Finančni praktikum

# Conjucture on annihilation vs. domination


Avtorici:

Eva Rozman, Lana Zajc

Univerza v Ljubljani

Fakulteta za matematiko in fiziko


Januar, 2020

# Kazalo

# 1 Uvod

V sklopu najinega projekta sva testirali domnevo, da vsak povezan netrivialen graf G zadošča neenakosti

$$\gamma_t(G) \leq a(G) + 1,$$

kjer je $\gamma_t(G)$ dominacijsko število grafa G, $a(G)$ pa anihilacijsko število grafa G.

Neenakost sva najprej preverili za netrivialne povezane grafe z majhnim številom vozlišč. Začeli sva z vsemi grafi na treh vozliščih, nato pa število vozlišč povečevali, dokler je program neenakost še lahko preveril za vse grafe. To se je zgodilo za grafe na osmih vozliščih.

Za grafe z velikim številom vozlišč sva napisali algoritem, ki generira naključni graf z večjim številom vozlišč in predpisanim številom povezav, izračuna razliko $a(G) + 1 - \gamma_t(G)$, nato pa z neko verjetnostjo grafu odstrani oz. doda naključno povezavo. Ponovno izračuna $a(G) + 1 - \gamma_t(G)$ in preveri, ali je razlika spremenjenega grafa manjša. Če je, se algoritem zažene na novem grafu. S testiranjem sva za različna števila vozlišč poskušali poiskati minimum in maksimum razlike $a(G) + 1 - \gamma_t(G)$ in poiskati skupne karakteristike grafov, pri katerih sta ta dosežena.

V splošnem naju je zanimalo, ali lahko najdeva graf, kjer je razlika $a(G) + 1 - \gamma_t(G) = 0$, ali celo graf, kjer je $a(G) + 1 - \gamma_t(G) < 0$, saj bi s tem hipotezo ovrgli.

Za testiranje sva uporabili orodje *CoCalc*.

# 2 Razlaga pojmov

## 2.1 Netrivialen povezan graf

Naj bo $G = (V, E)$ graf. $G$ je netrivialen povezan graf, če ima vsaj dve vozlišči in lahko iz poljubnega vozlišča pridemo v vsa druga vozlišča v $G$.

## 2.2 Anihilacijsko število (Annihilation number)

Naj bo $G = (V, E)$ graf, $n$ število njegovih vozlišč in $m$ število povezav. Stopnje vozlišč grafa naj bodo urejene v nepadajoče zaporedje $d_1 \leq d_2 \leq \cdots \leq d_n$. *Anihilacijsko število* $a(G)$ grafa $G$ je največje naravno število $k$, da velja:

$$d_1 + d_2 + \cdots + d_k \leq m.$$

## 2.3 Dominacijska množica grafa (Domination set)

*Dominacijska množica* $A$ grafa $G = (V, E)$ je podmnožica množice vozlišč $V$, če je vsako vozlišče iz $V$ bodisi element A, ali pa je sosednje vozlišče kakšnega vozlišča iz $A$.

## 2.4 Dominacijsko število (Domination number)

*Dominacijsko število* $\gamma_t(G)$ je število vozlišč v najmanjši dominacijski množici grafa $G$.

# 3 Rezultati testiranja za minimalno razliko

## 3.1 Testiranje na malih grafih

Najprej sva za vse povezane netrivialne grafe s številom vozlišč od 3 do 8 testirali ali neenakost $\gamma_t(G) \leq a(G) + 1$ drži. Izkaže se, da drži za vse grafe, pri čemer je minimalna razlika za grafe na 3, 4, 5 in 7 vozliščih 1, minimalna razlika za grafe na 6 in 8 vozlišč pa 2.

## 3.2 Testiranje na večjih grafih

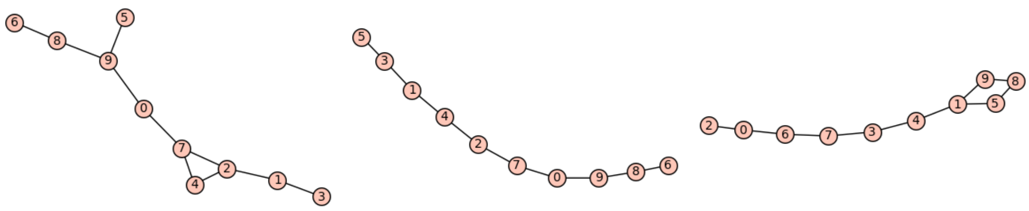
Za grafe z velikim številom vozlišč je nemogoče domnevo testirati na vseh grafih, zato sva se problema lotili z metodo simulated annealing. Najprej sva generirali naključen graf z določenim številom vozlišč in povezav, potem pa s spodnjo funkcijo naključno dodali ali odstranili povezavo.

```
from sage.graphs.connectivity import is_connected
def spremeni_graf(G):
    H = Graph(G)
    if random() < 0.5:
        i = 0
        while True:
            H.delete_edge(H.random_edge())
            if is_connected(H):
                H
                break
            else:
                H = Graph(G)
                i = i + 1
                True
            if i > 25:
                H.add_edge(H.complement().random_edge())
                break
    else:
        if H.complement().size() == 0:
            H.delete_edge(H.random_edge())
        else:
            H.add_edge(H.complement().random_edge())
    return H
```

