UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika

Žiga Mazej

Sigma totalna iregularnost dvodelnih grafov

Skupinski projekt

Poročilo

Mentorja: doc. dr. Janoš Vidali,
prof. dr. Riste Škrekovski

Domžale, 2024

1. Navodilo naloge

Želimo najti dvodelne grafe reda n z največjo možno sigma totalno iregularnostjo. Da bi dosegli to, naredimo naslednje:

(1) Prvič, za majhne vrednosti n poiščimo optimalne grafe z uporabo sistematičnega iskanja.
(2) Drugič, poskusimo posplošiti svoje ugotovitve za večje n in jih podrobno preizkusimo.
(3) Navedimo natančno izjavo o tem, kateri graf/i je/so optimalen/ni, in preizkusimo svojo domnevo s spreminjanjem kandidatov, tj., vedno bi morali dobiti graf z manjšo sigma totalno iregularnostjo. Tu lahko uporabimo nekaj metahevristike.

2. Opis problema

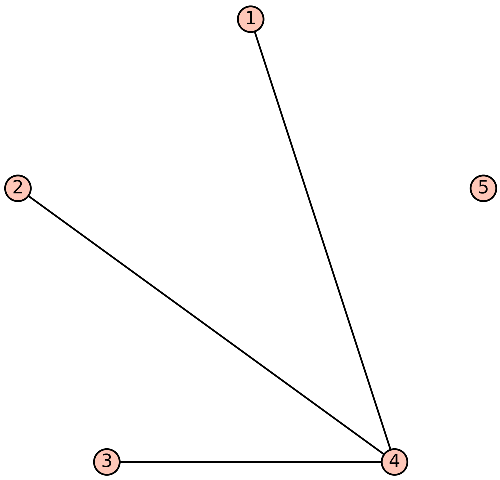
Skušam najti dvodelni graf z maksimalno sigma totalno iregularnostjo (v nadaljevanju STI), glede na red n, ki jo definira enačba

$$\sigma_t(G) = \sum_{\{u,v\}\subseteq V(G)} (d_G(u) - d_G(v))^2. \tag{1}$$

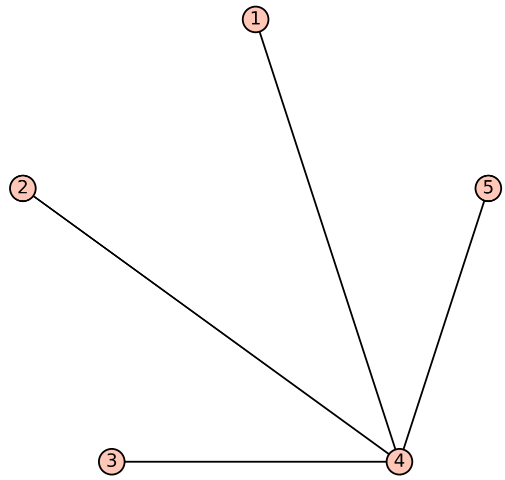
Problema se bom lotil s simatičnim iskanjem za majhne n, nadaljeval za večje n, na koncu pa bom poskušal najti čim bolj optimalne grafe za velike n z metahevristiko, oziroma z metodo simulated annealing.

3. Potek Dela

3.1. Sistematično iskanje. Najprej sem iskal STI za dvodelne grafe nizkega reda. Red 2 je trivialen, prav tako tudi vsi ostali regularni grafi, kar je očitno iz same definicije regularnosti. Nasledjnih nekaj redov se je dalo hitro preveriti ročno oziroma sistematično. Izračuni STI za te grafe so bili izvedeni ročno po formuli, saj so bile številke še relativno majhne. Izmed teh sem izbral tiste, katerih vrednosti STI bodo največje in poskušal iskati nek vzorec ter algoritem, ki bi deloval na večjih redih n. Izkaže se, da bo STI vedno večja pri polnih grafih



