Particle Swarm Optimization

- Technika ewolucyjna wzorowana na sposobie poruszania się i inteligencji roju owadów.
- Zaproponowana w 1995 roku przez Jamesa Kennedy'ego i Russella Eberharta.
- Bardzo prosty algorytm, łatwy w implementacji, wymagający doboru jedynie kilku parametrów.
- Podobnie jak w przypadku mrówek, rój składa się z pozornie losowo zachowujących się indywiduów, niemniej jako całość rój zachowuje się w sposób „zdyscyplinowany" i nakierowany na osiągnięcie zamierzonego celu.

Particle Swarm Optimization

- Agenci (cząstki) tworzące rój poszukują optymalnego rozwiązania poruszając się w jego pobliżu w przestrzeni rozwiązań.

- Każda cząstka jest traktowana jako punkt w D-wymiarowej przestrzeni poszukiwań, która dostosowuje swój „lot” zgodnie ze swoim dotychczasowym doświadczeniem oraz doświadczeniem pozostałych uczestników roju (roju jako całości).

- Każda cząstka kontroluje swoje współrzędne w przestrzeni poszukiwań, które są związane z najlepszym dotychczas przez nią znalezionym rozwiązaniem („personal best” – ***pbest***).

Particle Swarm Optimization

- Drugą wartością kontrolowaną przez PSO jest dotychczas najlepsza znaleziona pozycja w ramach roju (rozwiązanie) określana jako ***gbest*** (global best).

- Idea PSO polega na zmianie prędkości (przyspieszaniu) w kierunku jej pozycji *pbest* oraz globalnej pozycji *gbest* w każdym kroku czasowym.

- Każda cząstka modyfikuje swoją aktualna pozycję i prędkość zgodnie z odległością pomiędzy jej pozycją a pozycją *pbest* oraz odległością pomiędzy jej pozycją a pozycją *gbest*.

Particle Swarm Optimization

v_n : prędkość cząstki w iteracji *n*

x_n : pozycja cząstki w iteracji *n*

p_{best} : najlepsza dotychczasowa pozycja cząstki

g_{best} : najlepsza dotychczasowa pozycja cząstek w całym roju

c_1 : współczynnik przyspieszania związany z p_{best}

c_2 : współczynnik przyspieszania związany z g_{best}

W: inercja

rand( ): losowa liczba z przedziału (0, 1)

$$v_{n+1} = w \cdot v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

$$x_{n+1} = x_n + v_{n+1}$$

← to są wektory D-wymiarowe

Algorytm PSO

Dla każdej cząstki

Zainicjuj ją losowo (położenie i prędkość) - zwykle z określonego „dopuszczalnego” zakresu

Do *//w każdej iteracji*

{

Dla każdej cząstki *//aktualizacja pbest każdej cząstki*

{

Policz wartość dopasowania

Jeżeli wartość dopasowania jest większa od najlepszego dotychczasowego dopasowania (**pbest**) to ustaw **pbest** jako aktualną wartość dopasowania

}

Wybierz najlepsze dotychczasowe dopasowanie w całym roju jako **gbest** *//aktualizacja gbest roju*

Dla każdej cząstki *// aktualizacja położenia i prędkości cząstek*

{

Policz jej szybkość zgodnie zrównaniem szybkości

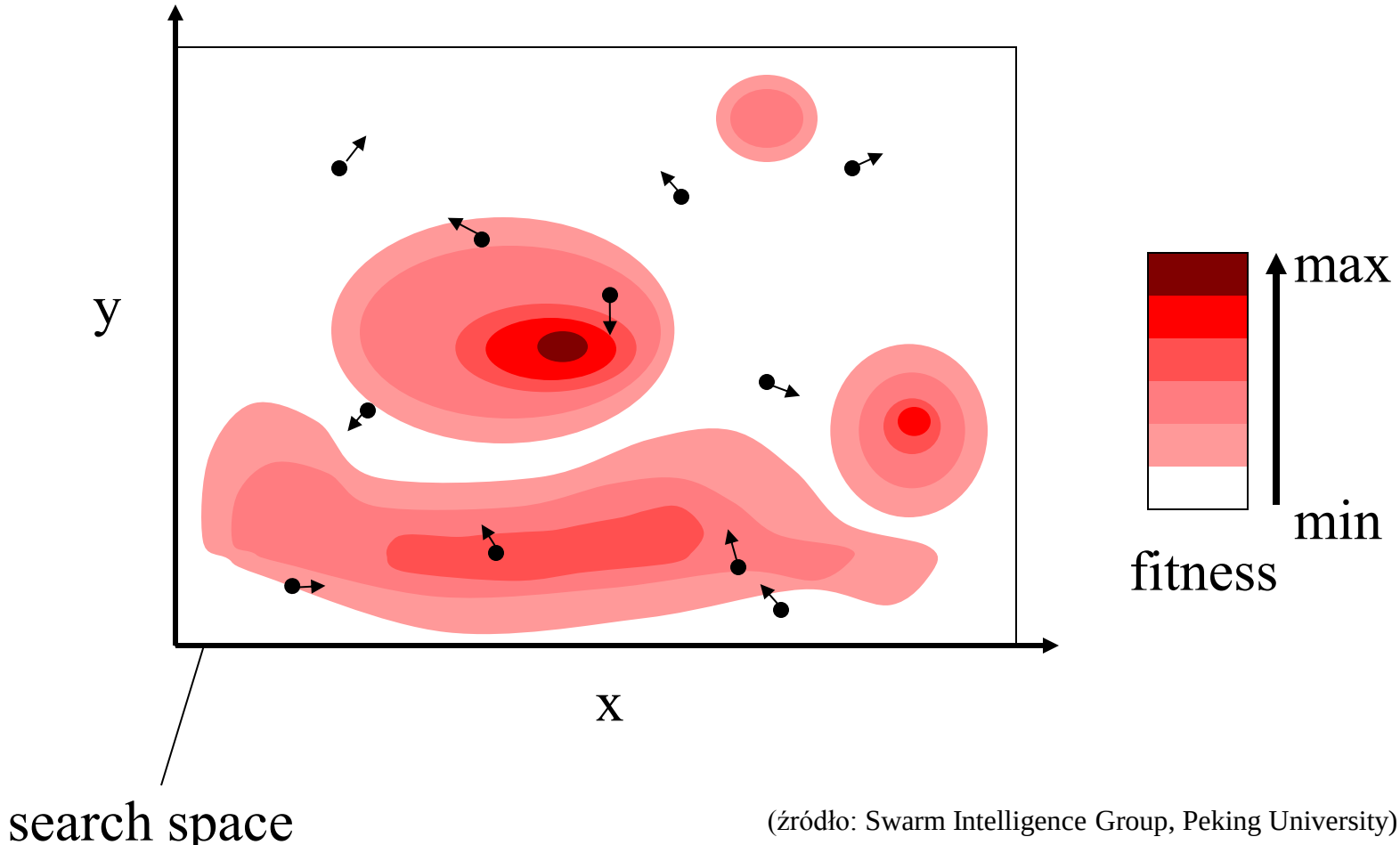
Zmodyfikuj jej pozycję zgodnie z równaniem zmiany pozycji

}

}

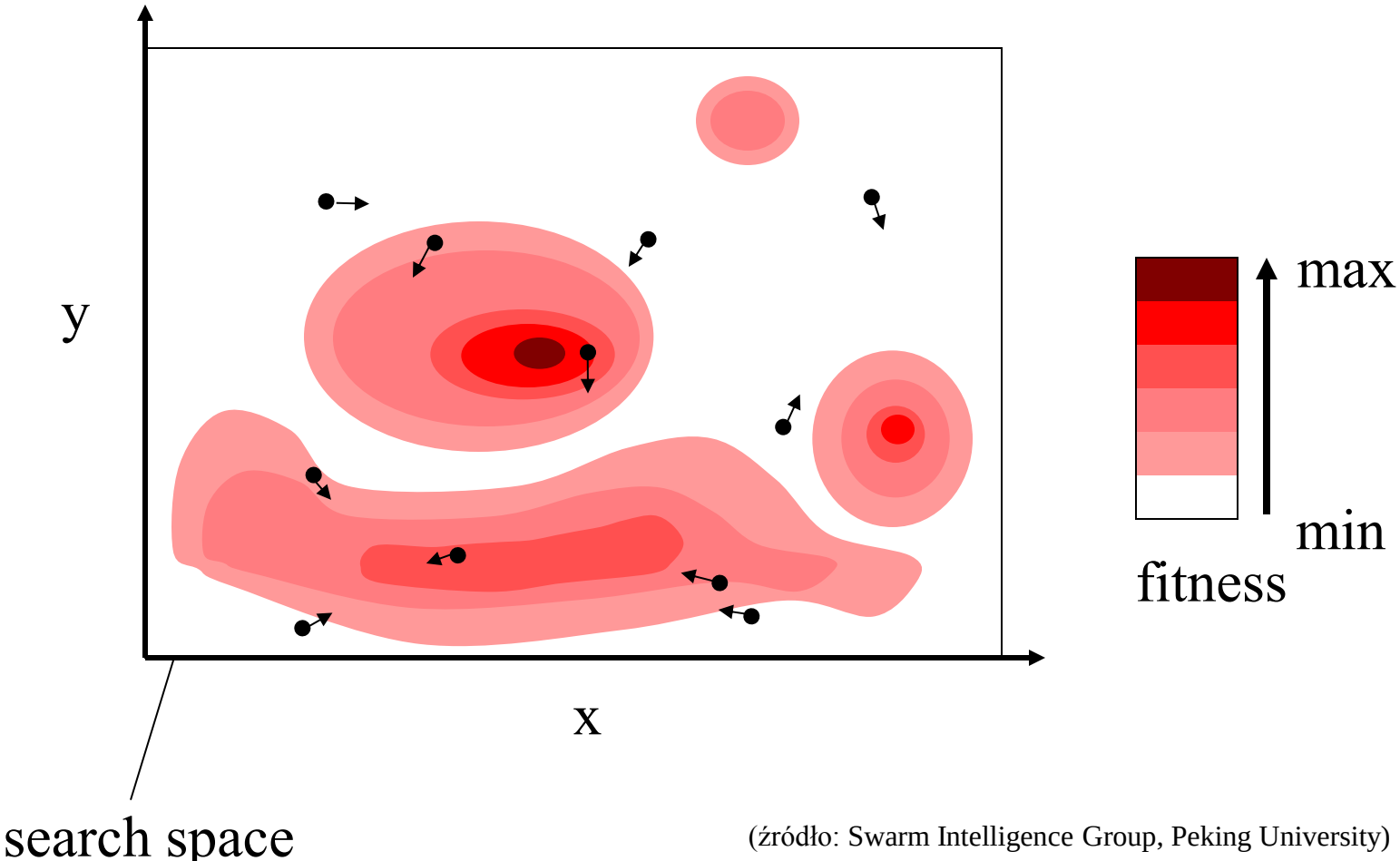
Dopóki nie przekroczono maksymalnej liczby iteracji lub błąd nie osiągnął zamierzonego minimum

Symulacja 1



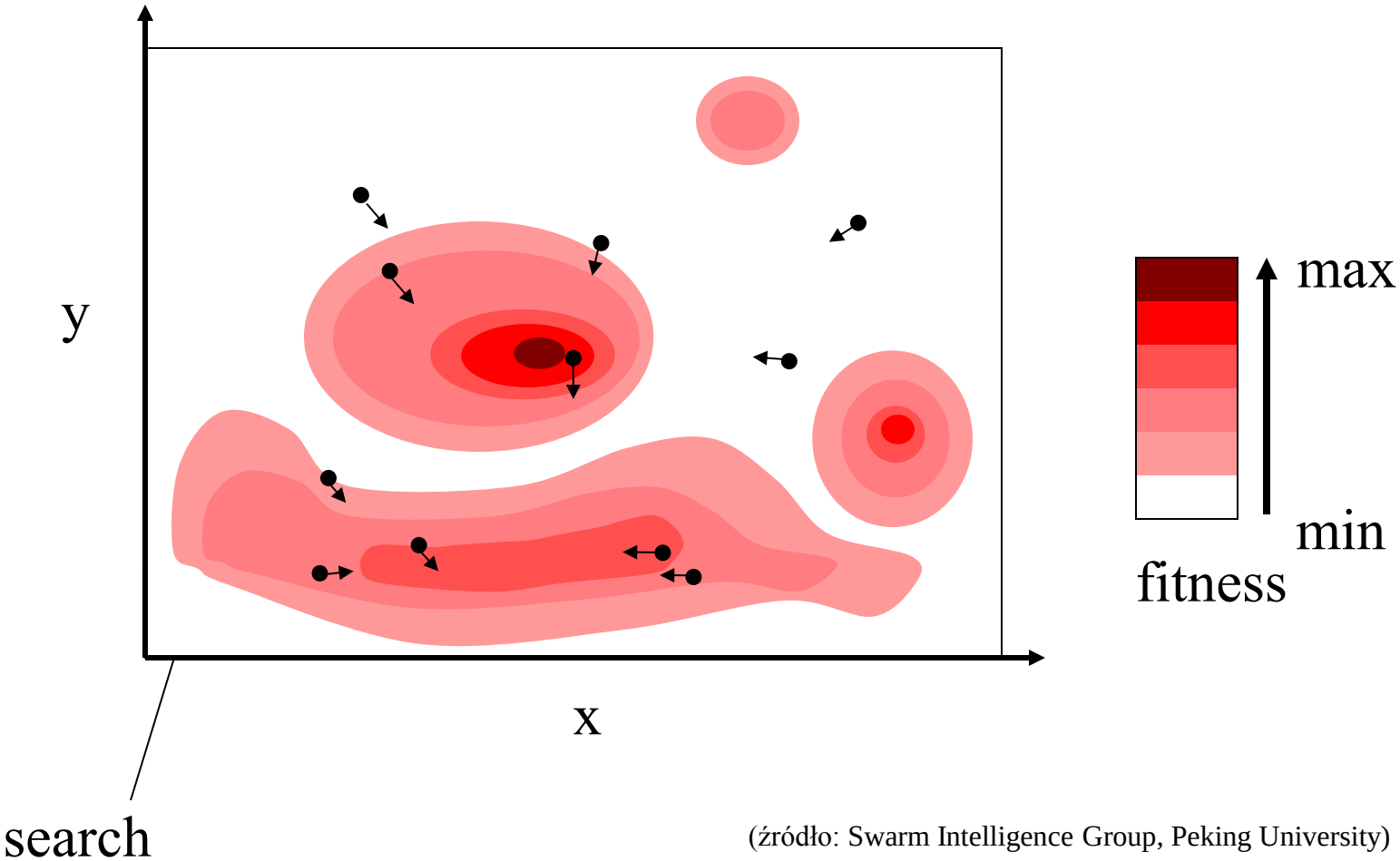
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 2