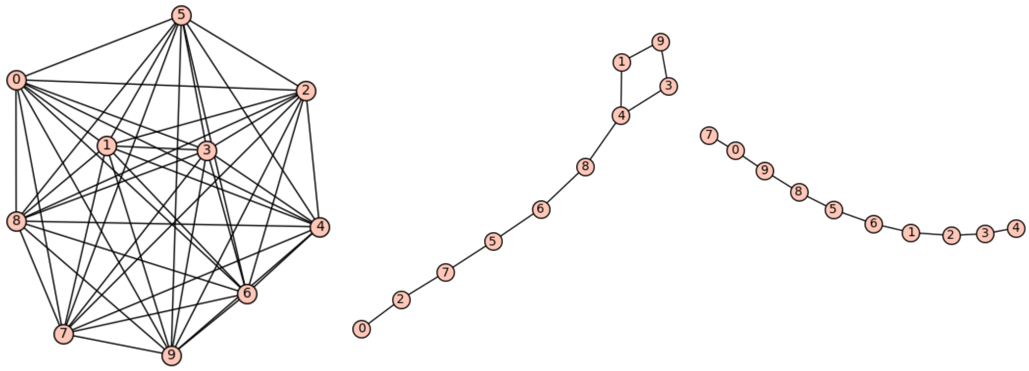
Za namen računanja minimuma razlike $a(G) + 1 - \gamma_t(G)$ sva definirali naslednjo funkcijo. Ta naredi 2000 korakov, na vsakem izračuna razliko, primerja razliki prejšnjega in novega grafa in nato spremeni graf z zgornjo funkcijo. Na koncu vrne minimalno razliko ter grafe, pri katerih je ta dosežena.

```
def najmanjse_razlike(G):
    stevilo_korakov = 2000
    trenutna_razlika = min_razlike = razlika(G)
    seznam_grafov_min = [G]
    for p in range(0, stevilo_korakov):
        T = stevilo_korakov / (p+1)
        nov_graf = spremeni_graf(G)
        nova_razlika = razlika(nov_graf)
        if nova_razlika < min_razlike:
            min_razlike = nova_razlika
            seznam_grafov_min = []
            seznam_grafov_min.append(nov_graf)
        elif nova_razlika == min_razlike:
            if all(not H.is_isomorphic(nov_graf) for H in seznam_grafov_min):
                seznam_grafov_min.append(nov_graf)
        if nova_razlika < trenutna_razlika or exp(-1 * (nova_razlika - trenutna_razlika) / T) >= random():
            G = nov_graf
            trenutna_razlika = nova_razlika

    return (min_razlike, seznam_grafov_min)
```

### 3.2.1 Grafi na 10 vozliščih



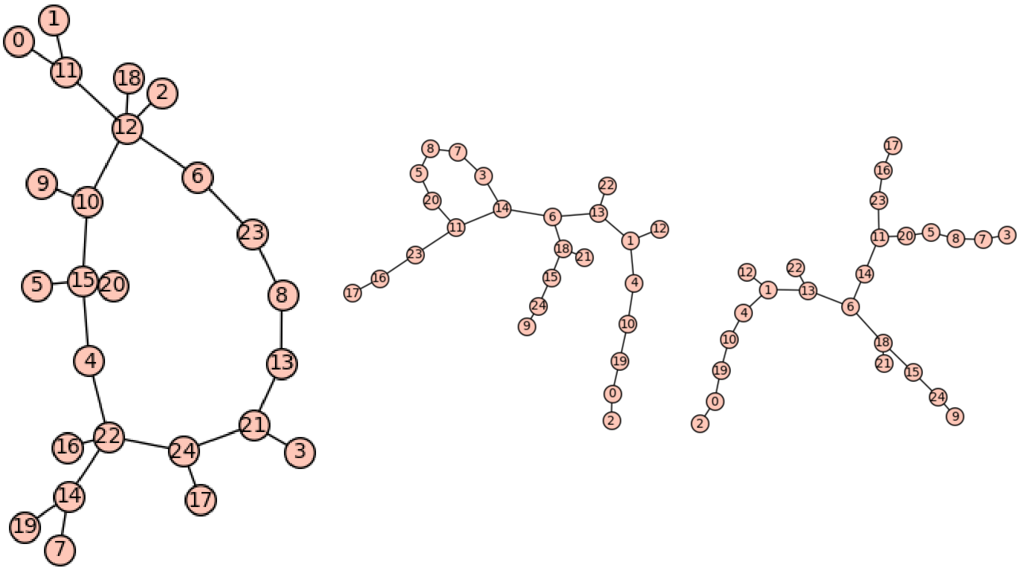
Slika 1: Začetni graf z 10 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 2$



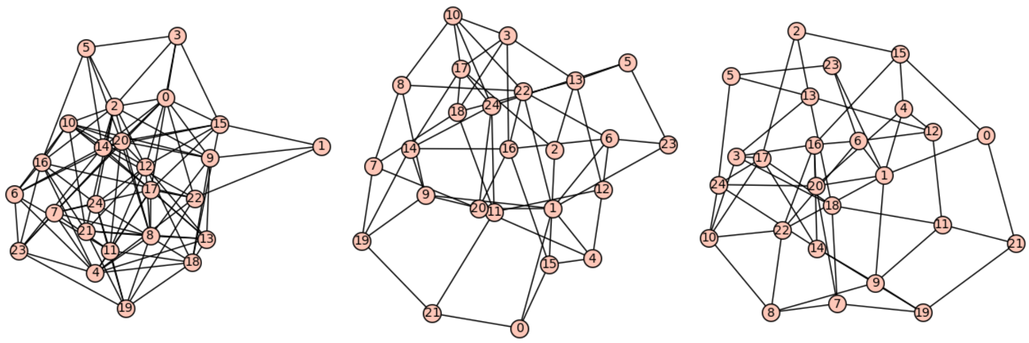
Slika 2: Začetni graf s 45 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 2$

Minimalno razliko za grafe na 10 vozliščih sva testirali za naključne grafe z 10, 20, 30 in 45 povezavami. Izkazalo se je, da je minimum razlike $a(G)+1-\gamma_t(G)$ za vsa števila začetnih povezav 2.

