

Principi optimizacije algoritmom roja čestica (PSO - Particle Swarm Optimization)

MILAN R. RAPAIĆ

Katedra za automatsko upravljanje
Departman za računarstvo i automatiku
Fakultet tehničkih nauka
Univerzitet u Novom Sadu

$$f(x) \rightarrow \max$$

- f - kriterijum optimalnosti - kriterijum prilagođenosti, **prilagođenost**
 - Sama potencijalna rešenja nazivaju se **česticama**
 - Ako imamo dve čestice, x_1 i x_2 , i ako je $f(x_1) > f(x_2)$ reći ćemo da je x_1 **bolja** čestica od x_2 (**prilagođenija**)
 - Skup čestica nazivamo **rojem**
 - Paralelno se koriste termini iteracija i generacija
 - Svaka čestica ima sposobnost memorisanja prethodnih događaja (vrlo ograničenu) i sposobnost komunikacije sa drugim česticama (vrlo ograničenu). U najčešćem slučaju, svaka čestica (i -ta čestica) je sposobna da pamti najbolju poziciju u kojoj se našla tokom čitavog svog života – p_i . U najčešćem slučaju, sve čestice su sposobne da prenesu informaciju o toj svojoj najboljoj poziciji drugima, i to tako da je svaka čestica svesna najbolje pozicije u kojoj se ikada našla bilo koja čestica u roju (najbolja pozicija na nivou roja) – g .
1. Izaberemo početne pozicije svih čestica u roju: odaberemo veličinu roja (broj čestica u roju, N) i na neki slučajan način “razbacamo” te čestice po prostoru pretrage. Označimo roj sa $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$
 2. Za svaku jedinku iz tekuće populacije, sračunamo njenju prilagođenost: $f(x)$. Ako je tekuća prilagođenost bolja od najbolje lične prilagođenosti, $f(x_i) > f(p_i)$, onda ažuriramo p_i tako da je $p_i = x_i$. Prosto ono p_i čije je $f(p_i)$ najveće je g . Prosto: u svakoj iteraciji g je najbolje p_i .
 3. Ažuriramo položaj svake čestice (videti sledeće slajdove!). Tako što zadržavamo procenat prethodne brzine kretanja (isti trik koji se koristi kod gradijentnog algoritma sa momentom), i na tu brzinu dodajemo komponente brzine koje pokazuju jedna ka sopstvenoj najboljoj poziciji p_i , a druga ka globalnoj najboljoj poziciji g : ove tri komponente se nazivaju inerciona, kognitivna i socijalna, tim redom.
 4. Ponavljamo korake 2 i 3 unapred definisani broj iteracija. Nakon toga, rešenje je g .

x_j^k - pozicija j -te čestice u k -toj iteraciji. Međutim, pošto kompletno pravilo ažuriranja važi za svaku česticu, donji indeks ćemo ispuštati, prosto ćemo pisati x^k .

v^k - “brzina” čestice u k -toj iteraciji. Ova veličina predstavlja skok, odnosno promenu pozicije.

$$x^{k+1} = x^k + v^k$$

Kako se računa brzina?

$$v^k = \underbrace{w v^{k-1}}_{\text{inerciona komponenta}} + \underbrace{c_p r_p (p^k - x^k)}_{\text{kognitivna komponenta}} + \underbrace{c_g r_g (g^k - x^k)}_{\text{socijalna komponenta}}$$

$0 < w < 1$ - **faktor inercije** ... obično se bira tako da je na početku 0,9, a kako odmiču iteracije lagano pada 0,4

$c_p > 0$ - **kognitivni faktor** ... obično se biraju vrednosti koje padaju tokom optimizacije od 2,5 do 0,5

