságát vizsgálták Wright és Herrington [47] tanulmányukban, akik azt tapasztalták, hogy már kisebb minták esetén is stabilabb becslés mondható e két paraméterre szimulációs eljárásokkal (ők a bootstrap eljárást használták), mint a paraméterek ismert standard hibájának felhasználásával.

  Mameli és munkatársai tovább is mennek alkalmazásaikban ennél: 2012-ben írt cikkükben nagy mintás[12] elemzéseken, orvosi alkalmazásokkal is kiegészítve (illetve valós adatokon tesztelve), összehasonlítják módszerüket a hagyományos, illetve egy paraméteres bootstrap eljárás eredményeivel.

– Kis minták esetén felmerülő anomáliák feloldására adnak támpontot az úgynevezett „breakdown point" elemzések (lásd alább). E témakör kutatásai arról adnak számot, hogy egyes becslések, illetve belőlük származtatott hipotézisvizsgálati eljárások miként viselkednek a minta egyes elemeinek torzulásakor.

[11] Gondoljunk itt arra az egyszerű feladatra, hogy például az átlag standard hibája a megismert $\frac{s}{\sqrt{n}}$ formulával határozható meg. Ha előírjuk a hibahatárt és ismert a szórás, akkor meg tudjuk mondani, hogy adott szórás mellett mekkora mintára van szükségünk annak érdekében, hogy várhatóan az előre megadott hibahatáron belül tudjuk tartani a becslésünket.

[12] A kis és nagy minták általában nem egzakt megfogalmazások. Egy 20-30 elemszámú mintát még kis mintának szokás nevezni, míg egy 80-100 esetet vizsgáló realizáció már tekinthető nagy mintának. A mintánk elemszáma, annak „nagysága" általában attól függ, hogy mit is vizsgálunk, vizsgálatunkban használt próbastatisztika mennyire érzékeny. Lásd például e fejezet következő pontjában található „breakdown point" analízist.

Megjegyzés. Gondoljunk itt arra, hogy például az átlag számítását egyetlen mintaelem megváltoztatása is tetszőlegesen módosíthatja – más megközelítésben a mintaátlag instabil paraméternek tekinthető e fent nevezett elmélet értelmében.

Ezzel szemben például a medián lényegesen nagyobb tűréshatárral bír akár még egészen kis minták esetén is (például egyetlen mintaelem akár végtelenbe tartása esetén sem fog nagyfokú ingadozást mutatni).

A „breakdown point" elemzés tehát az adott paraméterekre vonatkozóan a becslés egy olyan értéket adja meg, hogy az adott mintanagyságok mellett a minta mekkora hányada módosítható úgy, hogy a minta egésze a becslésre vonatkozóan ne váljon használhatatlanná.

– Átlag: miután az átlagot $\overline{X} = \frac{X_1+\cdots+X_n}{n}$ formulával határozhatjuk meg, világos, hogy ha az első $n-1$ értéket fixnek tekintjük és $X_n \to \infty$ feltételt nézzük, úgy az egész átlagra is teljesül, hogy $\overline{X} \to \infty$.

  Így a véges „breakdown point" $\frac{1}{n}$, míg asszimptotikusan 0.

– Medián: az átlaggal szemben ha elképzeljük, hogy az sorba rendezett minta $\left\lfloor \frac{n-1}{2} \right\rfloor$ legkisebb elemet fixáljuk, úgy látható, hogy a felső, ugyanennyi elem (mediánnál nagyobbak) szabadon növelhetőek, a medián értékét nem módosítják.

  Így a véges „breakdown point" $\left\lfloor \frac{n-1}{2n} \right\rfloor$, míg aszimptotikusan $\frac{1}{2}$.

Azaz érzékenység szempontjából a medián lényegesen jobban viselkedik, mint az átlag – hiszen az adataink közel felét megváltoztatva is stabilitást mutat ez az paramétere az eloszlásnak. Erre vonatkozóan a következőkben egy példával is érzékeltetni fogjuk a két középérték közötti különbséget egy versenyhelyzet értékelése kapcsán.

A „breakdown point" elemzésekről egy speciális esetben értekezik Camponovo és Otsu 2012-ben megjelent cikkükben [10], ahol a szerzők a későbbiekben még szintén tárgyalt bootstrap eljárás viselkedését figyelték az extrém értékek megjelenésének fényében.

Az ezen téma iránt érdeklődő Olvasó számára egy összefoglaló, a fenti példát is tartalmazó, az egymintás t-próba esetét taglaló jegyzetet ajánlhatunk kiindulópontnak, melyet 2006-ban publikált Geyer [18] – és mely jegyzetben e téma néhány alap eredményét foglalja össze, illetve ad támpontot további kutatásokhoz, számításokhoz.

Elmondható tehát, hogy bizonyos paraméterek esetén ismerjük azok becslésének eloszlását – így tudjuk, hogy várhatóan milyen hibát vétünk a becslési eljárás alkalmazásával. Azonban kis minták esetén, vagy olyan paraméterekre, melyek eloszlása nem ismert, ilyen információval nem rendelkezünk.

E helyzetek feloldására tűnik elfogadható empirikus megoldásnak a korábbiakban már említetteken túl a különböző szimulációs technikák alkalmazása.

3.1.3. Véges sokaságok esete

A véges sokaságokról több helyen is szerezhetünk információkat, pl. Lehmann becsléselméleti, továbbiakban is még idézett könyvében részint a mintavételezési problémákról (3. fejezet 6. alfejezet), részint például M-becslésekről (5. fejezet, 6. alfejezet), melyekre vonatkozóan tapasztalati eredményeket is találhatunk az idézett műben.

E fejezetben külön találhatunk számos információt a Huber-féle robusztus becslési eljárásról (Huber-féle simított becslésnek is nevezik). A Huber-féle eljárás során lényegében kombináljuk a medián és az átlag információit, ennek segítségével alkothatunk robusztus becslést az átlagra – azonban feltétele az eljárásnak az eloszlás szimmetriája.

3.3. Példa. Legyen $X_1, \ldots, X_n$ független, azonosan $P_{\mu,\sigma}$ eloszlásból származó minta, ahol

$$f_{\mu,\sigma} := \frac{1}{\sigma} f\left(\frac{x-\mu}{\sigma}\right)$$

alakú[13]. Ekkor az M-becslés a μ eltolás paraméterre azon t érték, melyre:

$$\sum_{i=1}^{n} \phi\left(\frac{X_i - t}{\sigma}\right) \to \min_t,$$

vagy más megközelítésben:

$$\sum_{i=1}^{n} \psi\left(\frac{x_i - t}{\sigma}\right) = 0,$$

ahol $\psi = \phi'$. Megjegyezzük, hogy a maximum-likelihood becslés esetén $\phi_f = -log(f)$, míg $\psi_f = -\frac{f'}{f}$.

