

# Usporedba algoritama baziranih na roju pri traženju globalnih ekstrema

Marko Stojanović,  
Vanja Vuković

11. veljače 2015.

- 1 Osnovni pojmovi i problematika
- 2 U znanstvenoj i stručnoj literaturi
- 3 O projektnom zadatku
- 4 Algoritmi
- 5 Testiranje i primjeri testnih funkcija
- 6 Rezultati testiranja
- 7 Grafičko sučelje
- 8 Literatura

# Problem globalnog optimuma za nelinearne probleme

Dvije osnovne vrste problema:

- traženje optimuma na segmentu domene (npr. Schafferova F6 funkcija na  $[-100, 100]^2$ )
- traženje optimuma funkcija ograničenih sa nekim jednakostima i/ili nejednakostima na cijeloj domeni (npr. G funkcije).

# Meta-heuristike

- Meta-heuristike su strategije koje efikasno pretražuju prostor rješenja te pronalaze ona rješenja koja su blizu optimalnih.
- Postoje dvije glavne podjele metaheuristika:
  - strategija pretraživanja
    - bazirane na lokalnom pretraživanju
    - bazirane na komponentama koje *uče*
  - dimenzija
    - jedno rješenje
    - skup mogućih rješenja.

# Populacijski algoritmi

- Velika klasa algoritama:
  - evolucijski algoritmi, genetski algoritmi, PSO, ...
- Algoritmi bazirani na rojevima (eng. swarm intelligence) su podklasa populacijskih algoritama koji se oslanjaju na kolektivno ponašanje decentraliziranih i samostalnih čestica ili jedinki u roju.
- Primjer takvih algoritama su: PSO, mravlji algoritam i pčelinji algoritam.

# U znanstvenoj i stručnoj literaturi



W. ZHU, J. CURRY, *Parallel Ant Colony for Nonlinear Function Optimization with Graphics Hardware Acceleration*, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (2009) 1803–1808

- Mravlji algoritam
- Ackley, Griewank, Penalty1, Penalty2, Quadratic, Rosenbrock, Rastrigin, Schwefel 1.2, Schwefel 2.22, Schwefel 2.21, Sphere, Step

# U znanstvenoj i stručnoj literaturi



B. ALATAS, *Chaotic bee colony algorithms for global numerical optimization*, Expert Systems with Applications 37 (2010) 5682–5687

- Pčelinji algoritam
- Griewank, Rosenbrock, Rastrigin

# U znanstvenoj i stručnoj literaturi



F. AGHAZADEH, M. R. MEYBODI, *Learning Bees Algorithm For optimization*, International Conference on Information and Intelligent Computing IPCSIT vol.18 (2011) 115–122

- Pčelinji algoritam
- De Jong<sup>1</sup> 1 i 2, Rastrigin, Ackley

---

<sup>1</sup>Postoji 5 De Jong funkcija: Sphere, Rosenbrock (Rosenbrock's vally), Step, Quadratic i Shekel's Foxholes



# U znanstvenoj i stručnoj literaturi



K. M. MALAN, A. P. ENGELBRECHT, *Characterising the searchability of continuous optimisation problems for PSO*, accepted for publication, Swarm Intelligence, SpringerLink vol.8 (2014)

- PSO
- Ackley, Griewank, Quadric, Rana, Rastrigin, Rosenbrock, Salomon, Spherical, Schwefel 2.26, Step

# U znanstvenoj i stručnoj literaturi



R. RAHMANI, R. YUSOF, *A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: Radial Movement Optimization*, Applied Mathematics and Computation 248 (2014) 287–300

- RMO – vrsta PSO algoritma
- De Jong 1-5, Schaffer F6, Rastrigin, Griewank, Hyper-Ellipsoid, Ackley

# Projektni zadatak

- Implementacija 3 algoritama iz klase algoritama baziranih na roju (ABC, PSO, OPSO) u programskom jeziku C.
- Izvšavanje programskog koda na apache serveru.
- Korisničko sučelje izrađeno web tehnologijama.
- Putem korisničkog sučelja korisnik može određivati neke zajedničke (broj iteracija, testna funkcija) i neke specifične parametre (koeficijenti koji se koriste u algoritmu).
- Grafički prikaz rezultata.

