

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Nina Smole, Martin Čadež

# Število geodetskih podpoti: ekstremalni grafi

Projektna naloga

Mentor: prof. dr. Riste Škrekovski

Ljubljana, 2025

## 1 Uvod

Namen te raziskave je bil sistematično preučiti ekstremalne grafe glede na število geodetskih podpoti (angl. *geodesic path number*, okrajšano *gpn*) v različnih strukturnih razredih povezanih grafov. Specifično nas je zanimala identifikacija grafov, ki v danem razredu dosežejo maksimalno možno vrednost *gpn* za fiksno število vozlišč  $n$ .

Osrednje raziskovalno vprašanje je bilo, kateri grafi maksimizirajo število geodetskih podpoti med vsemi povezanimi grafi na  $n$  vozliščih.

Reševanja problema smo se lotili v treh glavnih korakih. V prvem koraku smo izvedli izčrpno analizo grafov vseh navedenih razredov za majhna ( $\leq 10$ ) števila vozlišč  $n$ . To nam je omogočilo vpogled v obnašanje različnih tipov grafov. V drugem koraku smo na podlagi prejšnje analize postavili hipotezo o tem, kateri grafi so optimalni za poljubno število vozlišč  $n$ . V zadnjem koraku smo postavljeni hipotezo preverili z uporabo stohastičnih metod, za večja števila vozlišč.

## 2 Definicije in teoretično ozadje

Za natančno obravnavo problema najprej ponovimo ključne definicije.

**Definicija 1 (Pot).** Za povezan graf  $G$  je *pot* dolžine  $\ell$  zaporedje vozlišč  $(v_0, v_1, \dots, v_\ell)$  brez ponovitev, kjer je vsak par zaporednih vozlišč povezan z povezavo grafa  $G$  (tj.  $v_{i-1}v_i \in E(G)$  za vse  $1 \leq i \leq \ell$ ). Trivialna pot dolžine 0 vsebuje eno vozlišče.

**Definicija 2 (Geodetska pot).** *Geodetska pot* med vozliščema  $u$  in  $v$  v grafu  $G$  je pot med njima minimalne dolžine. Dolžino te poti označimo z  $d_G(u, v)$  in jo imenujemo razdalja med  $u$  in  $v$ .

**Definicija 3 (Število geodetskih podpoti).** Za povezan graf  $G$  je *število geodetskih podpoti*  $gpn(G)$  definirano kot število vseh geodetskih poti v grafu, vključno s trivialnimi potmi dolžine 0. Formalno:

$$gpn(G) = \sum_{u,v \in V(G)} \text{število geodetskih poti med } u \text{ in } v.$$

Za vsak par različnih vozlišč je vsaj ena geodetska pot, za vsako vozlišče pa štejemo tudi trivialno pot.

V raziskavi smo se osredotočali na naslednje strukturne razrede povezanih grafov:

**Definicija 4 (Dvodelni graf).** Graf  $G = (V, E)$  je *dvodelen*, če obstaja particija množice vozlišč  $V = A \cup B$ ,  $A \cap B = \emptyset$ , tako da za vsako povezavo  $uv \in E$  velja  $(u \in A \wedge v \in B) \vee (u \in B \wedge v \in A)$ . Vse povezave torej povezujejo vozlišča iz različnih partit.

**Definicija 5 (Graf brez trikotnikov).** Graf  $G$  je *brez trikotnikov*, če ne vsebuje nobenega cikla dolžine 3 kot podgrafa. Ta razred je širši od razreda dvodelnih grafov, saj vsak dvodelni graf ne vsebuje lihih ciklov, zlasti tudi ne trikotnikov.

**Definicija 6 (Kubični graf).** Graf  $G$  je *kubičen* (oz. 3-regularen), če ima vsako vozlišče stopnjo 3. Za kubične grafe na  $n$  vozliščih mora biti  $n$  sod in število povezav je  $3n/2$ .

