1. Uvod

Za lažje razumevanje poročila, sva definirala nekaj manj znanih pojmov.

Naj bo $\tilde{G} = v_1 v_2 \dots v_l$ podgraf grafa G. $\tilde{G}$ pravimo **nit**, če je v_l list in velja $d_G(v_1) = d_G(v_2) = \dots = d_G(v_{l-1}) = 2$.

Oznake:

- $V_{a,b}$ je množica vozlišč grafa s stopnjami $[a, b]$,
- $V_{\leq a}$ ($V_{\geq a}$) je množica vozlišč grafa s stopnjami največ (vsaj) a,
- $V^x_{\leq a}$ ($V^x_{\geq a}$) je množica sosednjih vozlišč vozlišča x s stopnjami največ (vsaj) a,
- $v_{\leq a} = |V_{\leq a}|$ oziroma $v^x_{\leq a} = |V^x_{\leq a}|$.

Definicija. Grafa sta sosednja, če lahko iz enega v drugega prideš tako, da spremeniš 1 povezavo (jo dodaš ali odstraniš).

2. Sigma iregularnost

Mere iregularnosti količinsko opredeljujejo, do kakšne stopnje je graf iregularen. Tako vrnejo vrednost 0 za regularne grafe, večja vrednost pa pomeni, da je graf bolj iregularen. Takih mer je ogromno, med njimi sta tudi naslednji dve:

- sigma iregularnost $\sigma(G)$ je definirana kot
$$\sigma(G) = \sum_{uv \in E(G)} (d_G(u) - d_G(v))^2,$$
- sigma totalna iregularnost je definirana kot
$$\sigma_t(G) = \sum_{\{u,v\} \subset V(G)} (d_G(u) - d_G(v))^2.$$

Razlika med slednjima je, da vsota pri sigma iregularnosti teče skozi vse *povezave*, medtem ko vsota pri sigma totalni iregularnosti teče po vseh parih *vozlišč*.

2.1. Sigma totalna iregularnost vs sigma iregularnosti.

Med zgoraj opisanima merama velja naslednja povezava:

Trditev 2.1. *Za vsako drevo T z n vozlišči velja*
$$\sigma_t(T) \leq (n-2)\sigma(T).$$
Enakost velja tedaj in le tedaj, ko je T pot.

Za dokaz si bomo pomagali z naslednjo lemo:

Lema 2.2. *Naj bo G drevo, x vozlišče stopnje $d \geq 3$, ki je del niti T_1 in T_2. Naj bo $G_0 = G(T_2 \circ T_1)$. Potem velja*

(1) $\sigma_t(G) - \sigma_t(G_0) = 2v_{2,d-1}$,
(2) $\sigma(G) - \sigma(G_0) = 2(d - v^x_{\geq d} - 1)$

Dokaz (Trditve 2.1 z indukcijo). Naj bo $n_1(G)$ število vozlišč grafa G s stopnjo 1.

- $n_1(G) = 0$

  Velja $G \simeq P_1$ in $\sigma(G) = \sigma_t(G) = 0$ ✓

- $n_1(G) = 2$

  Velja $G \simeq P_n$, $\sigma(G) = 2$ in $\sigma_t(G) = (n-2)2$ ✓
- Naj bo $n_1(G) > 2$

  Očitno ima graf G vozlišče x s stopnjo $d \geq 3$, ki je del vsaj dveh niti T_1 in T_2. Naj bo $G_0 = G(T_2 \circ T_1)$. Ker je $n_1(G_0) = n_1(G) - 1$, lahko predpostavimo, da velja

$$\sigma_t(G_0) \leq (n-2)\sigma(G_0). \tag{1}$$

Po Lemi 2.2 velja

$$\sigma_t(G_0) = \sigma_t(G) - 2v_{2,d-1} \quad \text{in} \quad \sigma(G_0) = \sigma(G) - 2(d - v^x_{\geq d} - 1) \tag{2}$$

