

# Algorytm FA

- Metaheurystyczna metoda poszukiwań (Xin-She Yang, 2008), inspirowana przez:
  - zachowania społeczne
  - zjawisko bioluminescencji robaczków świecących (świecących)
- Zastosowanie w zadaniach optymalizacji z ograniczeniami dla ciągłych dziedzin poszukiwań

# Algorytm FA

- Świetliki:

- latające chrząszcze
- bioluminescencja



do przyciągania ofiar i innych osobników

- „zimne” źródło światła
- chemicznie produkowane światło z dolnej części brzucha (żółte, zielone lub jasnoczerwone, o długości fali od 510 do 670 nm)

# Algorytm FA

- Wszystkie świetliki są obupłciowe - jeden świetlika będzie przyciągał inne świetliki
- Atrakcyjność jest proporcjonalna do ich jasności
- Świetlik o mniejszej jasności będzie przyciągany (przesuwany) do osobnika o większej jasności

# Algorytm FA

- Jasność może zmniejszać się ze zwiększającą się odległością
- Jeżeli nie ma żadnych osobników o jaśniejszym świetle wtedy porusza się losowo
- Jasność światła powinna być powiązana z funkcją kryterialną

# Algorytm FA

- Rozwiązanie problemu optymalizacji może być postrzegane jako agent (światlik)
- Sztuczny światlik „świeci” proporcjonalnie do jego jakości w rozważanym problemie optymalizacji
- Jaśniejsze światliki przyciągają partnerów (bez względu na ich płeć) - przestrzeń poszukiwań jest bardziej efektywnie badana

# Algorytm FA

- Mechanizmy komunikacji poprzez świecące i migające świetliki i ich synchronizacja została naśladowana skutecznie, np.:
  - technikach projektowania sieci bezprzewodowych
  - modelowania dynamiki cenowej na rynku
  - robotach mobilnych

# Algorytm FA

- Ciągłe z ograniczeniami zadanie optymalizacji

$$f(\mathbf{x}^*) = \min_{\mathbf{x} \in S} f(\mathbf{x})$$

gdzie

$f(\mathbf{x}) \in \mathbf{R}$  - funkcja celu (funkcja kosztów)

$$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T \in \mathbf{S} \subseteq \mathbf{R}^n$$

- wektor poszukiwanych parametrów w przestrzeni  $\mathbf{S}$

# Algorytm FA

- Rój  $m$  agentów (światlików)
  - rozwiązanie  $\mathbf{x}_i$   $i$ -tego światlika
  - wartość funkcja kosztów  $f(\mathbf{x}_i)$  światlika  $i$
- Funkcja atrakcyjności:  $\beta^i = \beta_0 e^{-\gamma r_{ij}^2}$ 
  - odległość pomiędzy  $i$ -tym i  $j$ -tym agentem  $r_{ij}$
  - maksymalna wartość atrakcyjności  $\beta_0$
  - współczynnik absorpcji  $\gamma$
- Intensywność  $I_i$  - odwrotność funkcji kosztów



# Algorytm FA

- Każdy światek zmienia swoją pozycję w sposób iteracyjny biorąc pod uwagę:
  - atrakcyjność innych członków roju o wyższej jasności  
$$I_j > I_i, \quad \forall_{j \neq i} j = 1, 2, \dots, m$$
  - ustalony losowy wektor przesunięcia  $u_j$
- Jeżeli nie znajdzie się w pewnym sąsiedztwie danego agenta o większej jasności wówczas wykonujemy losowe przesunięcie

# Algorytm FA

- Ruch świetlika  $i$  jest taki, że jest przyciągany do bardziej atrakcyjnego świetlika  $j$

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \mathbf{u}_i$$

gdzie:

$\mathbf{u}_i$  - wektor liczb losowych wygenerowanych z rozkładem równomiernym z przedziału  $[0,1]$

# Schemat Algorytmu FA

**Procedura FA**

**begin**

*Wylosuj populację m światełek*

**repeat**

**for  $i=1$  to  $m$  do**

**for  $j=1$  to  $m$  do**

*if  $f(\mathbf{x}_j) > f(\mathbf{x}_i)$  then      /\* przesun światełko  $i$  w kierunku  $j$  \*/*

