UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

FINANČNA MATEMATIKA – 1.stopnja

Matej Rojec, Vito Rozman, Ana Marija Belingar

Uravnotežen rdeče modri povezan podgraf

Poročilo pri predmetu Finančni praktikum

Ljubljana, 2022

Kazalo

1 Uvod **3**

2 Predstavitev problema **3**

3 Osnovni pojmi **3**

4 Algoritem **4**

5 Eksperimenti **5**
5.1 Delež vozlišč v BCS grafu glede na verjetnost p 5
5.2 Delež vozlišč v BCS grafu glede na velikost mreže 6
5.3 Hitrost BSC glede na velikost mreže 6

Slike

1 Prikaz porazdelitve za $n = 50$ ter $m \in \{1, 2, 3, 4\}$. 5
2 Prikaz deleža vozlišč v grafu pri $p = 0.5$ v odvisnosti od n. . . 6
3 Prikaz deleža vozlišč v grafu pri $p = 0.5$ v odvisnosti od n. . . 7

Tabele

1 Tabelirane vrednosti 6

1 Uvod

V poročilu bomo predstavili problem iskanja največjega povezanega uravnoteženega podgrafa na mrežah velikosti $1 \times n$ (pot), $2 \times n$, $3 \times n$ in $4 \times n$, pri čemer je n število stolpcev. Na začetku bomo predstavili kako smo se lotili reševanja problema. Nato bomo predstavili kodo s katero smo problem rešili, v zadnjem delu pa eksperimente, rezultate in ugotovitve do katerih smo prišli. Za zaključek bomo navedli še možne izboljšave.

2 Predstavitev problema

Naj bo $G = (V, E)$ graf. Vsako vozlišče $v \in V$ je obarvano rdeče ali modro. Najti želimo največji povezani podgraf $G' = (V', E')$, ki ima enako število rdečih in modrih vozlišč. Velikost podgrafa je število njegovih vozlišč. Ta problem je v splošnem NP-težek, kar pomeni da ga ne moremo rešiti v polinomskem času. Osredotočili se bomo na optimalen algoritem za reševanje problema na mrežah oblike $1 \times n$ (pot), $2 \times n$, $3 \times n$ in $4 \times n$. Naš algoritem je testiran na grafih, kjer so vožlišča obarvali rdeče z verjetnostjo $p \in (0, 1)$ in modro z verjetnostjo $1 - p$.

3 Osnovni pojmi

Definicija 1. (Induciran podgraf) Naj bo $G = (V, E)$ graf in naj bo $S \subseteq V$ podmnožica vozlišč grafa G. Graf $G[S]$ je induciran podgraf grafa G, natanko takrat ko $\forall u, v \in S$ velja, da sta u in v sosednja v $G[S]$, natanko takrat ko sta sosednja v G.

V našem primeru bomo iskali največji povezan inducirani podgraf $G' = (V', E')$ grafa $G = (V, E)$, kjer lahko možico volzlišč zapišemo kot:

$$V' = V_R \cup V_B,$$

za katero velja $V_R \cap V_B = \emptyset$ in $|V_R| = |V_B| = \frac{|V'|}{2}$. Tako bomo dobili največji uravnotežen povezan podgraf z enakim številom rdečih in modrih vozlišč.

Definicija 2. (Pot) Graf $P_n = (V, E)$ je pot, kjer je $V = \{v_1, ..., v_n\}$, če je množica povezav oblike: $E = \{(1, 2), (2, 3), ..., (n - 1, n)\}$.

Definicija 3. (Mreža) Graf $M_{m,n} = (V, E)$ je mreža velikosti $m \times n$, kjer je $V = \{v_{1,1}, ..., v_{1,n}, v_{2,1}, ..., v_{2,n}, ..., v_{m,1}, ..., v_{m,n}\}$, če je množica povezav oblike:

$$E = \{((i, i'), (j, j')) \mid |i - i'| + |j - j'| = 1\}$$

Definicija 4. (Particija vzorcev) Naj bo G graf. Za graf $G \square P_{n-1}$ je $\mathcal{V}$ particija množice povezanih komponent podgrafa grafa G.

Definicija 5. (Združljiva vzorca) Vzorca u in v sta združljiva, pišemo $u \sim v$, če ko pogledamo unijo induciranih podgrafov grafa G na vozliščih iz u oziroma v, potem za vsako povezano komponento v istem delu v velja, da pripada isti povezani komponenti v uniji (pri čemer upoštevamo povezanost komponent v istem delu u).

4 Algoritem

Ogledali si bomo problem na mrežah $(m \times n)$, pri čemer bomo uporabili dinamično programiranje kot je razvidno iz algoritma **??** . Algoritem sprejme kot vhod mrežo $m \times n$, kjer je vsako vozlišče obarvano bodisi modro (B) bodisi rdeče (R).

