σ-irregularity vs. total σ-irregularity

Nejc Ševerkar & Anja Trobec

3. december 2019

Kazalo

1 Uvod

V projektni nalogi se bova ukvarjala z merjenjem iregularnosti enostavnih neusmerjenih grafov z dvema na videz podobnima metodama, *σ-irregularity* in *total σ-irregularity*, definiranima kot

$$\sigma(G) = \sum_{(u,v)\in E(G)} (d_u - d_v)^2 \quad \text{in} \quad \sigma_t(G) = \sum_{(u,v)\in V(G)} (d_u - d_v)^2$$

Cilj naloge je maksimizacija razmerja, definiranega kot

$$\sigma_r(G) = \frac{\sigma_t(G)}{\sigma(G)}$$

pri danem redu grafa $n \in \mathbb{N}$ in tako ugotoviti stopnjo naraščanja $\sigma_r(G)$, v odvisnosti od reda.
Ker nas grafi, za katere to razmerje ni definirano, torej v primeru $\sigma(G) = 0$, ne zanimajo, definiramo $\sigma_r(G) = 0$. To je natanko tedaj, kadar so vse komponenete grafa G regularne.

1.1 Naloge

Najina glavna naloga je torej poiskati maksimalno razmerje $\sigma_r(G)$ med grafi na n vozliščih in jih temeljito preučiti. Najprej bova to počela za majhne grafe, potem pa bova z algoritmi problem reševala še za grafe reda n.

Po prvih oprijemljivih ugotovitvah naju čakajo naslednje naloge.

Za optimalne grafe je dana hipoteza, da so krajišča njihovih povezav vedno na absolutni razliki 1.
To bova preverila tako, da bova izračunala število vozlišč za katera to ne velja pri vsakem optimalnem grafu in ta prikazala z grafom.

Naslednja stvar, ki pa jo bova morala preveriti je ta, da je sup

$$\frac{\sigma_t(G)}{\sigma(G)}$$

reda $O(n^2)$. Poiskati bova morala še ustrezno konstanto c s katero bo aproksimacija cn^2 najustreznejša.

2 Osnovna Teorija

2.1 Maksimalna Stopnja Naraščanja

Ker bomo za analizo in primerjavo potrebovali zgornjo mejo stopnje naraščanja zaporedja $(\max(\sigma_r(G_n)))_n$ v odvisnosti od reda n si poglejmo naslednji račun.

$$\sigma_r(G) = \frac{\sigma_t(G)}{\sigma(G)} \leq \sigma_t(G) = \sum_{(u,v)\in V(G)} (d_u - d_v)^2 < \sum_{(u,v)\in V(G)} n^2 < n^2 n^2 = n^4,$$

Sledi, da je red naraščanja $O(n^4)$.

2.2 Nepovezani Grafi

Ker sva opazila, da družina nepovezanih grafov doseže maksimalno stopnjo naraščanja, se lahko po konstrukciji družine takšnih grafov G_n za $\forall n \in \mathbb{N}$ osredotočimo samo na povezane grafe.

Konstrukcije grafov G_n, za katere ima zaporedje $(\max(\sigma_r(G_n)))_n$ stopnjo naraščanja $O(n^4)$ je sledeča. Vzamemo $n/2$ vozlišč in iz njih konstruiramo poln graf medtem, ko v preostalih $n/2$ vozliščih povežemo 3 vozlišča z dvema povezavama. Po kratkem premisleku lahko formuliramo naslednjo neenakost.

$$\sigma_r(G_{2n}) > \frac{(n-3)n\binom{n}{2}}{2} = \frac{(n-3)nn(n-1)}{4} = \Theta(n^4)$$

.
S tem sva zaključila preučevanje nepovezanih grafov.

3 Ideja Iskanja Optimalnih Grafov

Problem bova reševala v Pythonu in si občasno pomagala s knjižnjico Networkx, s katerim bodo grafično predstavljeni grafi.

3.1 Metahevristike

Za optimalno vrednost σ_r na grafih reda n morava testirati vse neizomorfne grafe tega reda, katerih je $\Omega(2^n)$. Če upoštevamo, da izračun $\sigma_r(G)$ zahteva $\Omega(n^2)$ operacij, dobimo skupno časovno zahtevnost $\Omega(n^2 2^n)$. Očitno ta zahtevnost predstavlja problem že za grafe reda 10, torej bova morala poiskati alternativni pristop v obliki metahevrističnih algoritmov.

