

Univerza v Ljubljani  
Fakulteta za matematiko in fiziko

Finančni praktikum

# **Lokalno iregularno barvanje povezav**

Domen Flakus Bosilj in Lara Jagodnik

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

Ljubljana, 2020

# Kazalo

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Kaj sva delala in kako dela algoritem</b>	<b>3</b>
2.1	Ugotovitve po poganjanju . . . . .	4
<b>3</b>	<b>Prvo izboljšanje algoritma</b>	<b>4</b>
3.1	Ugotovitve po prvem izboljšanju . . . . .	4
<b>4</b>	<b>Drugo izboljšanje funkcije</b>	<b>6</b>
4.1	Ugotovitve po drugem izboljšanju . . . . .	6
<b>5</b>	<b>Zaključek</b>	<b>10</b>

# 1 Uvod

V projektu pri finančnem praktikumu si bova ogledala lokalno iregularno barvanje povezav. Pri obravnavi problema bova delala z neusmerjenimi grafi. Projekt bova izdelala v programu Sage.

Graf  $G$  je lokalno iregularen, če imata vsaki dve sosednji vozlišči  $u$  in  $v$  različno stopnjo,  $\deg(u) \neq \deg(v)$ . (Stopnja vozlišča nam pove število povezav, ki potekajo iz vozlišča.)

Barvanje povezav je lokalno iregularno, če vsaka barva inducira lokalno iregularen graf.

V projektu bova preverila ali drži domneva, da je vsak kubičen graf lokalno iregularno 3-obarvljiv. To pomeni, da lahko graf pobarvamo s tremi barvami tako, da razpade na tri lokalno iregularne grafe. (Kubični graf je graf v katerem imajo vsa vozlišča stopnjo 3.)

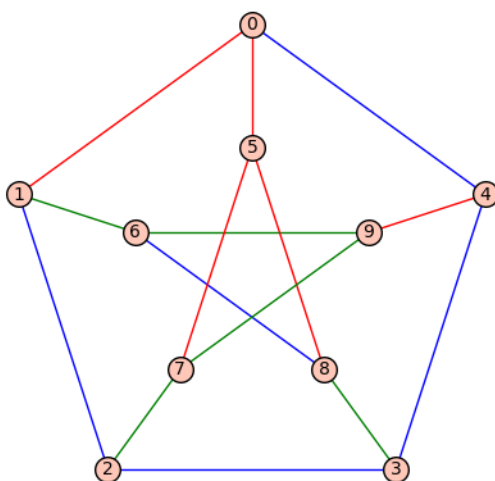
Lastnosti grafov, ki so povezani z najino temo:

- Poti lihe dolžine ne dopuščajo lokalno iregularnega barvanja.
- Graf je "decomposable" če dopušča lokalno iregularno barvanje.
- Najmanjši  $k$  za katerega obstaja lokalno iregularno barvanje v grafu  $G$  s  $k$  barvami se imenuje lokalni iregularni kromatični indeks in ga označimo z  $x'_{irr}(G)$
- Naj bo  $G$  "decomposable" graf. Potem je  $x'_{irr}(G) \leq 328$ .
- Domneva, da je  $x'_{irr}(G) \leq 3$  je že dokazana za posebne vrste grafov, kjer je najmanjša stopnja vozlišč vsaj  $10^{10}$  in za  $k$ -regularne grafe, kjer je  $k \geq 10^7$ .

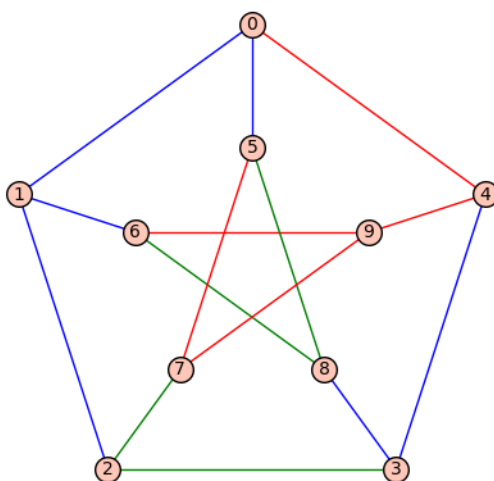
## 2 Kaj sva delala in kako dela algoritem

Problem sva najprej obravnavala na manjših grafih (do vključno 12 vozlišč), ki sva jih generirala s pomočjo funkcije `graphs.nauty_geng`. Ker pri tako majhnem številu vozlišč grafov še ni veliko, (kubičnih grafov na 10 vozliščih je 19, na 12 vozliščih pa 85) sva pregledala čisto vse možne.

