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 bova 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 bova najprej preverili za netrivialne, povezane grafe z majhnim številom vozlišč. Začeli bova z vsemi grafi s tremi vozlišči, nato pa število vozlišč povečevali, dokler bo neenakost izračunana v relativno kratkem času.

Za grafe z velikim številom vozlišč bova napisali algoritem, ki generira naključni graf z večjim številom vozlišč, 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. Cilj je za različna števila vozlišč poiskati minimum razlike $a(G) + 1 - \gamma_t(G)$ in poiskati skupne karakteristike teh grafov.

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

Kot pomoč pri računanju bova v orodju *CoCalc* definirali še funkciji:

- *dominacijsko število*, ki bo za dani graf izračunala dominacijsko število;
- *anihilacijsko število*, ki bo za dani graf izračunala anihilacijsko število.

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 vozliščev 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.

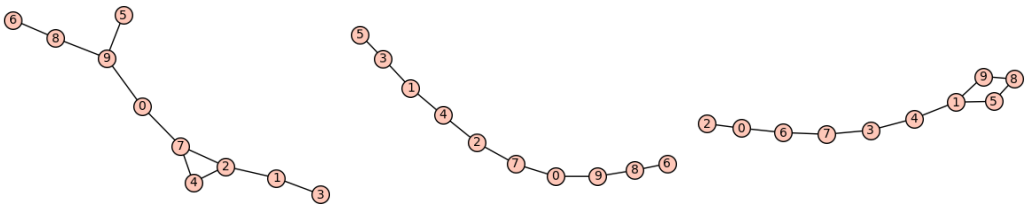
```
from sage.graphs.connectivity import is_connected
def spremeni_graf(G):
    H = Graph(G)
    if random() < 0.75:
        i = 0
        while True:
            H.delete_edge(H.random_edge())
            if is_connected(H):
                H
                break
            else:
                H = 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 za 2000 korakov izračuna razliko in nato spremeni graf z zgornjo funkcijo. Na koncu vrne minimalno razliko ter grafe, pri katerih je ta dosežena.

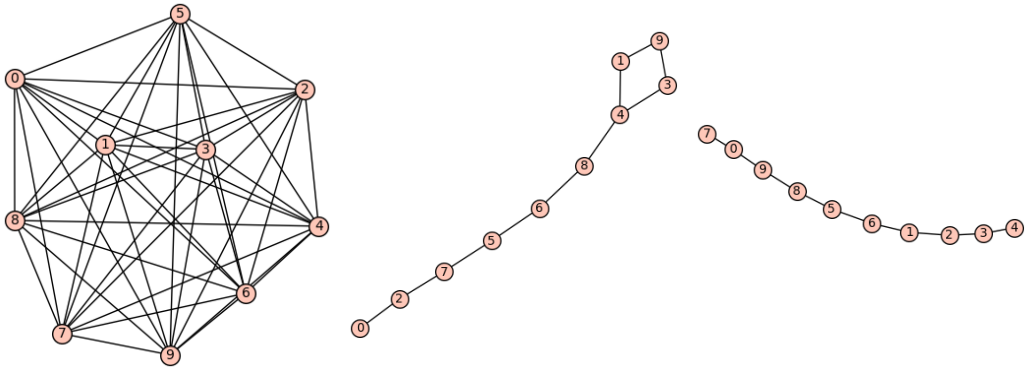
```
def najmanjse_razlike(G):
    stevilo_korakov = 2000
    trenutna_razlika = min_razlike = razlika(G)
    seznam_grafov_min = [G]
    for p in range(0, stevilo_korakov):
        T = stevilo_korakov / (p+1)
        nov_graf = spremeni_graf(G)
        nova_razlika = razlika(nov_graf)
        if nova_razlika < 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