Ideja bo torej sistematično postopati po prostoru povezanih enostavnih grafov reda n in tako iskati aproksimacijo grafa G, ki maksimizira vrednost σ_r na tem prostoru.
Za učinkovito delovanje teh procesov pa potrebujemo definirati ustrezno topologijo na prostoru, torej podati pojem bližine, saj jo zahteva večina hevrističnih algoritmov.
To bova naredila v obliki zaporednega dodajanja ali odstranjevanja naključnih povezav v danem grafu, pri čimer morava paziti, da ohranjava povezanost grafa, torej z drugimi besedami ne odstraniva mostov.
Za te namene je napisana knjižnjica, ki poda podporo za izbiro teh povezav in splošno generiranje naključnih povezanih grafov.

3.2 Simulated Annealing

Eden od algoritmov, ki naj bi rešil ta problem je *Simulated Annealing*, katerega implementacija je končana, a brez okolice, ki bi vrnila dobre rezultate. Iz tega razloga sva poskusila implementacijo drugega algoritma.

4 Reševanje Problema

4.1 Majhni grafi

Za majhne grafe sva na strani ** našla že generirane povezane neizomorfne grafe do stopnje 9. Na njih sva poiskala maksimalne $\sigma_r(G)$, ki so zapisane v naslednji tabeli.

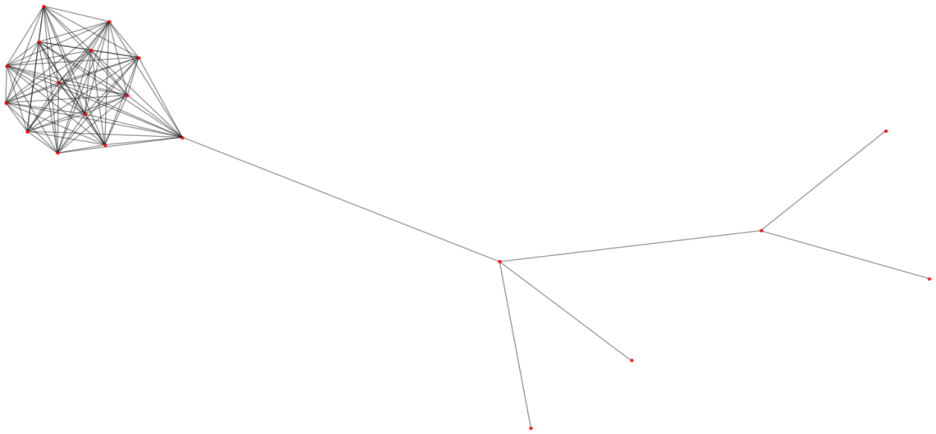
n	$\sigma_r(G)$
2	0
3	1
4	3
5	3
6	5
7	13
8	19
9	14.5

Grafov na večjemu številu vozlišč je preveč za posamično analizo, torej se lotimo implementacije *Simulated Annealing* algoritma.

4.2 Večji grafi

Med večjimi grafi bomo optimume iskali s pomočjo algoritma *Simulated Annealing*. Ta sprejme 3 argumente: število vozlišč, število simulacij in definicijo okolice grafa.
Okolice so lahko poljubne in se delijo na lokalne in globalne. Lokalne okolice spreminjajo graf po povezavah, globalne pa konstruirajo nov graf, ki ima morda kakšne podobne karakteristike kot prejšnji.
Izkazalo se je da aplikacija samo lokalnih okolic ni dovolj pri velikih grafih in sva zato iskanje izvedla v dveh delih. Prvi del je klicanje SA na globalni okolici, nakar pa še na lokalni, saj ta naredi na tem male optimizacije, ki jih globalne spremembe niso bile zmožne doseči.
Lokalno izboljšavanje nastopi v obliki minimizacije $\sigma(G)$ z osredotočanjem na povezave, kjer je razlika med $d_G(V)$ in $d_G(U)$ za neka $uv \in E(G)$ največja.

Po nekaj testih je začela biti očitna struktura optimalnih grafov in sicer njihova oblika je sestavljena iz dveh delov. V prvem nastopa polni podgraf, na drugem pa so povezave redke in je zato podgraf blizu drevesu, kot lahko vidimo na spodnji sliki grafa na 20ih vozliščih.



Ta sestava je primerna, saj razlike med polnim grafom in redkim prispevajo k vrednosti σ_t, medtem ko vrednost σ ostaja enaka, saj so v obeh delih grafa sosednje povezave podobnih stopenj. To je skoraj res, a naletimo na posebno povezavo, ki je prisotna v teh grafih in to je most med polnim in praznim podgrafom. Brez te povezave bi imeli stopnjo naraščanja $O(n^4)$, a njena prisotnos znatnot povišuje vrednost σ.