Grafe sva naključno pobarvala, nato sva poiskala povezave, kjer se pojavi težava. Najprej sva nameravala v vsakem obhodu funkcije čez vse povezave in ob tem pogledati ali se barvo spleta spremeniti. Ker sva ugotovila da tak algoritem na velikih grafih deluje zelo počasi, sva funkcijo randomizirala. Barvanje sva nato spreminjala tako, da sva za naključno povezavo pogledala, ali sprememba barve te povezave morda zmanjša število povezav, ki ne ustrezajo pogoju lokalne iregularnosti grafa. Če je bilo barvanje neuspešno, sva na istem grafu poskusila začeti z novim drugačnim barvanjem in v najslabšem primeru to ponovila 10-krat.



Slika 1: Naključno pobarvan *Petersenov graf*



Slika 2: Lokalno iregularno  
pobarvan *Petersenov graf*

Za primer prikaza delovanja algoritma sva vzela najbolj znan kubičen graf na 10 vozliščih, to je *Petersenov graf*. Na sliki 1 je prikazano naključno začetno barvanje, na sliki 2 pa končno barvanje ki je lokalno iregularno.

Nato sva konstruirala naključne "velike" kubične grafe z `graphs.RandomRegular`, na vozliščih 14, 16, 18, 20 in število vozlišč večala za 10 do 70. Tu nisva pregledala vseh možnih različnih grafov na določenem številu vozlišč ampak le naključnih 10. Popravljanje začetnega naključnega barvanja pa sva nadaljevala kot pri majhnih grafih. Ker je bil algoritem za grafe na 70 vozliščih

precej počasen sva tu dobila le rezultate šestih grafov, kjer noben ni bil uspešno pobarvan.

Grafe, ki jih najin algoritem ni uspel pobarvati tako, da bi ustrezali pogoju lokalne iregularnosti, sva shranila, saj le ti predstavljajo potencialen protiprimer, da domneva, ki sva jo preverjala, torej da je vse kubične grafe možno pobarvati s tremi barvami tako, da so lokalno iregularni, ne drži.

## 2.1 Ugotovitve po poganjanju

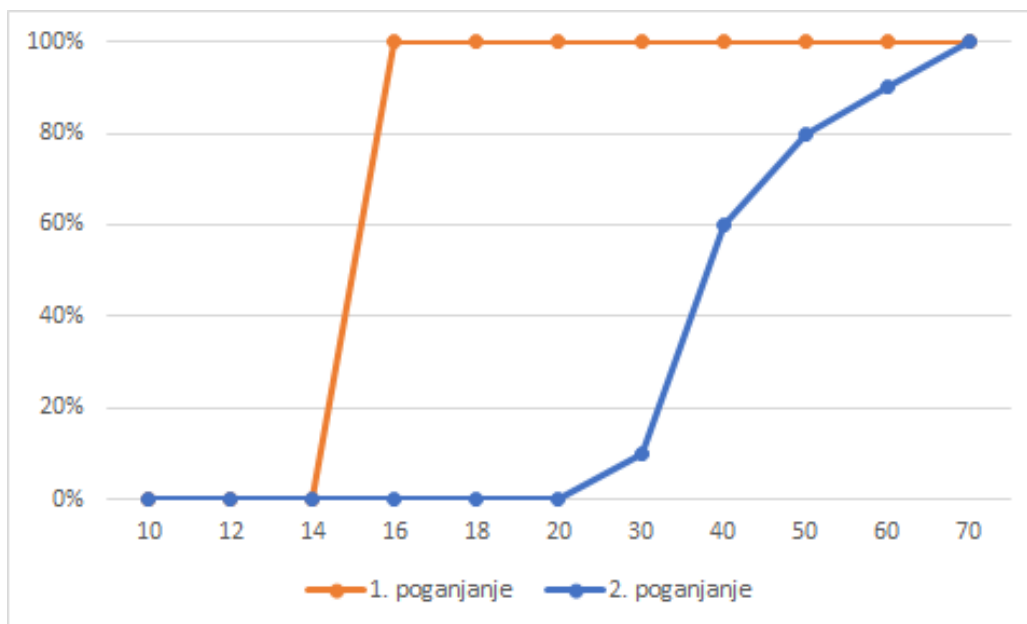
Vse majhne grafe sva uspela lokalno iregularno pobarvati. Tudi z grafi na 14 vozlišči ni bilo težav, saj je bil algoritem v vseh poskusih uspešen. Težave so se začele pojavljati pri grafih na 16 vozliščih, kjer najin algoritem ni uspel pobarvati nobenega od desetih slučajnih grafov. Prav tako barvanje ni uspelo na nobenem večjem grafu. V seznamu nepravilno pobarvanih grafov sva tako imela 76