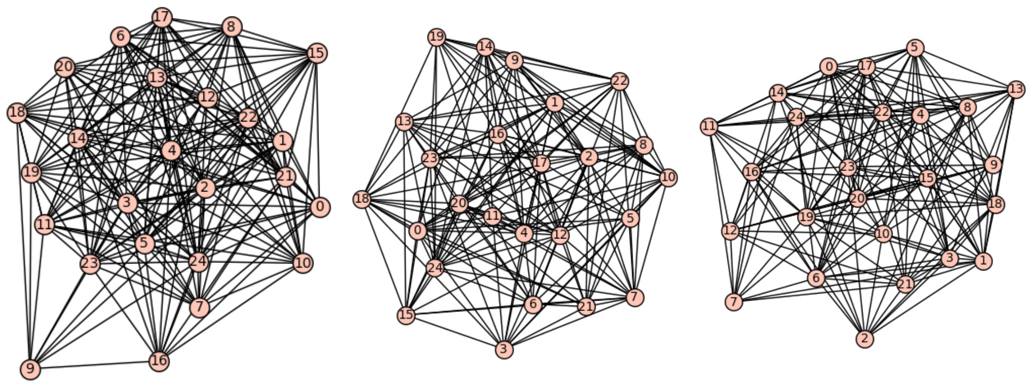
### 3.2.2 Grafi na 25 vozliščih



Slika 3: Začetni graf s 25 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 6$



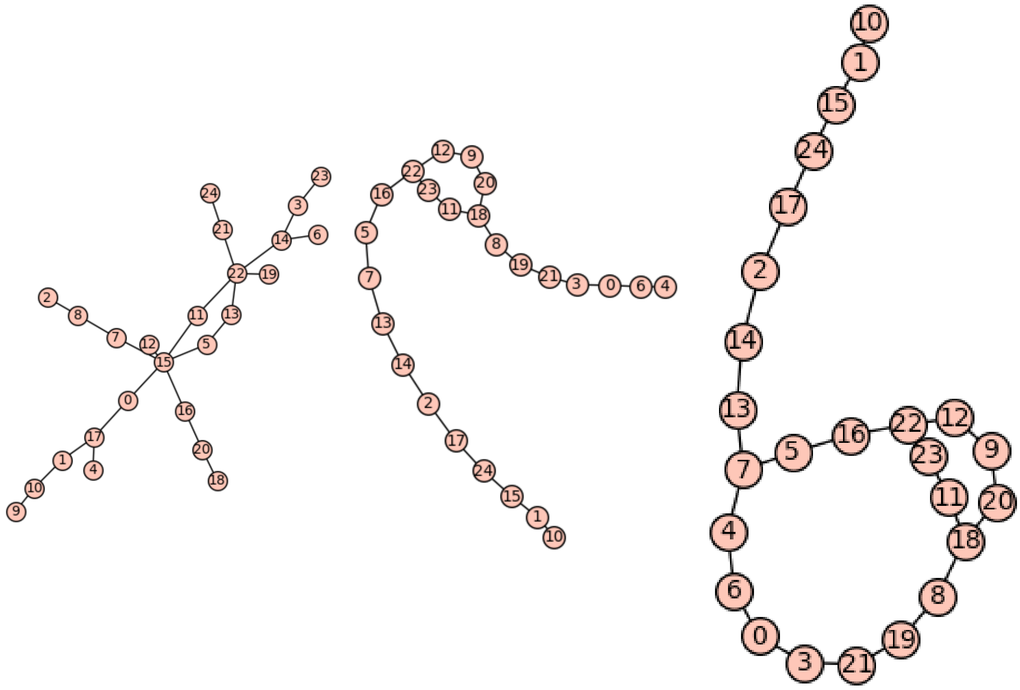
Slika 4: Začetni graf s 100 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 9$



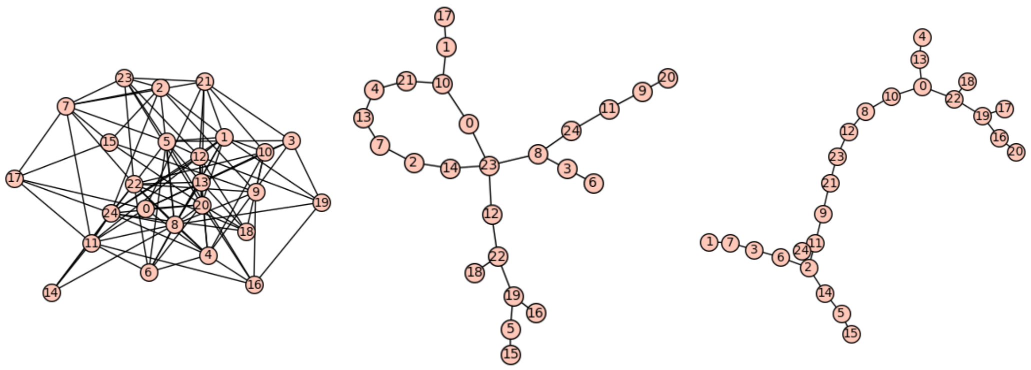
Slika 5: Začetni graf s 200 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 11$

Minimalno razliko za grafe na 25 vozliščih sva testirali za naključne grafe s 25, 50, 100, 200 in 300 povezavami. Najmanjša dobljena razlika $a(G)+1-\gamma_t(G)$ je bila 6.