**Definicija 7 (Ekstremalni graf).** Graf  $G$  je *ekstremalen* glede na invarianto  $P$  v razredu grafov  $\mathcal{C}$ , če med vsemi grafi iz  $\mathcal{C}$  doseže največjo (ali najmanjšo) možno vrednost  $P(G)$ . V naši študiji smo iskali grafe, ki maksimizirajo invarianto  $gpn(G)$  znotraj posameznega razreda.

### 3 Reševanje problema na majhnih grafih

V prvem delu smo definirali module za generiranje grafov treh zahtevanih tipov- dvodelnih, kubičnih in brez trikotnikov. Koda je uporabila knjižnice SageMath, NetworkX in Matplotlib za generiranje, analizo in vizualizacijo grafov.

#### 3.1 Generiranje grafov

Del (284 934) analiziranih grafov je bil pridobljen z uporabo spodaj opisanih modulov za generiranje grafov. Preostanek (11 721 442) je bil pridobljen iz spletne baze podatkov<sup>1</sup>.

##### Dvodelni grafi

Razred `BipartiteGraph` konstruira povezane dvodelne grafe. Zahteva tri parametre: število vozlišč v prvi množici (`num_u`), število vozlišč v drugi množici (`num_v`) in število povezav (`num_edges`). Validacijska metoda zagotavlja, da je število povezav med teoretičnimi mejami: minimalno število za povezan graf ( $U + V - 1$ ) in maksimalno za popoln dvodelni graf ( $U \times V$ ). Konstrukcija grafa poteka postopno z ustvarjanjem vozlišč  $u_i$  in  $v_j$ , dodelitvijo atributov za dvodelnost, vzpostavitvijo minimalne povezanosti in dodajanjem preostalih povezav. Razred ponuja lastnosti za dostop do sosednostne in incidencne matrike, zaporedja stopenj ter seznamov vozlišč po množicah. Vizualizacijska metoda uporablja bipartitno postavitev z različnima barvama za vsako množico vozlišč.

##### Kubični grafi

Razred `CubicGraphs` generira vse povezane kubične (3-regularne) grafe z danim številom vozlišč. Zahteva parameter `num_nodes`, ki mora biti sodo število vsaj 4. Generiranje poteka z uporabo SageMathove implementacije algoritma `nauty` s parametri `-d3 -D3 -c`, ki zagotavljajo 3-regularnost in povezanost grafov. Rezultat je seznam vseh neizomorfnih kubičnih grafov z danim številom vozlišč. Razred omogoča dostop do seznamov povezav, sosednostnih in incidencnih matrik za vse generirane grafe. Vizualizacija uporablja krožno postavitev vozlišč z enotno barvo.

##### Grafi brez trikotnikov

Razred `TriangleFreeGraphs` generira vse povezane grafe brez trikotnikov z določenim številom vozlišč. Zahteva parameter `num_nodes` brez posebnih omejitev. Generiranje poteka s pomočjo algoritma `nauty` s parametrom `-t`, ki izključuje grafe s trikotniki, in parametrom `-c` za povezanost. Poleg standardnih lastnosti (povezave, matrike) razred ponuja tudi metodo za pridobitev zaporedij stopenj vseh generiranih grafov. Vizualizacija uporablja krožno postavitev.

#### 3.2 Izračun števila geodetskih podpoti

Implementirali smo funkcijo `gpn`, ki izračuna število vseh najkrajših poti med vsemi pari vozlišč v grafu. Če je parameter `count_trivial` nastavljen na `True`, v štetje vključi tudi trivialne poti (poti dolžine 0 od vozlišča do samega sebe), sicer jih izloči. Algoritem deluje tako, da najprej inicializira števec na število vozlišč (za trivialne poti, če so vključene). Nato za vsak par različnih vozlišč  $(u, v)$  v grafu pobere vse najkrajše poti med njima z uporabo funkcije `nx.all_shortest_paths` in prišteje njihovo število k skupnemu rezultatu. Funkcija uporablja optimizacijo, da za vsak par vozlišč šteje poti samo v eno smer, saj je množica najkrajših poti od  $u$  do  $v$  enaka množici najkrajših poti od  $v$  do  $u$ .