## 3 Prvo izboljšanje algoritma

Začetek funkcije sva ohranila in dodala izboljšave. Če po 3000 poskusih nisva uspela dobiti pravilnega barvanja, sva na najboljšem barvanju grafa, ki sva ga dobila izvedla še lokalno spreminjanje. Če je bila napaka na povezavi  $uv$  sva pogledala, ali je možno kaj izboljšati na povezavah, ki imajo vsaj eno kajišče  $u$  ali  $v$ . S tem sva uspela algoritem precej izboljšati, saj se je glede na najine teste to izkazalo za najhitrejšo in najbolj učinkovito metodo. Če je funkcija `lokalno_spreminjanje` graf izboljšala, ni pa uspela odpraviti vseh težav, sva še enkrat poskusila z naključnim spreminjanjem barv, kjer je to mogoče.

### 3.1 Ugotovitve po prvem izboljšanju

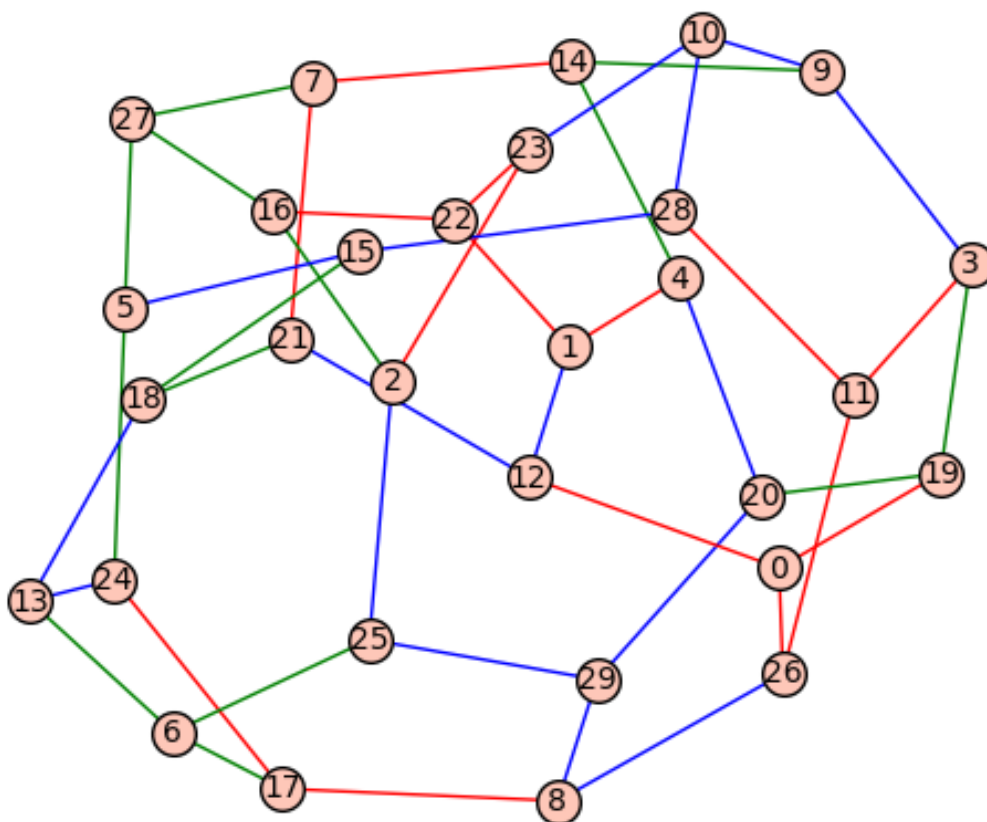
Ta funkcija je hitrejša in bolje deluje tudi na velikih grafih, kjer se je uspešnost pravilnega barvanja na naključnih desetih grafih precej zvečala.



