Fakulteta za matematiko in fiziko
Univerza v Ljubljani

Število podpoti pri grafih z danim ciklomatskim številom

Poročilo projektne naloge

Skupina 18: Maja Košmrlj, Jan Maradin

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 $(v_0, v_1, \ldots, v_\ell)$, 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 $P = v_1v_2 \ldots v_k$ pot v grafu G. *Podpot* grafa G je vsaka zaporedna podzaporedje oglišč te poti, torej vsaka pot oblike
$$v_iv_{i+1} \ldots v_j, \quad \text{kjer } 1 \leq i < j \leq k.$$
- **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 vozliščci 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.

Najini polomi

- Zagnala sva iskanje minimuma za $n = 9$ po Janovi metodi z nastavitvami `max_steps = 300`, `T_start = 3`, `T_end = 0.001`, `cooling = 0.99`. Računanje je trajalo približno 32 minut.
  naslednji ko sva pognala je trajalo 41 min (opomba: runnala sta dve funkcije hkrati)
  Problem: ker za $n = 9$ ne začnemo več z dobrim približkom iz $n = 8$, se zgodi, da se pri dodajanju ene povezave ostale povezave ne spremenijo in se vrednost `subpath_number` sploh ne izboljša. Sklepava, da je težava v premajhnem številu korakov, da bi metoda našla boljši rezultat, torej da sva obstala v nekem lokalnem minimumu.

- Pognala sva kodo `vecji_grafi` za $n = 8$, da bova rezultate primerjala z različico `_live`. Izvajala sva jo za minimum in maksimum ter za različno število ponovitev (300 in 4000). Prav tako sva v kodo dodala možnost nastavljanja N za poljuben n, tako da program sam izračuna $\mu_{\text{max, prev}}$ in $\mu_{\text{max, n}}$.
  za max je trajalo 8 minut pri 4000
  za max T=30 N= 300— 3,5min
  za min T= 6 N= 2000 —

to do– če bo čas

- prveri dodatne lastnosti v grafih
- naredi analizo 1. dela –> ali lahko damo v R v tabelo ali v excel?
- SA 1. dela za n = 10
- dodaj 2. del za mu = 9 in mu = 10
- poskusi v SA dodati samo random graf

uvod

V tej projektni nalogi sva za določeno ciklomatsko število in število vozlišč poiskala grafe z najmanjših številom podpoti in geodetskim številom, ter nato iskala kaj te grafe povezuje po izgledu. Ugotavljala sva torej kako izgledajo grafi ter kako se postopoma spremninjajo z večanjem štrevila vozlišč.

Projetne naloge sva se vbistvu lotila na dva različna načina, sicer se prepletata, vendar sva se pri prvem načinu osredotočala kako bi poiskala način da s pomočjo SA dobiva čim boljši približek za graf z manjmanjšim številom podpoti. v 2. delu naloge pa sva se osredotočila na grafe z manjšim ciklomatskim številom in nato bolj gledala oblike grafov.

Za velike grafe sva tako v 1. delu kot v 2. delu uporabljala Simulated aneling. Simulated annealing (simulirano ohlajanje) 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

$$\delta = |f(x') - f(x)|$$

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.

1. del projektne naloge

V tem delu projektne naloge sva za določeno število vozlišč in za vsa mogoča ciklomatska števila pri teh vozliščih poiskala grafe z najmanjšim subpath numberjem.

Ker sva iskala grafe za vsa mogoča ciklomatska števila, je bilo iskanje zelo dolgotrajno, zato sva to izvedla le za grafe z največ 10 vozlišči. Prave grafe z najmanjsim število podpoti sva za male grafe poiskala v datoteki 1del$_{m}ali_{g}rafi.ipynb(dografovzvelikostjo9)Natopasvaznajinimidobri$ 8..10.

V tem delu se nisva ukvarjala z geodetskimi podpotmi.

Postopek dela

Mali grafi

- Prva stvar, ki sva se je lotila, je bila pravilna definicija funkcije, 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 tistega z najmanjšim

oziroma največjim številom podpoti. Vsi dobljeni podatki se avtomatsko zapišejo v datoteko CSV.

- Nato sva to funkcijo 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, ki so zapisani v datoteki `glavno.ipynb`, sva shranjevala v datoteko `rezultati_subpath_live.csv`. V datoteki CSV so shranjeni stolpci s številom vozlišč, vrednostjo μ, najmanjšim $pn(G)$ in imenom grafa, ki to vrednost doseže, ter največjim $pn(G)$ in imenom grafa, ki doseže to vrednost. Celoten pogon kode je trajal približno 80 minut.
- S to funkcijo sva tako za vse možne pare $(\mu, |V|)$ (do $|V| = 9$) poiskala grafe z najmanjšim in največjim številom podpoti.

Veliki grafi

- Za velike grafe pa sva iskala grafe z najmajšim in največjim številom podpoti s pomčjo SA. Torej ni nujno da sva našla najboljše vrednosti, ampak dobre približke.
- Osredotočila sva se, da sva poiskala način uporabe SA in začetnih približkov, ki nama poišče razmerje med dovolj dobrimi približki in hitrostjo kode.
- Najprej sva želela z rekurzijo nareidti SA na grafih z n = 9 in vemi možnimi mu-ji.
- Rekurzija, ki sva jo uporabila vzame graf iz točnih vrednosti, ki sva jih izračunala z majhne grafe. vzela sva grafe z 8 vozlič in istim mu, ter dodala eno povezavo od dodanega vozlišča na poljubno vozlišče da sva ohranila povezan graf. ko pa je mu prevelik (saj je max mu pri n=8 enak 21) SA vzame približek ki ga je izračunal prej in doda eno povezavo tako da se mu za 1 poveča.
- **NAPIŠI UGOTOVITVE ZA N = 9**
- Nato sva izvajala SA za grafe z n = 8. Tu pa sva pregledovala tudi kako spreminjanje temperature in števila korakov vliva na rezultate in čas trajanja.
- Primerjala vsa kaj je razlika ko je začetna temperatura 30 ali 3. potem pa sva še pogledala kaj se spremeni če je število korakov 100, 2000 ali 4000
- **ugotoviitve 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_mal`
  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.
- V tej kodi sva prav tako z uporabo večih jedel povečala uporabo CPU-ja računalika, tako da je celotna koda tekla samo 3 minute
- **Ugotovitev:**

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.