Domination on Modular Product Graphs

Maks Rozman, Nea Lovrinčić, Rene Turk

31. januar 2025

1 Uvod

V projektni nalogi smo preučevali dominacijo (dominacijsko število) na modularnem produktu grafov. Modularni produkt smo označevali z znakom $\diamond$, dominacijsko število pa z γ. Kot je predvideno v navodilih, smo si delo razdelili na dva dela. Zaradi težav pri uporabi Sage-a na naših napravah (ter omejitev plačljive spletne storitve CoCalc), smo kodo pisali v Pythonu s pomočjo knjižnice `NetworkX`.

2 Definicije

Delali smo z neusmerjenimi grafi $G = (V, E)$, kjer je V množica vozlišč, E pa množica povezav grafa.

2.1 Dominacijska množica in dominacijsko število

Definicija 2.1. Dominacijska množica grafa $G = (V, E)$ je podmnožica $D \subseteq V$, za katero velja, da je vsako vozlišče, ki ni v D, sosed vsaj enemu vozlišču iz D. Dominacijsko število $\gamma(G)$ je moč najmanjše dominacijske množice grafa G.

2.2 Modularni produkt

Definicija 2.2. Modularni produkt $G \diamond H$ grafov G in H je graf z množico vozlišč $V(G) \times V(H)$, ki je unija kartezičnega produkta, direktnega produkta in direktnega produkta komplementov G in H:

$$G \diamond H = G \square H \cup G \times H \cup \overline{G} \times \overline{H}.$$

Natančneje, vozlišči (g, h) in (g', h') grafa $G \diamond H$ sta sosednji, če:

1. $g = g'$ in $hh' \in E(H)$ ali $gg' \in E(G)$ in $h = h'$; **ali**
2. $gg' \in E(G)$ in $hh' \in E(H)$; **ali**
3. (ko $g \neq g'$ in $h \neq h'$) $gg' \notin E(G)$ in $hh' \notin E(H)$.

3 Prvi del projektne naloge

V prvem delu smo iskali grafe G, za katere velja $\gamma(G \diamond G) \geq \gamma(G) + 2$.

3.1 Stohastično iskanje ustreznih grafov

Najprej smo v datoteko `funkcije.py` zapisali štiri funkcije, ki smo jih potem uporabljali za iskanje ustreznih grafov. in sicer:

- `modularni_produkt` za konstrukcijo grafa modularnega produkta dveh grafov,
- `dominacijsko_stevilo`, da smo dobili dominacijsko število grafa,
- `zmanjsaj_premer_na_2`, ki je danemu grafu dodajala povezave, dokler ni bil premer grafa enak 2,
- `stohasticno_iskanje_grafa`, ki je stohastično generirala grafe (dejansko smo na koncu imeli tri, ker smo poskusili tri nekoliko različne načine konstrukcije).

3.1.1 Modularni produkt grafov

Psevdokoda za konstrukcijo grafa modularnega produkta dveh grafov:

Algorithm 1 Modularni produkt grafov

function ModularniProdukt(G, H)
  GH $\leftarrow$ nov prazen graf
  GH_vozlišča $\leftarrow \{(g,h) \mid g \in V(G), h \in V(H)\}$ ▷ Vozlišča GH so kartezični produkt vozlišč G in H.
  Dodaj GH_vozlišča v GH
  for all $(g_1,h_1) \in$ GH_vozlišča **do** ▷ Inicializiramo dve for zanki za dodajanje povezav po treh pravilih modularnega produkta (glej definicijo 2.2).
    for all $(g_2,h_2) \in$ GH_vozlišča **do**
      if $g_1 = g_2$ in $(h_1,h_2) \in E(H)$ **then**
        Dodaj povezavo $((g_1,h_1),(g_2,h_2))$ v GH
      else if $(g_1,g_2) \in E(G)$ in $h_1 = h_2$ **then**
        Dodaj povezavo $((g_1,h_1),(g_2,h_2))$ v GH
      else if $(g_1,g_2) \in E(G)$ in $(h_1,h_2) \in E(H)$ **then**
        Dodaj povezavo $((g_1,h_1),(g_2,h_2))$ v GH
      else if $g_1 \neq g_2$ in $h_1 \neq h_2$ in $(g_1,g_2) \notin E(G)$ in $(h_1,h_2) \notin E(H)$ **then**
        Dodaj povezavo $((g_1,h_1),(g_2,h_2))$ v GH
      end if
    end for
  end for
  return GH ▷ Vrnemo graf GH modularnega produkta grafov G in H.
end function