V (1) vstavimo (2) in dobimo

$$(n-2)\sigma(G) \geq \sigma_t(G) - 2v_{2,d-1} + 2(n-2)(d - v^x_{\geq d} - 1). \tag{3}$$

Ker za vozlišče x velja $d \geq 3$ in je vsebovana v dveh nitih, lahko sklepamo $v^x_{\geq d} + 2 \leq d$ in torej $2(d - v^x_{\geq d} - 1) \geq 2$. Opazimo tudi, da velja $v_{2,d-1} \leq n-3$, zato je $2(n-2)(d - v^x_{\geq d} - 1) > 2(n-3) \geq 2v_{2,d-1}$. Skupaj s (3) nam premislek vrne $\sigma_t(G) \leq (n-2)\sigma(G)$.

□

Opomba. Ker je graf G drevo, velja $n_1(G) \neq 1$. Zato v dokazu uporabimo $n_1(G) = 2$.

Zgornji dokaz in dokaz Leme 2.2 si lahko natančneje ogledate v [1]. Ker nama tak dokaz ni bil dovolj, sva trditev tudi manualno preverila. Spisala sva kodo, ki jo lahko najdete v GitHub repozitoriju pod imenom `preverjanje_trditve`. Zaradi težave s spominom s Sage v CoCalc nama je uspelo preveriti le drevesa, ki imajo največ 8 vozlišč. Zanje se izkaže, da neenakost velja. Enakost pa velja le v primeru, da je drevo pot.

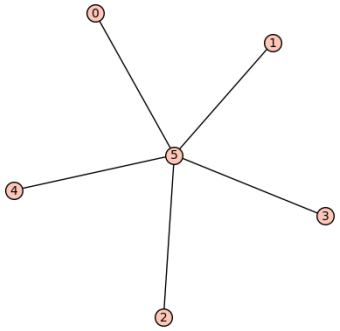
V nadaljevanju najinega raziskovanja sva se najprej osredotočila na majhne grafe. Zanimalo naju je, kako izgleda graf, ki ima maksimalno sigma totalno iregularnost (v nadaljevanju STI). Ob kratkem premisleku sva postavila hipotezo, da je to zvezda. S pomočjo kode imenovane `max_STI` sva hipotezo preverila na vseh grafih, ki imajo od 3 do vključno 7 vozlišč. Rezultati so naju presenetili. Kot sva pričakovala, je

STI z dodajanjem vozlišč naraščala.

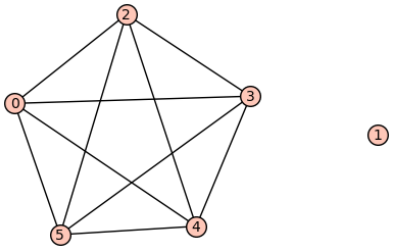
Tabela 1. STI in število vozlišč

Število vozlišč	Maksimalna STI
3	2
4	12
5	36
6	80
7	160

Rezultati so shranjeni tudi v `max_STI` na koncu kot komentar. Opazila sva 2 zelo zanimivi stvari. Maksimalna STI se pri posameznem številu vozlišč pojavi pri dveh različnih grafih. Grafi so do 6 vozlišč enake oblike; zvezda in poln graf, ki ima še dodatno vozlišče s stopnjo 0. Pri 7 vozliščih se stvar nekoliko spremeni. Namesto zvezde dobimo novo obliko grafa, zunaj polnega grafa se pojavita 2 vozlišči. Za boljšo vizualizacijo sva grafe izrisala.

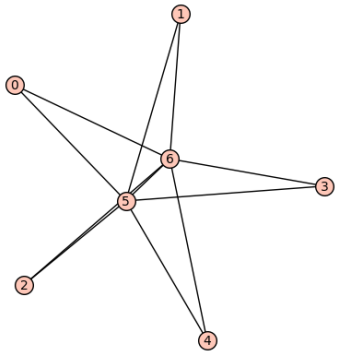


(a) Graf z maksimalno stopnjo 5.