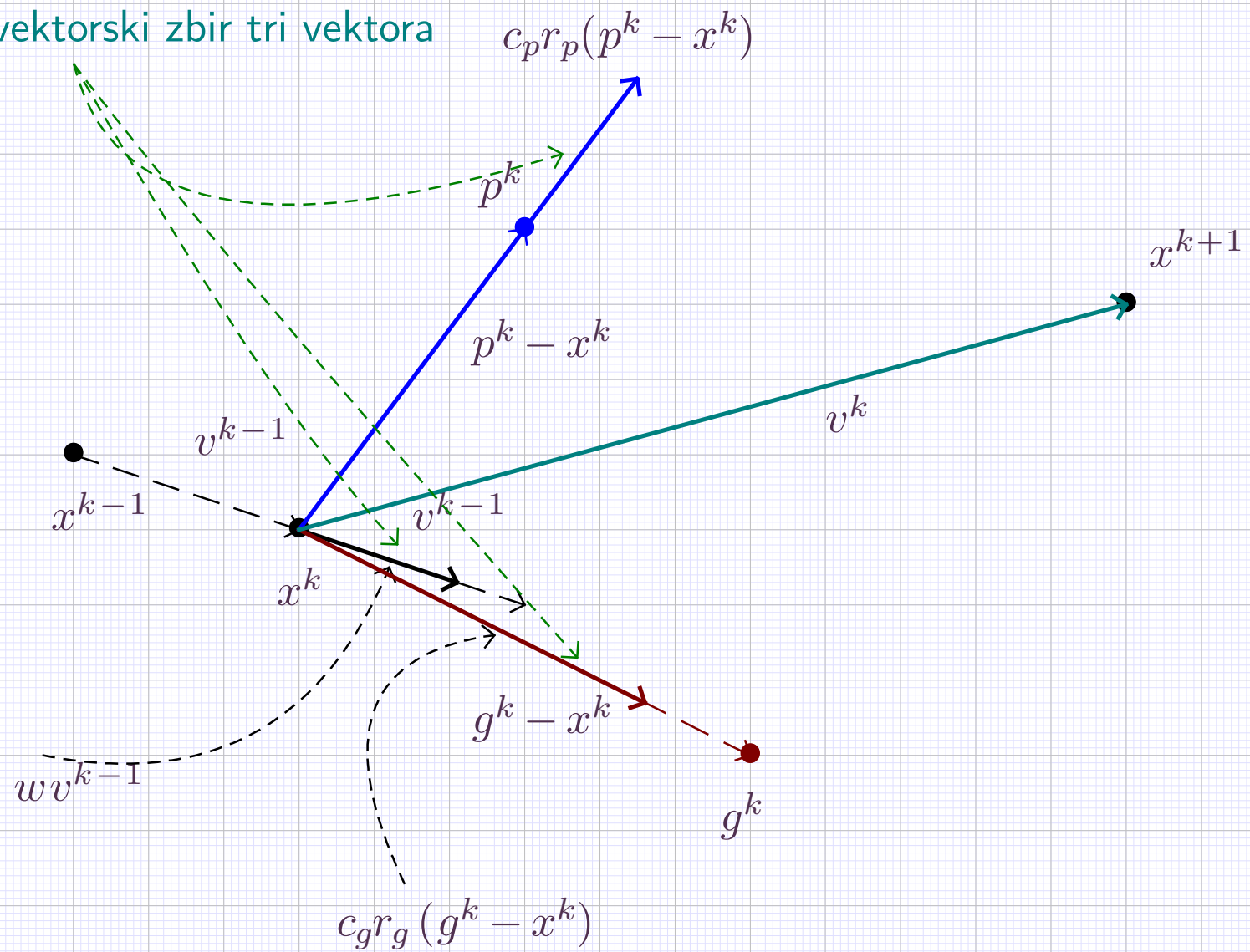
$c_g > 0$ - **socijalni faktor** ... obično se biraju vrednosti koje rastu tokom optimizacije od 0,5 do 2,5

Jednim imenom c_p i c_g se nazivaju **faktorima ubrzanja**.

