

UNIVERZA V LJUBLJANI

Fakulteta za matematiko in fiziko

Finančni praktikum

Največje neodvisne množice z lokalnim
iskanjem

Avtorja:

Jaka Mrak

Žiga Gartner

Mentorja:

prof. dr. Sergio CABELLO

doc. dr. Janoš VIDALI

Ljubljana, 9. januar 2022

Kazalo

1	Navodilo	2
2	Opis problema	2
3	Opis dela	3
3.1	Generiranje podatkov	3
3.2	Algoritmi	3
3.3	Analiza rezultatov	4
3.3.1	Analiza algoritmov na grafih $G(50, 0.3)$	4
3.3.2	Primerjava algoritmov na grafih s konstantno verjetnostjo p in spremenljivim številom vozlišč n	6
3.3.3	Primerjava algoritmov na grafih s konstantnim številom vozlišč n in spremenljivo verjetnostjo p	8
4	Zaključek	9
	Literatura	10

1 Navodilo

Naloga je iskanje največje neodvisne množice v grafu $G = (V, E)$ s pomočjo celoštevilkega linearnega programiranja. Velike neodvisne množice v grafu lahko poiščemo s pomočjo metode lokalnega iskanja. Začnemo s poljubno neodvisno množico $U \subseteq V$, kjer k vozlišč nadomestimo s $k + 1$ vozlišči tako, da ohranjamo neodvisnost množice U . Konstanta k je dana na začetku. Primerjali bomo metodi lokalnega iskanja in optimalne rešitve ter primerjali njune rešitve za nekatere preproste grafe.

2 Opis problema

Definicija 1. Naj bo $G = (V, E)$ graf. **Neodvisna množica** U , v grafu G , je taka podmnožica množice vozlišč V , kjer poljubni dve vozlišči iz množice U nista sosednji. **Maksimalna neodvisna množica** v grafu G pa je taka neodvisna množica, kjer ne obstaja vozlišče $v \in V$ in $v \notin U$, ki bi ga lahko dodali množici U in pri tem ohranili neodvisnost množice U . Torej je neodvisna množica U največja taka, če velja ena od naslednjih dveh lastnosti:

1. $v \in U$
2. $S(v) \cap U \neq \emptyset$, kjer je $S(v)$ množica sosedov v .

Največja neodvisna množica je neodvisna množica, največje možne velikosti, za dan graf G . Velikosti največje neodvisne množice, za graf G , pa pravimo **neodvisnostno število** in pogosto označimo $\alpha(G)$.

Definicija 2. Celostevilski linearni program v standardni obliki je dan z matriko $A \in \mathbb{R}^{m \times n}$, vektorjem $b \in \mathbb{R}^m$ in vektorjem $c \in \mathbb{R}^n$. Iščemo

$$\max \langle c, x \rangle,$$

da bodo zadoščeni pogoji

$$Ax \leq b, x \geq 0,$$

kjer je $x \in \mathbb{Z}^n$.

Posledica 1. Problem največje neodvisne množice v grafu $G = (V, E)$ lahko s celoštevilskim linearnim programiranjem modeliramo na sledeč način:

$$\max \sum_{v \in V} x_v,$$

da velja:

$$x_v + x_w \leq 1 \text{ za } \forall vw \in E,$$

$$x_v \in \{0, 1\}$$

$$x_u = \begin{cases} 1, & \text{za } u \in U \\ 0, & \text{za } u \notin U \end{cases}, U \text{ neodvisna množica v grafu } G.$$

Največjo neodvisno množico v množici vseh neodvisnih podmnožic grafa $G = (V, E)$ bomo iskali s pomočjo celoštevilkega linearnega programiranja in lokalnega iskanja. **Lokalno iskanje** temelji na izbiri začetne neodvisne podmnožice vozlišč $U \subset V$ v kateri k vozlišč zamenjamo s $k + 1$ vozlišči in pri tem ohranjamo neodvisnost množice U .