(A) Graf reda 5, ki ni poln



(B) Poln graf reda 5

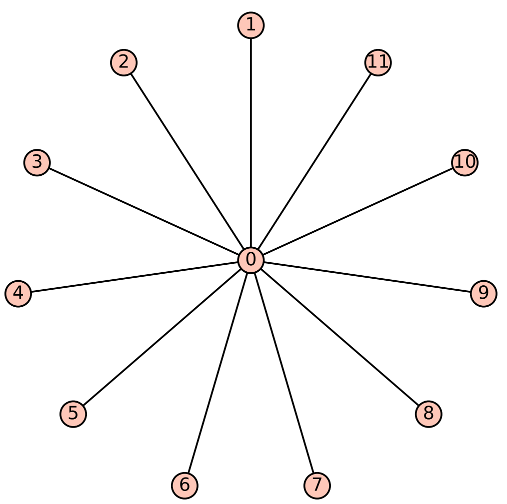
STI grafa A = 24, STI grafa B = 36,

kot se vidi na tem primeru. To pa lahko potrdimo tudi z izračunom za splošen n

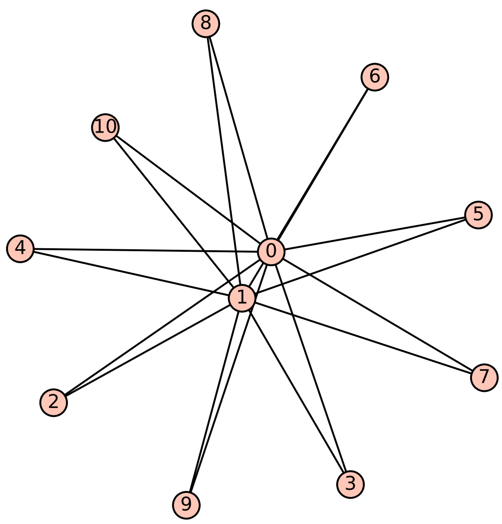
$$\begin{aligned}
(d_cB - d_zB)^2(n-1) &> (d_cA - d_zA)^2(n-2) + (d_cA - d_nA)^2 \\
(n-2)^2(n-1) &> (n-3)^2(n-2) + (n-2)^2 \\
(n-2)(n-1) &> (n-3)^2 + (n-2) \\
n^2 - 3n + 2 &> n^2 - 5n + 7 \\
-3n + 2 &> -5n + 7 \\
-3n &> -5n + 5 \\
2n &> 5 \\
n &> \frac{5}{2}.
\end{aligned}$$

Kjer d_c predstavlja stopnjo centra, d_z stopnjo zunanjega povezanega vozlišča, d_n pa stopnjo nepovezanega vozlišča. Podoben izračun bi uporabili za več nepovezanih vozlišč.

Naslednja hipoteza je bila, da bi lahko bili iskani grafi z največjo STI zvezdni grafi. To pride iz formule za STI, ki vzame kvadrat razlik med stopnjami vozlišč, pri zvezdnih pa so te razlike največje. Po nadaljnem premisleku in preizkušanju se to hipotezo lahko zavrne že pri grafih reda 11, kjer je optimalna rešitev "zvezdni graf z dvema vozliščema v centru".



(A) STI = 810



(B) STI = 882

Prehod iz enega vozlišča v centru na dva

Tu sem opazil trend dodajanja vozlišč v "center" na vsake nekaj dodanih vozlišč v celotni graf. Na podlagi te ideje, sem sestavil algoritem v okolju Sage Worksheet, ki je shranjen pod imenom *prvi_algoritem*, ki za izbran red grafa pregleda vse možne "zvezdne grafe z več vozlišči v centru", ki so smiselni. Ta algoritem je deloval za grafe do reda 100, potem pa zaradi omejitev Cocalc-a ni bil več zmožen procesirati. Rezultati so sledeči

Vozlišča	**Sigma Totalna Iregularnost**	**Razmerje**
3	2	0.333
4	12	0.25
5	36	0.2
6	80	0.167
7	150	0.143
8	252	0.125
9	392	0.111
10	576	0.1
20	9996	0.15
30	50336	0.133
40	159936	0.15
50	390096	0.14
60	809676	0.15
70	1500000	0.143
80	2558976	0.15
90	4100096	0.144
100	6247500	0.15

Tabela 1. Rezultati prvega algoritma.

