Рачунарска интелигенција

Оптимизација ројем честица - PSO

Александар Картељ

kartelj@matf.bg.ac.rs

Неки од материјала за ове слајдове су преузети до стране:
A/Prof. Xiaodong Li, School of Computer Science and IT, RMIT University Melbourne, Australia
.
Датум последње измене: 18.12.2019.

Оптимизација ројевима честица

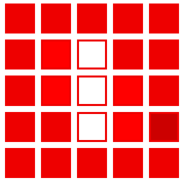
Ова метода (енг. Particle swarm optimization – PSO) има корене у социјалној психологији

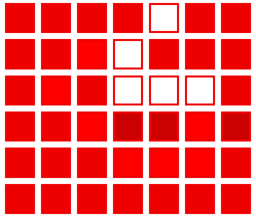
Ројеви честица су на неки начин слични целуларним аутоматима (енг. Celular automata - CA):

a) Свака појединачна ћелија ажурира своје стање паралелно са осталима

b) Свака нова вредност неке ћелије зависи од старих вредности и од вредности својих суседа

c) Све ћелије се ажурирају применом истог правила (Rucker, 1999).

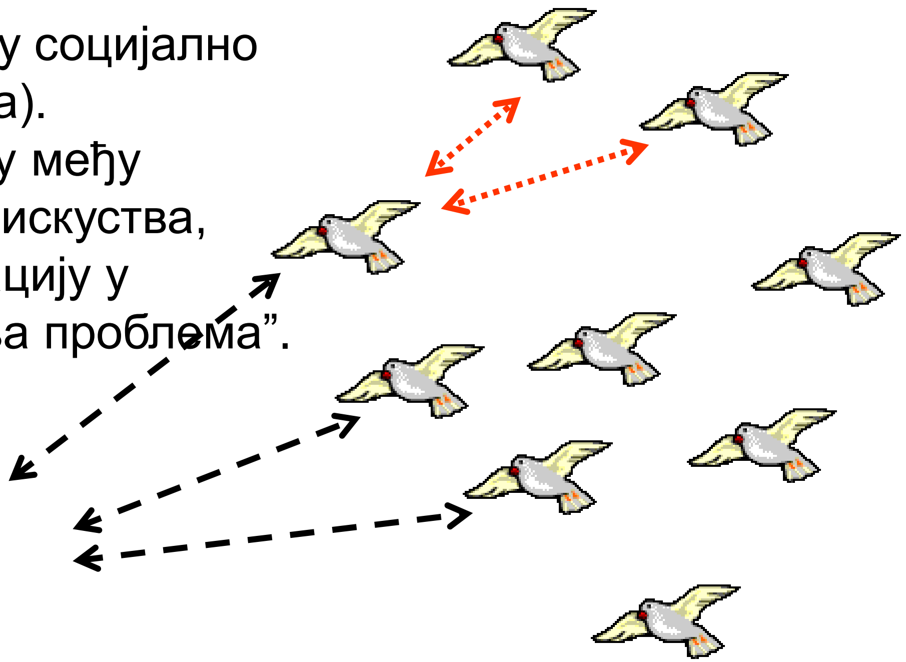

Blinker

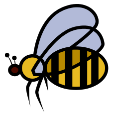

Glider

Честице унутар роја се могу поистоветити са ћелијама унутар CA, само се њихова стања мењају у много димензија истовремено.

Оптимизација ројем честица (2)

Према објашњењу аутора методе James Kennedy и Russell Eberhart:
“честице унутар роја имитирају социјално понашање људи (или инсеката).
Честице (јединке) интерреагују међу собоом док уче из сопственог искуства, што постепено помера популацију у правцу бољих региона решења проблема”.





Зашто име “честица”, а не “тачке”?
Обојица аутора су били мишљења да брзине и убрзања више приличе честицама него тачкама.

PSO примене

- Проблеми у:
  - реалном,
  - дискретном
  - или мешовитом простору претраге.
- Даље, проблеми са:
  - вишеструким локалним оптимумима,
  - ограничењима,.
- Проблеми:
  - вишециљне,
  - динамичке оптимизације,
  - ...

Оригинални PSO

$$\vec{v}_i \leftarrow \vec{v}_i + \vec{\varphi}_1 \otimes (\vec{p}_i - \vec{x}_i) + \vec{\varphi}_2 \otimes (\vec{p}_g - \vec{x}_i)$$

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$$

$\vec{x}_i$ је тренутна позиција *i*–те честице у роју;

$\vec{v}_i$ означава брзину *i*-те честице;

$\vec{p}_i$ је најбоље пронађена позиција коју је пронашла *i*-та честица до сада, тј., лични најбољи;

$\vec{p}_g$ најбоља позиција у суседству тј., глобални најбољи;

Симбол $\otimes$ означава векторски производ;

$\vec{\varphi}_1 = c_1\vec{r}_1$ and $\vec{\varphi}_2 = c_2\vec{r}_2$;

$\vec{r}_1$ и $\vec{r}_2$ су два вектора случајних бројева из равномерне расподеле [0, 1];

c_1 и c_2 су коефицијенти убрзања.

Оригинални PSO

Когнитивна компонента

Социјална компонента

$$\vec{v}_i \leftarrow \vec{v}_i + \boxed{\vec{\varphi}_1 \otimes (\vec{p}_i - \vec{x}_i)} + \boxed{\vec{\varphi}_2 \otimes (\vec{p}_g - \vec{x}_i)}$$

Моменат

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$$

Брзина $\vec{v}_i$ (удео промена) *i*-те честице је дефинисан са 3 компоненте:

- **моменат** – претходно стање брзине се преноси на ново;
- **когнитивна** компонента – тенденција враћања у лично најбоље;
- **социјална** компонента – тенденција ка кретању ка најбољем глобалном.

Различите топологије се могу користити за усмеравање тока информација између честица, нпр. топологија прстена, звезде, фон Нојманова топологија.

Псеудокод обичног PSO

```
random_initial_population();
repeat
    for i = 1 to population_size do
        if f(x_i) < f(p_i) then p_i = x_i;
        p_g = min(p_neighbours);
        for d =1 to dimensions do
            velocity_update();
            position_update();
        end
    end
until termination_criterion_met()
```

Инерцијална тежина

Вредности $\vec{p}_i$ и $\vec{p}_g$ се могу пребацити у једну вредност $\vec{p}$ без губитка информација:

$$\vec{v}_i \leftarrow \vec{v}_i + \vec{\varphi} \otimes (\vec{p} - \vec{x}_i)$$

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$$

Где је $\vec{\varphi} = \vec{\varphi}_1 + \vec{\varphi}_2$ и $\vec{p} = \dfrac{\vec{\varphi}_1 \otimes \vec{p}_i + \vec{\varphi}_2 \otimes \vec{p}_g}{\vec{\varphi}_1 + \vec{\varphi}_2}$

$\vec{p}$ представља пондерисани просек $\vec{p}_i$ и $\vec{p}_g$. Приметити да је оператор дељења по елементима.

