

Univerza v Ljubljani  
Fakulteta za matematiko in fiziko

# POROČILO

**Group 13: Subpath number – minimal cubic graphs**

Avtorja:

Lovro Levačič  
Žiga Obradović

Mentor: prof. dr. Riste Škrekovski

December 2025

# Kazalo

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Testiranje domneve 1. za majhne <math>n</math></b>	<b>2</b>
<b>3</b>	<b>Simulated annealing</b>	<b>3</b>
3.1	Delovanje algoritma . . . . .	3
3.2	Rezultati . . . . .	4
<b>4</b>	<b>Konstrukcije novih grafov in domneva</b>	<b>5</b>
4.1	Goseničasti konstrukciji . . . . .	6
4.2	Drevesna konstrukcija . . . . .	6
4.3	Domneva . . . . .	6
<b>5</b>	<b>Testiranje domneve 2.</b>	<b>7</b>
5.1	Simulated annealing . . . . .	9
<b>6</b>	<b>Zaključek</b>	<b>10</b>

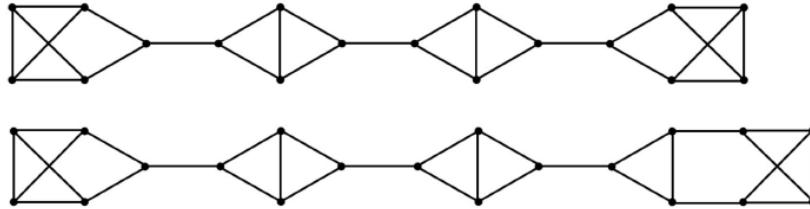
# 1 Uvod

V projektu obravnavamo domnevo, ki govori o minimizaciji števila poti v razredu kubičnih grafov, torej grafov, kjer ima vsako vozlišče stopnjo 3. Za dani povezani graf  $G$  definiramo *subpath number*, označen s  $pn(G)$ , kot število vseh preprostih poti v grafu, vključno s trivialnimi potmi dolžine 0. Preprosta pot je zaporedje vozlišč  $(v_0, v_1, \dots, v_\ell)$  brez ponovitev, kjer je vsak par zaporednih vozlišč povezan z robom.

V literaturi je bila postavljena naslednja domneva.

**Domneva 1.** Za vsako sodo število vozlišč  $n \geq 10$  obstaja natanko en kubični graf  $L_n$ , ki ima najmanjši subpath number med vsemi kubičnimi grafi na  $n$  vozliščih.

Grafi  $L_n$  so sestavljeni iz več kopij grafa  $K_4 - e$  (popolnega grafa na štirih vozliščih, iz katerega odstranimo en rob), ki jih povežemo v verižni strukturi. Na obeh koncih se tej verigi dodata še posebna *pendant bloka* - eden na 5, drugi pa na 7 vozlišč.



Slika 1:  $L_n$  grafa za  $n = 18$  in  $20$

Kasneje se je pokazalo, da domneva za dovolj velika  $n$  ne drži, vendar še ni znano, pri katerem najmanjšem  $n$  se pojavijo prvi protimeri. Namen našega projekta je torej:

- preveriti domnevo za manjša, obvladljiva števila vozlišč  $n$ ,
- in poskusiti konstruirati kubične grafe z manjšim *subpath number* kot  $L_n$ .

## 2 Testiranje domneve 1. za majhne $n$

Če smo domnevo želeli preveriti na vseh kubičnih grafi na  $n$  vozliščih, smo morali najprej te grafe generirati. Izkazalo se je, da številov kubičnih grafov na  $n$  vozliščih eksponentno raste s številom vozlišč, zato smo domnevo preverili na vseh kubičnih grafi le za  $n \leq 18$ .

$n$	# kubičnih grafov
10	19
12	85
14	509
16	4.060
18	41.301
20	510.489
22	7.319.447

Tabela 1: Število neizomorfnih povezanih kubičnih grafov na  $n$  vozliščih.