Slika 3: Število nepravilno pobarvanih grafov pred in po prvem izboljšanju algoritma

Slika 3 prikazuje število nepravilno pobarvanih grafov s funkcijo pred in po prvi izboljšavi. Vidimo, da nobena od funkcij do 14 vozlišč ni imela težav, le-te pa so se začele pred izboljšavami pojavljati že pri 16 vozliščih, kjer algoritem ni znal pravilno pobarvati nobenega grafa. Izboljšana funkcija pa je bila v vseh primerih uspešna vse do 20 vozlišč in tudi na več vozliščih je bila veliko boljša od prejšnje.

Prvi graf ki ga funkcija ni uspela pravilno pobarvati je bil na 30 vozliščih, zato sva ga dodala na seznam potencialnih protiprimerov, prikazan je na sliki 4. Graf je neuspešno pobarvan zaradi modre povezave (15, 28), saj imata zaradi tega obe prej omenjeni vozlišči po 2 povezavi modre barve. Potem je s številom vozlišč število neuspešno pobarvanih grafov strmo narastlo, kar bi lahko pomenilo, da nekaj grafov ni mogoče lokalno iregularno pobarvati s 3 barvami, možno pa je tudi, da je to mogoče vendar bi potrebovali zelo veliko časa ali nekoliko boljši algoritem. V seznamu neuspešno pobarvanih grafov je tako ostalo še 30 grafov.



Slika 4: Nepravilno pobarvan graf na 30 vozliščih, ki ga prvič izboljšana funkcija ni uspela pobarvati