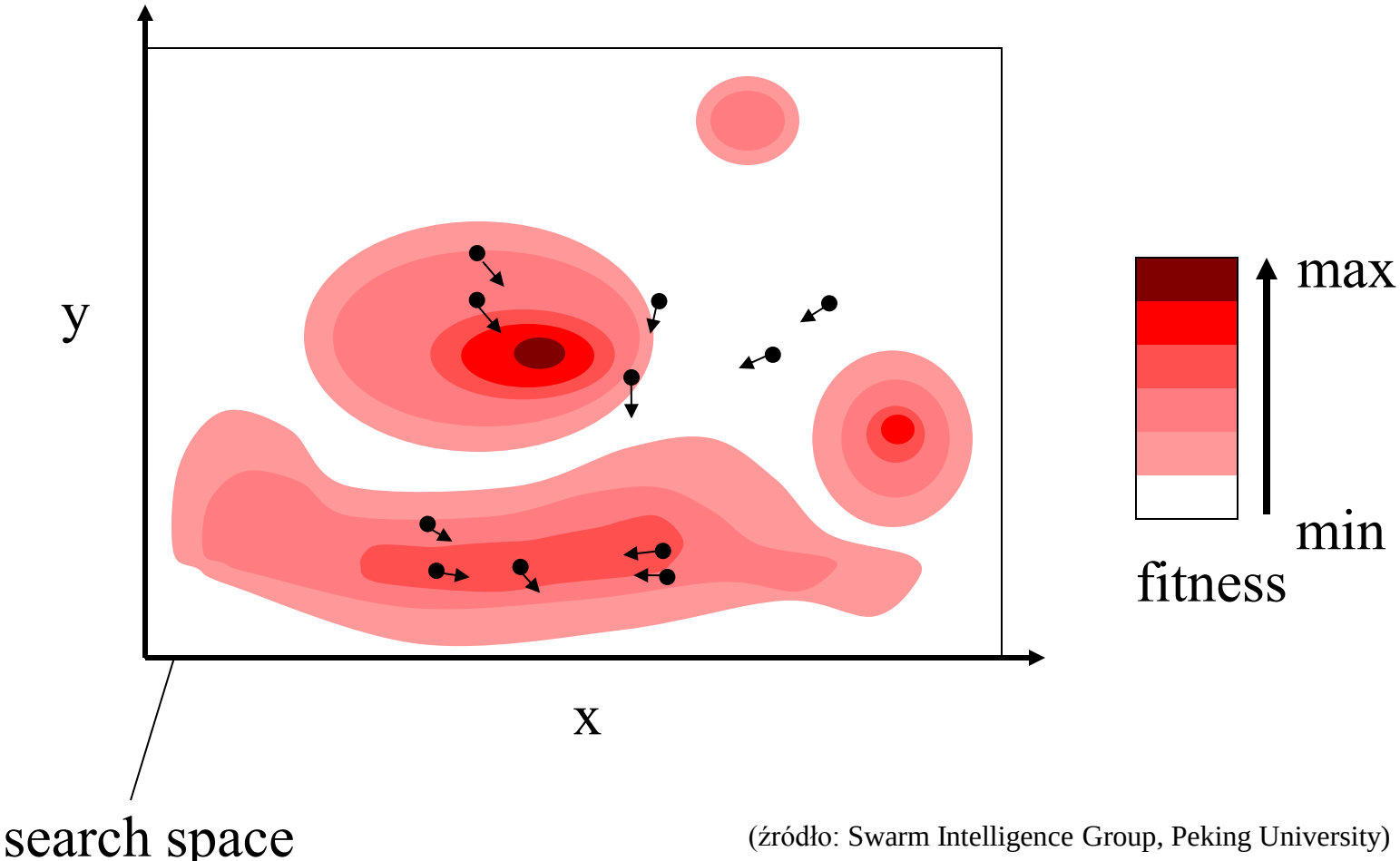
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 3



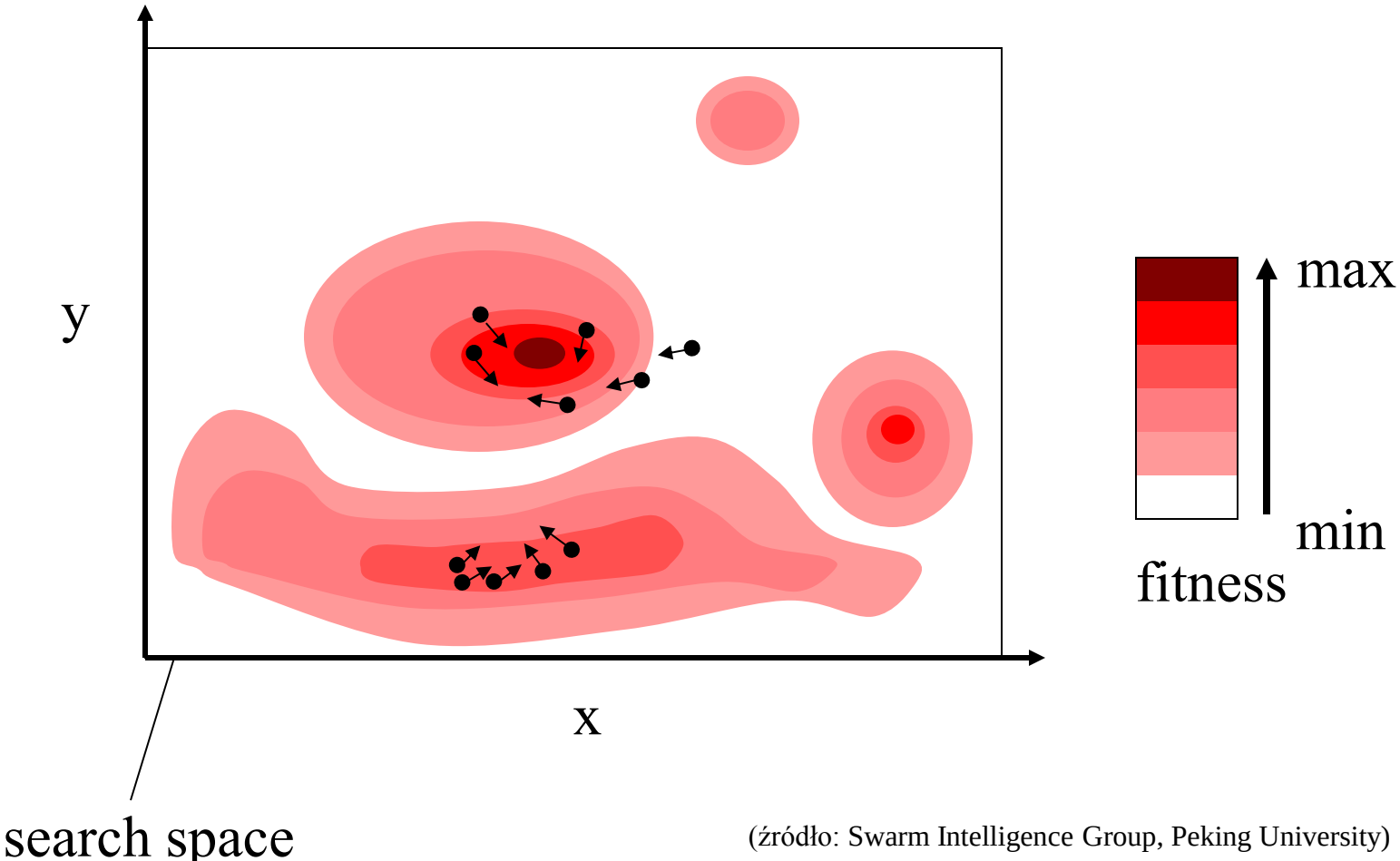
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 4



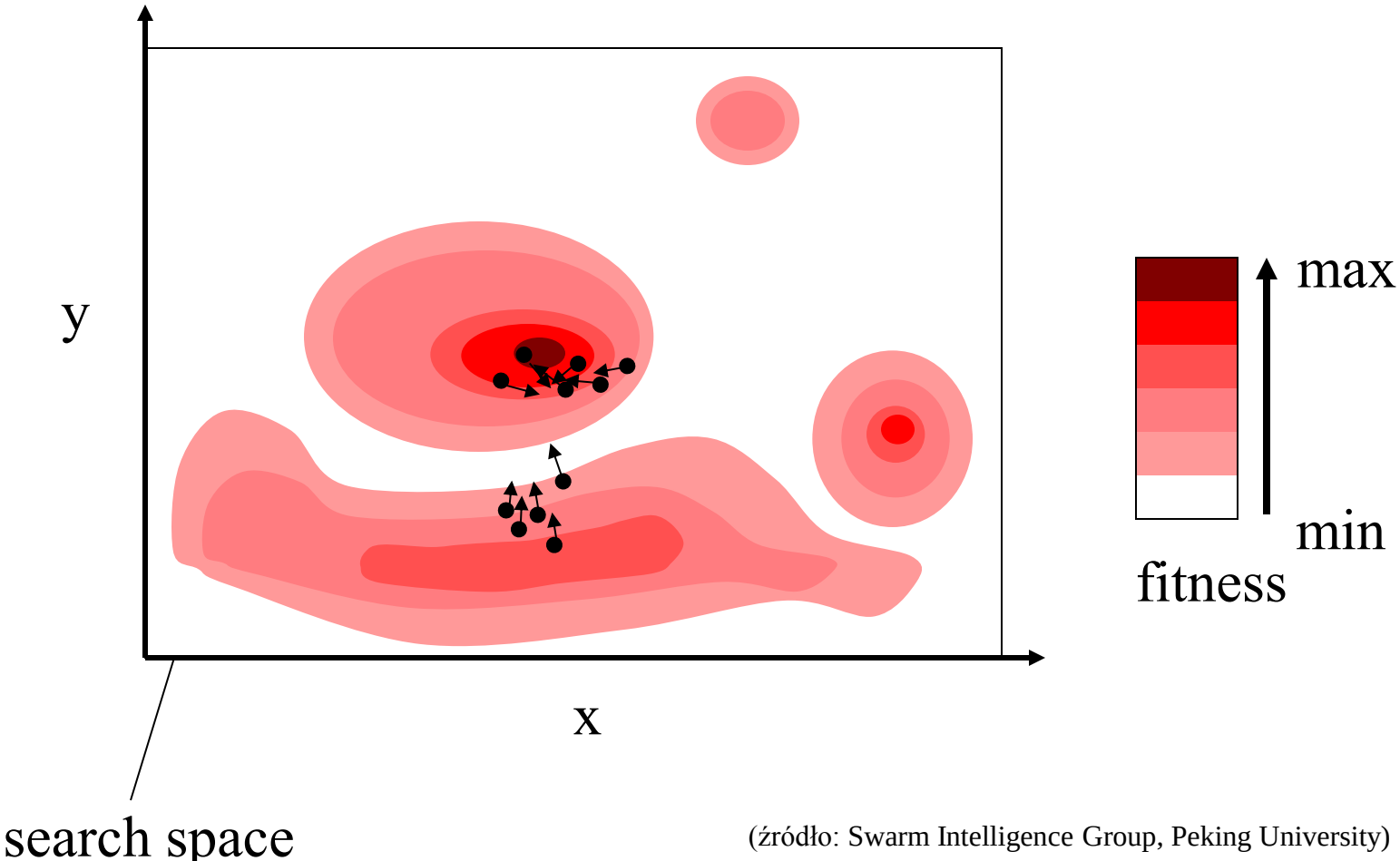
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 5