За контролисање утицаја претходне брзине на нову брзину,
може се користити додатни параметар који се зове инерција (инерцијална тежина) ***w***:

$$\vec{v}_i \leftarrow w\,\vec{v}_i + \vec{\varphi}_1 \otimes (\vec{p}_i - \vec{x}_i) + \vec{\varphi}_2 \otimes (\vec{p}_g - \vec{x}_i)$$

Инерцијална тежина (2)

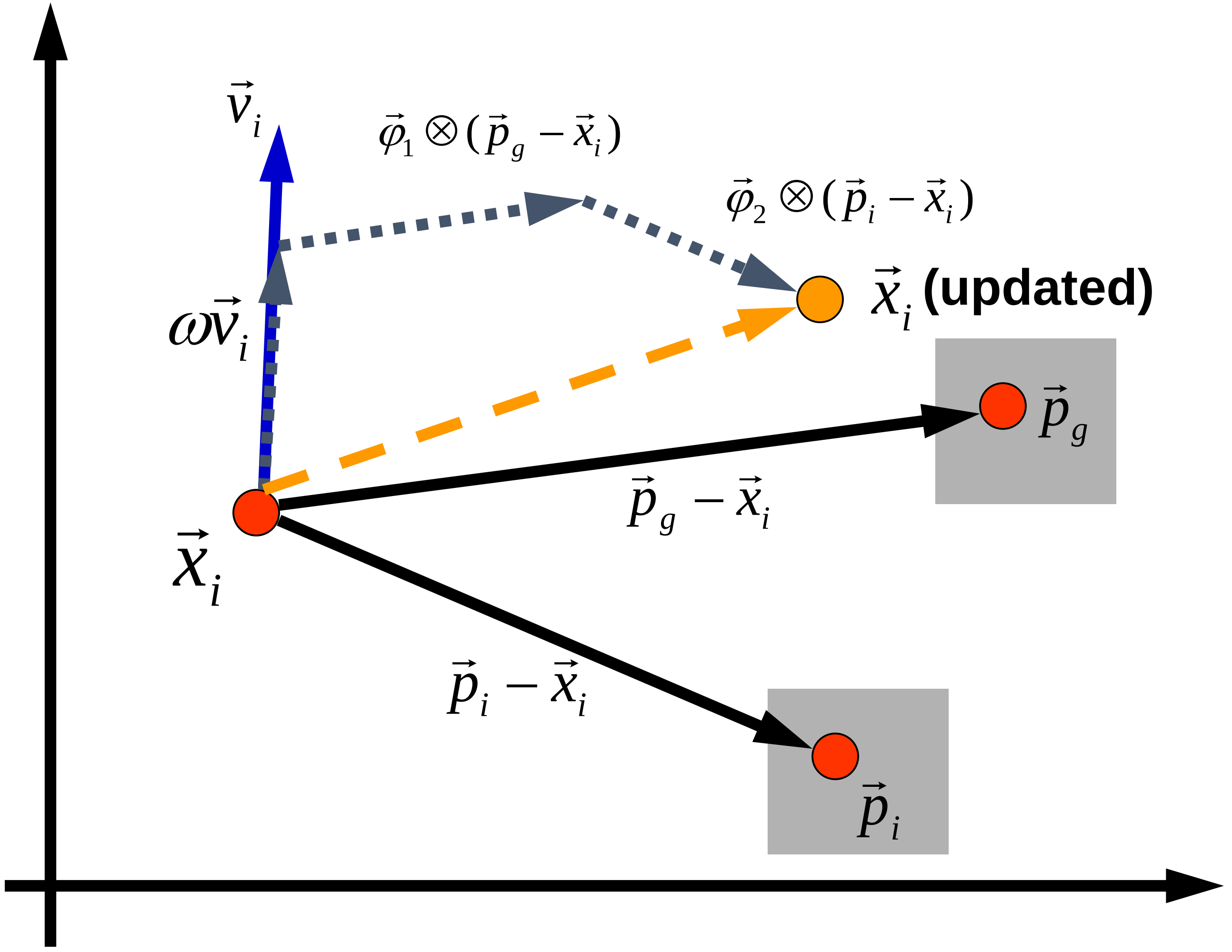
Инерција омогућава контролу експлорације и експлоатације

За $w \geq 1$: брзина расте током времена, па рој дивергира;
За $0 < w < 1$: честице успоравају па конвергенција зависи од вредности $\mathbf{c_1}$ и $\mathbf{c_2}$;
За $w < 0$: брзина се смањује током времена, на крају достиже 0 и тиме се зауставља алгоритам.

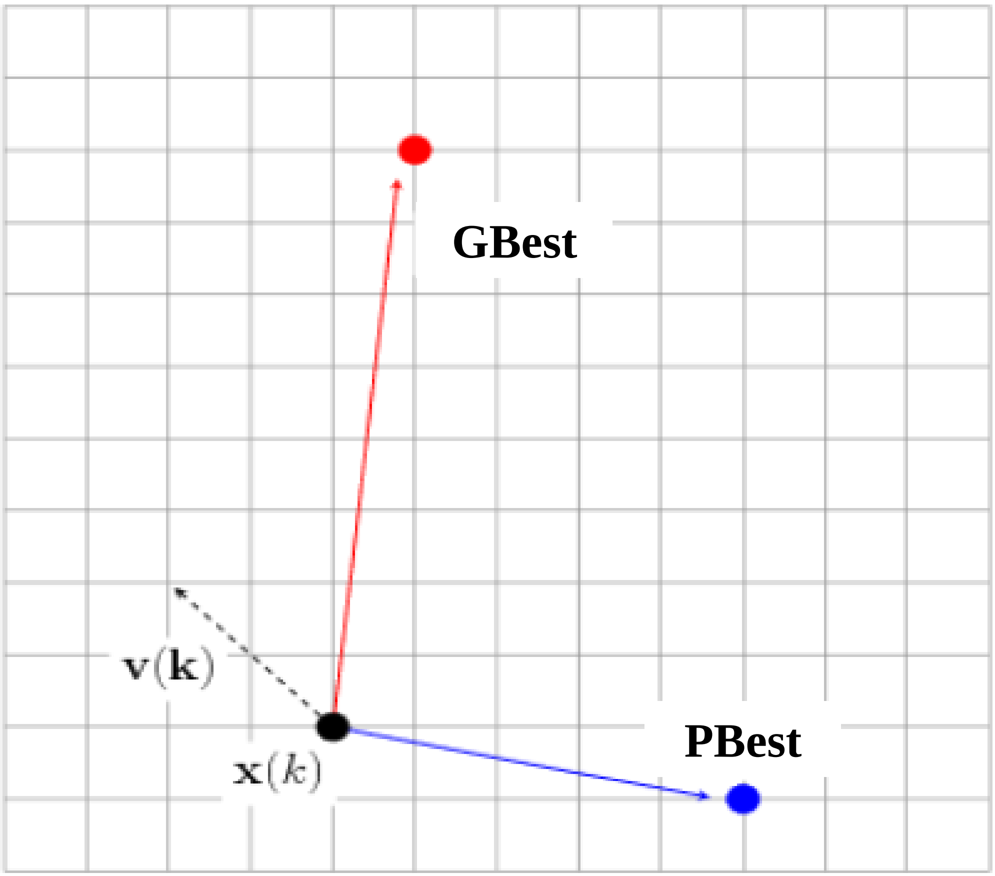
Емпиријски резултати указују да константна инерцијална тежина од w = 0.7298 и $\mathbf{c_1}=\mathbf{c_2}=1.49618$ дају добре резултате.