*Oblicz odległość:  $r_{ji}$*

*Zmodyfikuj atrakcyjność:  $\beta^i = \beta_0 e^{-\gamma r_{ij}^2}$*

*Wygeneruj losowy wektor:  $\mathbf{u}_i$*

*Utwórz nowe rozwiązanie:  $\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \mathbf{u}_i$*

**end**

**end**

*Wygeneruj losowy wektor:  $\mathbf{u}_k$*

*Najlepsze światełko jest przesuwane wg:  $\mathbf{x}_k = \mathbf{x}_k + \mathbf{u}_k$*

**until** *warunek zakończenia niespełniony*

**end**

# Techniczne szczegóły FA

- FA wykorzystuje synergiczne lokalne wyszukiwanie
- Każdy członek rój bada przestrzeń problemu, biorąc pod uwagę wyniki uzyskane przez innych
- Wpływ innych rozwiązań jest kontrolowany przez wartość atrakcyjności

# Techniczne szczegóły FA

- $\beta_0$  atrakcyjność dla  $r_j = 0$  (dwa świetliki znajdują się w tym samym punkcie przestrzeni poszukiwań  $\mathcal{S}$ )
- W ogólności  $\beta_0 \in [0,1]$  :
  - $\beta_0 = 0$  (brak współpracy świetlików – losowe poszukiwanie )
  - $\beta_0 = 1$  (poszukiwanie lokalnej współpracy z jaśniejszym świetlikiem – sąsiedztwo)

# Techniczne szczegóły FA

- $\gamma$  - zróżnicowanie atrakcyjności ze wzrostem odległości od przekazywanych informacji przez inne świetliki
- Zwykle  $\gamma \in [0,10]$ :
  - $\gamma = 0$  (brak zmiany lub stałej atrakcyjności)
  - $\gamma \rightarrow \infty$  (atrakcyjność bliska zeru co może powodować bardzo losowe przeszukiwanie przestrzeni)

# Techniczne szczegóły FA

- Dostosowanie współczynnika absorpcji - w oparciu o odległość w przestrzeni poszukiwań
- Proponuje się następujące metody ustalania parametrów

$$\gamma = \frac{\gamma_0}{r_{\max}} \quad \text{or} \quad \gamma = \frac{\gamma_0}{r_{\max}^2}, \quad \gamma \in [0,1]$$

$$r_{\max} = \max \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}$$

# Techniczne szczegóły FA

- Ograniczenia na wektor  $\mathbf{u}_i$ :
  - $\min \mathbf{u}_i = -0.5\alpha$  i  $\max \mathbf{u}_i = 0.5\alpha$
- Każda  $k$ -ta współrzędna wektora  $\mathbf{u}_i$  jest generowana według następującej formuły:

$$u_i^k = \alpha(\varepsilon - 0.5)$$

gdzie

$\alpha \in [0,1]$  - parametr algorytmu

$\varepsilon \in [0,1]$  - liczba wylosowana z rozkładem  
równomiernym



# Warianty algorytmu FA

- Dyskretny wersja algorytmu (DFA):
  - poszukiwane parametry kodowane binarnie
  - mogą efektywnie rozwiązywać zadania NP-trudne
  - przewyższają istniejące algorytmy, takie jak algorytm kolonii mrówek
  - problemy segmentacji obrazów ( FA - o wiele bardziej efektywna niż metody Otsu)

# Warianty algorytmu FA

- Wielokryterialny FA:
  - dla szerokiego zakresu problemów (np. polioptymalizacja problemu wysyłki obciążenia)
- Lagrang'a FA:
  - w problemach optymalizacji systemów jednostek mocy
- Chaotyczny FA
- Hybrydowe FA

# Zastosowania FA

- Cyfrowa kompresja obrazów
- Optymalizacja wektorów własnych
- Detekcja uszkodzeń i rozróżnianie
- Projektowanie anten
- Planowanie i TSP
- Semantyczna kompozycja stron WWW
- Clustering
- Inne dynamiczne problemy

