

Število podpoti pri grafih z danim ciklomatskim številom

*Poročilo projektne naloge
Skupina 18: Maja Košmrlj, Jan Maradin*

Uvod

V tej projektni nalogi sva za dano ciklomatsko število in število vozlišč poiskala grafe z najmanjšim in največjim številom podpoti ter geodetskim številom podpoti, nato pa sva analizirala njihove oblike in lastnosti. Zanimalo naju je predvsem, kako so videti takšni grafi in kako se njihove strukture postopoma spreminjajo z večanjem števila vozlišč ter ciklomatskega števila.

Projektne naloge sva se v bistvu lotila na dva med seboj povezana načina. V prvem delu sva se osredotočila na iskanje metod, s katerimi lahko s pomočjo simuliranega ohlajanja (SA) dobiva čim boljše približke grafov z najmanjšim oziroma največjim številom podpoti. V tem delu sva obravnavala grafe z manjšim številom vozlišč in vsemi možnimi vrednostmi μ . V drugem delu naloge pa sva se usmerila na grafe z manjšim ciklomatskim številom in podrobneje analizirala njihove oblike ter značilnosti.

Teoretično ozadje

- **Graf:** Graf $G = (V, E)$ je sestavljen iz množice vozlišč V in množice povezav E , kjer je vsaka povezava element oblike $\{u, v\}$ za $u, v \in V$.
- **Povezan graf:** Graf G je povezan, če med vsakim parom vozlišč obstaja pot.
- **Pot:** Pot v grafu G dolžine ℓ je zaporedje različnih vozlišč $(v_0, v_1, \dots, v_\ell)$, kjer za vsak $1 \leq i \leq \ell$ velja, da je $v_{i-1}v_i \in E$.
- **Ciklomatsko število:** Je število neodvisnih ciklov, ki jih graf vsebuje, oziroma nam pove koliko povezav bi morali odstraniti, da bi dobili gozd. Ter je definirano kot:

$$\mu(G) = |E| - |V| + 1$$

- **Podpot** Naj bo $G = (V, E)$ povezan graf in $P = v_1v_2 \dots v_k$ pot v grafu G . Podpot grafa G je vsaka zaporedna podzaporedje oglišč te poti, torej vsaka pot oblike

$$v_i v_{i+1} \dots v_j, \quad \text{kjer } 1 \leq i < j \leq k.$$

- **Število podpoti:** Je skupno število vseh enostavnih poti v grafu, vključno s trivialnimi potmi in ga označimo s $p_n(G)$.

- **Geodetsko število** Za $G = (V, E)$ povezan graf označimo geodetsko število podpoti z $gp_n(G)$ in definiramo kot skupno število vseh geodetskih enostavnih poti v grafu. Za vsaka dva vozlišča $u, v \in V(G)$ označimo z $d(u, v)$ razdaljo med njima, z $\sigma^*(u, v)$ pa število vseh enostavnih $u-v$ poti dolžine $d(u, v)$. Tedaj je

$$gp_n(G) = \sum_{u, v \in V(G)} \sigma^*(u, v).$$

Vračunane so tudi poti dolžine 0, torej $u = v$. Geodetsko število podpoti tako meri, koliko najkrajših enostavnih poti vsebuje graf.

- **Kaktusni graf** je graf, v katerem se poljubna dva cikla stikata v največ enem vozlišču. To pomeni, da se cikli lahko dotikajo, vendar se ne prekrivajo.
- **Graf PTC(n,k)** (angl. *Pseudo Triangle Chain*), oziroma psevdotrikotna veriga, je poseben primer kaktusnega grafa z n vozlišči in k cikli. Vsi cikli so trikotniki, povezani v verigo in zato graf dosega največje možno število podpoti med vsemi takšnimi grafi.
- **Izrek** (Knor, 2025) Graf $PTC(n, k)$ enolično maksimizira število podpoti med vsemi kaktusnimi grafi z n vozlišči in $k \geq 2$ cikli.

