Önálló Labor

Random konstrukciók programozása a Zarankiewicz problémához

Borsik Balázs

Témavezető: Dr. Héger Tamás

2025. május 25.

1. Alapfogalmak

- $\boldsymbol{K_{n,m}}$: Egy (N, M, E) teljes páros gráf, ahol $|N| = n$, $|M| = m$ és bármely $u \in N$ és $v \in M$ között van él behúzva.
- $\boldsymbol{C_4}$: $K_{2,2}$
- **Páros gráf mátrixos ábrázolása**: A dokumentumban a *gráf* kifejezés mindig páros gráfokra utal, még abban az esetben is, ha ez nincs külön kihangsúlyozva. Egy (N, M, E) páros gráfhoz gyakran hozzárendeljük a hozzá tartozó szomszédsági mátrixot, melynek mérete $|N| \times |M|$. Ebben a megfeleltetésben a mátrix n-edik sora egy N-beli csúcsot, az m-edik oszlopa pedig egy M-beli csúcsot reprezentál. A két csúcs között akkor van él, ha a szomszédsági mátrix (n, m) pozíciójában 1 található; ha pedig 0, akkor nincs köztük él.
- **Zarankiewicz-szám:** Egy $G = (A, B; E)$ páros gráf $K_{s,t}$-mentes, ha nem tartalmaz olyan s elemű csúcshalmazt A-ban és t elemű csúcshalmazt B-ben, amelyek egy $K_{s,t}$-vel izomorf részgráfot alkotnak. Az (m, n) méretű $K_{s,t}$-mentes páros gráf éleinek maximális számát $Z_{s,t}(m, n)$-nel jelöljük, és Zarankiewicz-számnak nevezzük. A dokumentumban amennyiben s, t nincs jelölve, akkor $Z(m, n)$ ekvivalens $Z_{2,2}(m, n)$-nel. [3]
- **Zarankiewicz-probléma (eredeti megfogalmazás):** Vegyünk egy $n \times m$-es mátrixot, amely csak 0 és 1 értékeket tartalmaz. Legkevesebb hány darab 1-es szükséges, hogy mindenképp tartalmazzon $s \times t$-es csupa 1-es részmátrixot? Ez a definíció megegyezik $Z_{s,t}(m, n) + 1$ -el (ez a fent bemutatott mátrixos áttérés után rögtön látszik).
- Egy C_4 mentes gráf **triviálisan kiegészíthető**, ha létezik olyan még nem behúzott él, melynek behúzása után nem keletkezik C_4.

2. Bemutatás

Mint látható, a Zarankiewicz probléma egy extremális gráfelméleti probléma. A problémára több képlet is ad értelmezhető felső becsléseket, de ezek a becslések általában a paraméterek nagyságának vagy aszimmetriájának növelésével jelentősen eltérhetnek a pontos értékektől.

Alsó becslésekre kevesebb képlet található; ezekben az esetekben különféle konstrukciók (pl. projektív síkok, vagy kimerítő keresések) adhatnak jobb közelítő értéket.

Az önálló labor célja a random konstrukciókkal kapott alsó becslések vizsgálata és programozása volt. A kipróbált módszerek többsége már ismert módszerek különböző heurisztikák alapján történő módosításainak eredménye.

3. Matematikai háttér

- Egy **incidenciastruktúra** egy $(\mathcal{P}, \mathcal{L}, I)$ hármas, ahol:
  - $\mathcal{P}$ egy halmaz, elemei a *pontok*,
  - L egy P-től különböző halmaz, elemei az *egyenesek* (vagy blokkok),
  - $I \subseteq P \times L$ az *incidenciareláció*, vagyis azon pont–egyenes párok halmaza, melyek „kapcsolatban állnak egymással”.

  Minden incidenciastruktúra megfeleltethető egy $G = (A, B, E)$ páros gráfnak, ahol $A = P$, $B = L$, és $a \in A$, $l \in L$ között akkor van él, ha $(a, l) \in I$.

- A **projektív sík** egy incidenciastruktúra $(\mathcal{P}, \mathcal{L}, I)$, ahol:
  - $\mathcal{P}$ a *pontok* halmaza,
  - $\mathcal{L}$ az *egyenesek* (vagy blokkok) halmaza,
  - $I \subseteq \mathcal{P} \times \mathcal{L}$ az *incidenciareláció*,

  amely az alábbi tulajdonságokkal rendelkezik:
  - Bármely két különböző pontra pontosan egy olyan egyenes létezik, amely mindkettővel incidens.
  - Bármely két különböző egyenesre pontosan egy olyan pont létezik, amely mindkettővel incidens.
  - Létezik négy pont, amelyek közül semelyik hármat nem tartalmazza ugyanaz az egyenes (vagyis nincs olyan egyenes, amely háromnál több közülük levő ponttal incidens lenne).

  A definícióból következik, hogy bármely egyeneshez ugyanannyi pont tartozik, mint ahány egyenes incidens egy adott ponttal. Ezt a közös számot $q + 1$-nek jelöljük, ahol q a sík *rendje*.

- $\boldsymbol{t - (v, K, \lambda)}$ **design:** Legyen $\emptyset \neq K \subset \mathbb{Z}^+$. Egy $(P, \mathcal{B})$ incidenciastruktúrát $t - (v, K, \lambda)$ *designnak* nevezünk, ha:
  - $|P| = v$,
  - minden $B \in \mathcal{B}$ esetén $|B| \in K$,
  - minden t különböző pont pontosan λ közös blokkban szerepel.