Eberhart и Shi такође саветују употребу инерције која се смањује током времена, обично у распону између 0.9 и 0.4. Ово утиче на смањење простора претраге током времена и постепено пребацивање из режима јаче експлорације ка режиму јаче експлоатације.

Визуализација PSO



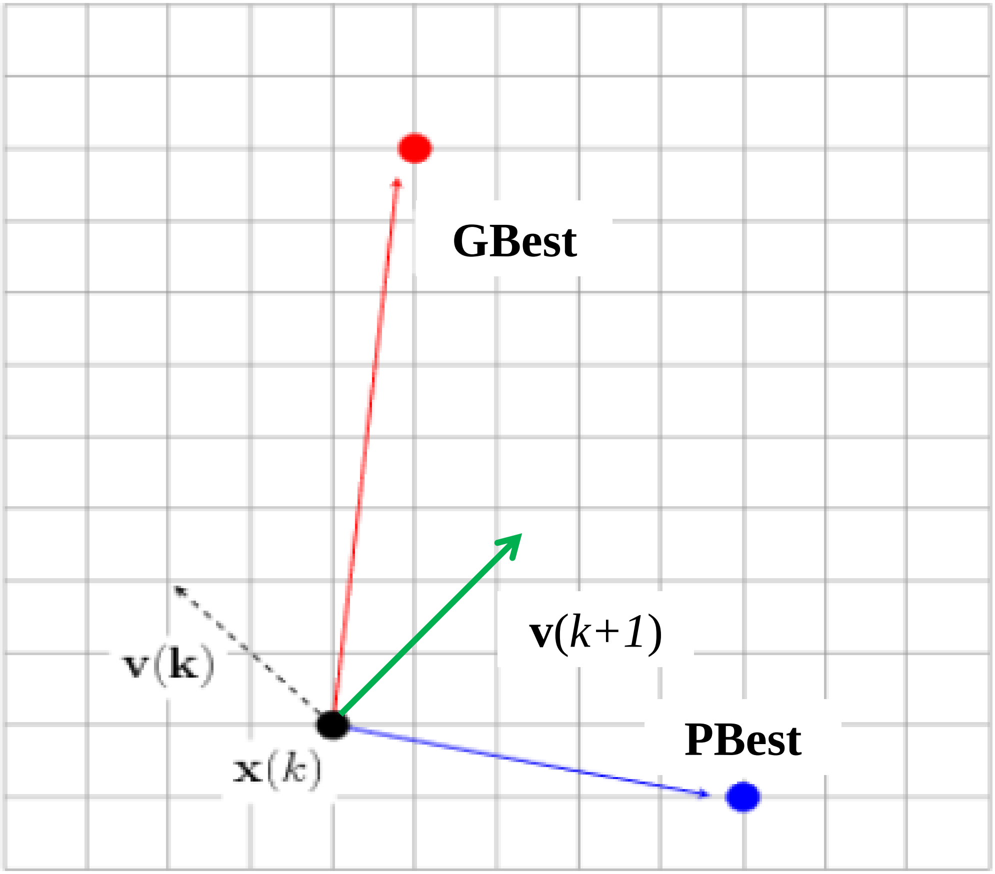
PSO нумерички пример



- Инерција: **v**(*k*)=(-2,2)
- Когнитивно: PBest-**x**(*k*)=(9,1)-(4,2)=(5,-1)
- Социјално: GBest-**x**(*k*)=(5,10)-(4,2)=(1,8)

- **x**(*k*) - Тренутно решење (4, 2)
- PBest – Лични најбољи (9, 1)
- Gbest - Глобални најбољи (5, 10)

PSO нумерички пример (2)

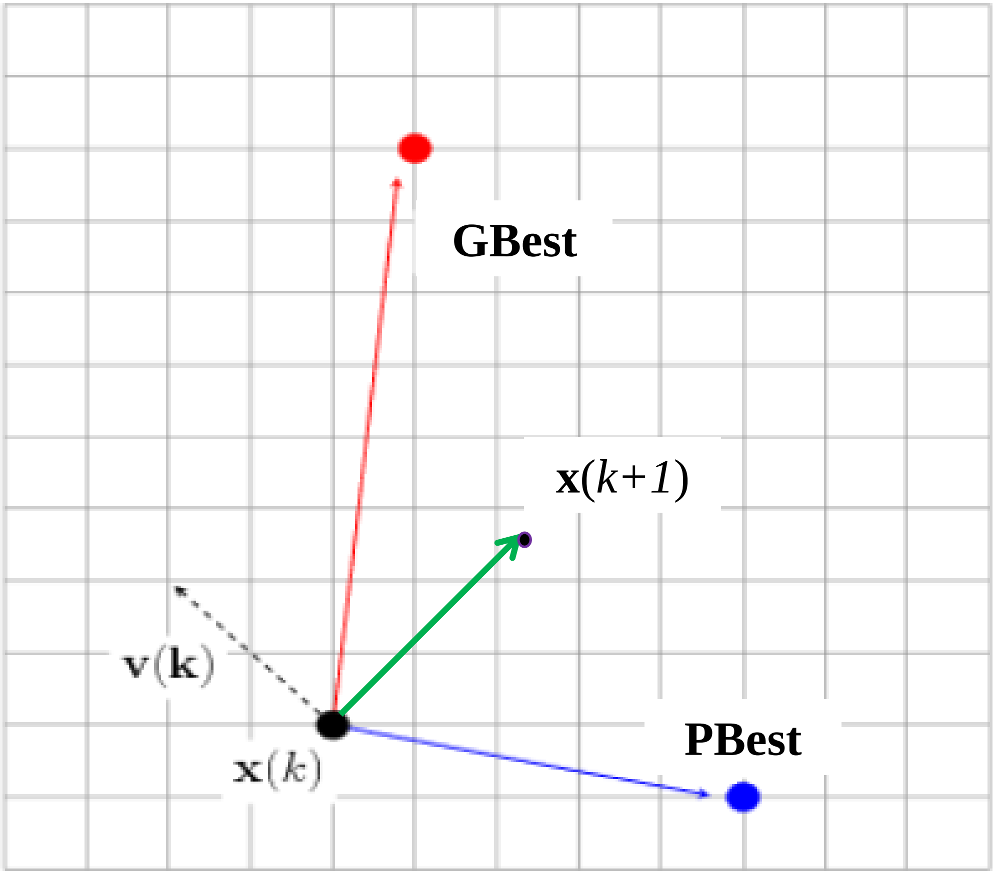


- Инерција: $\mathbf{v}(k)$=(-2,2)
- Когнитивно:
  PBest-$\mathbf{x}(k)$=(9,1)-(4,2)=(5,-1)
- Социјално:
  GBest-$\mathbf{x}(k)$=(5,10)-(4,2)=(1,8)

$\mathbf{v}(k+1)$=(-2,2)+0.8*(5,-1)
+0.2*(1,8) = (2.2,2.8)