Definirajmo spremenljivke:

- $x_{ik} = 1$, če je k-to vozlišče v i-tem stolpcu rdeče, in -1, če je modro
- p_{ijv} je največji podgraf do i-tega stolpca z razliko j, kjer je v zadnjem stolpcu uporabljeni vzorec v

Začetni pogoji so:

- $p_{10v} = 0$
- $p_{1jv} = \begin{cases} |v|, & \text{če } j = \sum_{k \in v} x_{1k} \\ -\infty, & \text{če } j \neq \sum_{k \in v} x_{1k} \quad \text{in} \quad j \neq 0 \end{cases}$
- $p_{ijv} = -\infty$, če je $\quad |j| > i|G|$

Rekurzivne enačbe:

$$p_{ijv} = |v| + \max\left(\{p_{i-1, j-\sum_{k \in v} x_{ik}, u} \mid u \sim v\} \cup \{-|v| \mid j = 0\}\right), \; (-i|G| \leq j \leq i|G|)$$

Optimalna rešitev:

$$p^* = \max\{p_{i0v} \mid i \leq 1 \leq n, v \text{ povezan}\}$$

Za programerski del naloge smo uporabljali Python. V nadaljevanju so predstavljenje glavne funkcije, ki smo jih za ta del definirali.

Glavni funkciji se nahajata v `algoritem.py`, ki sta `P`, ki izračuna vse vsrednosti p_{ijv} od zgoraj ter pripdajoč graf ter `max_BCS`, ki najde največji uravnotežen povezan podgraf.

Pomožna datoteka `modelBCS.py` vsebuje razred `BSC`, ki generira osnovne parametre za lažji izračun.

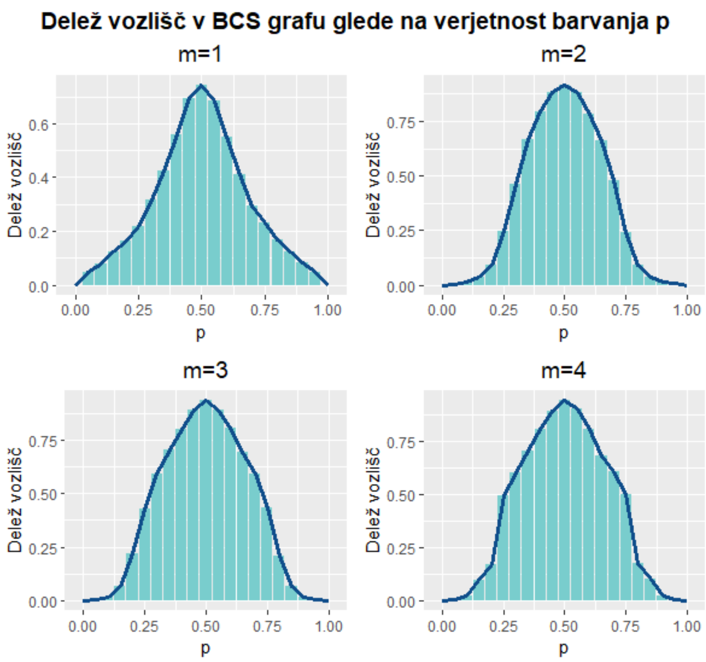
5 Eksperimenti

V tem delu seminarske naloge smo se ukvarjali s tremi ključnimi vprašanji:

- Kako se velikost rešitve spreminja glede na verjetnost p.
- Kako se velikost rešitve spreminja glede na dolžino mreže n.
- Kako se hitrost algoritma spreminja, če večamo n in m.

5.1 Delež vozlišč v BCS grafu glede na verjetnost p

Za odgovor na prvo vprašanje smo za $m = 1$ in $m = 2$ izvedli 500 ponovitev poskusa ter za $m = 3$ in $m = 4$ pa 250 ponovitev poskusa na mreži dolžine $n = 50$ pri različnih verjetnostih. Slika 1 prikazuje rezultate.