3 Opis dela

V nadaljevanju bomo največjo oz. maksimalno množico iskali s pomočjo *CLP*, v Sage, in s pomočjo implementiranega algoritma *nakljucni_MIS*(G) kasneje izboljšanega z lokalnim iskanjem. Algoritme bomo izvajali na grafih, generiranih s pomočjo Erdős-Rényijevega $G(n, p)$ modela. Primerjali bomo maksimalne (ali pa največje) neodvisne množice, ki jih algoritmi poiščejo, časovno zahtevnost algoritmov, kasneje pa še, kako vpliva spreminjanje števila vozlišč in verjetnosti, v modelu, na velikost maksimalne (ali pa največje) neodvisne množice in časovno zahtevnost algoritmov.

3.1 Generiranje podatkov

Kot omenjeno, bomo grafe generirali s pomočjo Erdős-Rényijevega $G(n, p)$ modela.

Definicija 3. *Erdős-Rényijev model $G(n, p)$ generira graf z n naključno povezanimi vozlišči. Vsaka povezava je v graf vključena neodvisno, z verjetnostjo p .*

Generirali bomo grafe s konstantima (n, p) , z naraščajočim n in konstantnim p ter s konstantim n in naraščajočim p . Za generacijo podatkov v Pythonu bomo uporabili knjižnico *NetworkX*, nato pa objekte grafov, s funkcijo *nx.to_dict_of_lists*(*graf*), kjer je *nx* okrajšava za *NetworkX*, pretvorili še v slovarje list, da bomo lahko grafe uporabljali še v okolju *Sage*. Grafe, na katerih bomo izvajali algoritme, bomo shranili še v *JSON* datoteke.

3.2 Algoritmi

Za iskanje maksimalne neodvisne množice smo implementirali algoritem *nakljucni_MIS*(G), čigar rešitev bomo poizkušali izboljšati še z algoritmom *lokalno_iskanje*(G, I). Oba algoritma bosta kot argument sprejela graf G , algoritem *lokalno_iskanje*(G, I) pa še neko maksimalno neodvisno množico I , grafa G .

Algoritem 1 *nakljucni_MIS*(G)

```
1:  $I \leftarrow \emptyset$ 
2:  $\forall v \in V$  dobi vrednost  $P(v) \in \text{permutacija}(V)$ 
3: if  $P(v) < P(w)$  za  $\forall w \in \text{sosedi}(v)$  then
4:    $I \leftarrow I \cup v$ 
5:  $V' \leftarrow V \setminus (I \cup \text{sosedi}(I))$ .
6:  $E' \leftarrow E \setminus \text{povezave}(I)$ .
7: return  $I \cup \text{MIS}(G' = (V', E'))$ 
```

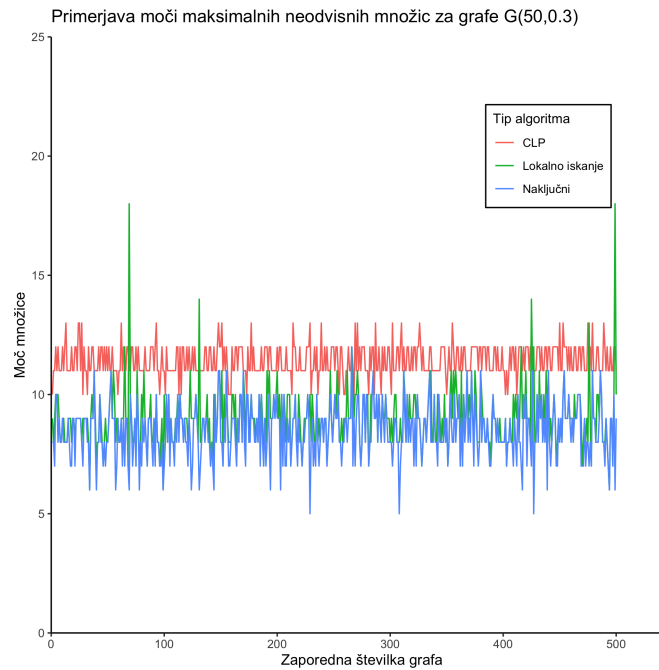
Algoritem 2 *lokalno_iskanje*(G, I)

```
1:  $I \leftarrow \emptyset$ 
2:  $\forall v \in V$  dobi vrednost  $P(v) \in \text{permutacija}(V)$ 
3: if  $P(v) < P(w)$  za  $\forall w \in \text{sosedi}(v)$  then
4:    $I \leftarrow I \cup v$ 
5:  $V' \leftarrow V \setminus (I \cup \text{sosedi}(I))$ .
6:  $E' \leftarrow E \setminus \text{povezave}(I)$ .
7: return  $I \cup \text{MIS}(G' = (V', E'))$ 
```