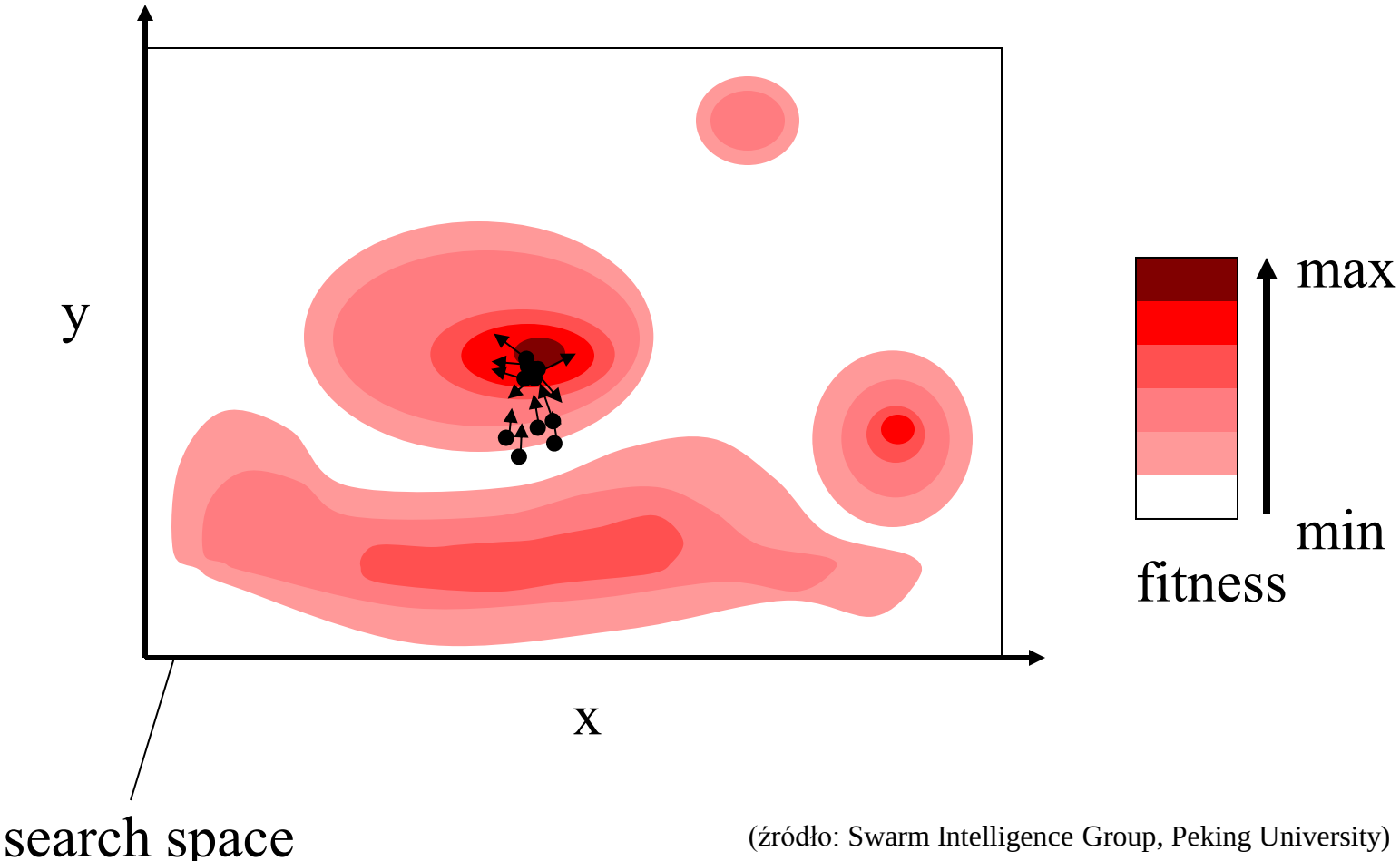
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 6



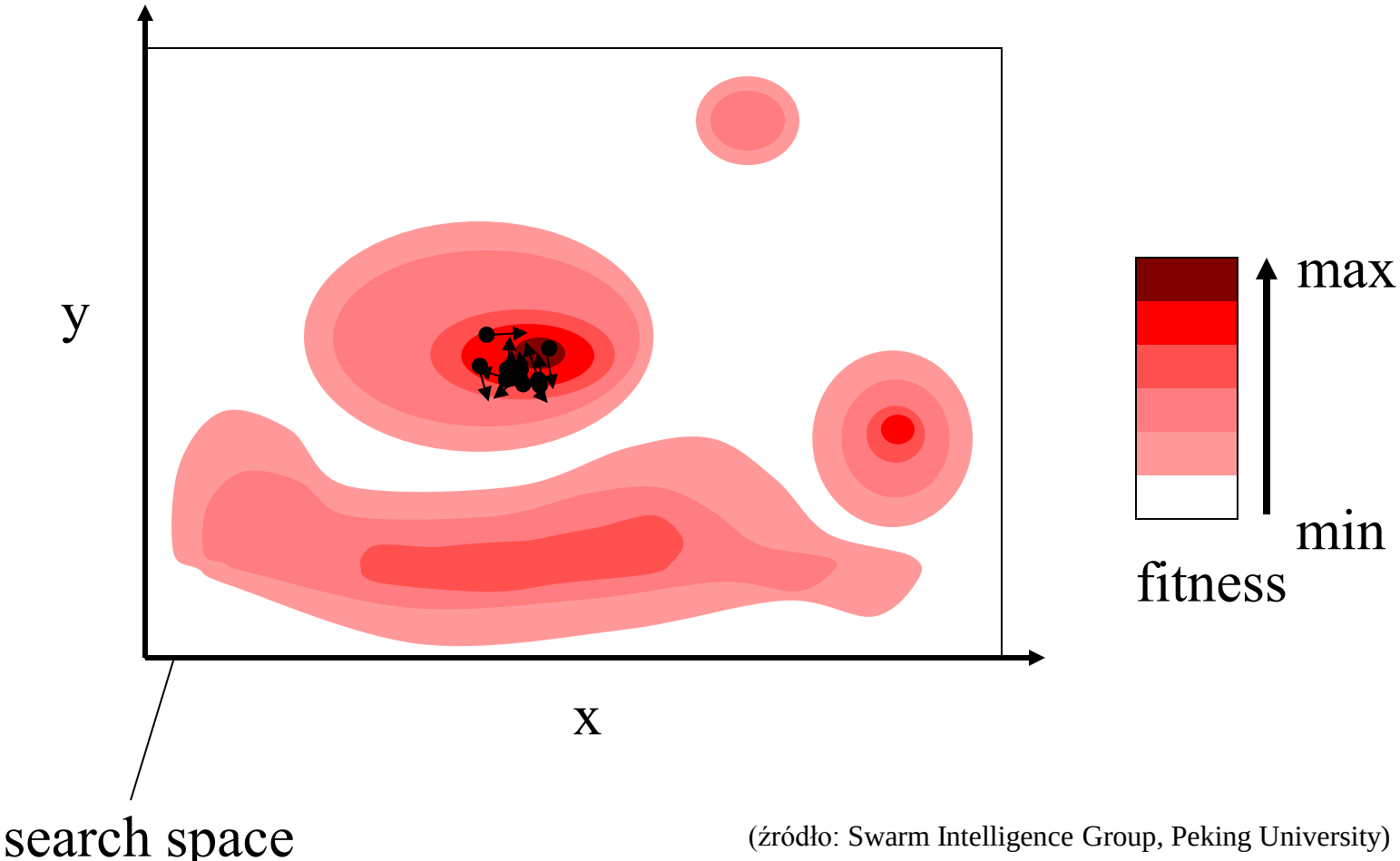
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 7



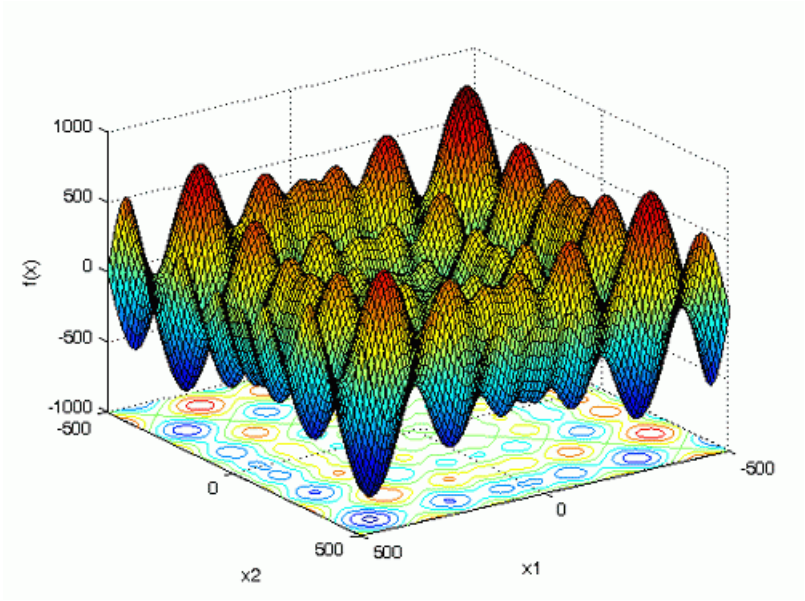
(źródło: Swarm Intelligence Group, Peking University)

Symulacja 8



(źródło: Swarm Intelligence Group, Peking University)