  Ha $K = \{k\}$, akkor egyszerűen $t - (v, k, \lambda)$ designról beszélünk. Ilyen design esetén:

$$b = |B| = \lambda \binom{v}{t} \Big/ \binom{k}{t}, \quad r = \frac{bk}{v} = \lambda \binom{v-1}{t-1} \Big/ \binom{k-1}{t-1}$$

ahol b a blokkok száma, r pedig az egy pontra eső blokkok száma. Feltételezzük, hogy $k < v$ és $\lambda \geq 1$.

A t–(v, k, λ) design incidenciagrafja $K_{t,\lambda+1}$-mentes, és ezek tartalmazzák a legtöbb élt az ilyen típusú gráfok közül.

A (t, v, k, λ) paramétereket akkor nevezzük *megengedettnek* (vagy admisszibilisnek), ha pozitív egészek, teljesül:

$$2 \leq t \leq k < v, \quad b = \lambda \binom{v}{t} \Big/ \binom{k}{t} \in \mathbb{Z}, \quad r = \lambda \binom{v-1}{t-1} \Big/ \binom{k-1}{t-1} \in \mathbb{Z}$$

Például egy q rendű projektív sík egy $2 - (q^2 + q + 1, q + 1, 1)$ design.

Forrás: *Damásdi, G., Héger, T. and Szőnyi, T., 2013. The Zarankiewicz problem, cages and geometries.* [2]

4. Felső becslések bemutatása

A felső becslésekhez alkalmazható például a **Jensen-egyenlőtlenség**, amely az alábbi módon fogalmazható meg:

$$\phi\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) \leq \frac{1}{n}\sum_{i=1}^{n} \phi(x_i),$$

ha ϕ konvex függvény. Ez az egyenlőtlenség segít az egyes Zarankiewicz-számokra adott becslések levezetésében.

A projekt során is használt felső becslés pedig **Roman becslése** volt, ami a Jensen-egyenlőtlenségből kapott paraméterezett változat, így sokkal kényelmesebben használható. A program során ez a becslés adta az élvalószínűségek alapját.

Roman-egyenlőtlenség: Legyen $I \subset \mathbb{R}$ egy intervallum, $f : I \to \mathbb{R}$ egy szigorúan növekvő konvex vagy szigorúan csökkenő konkáv függvény, $n \in \mathbb{N}$, $x_1, \ldots, x_n, p, p+1 \in I \cap \mathbb{Z}$. Ekkor

$$\sum_{i=1}^{n} x_i \leq \frac{\sum_{i=1}^{n} f(x_i)}{f(p+1) - f(p)} + n \cdot \frac{pf(p+1) - (p+1)f(p)}{f(p+1) - f(p)}.$$

Az egyenlőség pontosan akkor áll fenn, ha $x_i \in \{p, p+1\}$ minden $1 \leq i \leq n$ esetén, vagy ha $\{x_1, \ldots, x_n, p, p+1\} \subset I'$ egy olyan intervallumban, amelyen f lineáris.

Bizonyítás: Legyen $a, c \in \mathbb{R}$, ahol $a > 0$, továbbá legyen $F(x) = af(x) - x + c$

Válasszuk meg a és c értékét úgy, hogy teljesüljön:

$$F(p) = af(p) - p + c = 0, \quad F(p+1) = af(p+1) - (p+1) + c = 0.$$

Ebből következik:

$$a = \frac{1}{f(p+1) - f(p)}, \quad c = \frac{pf(p+1) - (p+1)f(p)}{f(p+1) - f(p)}.$$

Mivel f növő és konvex, ezért $a > 0$, így $F(x)$ is konvex.

Két eset lehetséges:

- Vagy $F(x) = 0$ egy $[p, p+1]$ intervallumot tartalmazó intervallumon, vagy
- $F(x) = 0$ csak a p és $p+1$ egész számokra teljesül.

Mindkét esetben $F(x) \geq 0$ minden egész x-re.

Ezért:

$$0 \leq \sum_{i=1}^{n} F(x_i) = \sum_{i=1}^{n} (af(x_i) - x_i + c) = a \sum_{i=1}^{n} f(x_i) - \sum_{i=1}^{n} x_i + nc,$$

amiből:

$$\sum_{i=1}^{n} x_i \leq a \sum_{i=1}^{n} f(x_i) + nc.$$

Az a és c behelyettesítésével megkapjuk az állított egyenlőtlenséget. Az egyenlőség pontosan akkor teljesül, ha minden i-re $F(x_i) = 0$, vagyis az ismertetett két eset valamelyike áll fenn.

Roman-féle felső korlát: Legyen $G = (A, B; E)$ egy $K_{s,t}$-mentes páros gráf, ahol $|A| = m$, $|B| = n$, és $p \geq s-1$. Ekkor G éleinek számára teljesül:

$$e(G) \leq \frac{t-1}{\binom{p}{s-1}} \binom{m}{s} + n \cdot \frac{(p+1)(s-1)}{s}.$$

Források: *STEVEN ROMAN, 1974. A Problem of Zarankiewicz* [4]
Damásdi, G., Héger, T. and Szőnyi, T., 2013. The Zarankiewicz problem, cages and geometries. [2]

Programváltozatok

Program alapja: A program minden változata `C++` nyelven készült. A gráfokat szomszédsági mátrixban tároljuk, mivel a vizsgált intervallumban a használt algoritmusokkal ez a módszer gyorsabbnak bizonyult az éllistás tárolásnál.