3.3 Analiza rezultatov

3.3.1 Analiza algoritmov na grafih $G(50, 0.3)$

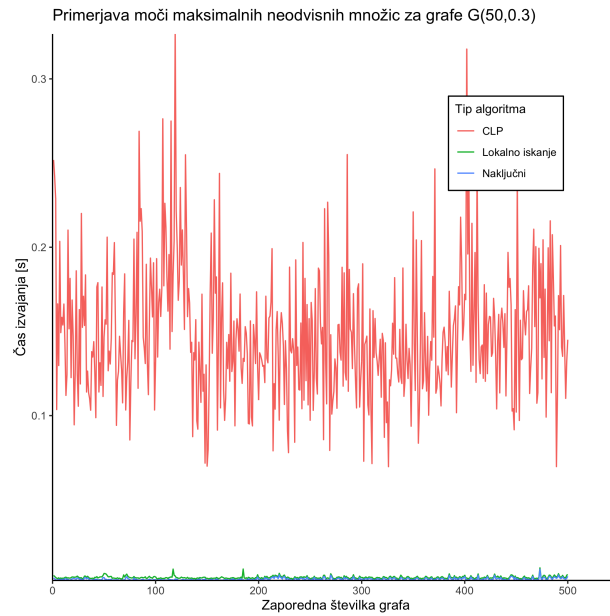
Najprej bomo primerjali algoritme na Erdős-Rényijevih $G(50, 0.3)$. Na tak način sva generirala 500 grafov in na njih izvedla algoritma $CLP(G)$, $\text{naključni_MIS}(G)$, za vsak graf pa sva slednjega poskušala izboljšati še z algoritmom $\text{lokalno_iskanje}(G, I)$, ki poleg grafa G sprejme še $I = \text{naključni_MIS}(G)$.



Slika 1: Moči neodvisnih množic za grafe $G(50, 0.3)$

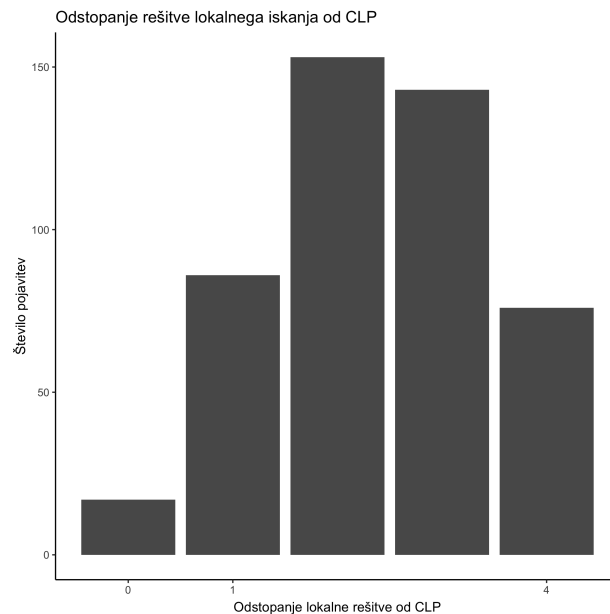
Opazimo, da neodvisne množice največjih moči najde CLP . Kljub temu so na grafu opazni osamelci, ko $\text{lokalno_iskanje}(G, I)$ vrne neodvisne množice izstopajočih moči. Teh pojavov si ne znava najboljše razložiti. Bodisi CLP nikoli ne vrne največje neodvisne množice, neodvisna množica $I = \text{naključni_MIS}(G)$ pa je v teh primerih odlična za lokalno iskanje.

Kljub temu, da je, za generirane grafe, *CLP* najpogosteje vrnil najboljše rešitev, lahko v naslednjem grafu vidimo njegovo slabost, *CLP* je občutno počasnejši od preostalih dveh algoritmov.