Simulated annealing- simulirano ohlajanje

Za velike grafe sva tako v 1. delu kot v 2. delu uporabljala Simulated annealing. To je optimizacijski algoritem, ki se uporablja za iskanje približno najboljših rešitev na grafih, zlasti kadar je iskalni prostor zelo velik in kompleksen. Metoda posnema proces fizičnega ohlajanja kovin: na začetku dovoljuje tudi „slabše“ premike, da lahko pobegne iz lokalnih ekstremov, nato pa postopno zmanjšuje verjetnost takšnih premikov, ko se temperatura znižuje. Prednost metode je, da lahko najde dobre rešitve tam, kjer bi druge metode hitro obtičale, saj s svojo naključnostjo bolj učinkovito raziskuje celoten prostor možnih rešitev. Pri vsakem koraku algoritem iz trenutne rešitve x generira novo kandidatno rešitev x' . Če je nova rešitev boljša, jo sprejmemo. V nasprotnem primeru jo sprejmemo z verjetnostjo

$$P = e^{-\frac{\delta}{T}},$$

kjer je δ razlika med novo vrednostjo ciljne funkcije in trenutno najboljšo vrednostjo. Parameter T predstavlja temperaturo, ki se skozi iteracije postopoma zmanjšuje. Temperatura se ohlaja po pravilu

$$T_{k+1} = \alpha T_k,$$

pri čemer sva v tej projektni nalogi izbrala $\alpha = 0.995$. Začetno temperaturo sva po potrebi prilagajala.

to do– če bo čas

- prveri dodatne lastnosti v grafih -zapiši funkcijo ki pregleduje tudi ostale lastnosti v 2.delu, npr.dvodelnost ali kaj takega
- še iz random v 2.delu?

1. del projektne naloge

V tem delu projektne naloge sva za dano število vozlišč in za vsa pripadajoča ciklomatska števila poiskala grafe z najmanjšim številom podpoti, torej se nisva osredotočala na geodetske podpoti.

Ker sva grafe iskala za vsa možna ciklomatska števila, je bil postopek iskanja zelo časovno zahteven, zato sva se omejila na grafe z največ 9 vozlišči. V prvem delu projektne naloge sva želela preveriti, kateri način simuliranega ohlajanja (SA) je najučinkovitejši, zato sva natančno iskanje za majhne grafe in algoritem SA izvajala na isti množici grafov.

Iskanje točnih vrenosti

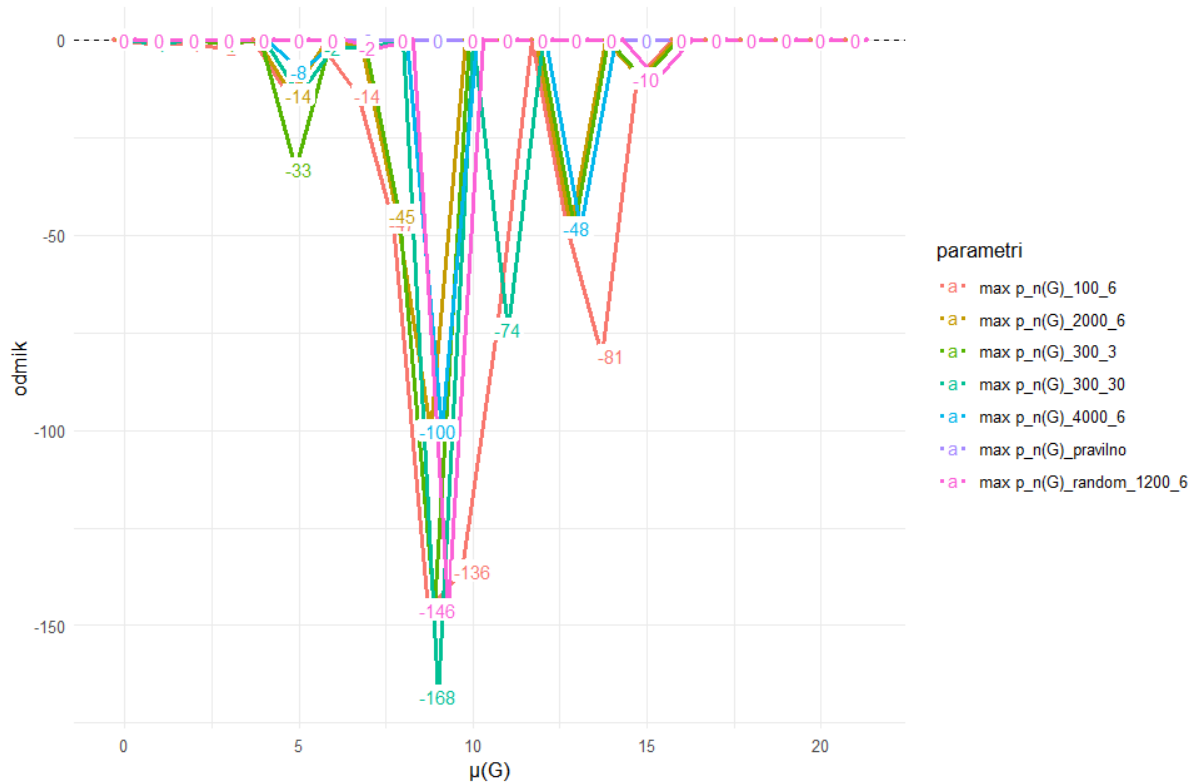
- Koda za ta del je zapisana v datoteki `1del_mali_grafi.ipynb`, rezultati pa so v `1del_mali.csv`
- Najprej sva definirala funkcijo, ki v podanem grafu poišče vse podpoti. To funkcijo sva nadgradila tako, da za dano ciklomatsko število in število vozlišč najprej generira vse možne grafe za ta par, nato pa izbere in shrani tistega z najmanjšim oziroma največjim številom podpoti. To funkcijo sva nato pognala na vseh grafih z največ 9 vozlišči in vsemi možnimi ciklomatskimi števili (to število je navzgor omejeno s polnostjo grafa; na primer pri $|V| = 9$ je maksimalni $\mu = 28$). Rezultate pa sva shranjevala v datoteko CSV.
- Celoten pogon kode je trajal približno 80 minut.