<sup>1</sup><https://hog.grinvin.org/><https://users.cecs.anu.edu.au/~bdm/data/graphs.html>

**Algorithm 1** Štetje vseh najkrajših poti v grafu (GPN)

---

```

1: procedure GPN( $G, count\_trivial$ )
2:   if  $count\_trivial$  then
3:      $total\_paths \leftarrow |V(G)|$ 
4:   else
5:      $total\_paths \leftarrow 0$ 
6:   end if
7:    $nodes \leftarrow$  seznam vseh vozlišč v  $G$ 
8:   for  $i \leftarrow 0$  to  $|nodes| - 1$  do
9:      $u \leftarrow nodes[i]$ 
10:    for  $j \leftarrow i + 1$  to  $|nodes| - 1$  do
11:       $v \leftarrow nodes[j]$ 
12:       $paths \leftarrow$  vse_najkrajše_poti( $G, u, v$ )
13:       $total\_paths \leftarrow total\_paths + |paths|$ 
14:    end for
15:   end for
16:   return  $total\_paths$ 
17: end procedure

```

---

Računanja  $gpn$  smo se lahko lotili tudi z uporabo celoštevilskega linearnega programa (z nelinearnimi omejitvami).

**Celoštevilski linearni program za maksimizacijo geodetskega števila:**

Naj bo  $G \in \mathcal{G}(V, E)$  neusmerjen neutrežen povezan enostaven označen graf nad  $n$ -timi vozlišči. Označimo  $V(G) = \{1, \dots, n\}$  in  $E(G) \subseteq \{e_{ij} \mid \forall i, j \in V(G) : e_{ij} = (i, j)\}$ . Najti smo želeli graf z največjim geodetskim številom.

$$\text{ciljna funkcija: } \max \left( \sum_{i=1}^n \sum_{j=1}^n y_{ij} \right)$$

kjer :

1. definiramo vse matrike sosednosti za enostavne in povezane grafe

- $\forall i, j \in \{1, \dots, n\} : x_{ij} = \begin{cases} 1 & ; e_{ij} \in E(G) \\ 0 & ; \text{sicer} \end{cases}$  (sestavimo matriko)
- $\sum_{i < j} x_{ij} \geq n - 1$  (graf je povezan)
- $\forall i, j \in \{1, \dots, n\} : x_{ij} = x_{ji}$  (graf je neusmerjen)
- $\forall i \in \{1, \dots, n\} : x_{ii} = 0$  (brez zank)
- $\forall i, j \in \{1, \dots, n\} : x_{ij} \in \{0, 1\}$  (brez večkratnih povezav)

2. definiramo funkcionalno spremenljivko odvisno od matrike sosednosti

- $\forall i, j \in \{1, \dots, n\} : y_{ij} = \{\text{število najkrajših poti med vozliščema } i \text{ in } j\}$
- $\forall i, j \in \{1, \dots, n\}, i < j : y_{ij} = 0$  (vsako pot štejemo le enkrat)
- $\forall i \in \{1, \dots, n\} : y_{ii} = 1$  (trivialne poti)

- $\forall i, j \in \{1, \dots, n\} : y_{ij} \in \mathbb{N}$

Če je podan CLP doposten za izbrano število vozlišč  $n$  in je  $(x_{ij}^*)_{i,j=1}^n$  optimalna rešitev, ki predstavlja graf  $G^*$ , je to rešitev problema. Pri tem je potrebno omeniti, da lahko za dani  $n$  obstaja več neizomorfnih grafov, ki dosežejo optimum.