Eredmények táblázat jelölései: Az eredménytáblázatban néhány különböző méretű, $n \times m$ méretű $Z(n, m)$ probléma szerepel a módszer bemutatására. Emellett két mérőszám is szerepel az értékeléshez, melyek kizárólag a $Z(10, 10)$ és $Z(40, 40)$ közötti méretekből számolódnak, hogy elkerüljük a kisebb, könnyebben megoldható esetek torzító hatását.
Referencia oszlop: A referencia oszlopot mindig az általunk talált legjobb táblázat adja, és a többi mérőszám is ezt veszi alapul. Ez első sorban azért van, mivel nem található jelenleg megfelelő szakirodalom a vizsgált intervallumban. Az általunk használt legfrissebb szakirodalom *Tan* [6] cikke, ami csak a $Z(24, 35)$-ig ad értékeket, de ezeknek nem mindegyike pontos felső becslés. Azokban az esetekben, ahol ismert az alsó becslés az általunk kapott táblázat is öt esettől eltekintve mindig pontos értéket ad.

Az **Él %** azt mutatja meg, hogy az adott módszer az intervallumon belül mennyire közelíti meg a legjobb általunk ismert eredményeket. Ehhez az adott módszer által talált legjobb gráfok élszámait összeadjuk, majd elosztjuk az intervallumban ismert legjobb eredmények élszámainak összegével.

A **Diff** azt mutatja, hogy a módszer hányszor adott pontos, egy híján, vagy két híján pontos megoldást az alábbi formátumban: `<pontos>, <pontos-1>, <pontos-2>`. A $Z(10, 10)$ és $Z(40, 40)$ közötti intervallumban összesen 496 vizsgált eset szerepel.

1. Mohó algoritmus

Algoritmus: Véletlen sorrendben végig megyünk az összes élen, ha egy él behúzása C_4-et hozna létre, akkor nem húzzuk be, különben behúzzuk. (Így a kapott gráf biztos, hogy triviálisan nem kiegészíthető, hiszen, ha az lenne, akkor a behúzható élt már behúztuk volna, amikor az algoritmus során vizsgáljuk.)

Motiváció: Az éleket csak sorban vizsgálva a gráf nagyon aszimmetrikus lenne, például az elsőnek vizsgált csúcs össze lenne kötve az egész másik csúcsosztállyal, és mivel a felső becslések is minél egyenletesebb fokszámeloszlást vesznek alapul, a projektív síkok, és abból képzett konstrukciók is ilyenek, ezért az élek random sorrendje lehetővé teszi egy kiegyensúlyozottabb gráf előállítását, miközben a gráf továbbra is triviálisan nem kiegészíthető marad.

Eredmények:

Z(n, m)	**Mohó algoritmus**	**Referencia**
$Z(9,30)$	63	63
$Z(14,29)$	85	87
$Z(20,20)$	88	96
$Z(20,37)$	130	136
$Z(30,30)$	150	175
$Z(40,40)$	222	252
Él %	92.87%	100%
Diff	69, 46, 33	496, 0, 0
Futásidő	1843s	-
Iteráció	18000	-

1. táblázat. Mohó algoritmus eredményei

2. Moser-Tardos

Moser Tardos módszer:

Definíció Legyenek $E_1, E_2, \dots, E_n$ események egy valószínűségi térből. A G gráfot *függetlenségi gráfnak* nevezzük, ha csúcshalmaza $\{1, 2, \dots, n\}$, és minden i-re az E_i esemény teljesen független az

$$\{E_j : j \neq i,\ (j,i) \notin E(G)\}$$

események halmazától.

Tétel (LLL (Lovász lokális lemma) egy változata): Legyenek $E_1, \dots, E_n$ események egy valószínűségi térben, G legyen egy függetlenségi gráf, $d \geq 1$, és teljesüljön:

- $\mathbb{P}(E_i) \leq p$ minden i-re,
- G-ben a csúcsok foka legfeljebb d,
- $4dp \leq 1$.

Ekkor $\mathbb{P}\left(\bigcap_{i=1}^n \overline{E_i}\right) > 0$.

Moser-Tardos algoritmus:

1. Kiértékeljük a $\xi \in \mathcal{P}$ változókat egy véletlen $\omega \in \Omega$ helyen. Ha egyik E_i sem következik be, akkor ω egy elemi esemény a $\bigcap_{i=1}^n \overline{E_i}$ metszetből, és az eljárás véget ér.

2. Ha valamelyik E_j bekövetkezett, akkor újra kiértékeljük a P_j-beli ξ változókat a saját eloszlásuk szerint, függetlenül. A $P \setminus P_j$ változókat változatlanul hagyjuk. Ezután ismét ellenőrizzük, hogy bekövetkezik-e valamelyik E_i, és szükség esetén ismételjük a 2. lépést.

Megjegyzés: Moser és Tardos bizonyították, hogy a 2. lépést várhatóan legfeljebb $\frac{n}{d-1}$ alkalommal kell elvégezni.