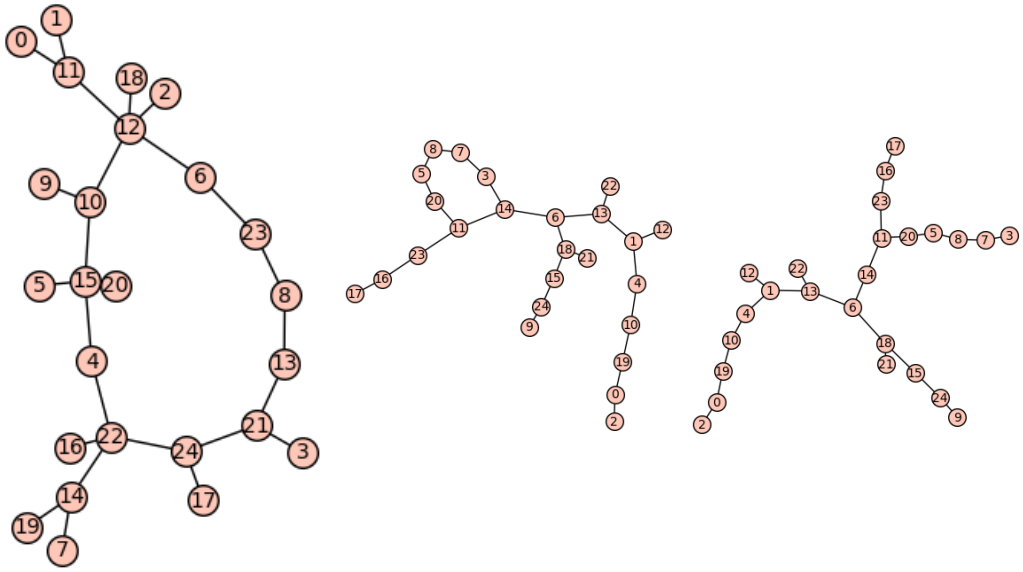
Slika 1: Začetni graf z 10 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 2$



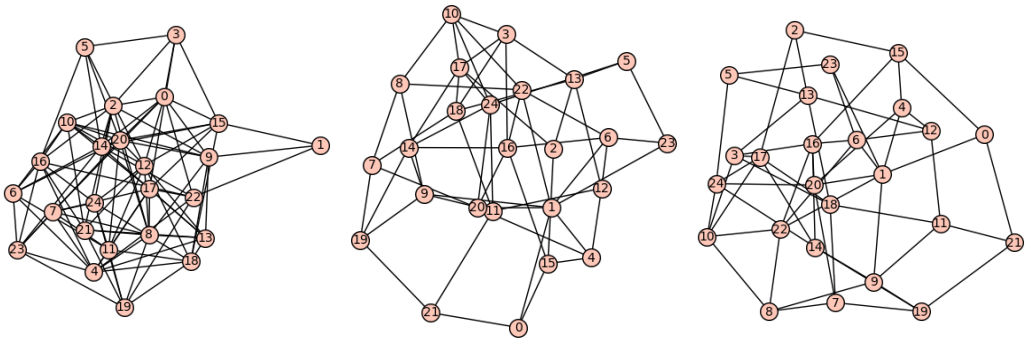
Slika 2: Začetni graf s 45 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 2$

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 $a(G)+1-\gamma_t(G)$ 2.