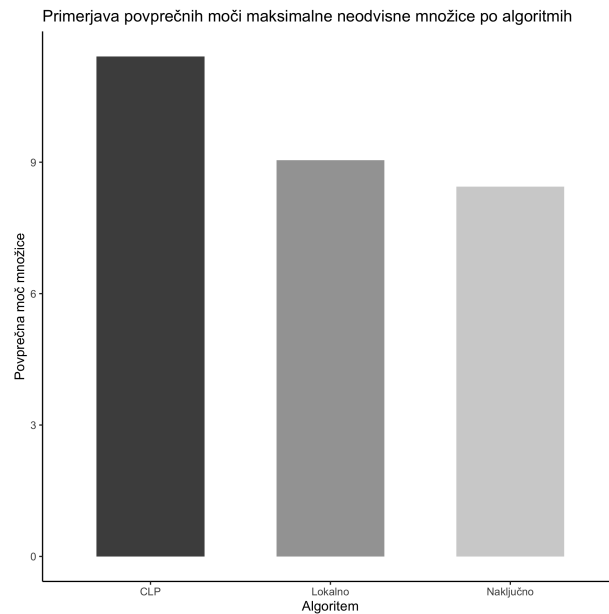
Slika 2: Časovne zahtevnosti algoritmov za $G(50,0.3)$

Če si pogledamo še odstopanja rešitev *lokalnega iskanja* od *CLP* opazimo, da je bila rešitev *lokalnega iskanja* največkrat za dve vozlišči manjša od rešitve *CLP*.



Slika 3: Odstopanja rešitev lokalnega iskanja do CLP.

Kot omenjeno, je maksimalne neodvisne množice z najvišjimi močmi vračal *CLP*, povprečju so bile te množice velikosti 12. Druge največje množice je vračalo *lokalno iskanje*, katerega množice so imele v povprečju 9 vozlišč, za eno več kot od povprečja moči neodvisnih množic pridobljenih z algoritmom *naključni MIS*. Rezultati so vidni na spodnjem grafu.

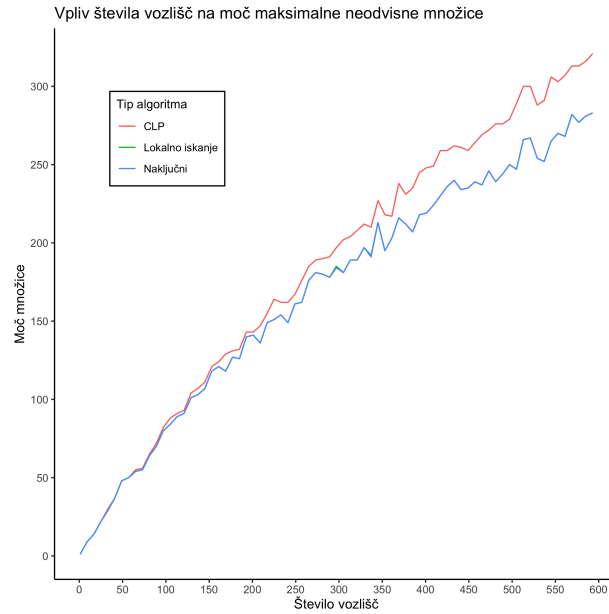


Slika 4: Povprečne vrednosti rezultatov posameznih algoritmov

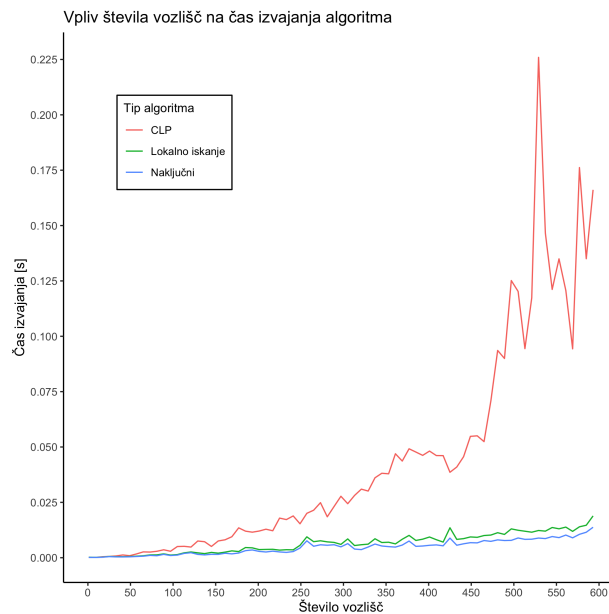
3.3.2 Primerjava algoritmov na grafih s konstantno verjetnostjo p in spremenljivim številom vozlišč n