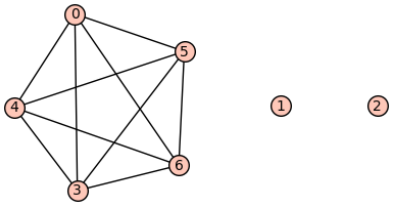


(b) Graf z maksimalno stopnjo 4.

Slika 1. Graf z maksimalno STI na 6 vozliščih.



(a) Graf z maksimalno stopnjo 6.



(b) Graf z maksimalno stopnjo 4.

Slika 2. Graf z maksimalno STI na 7 vozliščih.

Najina hipoteza torej velja le za grafe, ki imajo največ 6 vozlišč.

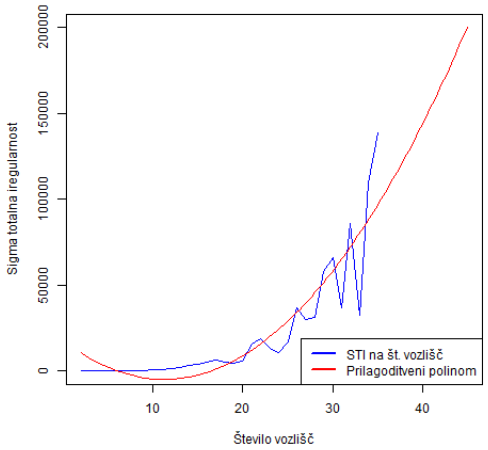
2.2. **Simulated annealing.**

Večjih grafov sva se lotila s pomočjo *simulated annealing*. Vse uporabljene kode, se na GitHub-u nahajajo v mapi `kode`, rezultati testiranj in analiza rezultatov v mapi `rezultati`.

2.2.1. *Splošni grafi.* Simulated annealing za splošne grafe za delovanje uporablja funkcijo `spremeni_n_povezav_povezano_splosno`, ki grafu spremeni n povezav naključno, s pogojem, da mora graf ostati povezan. Sprememba povezave pomeni, da izbere 2 naključni vozlišči in preveri ali je med njima povezava. Če te povezave ni, jo ustvari, če pa povezava obstaja, jo izbriše in preveri, ali je graf še vedno povezan. Če ni, povezavo doda nazaj.

Pri *simulated annealingu* funkcija prvo generira graf prek random Barabasi Albert metode. Za ta graf izračuna še STI. Nato opravimo for zanko z n ponovitvami, kjer je n parameter, ki ga izbere uporabnik. Grafu spremenimo število povezav, katerega program izbere naključno. Nato preverimo, ali je razlika med STI novega grafa in starega grafa dovolj velika. Če je, novi graf postane naš trenutni 'odgovor' za globalni maksimum. Ob vsaki ponovitvi sistem ohladimo s stopnjo 0.9. Funkcija vrne STI, ki predstavlja globalni maksimum, in seznam stopenj grafa s to STI. Zakomentirani del kode lahko po želji ta graf še izriše, lahko pa tudi izpiše zgodovino, torej vse grafe, ki jih je preverjala.

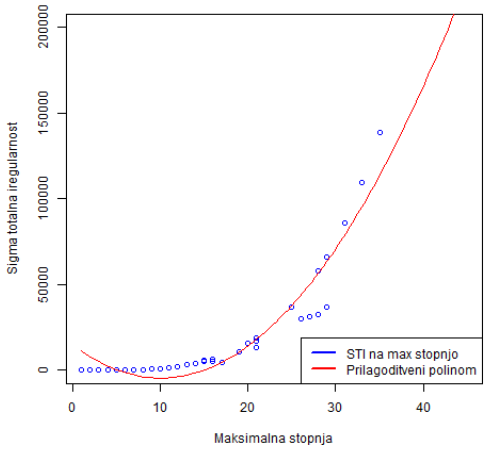
Kodo sva lahko stestirala na 35 vozliščih. Shranjena je pod imenom `splosni_grafi`. Kot pričakovano se je z večajnem števila vozlišč STI večala. Toda to velja le do 17 vozlišč. Ob nadaljnjem večajnem števila vozlišč opazimo, da se ponekod STI zmanjša. Za boljšo vizualizacijo sva narisala graf, ki prikazuje večanje STI z dodajanjem vozlišč.