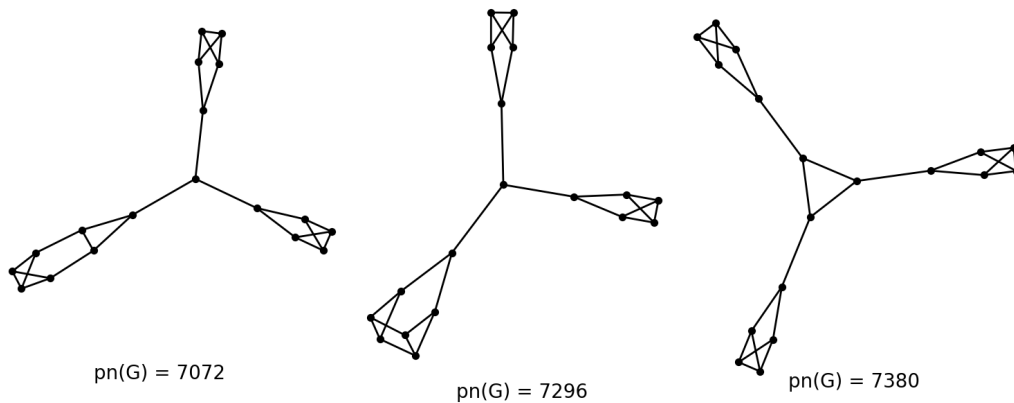
Za  $n \in \{10, 12, 14, 16, 18\}$  smo pridobili vse neizomorfne povezane kubične grafe na  $n$  vozliščih in za vsak graf  $G$  izračunali  $pn(G)$ . Za vsak  $n$  smo posebej primerjali število poti

generiranega grafa  $pn(G)$  s številom poti  $pn(L_n)$  grafa  $L_n$ . Rezultati so prikazani v spodnji tabeli.

$n$	število boljših grafov	minimalno število poti	$pn(L_n)$
10	0	1.276	1.276
12	0	3.076	3.076
14	0	5.504	5.504
16	6	3.640	12.744
18	23	7.072	22.532

Tabela 2: Število vseh grafov na  $n$  vozliščih z manjšim številom poti od grafa  $L_n$ .

Na podlagi rezultatov vidimo, da domneva velja za  $n \in \{10, 12, 14\}$ , medtem ko lahko domnevo za  $n \in \{16, 18\}$  zavrtnemo. Namreč v primeru ko je  $n = 16$  smo našli 6 grafov z manjšim številom poti od grafa  $L_{16}$ , v primeru  $n = 18$  pa že kar 23 grafov z manjšim številom poti od grafa  $L_{18}$ . Izrišimo 3 grafe na 18 vozliščih z najmanjšim številom poti.



Slika 2: Grafi z najmanjšim številom poti za  $n = 18$ .

Ker je za  $n \geq 20$  izračun *subpath number* vedno zahtevnejši, grafov pa vedno več, bomo za iskanje protiprimerov uporabili drugačno metodo. Ta metoda se mimenuje *simulated annealing*.

### 3 Simulated annealing

Za iskanje kubičnih grafov z manjšim številom poti od  $L_n$  grafa smo za  $n \geq 20$  uporabili metahevrstični algoritem *simulated annealing* (SA). Gre za probabilistično metodo globalne optimizacije, ki posnema proces fizikalnega ohlajanja kovin: sistem se sprva nahaja pri visoki temperaturi in lahko sprejema tudi poslabšanja, postopoma pa se temperatura znižuje, kar zmanjšuje verjetnost sprejemanja slabših rešitev. Pri dovolj počasnem ohlajanju se algoritem z visoko verjetnostjo približa globalnemu minimumu energetske funkcije, katera je v našem primeru *subpath number*.

#### 3.1 Delovanje algoritma

V našem primeru je prostor iskanja sestavljen iz vseh *povezanih kubičnih grafov* na  $n$  vozliščih. Prehod med grafi definiramo s t. i. *dvojnim prevezovanjem robov* (ang. *double-edge swap*). Naj bo  $G = (V, E)$  kubičen graf in naj bosta izbrani dve disjunktni povezavi  $\{u, v\}, \{x, y\} \in E$ . Zamenjava poteka tako, da se povezavi odstranita in nadomestita z novima paroma  $\{u, x\}$  in  $\{v, y\}$ .

ali s paroma  $\{u, y\}$  in  $\{v, x\}$ . S takimi menjavami povezav ohranimo 3-regularnost grafa. Energij-ska funkcija, ki jo minimiziramo, pa je v našem primeru podana z  $E(G) = \text{subpath\_number}(G)$ , kjer funkcija  $\text{subpath\_number}(\cdot)$  prešteje vse različne poti v grafu.

Proces začnemo z grafom, ki ga želimo izboljšati. V našem primeru je to graf  $L_n =: G$ . Pri vsakem koraku z double edge swap generiramo novega soseda  $G'$  in izračunamo  $\Delta E = E(G') - E(G)$ . Če je  $\Delta E \leq 0$ , rešitev sprejmemo. V nasprotnem primeru jo sprejmemo z verjetnostjo

$$p = \exp\left(-\frac{\Delta E}{T}\right),$$

kjer  $T > 0$  predstavlja trenutni temperaturni parameter. S tem omogočimo kontrolirano sprejemanje slabših rešitev zgodaj v postopku, kar preprečuje prezgodnjo ujetost v lokalne minimume. Če rešitev sprejmemo nastavimo  $G := G'$  in postopek ponavljamo.