Speciális esetben a fenti simítási eljárás a következőképpen módosítható, alkalmazható. Adott k konstans mellett az úgynevezett Huber-féle becslés vagy transzformáció az alábbi:

$$\psi_k(x) = \begin{cases} k & x > k, \\ x & -k \le x \le k, \\ -k & x < k. \end{cases}$$

Megjegyzés. A fenti függvény egyfajta trimmelésként[14] is felfogható. Azonban míg a trimmelés esetén a kiugró értékektől megszabadulunk – ezzel a minta

[13] Feltesszük, hogy az F eloszlás szimmetrikus, továbbá feltehető, hogy $\sigma = 1$. Az eljárásban tehát az eloszlásfüggvényt ismertnek tekintjük, a két fenti paramétert pontbecslés segítségével becsüljük.

[14] A trimmelés azon statisztikai eljárás, melyben a kiugró vagy extrém értékeket levágjuk, kihagyjuk a mintából – centralizálva így a mintánkat, illetve csökkentve annak szabadásfokát.

szabadságfokát, esetszámát is csökkentve – addig itt az esetszám megmarad, csak egy adott értéken túl a számunkra megválasztott szint (k) kerül az adott szintnél nagyobb és kisebb esetek helyére. Más megközelítésben a kiugró értékeket egy számunkra beállított toleranciaszintre kényszerítjük vissza, ha úgy tetszik centrálunk.

Ezt az eljárást vizsgálta, illetve módosította Hampel munkatársaival, melyet 2011-ben publikáltak. E simítás azért is lehet fontos számunkra, mert a simítás a mintanagyság figyelembe vételével történik. Tanulmányukban kitérnek arra is, hogy az eljárást mind a Huber-féle transzformációra, mind a maximum-likelihood becslésre, mind pedig egyéb M-becslésekre alkalmazzák – ráadásul a simítási eljárásukat minden esetben össze is hasonlítják az eredeti eljárásokkal. Tapasztalataik szerint a simított eljárás minden esetben jobb (vagy legalábbis nem rosszabb) eredményeket hozott, mint nem simított változatuk.

3.1.4. A véletlen változó eloszlása

A korábban, a bootstrap szimuláció kapcsán már említettük, hogy a becslés eloszlásának ismerete segítségével a bizonytalanság, az eljárásunk érzékenysége vizsgálható. Azonban azt is tudnunk kell, hogy a véletlen jelenség eloszlása nagyban befolyásolja a becslési eljárásunkat (egyáltalán, már azt is befolyásolja, hogy mely paraméterekre szeretnénk becslést mondani és mely paraméterek nem érdekesek számunkra).

Lehmann [29] több eloszlás esetén is tárgyalja különböző paraméterek becslési tulajdonságait, azok viselkedését konzisztencia, torzítatlanság szempontjából, illetve hatékonyságukat is vizsgálja. Sak és munkatársai ennél tovább is mennek egészen friss kutatási riportjuk [41] tanúsága szerint, melyben azt vizsgálják, hogy különböző eloszlások ferdeségi mutatója miként hat az átlag konfidenciaintervallumára, illetve ezt milyen empirikus módszerekkel lehet korrigálni. Azt tapasztalták, hogy Hall 1992-ben publikált transzformációja [20] hatékony eszköznek bizonyul annak érdekében, hogy az átlagra vonatkozó konfidenciaintervallumot továbbra is zárt formula segítségével, szimulációk nélkül határozhassuk meg.

3.4. Példa. Míg az eredeti t-próba próbastatisztikája

$$t = \frac{\overline{X} - \mu}{\frac{s}{\sqrt{n}}},$$

addig a Hall-féle transzformáció:

$$g_1(t) = t + \frac{1}{\sqrt{n}} G_1 \left(\frac{1}{3} t^2 + \frac{1}{6} \right) + \frac{1}{3n} \left(\frac{1}{3} G_1 \right)^2 t^3,$$

ahol G_1 a tapasztalati ferdeség, tehát ilyen szempontból Hall transzformációja a Johnson-féle, ferdeséget korrigáló eljárással rokon[15].

[15] A t-próbának, mint már említettük, feltétele, hogy a vizsgált változó normális eloszlású legyen. Ennek egyik lehetséges ellenőrzése is lehet, hogy a normális eloszlás – szimmetrikus

A t-próba korrekciójának akkor van csak ilyen jellegű jelentősége, ha a normalitás sérülése mellett továbbra is a várható értéket (átlagot) szeretnénk tesztelni. A felmerülő probléma érzékeltetésére képzeljük el a következő esetet.

3.5. Példa. Adott két középiskolai osztály, akik futóversenyt szeretnének rendezni. Az összehasonlítás alapja a két osztály átlagos futásteljesítménye lesz. Az egyik osztályban csupa élsportolót találunk: 27 atlétát és 3 szumóbirkózót. A másik osztályban sok átlagos diák mellett (29 fő) egyetlen nagyon túlsúlyos diák is tanul. Azonban e túlsúlyos diák – megismerve az ellenfél adottságait – cselt eszel ki. A futóversenyt a Margitszigeten rendezik meg, egyetlen kört kell futni. A diákok nekikezdenek – de túlsúlyos egyedünk csak sétál, mellette a 3 szumóbirkózóval. Az atléták természetesen gond nélkül gyorsabbak az átlagos középiskolás diákoknál – de a csel még nem teljesedett ki. Hősünk beszélgetést kezdeményez a birkózókkal és a beszélgetést a gasztronómia irányába tereli. Majd a sziget egy céltól és rajttól egyaránt távoli pontján lévő talponálló büféhez vezeti a gyanútlan birkózókat. Ott aztán pénzt nem kímélve etetni kezdi őket. A trükk ugyanis a következő: a 3 birkózó – még akár az utolsó pár méteren le is hajrázhatják majd a továbbra is sétáló, velük tartó egyetlen túlsúlyost – eredményei már úgyis olyannyira fogja az egész osztályuk átlagát rontani, hogy bármely, átlagot összehasonlító eljárásban toronymagas győztesként kerül majd ki a teljesen átlagos középiskolai osztályunk. Ez azonban nyilván amiatt alakulhat ki, hogy az eloszlásaink, melyek az osztályokat jellemzik ferdék (például túl sok jó/átlagos és aránylag kevés rossz futó van), továbbá az átlagot egyetlen extrém érték is bármilyen irányba el tudja mozgatni. Így az átlag helyett más mutatóval, eljárással kellene döntenünk a két osztály összehasonlításában (ahogy ezt a „breakdown point" elemzésben már megállapíthattuk). Ha azt a kísérletet végeznénk el, hogy páronként futtatjuk őket, mely párokat véletlenszerűen válogattuk ki egyik és másik osztályból, úgy érzékelhető, hogy a sporttagozatos osztály esetén csak minden 10. választás lesz olyan, ahol az átlagos középiskolából választott diáknak lenne valami esélye – feltéve, ha onnan nem a túlsúlyos egyedet választjuk. Ez utóbbi kísérletet sztochasztikus egyenlőség vizsgálatnak nevezzük és Wilcox már korábban is idézett könyve [46] tartalmaz ilyen – vagy hasonló – helyzetekre alkalmazható próbákat, eljárásokat.

