

# Rešavanje problema maksimalne k-zadovoljivosti optimizacijom rojem čestica

David Dimić  
Zorana Gajić

Mart 2019.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Formulacija problema</b>	<b>3</b>
<b>3</b>	<b>Optimizacija rojem čestica (PSO)</b>	<b>3</b>
3.1	Pseudokod PSO . . . . .	4
3.2	Reprezentacija rešenja . . . . .	5
3.3	Inicijalizacija . . . . .	5
3.4	Fitnes funkcija . . . . .	5
3.5	Lokalna pretraga i flip heuristika . . . . .	5
3.6	Kriterijum zaustavljanja . . . . .	6
3.7	Rezultati . . . . .	6

## 1 Uvod

## 2 Formulacija problema

Data je formula  $F$  u KNF obliku sa  $n$  promenljivih  $(x_1, x_2, \dots, x_n)$  i  $m$  klauza. Problem maksimalne  $k$ -zadovoljivosti može se definisati na sledeći način:

Klauza  $C_i$  dužine  $k$  je disjunkcija  $k$  literala:  $C_i = (x_1 \vee x_2 \dots \vee x_k)$ , gde je svaki literal promenjiva ili njegova negacija i može se pojavljivati više puta u izrazu. Cilj je pronaći istinitosne vrednosti promenljivih, valuaciju koja je vektor  $\vec{v} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$  tako da ova valuacija maksimizuje broj zadovoljenih klauza u formuli  $F$ .

Ako valuacija zadovoljava formulu, onda se ona naziva modelom formule. Max  $k$ -SAT problem može se definisati parom  $(\Omega, SC)$ , gde je  $\Omega$  skup svih potencijalnih rešenja iz  $\{0, 1\}^n$ , a  $SC : \Omega \rightarrow N$ , skor valuacije koji je jednak broju zadovoljenih klauza. Shodno ovome, problem max  $k$ -SAT je naći  $\vec{v} \in \Omega$  za koje je  $SC$  maksimalno:

$$\max_{\vec{v} \in \Omega} \{SC(\vec{v})\}$$

Očigledno ima  $2^n$  potencijalnih rešenja koji zadovoljavaju formulu  $F$ . Problem max  $k$ -SAT je NP-kompletna za svako  $k > 2$ .

## 3 Optimizacija rojem čestica (PSO)

Optimizacija rojem čestica (Particle swarm optimization – PSO) je jedna od tehnika pretraživanja zasnovana na populaciji kao što je genetski algoritam, ali ne koriste evolutivne algoritme kao što su mutacija i ukrštanje. PSO algoritmi su 1995. godine uveli Kenedi i Eberhart kao alternativu standardnim genetskim algoritmima.

Optimizacija rojem čestica je algoritam zasnovan na ponašanju pojedinačnih jedinki unutar određene grupe (na primer, jata ptica ili roja insekata). Ukoliko se, vođeno instiktom, jato prica uputi u određenom smeru u potrazi za hranom, očekivanje je da će čitavo jato slediti upravo onu pticu koja je pronašla izvor hrane. Međutim, i svaka ptica ponaosob može biti vođena sopstvenim instiktom i time na trenutak u potrazi za hranom napustiti jato. Tada se verovatno može desiti da, ukoliko pronađe bolji izvor hrane, čitavo jato upravo krene da sledi tu pticu.

PSO pripada skupu algoritama koji se zasnivaju na inteligenciji roja (swarm intelligence). Algoritam radi nad skupom jedinki, koji se naziva rojem. Elementi ovog skupa se nazivaju česticama. Svaka čestica predstavlja kandidatsko rešenje optimizacionog problema. Čestice se na unapred definisan način kreću po prostoru pretraživanja. Njihovo kretanje se usmerava imajući u vidu njihovu trenutnu poziciju, njihovu do sada najbolju poziciju, kao i do sada najbolju poziciju čitavog roja. Pod najboljom pozicijom čitavog roja se podrazumeva do sada najbolja pozicija, uzimajući u obzir sva njegova rešenja. Proces se ponavlja dok ne bude zadovoljen kriterijum zaustavljanja, a u svakoj iteraciji se ažurira najbolja vrednost rešenja za svaku česticu, kao i za roj u celini.