V našem primeru smo naredili 20.000 ponovitev postopka, začetni parameter  $T_0$  pa je bil nastavljen tako, da se je v zgodnjih ponovitvah slabša rešitev sprejela z verjetnostjo približno 40 %, proti koncu postopka pa skoraj nikoli.

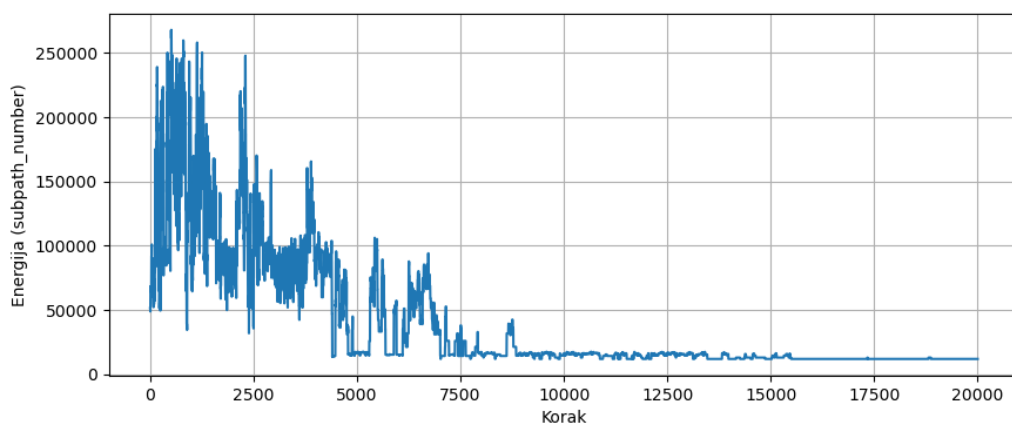
### 3.2 Rezultati

Z metodo *simulated annealing* smo za  $n \in \{20, 22, \dots, 30\}$  iskali grafe z manjšim številom poti od grafa  $L_n$ . Rezultati so v spodnji tabeli.

n	$\text{subpath\_number}(L_n)$	boljši $\text{subpath\_number}$	razlika
20	51.532	11.708	38.716
22	90.760	7.156	83.604
24	206.800	12.220	194.580
26	363.788	21.760	342.028
28	827.988	29.568	798.420
30	1.456.016	18.520	1.437.496

Tabela 3: Rezultati SA.

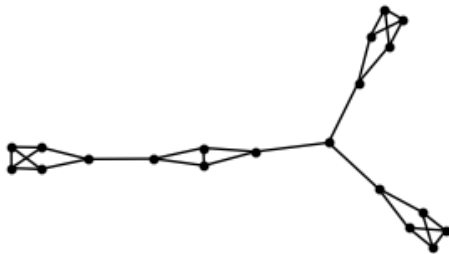
Ker smo našli grafe z manjšim številom poti, smo torej domnevo, da imajo grafi  $L_n$  najmanjše število poti, zavrnilo tudi za  $n \in \{20, 22, \dots, 30\}$ .



Slika 3: Potek energije med *simulated annealing* za  $n = 20$ .

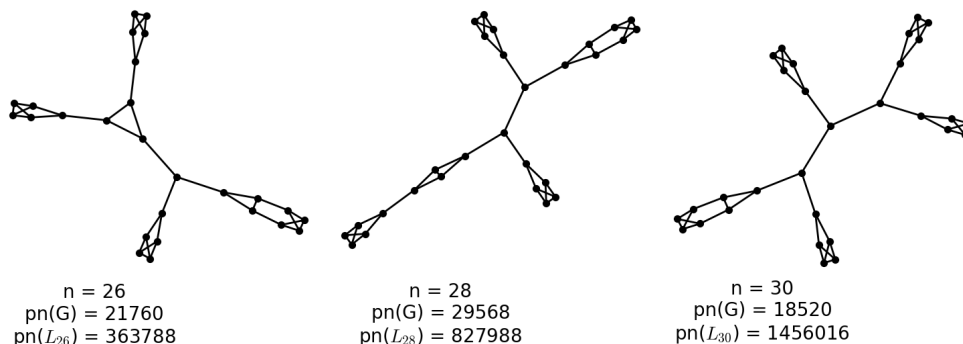
Na sliki lahko vidimo potek energije (števila poti) med procesom SA za  $n = 20$ . Kar lahko razberemo je, da so se med postopkom sprejele tudi slabše rešitve, saj energija ni strogo

padajoča. Program je uspešno sprejemal tudi slabše rešitve in se s tem izognil morebitnim lokalnim minimumom. Vselej pa smo na koncu našli graf s kar 5-krat manjšim številom poti ob grafa  $L_{20}$ . Čeprav samo na podlagi rezultatov SA še ne moremo trditi, da je 11.706 najmanjše število poti za  $n = 20$ , bomo kasneje dokazali, da je to pravzaprav res in da smo s postopkom SA našli graf, ki minimizira število poti za  $n = 20$ .