3.1.5. Összefoglaló megállapítások a becslésekhez

Megállapíthatjuk tehát az alábbiakat:

– Amennyiben ismert az eljárásunkból származó becslés eloszlása (pl. a mintaátlag alkalmazása ilyen), akkor zárt formulák segítségével meghatározható az eljárás standard hibája (vagy általánosságban hibája), melynek segítségével a becslésünk pontossága, konfidenciaintervalluma meghatározható. Ennek segítségével tehát képet kaphatunk arról, hogy a valószínűségi változó adott

lévén – $G_1 = 0$ értékkel rendelkezik, azaz a ferdesége 0. Magyarán, ha azt tapasztaljuk, hogy az eloszlásunk ferde – pozitív vagy negatív irányba „eldől", akkor a ferdeségi együttható segítségével korrigáljuk a próbastatisztikánk értékét.

paraméterének becslése esetén milyen hibákat követhetünk el: a véletlen jelenségre mennyire érzékeny a becslésünk.

- Amennyiben nem ismert az eljárásunk eloszlása, úgy szimulációs eljárások bevetésével tudunk képet kapni arról, hogy az adott minta sajátosságaiból következően milyen várható hibákat követünk el az adott paraméter vagy paraméterek becslése során.
- A szimulációk helyett – a kezdeti tapasztalatok sikeressége okán – említhetjük például Hampel és munkatársai, 2011-ben publikált simítási eljárását is [21], mely szintén alkalmazható lehet annak érdekében, hogy a becsléseink bizonytalanságát pontosabban meghatározhassuk.

Megjegyzés. Fontos kiemelni, hogy a fenti felsorolás messze nem teljes. Például nem szóltunk a Bayes-becslések problématikájáról, illetve azok érzékenységéről, ezen keresztül nem adtunk számot azokról az esetekről, amikor rizikó vagy információ (és nem közveltenül az eltérés) alapján akarjuk vázolni a becslés jóságát. Bayes-becslések érzékenységéről, annak vizsgálatáról és függéséről pl. az a-priori eloszlások[16] befolyásáról olvashatunk Lavine 1991-es cikkében [27].

Nem beszéltünk a hiányzó értékek problémájáról vagy arról, ha az adott változóval összefüggő más változókról is rendelkezünk információkról. Erről például Robins és munkatársai értekeznek könyvükben [39], ahol hiányzó értékek esetén való becslések érzékenységvizsgálatára találhatunk módszereket, lehetőségeket.

A témák szerteágazó volta miatt célunk nem is lehetett mindenre kiterjedő – továbbra is a kérdések felvetését tartjuk inkább fontosnak.

3.2. Hipotézisvizsgálatok

A hipotézisvizsgálatok nyilván jelentős mértékben összefüggnek az előző kérdéskörrel: amennyiben van becslésünk és tudjuk annak megbízhatóságát (konfidenciaintervallumát), akkor lényegében hipotézisekről is tudunk döntéseket hozni.

Azonban a hipotézisvizsgálat során több, egymástól funkciójában is igen eltérő hibát tudunk elkövetni.

3.6. Példa. Tegyük fel, hogy egy betegséget szeretnénk diagnosztizálni, melynél az is gondot jelent, ha valakit betegnek mondunk a vizsgálatok alapján – pedig nem az, illetve akkor is gondban vagyunk, ha kiengedjük kezelés nélkül, pedig szüksége lenne rá.

Gondolhatunk itt egy rákos megbetegedésre, aminél a hibás diagnózis bármely kimenetele veszélyeket rejt: ha nem kezeljük, akkor esetleg menthetetlenné válik a beteg, míg ha kezelünk egy egészséges pácienset például kemoterápiával, úgy könnyen megbetegíthetjük.

[16] A Bayes-féle becslésekben azt feltételezzük, hogy maga a vizsgált paraméter is egy véletlen változó, melynek az úgynevezett a-priori (tapasztalás előtti) eloszlása adott. A vizsgált paraméter a-posteriori (tapasztalás utáni) eloszlása nem más, mint az a-priori eloszlás minta esetén vizsgált feltételes eloszlása. A Bayes-becslés pedig az a-posteriori eloszlásból számított paraméterbecslés.

Nyilvánvalóan vannak betegségek, melyeknél valamely kimenetel nem hordoz ekkora kockázatot: ha megszúrom a mutatóujjamat egy tűvel és a baleseti sebész nem hajlandó egy teljes műtőstábot összehívni a problémám elhárítására, majd hazaküld – nagy valószínűséggel nem követ el végzetes hibát. Másik oldalról, ha egy egészséges embernek C-vitamint írok elő, várhatóan nem fog neki ártani, így nagyobb gondot sem fogok vele okozni.

A statisztikai érzékenységvizsgálatokra a hipotézisvizsgálatok során két területet fogunk bemutatni.

3.2.1. A próba erejének és szignifikanciájának vizsgálata

A próba ereje, illetve a szignifikancia minden esetben az eljárás érzékenységeként kezelhető.

- A szignifikancia és a korábban már tárgyalt konfidencia, (megbízhatóság) egymással lényegében megegyező fogalmak.

- A próba ereje egy bonyolultabb módon számolható paramétere a kiválasztott hipotézisvizsgálati eljárásnak. A próba ereje a vizsgálat úgynevezett másodfajú hibájával analóg fogalmak. A fenti példával élve, ha a nullhipotézisünk az, hogy a vizsgált páciensünk egészséges, úgy a másodfajú hibát akkor követjük el, amikor a betegeket nem részesítjük kezelésben.

3.7. Példa. A hibák kummulálódására az alábbi, általában ismert példát említjük. Több átlag összehasonlítását végezzük a varianciaanalízis során. Ekkor hagyományosan azt teszteljük, hogy több csoport átlaga egyezik-e egymással vagy sem. Világos, hogy a több átlag egyidejű, páronkénti összehasonlítása nem végezhető el független módon – és ilyen esetben az elsőfajú hibák valószínűségének viselkedéséről keveset tudunk.

A páros összehasonlítások úgynevezett „Post Hoc" tesztjeinek számos változata ismert, ezekből a teljesség igénye nélkül felsorolunk néhányat. A képletekben minden esetben szerepelni fog az MSE-érték, ami nem más, mint a csoportokon belüli átlagos négyzetes eltérés[17].

Továbbá általában feltételezzük, hogy ha k darab csoport van, akkor minden csoportban azonos, n esetszámmal dolgozunk (mutatunk egy olyan formulát is, ahol e feltételtől eltérhetünk).

Értelemszerűen két átlagot akkor nem fogunk szigkifikánsan különbözőnek tekinteni, ha a különbségük konfidenciaintervalluma tartalmazza a 0-t.

- Bonferroni-eljárás: Bonferroni azt javasolta, hogy ha α megbízhatósági szintű döntést szeretnénk hozni, de egymás után m darab tesztet kell végeznünk, me-

[17] Azaz a csoportok átlagaitól vesszük a csoportban lévő egyedek, részminták eltéréseinek négyzetösszegét és átlagoljuk – belső variancia, vagy hibavariancia néven is ismert. Szabadságfoka a fenti jelölésekkel $n-k$, azaz a teljes létszám és a csoportok számának különbsége.

lyek egymástól nem függetlenek, akkor α szint helyett $\frac{\alpha}{m}$ szinten döntsünk[18]. Fontos azonban megjegyeznünk, hogy ez általában feleslegesen szigorú eljárást jelent, így ezt általában finomítani szokás.

