

FAKULTETA ZA MATEMATIKO IN FIZIKO  
UNIVERZA V LJUBLJANI

# Zunajravninski grafi

Poročilo naloge

Jure Kraševac, Urh Videčnik

Februar 2026

# Kazalo

1	Teoretični uvod	2
2	Cilj naloge	2
3	Zunajravnski grafi na $n \leq 10$ vozliščih	2
4	Naključno generirani zunajravnski grafi	5
5	Preverjanje neenakosti za večja števila vozlišč	6

## 1 Teoretični uvod

Naj bo  $G = (V, E)$  graf, kjer je  $V$  množica vozlišč in  $E$  množica povezav med vozlišči. Zunajravninski graf (ang. Outerplanar graph) je graf, ki ga lahko narišemo v ravnini tako, da se nobeni dve povezavi ne sekata in da vsa vozlišča ležijo na zunanji strani oziroma zunanjem licu grafa. Takšni grafi so podmnožica ravninskih grafov.

Množica vozlišč  $S \subseteq V$  grafa  $G$  je liha neodvisna množica, če za vsako vozlišče  $u \in S$  velja, da je lihe stopnje ter za vsak par  $u, v \in S$  velja, da med njima ne obstaja povezava v  $E$ . Za liho neodvisno množico  $S$  velja, da vsako vozlišče  $v \in V \setminus S$  nima nobenega soseda v  $S$ :  $N(v) \cap S = \emptyset$  ali pa ima liho število sosedov v  $S$ :  $|N(v) \cap S| \equiv 1 \pmod{2}$ . Z  $\alpha_{od}(G)$  označimo največjo moč lihe neodvisne množice grafa  $G$ .

## 2 Cilj naloge

Cilj naloge je preveriti, veljavnost neenakosti

$$\alpha_{od}(G) \geq n/7,$$

za vsak zunajravninski graf  $G$  z  $n$  vozlišči. Nalogo smo razdelili na dva dela. Prvi del je namenjen preverjanju neenakosti na manjših zunajravninskih grafih - grafih z največ 10 vozlišči. V tem delu bomo implementirali algoritem, ki bo generiral vse zunajravninske grafe z največ 10 vozlišči in za vsak graf izračunal  $\alpha_{od}(G)$ . V drugem delu pa bomo s postopkom lepljenja zunajravninskih grafov, generirali zunajravninske grafe z več kot 10 vozlišči in poskušali najti protiprimer, torej graf, za katerega ne velja zgornja neenakost.

## 3 Zunajravninski grafi na $n \leq 10$ vozliščih

Najprej bomo preverili neenakost  $\alpha_{od}(G) \geq n/7$  za vse zunajravninske grafe z največ 10 vozlišči. Prvo bomo implementirali funkcijo, ki za dan graf preveri, ali je zunajravninski. Ker za vsak zunajravninski graf velja, da ga lahko narišemo kot krožno ravninski graf (angl. Circular planar graph), lahko za preverjanje zunajravninskosti uporabimo vgrajeno metodo `is_circular_planar`. Funkcijo za preverjanje zunajravninskosti smo testirali na nekaj grafih, da smo se prepričali o njeni pravilnosti. Za samo iskanje števila vseh zunajravninskih grafov smo definirali funkcijo `outerplanar_graphs(n)`. Znotraj funkcije smo uporabili metodo `planar_graphs(n)`, ki generira vse ravninske grafe z  $n$  vozlišči. Uporabo te metode nam omogoča program Plantri, katerega naložimo znotraj SageMath okolja. Na ta način prihranimo čas, saj je

število ravninskih grafov z  $n$  vozlišči veliko manjše od števila vseh grafov z  $n$  vozlišči. Za nadaljno filtracijo uporabimo pogoj `G.size() ≤ G.order() − 3`, ki je nujen pogoj za zunajravninske grafe. Potem za vsak graf, ki izpolnjuje ta pogoj, preverimo še, ali je zunajravninski z uporabo funkcije `preveri_outerplanarnost(G)`. Ker se želimo izogniti podvajanju izomorfnih grafov, shranjujemo le tiste grafe, ki niso izomorfni z že shranjenimi grafi, kar najlažje dosežemo z uporabo kanonične oblike grafa, ki jo dobimo z metodo `canonical_label()`.