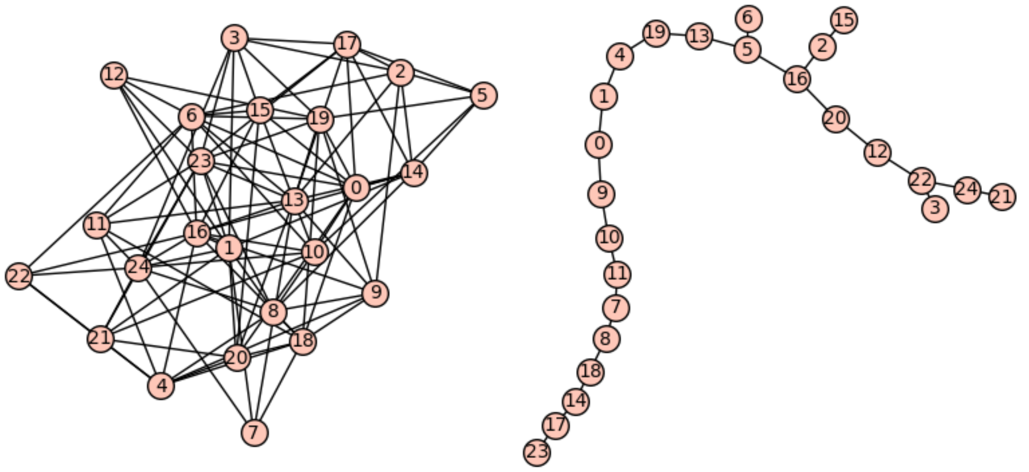
Osnovna funkcija za spreminjanje grafa ima enako verjetnost, da doda ali izbriše povezavo, torej število povezav v novih generiranih grafih ostaja blizu začetnega števila povezav. Z nekaj testiranja sva opazili trend, da se minimum razlike povečuje, ko povečujeva začetno število povezav, zato sva povečali verjetnost, da se povezava izbriše na 0,75 in na 0,9. Na ta način sva tudi pri večjem začetnem številu povezav prišli do enakega minimuma kot pri manjšem, prav tako pa sva celo dobili boljšo minimalno razliko in sicer 5.



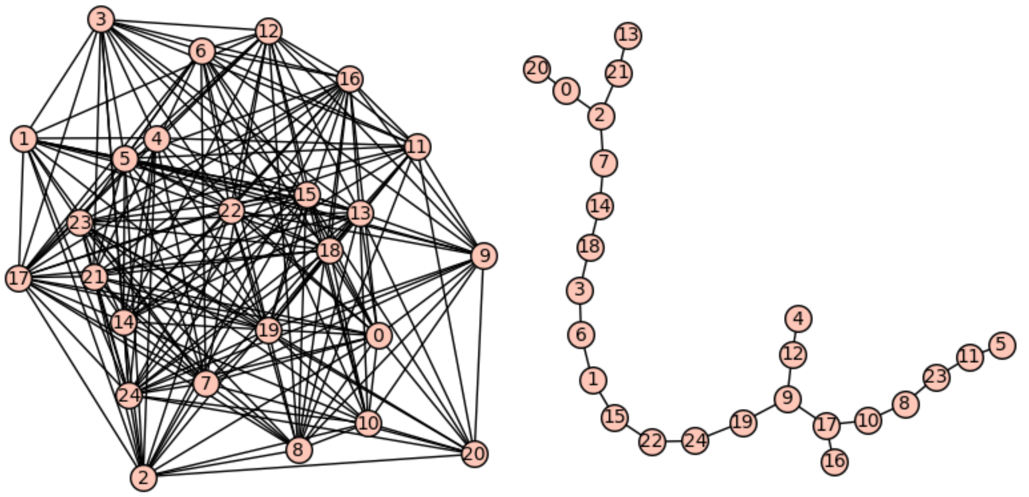
Slika 6: Začetni graf s 25 povezavami in rešitvi, $a(G)+1-\gamma_t(G)=5$, p=0.75



Slika 7: Začetni graf s 100 povezavami in rešitvi, $a(G)+1-\gamma_t(G)=6$, p=0.75

