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 $(v_0, v_1, \dots, v_\ell)$, 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
$$v_iv_{i+1} \dots v_j, \quad \text{kjer } 1 \leq i < j \leq k.$$
- **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 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

-

- ko sva probala za n = 9 in N = 4000 izvesti po Janovi metodi smo porabili do mu= 18 je rabil 20 minut kar je očitno čisto preveč počasi, saj bi s tem tempom delal za vse mu pri n = 9 rabil po vsej verjetnosti več ur

- 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.

- Sedaj bova za $n = 9$ uporabila še metodo za manjše grafe: za vsak par (n, μ) poiščeva vse možne povezane grafe, za vsakega izračunava najmanjši `subpath_number` in nato shraniva tistega z najmanjšo vrednostjo (če bodo rezultati še izvedljivi v razumnem času).
  v 80 minutah je funkcija, ki generira vse grafe dokončala izračune za vse grafe do vključno n = 9

  ogotovitev: srednje vrednosti se precej razlikujejo, ampakk ne ostalo je pa obiskuje

- sedaj poganjava kodo še za max, tako da vpisuje v isti csv kot sva že imela rezulatate za min to delava v PRAVILNO2, trajalo je 37 min

- 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 —

- Sedaj bova poskusila kodo pospešiti tako, da jo zaženeva za več vrednosti μ hkrati (da bolje izkoristiva 100% CPU-ja) ter obenem povečava število korakov, tako da bo čas izvajanja ostal približno enak.

Problemi

- ali koda ki ne zapisuje pravilno dela bolje ot tista v pravilno? ali je bilo takrat samo naključje

to do– če bo čas

- poženi za n= 9 še za 2000
- prveri dodatne lastnosti v grafih
- naredi analizo 1. dela –> ali lahko damo v R v tabelo ali v excel?

game plan

- Za $n = 9$ uporabiva rezultate iz `ne_live_8`.
- Napiši funkcijo, ki izračuna minimum in maksimum hkrati za poljuben n.
- Izboljšaj izkoriščenost CPU-ja v tej funkciji.
- Zapiši funkcijo, ki bo rezultate naključnih grafov za $n = 8, 9, 10$ zapisovala v CSV.
- Opraviva analizo podatkov.
- Zapiši funkcijo, ki bo rezultate kaktusnih grafov za $n = 8, 9, 10$ zapisovala v CSV.
- preverit je treba če so slučajno ptc grafi pri majhnih grafih

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.

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.

V tem delu se nisva ukvarjala z geodetskimi podpotmi.

Postopek dela

- 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 določeno ciklomatsko število in število vozlišč (iz teh dveh vrednosti posledično dobimo tudi točno določeno število povezav) izbere tisti graf, ki ima najmanjše oziroma največje število podpoti.
- Nato sva to funkcijo pognala na vseh grafih z največ 9 vozlišči in vsemi možnimi ciklomatskimi števili (to število je seveda navzgor omejeno s polnostjo grafa; npr. pri $|V| = 9$ je maksimalni $\mu = 28$). Rezultate te funkcije, ki so zapisani v datoteki `glavno.ipynb`, sva shranjevala v datoteko `rezultati_subpath_live.csv`. V datoteki CSV so stolpci s številom vozlišč, vrednostjo μ, najmanjšim $pn(G)$ ter imenom grafa, ki to vrednost doseže, ter največjim $pn(G)$ in imenom grafa, ki doseže to vrednost. Pogon celotne kode je trajal 80 minut