Forrás: *Rónyai Lajos: Véletlen és algoritmusok 2011* [5]

Programváltozatok: A Moser-Tardos módszer volt a kiindulási alap, így majdnem az összes kipróbált módszer használja egyes részeit. Kezdetben két fő programváltozat készült, mely pontosan a Moser-Tardos lépéseit követi.

1. E_i esemény: Egy csúcsosztályban két csúcsnak van két közös szomszédja.
2. E_i esemény: C_4 képződik.

A p (élbehúzási valószínűség) értéke kezdetben az LLL alapján lett megválasztva. Később, p-t növeltük jobb eredmények reményében. Ezért:

- Az 1. esetben p a Roman-féle felső becslés szerinti várható érték 80%-a.
- A 2. esetben 85%-a.

1. változat algoritmusa: Végig megyünk az éleken, és minden élt az LLL-ből kiszámolt/(megnövelt) valószínűséggel húzunk be, közben számon tartjuk a kisebbik/nagyobbik csúcsosztályban az összes csúcspár közös szomszédjainak számát. Ha egy (u, v) él behúzásra kerül, akkor az összes olyan (u, u') csúcspár közös szomszédjainak számát növeljük eggyel, ahol u' szomszédos v-vel. Ezután végigmegyünk minden csúcspáron, és ha a közös szomszédaik száma ≥ 2, akkor újrasorsoljuk a csúcspár összes élét. Ezt addig ismételgetjük, amíg van olyan csúcspár, ahol a közös szomszédok száma ≥ 2. (Triviális javítás a mohó algoritmus használata a folyamat vége után, ennek eredményeiről lentebb.)

LLL-ből számolt valószínűség: A gráf G csúcsainak foka: $2(m-2)$. *Magyarázat:* Vagy az egyik vagy a másik csúcsuk közös, csak így lehetnek összefüggők.

Ez alapján a maximum fok:

$$d = 2m - 4$$

A ε élbehúzás valószínűségére egy durva becslés adható:

$$p \leq \binom{n}{2} \cdot \varepsilon^4$$

Magyarázat: A $\binom{n}{2}$ lehetséges csúcspár mindegyikéhez két közös szomszéd választható, így egy adott csúcspárra akkor teljesül a feltétel, ha mind a négy él be van húzva. Mivel a vizsgált csúcspárok nem függetlenek, az ε-re kapott érték kisebb lesz, mint az LLL által meghatározható maximum érték, de nekünk most ez a becslés is elég lesz.

A feltétel:

$$\frac{1}{4d} = p = \binom{n}{2} \cdot \varepsilon^4$$

Innen az ε értéke:

$$\varepsilon = \sqrt[4]{\frac{1}{4d \cdot \binom{n}{2}}} = \sqrt[4]{\frac{1}{4(m-2) \cdot n(n-1)}}$$

Motiváció: A Moser-Tardos módszer jól ráillik a problémára és az újrasorsolások által pont két csúcs közös szomszédjait javítja ki, és ahol a gráf már C_4-mentes, azt kevésbé változtatja meg. Ezért ez a célzott javítgatás jobb eredményekre ad reményt.

Eredmények:

Z(n, m)	**Nagyobb + LLL**	**Nagyobb**	**Nagyobb + mohó**	**Kisebb + mohó**	**Referencia**
$Z(9,30)$	36	45	63	63	63
$Z(14,29)$	46	61	85	85	87
$Z(20,20)$	49	58	87	89	96
$Z(20,37)$	63	78	129	129	136
$Z(30,30)$	73	88	149	150	175
$Z(40,40)$	96	119	222	220	252
Él %	46.33%	57.68%	92.62%	92.52%	100%
Diff	0, 0, 0	0, 0, 0	61, 44, 38	53, 49, 38	496, 0, 0
Futásidő	1790s	1058s	1755s	1824s	-
Iteráció	100000	8000	8000	6000	-

2. táblázat. Moser-Tardos első változatának eredményei

3. Moser-Tardos 2. változata

Algoritmus: Végig megyünk az éleken, és minden élt az LLL-ből kiszámolt valószínűséggel behúzunk, közben számon tartjuk a keletkezett C_4-eket. Ezután végigmegyünk minden C_4-en és újrasorsoljuk az összes élét. Ezt addig ismételgetjük, amíg van C_4. Mohó javítás nélkül sajnos nem hozott jó eredményeket ez a módszer sem, így ezután a mohó javítás minden programváltozatban jelen lesz.

LLL-ből számolt valószínűség: E_i esemény: C_4 képződik.
A G függetlenségi gráfban a csúcsok foka:

$$d = \binom{m}{2}\binom{n}{2} - \left(\binom{m-2}{2}\binom{n}{2} + 2\binom{m-2}{1}\binom{n-2}{2} + \binom{n-2}{2}\right) - 1$$

$$4dp \leq 1$$

Mivel egy C_4-et négy él behúzásával kapunk, ezért:

$$p = \varepsilon^4$$

$$\varepsilon^4 \leq \frac{1}{4d}$$

$$\varepsilon = \left(\frac{1}{4d}\right)^{1/4} = \text{élbehúzási valószínűség}$$

A p (élbehúzási valószínűség) értéke kezdetben az LLL alapján lett megválasztva. Később, p-t növeltük jobb eredmények reményében. A tapasztalat azt mutatta, hogy p nagysága nem igazán számított a végeredmények tekintetében, mivel a Moser Tardos módszer önmagában nem közelítette meg az ismert felső értékeket.