## 4 Eksperimentiranje in postavitev hipoteze

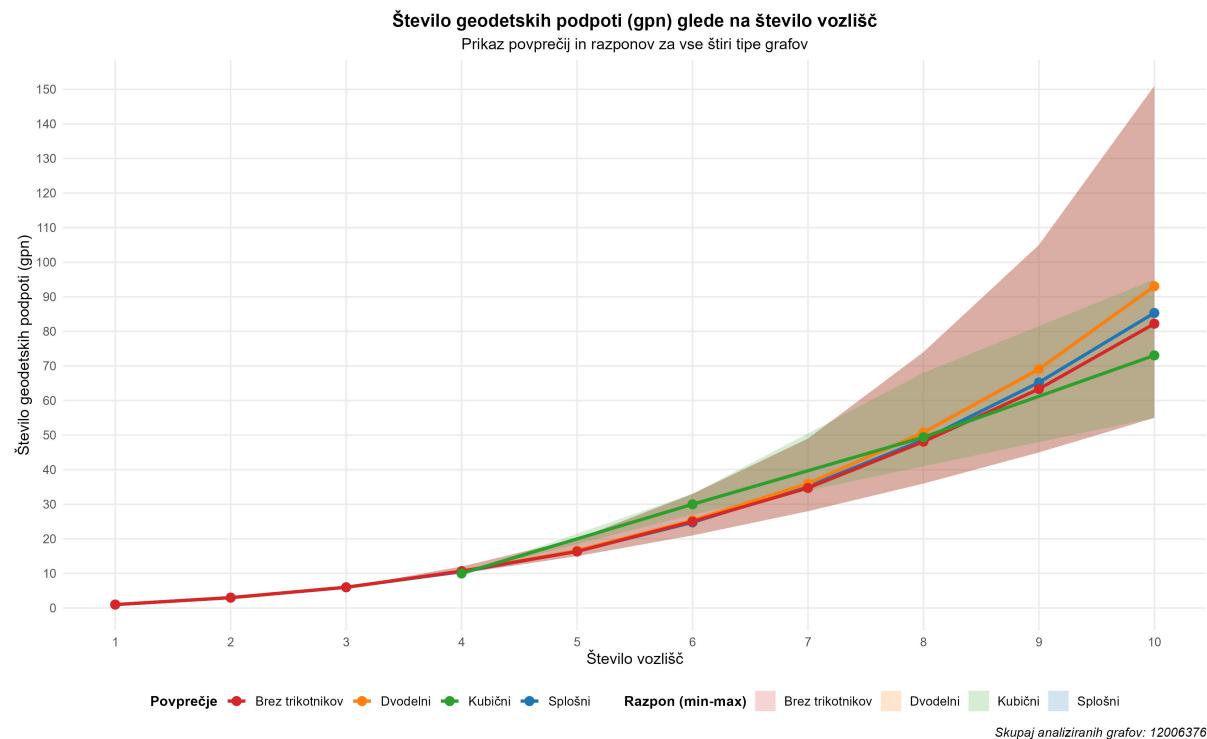
V eksperimentalnem koraku smo uporabili zgornje module za generiranje grafov in izračun  $gpn$  za različna števila vozlišč  $n$ . Za vsak razred generiranih grafov (dvodelni, kubični, brez trikotnikov in splošni grafi) smo zabeležili število vozlišč, povezav in prešteto število  $gpn$ . Za grafe pridobljene iz spletne baze smo zaradi odsotnosti podatka o številu povezav analizirali samo na podlagi vozlišč. Iz rezultatov smo izračunali povprečno, maksimalno in minimalno vrednost  $gpn$  za vsak razred grafov in vsako število vozlišč (manjše od 10). Rezultate smo predstavili v grafih 1 in 2. V tabeli 1 so predstavljeni grafi z največjim številom  $gpn$  za vsako število vozlišč od 1 do  $10^2$ .