– Átlagok Bonferroni-összehasonlítása:

$$\overline{X}_{i,\bullet} - \overline{X}_{j,\bullet} \pm t_{(1-\frac{\alpha'}{2},\nu)}\sqrt{\frac{2MSE}{n}},$$

ahol $\frac{\alpha'}{2}$ tehát $\frac{\alpha}{2(m-1)}$, ahol m az összehasonlítások száma. $\overline{X}_{i,\bullet}$ az i-edik csoport átlagát jelöli, míg ν az MSE szabadságfoka.

– Átlagok Bonferroni-összehasonlításának Sidák-féle módosítása:

$$\overline{X}_{i,\bullet} - \overline{X}_{j,\bullet} \pm t_{(1-\frac{\alpha_m}{2},\nu)}\sqrt{\frac{2MSE}{n}},$$

ahol $\frac{\alpha_m}{2} = \frac{1-(1-\alpha)^{\frac{1}{m}}}{2}$.

– Átlagok Dunnett-féle összehasonlítása:

$$\overline{X}_{i,\bullet} - \overline{X}_{j,\bullet} \pm D_{(1-\alpha,k-1,\nu)}\sqrt{\frac{2MSE}{n}},$$

ahol D az úgynevezett Dunnett-eloszlás[19], k a csoportok száma, míg ν továbbra is MSE szabadságfoka.

– Átlagok Hsu-féle (MCB) összehasonlítása:

$$\overline{X}_{i,\bullet} - \max_{i\neq j} \overline{X}_{j,\bullet} \pm OD_{(1-\alpha,k-1,\nu)}\sqrt{\frac{2MSE}{n}},$$

ahol OD az egyoldali Dunnett-eloszlás.

– Átlagok Fisher-féle (LSD) összehasonlítása:

$$\overline{X}_{i,\bullet} - \overline{X}_{j,\bullet} \pm t_{(1-\frac{\alpha}{2},\nu)}\sqrt{MSE\left(\frac{1}{n_i}+\frac{1}{n_j}\right)},$$

mely eljárás tehát alkalmazható különböző csoportlétszámok esetén is.

[18] Ilyenkor tehát a korábban már tárgyalt elsőfajú hiba valószínűségét drasztikusan lecsökkentjük.

[19] A Dunnett-eloszlásról általában táblázat segítségével döntenek [50]. A standard normális eloszlás esetén is az eloszlásfüggvény inverzének táblázatát használják a statisztikai számításoknál, hiszen az inverznek zárt alakja nincsen – így ez a táblázatos eljárás nem nevezhető szokatlannak. A táblázathoz használt, a standard normális eloszlás eloszlásfüggvényénél lényegesen bonyolultabb formula megtalálható például Dunlap és munkatársai cikkében [14], mely cikkben ráadásul több példát is bemutatnak ezen eloszlás alkalmazására.

Megfigyelhető volt, hogy az átlagok egyenlőségének tesztelésekor a részmintáink szórásának egyenlősége is feltételként szabható – vannak eljárások, ahol ez nem feltétel. Azonban a korábban már említett Lee és munkatársai is megfogalmazzák 2010-ben publikált anyagukban [28], hogy a szórások összehasonlítása – helyesebben a részminták szóródási mutatóinak egyezése vagy különbözősége – egyáltalán nem triviális kérdés. Ráadásul több tesztet is összehasonlítanak egymással szimulációk segítségével, így a különböző tesztek numerikus eredményeit is áttekinthetjük dolgozatukban.

3.8. Példa. Az alábbiakban összefoglalunk néhány tesztet Lee és munkatársainak cikkéből. Mindezt azért tesszük, hogy jobban rávilágíthasunk: amennyiben a vizsgált változónk normalitása sérül, úgy a már korábban elmondottak alapján nem csak a középértékek megválasztása lehet problematikus (átlag helyett például medián, átlagok összehasonlítása helyett sztochasztikus egyenlőség vizsgálat, lásd Wilcox könyvét [46]), hanem a szóródási mutatók megválasztása, vagy azok tesztelése sem egyértelmű.

Két szórás összehasonlítására a hagyományos eljárás az úgynevezett F-próba (a két variancia hányadosa alapján tesztel), melynek feltétele a normalitás és melynek megsértésre kifejezetten érzékeny (lásd például Klotz és Johnson dolgozatát, [26] akik – ahogyan a most idézett dolgozat is – az először ismertetendő tesztet, mint alternatívát ajánlják helyette).

Az alábbi tesztek tehát mind a $H_0 : \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2$ nullhipotézis eldöntésére szolgálnak.

– **Levene-teszt**: A Levene-teszt próbastatisztikája:

$$W = \frac{(N-k)\sum_{i=1}^{k} n_i \left(\overline{Z}_i - \overline{Z}\right)^2}{(k-1)\sum_{i=1}^{k}\sum_{j=1}^{n_i} \left(\overline{Z}_{ij} - \overline{Z}_i\right)^2},$$

ahol N a teljes mintanagyság, n_i az i-edik részminta nagysága, $Z_{i,j} = \left|Y_{ij} - \overline{Y}_i\right|$, $\overline{Y}_i$ az i-edik részminta átlaga, $\overline{Z}_i$ a Z_{ij}-k csoportjainak egyenkénti átlaga, míg $\overline{Z}$ a Z_{ij}-k főátlaga, azaz a Levene-teszt az átlagos abszolút eltéréssel számol az átlagos négyzetes eltérés helyett.

A fenti W-próbastatisztika H_0 fennállása esetén F-eloszlást követ $k-1$ és $N-k$ szabadságfokkal.

– **Módosított Levene-teszt**: lényegében azonos a fenti teszttel, csak átlagok helyett mindenhol a mediánt kell használni.

– **Z-variancia teszt**: Az Overall és Woodward által 1974-ben publikált [35] eljárás a következő alakot ölti. A próbastatisztika:

$$F = \frac{\sum_{i=1}^{k} Z_i^2}{k-1},$$

$$Z_i = \sqrt{\frac{c_i\,(n_i-1)\,s_i^2}{MSE}} - \sqrt{c_i\,(n_i-1) - \frac{c_i}{2}},$$

ahol $c_i = 2 + \frac{1}{n_i}$, s_i^2 a korrigált tapasztalati variancia, n_i az adott részminta mintanagysága, MSE pedig a már korábban ismertetett négyzetes eltérés.

Ekkor H_0 fennállása esetén Z_i eloszlása standard normális, tehát a fenti F-próbastatisztika eloszlása F-eloszlás, $k-1$ és ∞ szabadságfokkal.