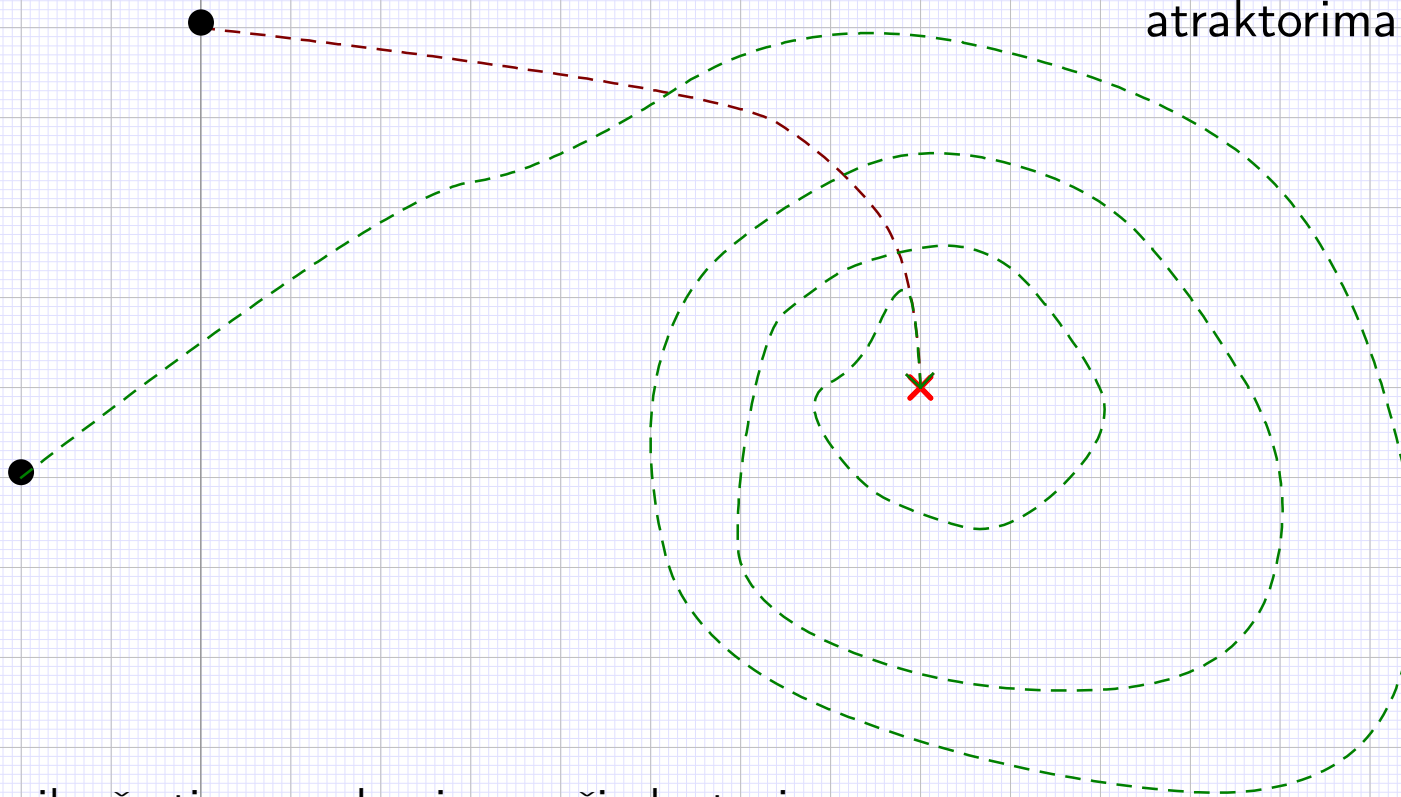
r_p, r_g - slučajni brojevi, uniformno raspodeljeni u opsegu $[0, 1]$.

Slučajni faktori r_p, r_g obezbeđuju da se u svakoj iteraciji, ponašanju čestica dodeli izvestan stepen individualnosti. Odnosno da logika smanjenja i povećanja parametara brzine važi, ali samo u srednjem, a ne obavezno za svaku pojedinačnu česticu u svakoj iteraciji.

Nova brzina je vektorski zbir tri vektora



Efikasno pretraživanje prostora
zahteva da se čestice kreću
oscilatorni kada prilaze svojim
atraktorima (p odnosno g).