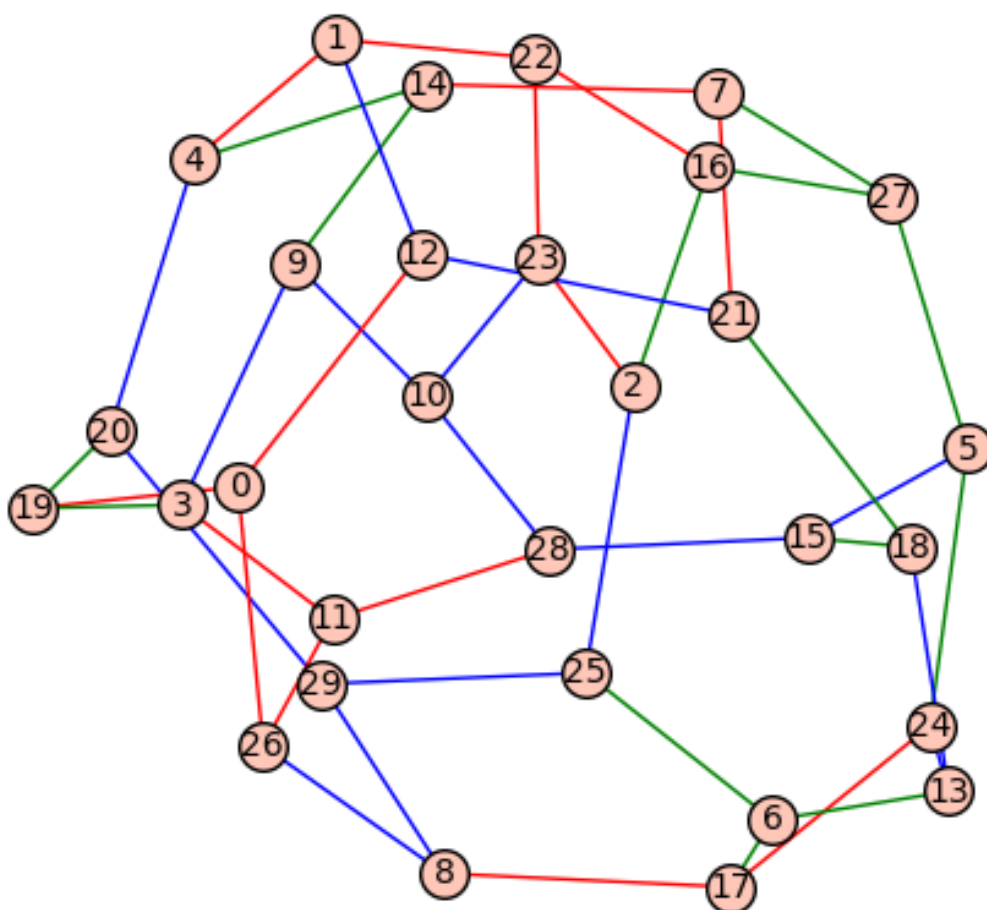
## 4 Drugo izboljšanje funkcije

Funkcijo sva spremenila tako, da je število, korakov, namesto fiksnih 3000, odvisno od števila vozlišč v grafu in sicer tako, da je število korakov enako produktu  $100 * \text{število vozlišč}$ . To sva uporabila le za velike grafe, saj se le tu še pojavljajo težave, število naključnih grafov, pa sva iz 10 zmanjšala na 5. Poleg tega sva s tako spremenjeno funkcijo poizkusila pravilno pobarvati še grafe, ki nama jih prej ni uspelo.

### 4.1 Ugotovitve po drugem izboljšanju

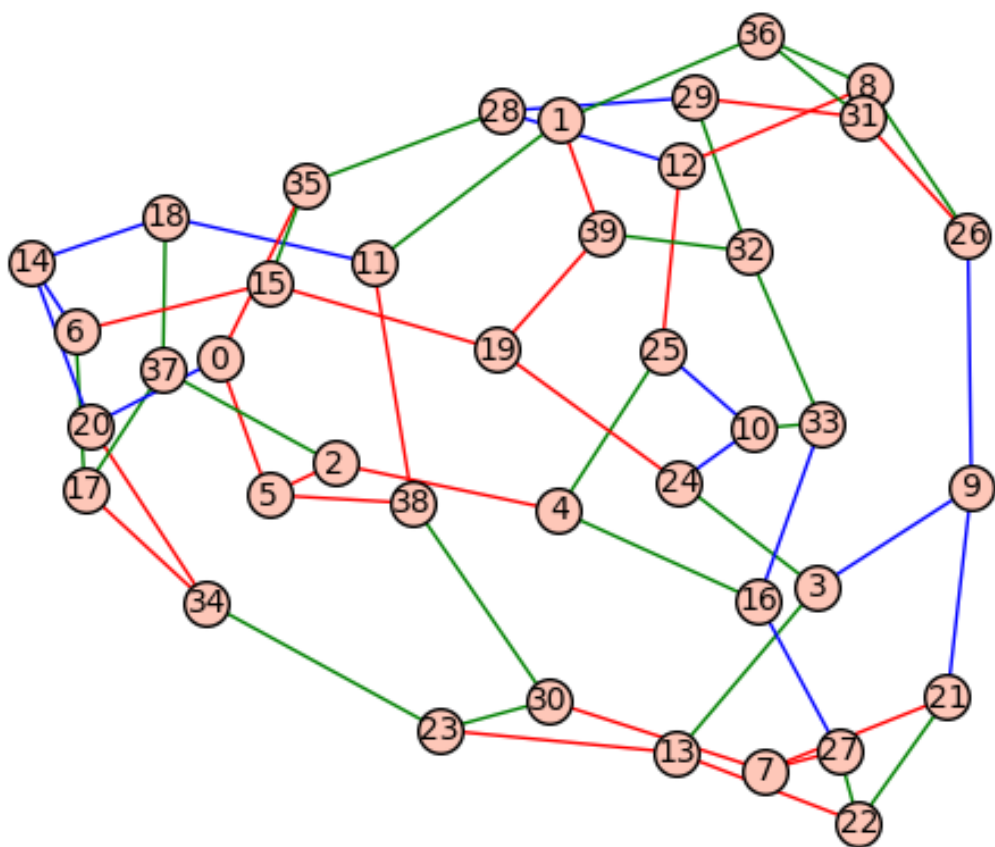
S tem algoritmom, sva uspela pravilno pobarvati vseh pet slučajnih grafov, ki jih je generiralo na vozliščih od 14 do 20. Težave so ze začele pojavljati pri grafih s 30 vozlišči, kjer 3 od 5 grafov ni bilo uspešno pobarvanih. Presenetilo