Naš problem se po tej analizi prevede v to kako zvezno, torej s čim manjšo razliko stopenj sosednih povezav, povezati poln podgraf z redkim. Za to potrebujemo veliko vozlišč, ki bodo poskrbeli za žvezno"pot med tima podgrafoma, a jih hkrati potrebujemo malo povezane, da maksimizirajo σ_t vrednost, saj se ta primerja z nastalim polnim grafom. Problem nam torej predstavljata tudi spodbijajoči optimizaciji σ ter σ_t.

V nadaljevanju si bomo pogledali katero optimizacijo favorizirajo optimalni grafi.

4.3 Kvadratična stopnja naraščanja

Preveriti morava še, če je stopnja narašanja zaporedja $(\max(\sigma_r(G_n)))_n$, kjer G_n graf stopnje $n \in \mathbb{N}$, $O(n^2)$ in poiskati ustrezno konstanto $c \in \mathbb{R}$, ki se naraščanju najbolj prilega.

Predpostavimo, da smo optimum izračunali za n grafov in označimo $X(n,p) = (1^p, 2^p, \ldots, n^p)^T \in \mathbb{R}^{nx1}$ in $a = (a_1, a_2, \ldots, a_n)^T$ vektor, kjer a_i predstavlja dobljeni optimalni $\sigma_r(G_i)$ za graf G_i na i vozliščih. Problem pri danem $p \in \mathbb{R}^+$ zahteva rešitev linearnega sistema

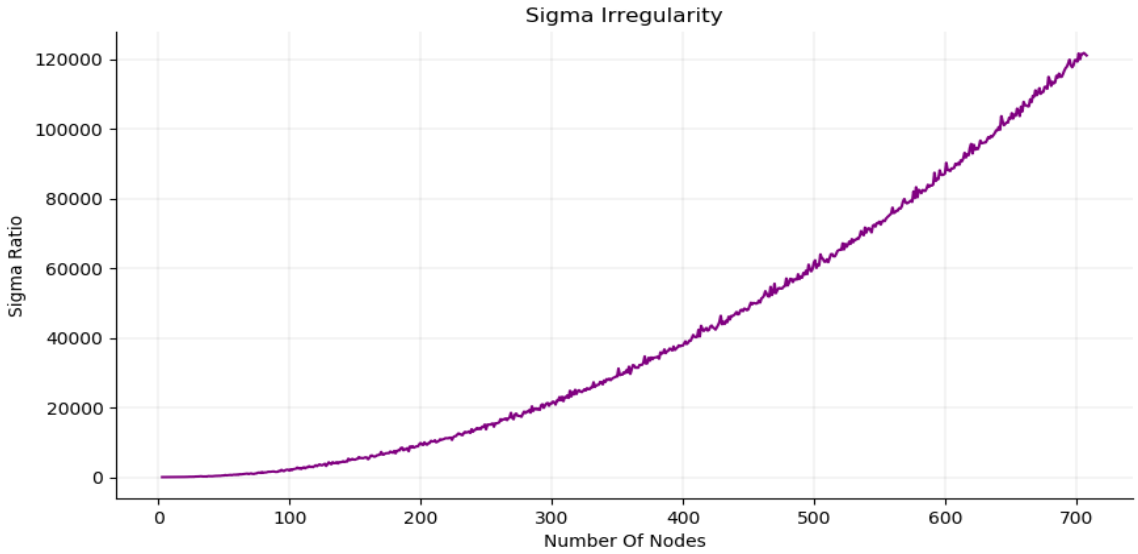
$$cX(n,p) = a.$$

Ta sistem seveda ne bo rešljiv, iskanja aproksimacije pa se bomo lotili z metodo najmanjših kvadratov, torej minimum 2-norme razlike obeh strani bo dosežen pri