Drugo skupino grafov sva, tako kot prvo, generirala z Erdős-Rényijevim $G(n, p)$ modelom, vendar pri tem spreminjala število vozlišč v grafih, verjetnost pa ohranila konstantno $p = 0.05$. Algoritme sva izvajala na grafih velikosti od 1 do 600 vozlišč.

Na spodnjem grafu si lahko ogledamo kako se, s povečevanjem števila vozlišč, spreminja moč maksimalnih neodvisnih množic. Pri konstantni verjetnosti p , se moč maksimalne neodvisne množice, v odvisnosti od moči množice vozlišč V , povečuje skoraj linearno, vendar moč maksimalnih neodvisnih množic dobljenih s *CLP* narašča hitreje od tistih pridobljenih z *naključni MIS* in *lokalnim iskanjem*.



Slika 5: Velikosti neodvisnih množic v odvisnosti od števila vozlišč n



Slika 6: Spreminjanje časovne zahtevnosti algoritmov v odvisnosti od števila vozlišč n

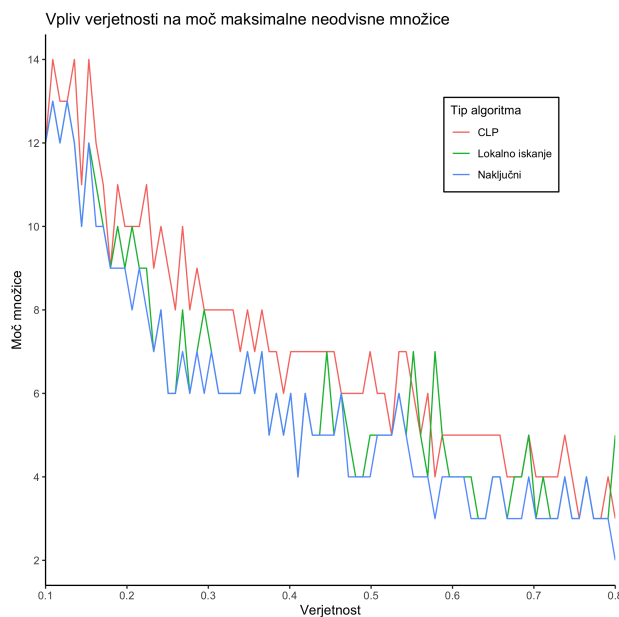
Na zgornjem grafu ponovno opazimo največjo slabost iskanja maksimalne neodvisne množice s *CLP*, tj. časovno zahtevnost. Čas, porabljen za izvajanje *CLP* je, v odvisnosti od števila vozlišč v množici V , naraščala eksponentno. Prav tako je naraščala časovna zahtevnost algoritma *naključni_MIS* in *lokalnega_iskanja*, vendar so bila naraščanja, na opazovanih grafih, občutno manjša. Glede na to, da je iskanje maksimalne neodvisne množice *NP*-težek problem lahko pričakujemo, da

bi z nadaljnjim povečevanjem števila vozlišč v množici V , časovna zahtevnost začela naraščati eksponentno tudi za zadnja dva algoritma. Zаметki tega so opazni, ko gledamo časovne zahtevnosti algoritmov *naključni_MIS* in *lokalnega iskanja* za največje moči množice V .