# Pčelinji algoritam (ABC) - 1

- Algoritam predstavlja roj pčela koje traže nektar u cvijetnom polju.
- Puno parametara:
  - izviđači,
  - radilice (dva parametra),
  - 'elitni' izviđači  $\subset$  izviđači (dva parametra),
  - veličina okoline za pretraživanje,
  - dubina istraživanje dane okoline.
- Ne konvergira.

## Pčelinji algoritam (ABC) - 2

- Nasumično odaberemo mjesto na kojem će izviđači tražiti.
- U  $\epsilon$  okolinu najboljih  $e$  izviđača (eltni) saljemo po  $ne$  radilica, a u kolinu sljedećih  $m$  najboljih šaljemo po  $nm$  radilica ( $\epsilon = (|b_d| + |b_g|)/n$ ).
- $k$  'poditeracija' pretražujemo tu okolinu (fiksirana vrijednost u testiranju  $k = 10$ ).
  - Ako je pronađena bolja vrijednost u  $\epsilon$  okolini, suzi okolinu  $\epsilon = \epsilon * 0.8$
- Zapamti najbolju vrijednost i ponovi ciklus.

# Particle swarm optimisation - 1

- Algoritam se bazira na kretanju jata riba (i ptica).
- 4 promjenjiva parametra:
  - broj čestica u roju,
  - inercija čestice ( $\omega$ ), osobna komponenta čestice ( $\rho_p$ ), socijalna komponenta čestice ( $\rho_g$ ).
- Algoritam konvergira.

# Particle swarm optimisation - 2

1. Nasumično raspodijeli čestice po prostoru pretraživanja i zadaj im brzinu
2. Izračunaj novu poziciju:  $x_i + v_i \mapsto x_i$
3. Izračinaj novu brzinu:  $v_i = \omega v_{i,d} + \rho_p(p_i - x_i) + \rho_g(g - x_i)$
4. Ako nije zadovoljen uvijet vrati se na 2.

# Orbital particle swarm optimisation - 1

- Algoritam se zasniva na klasičnom PSO algoritmu uz lokalnu pretragu kao u ABC.
- Uz varijable iz PSO algoritma dodana je i varijabla  $\epsilon$  koja određuje okolinu za lokalno pretraživanje za najbolju česticu
- Lokalno pretraživanje se vrši po orbitama dane čestice fiksiranjem jedne dimenzije i iteracijom po dimenzijama (za točku u dimenziji 30 dobijemo 30 novih točaka)



# Orbital particle swarm optimisation - 2

1. Nasumično raspodijeli čestice po prostoru pretraživanja i zadaj im brzinu
2. Izračunaj novu poziciju:  $x_i + v_i \mapsto x_i$
- 3.\* Za najbolju česticu izvrši pretragu po orbitama
4. Izračinaj novu brzinu:  $v_i = \omega v_{i,d} + \rho_p(p_i - x_i) + \rho_g(g - x_i)$
5. Ako nije zadovoljen uvijet vrati se na 2.

# Uvjeti testiranja

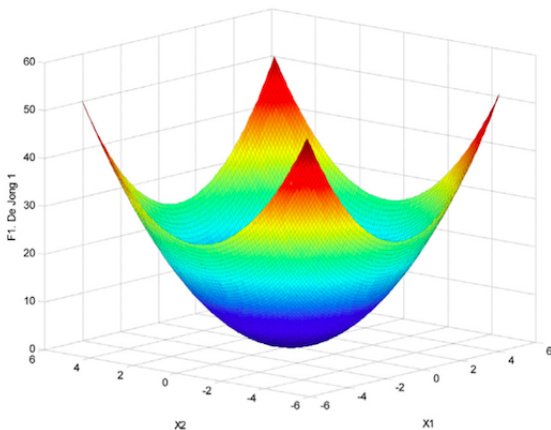
- 4 jezgri 64-bitni Intel procesor
- 8Gb RAM
- Linux operacijski sustav (2.6 kernel)
- Programski jezik C (kod prilagođen za 64-bitni sustav)
- Mersenne Twister generator pseudo-slučajnih brojeva (procesorsko vrijeme se koristi kao 'sjeme' za genereiranje)
- Apache2 server
- web tehnologije: Php i JavaScript

