

# Minimum metric dimension

Irina Marko

mi17243@alas.matf.bg.ac.rs

Anita Jovanović

mi17227@alas.matf.bg.ac.rs

Računarska inteligencija  
Matematički fakultet  
Univerzitet u Beogradu

Septembar 2024

- ① Definicija problema
- ② Primena grube sile
- ③ Optimizacija
  - Genetski algoritam
  - VNS
- ④ Optimizacija parametara
  - Primena optimizacije parametara na genetski algoritam
  - Primena optimizacije parametara na VNS
- ⑤ Testiranje i rezultati
  - Brute force
  - Poređenje
- ⑥ Veliki grafovi
  - Veliki grafovi
- ⑦ Zaključak

- Za dati graf  $G$ , problem **minimalne metričke dimenzije** je nalaženje najmanjeg mogućeg skupa čvorova  $S$  koji diferencira sve parove čvorova u grafu. To znači da svaka dva različita čvora  $u$  i  $v$  postoji barem jedan čvor  $w$  u skupu  $S$  tako da je rastojanje od  $u$  do  $w$  različito od rastojanja od  $v$  do  $w$ .

- Metoda grube sile(eng. *Brute-force*) za nalaženje metrike dimenzije grafova podrazumeva ispitivanje svih mogućih podskupova čvorova i proveru da li svaki podskup zadovoljava uslov rezolventnog skupa.
- Ovaj pristup je teoretski moguć za manje grafove, ali postaje neefikasan za veće grafove. Kod prolazi kroz sve kombinacije čvorova pomoću funkcije *combinations* iz biblioteke *itertools* i proveru da li dati čvorovi ispunjavaju uslove rezultujućeg skupa
- Funkcija *is\_resolving\_set* proverava da li dati skup čvorova može da razlikuje sve parove čvorova u grafu na osnovu njihovih udaljenosti. Ovo je ključni deo algoritma, jer određuje da li je skup čvorova validan rešavajući skup za problem minimalne metricke dimenzije.

- Genetski algoritam, inspirisan prirodnom evolucijom, započinje sa nasumično generisanom populacijom rešenja, od kojih svako ima svoju "prilagođenost" ili "fitnes". Bolja rešenja imaju veću verovatnoću da se reprodukuju i kombinuju, simulirajući ukrštanje genetskog materijala, dok nasumične promene predstavljaju genetsku mutaciju. Ovaj proces se ponavlja kroz generacije s ciljem poboljšanja populacije.
- Rezultat genetskog algoritma je tačno ili približno rešenje problema optimizacije. Reprezentacija jedinke naziva se hromozom ili genotip, a cilj je pronaći ekstremum funkcije cilja. U svakoj generaciji se procenjuje kvalitet jedinki pomoću funkcije prilagođenosti, pri čemu se kvalitetnije jedinke biraju za reprodukciju. Nakon selekcije, primenjuju se operatori ukrštanja i moguća mutacija.

- **Elitizam** - Na početku svake iteracije identifikuju se elitisti, odnosno najbolje jedinke.

**Selekcija** - Iz populacije se biraju jedinke koje će preživeti i imati priliku za reprodukciju, pri čemu bolje prilagođene jedinke imaju veće šanse za izbor.

**Ukrštanje** - Odabrane jedinke se kombinuju kako bi se stvorile nove jedinke. Ovaj proces predstavlja simulaciju ukrštanja genetskog materijala koji se dešava kod živih organizama.

**Mutacija** - U određenim situacijama, nova jedinka može doživeti mutaciju, što podrazumeva nasumične promene u njenom genetskom materijalu. Ova operacija doprinosi očuvanju raznolikosti unutar populacije, pri čemu nove jedinke zamenjuju prethodne.

- VNS (Variable Neighborhood Search) je metaheuristička metoda namenjena rešavanju kombinatornih optimizacionih problema. Osnovna ideja ove tehnike je istraživanje prostora rešenja kroz različite "okoline", čime se obezbeđuje šira pretraga i mogućnost pronalaženja boljih rešenja.
- Jedna od ključnih karakteristika VNS-a je korišćenje različitih okolina, pri čemu svaka okolina predstavlja poseban način modifikacije rešenja. Ova raznolikost omogućava istraživanje različitih delova prostora rešenja.
- Ova metoda kombinuje intenzivnu pretragu (usmerenu na lokalna poboljšanja) s diverzifikacijom (istraživanje novih rešenja) s ciljem pronalaženja optimalnog rešenja.
- glavna petlja se izvršava do definisanog broja iteracija (*max\_iterations*) ili dok ne istekne vremensko ograničenje (*time\_limit*)

- Shaking: zamućivanje rešenja-funkcija shaking nasumično menja trenutno rešenje dodavanjem, uklanjanjem ili zamenom čvora. Ovo omogućava istraživanje novog dela prostora rešenja.
- **Lokalna pretraga:** unapređenje rešenja-funkcija `local_search` se koristi za optimizaciju novog rešenja. Ova funkcija koristi tri operacije:
  - dodavanje čvora
  - uklanjanje čvora
  - zamena čvora



- Jedna od najčešće korišćenih tehnika za optimizaciju parametara je *grid search*. Ova metoda uključuje definisanje skupa vrednosti za svaki hiperparametar, a zatim sistematsko isprobavanje svake moguće kombinacije. Cilj je minimizovati ili maksimizovati određeni kriterijum, kao što je tačnost, preciznost ili vreme izvršavanja
- Optimizacija parametara je primenjena na genetski algoritam i VNS kako bismo identifikovali optimalne vrednosti koje poboljšavaju performanse u pronalaženju minimalne metricke dimenzije velikih grafova. Ova strategija nam je omogućila da poboljšamo efikasnost i tačnost naših algoritama, što je dovelo do boljih rešenja u kraćem vremenskom periodu.