Funkcja Schwefela



$$f(x) = \sum_{i=1}^{n} (-x_i) \cdot \sin(\sqrt{|x_i|})$$

gdzie

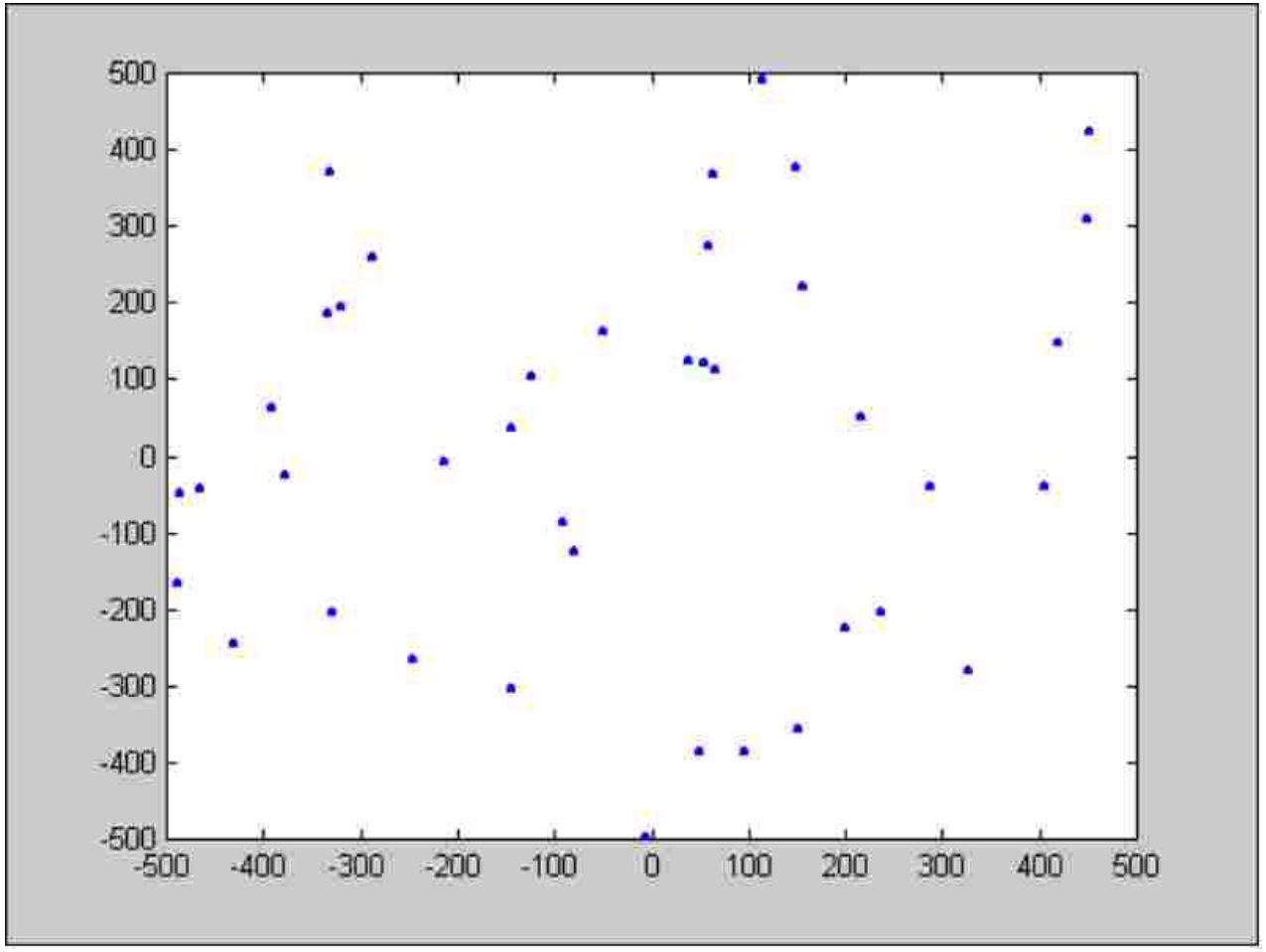
$$-500 \le x_i \le 500$$

minimum globalne : $-n \cdot 418.9829;$

$$x_i = -420.9687,\ i = 1,\ldots, n$$

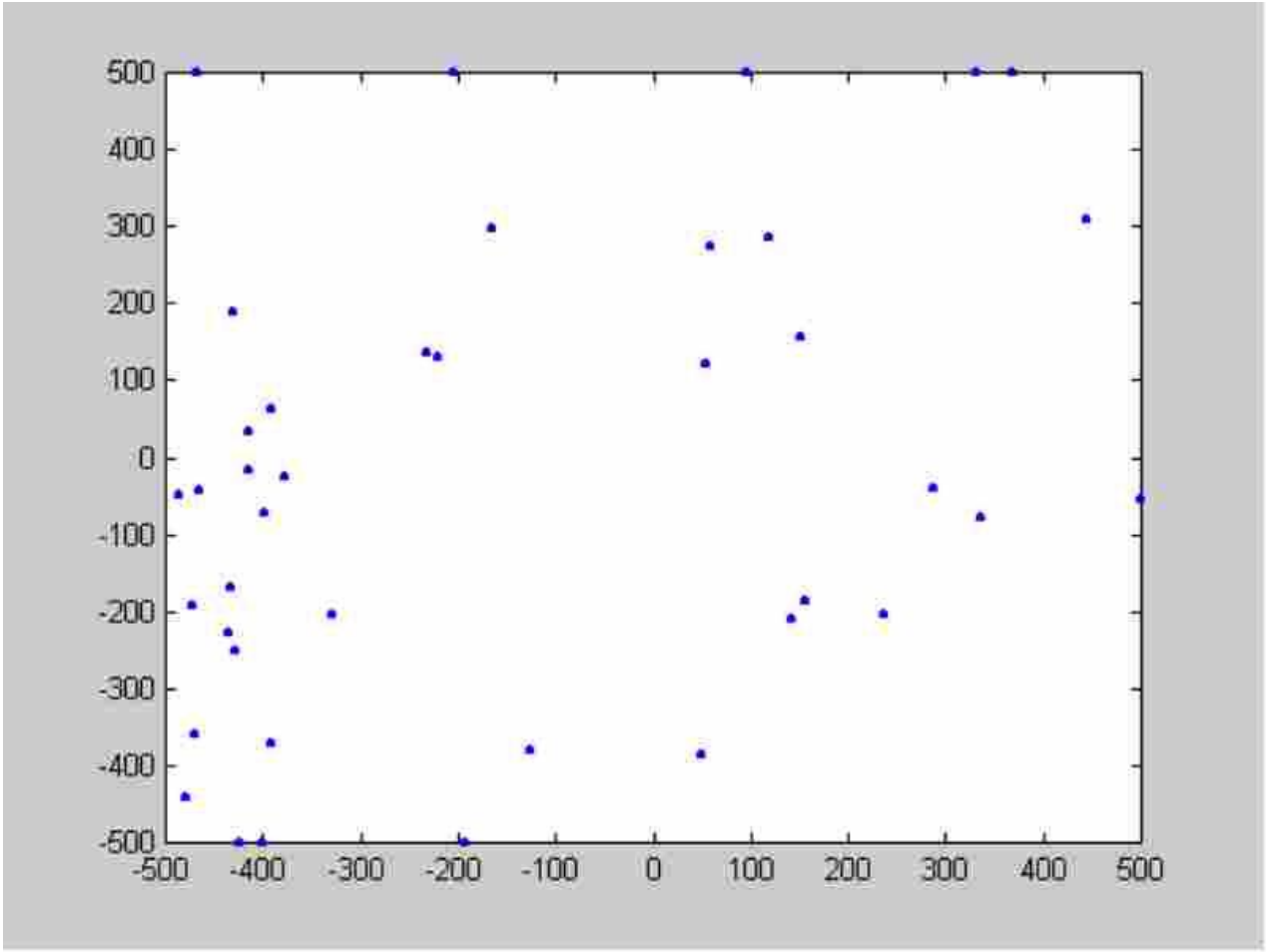
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—Inicjalizacja



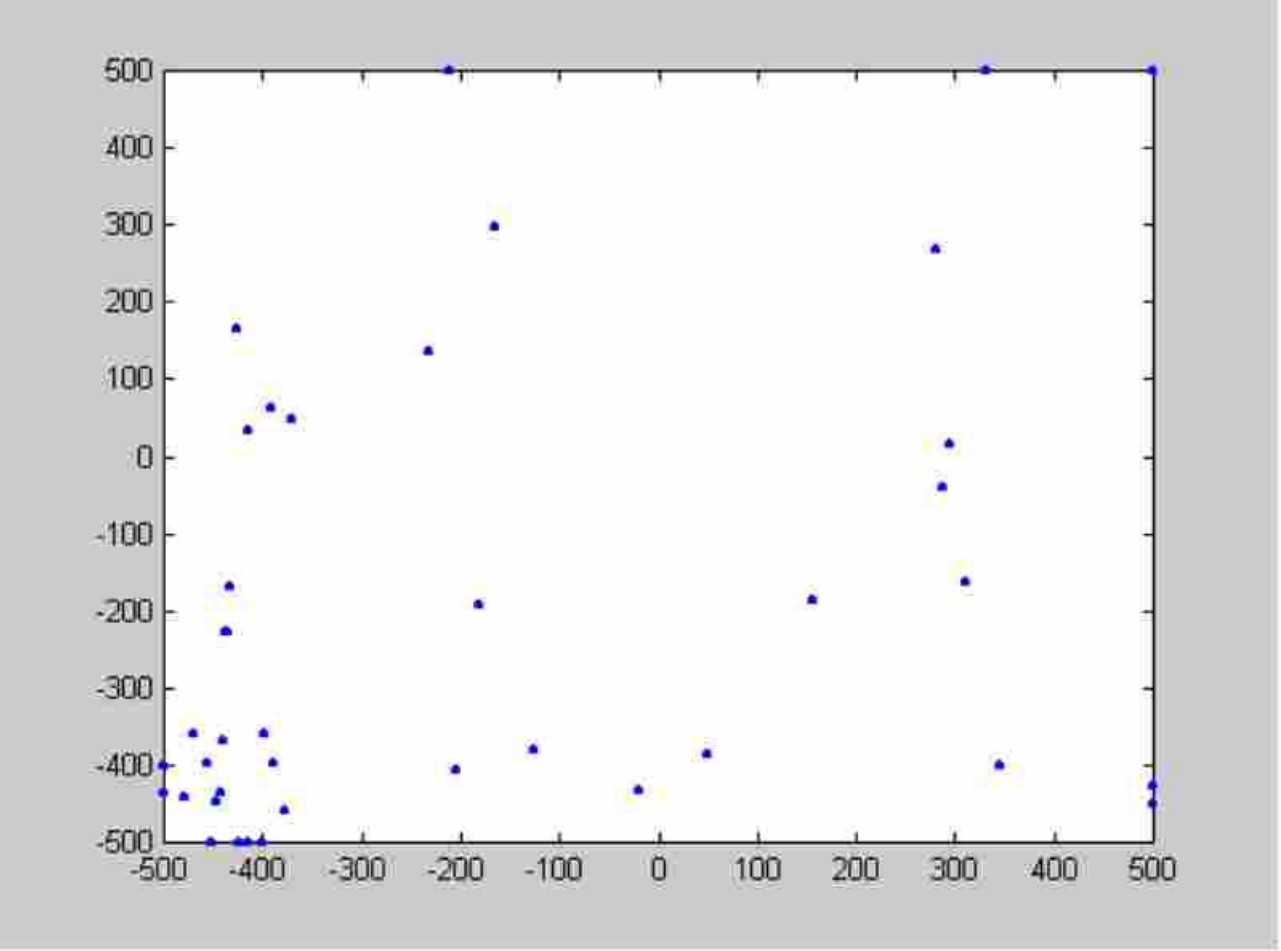
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—5 iteracja



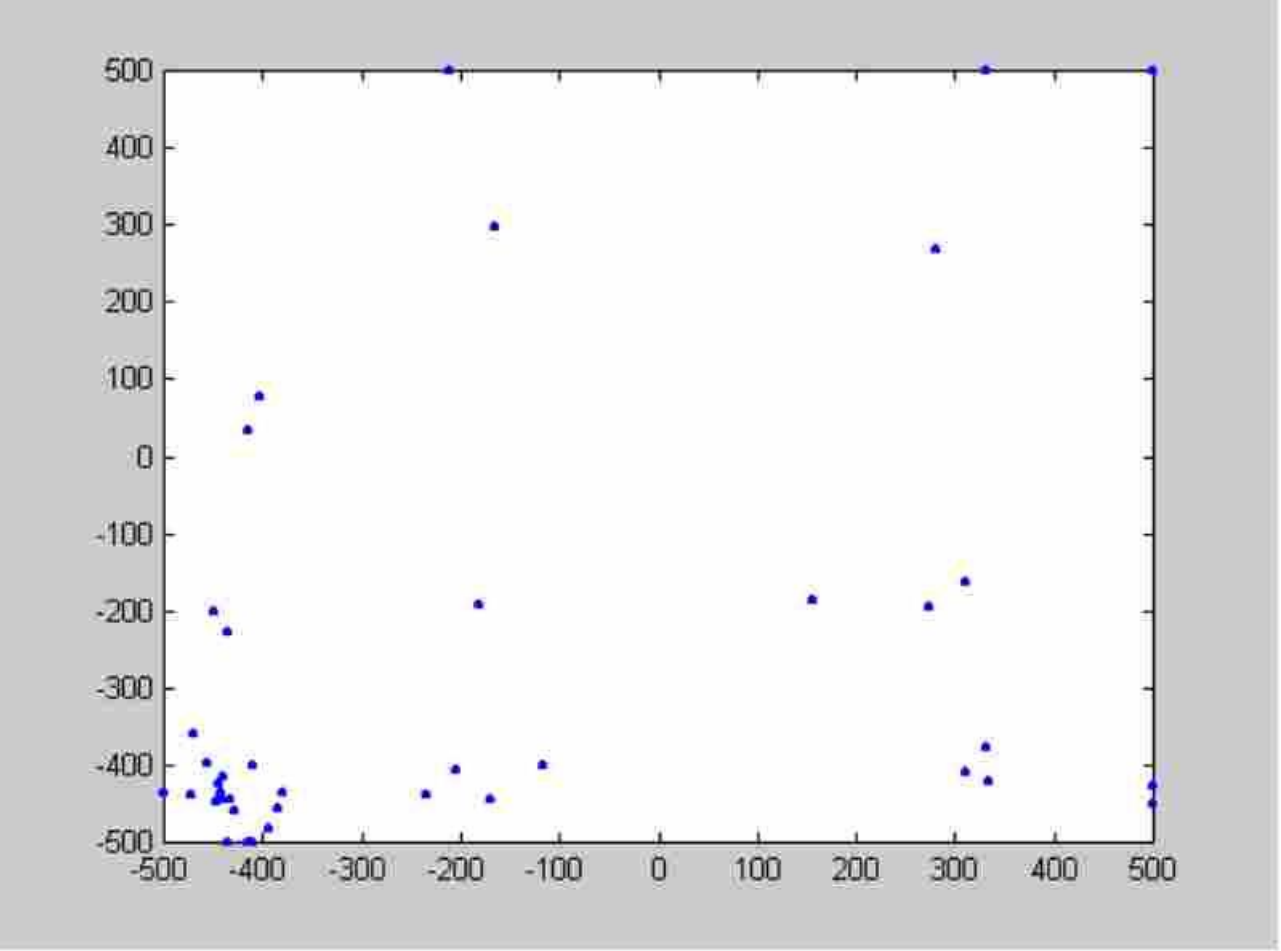
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—10 iteracja



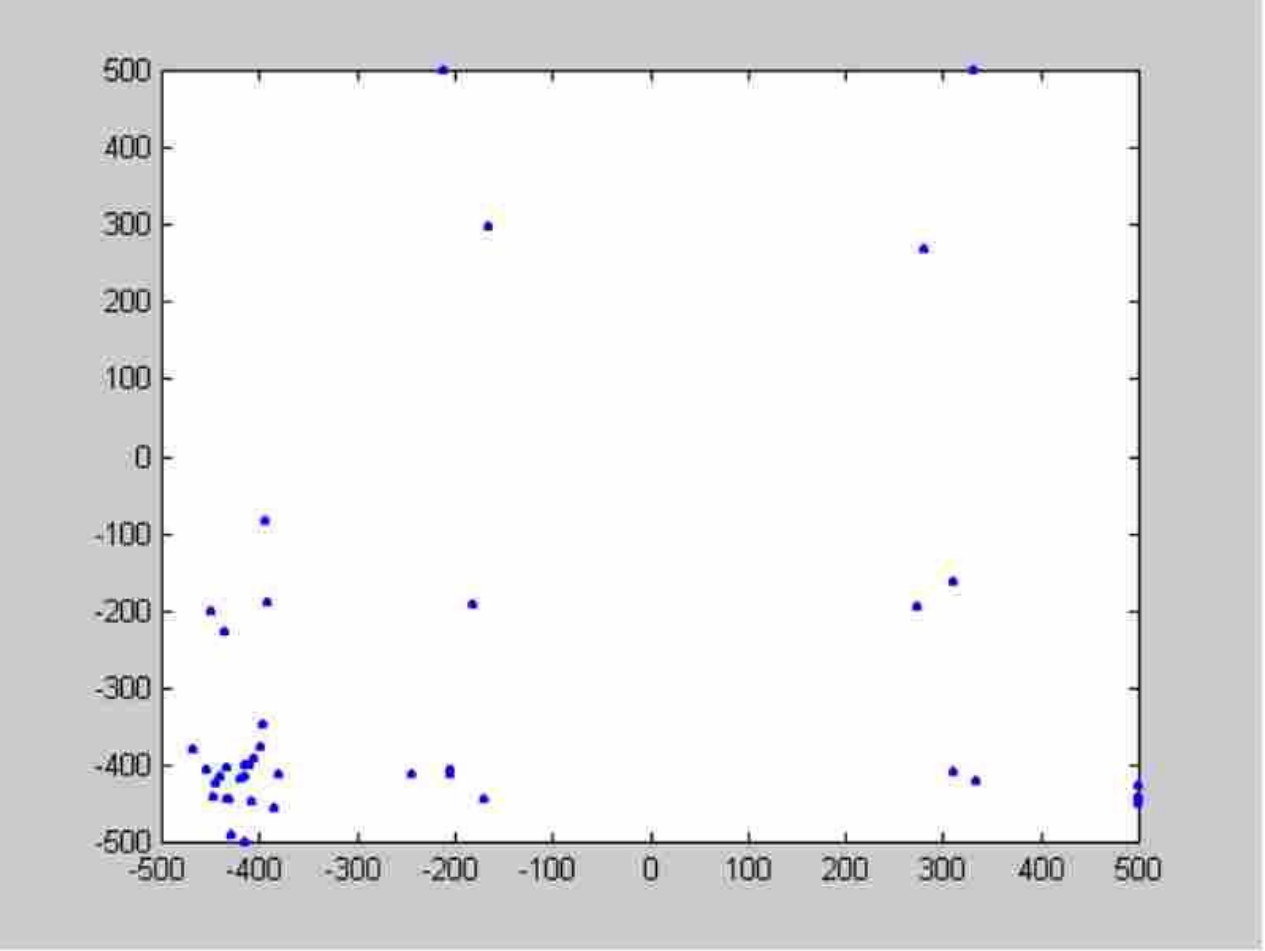
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—15 iteracja