# Parametri pri testiranju

- Broj čestica  $n=50, 100, 150, 200$  za svaki algoritam
- ABC:  $m = \lfloor \frac{n}{2} \rfloor, e = \lfloor \frac{m}{2} \rfloor, ne = 10, nm = 5, \epsilon$  varijabilni
- PSO:  $\omega = 0.9, \rho_p = 1, \rho_g = 0.9$
- OPSO:  $\omega = 0.9, \rho_p = 1, \rho_g = 0.9\epsilon = 1$

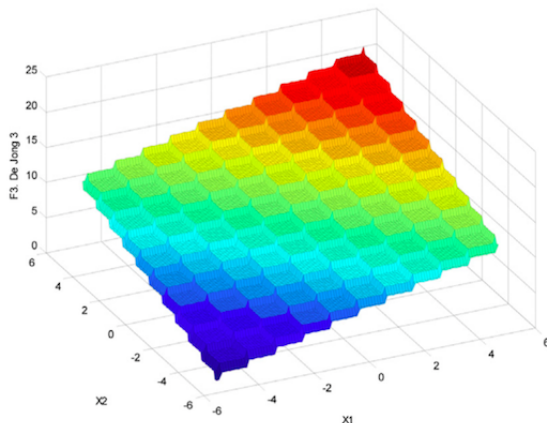
# Primjeri testnih funkcija - Sfera (De Jong 1)

$$f(x) = \sum_{i=1}^3 x_i^2, x \in \mathbb{R}^3, x_i \in [-5.12, 5.12]$$



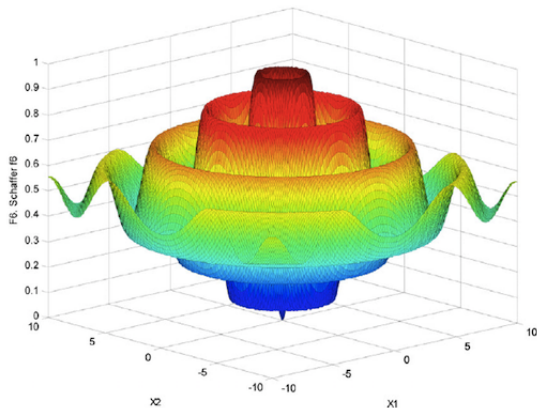
# Primjeri testnih funkcija - Step (De Jong 3)

$$f(x) = \sum_{i=1}^5 [x_i], x \in \mathbb{R}^5, x_i \in [-5.12, 5.12]$$



# Primjeri testnih funkcija - Schaffer F6

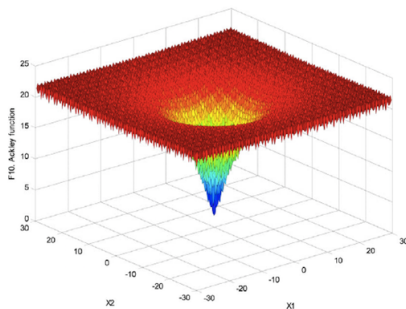
$$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{[1 + 0.001 \cdot (x_1^2 + x_2^2)]^2}, x \in \mathbb{R}^2, (x_i) \in [-100, 100]$$



# Primjeri testnih funkcija - Ackley

$$f(x) = -20 \cdot \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i)\right) + 20 + e$$