Slika 3. Naraščanje STI glede na število vozlišč rpi splošnih grafih.

Dodala sva še prilagoditveno premico, ki prikazuje, kam morda narašča STI v neskončnosti. Glede na najine podatke lahko edino sklepava, da STI z dodajanjem vozlišč kvečjemu narašča v neskončnost. Sklep sva potrdila tudi, ko sva primerjala največji STI pri maksimalni stopnji na posameznem vozlišču, kar lahko vidimo na Sliki 4. Opazila sva tudi, da je minimalna stopnja pri največji STI pri posameznem

številu vozlišč dokaj konstantna. Večinoma je 1, največjo minimalno stopnjo pa dosežemo pri 27 in 28 vozliščih, kjer je minimalna stopnja 4.



Slika 4. Največja STI glede na maksimalno stopnjo pri splošnih grafih.

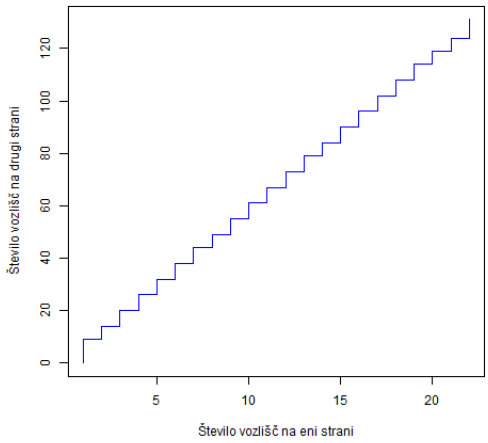
Sklep: splošnega grafa z maksimalno STI ni. Seveda bi sklep lahko dodatno podkrepila, če bi testirala na še večjem številu vozlišč. Žal zaradi Sageo-vih težav s spominom to ni možno.

2.2.2. *Dvodelni grafi.* Koda za dvodelne grafe je shranjena pod imenom `dvodelni_grafi`. Pri dvodelnih grafih funkcija za računanje STI deluje na isti princip kot funkcija pri `splosni_grafi`. Podamo graf, funkcija za vsako vozlišče izračuna kvadrat razlike v stopnji vozlišča in vseh nadaljnih vozlišč. Te vrednosti dodaja totalni sigmi. Na koncu vrne STI grafa.

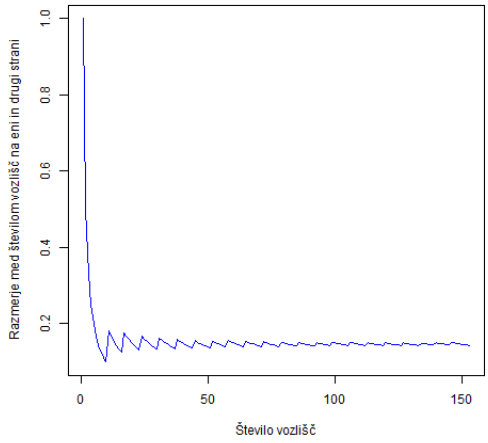
Frunkcija za dvodelne grafe s *simulated annealing* nam najprej generira naključen dvodelni graf, na podlagi naključne delitve števil vozlišč in naključnega števila med 0 in 1, ki je izbran iz enakomerne porazdelitve. Na vsakem koraku `for` zanke nam `proposal` zgenerira 2 števili; delitev (koliko vozlišč naj je na vsakem delu) in faktor, s katerim bo zgeneriral naslednji graf. Ko graf zgenerira, preveri, ali je razlika STI med trenutnim grafom in pravkar zgeneriranim grafom dovoj velika. Če je, zgenerirani graf postavimo na trenutnega. Sistem na vsakem koraku ohladimo s faktorjem 0.9, enako kot pri splošnih grafih. Na koncu funkcija izpiše število vozlišč na vsakem delu in kvocient manjšega dela s številom vseh vozlišč, saj nas zanima, ali se to razmerje z večanjem vozlišč čemu bliža.