– **Az Overall–Woodward-féle módosított Z-variancia teszt:** 1976-ban a már hivatkozott Overall és Woodward szerzőpáros újabb dolgozatukban [36] módosították az eredeti c_i értékeket az alábbira:

$$c_i = 2\left(\frac{2,9 + \frac{0,2}{n_i}}{K}\right)^{\frac{1,6\left(n_i - 1,8K + 14,7\right)}{n_i}},$$

ahol n_i továbbra is az i-edik részcsoport mintanagysága, továbbá:

$$Z_{i,j} = \frac{X_{i,j} - \overline{X}_i}{\sqrt{\frac{n_i-1}{n_i}}\,s_i^2},$$

$$K = \frac{\sum_{i,j} Z_{i,j}^4}{n_i - 2}.$$

– **O'Brien-teszt**: Az O'Brien által publikált próba [34] azt mondja, hogy a hagyományos F-próbát módosítsuk olymódon, hogy az eredeti próbában használt $Y_{i,j}$ értékeket módosítjuk az alábbi módszerrel:

$$V_{ij} = \frac{(n_i - 1,5)\,n_i\left(Y_{ij} - \overline{Y}_2\right)^2 - 0,5 s_i^2\,(n_i - 1)}{(n_i - 1)\,(n_i - 2)},$$

ahol az alábbi jelöléseket alkalmaztuk:

$$\overline{Y}_i = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i},$$

$$s_i^2 = \frac{\sum_{j=1}^{n_i}\left(Y_{ij} - \overline{Y_i}\right)^2}{n_i - 1},$$

a megfelelő részcsoportátlagok és részcsoportvarianciák, tehát lényegében Y_{ij}-ket a fenti V_{ij} értékekre cseréljük, és úgy alkalmazzuk az eredeti F-próbát.

Megjegyzés. Megjegyezzük, hogy ilyenkor fennáll az alábbi egyenlőség:

$$s_i^2 = \overline{V}_i = \frac{\sum V_{i,j}}{n_i}.$$

Megállapíthatjuk, hogy amennyiben a normalitás nem teljesül, úgy a szóródási mutatóknál sem feltétlenül a szórást kell választani, hiszen látható, hogy a szórás nem feltétlenül a lehetséges legjobb, valamely középértéktől való átlagos eltérést mérő, jól interpretálható mennyiség.

3.3. Egy biostatisztikai megközelítés: ROC-görbék alkalmazása

A másik megközelítés a hipotézisvizsgálatok esetén a biostatisztika[20] egyik bevett eljárása. A továbbiakban a következő jelöléseket fogjuk alkalmazni:

Er	Érzékenység
Fa	Fajlagosság
$N_{+,v}$	Nem hibás, pozitív tesztek száma
$N_{+,h}$	Hibás pozitív tesztek száma
$N_{-,v}$	Nem hibás, negatív tesztek száma
$N_{-,h}$	Hibás negatív tesztek száma

Megjegyzés. Tehát $N_{+,v} + N_{-,h}$ a betegek, míg $N_{+,h} + N_{-,v}$ az egészséges egyedek száma. Érzékenységnek (sensitivity) nevezik annak valószínűségét, hogy egy beteget a teszt valóban betegnek mutat. Más megközelítésben:

$$Er = \frac{N_{+,v}}{N_{+,v} + N_{-,h}}.$$

Megjegyzés. Megjegyezzük, hogy egy másik, ezzel analóg fogalom is gyakran használatos a biostatisztikában. A fajlagosság (specificity) megmutatja, hogy mi a valószínűsége annak, hogy negatív tesztet kapunk abban az esetben, ha az illető tényleg egészséges. A fenti jelölésekkel:

$$Fa = \frac{N_{-,v}}{N_{-,v} + N_{+,h}}.$$

[20]Fontos megjegyezni, hogy az úgynevezett „túlélési statisztikák" e bevett grafikus elemzési eszközét számos területen – így nem csak a biostatisztikában – alkalmazzák. Így például a pénzügyi statisztikai eljárásokban is számos felhasználása ismert: egy betegségben való elhalálozás a cégek számára a csődeljárásként fogható fel. A modellek ilyen szempontból tehát rokonságban állnak egymással.

Az érzékenység és fajlagosság témájában is fontos mérnünk, hogy e két mennyiség milyen hibahatáron belül mozoghat. Ez lényegében nem más, mint annak mérése, hogy bizonyos statisztikai próbák első és másodfajú hibája miként alakul.

E biostatisztikai témakörben számos publikáció készült – melyek olykor pont e terület érzékenységvizsgálatát nem érintik. Erre hozható példaként Bender és munkatársainak elemzése [3] Brenner és Gefeller dolgozatáról [5], ahol a számításokat reprodukálva mutattak arra rá, hogy a becslésekben, melyeket a szerzők tettek, számos megkérdőjelezhető pont van.

Az orvoslásban persze adott egy igen egyszerű – bár nem feltétlenül költséghatékony ellenszere a téves diagnózisok szűrésének. Ez pedig nem más, mint amit Diepgen és Coenraads feszeget cikkükben [13]: több tesztet futtatnak egy-egy diagnózis felállítására. A több teszt futtatása, összefüggéseinek matematika sajátosságaira, statisztikai hibáinak kummulálódására vagy éppen szűkítésére hívják fel munkájukban a figyelmet egy igen konkrét diagnosztikai eljárás kapcsán.

Az orvosi alkalmazások során nyilván nem csak ilyen helyzetek adódnak. Egyes betegségek esetén a döntést és a becsléseket általában logisztikus regresszió[21] alkalmazásával és úgynevezett ROC-görbék elemzésével szokták megoldani.

3.1. Definíció. Tegyük fel, hogy adott k darab, $X_1, \ldots, X_k$ véletlen változó, melyek segítségével az Y bináris változó lehetséges értékeinek bekövetkezési valószínűségét szeretnénk meghatározni adott $x_1, \ldots, x_k$ realizáció esetén.

Világos, hogy $P(Y = 1)$ meghatározása elegendő, hiszen

$$P(Y = 1) + P(Y = 0) = 1.$$

A logisztikus regresszió modellje azt mondja, hogy

$$P\left(Y = 1 | X_1, \ldots, X_k\right) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k}}$$

alakban keresendő.

Innen is világosan látszik, hogy a logisztikus regresszió egyfajta lehetséges modellje a bekövetkezési valószínűség meghatározásának, adott realizáció mellett. A lábjegyzetben is olvasható, Boros Endre és munkatársai által jegyzett [8] cikk éppen e helyzetek másfajta megközelítésére ajánl alternatívát egy, a logiszitkus regresszió modelljétől teljesen más megközelítés alkalmazásával.

[21] Jegyezzük meg, hogy nem csak logisztikus regressziót lehetne alkalmazni egy-egy ilyen osztályozási eljárás során. Például Boros és munkatársainál könyvet is olvashatunk [7] a Logical Analysis of Data (LAD) eljárásról, mely szintén egy bináris osztályozás, ahol azonban nem statisztikai, hanem optimalizálási technikák segítségével dolgoznak. Konkrét implementációit is adják Boros és szerzőtársai dolgozatukban [8], ahol pszichometriai, műszaki és gazdasági adatokon egyaránt bemutatják eljárásukat, numerikus eredményekkel alátámasztva. Érdemes tehát arról is tudnunk, hogy a logisztikus regresszió nem feltétlenül az egyetlen olyan eljárás, melynek segítségével bináris változók eloszlásáról szerezhetünk információt – sőt.

Megjegyzés. A ROC-görbékkel az egységnégyzetben ábrázolják a érzékenység (sensitivity) és fajlagosság (specificity) közötti összefüggéseket. Míg az x tengelyen az $1 - Fa$, addig az y tengelyen az Er érték (arány) helyezkedik el.