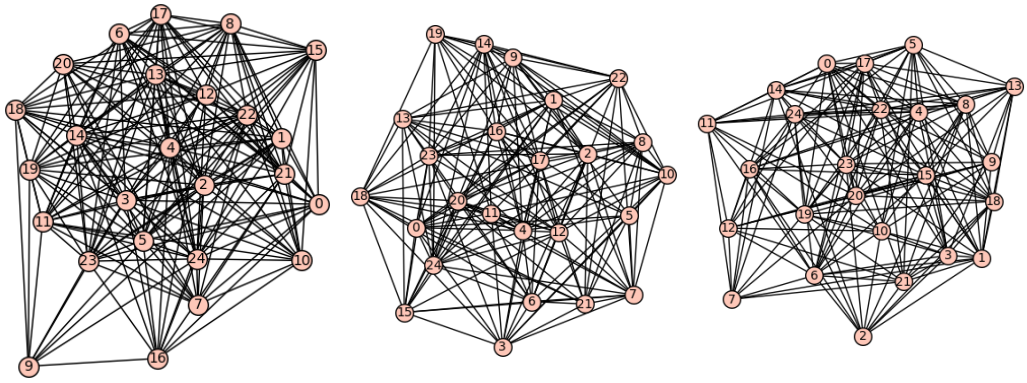
3.2.2 Grafi na 25 vozliščih



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



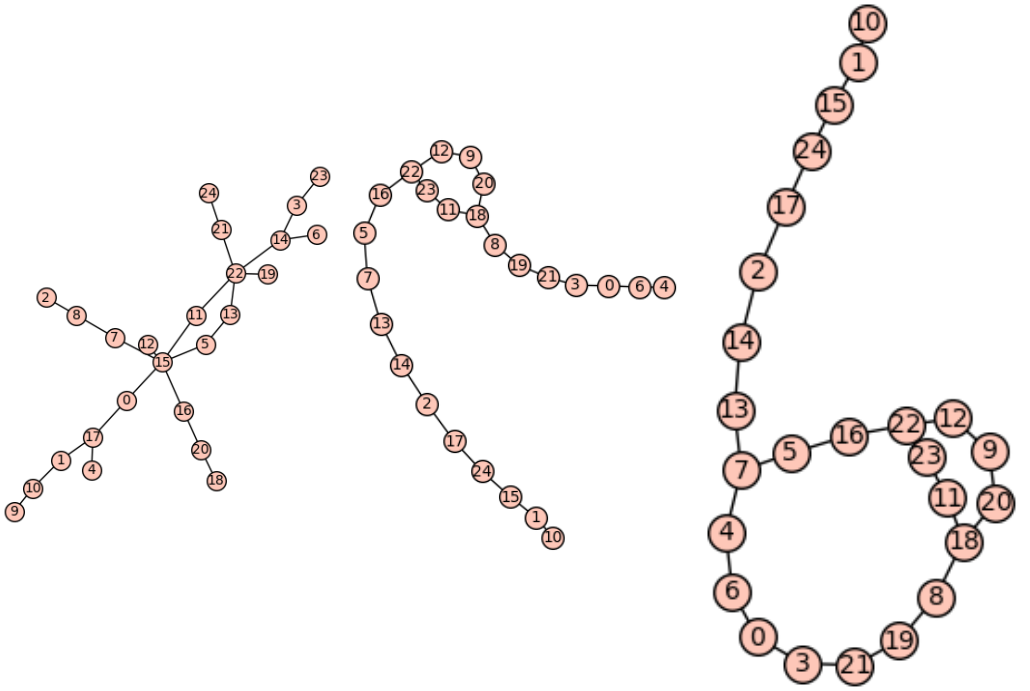
Slika 4: Začetni graf s 100 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 9$



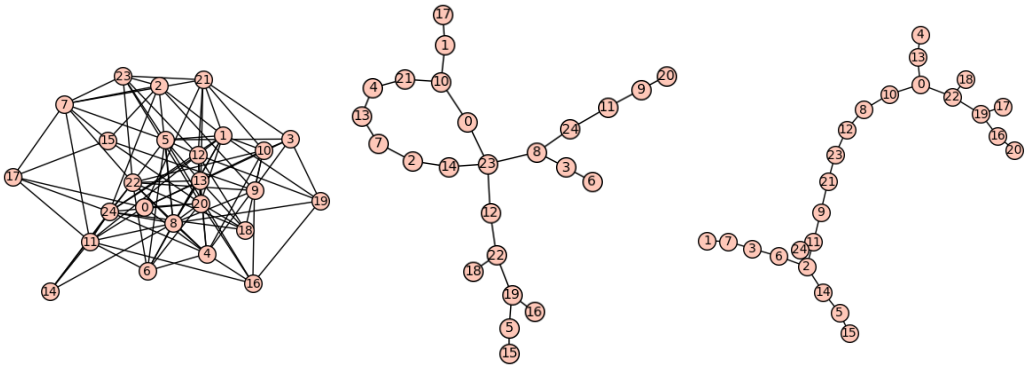
Slika 5: Začetni graf s 200 povezavami in rešitvi, $a(G) + 1 - \gamma_t(G) = 11$

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.