1.del z SA V bistvu naju je zanimalo kako dobro deluje SA, če spremeniva določene parametre. Rezultate pa sva primerjala z točnimi vrednostmi, ki sva jih izračunala v prejšnjem delu.

- Najprej sva se osredotočila na grafe z $n = 9$ in vemi možnimi μ -ji z nastavitvami `max_steps = 300`, `T_start = 3`, `T_end = 0.001`, `cooling = 0.99`. Torej sva v bistvu iskala kateri dober približek bi dala v SA kot začetno vrednost, da bi bila čim večja verjetnost da se algoritem nebi ustavil v lokalnem ekstremu.
- Koda za ta del je zapisana v datoteki `1del_SA_n_9.ipynb`, rezultati pa v `1del_SA_n_9.csv`,
- Začetne približke sva skonstruirala iz točnih vrednosti prejšnjega dela, tako da sva vzela graf z 8 vozlišč in istim ciklomatskim številom, ter mu dodala vozlišče in eno povezavo tako, da je graf ostal povezan. Ko pa sva pognala SA na ciklomatskem številu, ki je večji od 21 (to je maksimum ciklomatsko število za grafe z 8 vozlišči) sva določila da SA vzame približek ki ga je izračunal prej in doda eno "naključno" povezavo tako da se ciklomatsko število za 1 poveča.
- **NAPIŠI UGOTOVITVE ZA $N = 9$**
- Nato sva izvajala SA na grafih z osmimi vozlišči. Tu sva pregledovala kako spreminjanje temperature in števila korakov vpliva na rezultate in čas trajanja algoritma. Primerjala sva kaj je razlika ko je začetna temperatura 30 ali 3, ter kakšne so razlike v rezultatih ko je število korakov 100, 2000 ali 4000. Za konec pa sva pognala kodo

brez "dobrih" začetnih približkov, torej sva kot začetni graf dala v SA naključen graf z izbranim ciklomatskim številom in osmimi vozlišči.