Bár számos helyen fellelhető e módszer (lásd például [43]), egy egyszerű példán keresztül könnyen bemutatható mind az alkalmazás, mind pedig a görbe elkészítésnek metódusa. A ROC-görbéhez e feladat Buza Krisztián jegyzete [48] alapján készült.

3.9. Példa. Tegyük fel, hogy lázat szeretnénk mérni, láz alapján pedig valamely betegséget diagnosztizálni, mely betegség általában lázzal jár – de persze nem minden esetben, illetve nem minden lázas szenved ebben a betegségben.

A mintánkat már testhőmérséklet szerint sorrendbe rendeztük a jobb átláthatóság kedvéért.

V	-	-	-	-	-	-	-	-	+	-	+	+
M	36,4	36,4	36,5	36,6	36,6	36,6	36,7	36,8	37,5	37,6	39	39,2

Azaz: a valóságban (V) a „-” jel azt mondja, hogy egészséges, nem szenved e specifikus betegségben, míg a „+” azt mondja, hogy beteg. A modellben (M) pedig a testhőmérsékletekkel modellezünk, tehát azzal szeretnénk mérni, diagnosztizálni.

A ROC-görbéhez ki kell számolnunk az igazi pozitív ($N_{+,v}$), a hamis pozitív ($N_{+,h}$), igazi negatív ($N_{-,v}$) és hamis negatív ($N_{-,h}$) értékeket. Szükségünk lesz az igazi pozitívok (Er) és a fals pozitívok ($1 - Fa$) arányára a betegek és az egészségesek között a testhőmérséklet különböző, értelmes értékei esetén, hiszen az y és x tengelyek rendre ezeket az arányokat mutatják.

A testhőmérséklet különböző szintjein kell hát eldönteni, hogy hány helyes és hány helytelen diagnózis lenne a fent adott modellel a betegséget illetően (tehát a táblázat első 4 sorában az adott módon besorolt betegek számát jelöljük, az alsó két sorban pedig az arányokat). A sorok elején a már korábban definiált jelöléseket használjuk.

A táblázat első sorában az értelmes testhőmérséklet vágópontokat tüntettük fel. Amely értékből több is volt, azt zárójelben szerepeltetjük.

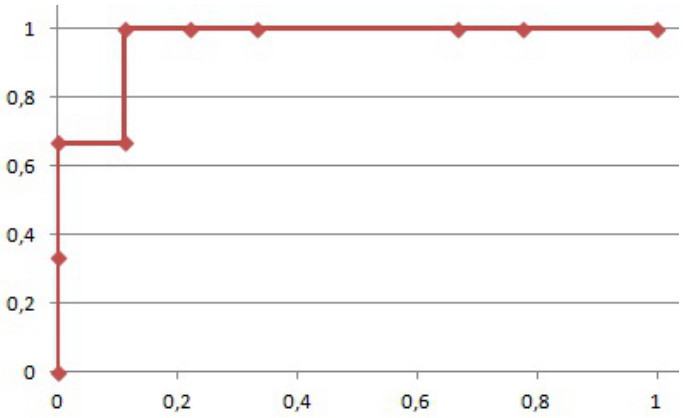
Hőmérséklet	36,4 (2)	36,5	36,6 (3)	36,7	36,8	37,5	37,6	39	39,2	FIN
$N_{+,v}$	3	3	3	3	3	3	2	2	1	0
$N_{+,h}$	9	7	6	3	2	1	1	0	0	0
$N_{-,v}$	0	2	3	6	7	8	8	9	9	9
$N_{-,h}$	0	0	0	0	0	0	1	1	2	3
Er	1	1	1	1	1	1	2/3	2/3	1/3	0
$1 - Fa$	1	7/9	6/9	3/9	2/9	1/9	1/9	0	0	0

A táblázat kitöltésének módjára vegyünk egy konkrét cellát. Az $N_{+,h}$ sorban tehát azt vizsgáljuk, hogy a testhőmérséklet adott értékének vágópontként való

definiálásával hány darab fals, pozitív eredményt kapnánk. Így például ha $36,5C^\circ$-os testhőmérésékletet vágópontként kezelve, a $36,4C^\circ$-os pácienset nem tekintenénk betegnek, azonban továbbra is maradna 7 darab fals, valóságban egészséges páciensünk, akiket betegnek jeleztünk (és lenne természetesen 3 darab, valóságban beteg páciensünk helyesen azonosítva).

Általánosságban: ha a görbe átmegy az egységnégyzet bal felső sarkán, akkor téves diagnózis nélküli eljárást sikerült alkotni. Minden görbe esetén fontos tehát annak alakja, hiszen minél jobban közelíti a görbe a bal felső sarkot, annál precízebb, pontosabb diagnózist lehet az eljárással felállítani. Azonban a görbe alakján kívül a görbe alatti területnek is jelentése van: lényegében a tesztünk hatékonyságának mérőszáma (a bal első sarkon átmenő esetben a terület 1, tehát ilyenkor a leghatékonyabb, míg egy olyan görbe esetén, ami a négyzet bal alsó sarkát a jobb felső sarokkal összekötő átlóját mutatja lényegében pénzt is dobálhatnánk döntéshozatal helyett).

A példánkhoz tartozó ábrát az utolsó két sor alapján elkészítettük, tehát az alsó két sorban található értékek a görbe koordinátái:



1. ábra. ROC-görbe az igazi pozitív és fals pozitív arányok szerint

Az ábra elég jól közelíti a bal felső sarkot, tehát azt mondhatjuk, hogy a fenti példában egy kellően jól viselkedő modellt tudtunk alkotni: a görbe alatti terület $\frac{26}{27}$, tehát a helyes diagnózisok valószínűsége magasnak mondható.

A döntéshozatalra, illetve alkalmazásukra számos példa hozható fel – pusztán a módszert és annak értelmezését láthatjuk Goldstein és munkatársai, 1906 öngyilkosságot túlélteken elvégzett pszichiátria kutatásában, illetve annak dokumentációjában [19].

Egy elméleti, a ROC-görbék elemzésében alkalmazott mennyiségek χ^2 statisztikák segítségével vizsgáló cikk olvasható Bennettől [4], aki teljesen elméleti megkö-

zelítésben tárgyalja – majd saját vizsgálati eredményein teszteli is a diagnosztikai eljárások ilyedtén való becslését, illetve becslésének jóságát.

3.4. Megjegyzések a hipotézisvizsgálatokhoz

Nem érintettük itt a hipotézisvizsgálatok során az összes létező lehetőséget a próbák lehetséges hibáinak tesztelésére. Világos, hogy minden statisztikai próba csak bizonyos – szigorúbb vagy kevésbé szigorú – feltételek mellett viselkedik optimálisan. E feltételek sérülése esetében különböző robusztus eljárások választhatók – azonban e választások során sem elhanyagolható, hogy a „hagyományos" eljárás feltételei, mely eljárás helyett most e robusztusat választottuk, milyen mértékben sérülnek.

