UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika

Jaša Pavčič, Liza Logar

Graovac-Ghorbani indeks

Skupinski projekt

Poročilo

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

Ljubljana, januar 2024

1. Navodilo naloge

Graovac-Ghorbani indeks $ABC_{GG}(G)$ povezanega grafa G z n vozlišči je dan z $ABC_{GG}(G) = \sum_{u,v \in E(G)} \sqrt{\frac{\nu_{u,v}+\nu_{v,u}-2}{\nu_{u,v}\nu_{v,u}}}$, kjer je $n_{u,v}$ število vozlišč, ki so bližje vozlišču u kot vozlišču v in $n_{v,u}$ je število vozlišč, ki so bližje v kot vozlišču u. Naš cilj je analizirati strukturo grafov na fiksnem številu vozlišč, ki imajo najmanjši oziroma največji možen Graovac-Ghorbanijev (G-G) indeks. Nato enak problem obravnavamo še na dvodelnih grafih, drevesih in grafih brez trikotnikov. Za manjše grafe bomo rezultat izračunali natančno, pri večjih pa bomo izračunali približek s pomočjo metahevristike. Za programiranje bomo uporabili okolje Sage.

2. Reševanje problema

Na začetku sva se lotila računanja G-G indeksa za manjše grafe, da bi dobila občutek, kateri grafi imajo večji in kateri manjši G-G indeks. Izračunala sva G-G indeks za vse možne (majhne) grafe danega tipa na fiksnem številu vozlišč, nato pa sva pri reševanju problema za večje grafe sva uporabila metahevristično metodo simuliranega ohlajanja. Na koncu sva eksperimentirala s parametri, ki jih sprejme metahevristika, da bi videla, pri katerih lahko dobiva najboljši rezultat. Nato sva za vsak tip grafa ustvarila svojo tabelo in v njih dokumentirala rezultate ugotovitev.

2.1. Prvi del: majhni grafi

V prvem delu naloge sva hotela izračunati vrednosti G-G indeksa za vse možne grafe danega tipa na fiksnem številu vozlišč. To sva storila s funkcijo `GGI_na_fiksnem_st_vozl(n, tip_grafa)`, ki sprejme število vozlišč n in tip grafa (običajen povezan (*op*), brez trikotnikov (*bt*), drevo (*dr*), dvodelen (*dv*)) in vrne seznam 'tuplov', kjer je prvi element vsakega tupla graf ustreznega tipa na n vozliščih, drugi pa pripadajoč G-G indeks. Seznam je urejen naraščajoče po indeksu. Na koncu za vsak element iz urejenega seznama grafično prikaževa graf in pripadajoč G-G indeks.

2.2. Drugi del: simulirano ohlajanje

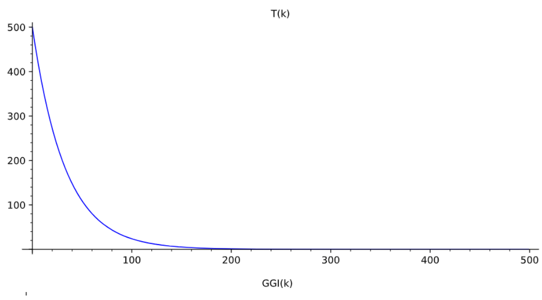
Za velike grafe nisva več računala vrednosti G-G indeksa za vse možne grafe na fiksnem številu vozlišč danega tipa, saj je to preveč računsko zahtevno. Namesto tega sva uporabila metahevristično metodo simuliranega ohlajanja, ki je generirala ustrezne 'sosede' danega grafa in računala njihove G-G indekse, dokler funkcija ni pretekla vseh korakov. Sosednja stanja sva dobila s funkcijo `neighbour(G, tip_grafa)`, ki do sosednjega grafa pride tako, da dodaja oziroma odstranjuje naključne povezave. Funkcija poskrbi, da je nov graf še vedno povezan, istega tipa in da ne doda povezave, kjer ta že obstaja.