3.3.3 Primerjava algoritmov na grafih s konstantnim številom vozlišč n in spremenljivo verjetnostjo p

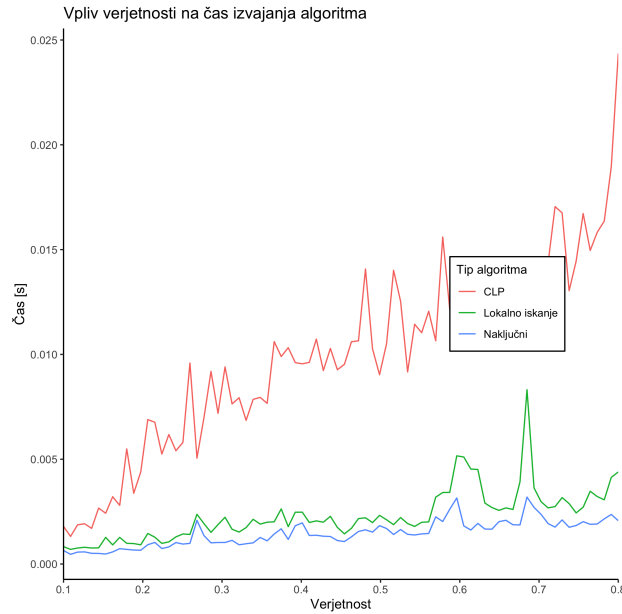
Tudi zadnjo skupino grafov sva generirala z Erdős-Rényijevim $G(n, p)$ modelom, pri tem pa ohranila konstantno velikost množice vozlišč V in spreminjala verjetnosti p . Vrednosti verjetnosti sva dobila s klicom funkcije `numpy.linspace(0.1, 0.8, 80)`.

Če je, s povečevanjem velikosti množice V , naraščala tudi velikost maksimalne neodvisne množice, opazimo, da je v primeru povečevanja verjetnosti p vzorec nasproten, saj velikost maksimalne neodvisne množice pada, ko povečujemo verjetnost p in posledično število povezav v grafu. Torej velikost maksimalne neodvisne množice ni odvisna le od števila vozlišč v grafu, ampak tudi od števila povezav. Ugotovitve so opazne na spodnjem grafu.



Slika 7: Vpliv verjetnosti na velikost neodvisnih množic

Tako kot velikost maksimalne neodvisne množice, je od števila povezav v grafu (oz. verjetnosti p), odvisna tudi časovna zahtevnost algoritmov. Časovna zahtevnost algoritmov ob povečevanju števila povezav v grafu se obnaša podobno, kot se je obnašala pri povečevanju vozlišč. Časovna zahtevnost vseh treh algoritmov narašča skupaj s verjetnostjo p , vendar pri *CLP* hitreje kot pri *naključni_MIS* in *lokalnim iskanjem*.



Slika 8: Vpliv verjetnosti na časovno zahtevnost algoritmov

4 Zaključek

Iz analize rezultatov lahko zaključimo, da v primeru generiranja grafov z Erdős-Rényijevim $G(n, p)$ modelom, na velikost maksimalne neodvisne množice v danem grafu vplivata tako število vozlišč v grafu, kot tudi število povezav. Kot alternativo iskanju maksimalne neodvisne množice s *CLP*-jem, smo uspeli implementirati algoritem *naključni_MIS* in ga kasneje izboljšati z lokalnim iskanjem. Čeprav so rešitve *CLP* malenkost boljše v večini primerov, sta *naključni_MIS* in *lokalno iskanje* boljša za grafe z večjim številom vozlišč, ali pa z večjim številom povezav, saj sta manj časovno zahtevna.

Literatura

- [1] Gary Miller. *Lecture 32: Luby's Algorithm for Maximal Independent Set*, dostopno na <http://www.cs.cmu.edu/afs/cs/academic/class/15750-s18/ScribeNotes/lecture32.pdf>.
- [2] Diogo Andrade, Mauricio G. C. Resende. *Fast Local Search for the Maximum Independent Set Problem*. Conference Paper in Journal of Heuristics, May 2008, dostopno na https://www.researchgate.net/publication/221131653_Fast_Local_Search_for_the_Maximum_Independent_Set_Problem.
- [3] *Maximal independent set*, v: Wikipedia: The Free Encyclopedia,[ogled 6. 1. 2022], dostopno na https://en.wikipedia.org/wiki/Maximal_independent_set.
- [4] *Independent set (graph theory)*, v: Wikipedia: The Free Encyclopedia,[ogled 6. 1. 2022], dostopno na [https://en.wikipedia.org/wiki/Independent_set_\(graph_theory\)](https://en.wikipedia.org/wiki/Independent_set_(graph_theory)).