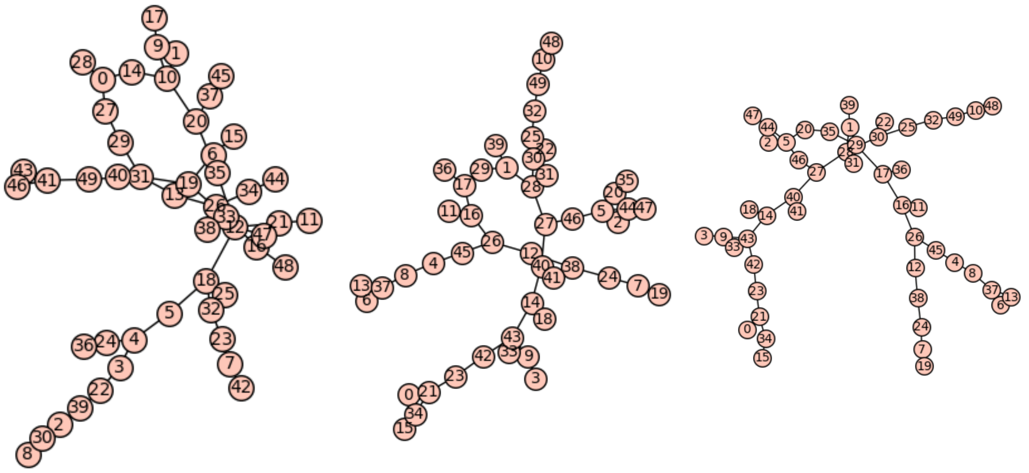


Slika 8: Začetni graf z 100 povezavami in rešitev, $a(G)+1-\gamma_t(G) = 5$, p=0.9

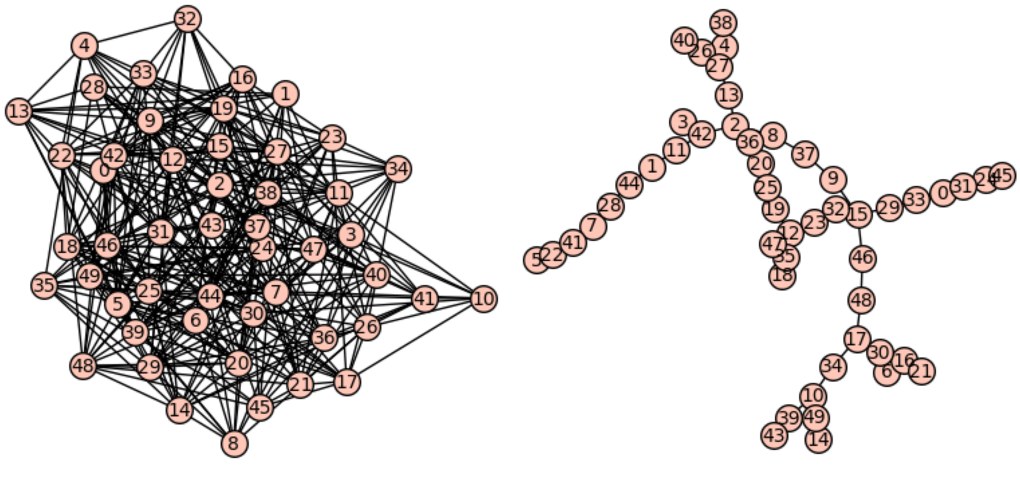


Slika 9: Začetni graf z 200 povezavami in rešitev, $a(G)+1-\gamma_t(G) = 5$, p=0.9

#### 3.2.3 Grafi na 50 vozliščih



Slika 10: Začetni graf s 50 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 12$, p=0.5



Slika 11: Začetni graf s 350 povezavami in rešitev, $a(G) + 1 - \gamma_t(G) = 11$, p=0.75

Minimalno razliko za grafe na 50 vozliščih sva testirala za naključne grafe s 50 in 350 povezavami. Pri verjetnosti 0,75, da funkcija *spremeni_graf* odstrani povezavo, sva izračunali minimalno razliko $a(G) + 1 - \gamma_t(G)$ 11.

# 4 Rezultati testiranja za maksimalno razliko

## 4.1 Testiranje na malih grafih

Za vse grafe s številom vozlišč 3 do 8 sva izračunali maksimalne razlike. Izkaže se, da je maksimalna razlika za grafe na treh vozliščih 2, za grafe na štirih vozliščih 3, na petih 4, na šestih 5, na sedmih 6 in za grafe na osmih vozliščih 7. Iz tega lahko sklepava, da se največja razlika povečuje, ko se povečuje začetni graf na katerem razliko preverjamo.