Tu sem začel opazovati tudi razmerje med vozlišči v "centru" in vsemi vozlišči, ki nam bo v prihodnje povedal več o vrsti in obliki grafov, ki imajo največjo STI, kot dejanska STI, saj je ta že pri grafih reda 100 ogromna. Opazimo lahko, da to razmerje pada do reda 10, kjer je najmanjše, in sicer 0.1, potem pa se za večje redove giblje med 0.14 in 0.15.

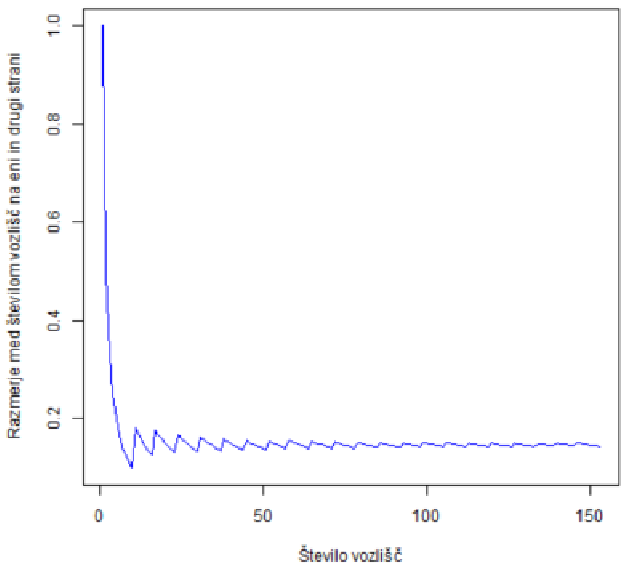
V nadaljnjem optimiziranju algoritma sem prišel do sklepa, da obravnavanje vozlišč glede na "centralna" in zunanja ni več smiselno, bolje bi jih bilo zgolj razdeliti na dve množici, kjer je prva, ki predstavlja prej imenovana "centralna" proporcionalno manjša od druge. Opazimo, da se ob povečanju reda za 1, število vozlišč v eni množici poveča za 1, v drugi pa ostane enako, nikoli se torej število vozlišč v eni množici s povečanjem reda ne zmanjša. Ta ugotovitev mi je omogočila, da sem sestavil bolj efektiven algoritem, ki se nahaja v datoteki z imenom *drugi_algoritem*. Tu za graf reda n algoritem gleda zgolj graf reda $n-1$ z največjo STI. Iz tega grafa naredi dva grafa, kjer enemu od njiju doda eno vozlišče v eno množico, drugemo pa v drugo množico, in v obeh poveže to vozlišče z vsemi iz nasprotne množice (kar sledi iz že prej dokazanih ugotovitev). Za oba izračuna STI, pogleda, pri katerem je večja in tega vzame kot optimalnega in kot bazo za red $n+1$ in tako dalje.

Ta algoritem deluje zelo efektivno in pri zmožnostih procesiranja Cocalc-a izračuna največjo STI za grafe do reda 550. Rezultati so sledeči

Vozlišča	Sigma Totalna Iregularnost	Število Vozlišč Nabora 2	Razmerje
50	390096	7	0.14
100	6247500	15	0.15
150	31640576	22	0.1467
200	99993276	29	0.145
250	244121856	37	0.148
300	506249216	44	0.1467
350	937874496	51	0.1457
400	1599943356	59	0.1475
450	2562886656	66	0.1467
500	3906225036	73	0.146
550	3906225036	81	0.1473