Slika 4: Graf dobljen s SA za  $n = 20$ .

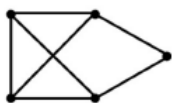
Vredno je poudariti, da v splošnem z najdenimi grafi še zdaleč ne minimiziramo števila poti za izbrani  $n$ , čeprav so za nekatere  $n$  najdeni grafi že precej dobri. Našli smo le primere grafov z manjšim številom poti od opazovanega  $L_n$  grafa. To se dobro vidi, ko primerjamo dobljeno minimalno število poti za  $n = 28$  in  $n = 30$ . Dobljeni graf za  $n = 30$  ima namreč kar za tretjino manj poti od dobljenega grafa za  $n = 28$ . Ta podatek nam za graf z  $n = 30$  sicer ne pove veliko, medtem ko smo lahko skoraj prepričani, da za  $n = 28$  obstaja graf z manjšim številom poti.



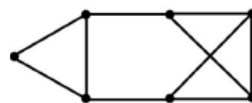
Slika 5: Grafi pridobljeni s SA za  $n = 26$ ,  $28$  in  $30$ .

## 4 Konstrukcije novih grafov in domneva

Ugotovili smo, da grafi  $L_n$  minimizirajo število poti v kubičnih grafih le za  $n \in \{10, 12, 14\}$ . Za sode  $n \geq 16$  pa smo našli po vsaj en protiprimer začetni domnevi. Ob testiranju smo opazili, da je večina grafov z manjšim številom poti od grafov  $L_n$  zgrajenih na podoben način. Grafi, ki smo jih našli, so skoraj vsi sestavljeni iz *pendant blokov* na 5 oziroma 7 vozliščih. Ker želimo konstruirati grafe z najmanjšim številom podpoti za vsak  $n$ , smo na osnovi začetnih rezultatov konstruirali grafe v gosenciasti (*ang. caterpillar*) in drevesni (*ang. tree*) strukturi.



Slika 6: Pendant blok na 5 vozliščih.

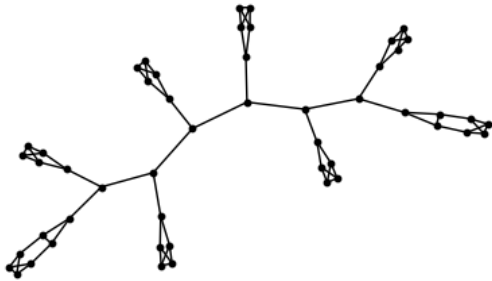


Slika 7: Pendant blok na 7 vozliščih.

## 4.1 Goseničasti konstrukciji

Na podoben način bomo konstruirali dve vrsti grafov. Obe bosta temeljili na goseničasti strukturi:

- Grafi *Caterpillar* $1_n$  ( $Cat1_n$ ) imajo v središču verigo povezanih vozlišč, na katera pripnemo pendant bloke tako, da so vsa vozlišča stopnje 3. Na vsako vozlišče verige pripnemo po en pendant blok na 5 vozliščih, kar je že dovolj, da so notranja vozlišča verige stopnje 3. Zunanijima dvema vozliščema pa dodamo še en pendant blok na 5 oziroma 7 vozliščih, glede na  $n$ .
- Grafi *Caterpillar* $2_n$  ( $Cat2_n$ ) imajo, podobno kot grafi  $Cat1_n$ , v središču verigo povezanih vozlišč, na katera pripnemo pendant bloke na 5 vozliščih. Razlika se pojavi le pri  $n = 6k + 2$  za  $k = 2, 3, 4, \dots$ , kjer namesto dveh pendant blokov na 7 vozliščih, na konec verige dodamo kopijo grafa  $K_4 - e$ . Tako dobimo podaljšano verigo, na katero so pripet le pendant bloki na 5 vozliščih.