Motiváció: A Moser-Tardos módszer jól ráillik a problémára és a kevesebb újrasorsolás miatt célzottabban a C_4-eket javítja ki, és ahol a gráf már C_4-mentes, azt kevésbé változtatja meg.

Eredmények:

Z(n, m)	MT C4 + mohó	MT C4 + LLL + mohó	Referencia
$Z(9,30)$	63	72	63
$Z(14,29)$	85	85	87
$Z(20,20)$	87	97	96
$Z(20,37)$	130	130	136
$Z(30,30)$	150	150	175
$Z(40,40)$	222	221	252
Él %	92.8%	92.82%	100%
Diff	72, 40, 32	68, 40, 38	496, 0, 0
Futásidő	1828s	1769s	-
Iteráció	13000	15000	-

3. táblázat. Moser-Tardos második változatának eredményei

4. Moser-Tardos további javítása

Algoritmus: Ugyanaz, mint a javított verziónál, csak számon tartjuk, hogy melyik él hány különböző C_4-ben szerepel, és azt az élt töröljük, ami a legtöbb C_4-ben szerepel, majd ezt a törlést iteráljuk, amíg van C_4.

Motiváció: A fenti javított változatban igazából az újrasorsolás miatt lehet, hogy egy C_4 összes élét kitöröljük, mikor lehet, hogy csak 1 élt kéne kitörölni ahhoz, hogy a gráf C_4-mentes legyen. Ez a javítás ezt a problémát megoldja, és mivel mindig a leggyakoribb élt töröljük, ezért ezt a törlést nem kell olyan sokszor megcsinálni. Továbbá valószínűleg a gráf sűrűbb részéről törlünk ki éleket.

5. Több iteráció futtatása

Algoritmus:

...
for belső iterációszám **do**
  véletlenszerű élek beszúrása p valószínűséggel
  while van benne $K_{2,2}$ **do**
    távolítsuk el azt az élt, amit a legtöbb $K_{2,2}$ tartalmaz
  end while
end for
...

Motiváció: A legtöbb C_4-ben szereplő élek eltávolítására épülő változat célja, hogy kevesebb él törlésével szüntesse meg a C_4-eket, miközben megőrzi a gráf kiegyensúlyozott szerkezetét. Ha a gráf egy része már eleve kiegyensúlyozott, akkor az új élek hozzáadása nem, vagy csak minimálisan ront a szerkezeten. Ezzel szemben, ha egy részgráf erősen aszimmetrikus, akkor az új élek beszúrása és a későbbi célzott törlések javíthatják annak struktúráját. Így a folyamat végeredményeként a teljes gráf kiegyensúlyozottabbá válik, ami nagyobb élhalmazt, azaz több megtartható élt eredményezhet.

Eredmények

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Számítástudományi és Információelméleti Tanszék
1117 Budapest, Magyar tudósok körútja 2. I.B.132.
Telefon: 463-2585, Fax: 463-3157
http:\\www.cs.bme.hu

Z(n, m)	**MT C4 jav.**	**MT C4 jav. + iter**	**Referencia**
$Z(9, 30)$	63	62	63
$Z(14, 29)$	85	84	87
$Z(20, 20)$	87	87	96
$Z(20, 37)$	130	130	136
$Z(30, 30)$	152	151	175
$Z(40, 40)$	223	224	252
Él %	92.9%	92.94%	100%
Diff	64, 45, 37	36, 60, 37	496, 0, 0
Futásidő	1739s	1846s	-
Iteráció	11000	120	-
Belső Iteráció	–	20	-

4. táblázat. 4. és 5. módszer összehasonlítása

6. Dinamikusan karbantartott valószínűségek

Algoritmus: Hosszas kísérletezés után a legjobb alsó korlátokat az alábbi módszer adta, a valószínűségek számítása az alábbi képlet alapján történt:

1. Algorithm élbehúzási valószínűségek kódrészlet

```
int expected_m = ((double)upper_bound / m+0.5); // m osztályú csúcs várható értéke
int expected_n = ((double)upper_bound / n+0.5); // n osztályú csúcs várható értéke
double expected_n_percent = (double)upper_bound / (n * m);

double get_multiplier(int v_m){
    if(degree_m[v_m] >= expected_m){
        if(degree_m[v_m] == expected_m)
            return 0.9;
        return 0.5;
    }
    return 6.7 - 5.6 * ((double)degree_m[v_m] / expected_m);
}

double get_p(int v_m, int v_n){
    int n = degree_n[v_n];
    return min(get_multiplier(v_m) * (expected_n_percent * 1.1
           - (n/expected_n) * expected_n_percent), 0.8);
}
```

Motiváció: A következő fejlesztés az élbehúzási valószínűségek futás közbeni módosítása volt. A cél az volt, hogy a csúcsosztályok fokszámai a Roman-féle felső becslés várható értékei felé konvergáljanak. Mivel a Roman-féle felső becslés is egy olyan kiegyensúlyozott gráfot vesz alapul, ahol a kisebbik csúcsosztályban két csúcs fokszámának eltérése legfeljebb egy lehet.