Ugotovimo, da se razmerje za velike n giblje med 0.145 in 0.1475, Torej bi lahko ocenili

$$\lim_{n\to\infty} (\text{razmerje}) \approx 0.1464$$



Slika 3. Razmerje med vozlišči v manjšem naboru in med vsemi vozlišči

3.2. **Simulated annealing.** Domnevo bom podprl z metahevristiko. Metodo, ki je uporabna za moj problem imenujemo Simulated annealing. Koda za le-to je v datoteki *algoritem_sa*. Funkcija 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 bo na vsakem delu) in faktor, s katerim bo zgeneriral naslednji graf. Ko ta graf zgenerira, preveri, da je razlika STI med prvotnim grafom in pravkar zgeneriranim grafom dovoj velika. Če je, potem zgenerirani graf postavimo na trenutnega. Sistem na vsakem koraku ohladimo s faktorjem 0.9 (ta je najbolj smiselen). Na koncu funkcija izpiše število vozlišč na vsakem delu in razmerje med manjšim delom in številom vseh vozlišč, saj nas zanima, ali se to razmerje z večanjem števila vozlišč čemu bliža. S funkijo *test_dvodelni_SA* dobimo sledeče rezultate

Vozlišča	**Nabor 1**	**Nabor 2**	**Razmerje**	**Sigma totalna iregularnost**
10	1	9	0.1	576
20	3	17	0.15	9996
30	20	10	0.3333	20000
40	21	19	0.475	12204
50	38	12	0.24	308256
60	9	51	0.15	809676
70	60	10	0.143	1400000
80	4	76	0.05	1575936
90	70	20	0.2222	3500000
100	15	85	0.15	6247500
110	16	94	0.1455	7341930
120	18	102	0.15	12954816
130	19	111	0.1462	17850576
140	97	43	0.3071	12162636
150	22	128	0.1467	31640576

Kot opazimo, so odstopanja od pričakovanj pri določenih grafih kar velika. Prav tako pa so v nekaterih primerih zelo nesmiselna. Tak primer sta grafa reda 30 in 40, kjer ima prvi večjo STI kot drugi, kar je zagotovo narobe. To se zgodi, saj lahko ta algoritem z neko verjetnostjo na vsakem koraku sprejme neko slabšo rešitev, kar je pogoj za delovanje in napredovanje skozi boljše možnosti pri tem metahevrističnem postopku.

Temu problemu se izognemo s for zanko in shranjevanjem najboljše rešitve do tistega koraka zanke. Tako algoritem skozi več poskusov z večjo verjetnostjo vrne boljšo rešitev in se tako izogne vračanju slabih rešitev, kot se je zgodilo v določenih primerih zgoraj. Ta popravek je izveden v funkciji *test_dvodelni_SA_natancen*, katere rezultati so sledeči

Vozlišča	**Nabor 1**	**Nabor 2**	**Razmerje**	**Sigma totalna iregularnost**
10	1	9	0.1	576
20	3	17	0.15	9996
30	26	4	0.1333	50336
40	34	6	0.15	159936
50	43	7	0.14	390096
60	9	51	0.15	809676
70	10	60	0.143	1500000
80	12	68	0.15	2558976
90	13	77	0.1444	4100096
100	15	85	0.15	6247500
110	16	94	0.1455	7341930
120	18	102	0.15	12954816
130	19	111	0.1462	17850576
140	21	119	0.15	24000396
150	22	128	0.1467	31640576
200	29	181	0.145	99993276

Ta funkcija dokaj časovno učinkovito deluje za grafe do reda 200, brez omejitev Cocalc-a pa bi tudi za večje. Rezultati so pričakovani, in potrdijo tiste iz sistematičnega iskanja največje STI. Torej

$$\lim_{n\to\infty} (\text{razmerje}) \approx 0.1464$$

4. KODA

Komentirana koda je dostopna v repozitoriju v mapi kode.