- **opomba** za max je trajalo 8 minut pri 4000
za max $T=30$ $N=300$ — 3,5min
za min $T=6$ $N=2000$ —
- **UGOTOVITVE** za $n=8$



Slika 1: Graf maksimalnih odmikov za $n=8$

2. del projektne naloge

V drugem delu projektne naloge sva se osredotočila na opazovanje oblik grafov. Fiksirala sva ciklomatsko število do $\mu=8$ in izvajala kodo na majhnih grafih do $n=10$, nato pa še simulirano ohlajanje (SA) na grafih z $n=11$ do $n=30$. Nato sva uporabila podani izrek Knorra in preverila, ali drži. Za konec sva tako za majhne grafe kot tudi za velike grafe z metodo SA izvedla analizo oblik tudi za grafe z najmanjšim oziroma največjim številom geodetskih podpoti.

Vsa koda za drugi del je zapisana v mapi 2_del. Za geodetske podpoti pa je znotraj te mape mapa `Geodesic_subpath_number`, kjer so zapisane vse datoteke, povezane z iskanjem grafov z največjim oziroma najmanjšim številom geodetskih podpoti.

Najmanjše/največje število podpoti

Mali grafi

- Tako kot v prvem delu sva najprej napisala kodo za majhne grafe in zanje poiskala točne rešitve. Za ciklomatska števila od $\mu = 1$ do $\mu = 8$ sva za grafe z največ 10 vozlišči generirala vse grafe ter izmed njih izbrala tiste z najmanjšim oziroma največjim številom podpoti.

Koda je zapisana v datoteki `2del_subpath_mali_grafi.ipynb`, rezultati pa v `2del_subpath_mali_grafi.csv`.

Za lažjo analizo oblik grafov sva si vse slike v drugem delu shranjevala v mapi `slike_max` in `slike_min`, kjer ime slike vsebuje par (μ, n) , na katerega se nanaša.

- Da se je koda iztekla hitreje, sva jo prilagodila za uporabo večih jeder.

- **Ugotovitev:**

Pri grafih z majhnim številom vozlišč in majhnim ciklomatskim številom sva ugotovila, da so se pri minimiziranju števila podpoti grafi nagibali k strukturi, sestavljeni iz ciklov dolžine 3, ki so imeli skupno eno vozlišče, kar se je pojavilo v primerih, ko je bilo glede na dano ciklomatsko število na voljo dovolj vozlišč. Ostala vozlišča so tvorila vsaj eno steblo, ki je v večini primerov izhajala iz istega središčnega vozlišča. Pri maksimiziranju števila podpoti sva opazila, da je imel pri dobro skonstruiranih grafih vsakemu vozlišču prirejeno vsaj stopnjo 2. To pomeni, da je bilo vsako vozlišče del nekega cikla. Poleg tega se je pri fiksnem ciklomatskem številu z dodajanjem novih vozlišč maksimalna stopnja grafa zmanjševala. To nakazuje, da se maksimalna stopnja grafa približuje vrednosti 3, minimalna stopnja pa vrednosti 2.

Veliki grafi

- Tudi v drugem delu sva za »večje grafe« uporabila metodo simulated annealing (SA). Za vsa μ od 1 do 8 sva izvedla SA za grafe z $n = 10$ do $n = 30$.
- Začetne približke sva izbrala podobno kot v prvem delu: vzela sva graf z enakim ciklomatskim številom, vendar z enim vozliščem manj, nato pa sva v ta graf dodala še eno vozlišče in eno povezavo, tako da je graf ostal povezan.
- Povezavo sva dodajala na dva različna načina. Najprej sva jo v graf dodala naključno, nato pa sva jo dodala v tisto vozlišče, ki ima največjo stopnjo.
- Koda za oba načina je zapisana v datoteki `2del_subpath_SA.ipynb`. Rezultati za naključno dodajanje povezave so zapisani v `2del_subpath_SA.csv`, za dodajanje v vozlišče z največjo stopnjo pa v `2del_subpath_SA_v_sredisce.csv`.

- **Ugotovitve za oba načina dodajanja povezave:**

Za minimum oba pristopa vrneta enako vrednost najmanjšega števila podpoti, razlikujeta pa se obliki grafov. To je smiselno, saj je število podpoti enako ne glede na to, ali so vsa vozlišča pripeta na eno »sredinsko« vozlišče ali pa so razporejena v verigo; tudi število ciklov ostane enako v obeh primerih. Na primer, pri $\mu = 3$

ima graf z minimalnim številom podpoti tri trikotne cikle, preostala vozlišča pa so razporejena v verigo.