Tabela 1: Grafi z največjim številom  $gpn$

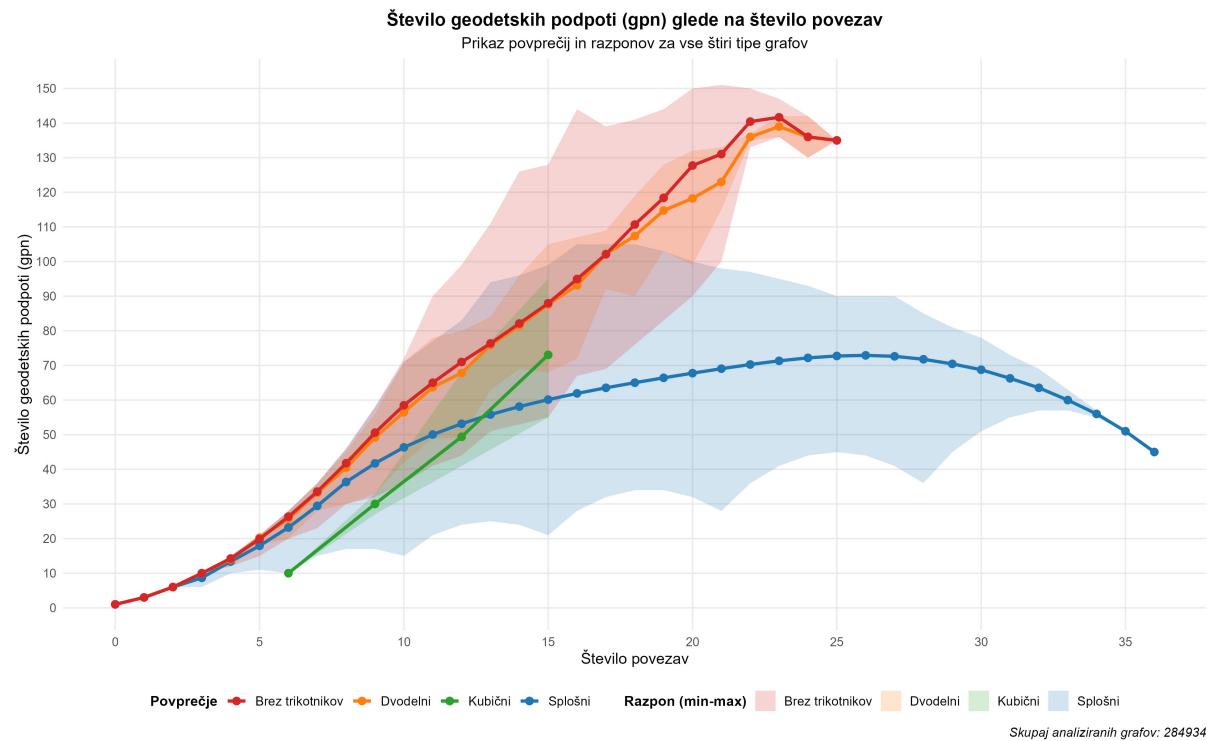
| Št. vozlišč | Ime grafa               | Št. povezav | Št. $gpn$ |
|-------------|-------------------------|-------------|-----------|
| 1           | bipartite_1_nodes_1     | 0           | 1         |
| 2           | bipartite_2_nodes_1     | 1           | 3         |
| 3           | bipartite_3_nodes_1     | 2           | 6         |
| 4           | bipartite_4_nodes_3     | 4           | 12        |
| 5           | bipartite_5_nodes_4     | 6           | 20        |
| 6           | bipartite_6_nodes_17    | 9           | 33        |
| 7           | bipartite_7_nodes_36    | 11          | 49        |
| 8           | bipartite_8_nodes_176   | 14          | 74        |
| 9           | bipartite_9_nodes_592   | 17          | 105       |
| 10          | bipartite_10_nodes_3816 | 21          | 151       |

V prvem grafu 1 smo opazili, da se je število geodetskih podpoti povečevalo z naraščajočim številom vozlišč za vse razrede grafov. Dvodelni grafi so kazali najvišje vrednosti  $gpn$ , sledili so jim splošni grafi, medtem ko so kubični grafi dosegali najnižje vrednosti  $gpn$  za enako število vozlišč. Ker je število geodetskih podpoti močno odvisno od števila povezav, smo v drugem grafu 2 prikazali  $gpn$  glede na število povezav. Tudi tukaj so dvodelni grafi izstopali z najvišjimi vrednostmi  $gpn$  za dano število povezav, medtem ko so kubični grafi ostajali na dnu lestvice. Opazili smo tudi, da  $gpn$  grafom s številom povezav najprej naraščal do neke točke, nato pa je začel upadati. To se je zgodilo, ker smo na začetku z dodajanjem povezav ustvarjali več alternativnih najkrajših poti med vozlišči, torej je  $gpn$  naraščal, hkrati pa s tem manjšali razdalje med njimi. Ko je število povezav preseglo določeno mejo (graf se je bližal polnemu grafu), so se razdalje med vozlišči zmanjšale do te mere, da je večina poti postala neposrednih (dolžine 1), kar je vodilo v zmanjšanje števila različnih najkrajših poti in s tem upad  $gpn$ .