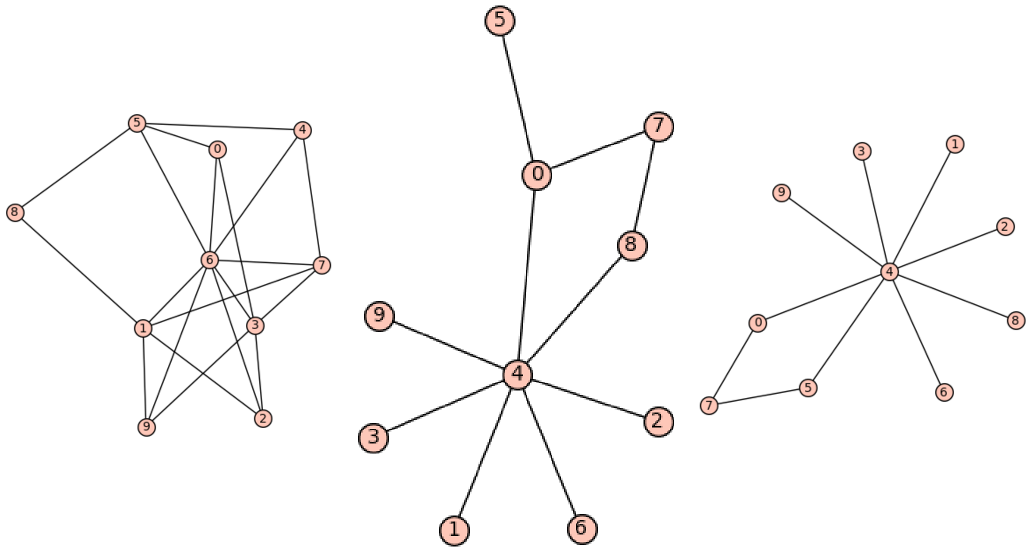
## 4.2 Testiranje na večjih grafih

Za namen računanja maksimuma razlike $a(G) + 1 - \gamma_t(G)$ sva definirali naslednjo funkcijo. Ta za 2000 korakov izračuna razliko in nato spremeni graf s funkcijo *spremeni_graf*. Na koncu vrne maksimalno razliko ter grafe, pri katerih je ta dosežena.

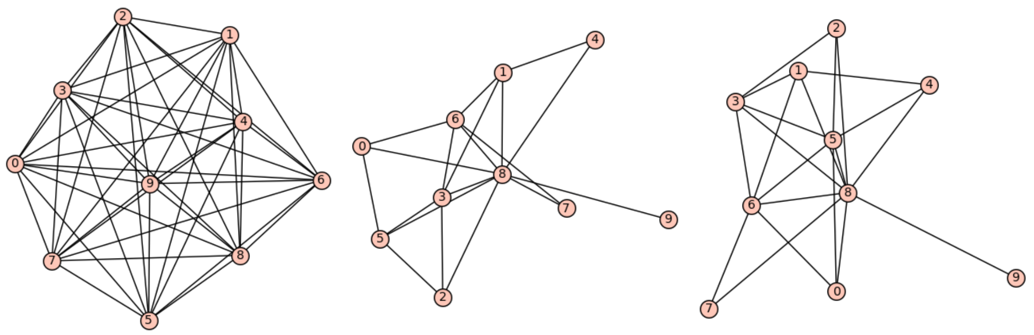
```
def najvecje_razlike(G):
    stevilo_korakov = 2000
    trenutna_razlika = max_razlike = razlika(G)
    seznam_grafov_max = [G]
    for p in range(0, stevilo_korakov):
        T = stevilo_korakov / ( p + 1 )
        nov_graf = spremeni_graf(G)
        nova_razlika = razlika(nov_graf)
        if nova_razlika > max_razlike:
            max_razlike = nova_razlika
            seznam_grafov_max = []
            seznam_grafov_max.append(nov_graf)
        elif nova_razlika == max_razlike:
             if all(not H.is_isomorphic(nov_graf) for H in seznam_grafov_max):
                seznam_grafov_max.append(nov_graf)
        if nova_razlika > trenutna_razlika or exp(-1 * (trenutna_razlika - nova_razlika) / T) >= random():
            G = nov_graf
            trenutna_razlika = nova_razlika
        G = nov_graf

    return (max_razlike, seznam_grafov_max)
```

Najprej sva največjo razliko računali pri enaki verjetnosti, da se povezava doda ali odstrani. Ker pa sva pri iskanju minimuma razlike ugotovili, da se z manjšanjem števila povezav v grafu razlika zmanjšuje, sva kasneje pri iskanju maksimalne razlike testirali tudi kaj se zgodi, če verjetnost, da se povezava odstrani, zmanjšava, in sicer na 0,25. Kljub spremenjeni verjetnosti se rezultati niso izboljšali.

### 4.2.1 Grafi na 10 vozliščih



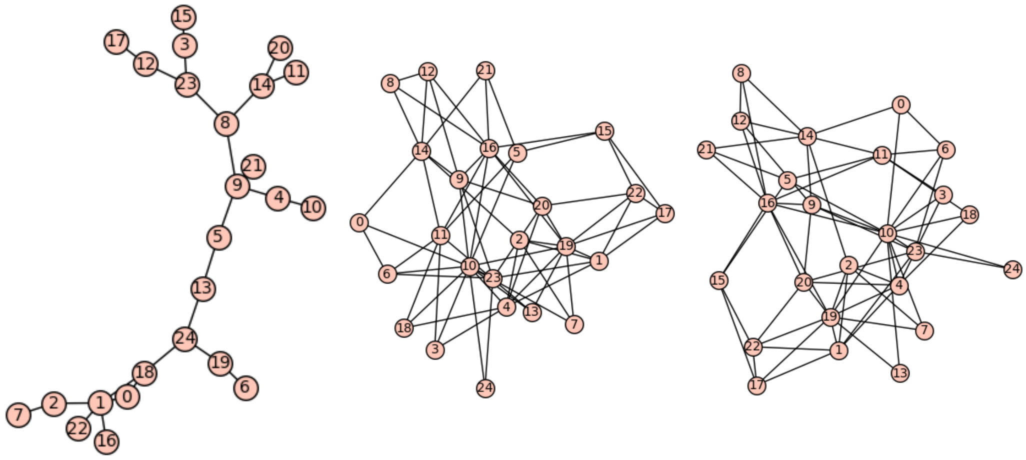
Slika 12: Začetni graf z 20 povezavami in rešitvi, $a(G)+1-\gamma_t(G) = 7$, p=0.5



