

1 Uvod

Pri tej nalogi smo imeli nekaj dela s teorijo grafov. Natančneje, opravka smo imeli z obojestransko povezanimi grafi. Z njimi lahko namreč natančneje obravnavamo populacijske modele, saj so taki grafi lokalizirani.

Modeliramo lahko na primer prenašanje vica/ideje/bolezni preko populacije. Povezave med ljudmi v taki populaciji so stohastične.

Poleg tega je bilo potrebno še preveriti Kapitzov populacijski model. Kot piše v navodilu je pričel s preprostimi predpostavkami in prišel do zanimivih rezultatov. To bomo obravnavali v nalogah 2 in 3.

2 Povezave

V tej nalogi smo pričeli z grafom, kjer je bilo vsako vozlišče povezano zgolj z najbližjimi sosedi. To število mora biti sodo, saj bi v nasprotnem primeru dobili orientirane grafe. Prav tako moramo predpostaviti, da je vsako vozlišče povezano samo s seboj. Tako lahko s potenciranjem matrike povezav pridemo do premera grafa.

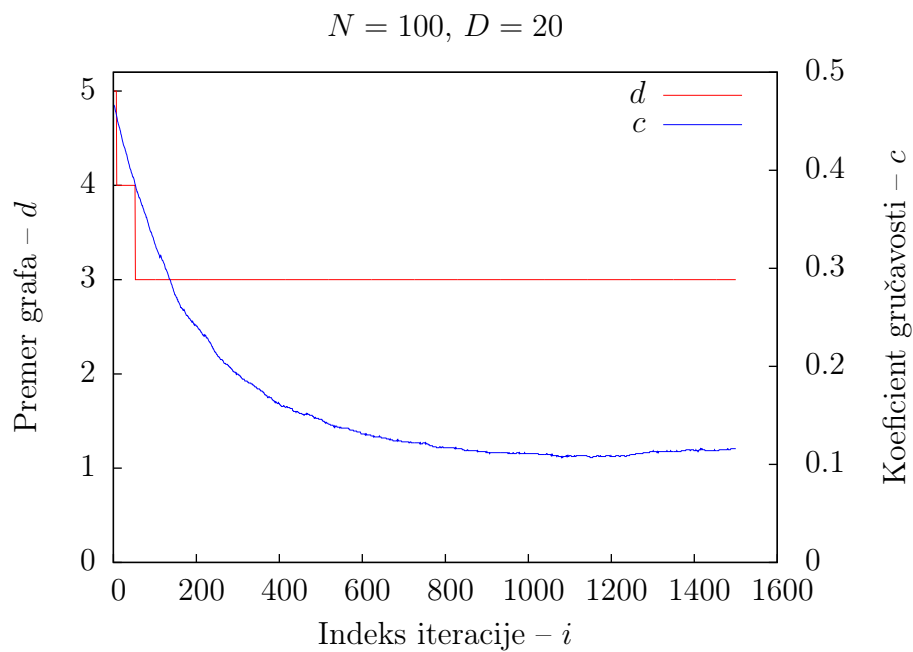
Tak graf seveda ni realen – da dobimo populacijo, ki nas bolje opiše moramo povezave stohastično premikati znotraj matrike povezav. Ta je simetrična, ima zgolj enice in ničle. Indeksi stolpcev predstavljajo vozlišča, indeksi vrstic pa predstavljajo vozlišče, na katerega je stolpec povezan.

Ker poznamo število povezav D , ki je majhno v primerjavi s številom vozlišč, je veliko prostora za optimizacijo. Sam sem na primer vse povezave pospravil v ločeno tabelo, naključno število, pa je predstavljalo indeks povezave. S tem se izognemo nepotrebnemu žrebanju števil. Dobljeno enico v matriki pobrišemo in prestavimo na novo mesto. Tu je spet prostor za optimizacijo: enic je veliko manj kot ničel, zato se teh ne splača imeti pospravljenih v ločeni tabeli. Zato sem najprej preveril, če je verjetnost, da izžrebamo ničlo večja od 0.5. Če je, potem lahko žrebamo, sicer pa jo enostavno premikamo po sosedih, dokler ne pridemo do nje in jo pretvorimo v enico.

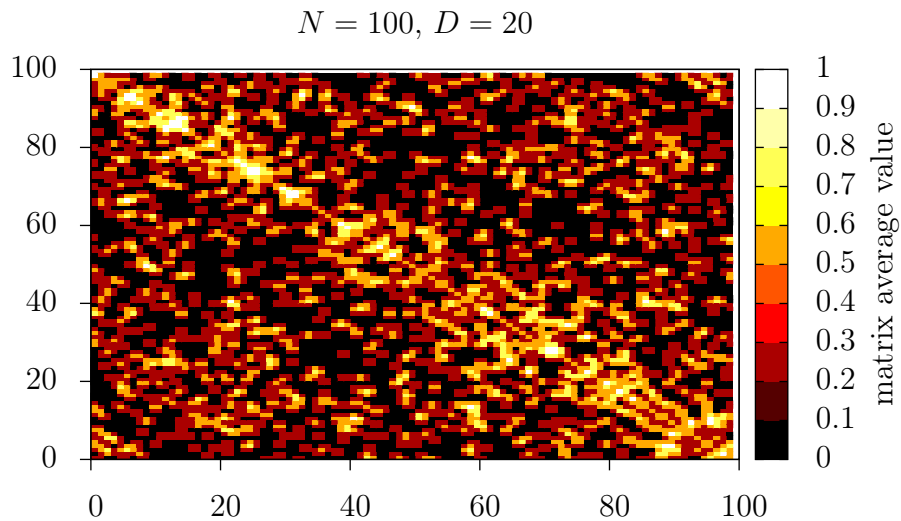
To počnemo, dokler so izpolnjeni vsi izmed sledečih pogojev:

- (a) število iteracij manjše od 10^4 ,
- (b) koeficient gručavosti je večji od 0.001,
- (c) gručavost se je zadnjih 100 iteracij spremenila za več kot 0.001.

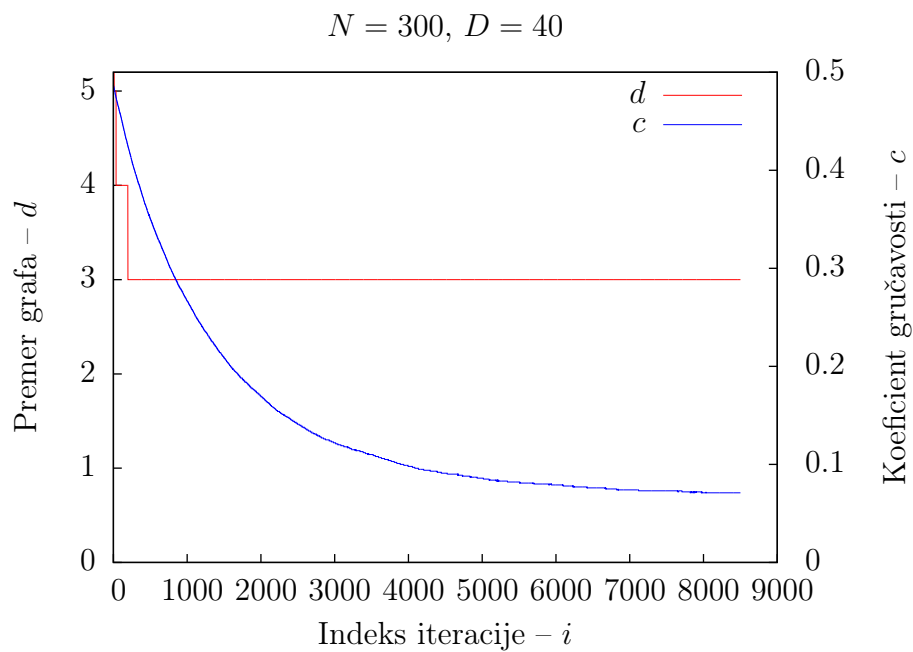
V kolikor je eden izmed teh pogojev neizpolnjen je se iteracijo prekine. Zadnji pogoj sem imel sprva nastavljen na vsakih 500 iteracij, vendar je na vsakih 100 zadosten pogoj.



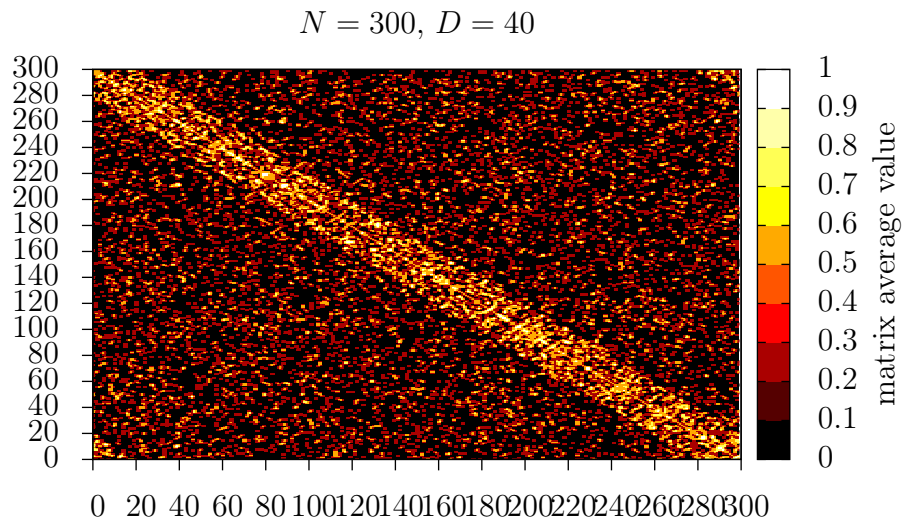
Slika 1: Za primerjavo lahko vidimo kako se s številom iteracij spreminjata koeficient gručavosti ter premer grafa.



Slika 2: Matrika na koncu je res videti bolj ali manj razpršena.



Slika 3: Za primerjavo lahko vidimo kako se s številom iteracij spreminjata koeficient gručavosti ter premer grafa.



Slika 4: Matrika na koncu je res videti bolj ali manj razpršena.