Pri poganjanju funkcije na preostalih 30 grafih, ki še niso bili uspešno pobarvani, je funkcija tokrat pravilno uspela pobarvati prejšnji graf na 30 vozličih, ki torej ne predstavlja potencialnega protiprimera. Pravilno barvanje je prikazano na sliki 5. Algoritem je bil uspešen še na 2 od 6 grafov na 40 vozliščih in 3 od 8 na 50 vozliščih. Na večjih grafih, je bil tudi tokrat algoritem neuspešen. V seznamu neuspešno pobarvanih grafov je tako ostalo še 24 grafov. najmanjši pa je na 40 vozliščih, eden od še vedno neuspešno pobarvanih grafov na 40 vozliščih je prikazan na sliki 6, graf ima le eno napako in sicer na zeleni povezavi (23, 30).



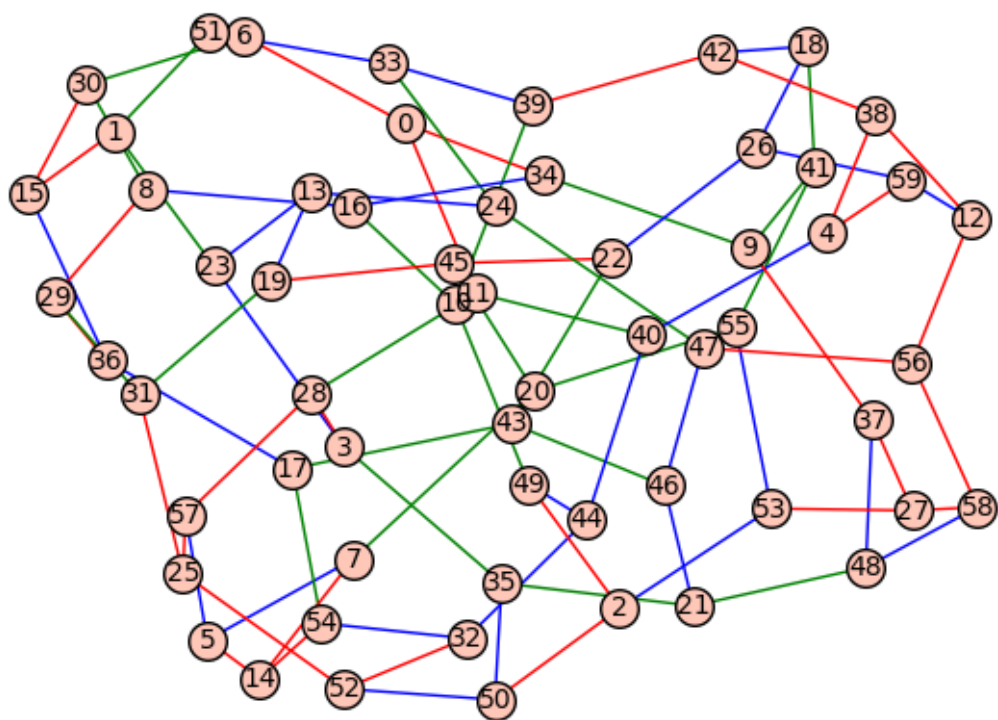
7





Slika 6: Graf na 40 vozliščih, ki ga funkcija ni uspela lokalno iregularno pobarvati