# Primena optimizacije parametara na genetski algoritam

- Funkcija `optimize_parameters` isprobava različite kombinacije parametara (veličina populacije, broj generacija, stopa mutacije, strategije selekcije) da bi se našla najbolja kombinacija za dati graf.
- U glavnoj petlji: `run_genetic_algorithm_with_params(G, population_size, generations, mutation_rate, selection_strategy, time_limit)` pokreće genetski algoritam, iterira kroz generacije, selektuje roditelje, vrši crossover, mutira potomstvo, i održava najbolju soluciju tokom procesa.

# Primena optimizacije parametara na VNS

- Funkcija `optimize_parameters(G, max_iterations_list, time_limits)` testira različite kombinacije maksimalnog broja iteracija i vremenskih ograničenja kako bi pronasla najbolje parametre za VNS.
- `VNS(G, max_iterations, time_limit)` pokreće varijantno nasumično pretraživanje i beleži najbolje rešenje tokom iteracija  
`local_search(solution, nodes_list, G, shortest_paths)` pokušava da poboljša trenutnu soluciju putem operacija dodavanja, uklanjanja ili zamene čvorova.

- Bruteforce metoda je veoma korisna za tačna rešenja, posebno na malim i jednostavnim grafovima. Međutim, zbog eksponencijalne složenosti, postaje nepraktična za veće grafove, što ukazuje na potrebu za razvojem efikasnijih algoritama za analizu metricke dimenzije.
- Za jednostavne grafove (npr. linijski ili ciklični) vreme izvršavanja može biti relativno kratko, dok su rezultati često lakši za analizu. Grafovi poput potpunih ili zvezdastih, koji imaju više veza i čvorova, mogu značajno povećati vreme izvršavanja, ali rezultati i dalje ostaju tačni.

- textbf-Brzina i efikasnost

VNS se pokazao bržim za manje grafove (do 20 čvorova) zbog svoje jednostavnosti i efikasnosti u pretrazi lokalnog prostora.

*Genetski algoritam* je bio konkurentniji na većim grafovima, posebno kada je potrebno istražiti raznolike rešenja, ali je zahtevao više vremena za konvergenciju.

## **-Kvalitet rešenja**

VNS je često pronalazio rešenja sa manjim brojem čvorova u metrickim dimenzijama za jednostavnije strukture, kao što su linijski i ciklični grafovi. GA je uspeo da pronađe dobar balans između brzine i kvaliteta rešenja u složenijim strukturama poput potpunih i zvezdastih grafova.

- **-Stabilnost**

VNS je imao manje varijabilnosti u rešenjima kada je testiran više puta na istim grafovima, dok su rezultati GA pokazali veću raznolikost, što može biti pozitivno u potrazi za globalno optimalnim rešenjem.

- **-Ponašanje sa različitim grafovima**

VNS se bolje snalazio u grafovima sa jasnim lokalnim minimumima (npr. ciklični grafovi). GA je bio efikasniji u kompleksnijim strukturama gde su potrebne raznolike strategije pretrage.

- **-Praktična upotreba**

Za manje i jednostavne grafove, preporučuje se korišćenje VNS-a zbog brzine i efikasnosti.

Za veće i kompleksnije grafove, GA bi mogao biti bolji izbor zbog svoje sposobnosti da istražuje širi prostor rešenja.

- Kada pričamo o velikim grafovima i optimizaciji parametara, VNS može bolje raditi na manje kompleksnim grafovima zbog svoje jednostavne strukture pretrage. GA može bolje raditi na grafovima sa više složenosti (npr. Barabási-Albert) zbog svoje sposobnosti da istražuje veći prostor rešenja.
- U većim i složenijim grafovima, GA bi mogao biti bolji zbog svoje sposobnosti da pronade raznovrsna rešenja kroz mutaciju i ukrštanje. U manje složenim ili jednostavnim grafovima, VNS bi mogao pružiti brža rešenja

- Ovaj rad istražuje različite pristupe za rešavanje problema pronalaženja minimalne metričke dimenzije. Svaka od ovih metoda ima svoje prednosti i nedostatke, koji se manifestuju u zavisnosti od karakteristika ulaznog grafa i zahteva aplikacije.
- **Bruteforce** metoda se pokazala kao najpouzdaniji pristup za male grafove, jer garantuje pronalaženje optimalnog rešenja. Međutim, njena vremenska složenost brzo postaje neizvodljiva za veće grafove, što je značajan nedostatak.  
**VNS** pruža efikasniji okvir za srednje velike grafove, omogućavajući brže konvergiranje ka dobrom rešenju uz razumnu garanciju kvaliteta. Njena sposobnost da se prilagodi raznim operacijama pretraživanja čini je fleksibilnim alatom za različite vrste grafova.



- **Genetski algoritam** nudi visoku fleksibilnost i može se koristiti za rešavanje kompleksnih problema u velikim grafovima. Iako ne garantuje optimalnost, njegov kapacitet za pronalaženje dobrih rešenja u razumnom vremenu čini ga pogodnim za širok spektar aplikacija. Međutim, zahteva pažljivo podešavanje parametara, što može biti izazovno.
- Na kraju, izbor metode zavisi od specifičnih potreba problema. Za male grafove, brute force ostaje najbolji izbor. Za srednje velike grafove, VNS pruža optimalnu ravnotežu između brzine i tačnosti. Dok je GA najprikladniji za velike i složene grafove, posebno kada je vreme izvršavanja kritično. Dalja istraživanja mogla bi se usmeriti na optimizaciju ovih algoritama i njihovu kombinaciju kako bi se postigli još bolji rezultati.