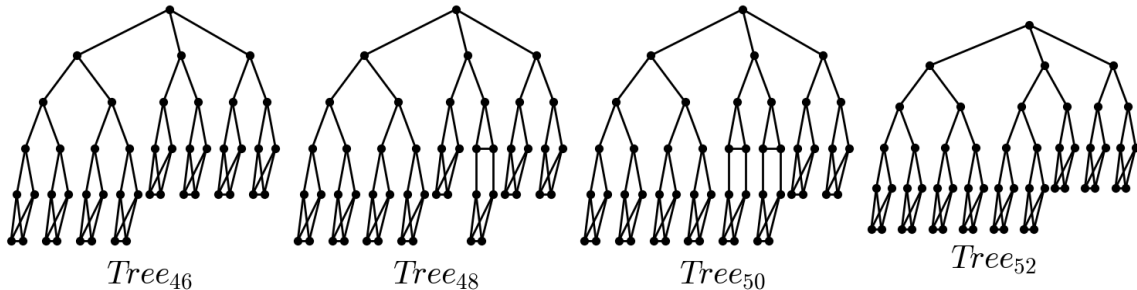
Slika 8: Graf  $Cat1_{50}$  ( $n = 50$ ).



Slika 9: Graf  $Cat2_{50}$  ( $n = 50$ ).

## 4.2 Drevesna konstrukcija

Če sta prvi dve konstrukciji grafov temeljili na verižni strukturi, bo tretja temeljila na drevesni strukturi. Grafi  $Tree_n$  (za sode  $n \geq 16$ ) imajo v središču torej drevesno strukturo, ki je razvejana tako, da je vsako notranje vozlišče stopnje 3. Na vsak list, ki je v začetni strukturi stopnje 1, pa pripnemo dva pendant bloka na 5 vozliščih. Glede na  $n$  nato enega oziroma dva pendant bloka na 5 vozliščih zamenjamo s pendant blokom na 7 vozliščih. Da lahko dosežemo željeno drevesno obliko, grafe  $Tree_n$  definiramo samo za sode  $n \geq 16$ .



Slika 10: Grafi  $Tree_n$  za  $n \in \{46, 48, 50, 52\}$ .

## 4.3 Domneva

Ker smo našo začetno domnevo ovrgli že pri precej majhnih  $n$ , sedaj šelimo postaviti boljšo domnevo. V dosedanjih testih smo ugotovili da graf  $Cat2_n$  minimizira število poti za  $n \in$

$\{10, 12, 14, 16, 18\}$ . Želimo vedeti, če je  $Cat2_n$  graf, ki minimizira število poti za vse sode  $n \geq 10$ . Prav tako pa nismo prepričani glede obnašanja grafa  $Cat2_n$  za velike  $n$  in zdi se nam, da je za večje  $n$  ustrežnejši eden izmed grafov  $Cat1_n$  ali  $Tree_n$ . Od tod sledi naslednja domneva.

**Domneva 2.** *Graf, ki minimizira število poti v grafu v razredu kubičnih grafov na  $n$  vozliščih, za  $n \geq 10$ , je enden izmed grafov  $Caterpillar1_n$ ,  $Caterpillar2_n$  ali  $Tree_n$ .*

Odslej bo torej naša naloga, da s testiranjem preverimo pravilnost naše domneve.

## 5 Testiranje domneve 2.

Število poti na novo konstruiranih grafov smo sedaj primerjali neposredno z grafi  $L_n$ . Rezultati izračunov za  $n \in \{16, 18, \dots, 40\}$  so predstavljeni v spodnji tabeli.

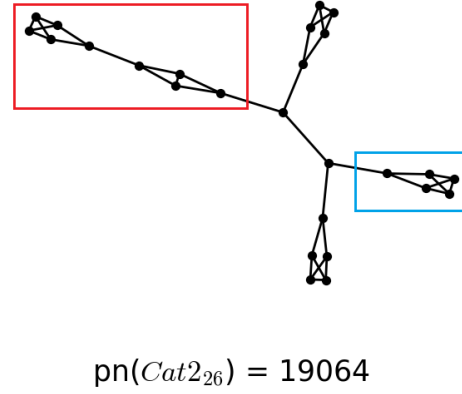
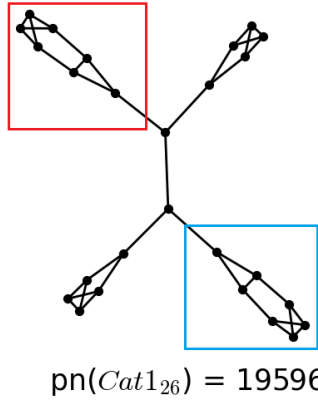
$n$	$pn(L_n)$	$pn(Tree_n)$	$pn(Cat1_n)$	$pn(Cat2_n)$
16	12.744	3.640	3.640	3.640
18	22.532	7.072	7.072	7.072
20	51.532	12.816	12.816	11.708
22	90.760	7.156	7.156	7.156
24	206.800	12.220	12.220	12.220
26	363.788	19.596	19.596	19.064
28	827.988	11.824	11.824	11.824
30	1.456.016	18.520	18.520	18.520
32	3.312.856	27.528	27.528	27.572
34	5.825.044	17.644	17.644	17.644
36	13.252.444	25.972	25.972	25.972
38	23.301.272	36.612	36.612	37.232
40	53.010.912	24.616	24.616	24.616

Tabela 4: Primerjava  $pn(L_n)$  s tremi različnimi konstrukcijami.