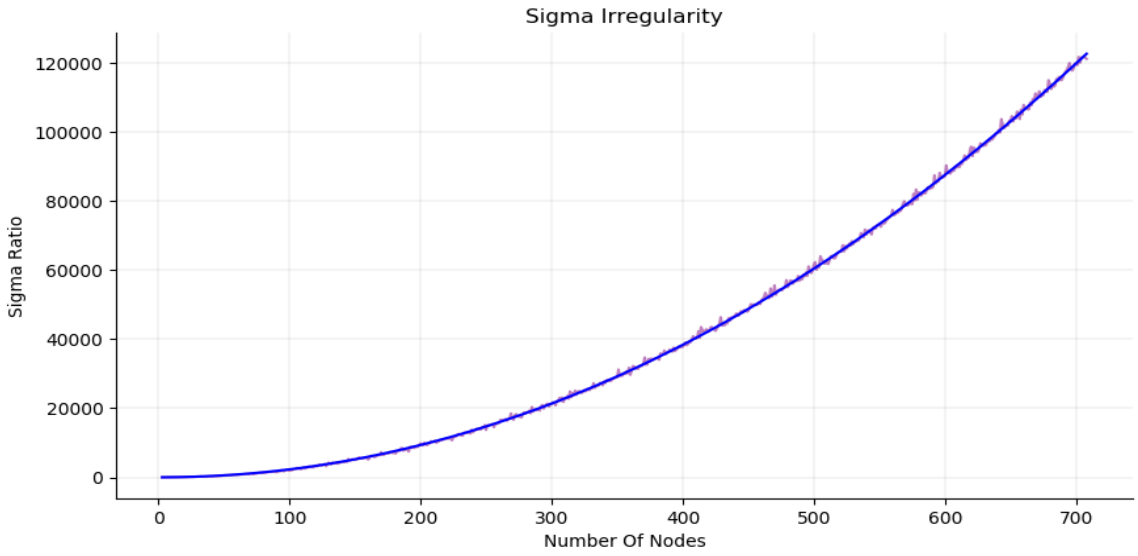
$$c = ||X(n,p)||_2^2 / < a, X(n,p) > .$$

Najboljšo aproksimacijo naraščanja σ_r bomo poiskali z diskretizacijo $D \subset [0,4]$ in za $\forall p \in D$ izračunali prej definirani c, na koncu pa izbrali tisto kombinacijo (c,p), za katero je $||cX(n,p) - a||_2$ najmanjši.

Tako dobimo po testiranju na vozliščih od 3 do 700 naslednji graf.

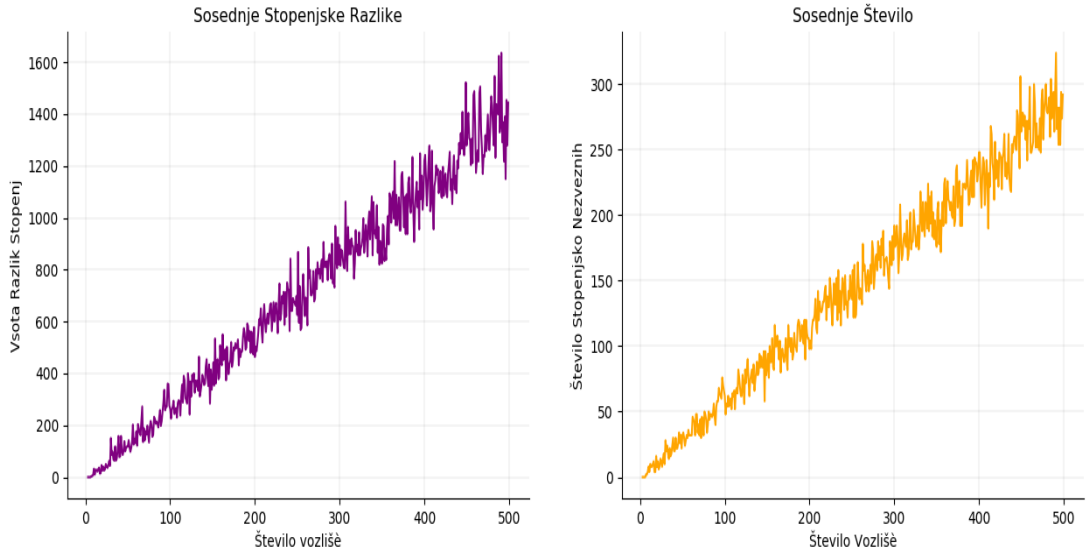


Po prej opisani metodi dobimo par (c, p), za katerega cn^p najbolje opisuje σ_r naraščanje. Ta par je izračunan kot $(0.19319714558507617, 2.0360000000000014)$ in njegova aproksimacija naraščanja je prikazana na naslednjem grafu.