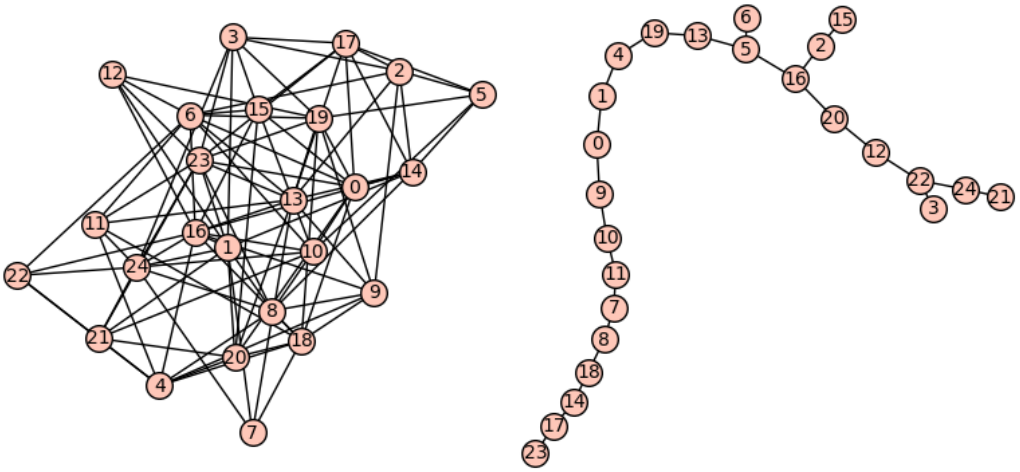
Osnovna funkcija za spreminjanje grafa ima enako verjetnost, da doda ali izbriše povezavo, torej število povezav v novih generiranih grafih ostaja blizu začetnega števila povezav. Z nekaj testiranja sva opazili trend, da se minimum razlike povečuje, ko povečujeva začetno število povezav, zato sva povečali 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 dobili boljšo minimalno razliko in sicer 5.



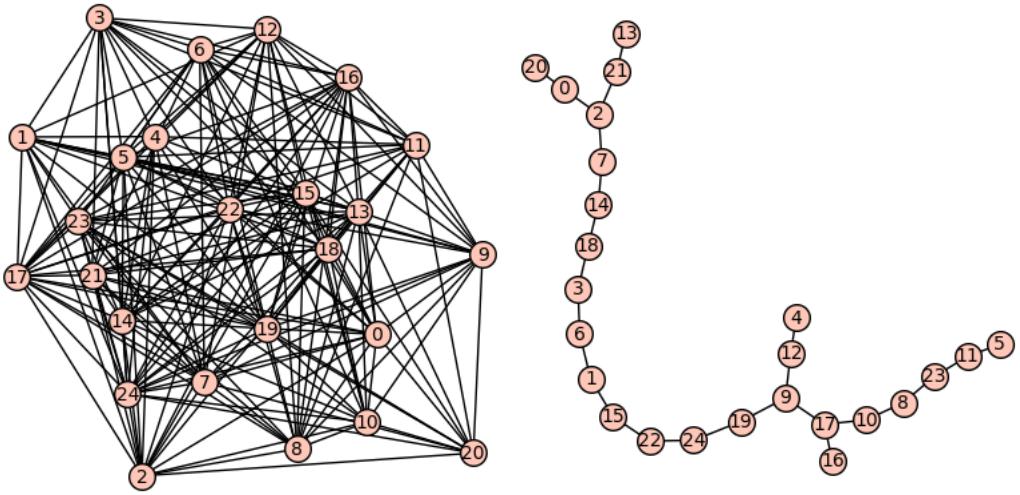
Slika 6: Začetni graf s 25 povezavami in rešitvi, $a(G)+1-\gamma_t(G) = 5$, p=0.75