Neka je dat roj sa  $\vec{S}$  čestica. Svaka čestica se sastoji od tri elementa:

- Pozicija u prostoru za pretragu  $\vec{x}_i$
- Brzina, vektor  $\vec{v}_i$
- Sećanje, koje se koristi za skladištenje elitnih čestica globalne pretrage  $\vec{P}_g$ , kao i najboljih individualnih rešenja  $\vec{P}_i$  koja su do sada pronašle zasebne čestice

Nije neophodno da se u budućim populacijama nalazi bilo koji elitni pojedinac, iako svaka čestica u populaciji pokušava da bude blizu svog najboljeg rešenja i globalnog najboljeg rešenja.

Osnovni oblik PSO algoritma dat je sledećim samoažurirajućim jednačinama:

$$\vec{v}_i^{t+1} = w \cdot \vec{v}_i^t + c_1 \cdot \vec{r}_1 \times (\vec{P}_i^t - \vec{x}_i^t) + c_2 \cdot \vec{r}_2 \times (\vec{P}_g^t - \vec{x}_i^t) \quad (1)$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \quad (2)$$

Jednačina 1 opisuje kako se ažurira brzina  $i$ -te čestice, a 2 koja je sledeća pozicija  $i$ -te čestice, pri čemu je:

- $w$  - faktor inercije
- $c_1, c_2$  - faktori učenja: kognitivna i socijalna
- $\vec{v}_i^t$  - brzina  $i$ -te čestice u iteraciji  $t$
- $\vec{x}_i^t$  - pozicija  $i$ -te čestice u iteraciji  $t$
- $\vec{r}_1, \vec{r}_2$  - pseudoslučajni brojevi iz uniformnog intervala  $[0, 1]$
- $\vec{P}_i$  - najbolje individualno rešenje čestice  $i$
- $\vec{P}_g$  - trenutno najbolje globalno rešenje

Kako je max k-SAT problem diskretan potrebno je prilagoditi jednačinu 2. Izračunata brzina  $\vec{v}_i$  je iz  $\mathbb{R}^n$ , pa je potrebno da se svede na  $\{0, 1\}^n$ . Jedan predlog za ažuriranje položaja čestice, izložen u radu [1], dat je sigmoidnom transformacijom. Sada  $v_i^t$  predstavlja verovatnoću da bit  $x_i^t$  uzme vrednost 1.

$$x_i^t = \begin{cases} 1, \text{rand}(0, 1) < \text{sigmoid}(v_i^t) \\ 0, \text{inace} \end{cases} \quad (3)$$

$$\text{sigmoid}(v_i^t) = \frac{1}{1 + e^{-v_i^t}} \quad (4)$$

### 3.1 Pseudokod PSO

**Input** : Formula  $F$  u KNF-u,  $n$  i  $m$

**Output**: Najbolja procenjena valuacija i broj zadovoljenih klauza

inicijalizacija populacije: pozicije i brzine;

$t = 0$ ; // **tekuca iteracija**

**while** nije zadovoljen uslov zaustavljanja **do**

$t = t + 1$ ;

**for**  $i \leftarrow 0$  **to** broj čestica u roju **do**

        Izračunaj  $\text{Fitness}(\vec{P}_i^t)$ ;

        Sačuvaj individualni najbolji rezultat kao globalni  $\vec{P}_g$ ;

        Ažuriraj brzine na osnovu  $\vec{P}_i$  i  $\vec{P}_g$ ;

        Ažuriraj pozicije  $\vec{v}_i^t$ ;

        Ažuriraj individualni najbolji rezultat  $\vec{P}_i$ ;

        Ažuriraj globalni najbolji rezultat  $\vec{P}_g$ ;

**end**

**end**

**Algorithm 1:** Uopšteni PSO algoritam

### 3.2 Reprezentacija rešenja

Za dobar algoritam je takođe važno dobro predstavljanje rešenja. Postoje više načina reprezentacija, ali je ovde odabran prirodna binarna reprezentacija. Svaka čestica je predstavljena binarnim nizom dužine  $n$ .

