

# Optimizacija rojem čestica

Seminarski rad u okviru kursa  
Metodologija stručnog i naučnog rada  
Matematički fakultet

Nevena Soldat, Milena Kurtić, Tijana Živković, Ana Miloradović  
nevenasoldat@gmail.com, mimikurtic67@gmail.com, tijanazivkovic6@gmail.com, ana.miloradovic@gmail.com

16. mart 2020.

## Sažetak

U ovom radu opisaćemo osnovni algoritam za optimizaciju rojem čestica, kao i neke od postojećih varijacija. Objasnićemo jednostavnije algoritme sa jedinstvenim rešenjem, ali i neke naprednije. Tema će biti i socijalne strukture na kojima se on zasniva, kao i koje su moguće primene.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Osnovni algoritam za optimizaciju rojem čestica</b>	<b>2</b>
2.1	Globalno najbolji PSO . . . . .	2
2.2	Lokalno najbolji PSO . . . . .	3

# 1 Uvod

Inteligencija rojeva predstavlja jednu od pet glavnih paradigmi Računarske Inteligencije (Computation Intelligence - CI). Jedinke u okviru grupe (roja) dele prikupljene informacije u zajedničkom cilju da reše neki problem, koje se propagiraju kroz celu grupu tako da se problem rešava mnogo efikasnije nego što bi to mogla pojedinačna jedinka. Prvi i dosta značajan doprinos u polju inteligencije rojeva imao je južnoafrički pesnik Eugene N Marais koji je proučavao socijalno ponašanje kako majmuna, tako i mrava. Posle njega, ranih 1990-ih godina, Marco Dorigo modeluje ponašanje kolonija mrava. Zatim, 1995, Eberhart i Kennedy razvijaju algoritam optimizacije rojem čestica, na osnovu posmatranog jata ptica. Optimizacija rojem čestica (eng. Particle Swarm Optimization - PSO) je stohastička optimizaciona tehnika zasnovana na veoma inteligentnom kolektivnom ponašanju nekih organizama kao što su insekti, ptice i ribe. Algoritam optimizacije rojem čestica otkriven je sasvim slučajno od strane Eberharta i Kenedija, pri pokušaju da se na računaru simulira kretanje jata ptica. Prvobitna namera bila je da se grafički prikaže nepredvidiva koreografija jata ptica, sa ciljem da se otkriju obrasci koji omogućavaju pticama da lete sinhronizovano, i da zadrže optimalnu formaciju pri naglim promenama pravaca. Sada je cilj kreiranje jednostavnog i efikasnog optimizacionog algoritma. Od kada je prvi put predstavljen 1995. doživeo je niz poboljšanja i nastale su brojne varijacije ovog algoritma.

## 2 Osnovni algoritam za optimizaciju rojem čestica

Algoritam za optimizaciju rojem čestica sadrži jedinke (čestice) koje se se kreću kroz višedimenzioni prostor pretrage, a pozicije jedinki se menjaju u skladu sa sopstvenim iskustvom, kao i sa iskustvom susednih jedinki. Svaka od tih čestica predstavlja jedno moguće rešenje. Neka je

$$x_i(t)$$

pozicija čestice  $i$  u prostoru pretrage u trenutku  $t$ . Pozicija čestice se menja dodavanjem brzine,

$$v_i(t)$$

na trenutnu poziciju.

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

Njihovo kretanje se usmerava imajući u vidu njihovu trenutnu poziciju, njihovu do sada najbolju poziciju, kao i do sada najbolju poziciju čitavog roja. Znanje čestice stečeno sopstvenim iskustvom predstavlja kognitivnu komponentu algoritma, i proporcionalno je udaljenosti čestice od svoje najbolje pozicije do sada. Socijalna komponenta algoritma predstavlja informacije koje su stečene od suseda posmatrane čestice. Prvobitno su razvijene dve varijante PSO algoritama koje se razlikuju u odnosu na broj susednih jedinki, a to su gbest PSO i lbest PSO.

### 2.1 Globalno najbolji PSO

U slučaju globalno najboljeg PSO algoritma, susedi za svaku česticu su zapravo čitav roj. On implementira topologiju zvezde. U topologiji zvezde

socijalna komponenta brzine čestice predstavlja informaciju dobijenu od svih čestica u roju. Brzina čestice  $i$  se računa kao:

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

## 2.2 Lokalno najbolji PSO

Lokalno najbolji PSO koristi topologiju prstena, gde svaka čestica  $i$  ima manji broj suseda. Socijalna komponenta predstavlja informaciju koju razmenjuju susedi čestice. Doprinos jedinice brzini je proporcionalna razdaljini između čestice  $i$  i najbolje pozicije koju su pronašli susedi. Brzina se računa kao:

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