$$x \in \mathbb{R}^{30}, x_i \in [-30, 30]$$

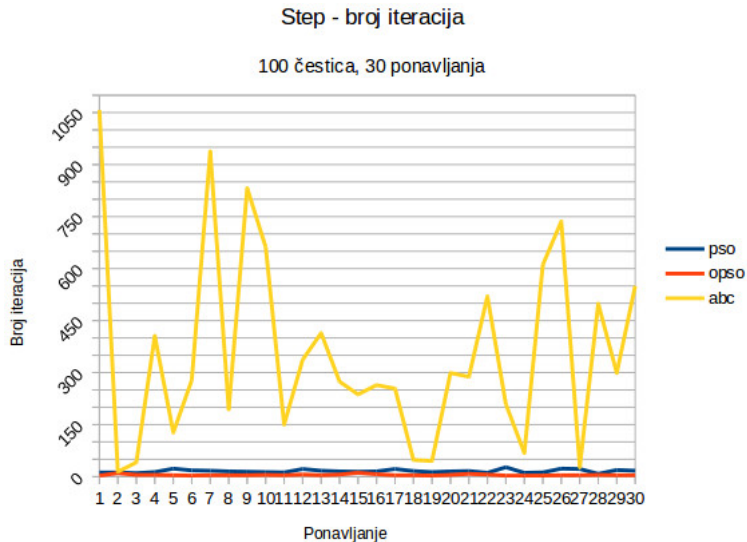


# Tablični prikaz broja iteracija - izbor

	Sfera	Step	Schaffer F6	Ackley
ABC	38.03	475.53	2	N/A
	224.60	355.93	1.93	N/A
	447.23	273.33	2.70	N/A
	630	323.20	2.57	N/A
PSO	50.73	350.33	234.30	10 000
	51.67	15.80	130.03	10 000
	52	16.63	135.07	10 000
	53.5	14.10	118.30	10 000
OPSO	28.63	6.27	76.83	8678.27
	25.73	4.83	51.97	4378.37
	23.67	4.87	36.13	2714.97
	25.73	4.87	38.4	1393.00



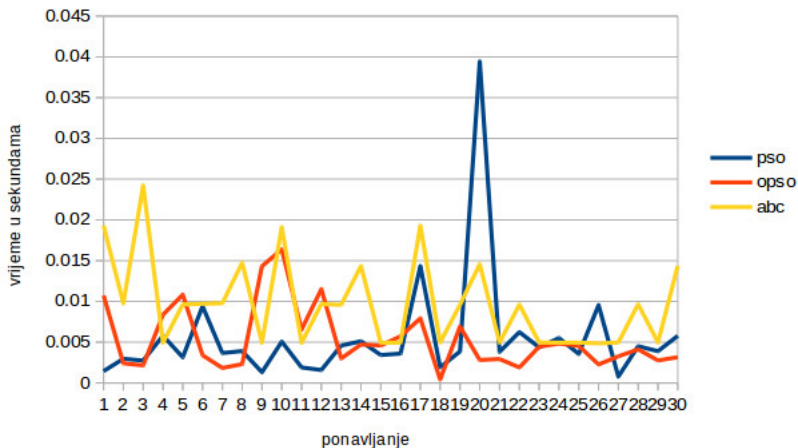
# Grafički prikaz prikaz - 1



# Grafički prikaz prikaz - 2

## Schaffer F6 - vrijeme izračuna

50 čestica, 30 ponavljanja








# Tehnologije

- Php i Javascript skriptni jezici
- Php funkcija `exec()` za poziv programa sa danim parametrima
- JavaScript funkcije za iscrtavanje grafova dobivenih rezultata
- Dva glavna dijela:
  - odabir funkcije i varijabli
  - grafiki prikaz rezultata

# Upotreba

- Korisnik unese parametre u za to predviđen prostor, uzimajući u obzir granice koje program prihvaća
- Klikom na gumb *Pošalji* poziva se `exec()` koja se nalazi u drugom php modulu
- Korisnik čeka da se završe izračuni nakon čega se fokusira dio stranice sa grafovima koji prikazuju dobivene rezultate

# Literatura

-  R. RAHMANI, R. YUSOF, *A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: Radial Movement Optimization*, Applied Mathematics and Computation 248 (2014) 287–300
-  F. AGHAZADEH, M. R. MEYBODI, *Learning Bees Algorithm For optimization*, International Conference on Information and Intelligent Computing IPCSIT vol.18 (2011) 115–122
-  S. LUKE, *Essentials of Metaheuristics*, Lulu, second edition, <http://cs.gmu.edu/~sean/book/metaheuristics/>
-  Z. MICHALEWICZ, D. B. FOGEL, *How to Solve It: Modern Heuristics*, Springer-Verlag Berlin Heidelberg (2000)
-  [en.wikipedia.org/wiki/Particle\\_swarm\\_optimization](http://en.wikipedia.org/wiki/Particle_swarm_optimization), listopad 2014.



W. ZHU, J. CURRY, *Parallel Ant Colony for Nonlinear Function Optimization with Graphics Hardware Acceleration*, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (2009) 1803–1808



B. ALATAS, *Chaotic bee colony algorithms for global numerical optimization*, Expert Systems with Applications 37 (2010) 5682–5687



F. AGHAZADEH, M. R. MEYBODI, *Learning Bees Algorithm For optimization*, International Conference on Information and Intelligent Computing IPCSIT vol.18 (2011) 115–122



K. M. MALAN, A. P. ENGELBRECHT, *Characterising the searchability of continuous optimisation problems for PSO*, accepted for publication, Swarm Intelligence, SpringerLink vol.8 (2014)