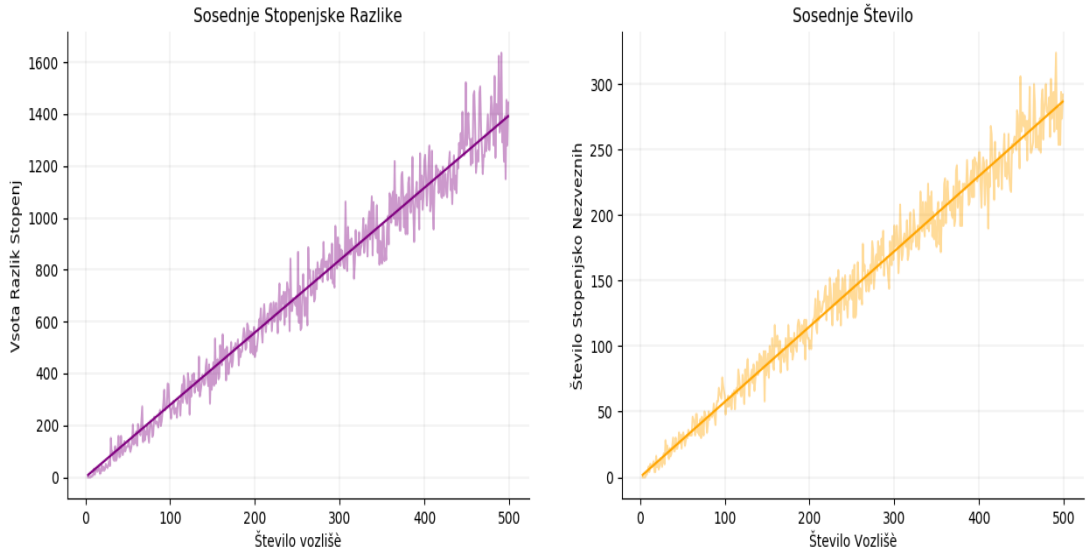


4.4 Hipoteza o stopnjah vozlišč

Hipotezo, ki pravi da so vsa sosednja vozlišča stopnje, ki se razlikuje za največ 1, bova testirala z izračunom števila parov vozlišč, ki nimajo ustrezne razlike stopenj in seštela še vse njihove razlike. Rezultati so opisani na naslednjem grafu.



Naraščanje je predvsem linearno in sicer sta najboljši linearni aproksiciji koeficientov enaki 2.7894615404336047 in 0.5748786913019662, glede na horizontalno zaporedje slik. Spodaj je prikazana še linearna aproksimacija.

