

# Heurističko rešavanje problema minimalnog broja zadovoljivih formula

---

POREĐENJE GENETSKOG I MEMETSKOG ALGORITMA

# Opis problema

---

Dat je skup iskaznih formula u 3KNF, pronaći valuaciju koja minimizuje broj zadovoljenih formula.

## Razlike i sličnosti sa MIN-SAT

Uopštenje MIN-SAT problema, specijalni slučaj kada su sve iskazne formule dužine jedan (broj klauza je jednak 1).

Razvijene relativno efikasne egzaktne metode za MIN-SAT, mada problem je dosta manje izučavan od MAX-SAT-a (primer MinSatz).

Uopštenje (kojim se mi bavimo) nije toliko izučavano.

# Pristupi u rešavanju

---

## Algoritam grube sile

Algoritam grube sile je neupotrebljiv za iole veće instance problema zbog eksponencijalne složenosti –  $2^n$  valuacija (gde je  $n$  broj promenljivih), proveru svake valuacije podrazumeva proveru  $k$  iskaznih formula, svaka dužine  $l$  (u smislu broja klauza). Dobija se algoritam vremenske složenosti  $O(2^n * k * l)$ .

Pribegavamo heurističkim algoritmima:

- ❖ Genetski

- ❖ Memetski

# Genetski algoritam

---

Ocena kvaliteta jedinke (fitnes funkcija):  $1 / (sat(v) + 1)$ , gde je  $v$  jedinka (valuacija), a  $sat(v)$  broj zadovoljenih iskaznih formula.

Dobijanje broja zadovoljenih formula iz fitnes funkcije:  $round(1 / fitness(v) - 1)$ .

Ruletska selekcija, jednopoziciono ukrštanje, mutacija inverzijom nasumično izabrane promenljive, elitizam.

Uslov zaustavljanja: maksimalan broj iteracija i maksimalan broj iteracija bez promene najbolje jedinke.

## Rezultati?

Rešenje skoro uvek optimalno za manje instance problema za kratko vreme izvršavanja.

# Memetski algoritam

---

Može se posmatrati kao proširenje evolucionih algoritama, u ovom slučaju genetskog algoritma. Razlika je u dodatom operatoru optimizacije – povremeno optimizujemo pojedinačne jedinke nekom S-metaheuristikom.

Isprobane S-metaheuristike:

- ❖ Lokalna pretraga
- ❖ Simulirano kaljenje
- ❖ Redukovana metoda promenljivih okolina

Motivacija – nadamo se se da će memetski algoritam uspeti da nadmaši genetski tako što će se optimizacijom izabrati sa manjim promenama u jedinkama potrebnim za optimalno rešenje.

# Memetski algoritam – dodatni parametri

---

Kao dodatne parametre uvodimo optimizator i frekvenciju optimizacije.

Frekvencija optimizacije – ne želimo da previše procesorskog vremena trošimo u optimizaciji jedinki, pa je vršimo samo povremeno.

# Način sprovođenja eksperimenta

---

Za svaku instancu problema pokrećemo određeni broj izvršavanja nekog algoritma, pratimo najbolje nađeno rešenje u svim izvršavanjima, najgore nađeno rešenje (od najboljih iz pojedinačnih izvršavanja) i prosek. Pratimo i prosečno vreme izvršavanja algoritma na datoj instanci problema.

## Rezultati?

Memetski algoritam se ponaša gore od genetskog. Prosečno vreme izvršavanja (što je bilo i očekivano) je višestruko gore od prosečnog vremena za genetski algoritam, ali isto tako su gora i nađena rešenja. Razlog za ovo je verovatno prebrza konvergencija ka lokalnom optimumu usled koraka optimizacije.

Od isprobanih optimizatora, najgore se pokazao optimizator na bazi redukovane metode promenljivih okolina, dok su optimizatori na bazi lokalne pretrage i simuliranog kaljenja imali slične performanse.

# Memetski algoritmi su loši?

---

Memetski algoritmi nisu loši. Ne postoji heuristički algoritam koji je optimalan za svaki moguć optimizacioni problem (No Free Lunch Theorem).

Drugačije varijante memetskih algoritama bi se vrlo moguće pokazale bolje. Neke varijante vrše optimizaciju S-metaheuristikama tek na kraju izvršavanja dela koji odgovara datom evolutivnom algoritmu, pa nije moguće dobijati gora rešenja od onih dobijenih genetskim. Neke naprednije varijante smanjuju verovatnoću zaglavljivanja u lokalnom optimumu podelom populacije na više delova koji se zasebno optimizuju, pa se međusobno mešaju (SFLA – Shuffled Frog Leaping Algorithm).



# Linkovi

---

Implementacija i rad: [https://github.com/MistFuror/ri\\_minsat](https://github.com/MistFuror/ri_minsat)

CNFgen biblioteka za generisanje iskaznih formula u KNF-u: <https://pypi.org/project/CNFgen/>