- $\mathbf{x}(k)$ – Тренутно решење (4, 2)
- PBest – Лични најбољи (9, 1)
- GBest- Глобални најбољи (5, 10)

PSO нумерички пример (3)



- Инерција: $\mathbf{v}(k)$=(-2,2)
- Когнитивно:
  PBest-$\mathbf{x}(k)$=(9,1)-(4,2)=(5,-1)
- Социјално:
  GBest-$\mathbf{x}(k)$=(5,10)-(4,2)=(1,8)
- $\mathbf{v}(k+1)$=(2.2,2.8)

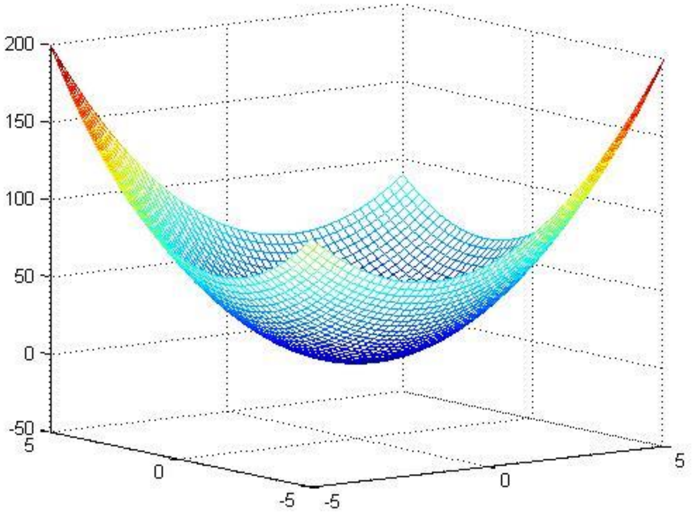
$\mathbf{x}(k+1)=\mathbf{x}(k)+\mathbf{v}(k+1)=$
(4,2)+(2.2,2.8)=(6.2,4.8)

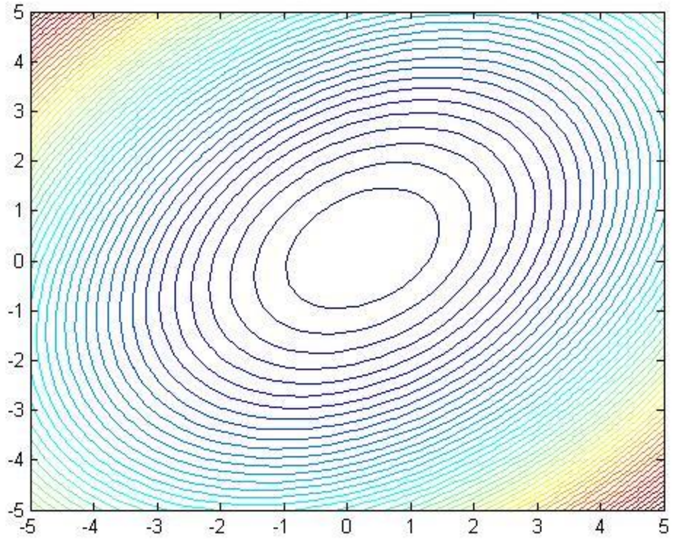
- $\mathbf{x}(k)$ – Тренутно решење (4, 2)
- PBest – Лични најбољи (9, 1)
- GBest- Глобални најбољи (5, 10)

Потпунији пример

- Пронаћи минимум функције

$$f(\mathbf{x}) = 3x_1^2 - 2x_1x_2 + 3x_2^2 - x_1 - x_2$$





Потпунији пример (2)

$$\mathbf{x}_1 = \begin{bmatrix} 2.2824 & 0.6238 & 4.0005 & 3.1717 & -4.0058 \\ -0.4894 & -2.7580 & -2.7043 & -3.3118 & 1.5771 \end{bmatrix}$$

$$\mathbf{v}_1 = \begin{bmatrix} -0.6321 & 0.1712 & 0.6942 & 0.0264 & 0.2207 \\ 0.2133 & -0.5598 & -0.2500 & 0.6079 & 0.3122 \end{bmatrix}$$

$$\Downarrow$$

$$\mathbf{x}_2 = \begin{bmatrix} 1.7767 & 1.4300 & 2.5656 & 2.2018 & 3.3541 \\ -0.3187 & -2.2903 & -0.3385 & 0.3199 & -0.5338 \end{bmatrix}$$

$$\mathbf{v}_2 = \begin{bmatrix} -0.5057 & 0.8063 & -1.4349 & -0.9700 & 7.3599 \\ 0.1706 & 0.4677 & 2.3657 & 3.6317 & -2.1109 \end{bmatrix}$$

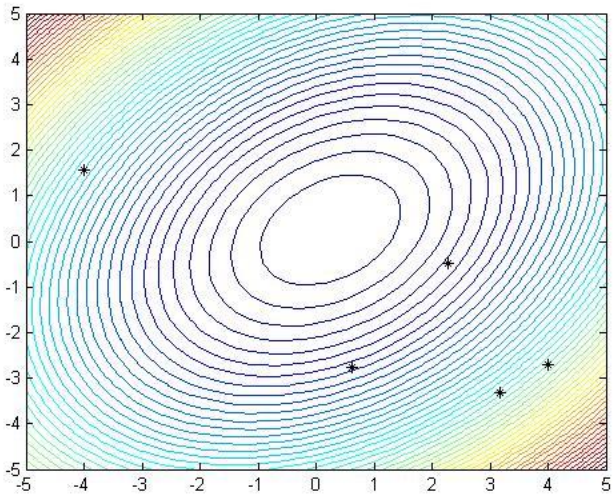
$$\Downarrow$$

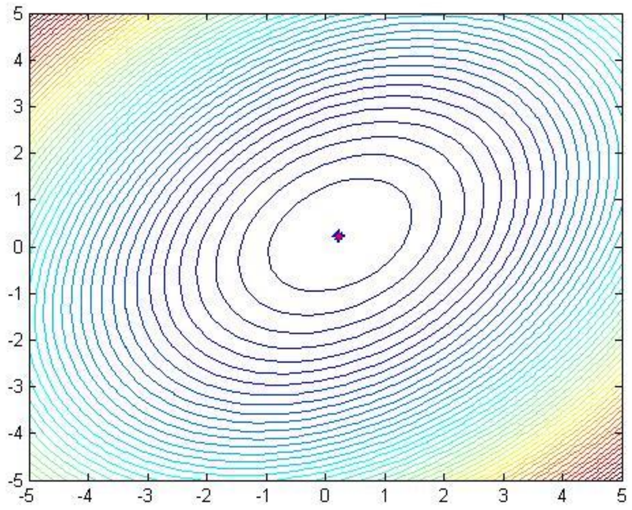
$$\mathbf{x}_3 = \begin{bmatrix} 1.3721 & 2.4464 & 1.0728 & 1.1350 & 7.9656 \\ -0.1822 & 0.1959 & 1.5627 & 2.7884 & -2.0485 \end{bmatrix}$$

$$\mathbf{v}_3 = \begin{bmatrix} -0.4046 & 1.0163 & -1.4928 & -1.0667 & 4.6114 \\ 0.1365 & 2.4862 & 1.9012 & 2.4685 & -1.5146 \end{bmatrix}$$

$$\vdots$$

$$\mathbf{x}_t = \begin{bmatrix} 0.2230 & 0.2197 & 0.2400 & 0.2293 & 0.2167 \\ 0.2056 & 0.2436 & 0.2378 & 0.2156 & 0.2106 \end{bmatrix}$$





$$GBest = \begin{bmatrix} 0.2227 \\ 0.2057 \end{bmatrix} \quad fitness = -0.25$$

Путања честице

Питање: колико су битне интеракције између честица унутар PSO?