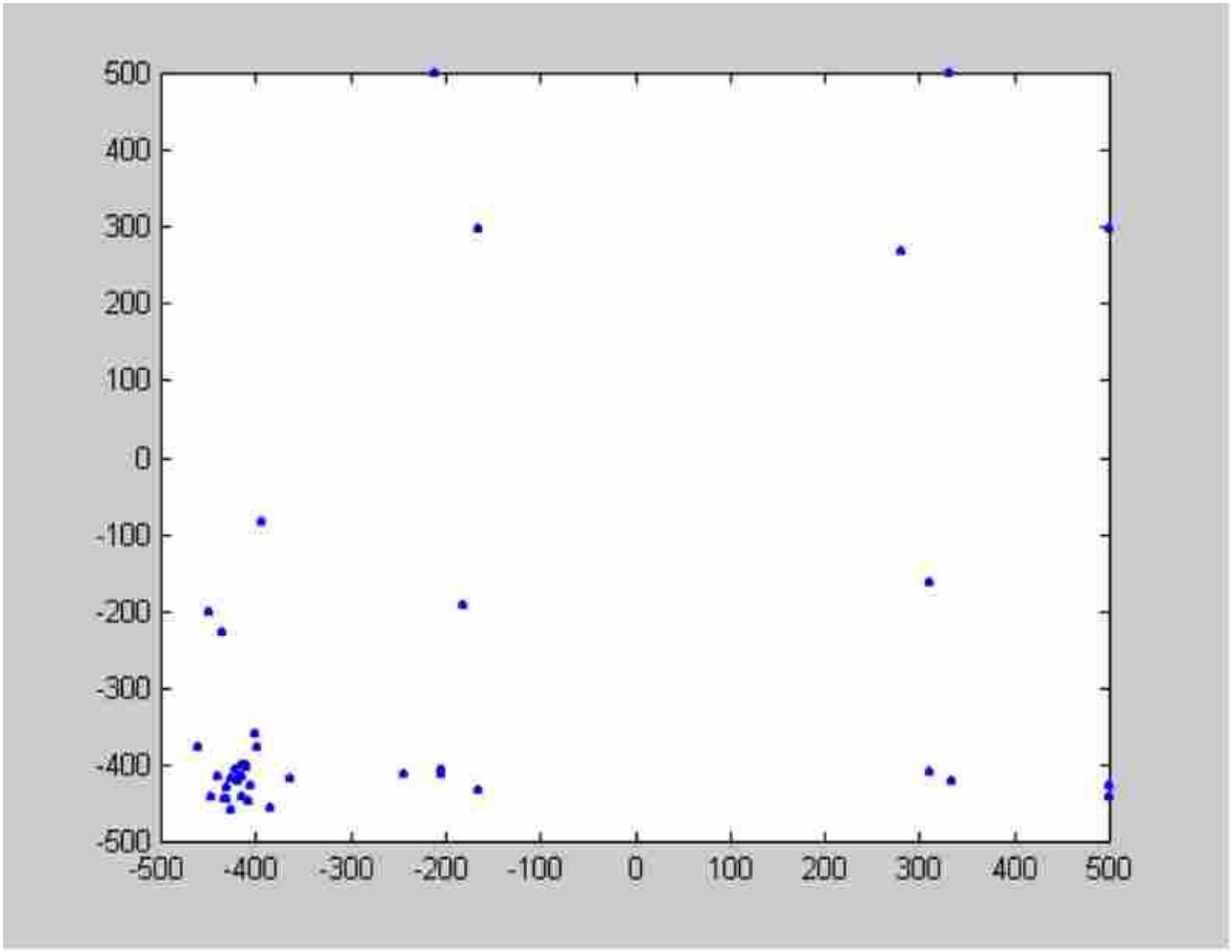
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—20 iteracja



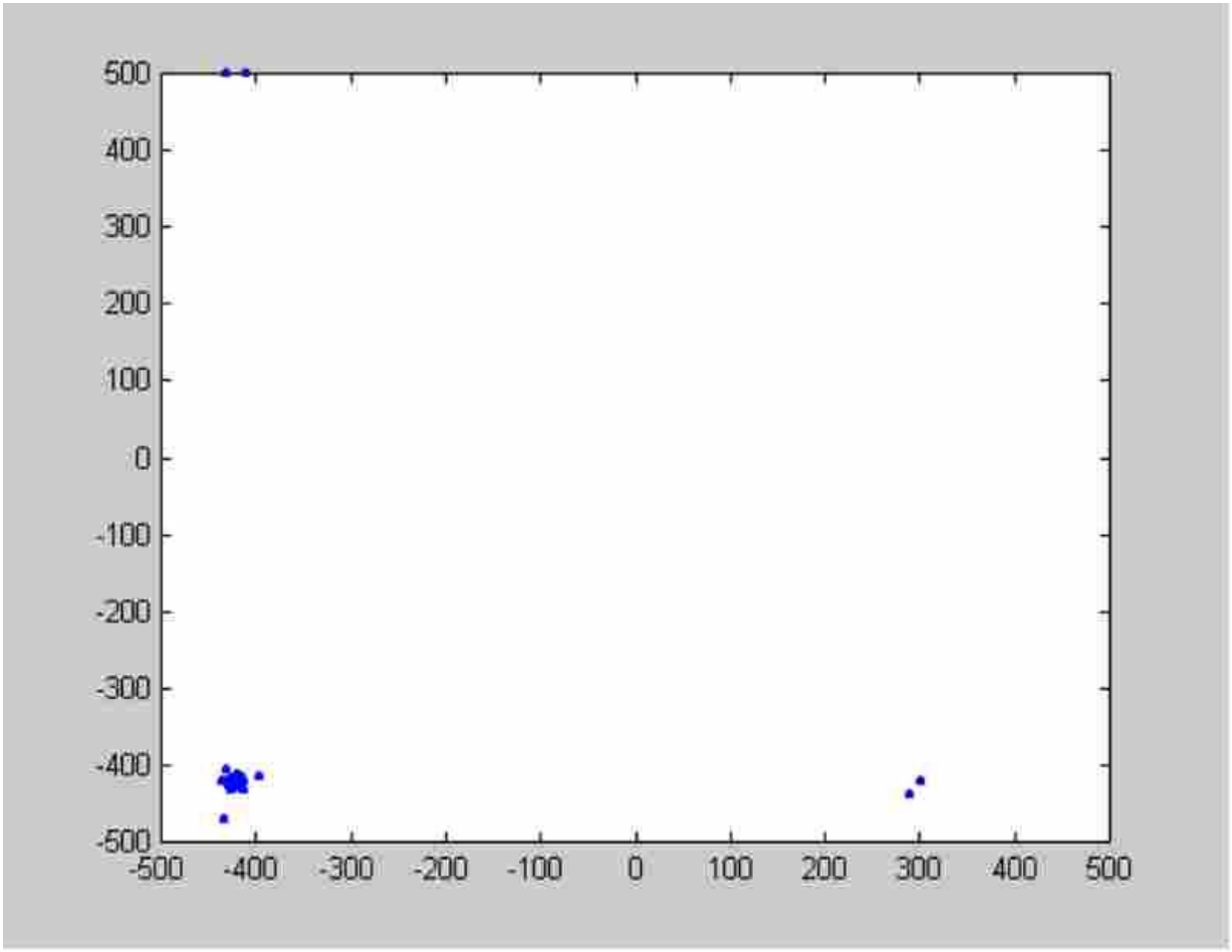
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—25 iteracja



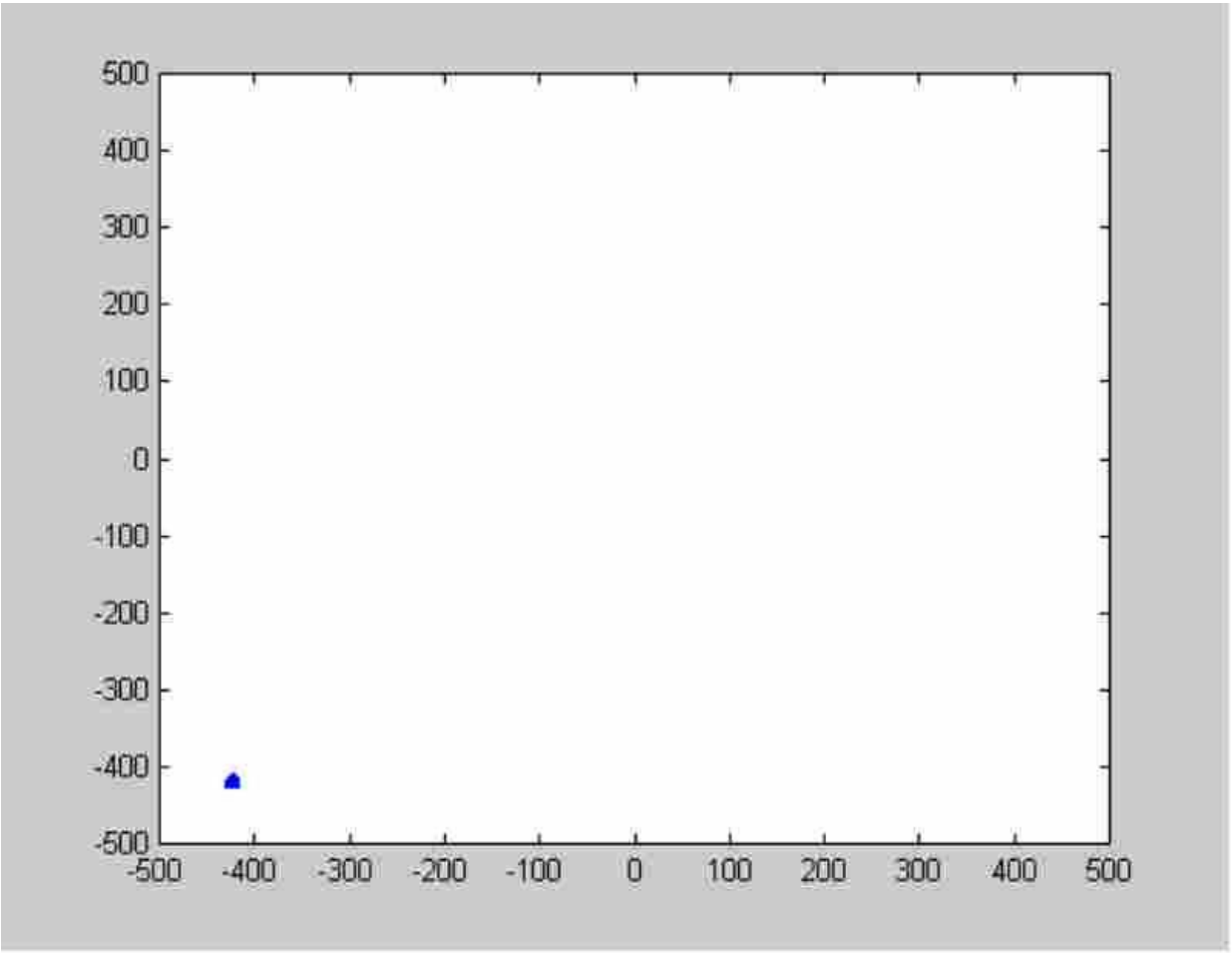
(źródło: Swarm Intelligence Group, Peking University)

Ewolucja—100 iteracja



(źródło: Swarm Intelligence Group, Peking University)