A sérülés mértékének, minőségének következményeire ritkán találhatunk egzakt módon is igazolható, megbízható és kalkulálható eljárásokat – azaz, amit például a student-féle t-próba esetén jól körüljárható területnek gondolunk.

A t-próba esetén a ferdeség, csúcsosság – vagy általánosabban a normalitás hiánya esetén választható robusztus tesztek megbízhatóságára Vargha 2003-as cikke [45], vagy a próba erejének vizsgálatára a normalitás sérülése esetén Srivastava 1958-as dolgozata [42] lehet példa. Ez más hipotézisvizsgálati módszerek esetén messze nem tűnik kérdések nélküli területnek, illetve elméleti háttere – a fellelhető szakirodalmak alapján – nem látszik ennyire körüljártnak.

4. Összefoglalás

E témában több összefoglaló mű is született, melyek támpontot, kiindulási alapot adhatnak a különböző statisztikai tesztek, illetve azok robusztus változatainak megismeréséhez (példaként említhetjük összefoglaló anyagként Wilcox könyvét [46], melyből számos hagyományos módszert, és azok több robusztus változatát is megismerhetjük).

Megállapítható, hogy a statisztikai vizsgálatok jelentős hányada a bemeneti adatok változásait vagy változékonyságát – amiatt, hogy eleve valószínűségi változókkal dolgozik, melyek szükségszerűen változékonyak kisebb-nagyobb mértékben – kezelik valamilyen formában. A leggyakrabban ez olymódon jelenik meg, hogy az eljárások megfelelő biztonsági szinten való alkalmazását feltételekhez kötik (a véletlen változó eloszlásának pl. normális volta, csoportok szórásának homogenitása, stb). Amennyiben e feltételek sérülnek, úgy az adott eljárás valamely korrekciós – robusztus – változatát javasolják. Ezen esetben az eljárásokban mindenképpen jelen lévő hibákat (hiszen véletlen jelenségek alapján hozunk döntéseket) általában megfelelő szinten lehet tartani.

Más esetekben viszont nem ismertek azok a matematikai alapok és vizsgálatok, melyek biztosítanák az eljárást alkalmazók számára azokat a stabilitási kritériumokat, melyekkel a hibás döntések valószínűsége meghatározható, uralható. Így pl.

empirikus eszközök segítségével – szimulációs eljárások – az adott tapasztalati eloszlások vizsgálatával kimérhetők az alkalmazott eljárások hibái. Ha a hibákat e módszerekkel nem is tudjuk kiküszöbölni, azok mértékével tisztában lehetünk – és így továbbra is megalapozott döntések hozhatók.

Szintén empirikusak, de nem feltétlenül igényelnek nagyobb gépigényt – illetve a kezdeti tapasztalatok alapján kis minták esetén is működőképes alternatívát jelenthetnek – a simítási eljárások. Segítségükkel robusztus becslések készíthetők – stabilabbá, kevésbé érzékennyé téve így az eljárásunkat, illetve a segítségükkel meghozott döntéseinket.

Természetesen – ahogy jeleztük, nem törekedtünk cikkünkben a statisztika minden területének lefedésére. Nem beszéltünk például a különböző regressziós technikák megbízhatóságáról, a Bayes-becslések érzékenységéről vagy az idősorok esetén alkalmazható különböző technikákról és felmerülő problémákról. Célunk pusztán az volt, hogy két, egyszerűbb területet kiragadva, azok segítségével vázoljuk a probléma általános mivoltát, nagyságát és fontosságát.

Hivatkozások

[1] Bayne, W.; Tobet, S.; Mattiace, L. A.; Lasco, M. S.; Kemether, E.; Edgar, M. A.; Morgello, S.; Buchsbaum, M. S.; Jones, L. B.: *The interestitial Nuclei of the Human Anterior Hypothalamus: An Inverstigation of Variation with Sex, Sexual Orientation, and HIV Status*, Hormones and Behavior, Vol. **40/2**, pp.: 86–92, 2001.

[2] Belia, S.; Fidler, F.; Williams, J.; Cumming, G.: *Researchers Misunderstand Confidence Intervals and Standard Error Bars*, Psychological Methods, Vol.: **10/4**, pp.: 389–396, 2005.

[3] Bender, R.; Langue, S.; Freitag, G.; Trampisch, H. J.: *Letters to the Editor on „Variation of sensitivity, specificity, likelihood ratios and predictive values with disease prevalence*, Statistics in Medicine, Vol. **16**, pp.: 981–991, 1997.", Statistics in Medicine, Vol. **17**, pp.: 945–950, 1998.

[4] Bennett, B. M.: *On comparisons of sensitivity, specificity and predictive value of a number of diagnostic procedures*, Biometrics, Vol. **28**, pp.: 793–800, 1972.

[5] Brenner, H.; Gefeller, O.: *Variation of sensitivity, specificity, likelihood ratios and predictive values with disease prevalence*, Statistics in Medicine, Vol. **16**, pp.: 981–991, 1997.

[6] Bolla, M.; Krámli, A.: *Statisztikai következtetések elmélete*, Typotex, 2005.

[7] Boros, E.; Hammer, P. L.; Ibaraki, T.: *Logical Analysis of Data*, IGI-Global, 2005.

[8] Boros, E.; Hammer, P. L.; Ibaraki, T.; Kogan, A.; Mayoraz, E.; Muchnik, I.: *An implementation of logical analysis of data*, Knowledge and Data Engineering, Vol. **12/2**, pp.: 292–306, 2000.

[9] Borovkov, A. A.: *Matematikai Statisztika*, Typotex, 1999.

[10] Camponovo, L.; Otsu, T.: *Breakdown pont theory for implied probability bootstrap*, The Econometrics Journal, Vol. **15/1**, pp.: 32–55, 2012.

[11] Cochran, W. G.: *The distribution of quadratic forms in a normal system, with applications to the analysis of covariance*, Mathematical Proceedings of the Cambridge Philisophical Society, Vol. **30/2**, pp.: 178–191, 1934.

[12] Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*, New York, 1988.

[13] Diepgen, T. L.; Coenraads, P. J.: *Sensitivity, specificity and positive predictive value of patch testing: the more you test, the more you get?*, Contact Dermatitis, Vol. **42/6**, pp.: 315–317, 2000.

[14] Dunlap, W. P.; Marx, M. S.; Agamy, G,J.: *Fortain IV functions for calculating probabilities associated with Dunnett's test*, Behavior Research Methods and Instrumentation, Vol. **13/3**, pp.: 363–366, 1981.

[15] Efron, B.; Tibshirani, R.: *Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy*, Statistical Science, Vol. **1/1**, pp.: 54–75, 1986.

[16] Fletcher, D.; Webster, R.: *Skewness-Adjusted confidence Intervals on Stratified Biological Surveys*, Journal of Agricultural, Biological and Environment Statistics, Vol. **1/1**, pp.: 120–130, 1996.

[17] Gayen, A. K.: *The distribution of „Student's" t in random samples of any size drawn from non-normal universes*, Biometrika, Vol. **36**, pp.: 353–369, 1949.

[18] Geyer, C. J.: *Breakdown Point Theory Notes*, http://www.stat.umn.edu/geyer/5601/notes/break.pdf, (letöltés: 2012. 10. 16.)