7. További dinamikus valószínűségi módszerek

1. Csak a nagyobb csúcsosztály fokszámainak figyelembevétele: A Roman becslés is csak a nagyobbik osztály fokszámait használja, a kisebbik osztály fokszámai nincsenek figyelembe véve.
2. Valószínűségek fixálása minden iterációnál: Iterációk előtt a valószínűségek fixálása úgy, hogy a behúzott élek várható száma a felső becslés adott százaléka legyen. Így például, ha a felső becslés 100%-ára állítjuk be a várható értéket, akkor törlések előtt a gráf élszámának várható értéke minden iteráció után a felső becslés lesz. Ennél több élt úgysem húzhatnánk be anélkül, hogy a gráfban ne képződne biztosan C_4.

Eredmények: Látni lehet, hogy több iteráció futtatása igazán a dinamikusan karbantartott módszerrel kombinálva hatékony, a lenti eredmények is ezt támasztják alá, a későbbi módszerek mind használják ezt.

Z(n, m)	Dinamikus	Dinamikus + iter	Egy oldal	Fixált	Referencia
$Z(9,30)$	63	63	62	63	63
$Z(14,29)$	86	86	86	86	87
$Z(20,20)$	88	93	90	90	96
$Z(20,37)$	131	132	131	130	136
$Z(30,30)$	153	153	152	152	175
$Z(40,40)$	223	225	224	224	252
Él %	93.34%	94.52%	93.76%	94.28%	100%
Diff	87, 38, 37	133, 33, 38	92, 47, 36	112, 46, 34	496, 0, 0
Futásidő	1724s	1867s	1689s	1800s	-
Iteráció	5300	1000	900	1500	-
Belső Iteráció	–	20	30	20	-

5. táblázat. Dinamikus módszerek és variációik összehasonlítása

8. Már megtalált gráfok felhasználása bemenetként

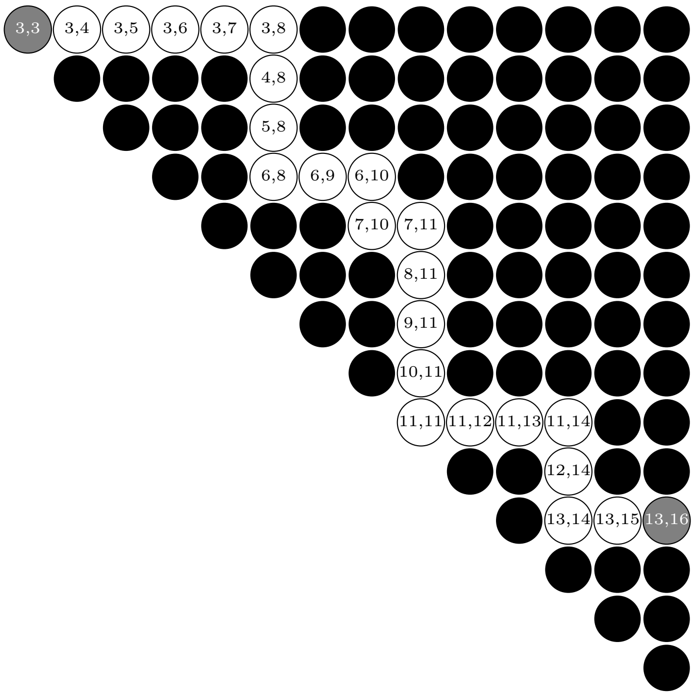
Algoritmus: Mielőtt elkezdenénk futtatni az algoritmusunkat $Z(m,n)$-re, keressük meg az eddigi legjobb $Z(m-1,n)$ vagy $Z(m,n-1)$ értéket, és olvassuk be a hozzá tartozó gráfot. (A program tesztelésénél ötször $Z(m-1,n)$, ötször $Z(m,n-1)$ adja a kiinduló gráfot, ezeket felváltva futtatjuk). Miután megvan a kiinduló gráf az újonnan hozzávett m' csúcsból kimenő éleket véletlen sorrendben megpróbáljuk hozzávenni a gráfhoz. Ehhez először végigmegyünk véletlen sorrendben az érintetlen N csúcsosztály csúcsain. Ha az eddig kiválasztott N' csúcsoknak nincs közös szomszédja az éppen vizsgált $n' \in N$ csúccsal, akkor kiválasztjuk, ellenkező esetben nem. Így kapunk egy N' csúcshalmazt az érintetlen N csúcsosztályból, ezt a folyamatot $\frac{|N|*|M|}{2}+5$-ször megismételjük, majd ezután a legnagyobb elemszámú csúcshalmaz lesz összekötve az újonnan hozzávett csúccsal. Látható, hogy az így kapott kiinduló gráf továbbra is C_4 mentes marad, hiszen N'-ben minden n_i, n_j csúcspárnak pontosan egy közös szomszédja van, mégpedig a most hozzávett m' csúcs.

Motiváció: Magától értetődő, hogy ha már van egy jó gráf, amit valamilyen módon újra tudunk használni a futás során, akkor nem éri meg nulláról indulni. Hiszen a meglévő gráf felépítése is időt vesz igénybe, így ez a költség többé-kevésbé megspórolható. A másik, megalapozottabb érv egy csúcs hozzáadására a következő:

> Ha $G=(A,B,E)$ egy $K_{2,2}$-mentes, (m,n) méretű páros gráf, és $\delta(A)$ az A osztályban vett minimális fokszám, akkor
>
> $$Z(m-1,n) \geq |E(G)| - \delta(A),$$

$$Z(m, n-1) \geq |E(G)| - \delta(B).$$

Az állítás magától értetődő, ami viszont meglepő lehet, hogy sok esetben a bemutatott becslés használata elég a felső korlát eléréséhez. Ezt jól szemlélteti, hogy a $Z(13, 16)$-tól egészen $Z(3, 3)$-ig el lehet jutni ezzel az egy becsléssel, úgy hogy közben csak $Z(m, n)$ pontos értékein haladunk át.