Pri dvodelnih grafih sva namesto naraščanja STI gledala razmerje med številom vozlišč na eni in drugi strani. Želela sva najti nek vzorec, za kar bi lahko posplošila na še večje grafe, kot na katerih sva testirala. Sage nama je pri dvodelnih grafih 'dovolil' testirati kar na 153 vozliščih. Najprej sva si ogledala, kako narašča število vozlišč na eni in na drugi strani. Rezultat je kar precej zanimiv stopničast graf.

V povprečju je 'zlato razmerje' 0,156813146. Razmerje, kjer doseže maksimalno vrednost je (seveda) pri samo 1 vozlišču (kjer je vrednost 1), minimalno razmerje pa pri 10 vozliščih, kjer je 9 vozlišč na eni in samo 1 vozlišče na drugi strani (razmerje je torej 0.1). Graf, ki prikazuje spreminjanje razmerja z dodajanjem vozlišč je spodaj.



Slika 5. Naraščanje števila vozlišč na eni in drugi strani pri dvodelnih grafih.



Slika 6. Razmerje med vozlišči na eni in drugi strani.

Iz grafa se zdi, da se z dodajanjem vozlišč iskano razmerje nekako stabilizira, iz podatkov lahko razberemo, da bo to število verjetno ležalo nekje med 0.145 in 0.15.

Sklep: razmerje, pri katerem dvodelni grafi v povprečju dosežejo maksimalno STI, je nekje med 0.145 in 0.15.

2.2.3. *Grafi s fiksno maksimalno stopnjo.* Za *simulated annealing* pri grafih s fiksno maksimalno stopnjo potrebujemo nekaj več funkcij. Shranjene so pod datoteko, imenovano `fix_grafi`.

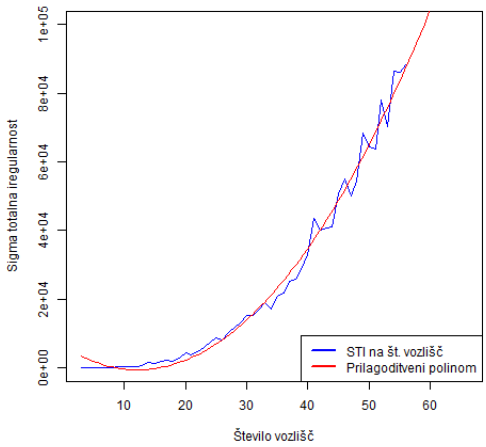
Funkcija `generiraj_nakljucni_graf_z_maks_stopnjo` potrebujemo, da prvič zgeneriramo ustrezen graf. Deluje tako, da najprej zgenerira naključen graf, pridobljen z metodo random Barabasi Albert. Če ugotovi, da ima tako dobljen graf večjo maksimalno stopnjo, kot jo želimo imeti na koncu, postopoma odstranjuje povezave in zagotovi, da je graf še vedno povezan. Če pa je maksimalna stopnja manjša od željene, postopoma varno dodaja povezave med naključno izbranima vozliščema,

kjer ponovno preverja, da maksimalna stopnja ne preseže željene. Funkcija nam vrne prvi graf, kjer maksimalna stopnja vozlišč doseže željeno.

Funkcija `sprmeni_n_povezav_varno` grafu s fiksno največjo stopnjo spremeni n povezav na način, ki ohranja največjo stopnjo in povezanost grafa. Deluje na enak princip kot funkcija `spremeni_n_povezav_povezano_splosno`.