[19] Goldstein, R. B.; Black, D. W.; Nasrallah, A.; Winkour, G.: *The Prediction of Suicide, Archives of General Psychiatry*, Vol. **48/5**, pp.: 418–422, 1991.

[20] Hall, P.: *On the removal of skewness by tranformation*, Journal of the Royal Statistics Society, Vol. **54**, pp.: 221–228, 1992.

[21] Hampel, F.; Hennig, C.; Ronchetti, E.: *A smoothing principle for the Huber and other location M-estimators*, Computational Statistics and Data Analysis, Vol. **55**, pp.: 324–337, 2011.

[22] Huber, P. J.: *Robust Estimation of a Location Parameter*, Annals of Matematical Statistics, Vol. **35/1**, pp.: 73–101, 1964.

[23] Johnson, N. J.: *Modified t tests and confidence intervals for asymmetrical distributions*, Journal of the American Statistical Association, Vol. **73/363**, pp.: 536–544, 1978.

[24] Jones, D. N.; Gill, C. A.: *Comparing Measures of Sample Skewness and Kurtosis*, The Statistician, Vol. **47/1**, pp.: 183–189, 1998.

[25] Judkins, D. R.: *Fay's method for variance estimation*, Journal of Official Statistics, Vol. **6**, pp.: 223–239, 1990.

[26] Klotz, S.; Johnson, N. L.: *Breakthroughs in Statistics*, Foundations and Basic Theory, Vol. **1**, pp.:680, 1993.

[27] Lavine, M.: *Sensitivity in Bayesian Statistics: The Prior and the Likelihood*, Journal of the American Statistics Association, Vol. **86/414**, pp.: 396–399, 1991.

[28] Lee, H. B.; Katz, G. S.; Restori, A. F.: *A Monte Carlo Study of Seven Homogeneity of Variance Tests*, Journal of Mathematics and Statistics, Vol. **6/3**, pp.: 359–366, 2010.

[29] Lehmann, E. L.: *Theory of Point Estimation*, John Wiley and Sons, New York, 1983.

[30] Lehmann, E. L.: *Testing Statistical Hypotheses*, John Wiley and Sons, New York, 1959.

[31] LeVay, S.: *A difference in Hypothalamic Structure between Heterosexual and Homosexual Man*, Science, New Series, Vol. **253**, No. **5023**, pp.: 1034–1037, 1991.

[32] Mameli, V.; Music, M.; Sauleau, E.; Biggeri, A.: *Large sample confidence intervals for the skewness parameter of the skew-normal distribution based on Fisher's information*, Journal of Applied Statistics, Vol. **39/8**, pp.: 1693–1702, 2012.

[33] Mogyoródi J.; Michaletzky Gy.: *Matematikai statisztika*, ELTE, TTK, Nemzeti Tankönyvkiadó, Budapest, 1995.

[34] O'Brien, R. G.: *Robust tschniques for testing heterogeneity of variance effects in factorial designs*, Psychometrika, Vol. **43**, pp.: 327–342, 1978.

[35] Overall, J. E.; Woodward, J. A.: *A simple test for homogeneity of variance in complex factorial design*, Psychometrika, Vol. **39**, pp.: 311–318, 1974.

[36] Overall, J. E.; Woodward, J. A.: *A robust and powerfull test for heterogeneity of variance*, University of Texas, Medical Branch Psychometric Laboratory, 1976.

[37] Prékopa, A.: *Valószínűségelmélet műszaki alkalmazásokkal*, Műszaki Könyvkiadó, Budapest, 1962.

[38] Rényi A.: *Valószínűségszámítás*, Tankönyvkiadó, Budapest, 1968.

[39] Robins, J. M.; Rotniczky, A.; Scharfstein, D. O.: *Sensitivity analysis for selection bias and unmeasured confounding in missing data and causal inference models*, SZERK: Holloran, M. E.; Berry, D.: *Statistical Models in Epidemiology, The Environment, and Clinical Trials*, Vol. **116**, Springer, 2000.

[40] Saavedra, P. J.: *An extension of Fay's method for Variance Estimation to the Bootstrap*, Proceeding to the Annual Meeting of the American Statistical Association, August 5–9, 2001.

[41] Sak, H.; Hörman, W.; Leydold, J.: *Better Confidence Intervals for Importance Sampling*, Research Report Series, Rep. 106, Institute for Statistics And Mathematics, Wirtschafts Universitat Wien (Vienna University of Economics and Business), 2010.

[42] Srivastava, A. B. L.: *Effect of non-normality on the power function of t-test*, Biometrika, Vol. **45/3/4**, pp.:421–430, 1958.

[43] Takahashi, K.; Uchiyama, H.; Yanagisawa, S.; Kamae, I.: *The Logistic Regression and ROC Analysis of Group-based Screening for Predicting Diabetes Incidence in Four Years*, Kobe J. Med. Sci., Vol. **52** (6), pp.: 171–180, 2006.

[44] Takács Sz.: *Egy nem hagyományos statisztikai eljárás bemutatása az OECD PISA adatbázison – esettanulmány*, Alkalmazott Matematikai Lapok, Vol. **27**., 157–174, 2010.

[45] Vargha, A.: *Robusztussági vizsgálatok az egymintás t-próbával*, Statisztikai Szemle, Vol. **81/10**, pp.: 872–890, 2003.

[46] Wilxoc, R. R.: *Applying Contemporary Statistical Techniques*, Academic Press, 2003.

[47] Wright, D. B.; Herrington, J. A.: *Problematic standard errors and confidence intervals for skewness and kurtosis*, Behavior Research Methods, Vol. **43/1**, pp.: 8–17, 2011.

[48] http://cs.bme.hu/ buza/edu/dm techn/dm feladatok.pdf (letöltve: 2012. 11. 16.).

[49] http://www.nsf.gov/statistics/nsf03302/pdf/setables.pdf (letöltés ideje: 2012. 10. 02.).

[50] http://www.watpon.com/table/dunnetttest.pdf (letöltés ideje: 2012. 11. 16.).

(Beérkezett: 2012. november 30.)

TAKÁCS SZABOLCS
Károli Gáspár Református Egyetem
Bölcsésztudományi Kar, Pszichológiai Intézet, Általános lélektani és módszertani tanszék
1037, Budapest, Bécsi út 324, 5. épület, fszt.
e-mail: tretarkhon@gmail.com

SENSITIVITY ANALYSIS IN A STATISTICAL PROCESSES

Szabolcs Takács

An important aspect of many mathematical process is sensitivity analysis. In these analysis we investigate the change of output data – result and behavior – when changes are made to the input. It is of interest what type of changes in the input doesn't affect the results – or which type of modifications in the inputs results in larger or smaller scale changes to the output.

In the various fields of statistical processes, sensitivity has different a meaning. As an example, it has different meaning in estimation or in hypothesis theory – or in the different modelling processes.

In this paper we are not aiming to address all the various questions about sensitivity in the fields of statistics – instead we embark on providing an insight to the wide spectrum of the applications involved with sensitivity analysis, while also drawing attention to the importance of these analysis.

The paper will not state new theorems – but rather it raises several open questions of interest which have arisen in recent statistical research projects.