<sup>2</sup>V analizi je bilo vsakemu grafu dodeljeno unikatno ime. V primeru dvodelnega grafa je to *bipartite\_x\_nodes\_y*, kjer je x število vozlišč in y zaporedna številka tvorjenja grafa.



Slika 1: Število geodetskih podpoti v odvisnosti od števila vozlišč.



Slika 2: Število geodetskih podpoti v odvisnosti od števila povezav.

Hipotezo smo temeljili na osnovi opažanja, da so dvodelni grafi dosegali najvišje vrednosti  $gpn$ . Ker so dvodelni grafi razred grafov brez trikotnikov (brez lihih ciklov) in ker

so grafi brez trikotnikov dosegali visoke vrednosti  $gpn$ , smo sklepali, da odsotnost trikotnikov prispeva k večjemu številu geodetskih poti. Poleg tega smo iz analize rezulatov opazili, da so optimalni tisti dvodelni grafi, ki so bili sorazmerno uravnoteženi glede na število vozlišč v obeh particijah in so imeli zmerno število povezav, kar je omogočalo več alternativnih najkrajših poti med vozlišči.

Iz teh rezultatov smo postavili sledečo hipotezo. *Maksimalno število geodetskih podpoti v grafu z danim številom vozlišč  $n$  dosežejo uravnoteženi dvodelni grafi s particijo vozlišč čim bolj enakomerne velikosti. Pri čemer velja: za sod  $n = 2k$  je optimalna particija  $(k, k)$ , za lich  $n = 2k + 1$  je optimalna particija  $(k, k + 1)$ . Poleg tega mora biti graf čim bolj regularen znotraj vsake particije, to je, da imajo vozlišča znotraj iste particije čim bolj enako stopnjo..*

## 5 Testiranje hipoteze na velikih grafih

Za grafe z od 11 do 30 vozlišč smo uporabili metodo naključnega iskanja (angl. *random search*), ki je izhajala iz hipoteze, da maksimalno število geodetskih podpoti dosegajo uravnoteženi dvodelni grafi. Pri vsaki velikosti smo generirali 10 000 naključnih grafov iz tega razreda in izračunali njihovo vrednost  $gpn$ . Končno rešitev je predstavljal graf z največjo vrednostjo  $gpn$ . Za manjše grafe (do 10 vozlišč) je ta pristop že podajal optimalne rezultate, kar je potrjevalo upravičenost izbora dvodelnih grafov.

Za nadaljnjo optimizacijo smo razvili dvostopenjski hibridni postopek, ki je združeval naključno iskanje in simulirano žarjenje (angl. *simulated annealing*, okrajšano SA). Najprej smo z naključnim iskanjem določili kvalitetnega začetnega kandidata, ki je nato postal začetno stanje za SA. Simulirano žarjenje je iterativno spremnjalo graf z osnovnimi operacijami: dodajanjem, odstranitvijo ali zamenjavo ene povezave, pri čemer je ohranilo dvodelnost in povezanost.

Sprememba stanja je bila sprejeta, če je vrednost  $gpn$  narasla; hkrati pa smo s pomočjo verjetnostnega mehanizma, ki je upadal s temperaturo, omogočali tudi prehod v stanje z nižjim  $gpn$ . Ta pristop je preprečeval ujetje v lokalne optimume in omogočal sistematično raziskovanje prostora rešitev. Pri grafih do 20 vozlišč je hibridna metoda dosegla povprečno izboljšanje  $gpn$  za približno 5 % v primerjavi s čistim naključnim iskanjem, kar je dokazovalo njeno učinkovitost.