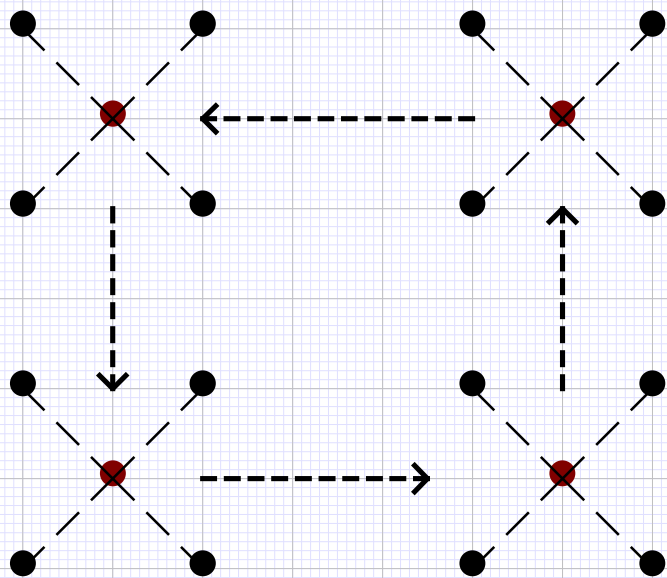
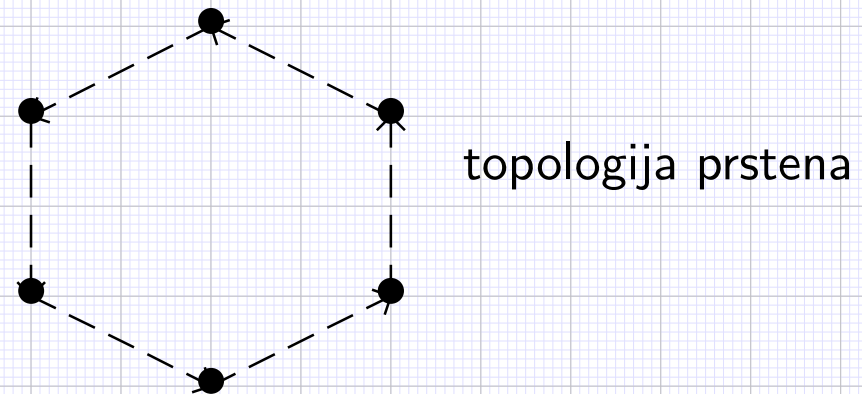
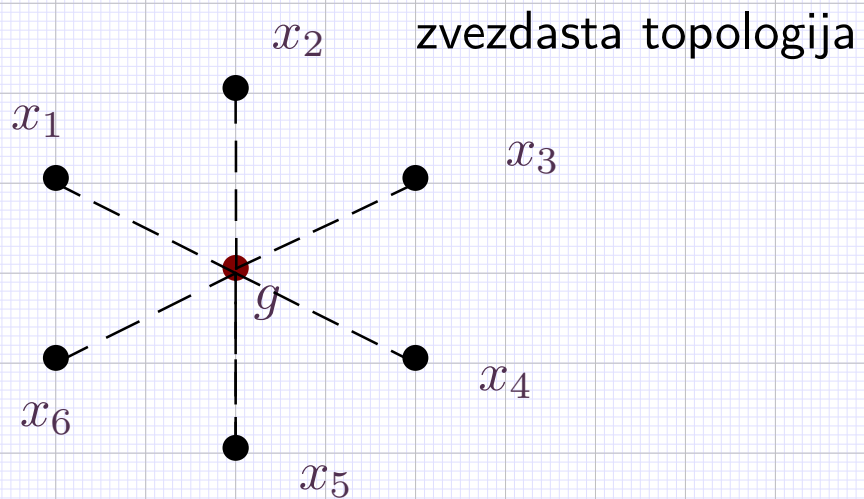


Dinamika čestica se odnosi na način kretanja
čestica u prostoru pretrage. Ukoliko fiksiramo c_p i c_g , te zamenimo r_p i r_g njihovim srednjim vrednostin
(koje su $1/2$), onda se jednačina kretanje svake pojedinačne čestice može
posmatrati kao linearan stacionaran vremenski diskretan sistem drugog reda.
Dinamika ovakvog sistema se može analizirati posmatranjem položaja polova.

Topologija roja se odnosi na način na koji čestice komuniciraju između sebe (razmenjuju svoje p vrednosti).

Do sada opisani PSO algoritam pripada grupi algoritama sa **zvezdastom topologijom** (eng. *star topology*). Ovo je topologija sa najbržom razmenom informacija. Postoje i druge topologije, među kojima su najvažnije: topologija prstena (jednostrukog i dvostrukog) i složena (hijerarhijska) topologija (u kojoj je roj podeljen na više podrojeva, topologija šitavog super-roja i svakog od pojedinih pod-rojeva može biti različita).

Veća povezanost čestica u roju doprinosi bržoj konvergenciji. Sa druge strane, brža konvergencija dovodi do slabije otpornosti na “zaglavljivanje” u lokalnim minimumima. Pravilan izbor parametara i topologije mora balansirati između ove dve međusobno suprotstavljene težnje: robusnosti, sa jedne, i performanse, sa druge strane.



Složena topologija, sa više podrojeva, od kojih je svaki organizovan zvezdasto, a sami podrojevi su između sebe organizovani u prsten.