

Optimizacija rojem čestica

Nevena Soldat, Tijana Živković, Ana Miloradović, Milena Kurtić

Matematički fakultet, Univerzitet u Beogradu

April 20, 2020

- Uvod u optimizaciju rojem čestica
- Osnovni algoritam
- Varijacije parametara
- Primena
- Topologije
- Literatura

Šta je optimizacija rojem čestica?

- optimizaciona tehnika zasnovana na inteligentnom ponašanju nekih organizama, kao što su insekti, ptice i ribe

Nastanak:

- Eugene Marais - The Soul of the White Ant (1926)
- Marco Dorigo - ponašanje kolonije mrava (1990-ih)
- Eberhart i Kennedy - algoritam optimizacije rojem čestica (1995)

Algoritam za optimizaciju rojem čestica je otkriven sasvim slučajno, pri pokušaju da se na računaru simulira kretanje jata ptica.

Osnovni algoritam

Osnovni koncepti:

- čestice se kreću kroz višedimenzioni prostor pretrage
- svaka čestica predstavlja jedno moguće rešenje

Neka je $x_i(t)$ pozicija čestice i u trenutku t . Pozicija se menja dodavanjem brzine $v_i(t)$ na trenutku poziciju:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

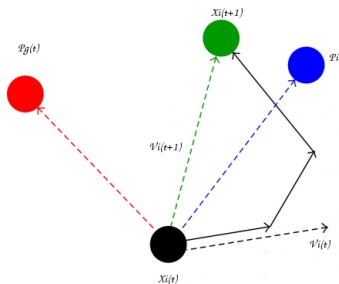
Brzina se računa kao:

$$v_i(t+1) = v_i(t) + c_1 r_1 (p_i(t) - x_i(t)) + c_2 r_2 (p_g(t) - x_i(t))$$

gde su:

- $p_i(t)$ - najbolja pozicija koju je čestica i pronašla do trenutka t
- $p_g(t)$ - najbolja pozicija u čitavom roju do trenutka t
- r_1, r_2 - nasumične vrednosti iz $U[0,1]$
- c_1, c_2 - konstante

Komponente brzine



Slika: Komponente brzine

- **moment** - prethodno stanje brzine
- **kognitivna** komponenta - tendencija vraćanja u lično najbolje
- **socijalna** komponenta - tendencija ka kretanju ka naboljem globalnom

Osnovne varijacije: Smanjenje brzina

Modifikacije osnovnog algoritma optimizacija rojem čestica

- **Eksploracija** - istraživanje različitih delova prostora pretrage
- **Eksploatacija** - koncentrisanje oko regije koje garantuje nalaženje rešenja

Smanjenje brzina uvodi se da bi se sprečilo da čestice napuste prostor pretrage usled čestog ažuriranja položaja

$$v_i(t+1) = \begin{cases} v'_i(t+1), & v'_i(t+1) < V_{max} \\ V_{max}, & v'_i(t+1) \geq V_{max} \end{cases} \quad (1)$$

Osnovne varijacije: Inercijske težine

U osnovnu formulu ažuriranja brzine uvodi se težina w koji kontroliše koliko će prethodni pravac leta uticati na novu brzinu, kao i sam balans izmeu istraživanja i eksploatacije

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)]$$

Veće vrednosti w olakšavaju eksploraciju, dok manje vrednosti w podstiču lokalnu eksploataciju

- $w \geq 1$ - roj divergira, brzine idu ka maksimalnoj brzini
- $0 < w < 1$ - čestice usporavaju, pa konvergencija zavisi od c_1 i c_2
- $w < 0$ - brzine opadaju sve dok ne stignu do nule i time se algoritam zaustavlja

Osnovne varijacije: Koeficijent suženja i model brzine

Ovaj princip je sličan inercijskoj težini, ali se brzine ograničavaju konstantom χ

$$v_{ij}(t+1) = \chi[v_{ij}(t) + \phi_1(y_{ij}(t) - x_{ij}(t)) + \phi_2(\hat{y}_j(t) - x_{ij}(t))]$$

gde je

$$\chi = \frac{2k}{|2 - \phi - \sqrt{\phi(\phi - 4)}|},$$

sa $\phi = \phi_1 + \phi_2$, $\phi_1 = c_1 r_1$ i $\phi_2 = c_2 r_2$

Modeli brzina se razlikuju u komponentama koje su uključene u jednačinu brzine

- **Kognitivni** model - isključuje socijalnu komponentu
- **Socijalni** model- isključuje kognitivnu komponentu
- model u kojoj sama **čestica neće sebe izabrati za najbolju** - najbolje rešenje se bira iz susedstva

- Kontinualni i diskretni problemi
- **Oblasti primene:** Neuronske mreže, telekomunikacije, istraživanje podataka, dizajn(antene, bluetooth), kombinatorna optimizacija, biomedicina itd.
- **Primer:** PSO koji rešava problem planiranja razvoja distributivnih mreža
- Šta je **distributivna mreža**?
 - deo elektroenergetskog sistema koji omogućava da se električna energija distribuira do srednjih i malih potrošača.
- **Proces planiranja** razvoja distributivnih mreža:
 - Šta?
 - Gde?
 - Kada?

- **Formulacija problema**

- **Faktori** - minimizacija gubitaka energije, ulaganja u nove objekte i distributivne vodove, maksimizacija pouzdanosti sistema
- **Ograničenja** - kapacitet vodova, naponski nivo opterećenja u čvorovima, radijalnost mreže
- **Funkcije cilja** - gubitak energije, pouzdanost sistema
- **Šema kodiranja čestica** - kodira topologiju mreže, direktno i indirektno kodiranje
- **Šema dekodiranja čestica** - troškovno pristrasno dekodiranje

Algoritam 2 *PSO koji rešava problem planiranja razvoja distributivne mreže*

Početak: /*Inicijalizacija - podešavanje veličine populacije i maksimalnog broja iteracija k_{max} */
 $k = 0$;
 $s^k = s^0$; /*Generisanje početne tačke pretrage za celu populaciju po predloženoj šemi kodiranja */
 $v^k = v^0$; /*Generisanje početne brzine pretrage za celu populaciju po predloženoj šemi kodiranja */
Dekodiranje svih čestica; /*Po predloženoj šemi dekodiranja*/
Oceniti $f(k)$; /* Izračunati vrednosti funkcije cilja svake čestice u tekućoj populaciji */
Ponavljati: /*Globalna iteracija, k*/
 $v_i^k + 1 = wv_i^k + c_1rand_1x(pbest_i - s_i^k) + c_2rand_2x(gbest_i - s_i^k)$; /*Ažuriraj brzinu za sve čestice */
 $s_i^{k+1} = s_i^k + v_i^{k+1}$; /*Ažuriraj poziciju za sve čestice*/
Oceniti $f(k)$; /* Izračunati vrednost funkcije cilja svake čestice u tekućoj populaciji */
if $f(k) > pbest(k)$ then
 $pbest(k+1) = f(k)$ /*Prihvatiti funkciju cilja $f(k)$ čestice kao $pbest(k+1)$ za tu česticu*/
end if
if $pbest(k) > gbest(k)$ then
 $gbest(k+1) = pbest(k)$ i pamti k-tu česticu /*Prihvatiti $pbest(k)$ kao $gbest(k+1)$ i sačuvaj k-tu česticu kao najbolju.*/
end if
 $k = k + 1$;
Dok nije ispunjen zahtev za zaustavljanje; /* $k > k_{max}$ */
Izlaz: Najbolje nađeno rešenje.