3.1.2 Dominacijsko število

Sledi zapis linearnega programa za izračun dominacijskega števila grafa. Prva oblika te funkcije je bila "brute-force". Za vsako število i od 1 do vključno n smo generirali vse možne podmnožice vozlišč moči i. Ko je funkcija za dano podmnožico vozlišč ugotovila, da je dominacijska, je vrnila število i (ki je avtomatsko dominacijsko število, saj je i moč najmanjše dominacijske množice grafa). Funkcija je pravilno delovala, vendar je bila časovno zelo potratna ($O(2^n)$). Po posvetovanju s prof. Škrekovskim smo za računanje dominacijskega števila vpeljali časovno veliko učinkovitejši linearni program (v Python funkcijo smo ga implementirali z orodjem PuLP):

$$x_v = \begin{cases} 1; & v \in V(G) \text{ je v dominacijski množici} \\ 0; & v \in V(G) \text{ ni v dominacijski množici} \end{cases}$$

Hočemo

$$\min \sum_{v \in V(G)} x_v,$$

pri pogoju

$$x_v + \sum_{u \in N(v)} x_u \geq 1, \quad \forall v \in V(G),$$

kjer je $N(v)$ množica vseh sosedov vozlišča v v grafu G:

$$N(v) = \{u \in V(G) \mid (u, v) \in E(G)\}.$$

Končna vrednost dominacijskega števila je

$$\sum_{v \in V(G)} x_v.$$

3.1.3 Zmanjšaj premer na 2

Znano je, da če obstajajo grafi, ki zadovoljijo ta pogoj, mora biti njihov premer enak 2. Zato smo grafe, ki smo jih stohastično konstruirali, s pomočjo naslednje metode preuredili, da so imeli premer enak 2:

1. Če vhodni graf ni bil povezan, je funkcija vrnila napako. Zato smo najprej naključno dodajali povezave, dokler graf ni bil povezan.

2. Če je bil premer grafa večji od 2, je metoda najprej poiskala dve vozlišči na grafu, ki sta bili na največji razdalji. Med njima je nato naključno dodala povezavo.

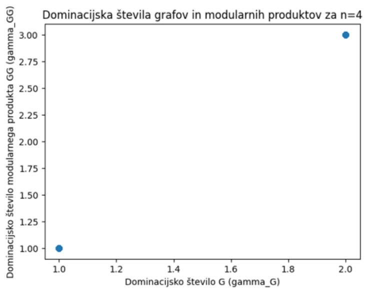
3. Nato je izbrala drugi dve vozlišči, ki sta bili na maksimalni povezavi, in tako naprej, dokler ni bil premer grafa enak 2.

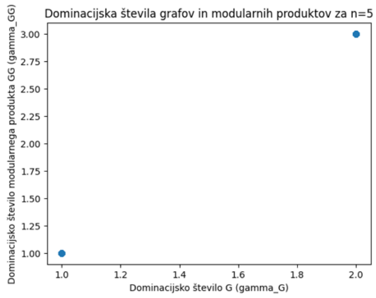
3.1.4 Stohastična konstrukcija na tri načine

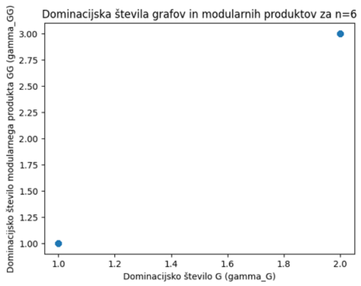
Grafe, ki ustrezajo pogoju, smo iskali stohastično na tri podobne načine. Za vsako število vozlišč n smo konstruirali 1000 grafov. Načini so se razlikovali v tem, da smo pri prvem načinu vsak graf na novo konstruirali, kjer je bilo n število vozlišč, število povezav pa je bilo naključno. Če graf ni bil povezan, smo iteracijo zanke preskočili. Pri drugem načinu smo konstruirali graf na n vozliščih brez povezav, in nato naključno dodajali povezave, dokler graf ni bil povezan. Hoteli smo poskusiti, če na ta način morda dobimo drugačne grafe kot v prvem primeru. Pri obeh načinih smo potem enako uporabili funkcijo za zmanjšanje premera grafa na 2, in potem testirali, če je graf ustrezen. Pri tretjem načinu pa smo graf kot pri drugem načinu konstruirali in ga testirali, nismo pa v naslednji iteraciji naredili novega grafa, temveč smo staremu odstranili nekaj povezav, spet uporabili funkcijo za zmanjšanje premera na 2 in tako naprej.