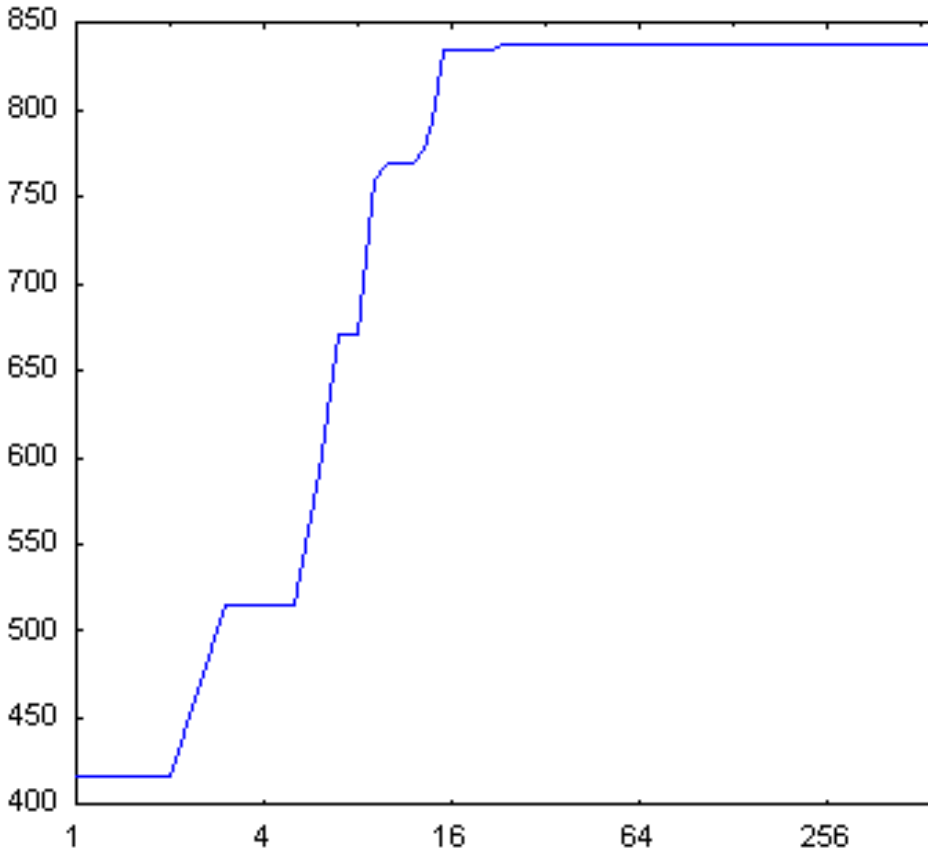
Ewolucja—500 iteracja



(źródło: Swarm Intelligence Group, Peking University)

Wyniki

Iteracja	Znalezione rozw.
0	-416.245599
5	-515.748796
10	-759.404006
15	-793.732019
20	-834.813763
100	-837.911535
5000	-837.965771
Globalne	-837.965745



(źródło: Swarm Intelligence Group, Peking University)

gbest / lbest (sąsiedztwo)

- Wersja globalna:

$$v_{n+1} = w \cdot v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

- Wersja lokalna:

$$v_{n+1} = w \cdot v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (l_{best} - x_n)$$

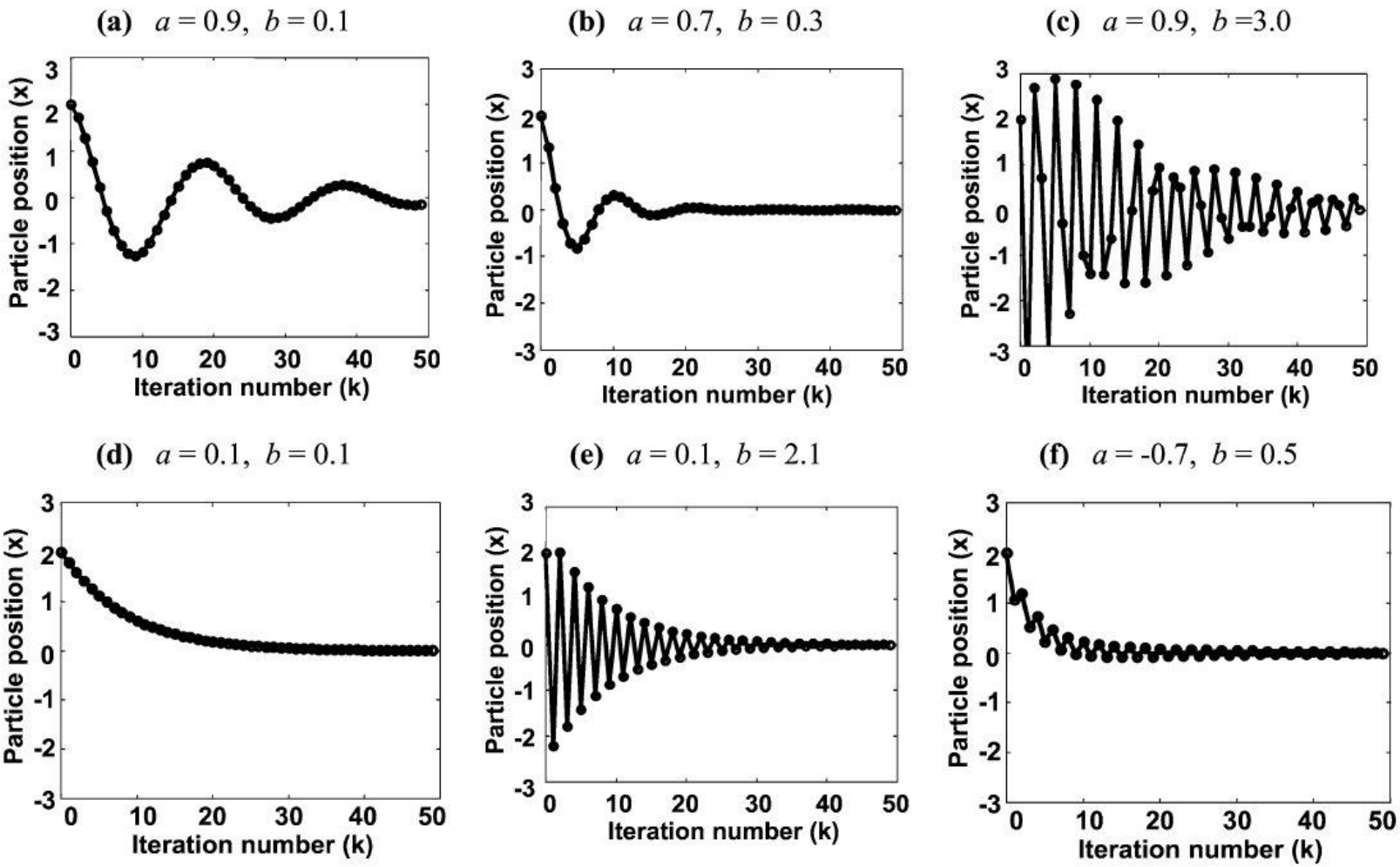
- Wersja lokalna_2:

$$v_{n+1} = w \cdot \left[v_n + \sum_{k \in Neighbors} c \cdot rand() \cdot (p_k - x_n) \right]$$

- Wersja lokalna/globalna ogólnie:

$$v_{n+1} = a \cdot v_n + b \cdot (r - x_n) \quad gdzie \quad r \approx p_{best} \quad \text{lub} \quad r \approx g_{best}$$

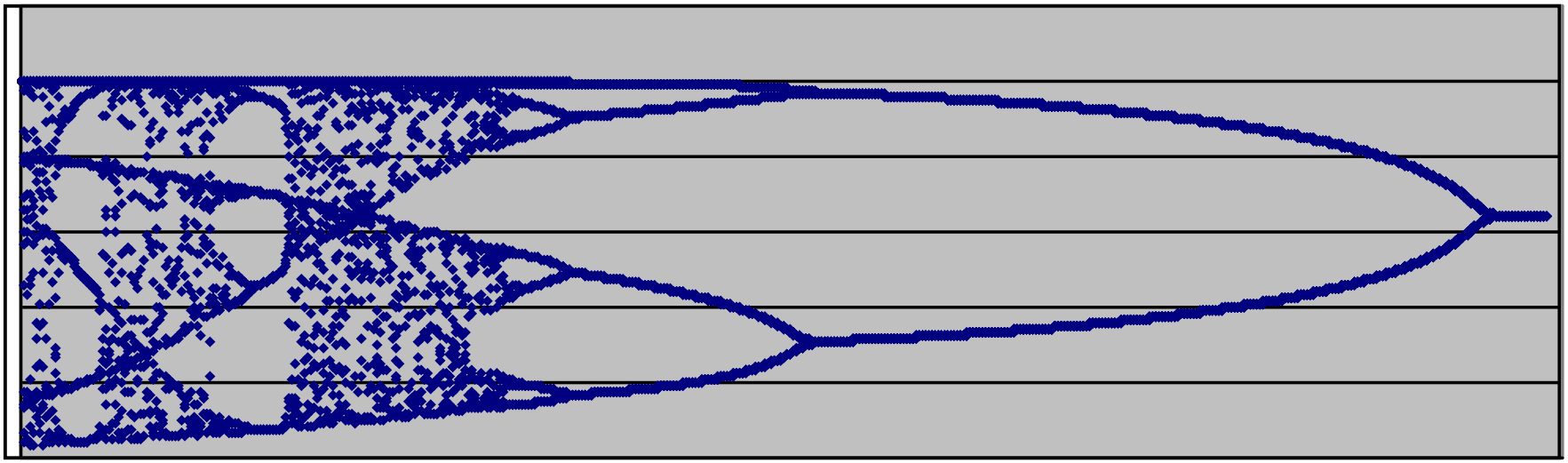
$$v_{k+1} = a \cdot v_k + b \cdot (r - x_k)$$



Pojęcie chaosu deterministycznego

System chaotyczny charakteryzuje się nieregularnym, nieprzewidywalnym zachowaniem (tzw. ***butterfly effect***).

System chaotyczny jest bardzo wrażliwy na warunki początkowe. Trajektorie zaczynające się w punktach położonych blisko siebie bardzo szybko rozchodzą się.



Przejście od zachowania liniowego do chaotycznego (bądź odwrotnie) zawiera bardzo często fazę *bifurkacji* po niej fazę *quadruplikacji*, itd. Możliwe są też inne zachowania chaotyczne.

Równanie logistyczne – przykład systemu chaotycznego

Jeden z klasycznych przykładów systemów chaotycznych to równanie logistyczne:

$$x_{n+1} = r x_n (1 - x_n),$$

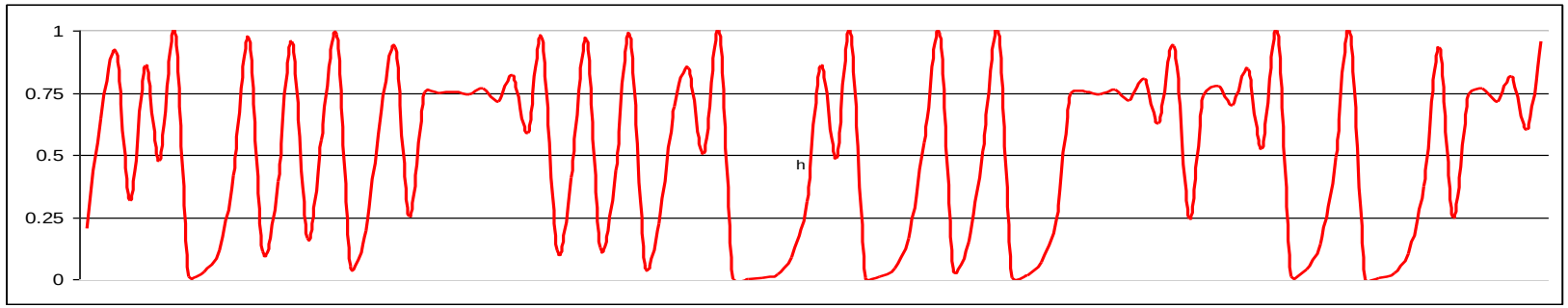
gdzie r > 0 jest stałą.

Własności:

- Zależnie od doboru stałej *r* oraz wyboru x_0 otrzymuje się różne zbiory punktów, które reprezentują różne zachowania chaotyczne

- Dla $x_0 \in [0,1]$ system utrzymuje się w przedziale [0,1] wtw., gdy $r \in (0,4]$

Równanie logistyczne – przykład (r = 4)



Przykłady innych systemów chaotycznych

Sunspot series: liczba zaobserwowanych plam na słońcu w kolejnych dniach (dostępne dane od XVIII wieku)

Mackey-Glass equation (model różnych zburzeń fizjologicznych, np. produkcji białych krwinek u pacjentów chorych na leukemię):

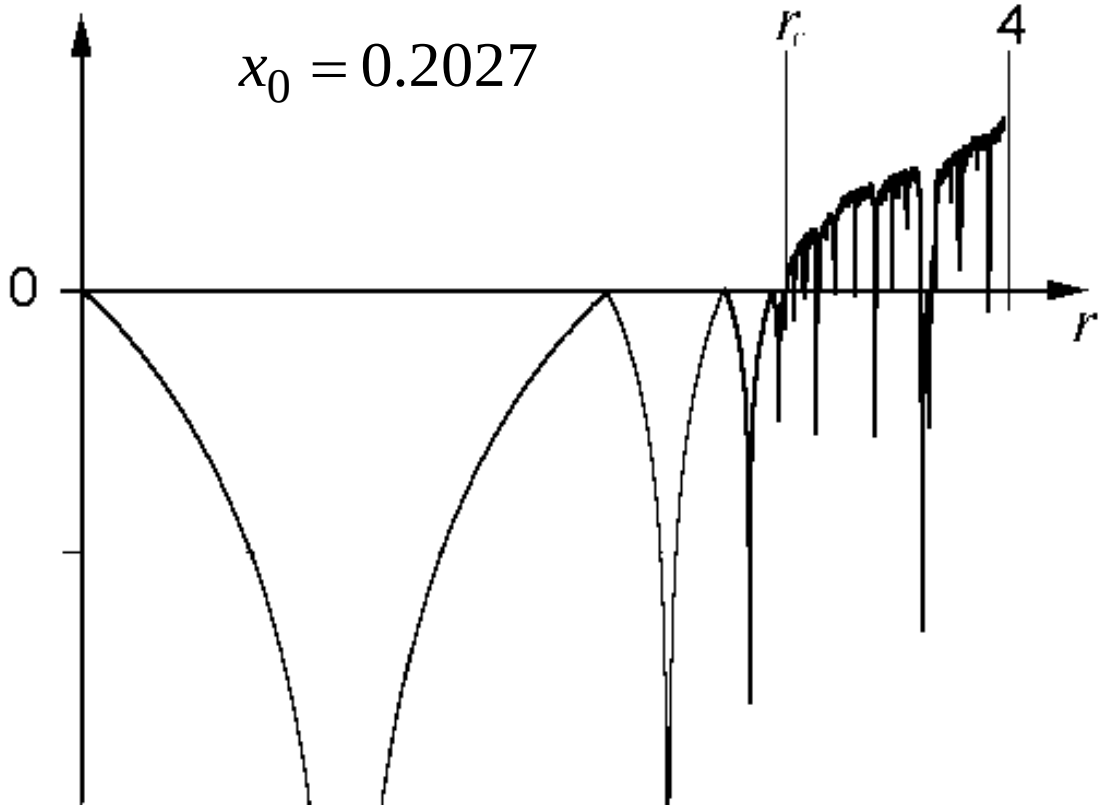
$$x'(t) = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t),$$

gdzie, np. $a = 0.2, b = 0.1, c = 10, \tau = 30.$

„Miara chaotyczności” systemu

„Miarą chaotyczności” systemu jest tzw. wykładnik Lapunova, który mierzy ekspotencjalne odchylenie trajektorii systemu chaotycznego wynikające z nieznacznego odchylenia (zaburzenia) warunków początkowych.