Да би одговорили на ово питање, посматрајмо једноставни PSO, и случај када је рој сведен на само 2 честице. Овако поједностављени PSO претпоставља да:

- Нема стохастичке компоненте;
- Постоји једна димензија;
- Унапред одређене почетне позиције и брзине.

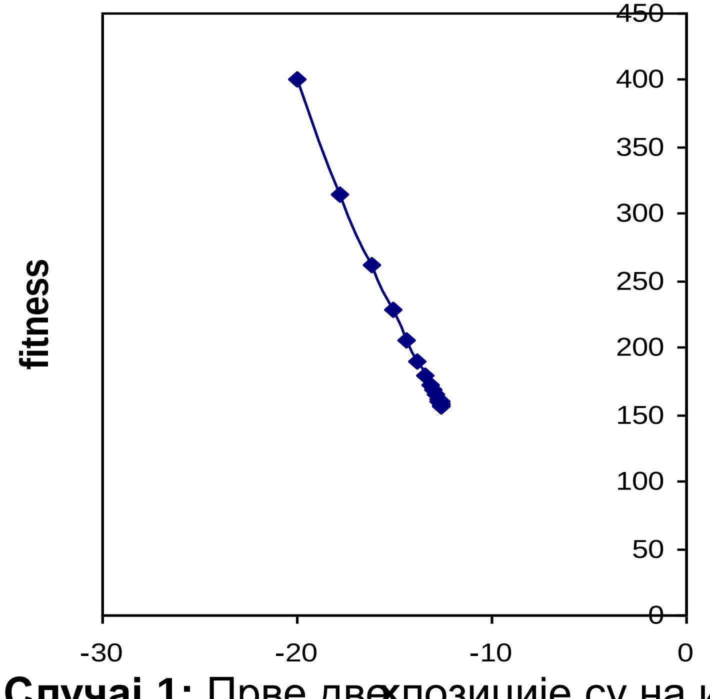
$$v \leftarrow w\, v + c_1(p_i - x) + c_2(p_g - x)$$
$$x \leftarrow x + v$$

У наредним примерима, претпостављамо да је ***w***=0.7, $c_1=c_2=0.7$.
Чак и у случају само једне честице, знамо позиције **x** и $\mathbf{p_i}$.

Нека је функција коју посматрамо $f(x)=x^2$, дефинисана на [-20, 20].
Имамо два случаја:

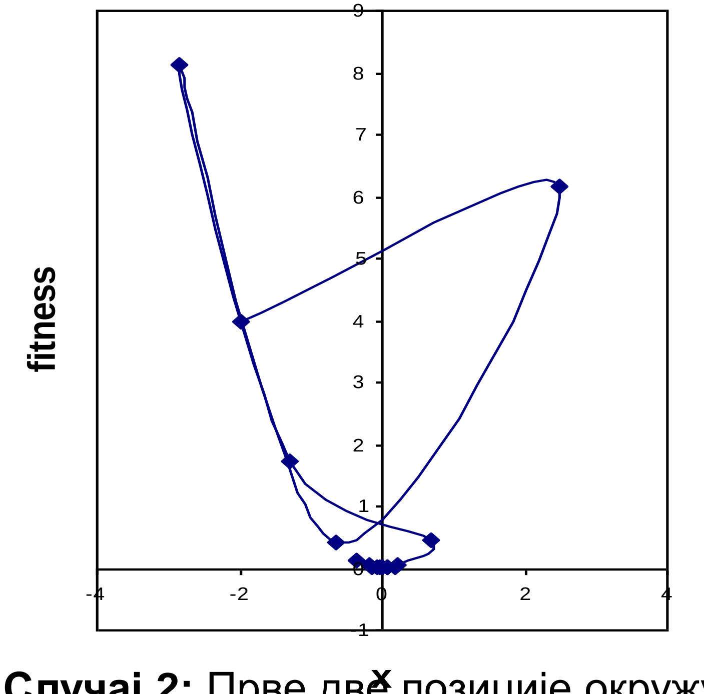
1) Прве две позиције су на истој страни минимума (иницијална позиција x= -20, v=3.2)
2) Прве две позиције окружују минимум (иницијална позиција x=-2, v=6.4).

Путања једне честице



Случај 1: Прве две позиције су на истој страни минимума.

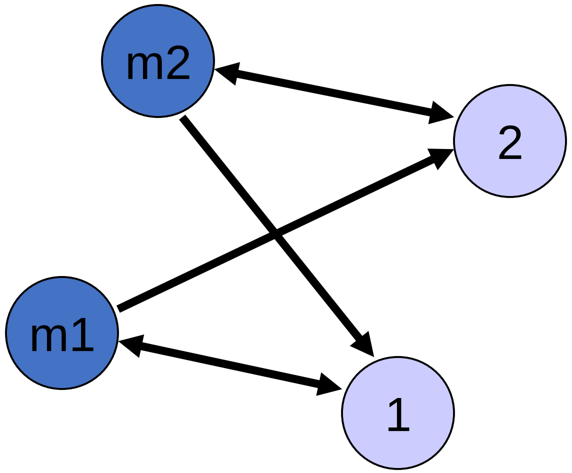
Пошто је лични најбољи увек једнак x, честица никад неће моћи да достигне минимум (прерана конвергенција).



Случај 2: Прве две позиције окружују минимум.

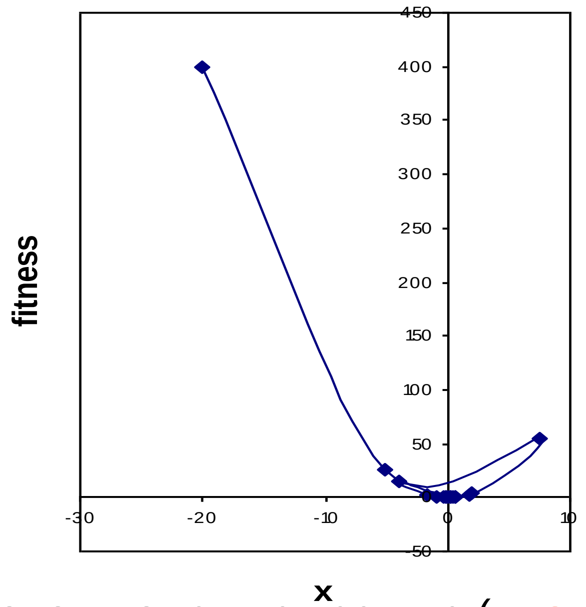
Честица осцилира око минимума пошто лични најбољи није увек x, што резултује бољим понашањем.