Število zunajravninskih grafov na  $n \leq 10$  vozliščih je prikazano v spodnji tabeli:

$n$	število zunajravninskih grafov na $n$ vozliščih
1	1
2	1
3	2
4	5
5	13
6	46
7	172
8	777
9	3 783
10	20 074

Tabela 1: Število zunajravninskih grafov na  $n \leq 10$  vozliščih.

Potem smo s funkcijo `check_alpha_od(G)` preverili, ali za vse zunajravninske grafe z največ 10 vozlišči velja neenakost  $\alpha_{od}(G) \geq n/7$ . Za izračun  $\alpha_{od}(G)$  smo implementirali funkcijo `alpha_od(G)`, ki izračuna največjo moč lihe neodvisne množice grafa  $G$  z uporabo celoštevilskega linearne programiranja. Najprej definiramo binarne spremenljivke  $x_v$ , ki označujejo, ali je vozlišče  $v$  vključeno v liho neodvisno množico ter  $y_v$ , ki označuje, ali ima vozlišče  $v$  sode, ki so vključeni v liho neodvisno množico ter celoštevilsko spremenljivko  $z_v$ , ki predstavlja števec za vozlišče  $v$ . Cilj funkcije je maksimizirati vsoto  $x_v$  za vsa vozlišča  $v \in V$ , kar predstavlja moč lihe neodvisne množice. Ugotovili smo, da neenakost velja za vse zunajravninske grafe z največ 10 vozlišči.

Poskusili smo tudi preveriti neenakost za zunajravninske grafe z največ 15 vozlišči. Najprej smo prvotno funkcijo `outerplanar_graphs(n)` prilagodili tako, da ne uporablja kanoničnega zapisa grafov in potem tudi ne preverjamo izomorfizma grafov, kar je sicer računsko zahtevno, vendar se je izkazalo, da je število zunajravninskih grafov z večjim številom vozlišč preveliko, da bi jih

lahko sistematično pregledali. Ukazno vrstico, ki shrani sezname zunajravninskih grafov z največ 15 vozlišči v en slovar,

```
slovar_zunajravninskih_grafov2 =
    {n: allouterplanar_graphs(n) for n in range(1, 16)}
```

smo pustili teči približno 50 ur, vendar v tem času žal nismo uspeli pridobiti vseh zunajravninskih grafov z 15 vozlišči, saj se je število zunajravninskih grafov z večjim številom vozlišč zelo povečalo. Zato smo poskusili s sprotnim shranjevanjem zunajravninskih grafov v slovar:

```
slovar_zunajravninskih_grafov2 = {}
for n in range(1, 16):
    print(f"Iskanje grafov za število vozlišč n = {n}")
    slovar_zunajravninskih_grafov2[n] = allouterplanar_graphs(n)
```

Vseeno smo uspeli pridobiti le zunajravninske grafe z največ 10 vozlišči, saj se je koda izkazala za prepočasno, da bi lahko pridobili zunajravninske grafe z večjim številom vozlišč. Spodnja tabela prikazuje število vseh zunajravninskih grafov (brez preverjanja izomorfности grafov) z največ 10 vozlišči, ki smo jih uspeli pridobiti:

$n$	število zunajravninskih grafov na $n$ vozliščih
1	1
2	1
3	2
4	5
5	18
6	100
7	738
8	6 823
9	70 547
10	781 328

Tabela 2: Število vseh zunajravninskih grafov na  $n \leq 10$  vozliščih.

Opazimo, da se število zunajravninskih grafov brez preverjanja izomorfности grafov zelo poveča. Za  $n = 10$  vozlišč smo uspeli pridobiti 781 328 zunajravninskih grafov, ko pa preverimo izomorfizem grafov, se število zunajravninskih grafov z 10 vozlišči zmanjša na 20 074.

Sedaj smo še s funkcijo `check2_alpha_od(G)` preverili, ali za vse zunajravninske grafe iz zgornje tabele velja neenakost  $\alpha_{od}(G) \geq n/7$ . Ugotovili

smo, da neenakost velja za vse zunajravninske grafe z največ 10 vozlišči, tudi za tiste, ki so izomorfni.