Hitro opazimo, da imajo vse tri konstrukcije veliko manjše število poti kot graf  $L_n$ . V primeru  $n = 40$  imajo naše konstrukcije kar 2.153-krat manjše število poti kot graf  $L_{40}$ . Opazimo lahko tudi, da imata grafa  $Cat1_n$  in  $Tree_n$  isto število poti za vsak  $n$ . Njun *subpath number* se namreč ujema vse do  $n = 300$  (testirali smo samo do  $n = 300$ ). To glede na njuno konstrukcijo ni nič presenetljivega, saj ju sestavlja isto število glavnih gradnikov in tudi število povezav med posameznimi gradniki je enako.

Poleg vsega opazimo tudi razliko v *subpath number* med grafoma  $Cat1_n$  in  $Cat2_n$ . Razlike se pojavijo pri  $n = 6k + 2$  za  $k = 2, 3, 4, \dots$ , kar bi lahko pričakovali, saj sta grafa različno konstruirana ravno pri teh  $n$ . Kot vidimo na spodnji sliki, se konstrukciji pri  $n = 26$  razlikujeta v treh ključnih gradnikih. Če izhajamo iz konstrukcije  $Cat1_{22}$  oziroma  $Cat2_{22}$  na 22 vozliščih (konstrukciji sta za  $n = 22$  enaki), opazimo, da pri konstrukciji  $Cat1_{26}$  4 nova vozlišča vpeljemo tako, da dva skrajna pendant bloka na 5 vozliščih zamenjamo z dvema pendant blokoma na 7 vozliščih. Pri konstrukciji  $Cat2_{26}$  pa 4 nova vozlišča vpeljemo tako, da središčno verigo podaljšamo z grafom  $K_4 - e$ .





Slika 11: Konstrukciji  $Cat1_n$  (levo) in  $Cat2_n$  (desno) za  $n = 26$ .

Zanimivo je, da ni ena izmed konstrukcij zmeraj boljša. Za  $n \in \{14, 20, 26\}$  ima namreč najmanjše število poti  $Cat2_n$ , medtem ko je za  $n \in \{32, 38, 44, \dots\}$  boljši graf  $Cat1_n$ .

$n$	$\text{pn}(Cat1_n)$	$\text{pn}(Cat2_n)$
14	7.188	5.504
20	12.816	11.708
26	19.596	19.064
32	27.528	27.572
38	36.612	37.232
44	46.848	48.044
50	58.236	60.008
56	70.776	73.124
62	84.468	87.392
68	99.312	102.812
74	115.308	119.384
80	132.456	137.108

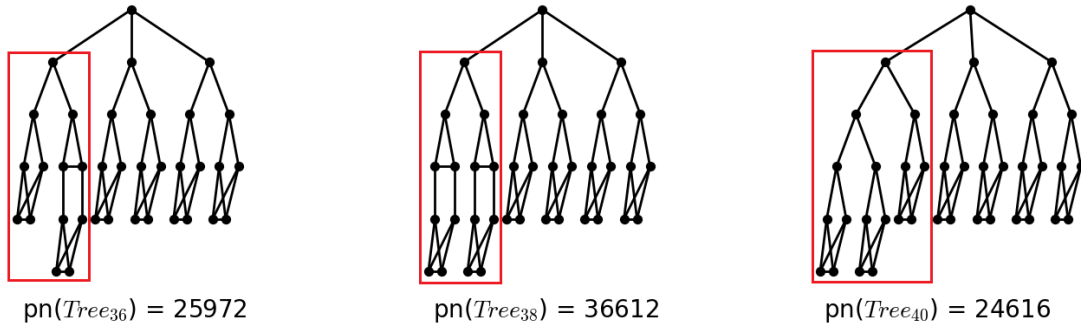
Tabela 5: Primerjava števila poti med strukturama  $Cat1_n$  in  $Cat2_n$ .