2.2.1. *Funkcija neighbour za običajne povezane grafe.* Funkcija je zastavljena tako, da načeloma z verjetnostjo 0.5 doda naključno povezavo, z verjetnostjo 0.5 pa odstrani naključno povezavo. V primeru, da je graf drevo, ne moremo nobene povezave odstraniti, saj bi tako dobili nepovezan graf. V tem primeru povezavo dodamo. Če pa je graf poln, moramo eno povezavo odstraniti.

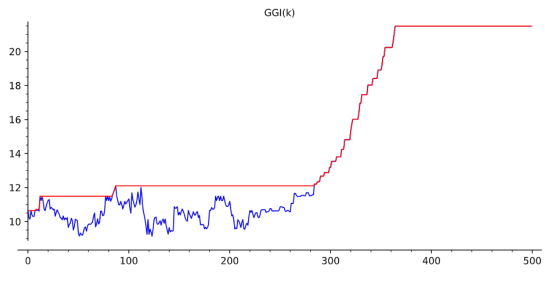
2.2.2. *Funkcija neighbour za drevesa.* V tem primeru funkcija neighbour deluje tako, da najprej izbere naključno povezavo in jo odstrani. Dobimo dva grafa, ki sta med sabo nepovezana. Nato izberemo eno naključno vozlišče iz prvega in eno iz drugega grafa in ju povežemo. Dobimo graf, ki je zagotovo drevo in je zagotovo povezan.

2.2.3. *Funkcija neighbour za dvodelne grafe.* V dvodelnih grafih lahko vozlišča razdelimo v dve skupini in dodajamo oziroma odstranjujemo povezave samo med vozlišči iz nasprotnih skupin. Funkcija posebej obravnava primer, ko imamo v eni skupini samo eno vozlišče. V tem primeru brez spremembe skupin vozlišč ne moremo nobene povezave odstraniti (saj bi bil graf nepovezan) ali dodati (saj je graf že poln). V ostalih primerih pa bodisi nekoliko spremenimo skupini vozlišč, bodisi dodamo oziroma odstranimo povezavo. Na ta način dobimo zelo raznolike sosednje grafe, kar nam vrne boljše rezultate optimizacije.

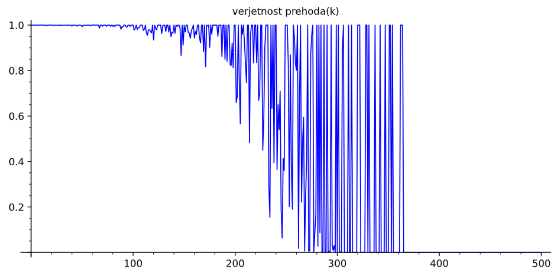
2.2.4. *Funkcija neighbour za grafe brez trikotnikov.* V tem primeru pa funkcija najprej z verjetnostjo 0.5 doda, z verjetnostjo 0.5 pa odstrani naključno povezavo. Nato graf 'popravlja' z odstranjevanjem in dodajanjem povezav, dokler nista izpolnjena oba potrebna pogoja: da je graf povezan in da je brez trikotnikov.
Funkcija simuliranega ohlajanja
`simulirano_ohlajanje(G_0, k_max, T_0, a, tip_grafa, kaj_iscem, show_plot=True)` sprejme začetni graf (G_0), največje število korakov (k_{max}), začetno temperaturo (T_0), parameter ohlajanja (a), tip grafa (tip_grafa) in podatek o tem, ali iščem minimum ali maksimum (kaj_iscem). Poleg tega sprejme tudi parameter $show\_plot$, ki določi, ali rezultate tudi grafično predstavimo ali ne. Vrne končni graf, torej graf, za katerega je funkcija izračunala, da ima največji oziroma najmanjši G-G indeks. Primer pomožnih grafov, ki jih dobimo s klicem funkcije `simulirano_ohlajanje(graphs.RandomBipartite(6, 7, p = 0.5), 500, 500, 0.97, 'dv', 'max')`:



Slika 1. Funkcija temperature T v odvisnosti od števila korakov k



Slika 2. Funkcija GGI v odvisnosti od števila korakov k



Slika 3. Funkcija P (verjetnost prehoda) v odvisnosti od števila korakov k

2.3. Tretji del: izbira parametrov in eksperimentiranje

2.3.1. *Eksperimentiranje.* Definirala sva sezname možnih vrednosti parametrov iz katerih se ustvarijo vse možne kombinacije teh parametrov. Nato se z zanko sprehodiva skozi vse možne kombinacije, ter za vsako iteracijo posodobiva vrednost oz. ekstrem glede na parameter kaj_iscem. To naredimo za graf brez trikotnika, dvodeln graf, obicajen povezan graf in drevo. S tem sva si pomagala ustvariti sliko o tem kateri parametri so primerni za izbiro in kako približno izgleda struktura grafa z G-G indeksom zelo majhnim in G-G indeksom zelo velikim.

2.3.2. *Izbira.* S skrbnim postopkom prehajamo skozi razpon parametrov, ki sega od največjega števila iteracij (k_{max}) do začetne temperature (T_0) in stopnje hlajenja (a). Za vsako kombinacijo se izvede simulirano hlajenje na grafih različnih velikosti, pri čemer je cilj optimizacija G-G indeksa. Pri iskanju zelo malega in zelo velikega indeksa na 8 vozliščih, je za spisano funkcijo simuliranega ohlajanja dovolj vzeti $T_0 = 1000$, $a \in (0.5 - 0.9)$ in $k_{max} \in [50 - 500]$. Najmanj časa program porabi za implementacijo algoritma na drevesih in na dvodelnih grafih. Največ časa pa pri grafih brez trikotnika in obicajnih povezanih grafih.

3. Rezultati

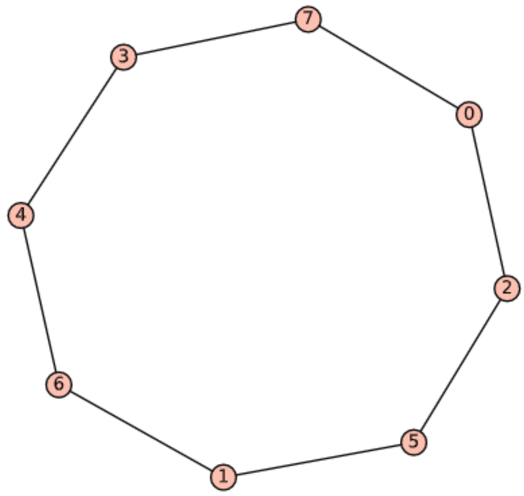
3.1. Vpliv parametrov

Vpliv parametrov je izrazito odvisen od največjega števila iteracij (k_{max}), kar globoko vpliva na konvergenco in končni G-G indeks. Začetna temperatura (T_0) se pojavi kot ključni dejavnik, ki vpliva na raziskovanje prostora rešitev, medtem ko stopnja hlajenja (a) usmerja hitrost konvergence in kakovost rešitve. Zanimivo je, da optimalni parametri kažejo tako skupne značilnosti kot tudi razhajanja med različnimi vrstami grafov, kar poudarja potrebo po prilagojenih parametrizacijah glede na značilnosti posameznega grafa. Za običajen povezan graf smo opazili, da za premajhen parameter k_{max} funkcija ne najde vedno minimuma. Pri drevesih pa, je za iskanje maksimuma potreboval algoritem največ korakov ($k_{max} = 500$)