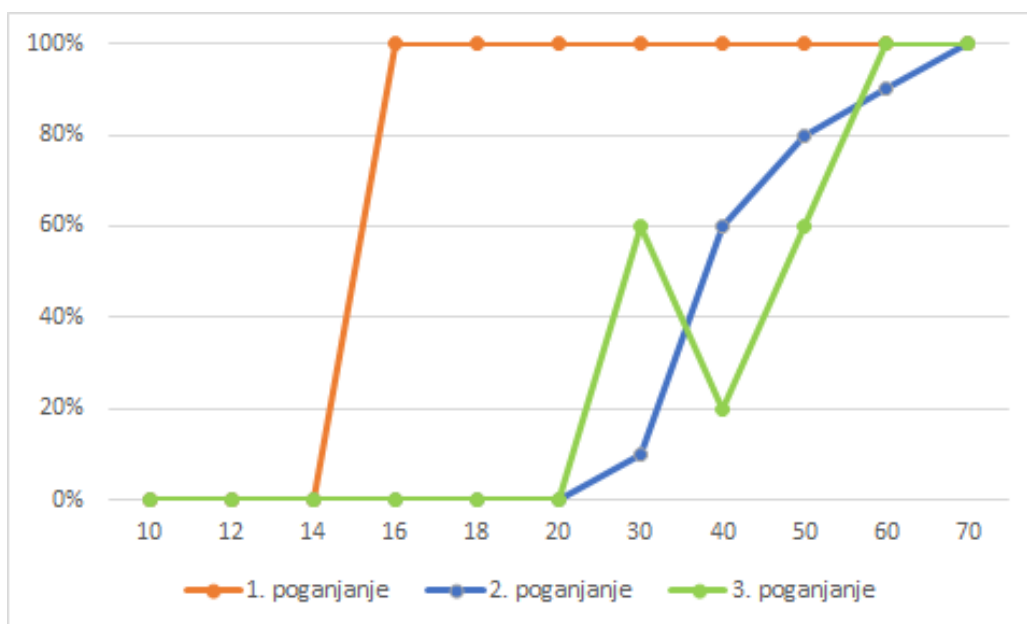
Za največji graf sva na sliki 7 prikazala graf na 60 vozliščih, saj za graf na 70 vozliščih nisva uspela dobiti grafa, kjer bi bile napake dovolj razvidne. Od 20 grafov, ki sva jih generirala in poskusila pravilno pobarvati je najin algoritem število napak v najboljšem primeru uspel zmanjšati na 2 napaki. Napaki sta ostali na povezavi (28, 57), kjer imata obe vozlišči po dve rdeči povezavi, in na povezavi (21, 35), kjer imata obe vozlišči po dve zeleni povezavi.



Slika 7: Graf na 60 vozliščih, ki ga funkcija ni uspela lokalno iregularno pobarvati

## 5 Zaključek

Algoritem, ki sva ga napisala je za manjše grafe precej uspešen, za grafe na vozliščih od 30 naprej pa deluje nekoliko slabše. Kot prikazuje slika 8 je bil začetni algoritem skoraj neuspešen, uspešnost po prvem in drugem popravljajanju pa je bila precej podobna, s presenetljivim odstopanjem za grafe na 40 vozliščih. Za grafe na 70 vozliščih noben algoritem ni bil uspešen.



Slika 8: Število nepravilno pobarvanih grafov po treh poganjanjih

Po najinih ugotovitvah njegova uspešnost ni veliko odvisna od števila začetnih korakov, kolikokrat funkcija naključno izbere povezavo in jo poizkusi spremeniti, če je le to število dovolj veliko. Je pa precej odvisna od začetnega naključnega barvanja grafa. Ko sva povečala število možnih začetnih naključnih barvanj, se je namreč tudi uspešnost pravilnega barvanja povečala. Zdi se nama, da bi z zadostnim številom poskusov barvanja grafov uspeli pobarvati večino na 30, 40 in 50 vozliščih, kasneje pa se res lahko pojavijo težave, kot kažejo najini potencialni protiprimeri, ki jih je na grafih z veliko vozlišči ostalo kar precej.