Za vsak  $n$  smo si še ogledali najmanjšo moč lihe neodvisne množice ter število grafov, pri katerih je ta vrednost dosežena. Tu smo uporabili zunajravninske grafe pridobljene s funkcijo `outerplanar_graphs`, torej s preverjanjem izomorfnosti. Rezultati so prikazani v spodnji tabeli:

$n$	$\min \alpha_{od}$	število zunajravninskih grafov
1	1	1
2	1	1
3	1	2
4	1	3
5	1	5
6	1	4
7	1	2
8	2	204
9	2	428
10	2	687

Tabela 3: Število zunajravninskih z minimalno vrednostjo  $\alpha_{od}$  na  $n \leq 10$  vozliščih.

## 4 Naključno generirani zunajravninski grafi

Ker je število zunajravninskih grafov z večjim številom vozlišč zelo veliko, v drugem delu naloge ne moremo več pregledati vseh možnih primerov. Namesto tega uporabimo pristop z naključnim generiranjem zunajravninskih grafov in eksperimentalnim preverjanjem neenakosti  $\alpha_{od}(G) \geq \frac{|V(G)|}{7}$ .

Za ta namen smo implementirali funkcijo `random_outerplanar_graph(n)`, ki za dano število vozlišč  $n$  generira naključen zunajravninski graf. Algoritem temelji na znani strukturni lastnosti zunajravninskih grafov, da jih lahko razstavimo na 2-povezane zunajravninske bloke, ki so med seboj povezani v drevesno strukturo.

Najprej ustvarimo 2-povezan zunajravninski blok kot podgraf triangulacije konveksnega mnogokotnika. To dosežemo s postopkom dodajanja t. i. uhelj, pri katerem zaporedoma vstavljamo nova vozlišča na robove zunanega cikla in jih povežemo z ustreznima krajiščema. Tako dobimo maksimalen zunajravninski graf. Da povečamo raznolikost generiranih primerov, nato naključno odstranimo del diagonal, pri čemer vedno ohranimo robove zunanega cikla. Ker odstranjujemo le robove, zunajravninskost ostane ohranjena.

Ko ustvarimo začetni blok, nove bloke postopoma dodajamo, dokler ne dosežemo želenega števila vozlišč. Pri vsakem dodajanju se naključno odločimo med dvema načinoma lepljenja:

- blok dodamo disjunktno in ga z obstoječim grafom povežemo z enim samim robom (mostom),
- blok prilepimo tako, da identificiramo eno njegovo vozlišče z izbranim vozliščem že zgrajenega grafa.

Oba postopka ohranjata zunajravninskost grafa. Prvi ustvari mostove, drugi pa rezalna vozlišča, zato dobljeni grafi praviloma niso 2-povezani, kar je skladno s splošno strukturo zunajravninskih grafov.

Če na koncu ostane manjše število vozlišč, ki ne zadostuje za tvorbo novega bloka, jih dodamo kot liste, povezane z enim robom na že obstoječi graf.

## 5 Preverjanje neenakosti za večja števila vozlišč

Za eksperimentalno preverjanje neenakosti smo definirali novo testno funkcijo `test_outerplanar_alpha_k_trials(n,m,k)`, ki za vsako naravno število v danem območju velikosti grafov  $n \leq |V(G)| \leq m$  generira  $k$  naključnih zunajravninskih grafov in za vsakega izračuna vrednost  $\alpha_{od}(G)$ . Za vsak generiran graf preverimo, ali velja  $\alpha_{od}(G) \geq \frac{|V(G)|}{7}$ .

S tem pristopom ne moremo dokazati veljavnosti neenakosti za vse zunajravninske grafe, lahko pa z veliko verjetnostjo zaznamo morebitne protiprimere. Pognali smo funkcijo za  $n = 15$ ,  $m = 30$  in  $k = 10$ , kjer pa nismo našli nobenega protiprimera, kar dodatno potrjuje pravilnost domnevane neenakosti.

Treba je poudariti, da je ta del naloge povsem eksperimentalen. Naključno generiranje grafov ne zagotavlja enakomerne porazdelitve po vseh zunajravninskih grafih, zato rezultatov ne moremo obravnavati kot dokaz.