3.2. Hipoteza in ugotovitve

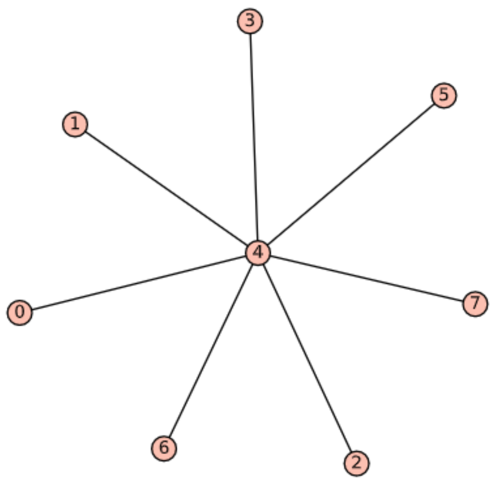
3.2.1. *Običajen povezan graf.* Pri običajnem povezanem grafu na n vozliščih iz eksperimenta sklepava in postavljava hipotezo , da je struktura grafa z najmanjših G-G indeksom graf z n povezavami in da za vozlišča velja, da je $d(v) = 2$ kar pomeni, da tak graf predstavlja n kotnik. Maksimalen G-G indeks pa doseže struktura grafa za kjer za vozlišča velja, $d(v) = n - 2$.

3.2.2. *Graf brez trikotnika.* Pri grafih, ki ne vsebujejo trikotnika se izkaže, da je primer zelo podoben problemu z običajnim povezanim grafom iz vidika majhnega G-G indeksa. Če ne bi zahtevali, da je graf povezan bi za rešitev problema dobili pot na n vozliščih. Kar se kasneje izkaže kot struktura, ki spada v drevesne grafe z zelo majhnim G-G indeksom. Če iščemo veliki G-G indeks, opazimo, da za vsako vozlišče velja, da je $d(v) = 4$ ampak ima ta graf skoraj polovico nižji G-G indeks na 8 vozliščih v primerjavi z običajnim povezanim grafom ($GGI(op) = 16.97$ in $GGI(bt) = 9.79$)



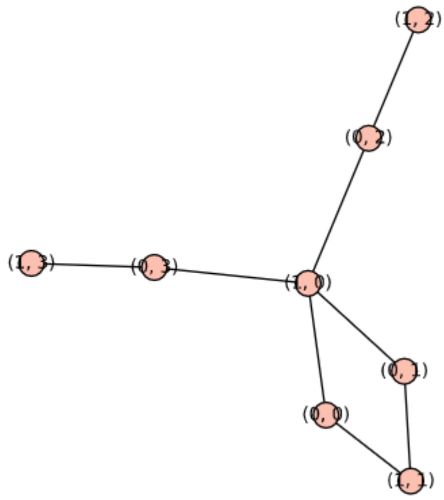
SLIKA 4. 8-kotnik

3.2.3. *Drevesa.* Opaziva, da je struktura grafa enaka poti na n vozliščih. A ta ima glede na običajen povezan graf in dvodelen graf večjo vrednost minimalnega G-G indeksa. Razlika je zelo majhna ampak opazna (($GGI(op) = 4.89$ in $GGI(dr) = 5.14$)). Zanimiva je struktura, ki se pokaže kot optimum pri iskanju maksimuma G-G indeksa na drevesih. Zanjo velja, da eno (centralno) vozlišče povezano z vsemi drugimi, druga pa ne med sabo. To je graf zvezde na n vozliščih.

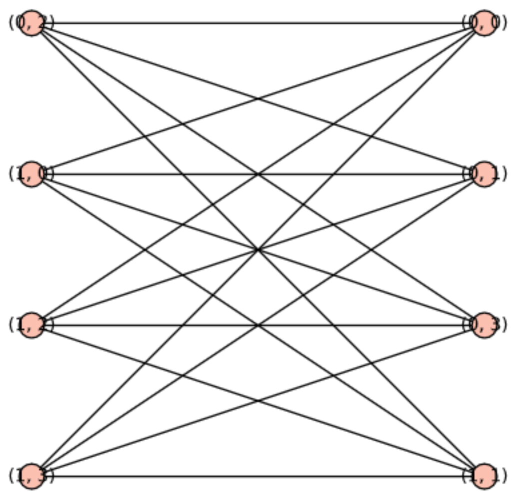


SLIKA 5. zvezda

3.2.4. *Dvodelen graf.* Iz eksperimenta na dvodelnem grafu se lahko sklepa, da je G-G indeks največji pri polnem dvodelnem grafu. To je dvodelen graf za katerega velja, da so vozlišča iz leve particije povezana z vsemi vozlišči iz desne particije in obratno. Struktura grafa pri katerem je G-G indeks najmanjši pa se z velikostjo problema zelo razlikuje tako, da je težko opisljiva v primeru ko je vozlišč 8, zgleda tako kot na sliki prikazani spodaj.