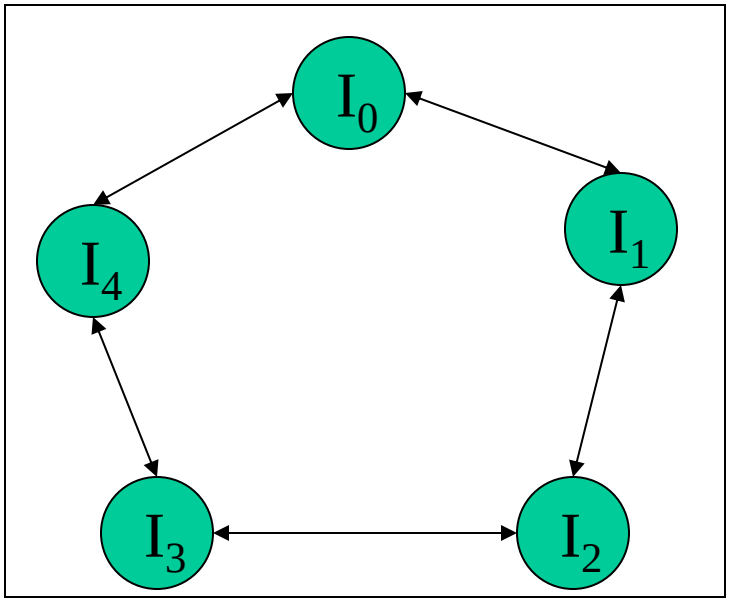
$$\lambda = \lim_{m\to\infty} \frac{1}{m} \sum_{k=1}^{m-1} \ln \left| \frac{dx_{k+1}}{dx_k} \right|$$

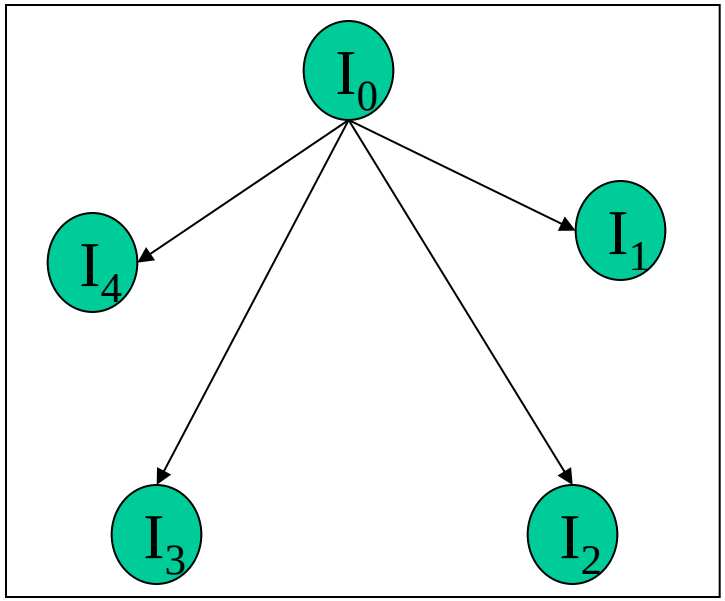


PSO - topologia roju

Dwie podstawowe topologie roju rozpatrywane w literaturze:

- pierścień (sąsiedztwo lokalne - 3 cząstki)
- gwiazda (sąsiedztwo globalne)





Inne warianty metody PSO

- Klasyczna (globalna) PSO:

$$v_{n+1} = v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

lub

$$v_{n+1} = w \cdot v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

- LPSO (Liniowa zmiana współczynnika bezwładności)

$$v_{n+1} = w_n \cdot v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

$$w_n = (w_{init} - w_{final}) \frac{it_{\max} - n}{it_{\max}} + w_{final}$$

Większe wartości na początku (większa eksploracja), stopniowo malejące („dopieszczanie” rozwiązania)

Inne warianty metody PSO

- RPSO (Losowa zmiana współczynnika bezwładności)

$$w_n = 0{,}5 + \frac{rand()}{2}$$

Wartość średnia = 0,75

- HPSO (Hybrid PSO)
  - Wykorzystanie idei PSO w połączeniu z ideą ewolucji (selekcji) z AE

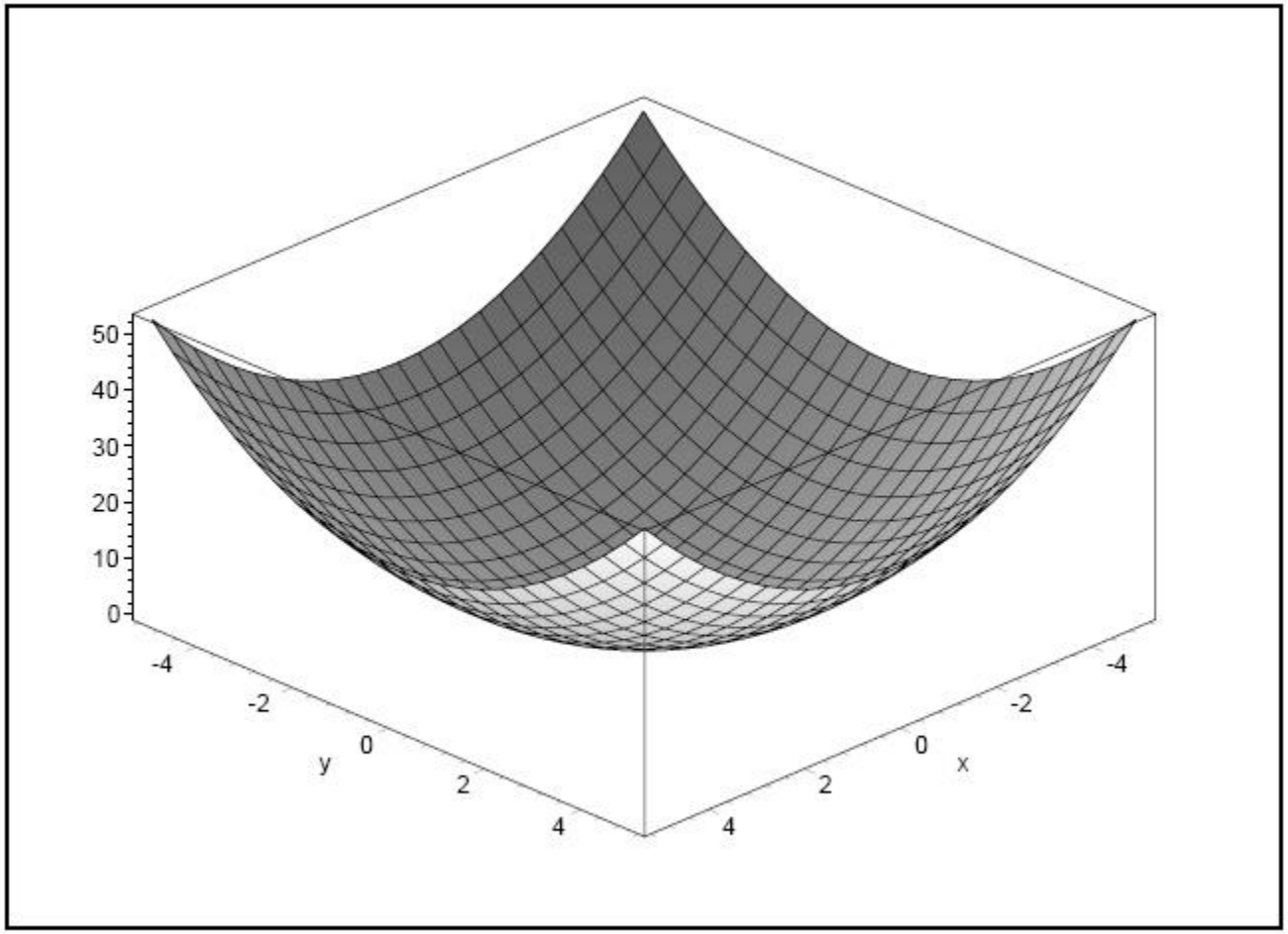
- Algorytm odkrywców/wynalazców (PSOO)
  - Idea: część populacji nie jest ograniczona przez g_best oraz l_best.
  - Większa swoboda w poszukiwaniu nowych atrakcyjnych rejonów. Jeżeli znajdą obiecujący obszar, to populacja może się w ten rejon przenieść.