### 3.3 Inicijalizacija

Potrebno je inicijalizovati pozicije čestica i vektora brzine. Pozicije su inicijalizovane pseudo-slučajnim brojevima  $\{0, 1\}$ , a vektor brzine realnim brojevima iz intervala  $[-V_{min}, V_{max}]$ , gde su granice intervala jedan od parametara PSO algoritma.

### 3.4 Fitnes funkcija

Fitnes funkcija je veoma važna za performanse algoritma. Prva fitnes funkcija koja se sama nameće jeste broj zadovoljenih klauza. Ova funkcija se nije pokazala kao dovoljno dobra, a bolji mehanizam je stepenasto ažuriranje težina (SAW - stepwise adaptation weights) uvedena od strane Eiben-a [2]. Fitnes funkcija je data sledećom formulama:

$$FitnessSAW(x) = \sum_{i=1}^m W_i C_i(x) \quad (5)$$

$$W_{i+1} = W_i + 1 - C_i(x^*) \quad (6)$$

Svakoј klauzi  $C_i$  dodeljuje se težina  $W_i$ . Ova funkcija ima za cilj identifikovanje težih klauza u procesu učenja koja je predstavljena većom vrednošću  $W_i$ . Na početku su težine inicijalizovane na 1, pa se potom ažuriraju jednačinom 6.  $x^*$  je tekuće najbolje rešenje.

### 3.5 Lokalna pretraga i flip heuristika

Da bi se unapredio standardni PSO algoritam uvedena je, umesto ažuriranja pozicija jednačinom 3, stohastička lokalna pretraga flip heuristikom.

**Input** : pozicija čestice  $p$ , Formula  $F$  u KNF-u,  $maxFlip$

**Output**: nova pozicija čestice  $p$

```
improvement = 1;
numFlip = 0;
while improvement > 0 and numFlip < maxFlip do
    improvement = 0;
    for  $i \leftarrow 0$  to  $n$  do
        flip  $p[i]$ ;
        numFlip += 1;
        Izračunaj dobit: gain;
        if gain >= 0 then
            | prihvati flip;
        end
        else
            | odbaci flip, vrati na staru vrednost  $p[i]$ ;
        end
    end
end
```

**Algorithm 2:** Funkcija lokalne pretrage

### 3.6 Kriterijum zaustavljanja

Jedini kriterijumi zaustavljanja koji se mogu koristiti u opštem slučaju su: da li je dostignut maksimalan broj unapred zadatih iteracija ili, da li u poslednjih nekoliko iteracija nema značajnog napretka. Ali, u test primerima za koje unapred znamo da je formula zadovoljiva, možemo koristiti i kriterijum da je broj zadovoljenih klauza dostigao ukupan broj klauza  $m$ .

### 3.7 Varijante PSO algoritma

### 3.8 Rezultati

## Literatura

- [1] Kennedy J, Eberhart RC. “A discrete binary version of the particle swarm algorithm,” In: Proceedings of IEEE conference on systems, man and cybernetics , p. 4104–4109, 1997.
- [2] Gottlieb J, and Voss N. “Adaptive fitness functions for the satisfiability problem”. In: Parallel Problem Solving from Nature - PPSN VI 6th International Conference, Paris, France. Springer Verlag. LNCS 1917, 2000.
- [3] Abdesslem Layeb, “A particle swarm algorithm for solving the maximum satisfiability problem”, MISC Lab., Computer science department, University mentouri of Constantine, Algeria