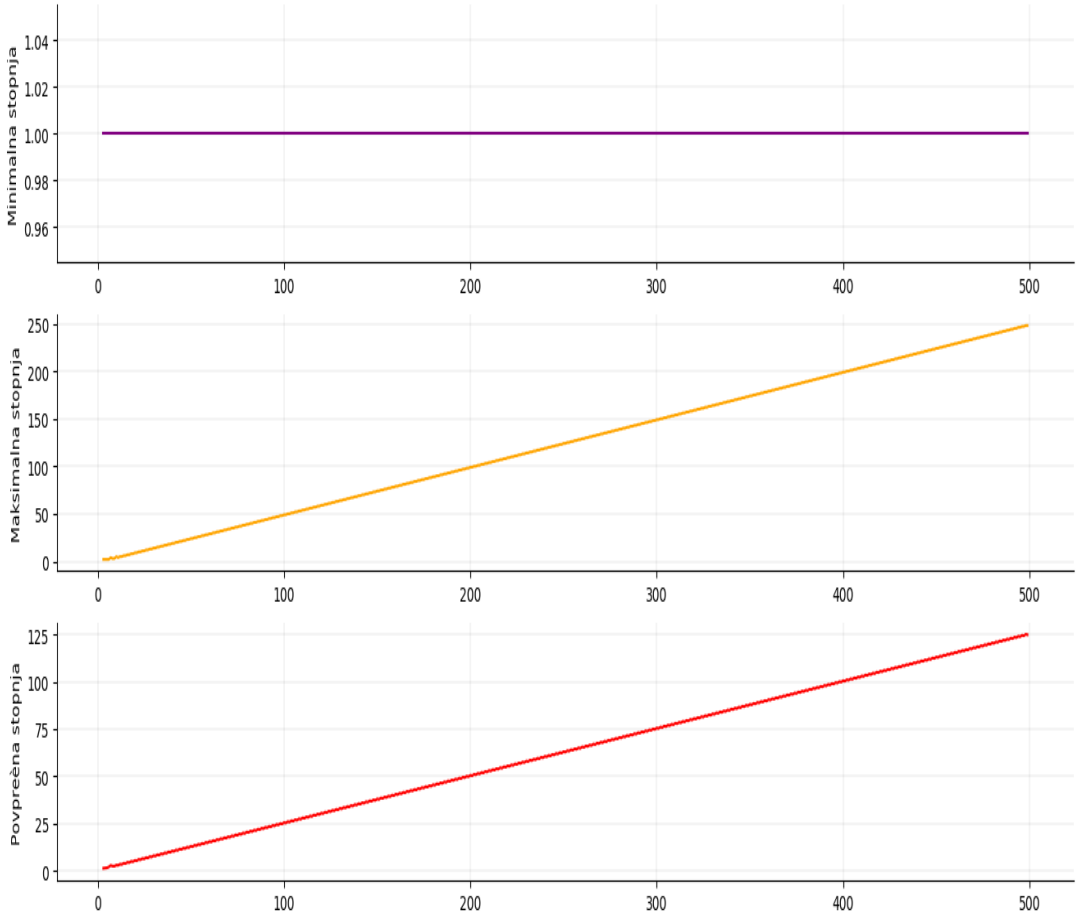


Na levem grafu vidimo, da je vsota stopenj vozišč majhna, saj je povprečno njena vrednost le trikratnik števila vozlišč, torej ima vsako vozlišče povprečno stopnjo 3, kot tudi namiguje koeficient. Na desnem grafu takoj opazimo, da je naraščanje števila parov vozlišč, ki ne ustrezajo pogoju zelo majhno, saj narašča počasneje kot stopnja vozlišč, čeprav je njeno potencialno naraščanje $\binom{n}{2}$. Seveda je rezultat smiseln, saj poskušamo minimizirati σ vrednost, katera pa narašča skupaj z naraščanjem razlik stopenj sosednih vozlišč.

Vrnimo se nazaj na problem optimalnega povezanja polnega podgrafa s praznim. Sva mnenja, da so rezultati pozitivna indikacija na to, da se ta pot naravno formira na način, ki favorizira minimizacijo σ vrednosti napram maksimizacije σ_t vrednosti. Torej grafi poskušajo poskrbeti za zvezni prehod iz polnega podgrafa k redkemu in za to žrtvujejo nekaj vozlišč, kar pa se na koncu pozna pri majhnih razlikah stopenj vozlišč.

4.5 Distribucija stopenj vozlišč

Ideja bo prikazati minimalne, maximalne in povprečne stopnje vozlišč pri danem optimalnem grafu na n vozliščih. Spodaj je prikazan graf, ki prikazuje prej povedano.



5 Opisi Implementiranih Algoritmov

Podali bomo kratek opis vseh vključenih algoritmov v napisani knjižnici in njihovo časovno zahtevnost. V tabeli bo n označeval število vozlišč grafa, m pa število povezav.

Ime	Kratek Opis	T
sigma	izračuna vrednost $\sigma(G)$ na danem grafu G	$O(n+m)$
sigma_t	izračuna vrednost $\sigma_t(G)$ na danem grafu G	$O(n^2)$
sigmaRatio	izračuna vrednost $\sigma_r(G)$ na danem grafu G	$O(n^2)$
sigmaUpdate	izračuna razliko σ po odstranjeni ali dodani povezavi v G	$O(n)$
sigmaArgmax	vrne povezavo, ki maksimizira σ vrednost v danem grafu G	$O(n+m)$
randomConnectedGraph	konstruira naključni graf na n vozliščih	$O(n^2 \log^*(n))$
randomTree	konstruira naključno drevo na n vozliščih	$O(n)$
randomPath	konstruira naključno pot na n vozliščih	$O(n)$
randomSubtree	poišče naključno poddrevo danega grafa	$O(n)$
randomSigmaOptAprox	konstruira graf, ki naj bi bil grob približek optimalnemu	$O(n^2)$
nonBridges	poišče k povezav, ki niso mostovi v danem grafu	$O(m+n)$
nonEdges	poišče k povezav, ki niso vključene v dani graf	$O(n^2)$
localBasicNeighbor	graf spremeni z dodajanjem in odstranjevanjem povezav	$O(n^2)$
globalBasicNeighbor	uporabi *randomConnectedGraph* za konstrukcijo okolice	$O(n^2 \log^*(n))$
globalTwoPartNeighbor	uporabi *randomSigmaOptAprox* za konstrukcijo okolice	$O(n^2)$
maxSigmaRatio_annealing	uporabi Simulated Annealing za iskanje optimalnega grafa	odvisno od argumentov