

Tukaj pišem svoj del, da bova kasneje združila

9. december 2025

## 1 Simulated annealing

Za iskanje kubičnih grafov z manjšim številom podpoti od  $L_n$  grafa smo uporabili metahevristični 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*.

### 1.1 Delovanje

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. Energijška 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 podpoti 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

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

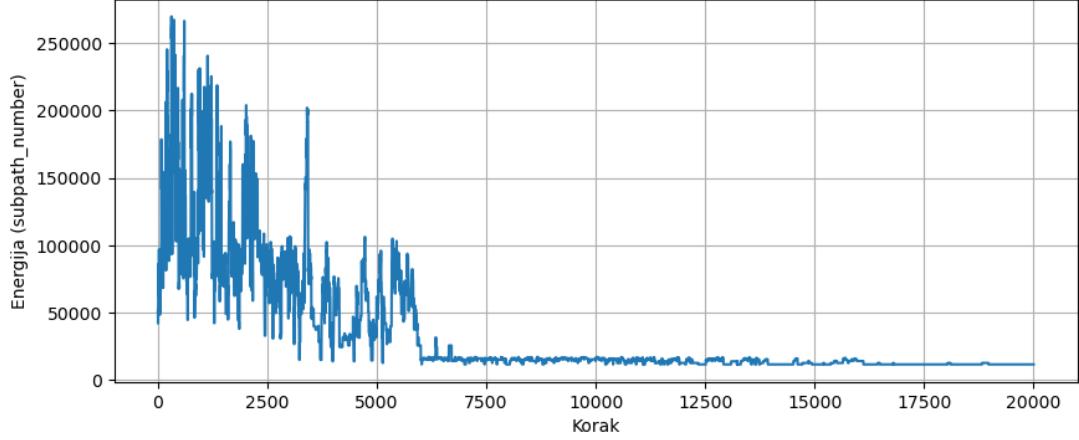
kjer  $T > 0$  predstavlja trenutni temperaturni parameter. S tem omogočimo kontrolirano sprejemanje slabših rešitev zgodaj v postopku, kar preprečuje prezgodnjo ujetost v lokalne minime. Če rešitev sprejmemo nastavimo  $G := G'$  in postopek ponavljam.

V našem primeru smo naredili 20000 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.

### 1.2 Rezultati

Z metodo *simulated annealing* smo za  $n \in \{10, 12, \dots, 30\}$  iskali grafe z manjšim številom podpoti od grafa  $L_n$ .

Na sliki lahko vidimo potek energije (števila podpoti) med procesom SA za  $n = 20$ . Kar lahko razberemo je, da so se med postopkom sprejele tudi slabše rešitve, saj energija (število podpoti) ni strogo padajoča. Program je uspešno sprejemal tudi sabše rešitve in se s tem izognil morebitnim lokalnim minimumom. Vselej pa smo na koncu našli graf...(nadaljuj tu)



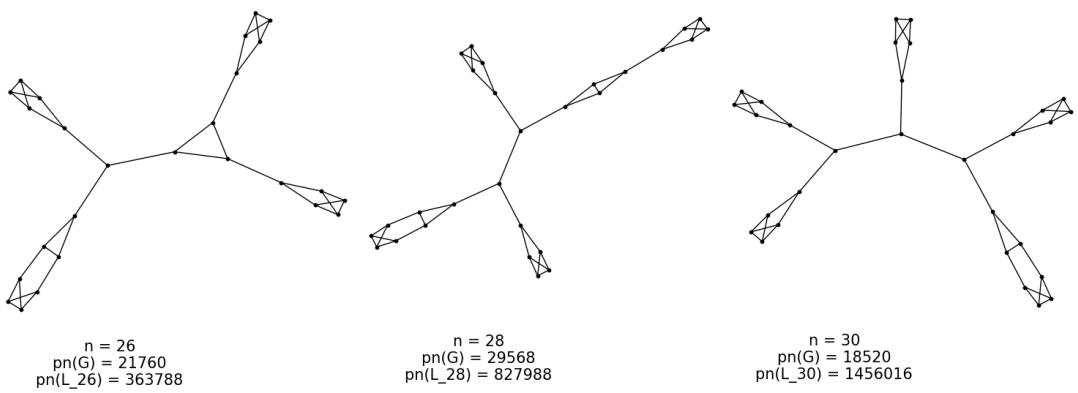
Slika 1: Potek energije med simulated annealing za  $n = 20$ .

Kot pričakovano za  $n \in \{10, 12, 14\}$  protiprimerov nismo našli, medtem ko smo za vse večje  $n$  našli grafe s precej nižjim številom podpoti. Rezultati so prikazani v spodnji tabeli.

$n$	$subpath\_number(L_n)$	boljši $subpath\_number$	razlika
10	1276	1276	0
12	3076	3076	0
14	5504	5504	0
16	12744	3640	9104
18	22532	7072	15460
20	51532	11708	38716
22	90760	7156	83604
24	206800	12220	194580
26	363788	21760	342028
28	827988	29568	798420
30	1456016	18520	1437496

Tabela 1: Rezultati SA.

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



Slika 2: Grafi pridobljeni s SA za  $n = 26, 28$  in  $30$ .