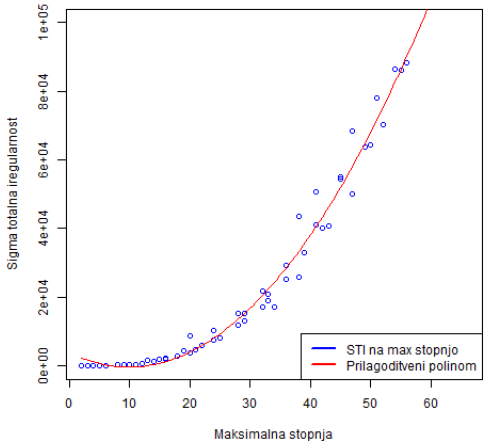
Simulated annealing pri grafih s fiksno maksimalno stopnjo na začetku generira ustrezen graf s funkcijo `generiraj_nakljucni_graf_z_maks_stopnjo`. Deluje na enak način kot funkciji za *Simulated annealing* pri splošnih in dvodelnih grafih.

Kodo sva lahko stestirala na največ 56 vozliščih. Tudi pri tej vrsti grafa se zdi, da STI z dodajanjem vozlišč narašča preko vsake meje, kar lahko vidimo tudi na spodnjem grafu.



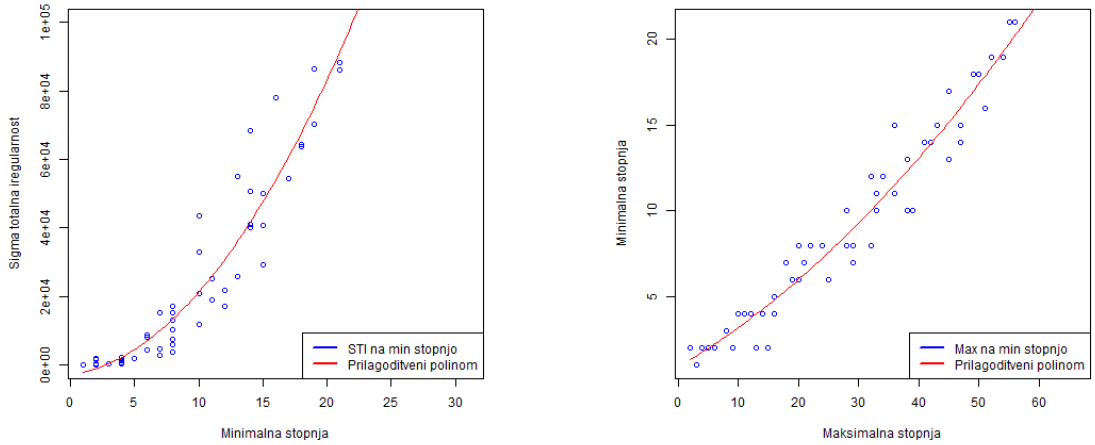
Slika 7. Naraščanje STI glede na število vozlišč pri 'fiksnih' grafih.

Tudi pri tem tipu grafa je zelo zanimivo, da se maksmalna STI z dodajanjem vozlišč včasih tudi pade. Raziskala sva tudi razmerje med maksimalno stopnjo grafa in STI.



Slika 8. Največja STI glede na maksimalno stopnjo pri 'fiksnih' grafih.

Opazila sva, da je v povprečju največja STI pri maksimalni stopnji, ki je za 1 manjša od števila vozlišč. Zanimivo se nama je zdelo tudi naraščanje minimalne stopnje z dodajanjem vozlišč. Ugotovila sva, da število narašča skoraj linearno z naraščanjem maksimalne stopnje. Kljub temu maksimalna stopnja narašča hitreje, kot minimalna. To se vidi že s samim naklonom prilagoditvenega polinoma.



(A) Največja STI glede na minimalno stopnjo pri 'fiksnih' grafih.

(B) Naraščanje minimalne glede na maksimalno stopnjo.

Slika 9. STI glede na minimalno stopnjo.

Sklep: tudi pri grafih s fiksno maksimalno stopnjo STI narašča preko vsake mere. Minimalna stopnja narašča z dodajanjem vozlišč (in posledični večanjem maksimalne stopnje) skoraj linearno.

Literatura

[1] R. Dimitrov, D. in Škrekovski. Comparing the irregularity and the total irregularity of graphs. *Ars Mathematica Contemporanea*, 9:45–50, 01 2015.