Pierwsza funkcja de Jonga

$$f(x) = \sum_{i=1}^{n} x_i^2$$

$$x^* = (0,0,...,0)$$

$$f(x^*) = 0$$

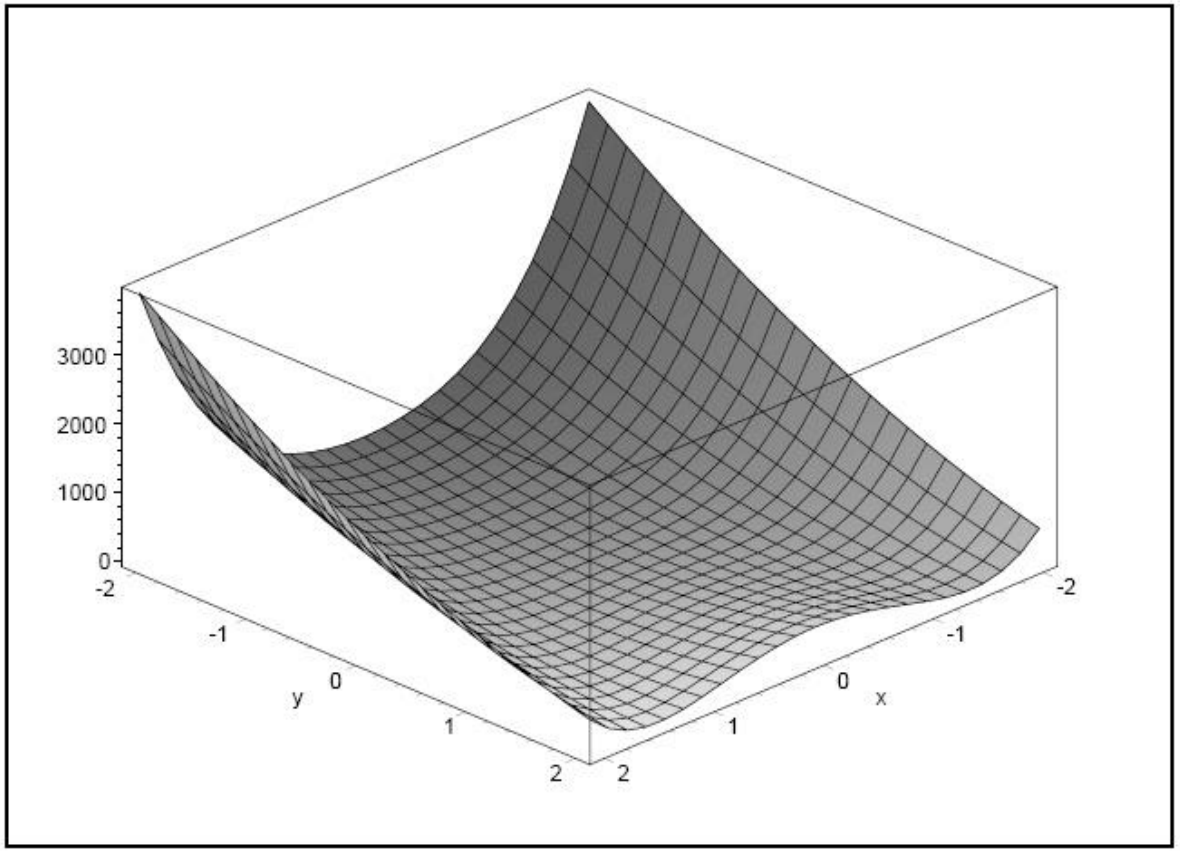


Dolina Rosenbrocka

$$f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right]$$

$$x^* = (1,1,\ldots,1)$$

$$f(x^*) = 0$$

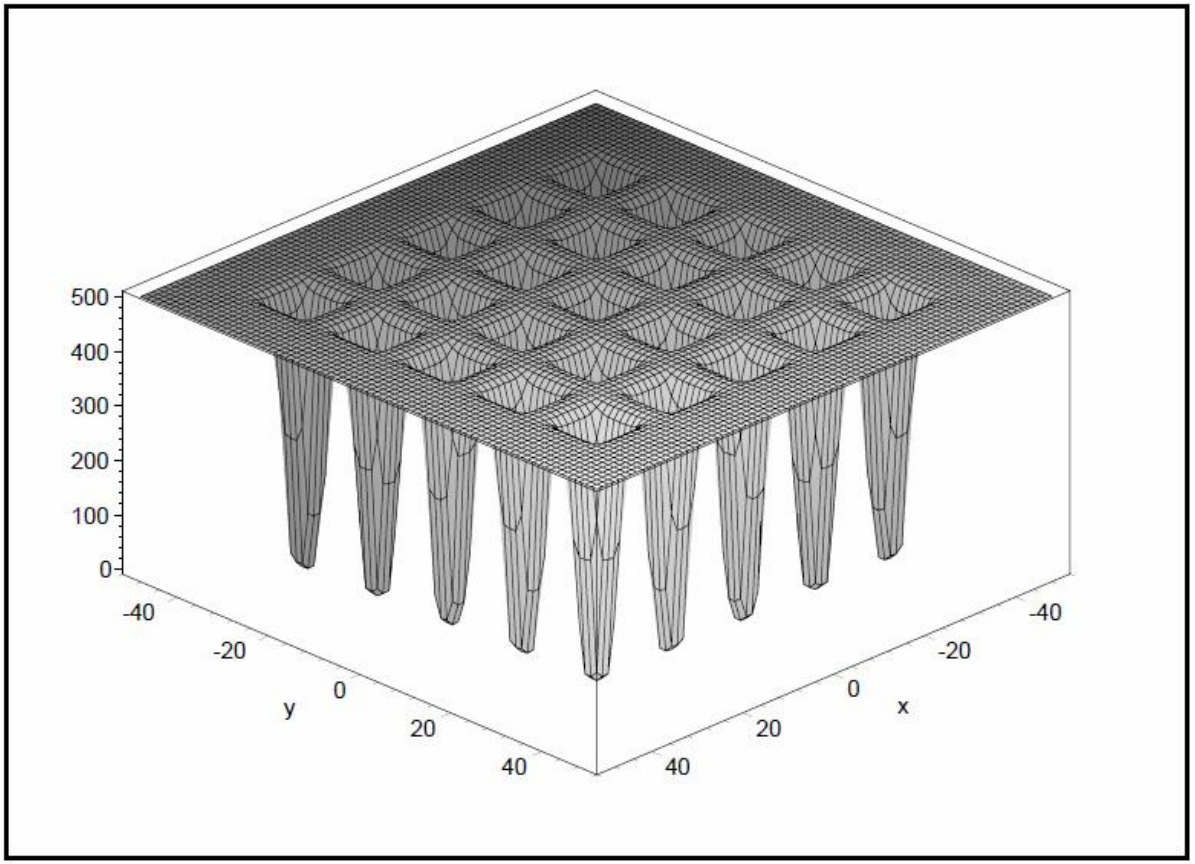


Piąta funkcja de Jonga

$$f(x) = \left(\sum_{i=-2}^{2} \sum_{j=-2}^{2} \left(5(i+2) + j + 3 + (x_1 - 16j)^6 + (x_2 - 16i)^6 \right)^{-1} \right)^{-1}$$

$$x^* = (-32, -32)$$

$$f(x^*) = 0{,}998$$

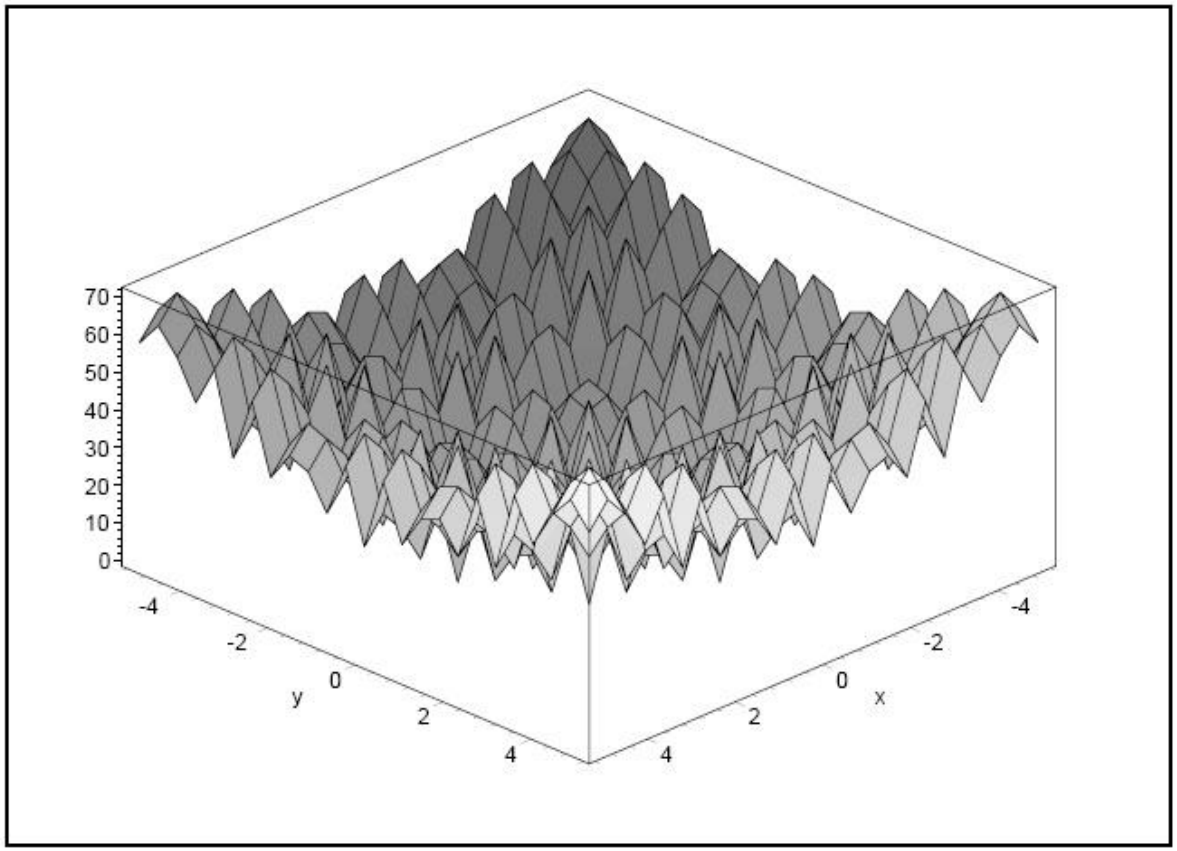


Funkcja Rastrigina

$$f(x) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\Pi x_i) \right]$$

$$x^* = (0,0,...,0)$$

$$f(x^*) = 0$$



Wyniki skuteczności PSOO / PSO

- Standardowa początkowa przestrzeń poszukiwań
- Początkowa przestrzeń poszukiwań nie zawiera minimum globalnego
- Początkowa przestrzeń przeszukiwań położona jest w bliskim sąsiedztwie minimum lokalnego

- Średnia ze 100 przebiegów algorytmu
- Każdy przebieg dotyczył 100 cząsteczek i trwał 100 iteracji
- Przebieg był uznawany za sukces jeżeli

$$\left\| x_{zn}^{*} - x^{*} \right\| < 0{,}05 \qquad \text{lub} \qquad \sqrt{f\left(x_{zn}^{*}\right) - f\left(x^{*}\right)} < 0{,}05$$

Wyniki skuteczności PSOO / PSO

	% sukcesu			wartość funkcji			średnia liczba iteracji		
FUNKCJA	**test 1**	**test 2**	**test 3**	**test 1**	**test 2**	**test 3**	**test 1**	**test 2**	**test 3**
Pierwsza funkcja de Jonga	100/ 100	100/ 100	-	0,00/ 0,00	0,00/ 0,00	-	27/22	32/26	-
Dolina Rosenbrocka	2/1	1/0	-	0,05/ 0,05	0,05/ 0,20	-	48/26	95/X	-
Piąta funkcja de Jonga	71/96	21/12	25/0	1,00/ 1,00	1,00/ 1,00	1,00/ 23,81	32/27	43/49	66/X
Funkcja Rastrigina	4/12	1/0	0/0	0,11/ 0,01	0,13/ 1,01	1,08/ 4,14	30/41	21/X	X/X

- Skuteczność różna dla różnych funkcji (dobór parametrów?)
- Bliskie sąsiedztwo minimum lokalnego ewidentnie przeszkadza (test 3)
- Wskazana jest inicjalizacja w obszarze zawierającym minimum globalne (testy 1 i 2)

Inne warianty metody PSO

- PSOwPPE – PSO with Particle Performance Evaluation

$$v_{n+1} = w \cdot v_n + c_1 rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

$$v_{n+1} = w \cdot v_n + c_1() \cdot rand() \cdot (p_{best} - x_n) + c_2 rand() \cdot (g_{best} - x_n)$$

$$c_1() = \begin{cases} 2 \\ 0{,}05 \end{cases}$$

zależnie od tego czy ***gbest*** była/nie była aktualizowana przez daną cząstkę w okresie ostatnich k iteracji

Większe znaczenie przypisywane jest jednostkom „aktywnie poprawiającym" rozwiązanie globalne, ponieważ to ***gbest*** stanowi „współdzieloną wiedzę w ramach roju" dotyczącą poszukiwanego rozwiązania

Algorytm PSO dla problemów z ograniczeniami

Dla każdej cząstki

Zainicjuj ją losowo ale tak by spełniała ograniczenia problemu

Do

{

Dla każdej cząstki

{

Policz wartość dopasowania

Jeżeli wartość dopasowania jest większa od najlepszego dotychczasowego dopasowania (**pbest**) i cząstka jest w dopuszczalnej podprzestrzeni (spełnia ograniczenia) to ustaw **pbest** jako aktualną wartość dopasowania

}

Wybierz cząstkę z najlepszym dopasowaniem w całym roju jako **gbest**

Dla każdej cząstki

{

Wybierz cząstkę z największą wartością **pbest** spośród jej sąsiadów i ustaw ją jako **lbest**

Policz jej szybkość zgodnie z równaniem szybkości

Zmodyfikuj jej pozycję zgodnie z równaniem zmiany pozycji

}

}

Dopóki nie przekroczono maksymalnej liczby iteracji lub błąd nie osiągnął zamierzonego minimum

Przykład: rekomendacja filmów

- Baza filmów wraz z profilami użytkowników
- Dobór na podstawie danych personalnych, oceny obejrzanych filmów oraz porównań z profilami innych użytkowników
- Baza MovieLens dostępna pod http://www.grouplens.org
  - 100,000 ocen (1-5) przez 943 użytkowników dla 1682 filmów
  - Każdy użytkownik ocenił co najmniej 20 filmów
- Pojedynczy wektor: 22 wymiary: ocena filmu, dane personalne (wiek, płeć, zawód), informacje o gatunkach filmów (18 cech)
- Odpowiednio dobrana funkcja odległości pomiędzy profilami.
- PSO z mutacją osobników

(D)CVRP(wTJ)

- $G=(V,E)$; $V=\{v_0, v_1, \ldots, v_n\}$
- $E=\{(v_i, v_j) \mid v_i, v_j \in V, i < j\}$
- v_0 – *centralny magazyn*,
- q_i – *zapotrzebowanie klienta w lokalizacji* v_i ;$(q_i << Q)$
- d_{ij} – *zmienia się w funkcji korka*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \sum_{k=1}^{n} x_{ij}^{k}$$

$$x_{ij}^{k} = \begin{cases} 1, & (v_i, v_j) \text{ jest fragmentem drogi } k-\text{tej} \\ 0, & \text{w p.p.} \end{cases}$$

Porównanie PSO z algorytmem mrówkowym

- Podobieństwa:
  - metoda heurystyczna,
  - iteracyjny schemat działania,
  - na zróżnicowanie populacji ma wpływ kilka parametrów, których dobór ma kluczowe znaczenie.

Porównanie PSO z algorytmem mrówkowym

- Różnice:
  - dane wejściowe - populacja losowych (PSO) / nielosowych (ANT) osobników,
  - cząstki posiadają (PSO) / nie posiadają (ANT) pamięci o najlepszym dotychczasowym położeniu (rozwiązaniu),
  - wymiana informacji poprzez interakcje z g/l best (PSO) vs wiedza całego zbiorowiska (ANT).

Porównanie PSO z algorytmem ewolucyjnym

- Podobieństwa:
  - metoda heurystyczna;
  - iteracyjny schemat działania;
  - dane wejściowe - populacja losowych osobników;
  - na zróżnicowanie populacji ma wpływ kilka parametrów, których dobór ma kluczowe znaczenie.

Porównanie PSO z algorytmem ewolucyjnym

- Różnice:
  - brak ewolucyjnego operatora mutacji;
  - w AE następowała wymiana informacji poprzez krzyżowanie (każdy z każdym), w PSO schemat (każdy z jednym: *gbest* lub *lbest*);
  - cząstki posiadają pamięć o najlepszym dotychczasowym położeniu (rozwiązaniu).

Pytania ?