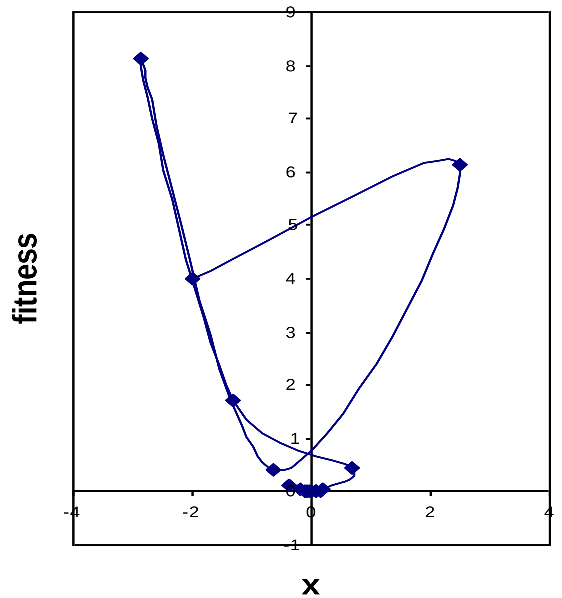
Путања две честице



Граф утицаја. У овом случају, имамо два истраживача и две меморије. Сваки истраживач прима информације од обе меморије, али информише само једну меморију (Clerc, 2006).

Путања две честице (2)

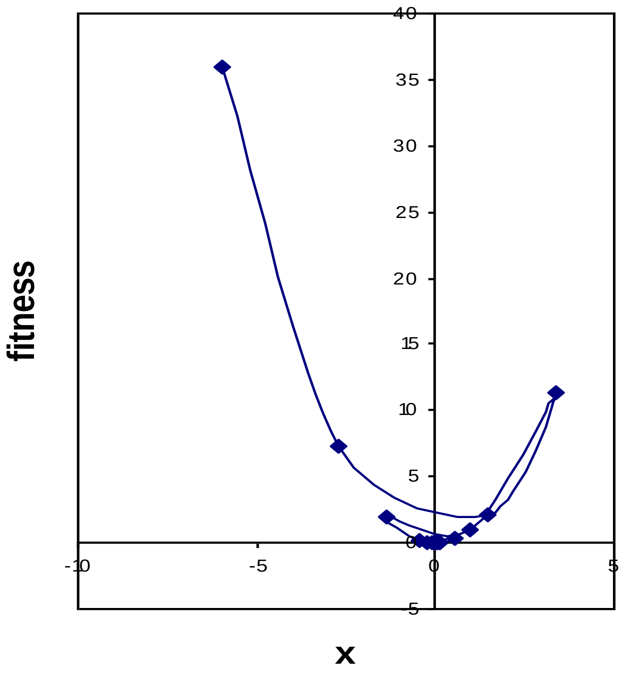


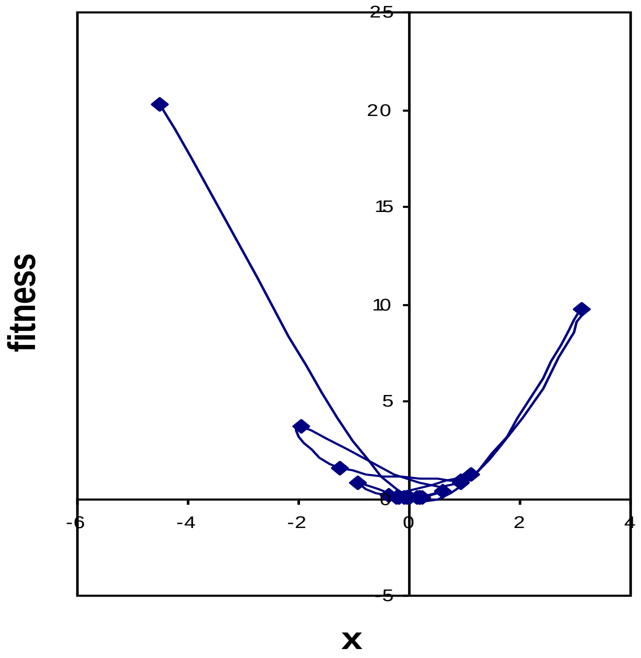


Сада имамо две честице (**два истраживача и две меморије**). Почетне позиције честица су исте као у случајевима 1 и 2 раније. Међутим, честице сада раде заједно (Clerc, 2006).

Приметити да је меморија честице 2 увек боља од меморије 1, па се честица 2 понаша исто као и раније када је била сама. Међутим, честица 2 сада има користи од меморије честице 2 што на крају изазива конвергенцију (лева слика) .

Путања две честице (3)





Две честице и две меморије.
Општији случај је када је свака честица под утицајем туђе меморије само повремено.
Конвергенција ка глобалном оптимуму је тада вероватнија,
мада цео процес може бити спорији.

Потенцијално опасно својство

- Шта се дешава када

$$\vec{x}_i = \vec{p}_i = \vec{p}_g$$

- Тада ажурирање брзине зависи само од

$$w\vec{v}_i$$

- Ако се ова ситуација настави више итерација,

$$w\vec{v}_i \rightarrow 0$$

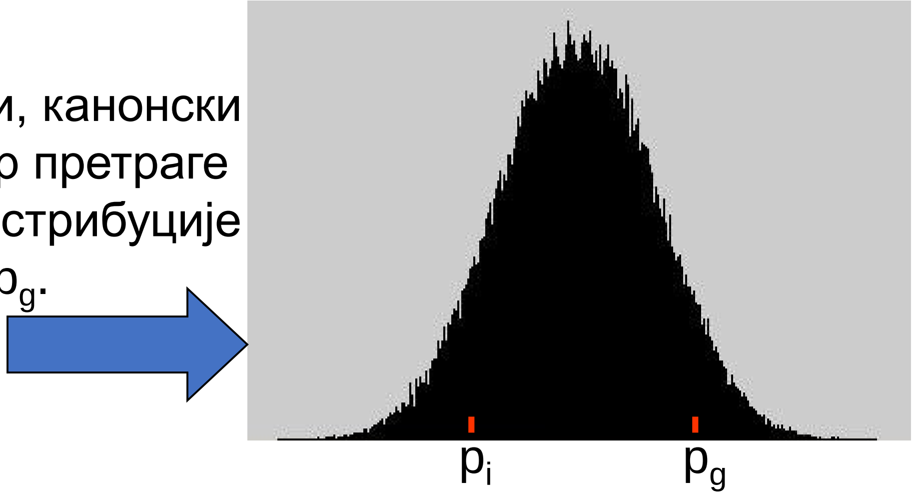
- **Решење**: Омогућити да глобално најбоља честица врши локалну претрагу и користи мутацију да прекине ово стање.

Огољени PSO

Шта ако избацимо брзину? Да ли је она неопходна?

Kennedy (2003) је вршио експерименте са варијантом PSO која не користи уопште брзине.

Ако су p_i и p_g константни, канонски PSO претражује простор претраге праћењем нормалне дистрибуције са центром између p_i и p_g.