Če podrobno pogledamo rezultate primerjave števila poti grafov  $L_n$  z našimi konstrukcijami opazimo, da prav za  $n = 6k + 2$  za  $k = 2, 3, 4, \dots$  število poti v naših konstrukcijah doživi velik narast in ima višjo vrednost kot njegov naslednik  $n + 2$  ali celo  $n + 4$ . Ker se nam to ni zdelo smiselno, smo za  $n = 20$  vrednost  $\text{pn}(Cat2_{20})$ , katera je za ta  $n$  med našimi konstrukcijami najnižja, primerjali z vsemi kubičnimi grafi na 20 vozliščih. Ugotovili smo, da graf  $Cat2_{20}$  med kubičnimi grafi na 20 vozliščih minimizira število poti. To pomeni, da glede tega skoka ne moremo narediti nič. To pomeni, da število poti ni povsem odvisno od števila vozlišč, pomembni so le posamezni gradniki znotraj grafa, katerih za  $n = 6k + 2$  ne moremo konstruirati.

$n$	$\text{pn}(L_n)$	$\text{pn}(Tree_n)$	$\text{pn}(Cat1_n)$	$\text{pn}(Cat2_n)$
16	12.744	3.640	3.640	3.640
18	22.532	7.072	7.072	7.072
20	51.532	12.816	12.816	11.708
22	90.760	7.156	7.156	7.156
24	206.800	12.220	12.220	12.220
26	363.788	19.596	19.596	19.064
28	827.988	11.824	11.824	11.824
30	1.456.016	18.520	18.520	18.520
32	3.312.856	27.528	27.528	27.572
34	5.825.044	17.644	17.644	17.644
36	13.252.444	25.972	25.972	25.972
38	23.301.272	36.612	36.612	37.232
40	53.010.912	24.616	24.616	24.616

Tabela 6: Skok števila poti za  $n = 6k + 2$ .

Za skok števila poti so zagotovo krivi pendant bloki na 7 vozliščih. V konstrukcijah  $Cat1_n$  in  $Tree_n$ , ko iz  $n = 6k$  preidemo v  $n = 6k + 2$ , dve dodatni vozlišči vpeljemo tako, da enemu pendant bloku na 7 vozliščih dodamo še drugega (namesto enega pendant bloka na 5 vozliščih). Tu se v smislu števila poti naredi največji skok. Ko pa  $n$  povečamo še za 2, torej  $n = 6k + 4$ , ta dva pendant bloka na 7 vozliščih zamenjamo s tremi pendant bloki na 5 vozliščih, kar nam število poti ponovno zmanjša. Vse tri konstrukcije in njihove razlike so predstavljene na spodnji sliki.

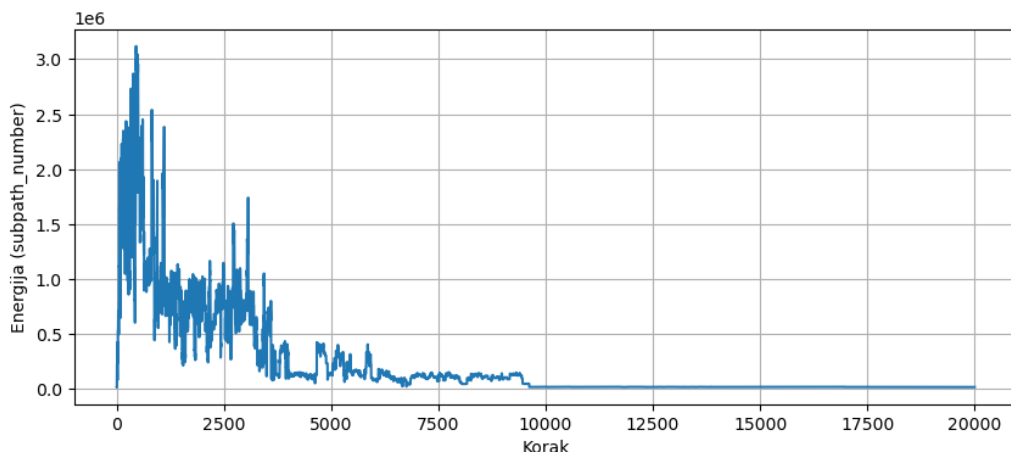


Slika 12: Razlike v konstrukcijah  $Tree_n$  za  $n = 36$  (levo),  $n = 38$  (srednji) in  $n = 40$  (desno).

## 5.1 Simulated annealing

Želimo vedeti, ali naše konstrukcije minimizirajo število poti za kubične grafe na  $n$  vozliščih. Ker smo za  $n \in \{10, 12, 14, 16, 18, 20\}$  pregledali vse kubične grafe, lahko z gotovostjo trdimo, da konstrukcija  $Cat2_n$  minimizira *subpath number* za kubične grafe na  $n$  vozliščih za  $n \in \{10, 12, 14, 16, 18, 20\}$ .