Tudi za maksimum sta pristopa primerljiva: v nekaterih primerih ena metoda vrne nekoliko večje število podpoti, v drugih pa druga. Večjo vrednost navadno vrne pristop, ki najde graf, kjer so nova vozlišča vključena v cikel in ne kot listi, pripeti na eno vozlišče. To obnašanje je posledica stohastične narave metode SA in ga ne moreva povsem odpraviti.

- **problem: apparently nimava vseh slik:** v mapo so shranjene slike, ki jih generira druga funkcija za $\mu \leq 4$, za $\mu = 5, 6, 7, 8$ pa so shranjene slike, ki so nastale z uporabo prve funkcije.

lastnosti: V drugem delu projektne naloge sva prav tako želela preveriti ali Izrek Knor-a zares trdi. V drugem smislu nisva našla smiselne ideje kako bi imelementirata ta izrek v našino projektno nalogo, saj sva midva iskala grafe z največjim in najmanjšim številom podpoti na vseh grafi in ne samo na kakusnih. ZA vse grafe v datoteki `2del_subpath_SA.csv`, sva v datoteki `pregledovanje_lastnosti_subpath.ipynb` Zapisovala funkcije, ki za vsak max in vsak min graf ali je kaktusni in ali je PTC graf. Rezulati so zapisani v datoteki `pregledovanje_lastnosti_subpath.csv` Izrek Knor-a trdi, da v primeru da je graf z največjim številom podpoti kaktusen mora biti tudi PTC. Da sva preverila če ta implikacija res trdi sva zapisala funkcijo, ki nam glede narezultate lastnosti preveri če drži in rezultat je bil da je izrek res

Najmanjše/največje število geodetskim podpoti

Mali grafi

- Za geodetske podpoti sva uporabila isto kodo kot sva jo zapisala v 2.delu za majhne grafe, torej kot je zapisana v `2del_subpath_mali_grafi.ipynb`, na novo sva definirala samo funkcijo, ki išče geodetske podpoti
- koda je zapisana v datoteki `2del_geodesic_mali_garfi.ipynb.ipynb`. Rezultati so zapisani v `2del_geodesic_mali_garfi.csv` Slike pa so shranjene v mapah `slike_geodesic_max` in `slike_geodesic_min`
- kodo sva podgnala za Za ciklomatska števila od $\mu = 1$ do $\mu = 8$ sva za grafe z največ 10 vozlišči

Veliki grafi

- za velike grafe pri iskanju geodetskih poti sva prav tako uporabila isto kodo kot v 2.delu za isanje podpoti, le da sedaj koda išče najmanjše število geodetskih podpoti
- koda je zapisana v datoteki `2del_geodesic_SA.ipynb.ipynb`. Rezultati so zapisani v `2del_geodesic_SA.csv` Slike pa so shranjene v mapah `slike_geodesic_max` in `slike_geodesic_min`

- za geodetske podpote sva začetnim približkom povezala samo randomly in ne v vozlišče z največjo stopnjo.

- **ugotovitve**

Pri minimiziranju geodetskega števila poti v grafih se je pri $\mu \geq 3$ začelo pojavljati, da grafi niso več vsebovali zgolj ciklov dolžine 3, temveč so začeli tvoriti strukture, ki jih lahko opišemo kot nekakšne piramide v prostoru. Pri $\mu = 3$ ima graf eno tristrano piramido in eno centralno vozlišče. Pri $\mu = 4$ se na to centralno vozlišče doda še en cikel dolžine 3, pri $\mu = 5$ pa še en tak cikel, dokler se pri $\mu = 6$ ne ustvari ena štiristrana piramida. Pri nadaljnjem povečevanju ciklomatskega števila se vzorec nadaljuje podobno: najprej nastajajo novi cikli dolžine 3, dokler se ponovno ne oblikuje nova piramidna struktura. Pri maksimiziranju sva ugotovila, da se v odvisnosti od geodetskega števila oblikujejo grafi, v katerih so cikli povezani zaporedno in skupaj tvorijo verigo.