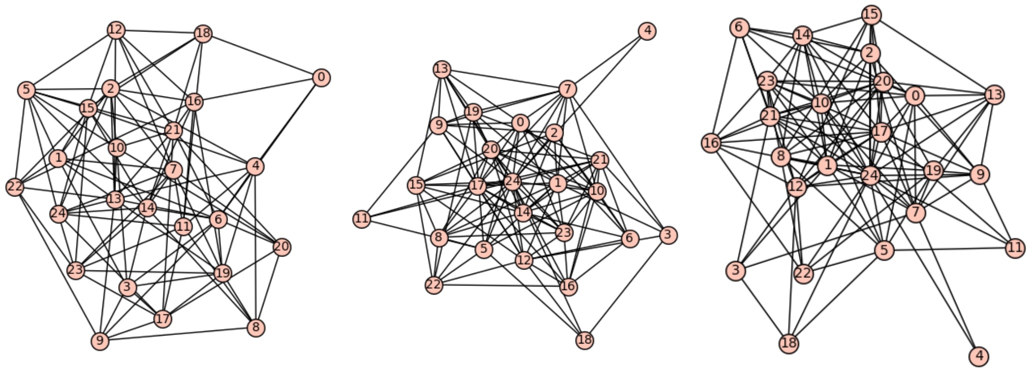
Slika 13: Začetni graf s 45 povezavami in rešitvi, $a(G)+1-\gamma_t(G) = 7$, p=0.5

Za grafe na 10 vozliščih sva maksimalno razliko testirali na grafih z začetnim številom povezav 10, 20, 30, 45. V vseh primerih sva dobili največjo razliko 7.

#### 4.2.2 Grafi na 25 vozliščih



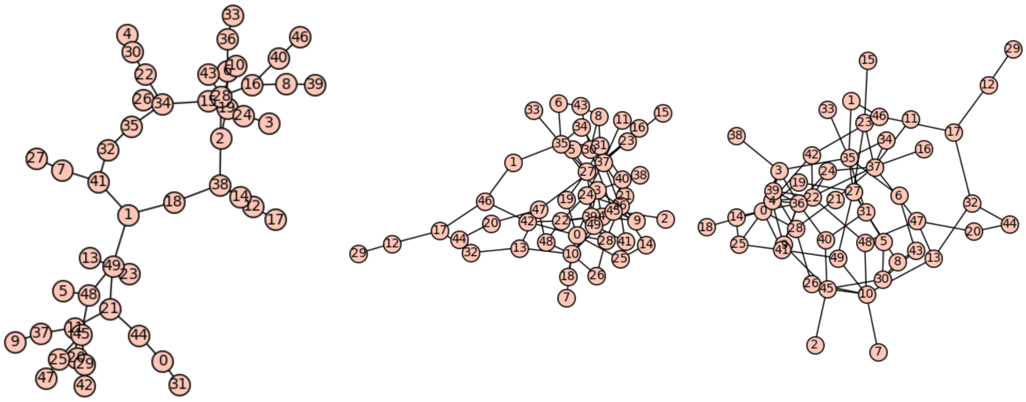
Slika 14: Začetni graf s 25 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 15$, p=0.5



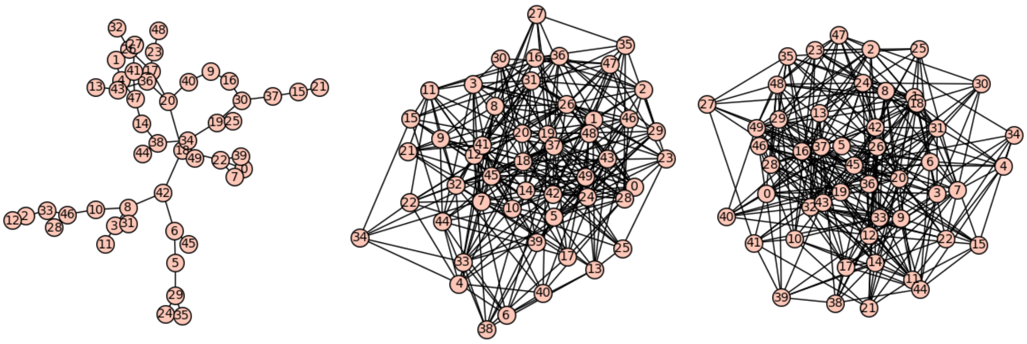
Slika 15: Začetni graf s 100 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 15$, p=0.5

Za grafe z začetnim številom povezav 25 in 100 sva dobili maksimalno razliko 15, število povezav v grafih rešitev pa se je glede na začetni graf povečalo. Za grafe z začetnim številom povezav 200 in 300 pa sva dobili maksimum razlike 14 oz. 13, medtem ko se je povezav v grafih rešitev zmanjšalo v primerjavi z začetnim. Domnevava, da bi tudi za slednje lahko dobili maksimalno razliko 15 ali več, vendar je zaradi večje časovne zahtevnosti problema program pregledal manj grafov.