1. ábra. $Z(13, 16)$-ból $Z(3, 3)$-ba vezető út kizárólag csúcselvételekkel

Eredmények:

Z(n, m)	**Eredeti**	**Fixált**	**Referencia**
$Z(9, 30)$	63	63	63
$Z(14, 29)$	87	87	87
$Z(20, 20)$	96	96	96
$Z(20, 37)$	135	135	136
$Z(30, 30)$	175	175	175
$Z(40, 40)$	247	248	252
Él %	99.5%	99.29%	100%
Diff	339, 80, 39	274, 109, 52	496, 0, 0
Futásidő	1709s	1713s	-
Iteráció	1600	1600	-
Belső Iteráció	[1,3,6,12]	[1,3,6,12]	-

6. táblázat. Legjobb eddigi gráfhoz egy csúcs hozzáadását kihasználó módszerek eredményei

9. Öt eddigi legjobb gráf felhasználása bemenetként

Algoritmus: Ugyanaz, mint az előző algoritmus, csak minden iterációban az eddigi legjobb öt kiinduló gráfot olvasunk be, és mindegyikből lefuttatjuk az algoritmust. Továbbá ebben a változatban nem csak a legjobb, hanem az öt legjobb triviálisan nem kiegészíthető gráfot mentjük el.

Motiváció: Mi történik, hogyha $Z(m,n)$ és $Z(m,n+1)$ struktúrája lényegesen eltérő? Az eredeti kiegészítős módszer esetén hiába találjuk meg az optimális gráfot (m,n)-re, az $(m,n+1)$-hez használt kiinduló gráfon nagyon sok módosítást kellene végrehajtanunk. Ilyen esetben nem is biztos, hogy jól járunk a kiinduló gráf használatával. Öt, élszámra már $Z(m,n)$-hez közel álló, de struktúrájukban eltérő gráf használatával sokkal nagyobb esélyünk van arra, hogy kezelni tudjuk ezeket az eseteket.

Eredmények: (Megjegyzés: az iteráció számnál figyelembe kell venni, hogy iterációnként öt gráf feldolgozása történik)

Z(n, m)	**Eredeti**	**Fixált**	**Referencia**
$Z(9,30)$	63	63	63
$Z(14,29)$	87	87	87
$Z(20,20)$	96	96	96
$Z(20,37)$	134	134	136
$Z(30,30)$	175	166	175
$Z(40,40)$	247	248	252
Él %	99.41%	98.98%	100%
Diff	320, 82, 47	244, 89, 63	496, 0, 0
Futásidő	1824s	1713s	-
Iteráció	320	360	-
Belső Iteráció	[1,3,6,12]	[1,3,6,12]	-

7. táblázat. Öt legjobb eddigi gráfhoz egy csúcs hozzáadását kihasználó módszerek eredményei

Végső táblázat, értékelés

A módszer már egyórányi futtatás után is közel került a jelenleg ismert felső korlátokhoz. A végső eredményeket a **top 5-ös módszer** többszöri, hosszú futtatása, valamint ennek egy kissé módosított változata adta, mivel nem minden gráf struktúrája azonos. A minimum- és maximumfokszám az optimumban bizonyos esetekben nagyon távol lehet egymástól. A Roman becslés, és a bemutatott és használt dinamikus valószínűségi logika is a teljesen egyenletes fokszámeloszlást veszi alapul, ezért azoknak az eseteknek a megtalálása kevésbé valószínű a fenti módszerrel. Emiatt egy kicsit módosított valószínűségi logikával futó kód jobb eredményeket produkált azokban az esetekben, ahol a gráf optimális fokszámai erősen aszimmetrikusak.

Módosított logika: Minden élhez futtatás előtt véletlen eltérítési értéket generálunk, amely pozitív vagy negatív irányba módosítja az élbehúzási valószínűséget. Ez növeli az esélyt az aszimmetrikus, de optimális struktúrák megtalálására.

Eredmények:

Jelenleg ismert alsó becsléseket *Collins, Riasanovsky, Wallace, Radziszowski.* [1] cikkéből vettük. A cikk egészen $Z(31,31)$-ig ad pontos alsó becsléseket, viszont csak a főátlóra. Ezekben az esetekben az általunk bemutatott módszerekkel létrehozott fenti táblázat is mindig pontos. Egyik legfrissebb felső becslések *Tan* [6] cikkében találhatók, ahol $Z(m,n)$ értékei $2 \le m \le 24$ és $2 \le n \le 35$ tartományban szerepelnek. A táblázat $m \le 20$ és $n \le 26$ esetén minden értéket pontosan tartalmaz, illetve az aszimmetrikusabb párok esetén e tartományon túl is.

Az általunk alkalmazott módszerekkel előállított táblázat mindössze öt olyan esetet tartalmaz, ahol az ismert pontos értékektől elmarad. Ezen esetek: $Z(14,28)$, $Z(15,28)$, $Z(15,29)$, $Z(16,28)$, $Z(16,29)$.