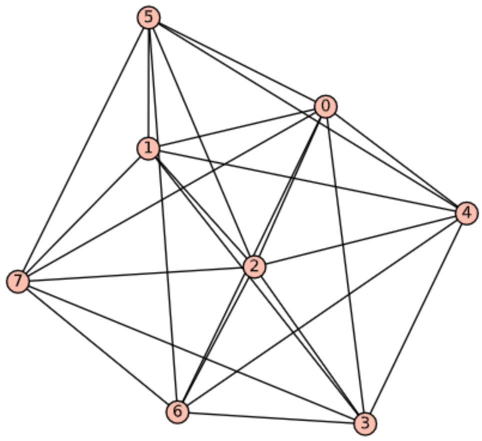


Slika 6. dvodelen graf na 8 vozliščih z majhnim G-G indeksom



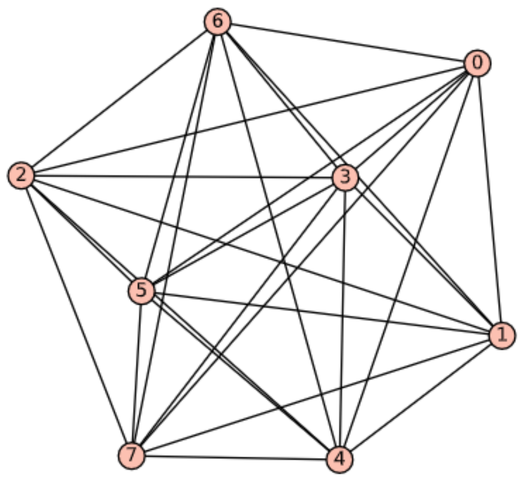
Slika 7. pol dvodelen graf na 8 vozliščih

3.2.5. *Struktura z zelo velikim indeksom na n-vozliščih.* Izkaže se, da je struktura z zelo velikim indeksom, struktura grafa na n vozliščih za katere velja, da imajo $d(v) = n - 2$. Ta struktura v sklopu eksperimenta spada pod kategorijo običajnih povezanih grafov.

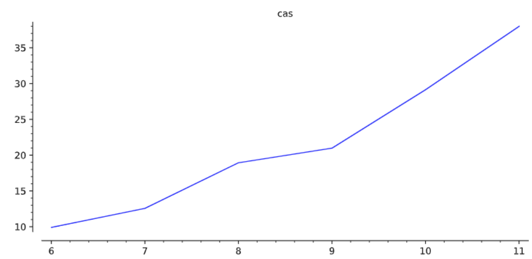


SLIKA 8. graf na 8 vozliščih z velikim G-G indeksom

3.2.6. *Struktura z zelo majhnim indeksom na n-vozliščih.* Pri testiranju G-G indeksa na grafih z majhnim številom vozlišč neodvisno od skupnega eksperimenta se izkaže , da je struktura z zelo majhnim indeksom struktura polnega grafa na n vozliščih, kar ovrže hipotezo eksperimenta iz katerega je bilo mogoče predvidevati, da je takšna struktura n kotnik. Saj algoritem pri iskanju zelo majhnega G-G indeksa favorizira odstranjevanje povezav in ne dodajanje. Lahko razmislimo, da je G-G indeks za graf na n vozliščih velja : $GGI(G) \geq 0$ in če X predstavlja graf na n vozliščih velja $min(GGI(X)) \leq n$.



SLIKA 9. graf na 8 vozliščih z majhnim G-G indeksom



Slika 10. Primer trajanja izvedbe zanke v odvisnosti od izbire števila vozlišč za graf brez trikotnika