Бинарни PSO

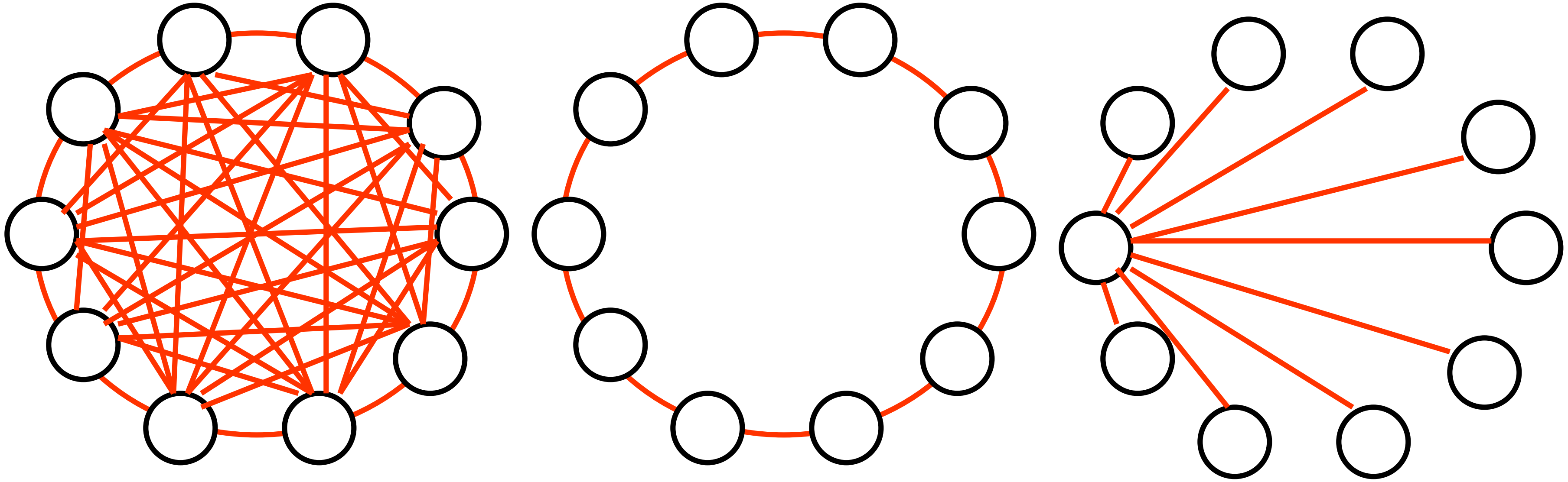
- Позиције се ажурирају према формули:

$$x_{ij}(t+1) = \begin{cases} 1 & if\ U(0,1) < sig(v_{ij}(t+1)) \\ 0 & otherwise \end{cases}$$

где је:

$$sig(v) = \frac{1}{1+e^{-v}}$$

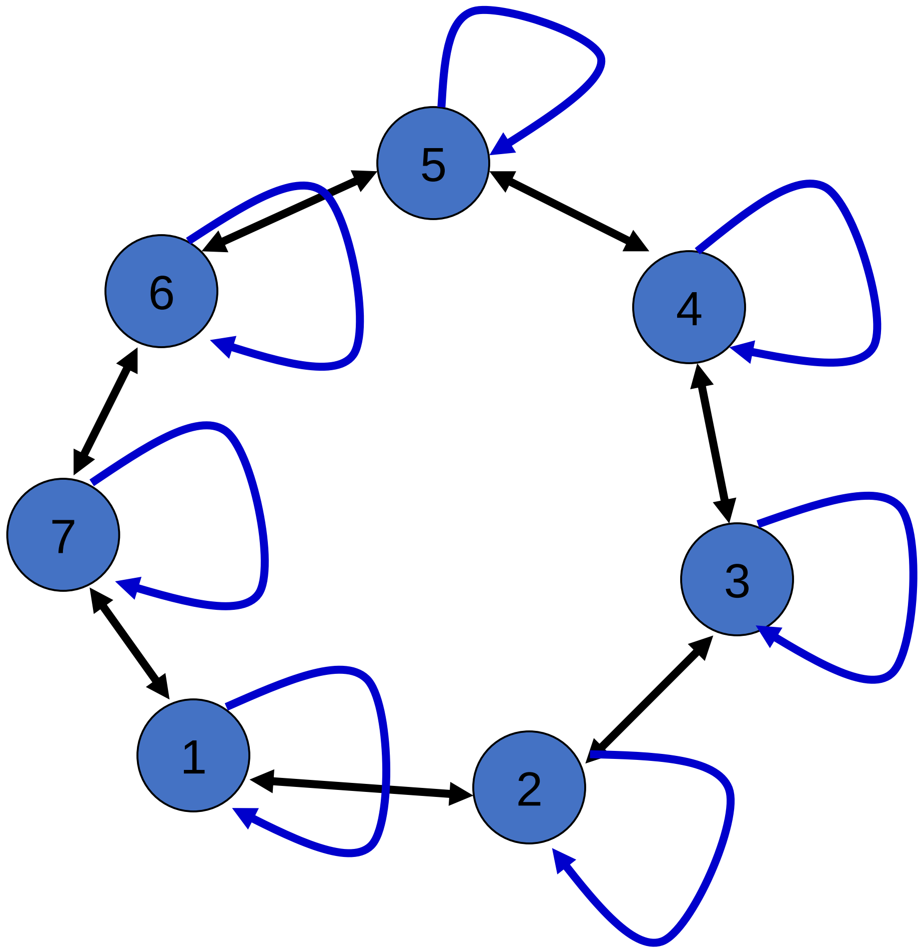
Топологије утицаја



Два најчешће употребљавана модела:

- **gbest**: свака честица је под утицајем најбоље јединке из читавог роја.
- **lbest**: свака честица је под утицајем најбољих јединки из неке своје локалне околине.

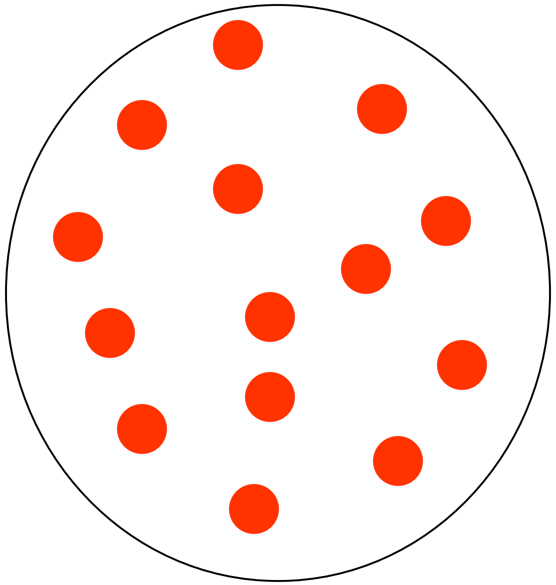
Топологије утицаја (2)