Sedaj trdimo, da so naše konstrukcije tiste, ki minimizirajo število poti na kubičnih grafih. Podobno kot smo to domnevo testirali za  $L_n$  grafe, smo tudi za te grafe protiprimere iskali s *simulated annealing* metodo. Metodo smo preizkusili za  $n \in \{22, 24, 26\}$ , kjer smo prvič za začetni približek uporabili konstrukcijo  $Cat1_n$ , drugič pa še konstrukcijo  $Tree_n$ . V obeh primerih nam je metoda za  $n = 22$  in  $n = 24$  vrnila kar  $\text{pn}(Cat1_n)$  oziroma  $\text{pn}(Tree_n)$ , prav tako pa je v primeru  $n = 26$  našla nižji *subpath number*, kateri se ujema s  $\text{pn}(Cat1_n)$ . Vse vrednosti so nam bile tako že znane, kar pomeni, da z metodo *simulated annealing* nismo našli kubičnih grafov na  $n$  vozliščih, z nižjim številom poti od naših konstrukcij za  $n \in \{22, 24, 26\}$ .



Slika 13: Potek energije med *simulated annealing* za  $n = 26$  z začetnim približkom  $Cat1_n$ .

Kar bi nas pri tej ugotovitvi lahko skrbelo je, da se je algoritem SA zataknil v nekem lokalnem minimumu. Vendar iz grafa poteka energije na sliki lepo vidimo, da je algoritem obravnaval tudi druge grafe s precej višjim številom poti. V prvih 2.000 korakih je namreč sprejel približno 35 % vseh slabših rešitev, kasneje pa vedno manj. Vsekakor pa to ne pomeni, da lahko z gotovostjo trdimo, da naše konstrukcije minimizirajo število poti med kubičnimi grafi, saj bi morali za to obravnavati čisto vse grafe, kot smo jih za  $n \leq 20$ .

## 6 Zaključek

V projektu smo preučevali domnevo, da grafi  $L_n$  minimizirajo *subpath number* med vsemi povezanimi kubičnimi grafi na  $n$  vozliščih, kjer je  $n \geq 10$  sodo. S popolnim pregledom vseh neizomorfni kubičnih grafov za  $n \leq 18$  smo ugotovili, da domneva velja le za  $n \in \{10, 12, 14\}$ , medtem ko se za  $n = 16$  in  $n = 18$  pojavijo protiprimeri. Za večje vrednosti  $n$  pa smo uporabili metahevrstični algoritem *simulated annealing*, s katerim smo za  $n \in \{20, 22, \dots, 30\}$  našli grafe z mnogo manjšim številom poti od grafa  $L_n$ . Izkazalo se je, da se razlika med številom poti grafa  $L_n$  in najdenih boljših grafov hitro povečuje z rastjo  $n$ . S tem smo dokončno ovrgli začetno domnevo.

Opazili smo, da imajo vsi najboljši najdeni grafi izrazito podobno strukturo, zgrajeno predvsem iz pendant blokov na 5 in 7 vozliščih. Na osnovi teh opažanj smo konstruirali tri grafe:  $Cat1_n$ ,  $Cat2_n$  in  $Tree_n$ . Pokazali smo, da imajo vse tri konstrukcije bistveno manjše število poti od grafov  $L_n$ ; v številnih primerih celo več tisočkrat manj. Primerjava konstrukcij razkrije tudi značilen skok v *subpath number* pri  $n = 6k + 2$ , ki ga povzroča nujna uporaba pendant blokov na 7 vozliščih. Pri manjših  $n$  smo lahko dokazali, da konstrukcija  $Cat2_n$  dejansko minimizira število poti med vsemi kubičnimi grafi (za  $n \leq 20$ ). Pri večjih  $n$  pa smo novo domnevo preverjali z metodo *simulated annealing*, ki ni našla boljših grafov od naših konstrukcij, kar dodatno potrjuje pravilnost postavljene domneve.

Ugotavljamo torej, da začetna domneva ne drži in da je minimizacija števila poti v kubičnih grafih tesno povezana s kombinatoriko pendant blokov in njihovo razmestitvijo. Nova domneva, da optimalni graf za dano sodo  $n \geq 10$  pripada eni izmed konstrukcij  $Cat1_n$ ,  $Cat2_n$  ali  $Tree_n$ , se skladno ujema z vsemi eksperimentalnimi rezultati. Za dokončno potrditev pa bo potrebna še nadaljnja analiza, ki bi v idealnem primeru v celoti opisala strukturo vseh grafov, ki minimizirajo *subpath number* za poljubno sodo število vozlišč.