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

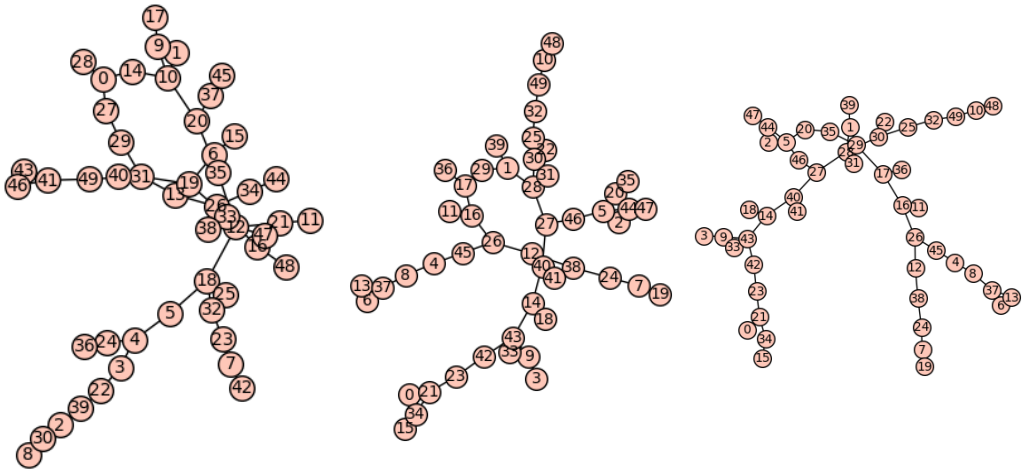


Slika 8: Začetni graf z 100 povezavami in rešitev, $a(G)+1-\gamma_t(G) = 5$, p=0.9

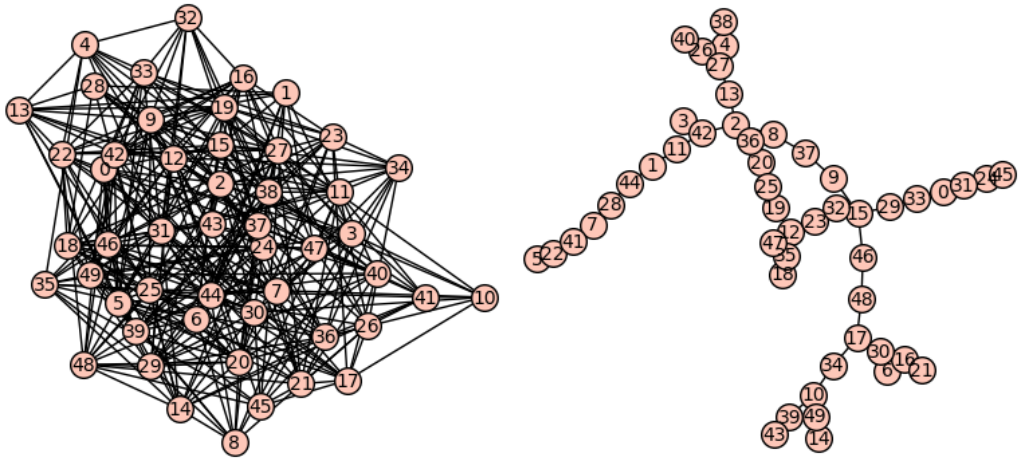


Slika 9: Začetni graf z 200 povezavami in rešitev, $a(G)+1-\gamma_t(G) = 5$, p=0.9

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