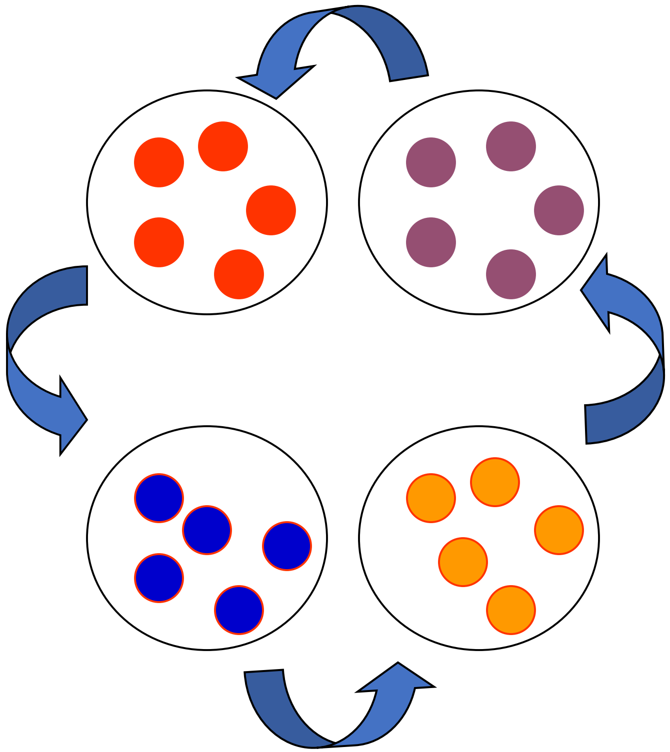
Граф утицаја над ројем од 7 честица. Свака честица зависи од саме себе и од своја два суседа (Clerc, 2006)

Топологије утицаја (3)

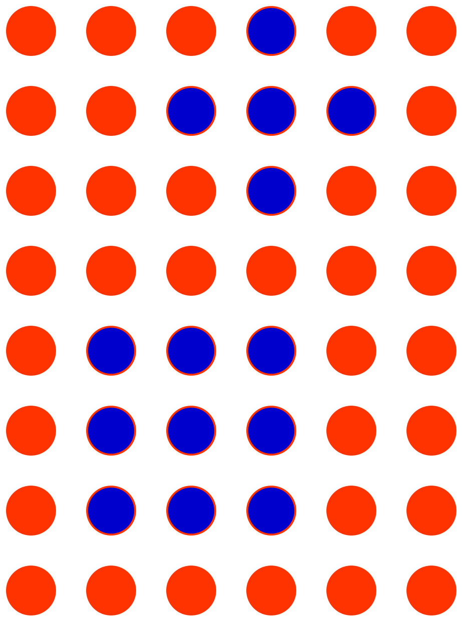
Глобална



Модел острва



Фино градуирана



Топологије утицаја (4)

Коју користити?

Правити компромис између експлоатације и експлорације…

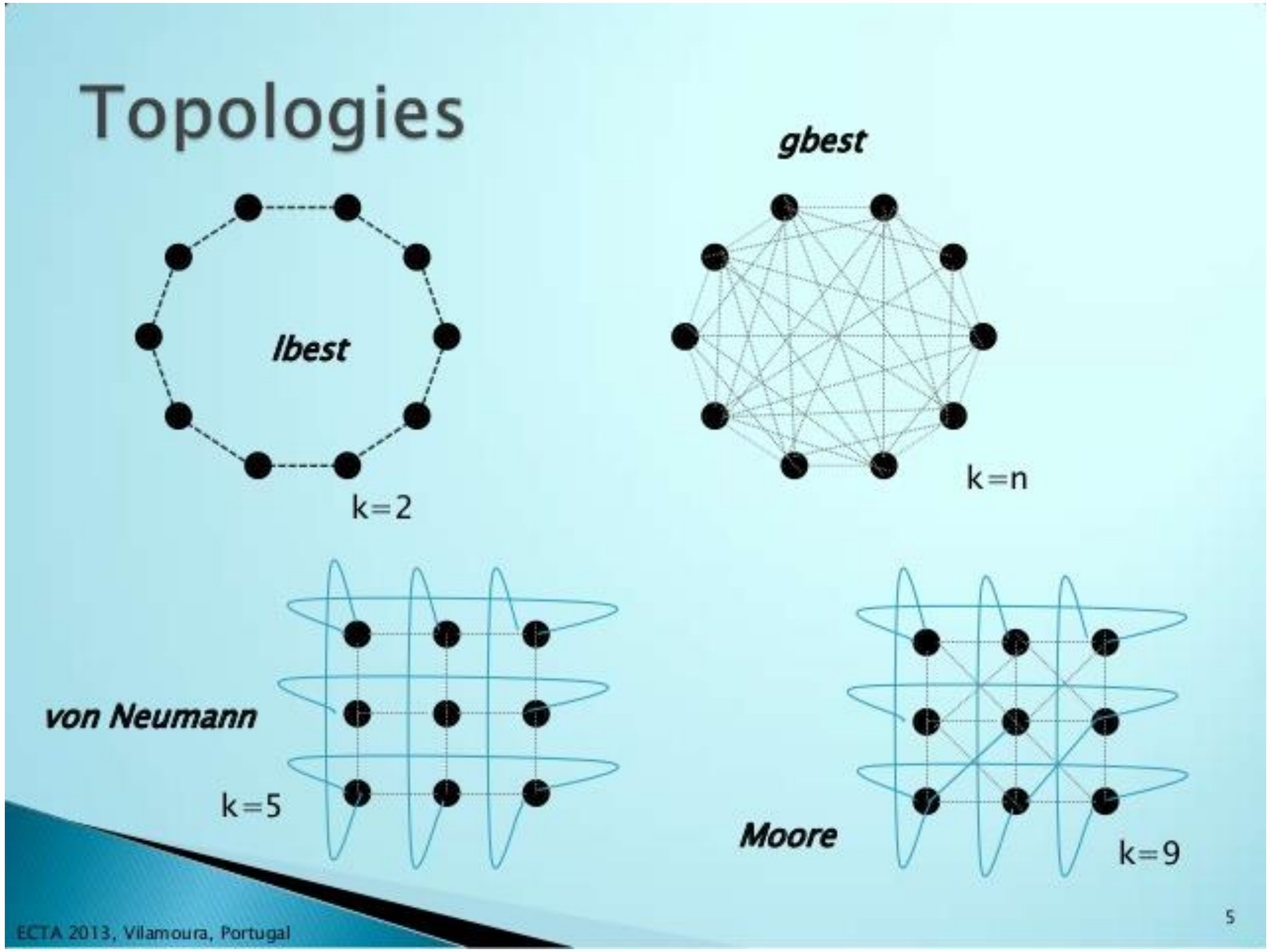
gbest модел најбрже шири информацију широм популације

lbest модел који користи топологију прстена најспорије

За сложене вишемодалне функције, брза пропагација није пожељна. Међутим, спорије ширење информација успорава конвергенцију!

Mendes и Kennedy (2002) су закључили да вон Нојманова топологија (северна, јужна, источна и западна честица у дводимензионој решетци) најбоље понаша међу много различитих топологија.

Топологије утицаја (5)



PSO литература

- Riccardo Poli, James Kennedy, and Tim Blackwell (2007), "Particle swarm optimization - An Overview", Swarm Intelligence, 1: 33–57
- Kennedy, J. Eberhart, R.C., and Shi, Y. (2001), Swarm Intelligence, New York: Morgan Kaufmann Publishers.