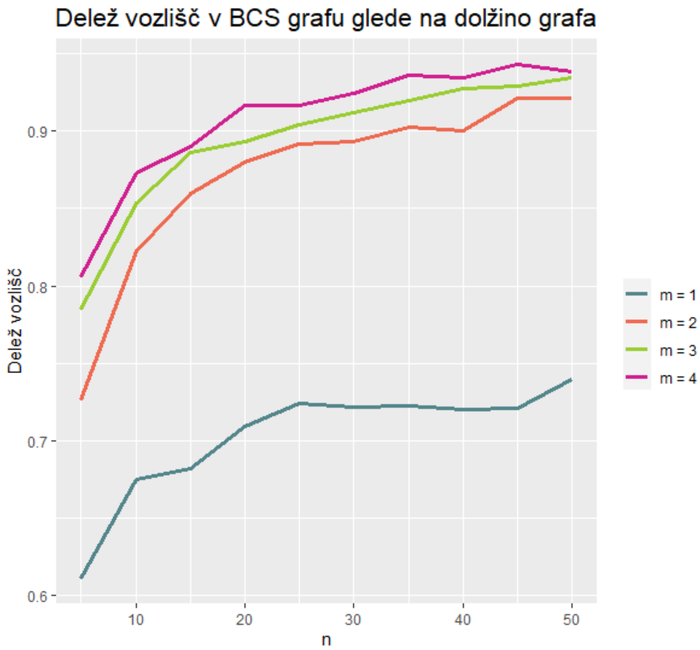


Slika 1: Prikaz porazdelitve za $n = 50$ ter $m \in \{1, 2, 3, 4\}$.

Prva ugotovitev je, da je za posamezen m največja rešitev pri $p = 0.5$, kar je pričakovano, saj je takrat število modrih in rdečih vozlišč v grafu v povprečju enako, zato je verjetnost da dobimo uravnotežen povezan podgraf največja. Bolj kot se bližamo robovom ($p = 0$ ali $p = 1$), manjšo rešitev dobimo. Iz eksperimentov se zdi , da se z dodajanjem vrstic (večanjem m-ja) veča tudi delež vozlišč glede na velikost celotnega grafa pri fiksnem p.

5.2 Delež vozlišč v BCS grafu glede na velikost mreže

Za odgovor na drugo vprašanje smo za vsak m posebej izvedli 250 ponovitev pri verjetnosti barvanja grafa $p = 0.5$ in pri različnih dolžinah grafa (tj. različnih n-jih).



Slika 2: Prikaz deleža vozlišč v grafu pri $p = 0.5$ v odvisnosti od n.

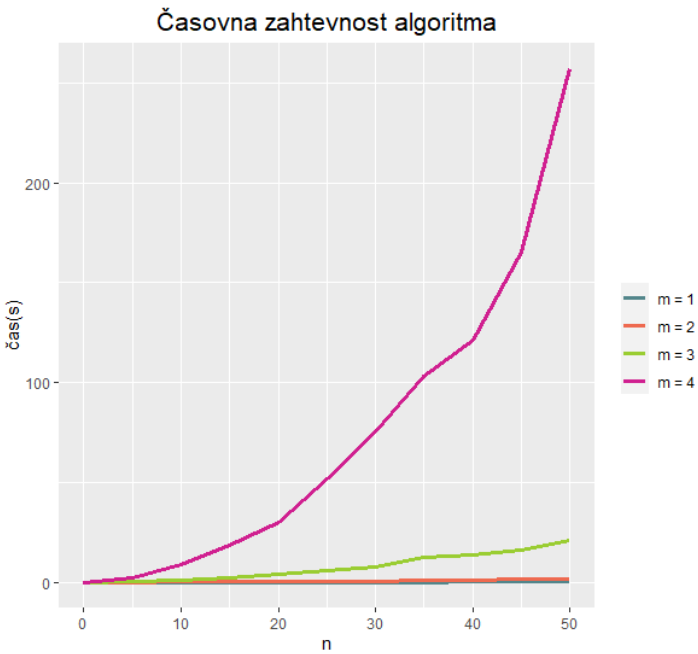
Kot vidimo iz slike 2 je delež vozlišč glede na velikost celotne poti veliko manjši kot pri ostalih m-jih. Iz grafa vidimo, da ne gre za linearno rast, ampak morda za logaritmično.

5.3 Hitrost BSC glede na velikost mreže

Kot zadnji eksperiment smo raziskovali časovno zahtevnost našega algoritma kar je razvidno iz slike 3 ter tabele 1. Podatki prikazujejo povprečen čas izvajanja algoritma v sekundah.

Tabela 1: Tabelirane vrednosti

$m \setminus n$	0	5	10	15	20	25	30	35	40	45	50
1	0.000	0.003	0.012	0.039	0.074	0.083	0.107	0.119	0.132	0.147	0.158
2	0.000	0.035	0.127	0.209	0.301	0.481	0.691	0.903	1.215	1.527	1.859
3	0.000	0.286	1.046	2.260	3.968	5.861	7.481	12.710	13.544	16.206	20.785
4	0.000	2.309	9.090	18.551	30.035	51.521	75.599	103.006	121.200	165.482	257.030



Slika 3: Prikaz deleža vozlišč v grafu pri $p = 0.5$ v odvisnosti od n.

Iz grafa 3 je razvidno, da je algoritem bistveno počasnejši za velike n .