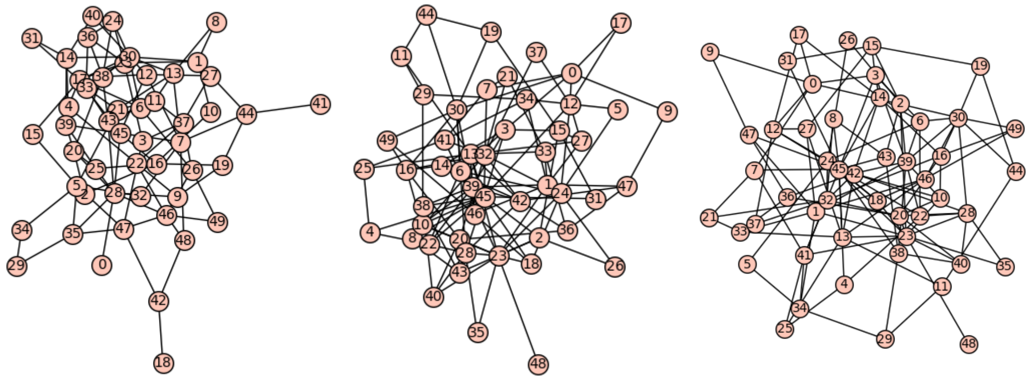
### 4.2.3 Grafi na 50 vozliščih



Slika 16: Začetni graf s 50 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 25$, p=0.5



Slika 17: Začetni graf s 50 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 26$, p=0.25

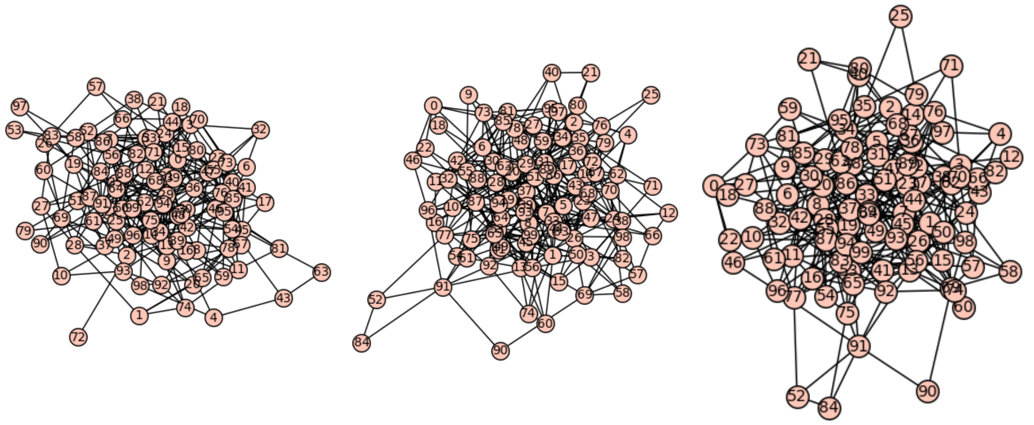


Slika 18: Začetni graf s 100 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 26$, p=0.50

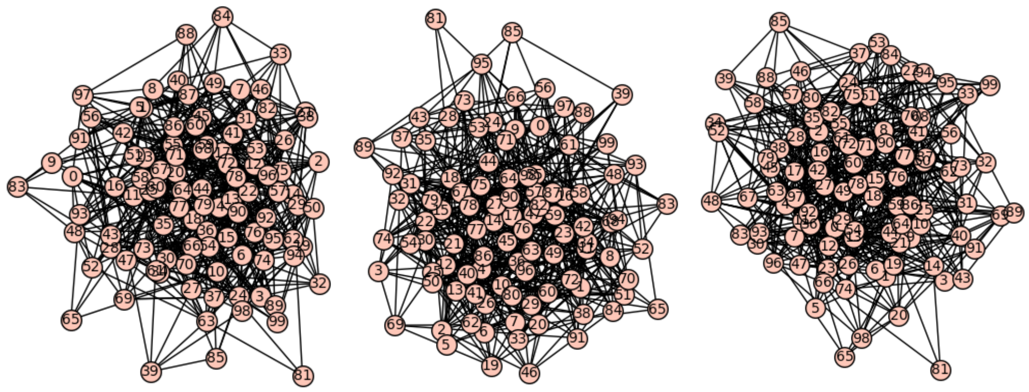
Za grafe z začetnim številom povezav 50 sva dobili maksimalno razliko 25, pri 100 začetnih povezavah pa razliko 26. Pri 50 začetni povezavah sva verjetnost odstranitve povezave zmanjšali na 0,25 in s tem razliko izboljšali na 26. Opazili sva, da se je število povezav v grafih rešitev pri verjetnosti odstranitve povezav 0,5 gibalo med 60 in 80, medtem ko se je število povezav pri verjetnosti odstranitve 0,25 močno povečalo na 180 do 400. Zanimivo je, da se je kljub veliki spremembi v številu povezav v grafih rešitev razlika izboljšala samo za 1.

### 4.2.4 Grafi na 100 vozliščih

Preverili sva še grafe na 100 vozliščih s 300 in 600 povezavami. Maksimalna razlika pri 300 povezavah je bila 51, pri 600 pa 52.



Slika 19: Začetni graf s 300 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 51$, p=0.50



Slika 20: Začetni graf s 600 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 52$, p=0.50

# 5 Zaključek

Najina naloga je bila potrditi ali zavreči domnevo o povezavi med anihilacijskim in dominacijskim številom. Ugotovili sva, da večji kot je bil začetni graf, večja je bila minimalna razlika $a(G) + 1 - \gamma_t(G)$, najmanjša razlika, ki sva jo v testiranjih dobili, pa je bila 1. Glede na najine rezultate, hipoteze ne moreva ovreči.