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Számítástudományi és Információelméleti Tanszék
1117 Budapest, Magyar tudósok körútja 2. I.B.132.
Telefon: 463-2585, Fax: 463-3157
http:\\www.cs.bme.hu

Ugyanakkor összesen 16 esetben sikerült olyan felső becslést elérni, amelyekről [6] alapján korábban nem volt ismert, hogy valóban elérhetők-e. Ezek az esetek:

m	16	16	16	17	17	17	17	17	17	20	21	22	22	23	23	24
n	33	34	35	30	31	32	33	34	35	29	29	29	30	29	30	30
$Z_{2,2}(m,n)$	106	108	110	105	107	109	111	113	115	120	125	130	132	135	138	144

8. táblázat. Esetek, amikor a kapott eredményünk beállítja a [6] beli nem egzakt felső becsléseket

Végső táblázat:

$m\backslash n$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
3		6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
4			9	10	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
5				12	14	15	17	18	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
6					16	18	19	21	22	24	25	27	28	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
7						21	22	24	25	27	28	30	31	33	34	36	37	39	40	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
8							24	26	28	30	32	33	35	36	38	39	41	42	44	45	47	48	50	51	53	54	56	57	58	59	60	61	62	63	64	65	66	67	68
9								29	31	33	36	37	39	40	42	43	45	46	48	49	51	52	54	55	57	58	60	61	63	64	66	67	69	70	72	73	74	75	76
10									34	36	39	40	42	44	46	47	49	51	52	54	55	57	58	60	61	63	64	66	67	69	70	72	73	75	76	78	79	81	82
11										39	42	44	45	47	50	51	53	55	57	59	60	62	63	65	66	68	69	71	72	74	75	77	78	80	81	83	84	86	87
12											45	48	49	51	53	55	57	60	61	63	65	66	68	70	72	73	75	76	78	79	81	82	84	85	87	88	90	91	93
13												52	53	55	57	59	61	64	66	67	69	71	73	75	78	79	81	82	84	85	87	88	90	91	93	94	96	97	99
14													56	58	60	63	65	68	70	72	73	75	78	80	82	84	85	87	89	91	92	94	96	98	99	101	102	104	105
15														61	64	67	69	72	75	77	78	80	82	85	86	88	90	92	95	96	98	100	102	105	106	108	109	111	112
16															67	70	73	76	80	81	83	85	87	90	91	93	95	97	100	102	103	106	108	110	111	113	115	117	118
17																74	77	80	84	85	87	89	91	94	96	98	101	102	105	107	109	111	113	115	117	119	121	123	125
18																	81	84	88	90	91	93	96	99	101	103	106	108	109	112	114	116	118	120	123	124	127	128	130
19																		88	92	95	96	98	100	103	106	108	111	114	115	117	119	121	124	126	129	130	132	134	136
20																			96	100	101	103	105	108	111	113	116	120	121	123	125	127	129	131	135	136	138	140	142
21																				105	106	108	110	112	116	118	121	125	126	128	130	132	135	137	141	142	144	147	149
22																					108	110	114	117	120	123	126	130	132	133	136	138	140	143	147	148	150	153	155
23																						115	118	121	125	128	131	135	138	139	141	143	146	148	151	154	156	159	161
24																							122	126	129	133	136	140	144	145	147	149	151	154	156	158	161	164	166
25																								130	134	138	141	145	150	151	153	155	157	160	162	164	167	170	172
26																									138	142	146	150	155	156	158	161	163	165	168	170	172	176	178
27																										147	151	155	160	162	163	166	168	170	173	175	178	181	184
28																											156	160	165	168	169	171	174	176	178	181	183	187	190
29																												165	170	174	175	177	180	182	184	187	189	192	195
30																													175	180	181	183	186	188	190	193	195	198	201
31																														186	187	189	192	194	196	199	201	204	207
32																															189	191	194	196	200	203	206	208	212
33																																195	198	201	204	207	210	213	217
34																																	202	207	210	213	216	218	222
35																																		213	216	219	222	225	228
36																																			221	224	227	231	234
37																																				228	231	235	238
38																																					235	239	243
39																																						243	248
40																																							252

9. táblázat. $Z_{2,2}(m,n)$ alsó becslések

`C++` kódok, talált gráfok:

A projekttel kapcsolatos fájlok megtalálhatóak az alábbi linken:
`https://github.com/balazsborsik/Onlab`

Hivatkozások

[1] Alex F. Collins és tsai. "Zarankiewicz Numbers and Bipartite Ramsey Numbers". *arXiv:2002.00903* (2020).

[2] Gábor Damásdi, Tamás Héger és Tamás Szőnyi. "The Zarankiewicz problem, cages and geometries". (2013).

[3] Wayne Goddard, Michael A. Henning és Ortrud R. Oellermann. "Bipartite Ramsey numbers and Zarankiewicz numbers". *European Journal of Combinatorics* 23.7 (2002), 761–766. old.

[4] Steven Roman. "A Problem of Zarankiewicz". *Journal of Combinatorial Theory, Series B* 18.3 (1975), 289–292. old.

[5] Lajos Rónyai. *Véletlen és algoritmusok*. Budapest: Typotex, 2011.

[6] Jeremy Tan. "An attack on Zarankiewicz's problem through SAT solving". *arXiv preprint arXiv:2201.12345* (2022).

Budapest, 2025. május 25.