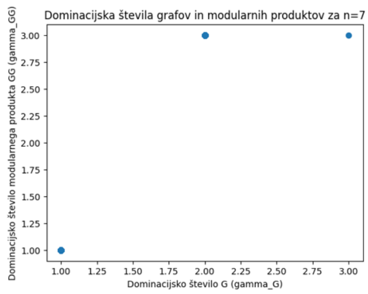
3.2 Ugotovitve

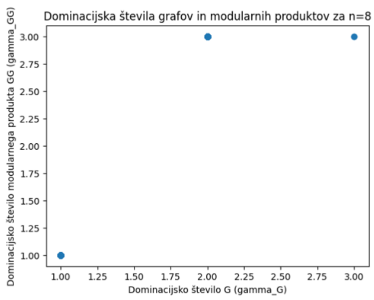
Pri iskanju grafov žal nismo bili uspešni, saj kljub testiranju na velikem številu grafov (od 2 do vključno 22 vozlišč) nismo našli nobenega, ki bi ustrezal pogoju. S funkcijami, ki so delovale na isti način, kot funkcije za stohastično generiranje grafov, smo konstruirali 1000 grafov, za vsako število vozlišč od 3 do 10. Potem smo naredili prikaz, kjer smo za vsak graf G izračunali njegovo dominacijsko število (x-os) ter dominacijsko število modularnega produkta $(G \diamond G)$ (y-os):

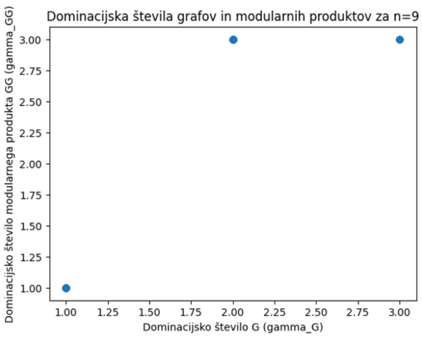


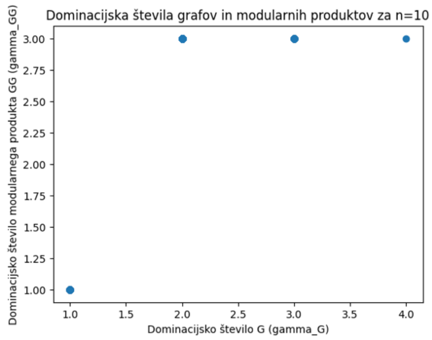












Vidimo lahko, da smo zajeli maksimalno možno dominacijsko število povezanega grafa za vsako število vozlišč (kar nam pove, da smo uspeli generirati med sabo dovolj različne grafe). Vidimo lahko tudi da je dominacijsko število modularnega produkta $(G \diamond G)$ kvečjemu za ena večje od dominacijskega števila G, za grafe, ki smo jih uspeli generirati.

4 Drugi del projektne naloge

4.1 Uvod

V tej nalogi smo preverjali veljavnost neenakosti $\gamma(G \diamond H) \leq 4$ za modularni produkt grafa G s premerom vsaj 3 in grafa H tipa (SB). Pri tem smo uporabili podatke in funkcije, ki so jih razvile skupine 20 in 21. Cilj je bil najti pare grafov G in H, pri katerih se doseže zgornja meja $\gamma(G \diamond H) = 4$, ter preveriti, ali obstajajo pari, kjer neenakost ne drži.

4.2 Metodologija

Za generiranje grafov smo uporabili naslednje pristope:

- Graf G je bil naključno generiran z uporabo modela Erdős–Rényi, pri čemer smo preverili, da je povezan in da ima premer vsaj 3.
- Graf H je bil generiran kot graf tipa (SB) s pomočjo funkcije `generate_random_sb()`.
- Izračunan je bil modularni produkt $G \diamond H$.
- Dominacijsko število $\gamma(G \diamond H)$ je bilo izračunano s funkcijo `dominacijsko_stevilo()`.
- Za vsako kombinacijo velikosti grafov (do vključno 20 vozlišč) je bilo izvedenih 1000 iteracij.

4.3 Rezultati

Po izvedenih simulacijah so bili dobljeni naslednji rezultati:

- Najden ni bil noben par grafov G in H, za katerega bi neenakost $\gamma(G \diamond H) \leq 4$ ne držala.
- Prav tako ni bil najden noben par, za katerega bi držala enakost $\gamma(G \diamond H) = 4$.
- Vsi generirani primeri so potrjevali domnevo, da $\gamma(G \diamond H) \leq 4$.

4.4 Zaključek

S pomočjo funkcij drugih skupin smo uspešno preverili neenakost $\gamma(G \diamond H) \leq 4$ za različne pare grafov. Rezultati kažejo, da v nobenem primeru dominacijsko število modularnega produkta ni preseglo vrednosti 4, kar potrjuje domnevo. Prav tako nismo našli primera, kjer bi bila dosežena zgornja meja, kar nakazuje možnost, da bi se dalo neenakost dodatno zožiti. Nadaljnje delo bi lahko vključevalo povečanje števila iteracij ali analizo posebnih razredov grafov, pri katerih bi morda lahko dosegli $\gamma(G \diamond H) = 4$.