# Optymalizacja rojem świetlików (GSO)

- Algorytm inteligencji roju rozwijany w oparciu o zachowania społeczne nieskrzydlatych świetlików (ang. **glowworms** )
- Wzorzec zachowania świetlików zastosowany w algorytmie jest ich widoczną zdolnością do zmiany intensywności emisji **lucyferyny** wydając tym samym blaski o różnym natężeniu



# Optymalizacja rojem świetlików (GSO)

- Algorytm GSO intensywność świecenia świetlika jest w przybliżeniu proporcjonalna do wartości funkcji kryterialnej
- Świetliki o jaśniejszej intensywności przyciągają inne świetliki o niższej emisji światła
- Dynamiczny zakres decyzji (percepcji) dotyczący wystarczającej liczby sąsiadów

# Optymalizacja rojem świetlików (GSO)

- W dróżnieniu od FA:
  - Nieograniczona “wystarczająca liczba” sąsiadów
  - nieograniczona percepcja wg odległości
- „Ograniczenia poznawcze” pozwalają rojowi świetlików na dzielenie się na podgrupy i osiągnięcie maksimum lub minimum
- Możliwość znalezienia wszystkich lokalnych optimum

# Optymalizacja rojem świetlików (GSO)

- Algorytm GSO zaproponowali K.N. Krishnanand i D. Ghose (2005)
- Algorytm GSO skuteczne narzędzie optymalizacji w różnych aplikacjach zaprezentowanych w literaturze

# Optymalizacja rojem świetlików (GSO)

- Sąsiedztwo reprezentuje lokalną przestrzeń decyzji
- Zakres zmiennej sąsiedztwa  $r_d^i$  jest związany z radialnym zakresem  $r_s$  ( $0 < r_d^i \leq r_s$ )
- Świetlik  $i$  bierze pod uwagę świetlika  $j$  jako sąsiada, jeżeli  $j$  jest wewnątrz obszaru sąsiedztwa rozwiązania  $i$  oraz poziom lucyferyny jest wyższy dla  $j$ -ego niż  $i$ -tego



# Optymalizacja rojem świetlików (GSO)

- Dziedzina zmiennych decyzyjnych umożliwia:
  - selekcyjne wybieranie interaktywnego sąsiada
  - zapewnienie interaktywnych reakcji dla danego sąsiedztwa
  - każdy osobnik z określonym wyższym poziomem emisji światła wewnątrz sąsiedztwa przyciąga inne
- Agentci w GSO posiadają tylko informację o swoim sąsiedztwie dla wyboru kolejnego położenia

# Optymalizacja rojem świetlików (GSO)

- Każdy świetlik wybiera, używając stochastycznego mechanizmu, sąsiada, który posiada wyższą wartość lucyferyny i przemieszcza się w jego kierunku
- Takie przemieszczenia (oparte na informacjach lokalnych i selektywnej interakcji sąsiada) pozwalają na podział w rozłączne podgrupy które poruszają się w kierunku innych lokalnych optimów funkcji celu

# Optymalizacja rojem światełków (GSO)

- GSO rozpoczyna od losowego rozmieszczenia populacji  $n$  agentów w przestrzeni poszukiwań
- Początkowo wszyscy agenci posiadają wielkość lucyferyny równą 0
- Każdy cykl GSO składa się z trzech głównych kroków:
  - Faza modyfikacji lucyferyny
  - Faza ruchu w oparciu o zasady przejścia
  - Faza modyfikacji zakresu sąsiedztwa

# Optymalizacja rojem świetlików (GSO)

## *Faza modyfikacji lucyferyny*

- Uwarunkowana wartością funkcji kryterialnej dla danego położenia świetlika
- Każdy świetlik dodaje do wcześniejszej wartości lucyferyny, wielkość lucyferyny, która jest proporcjonalna do wartości funkcji celu (przystosowania) dla danej aktualnej pozycji
- Pewna wartość lucyferyny jest odejmowana w celu zasymulowania mechanizmu rozkładu tej substancji w czasie

# Optymalizacja rojem światełków (GSO)

- Modyfikacja lucyferyny:

$$\ell_i(t+1) = (1-\rho)\ell_i(t) + \gamma J(\mathbf{x}_i(t+1))$$

gdzie

$\ell_i(t)$  - poziom lucyferyny dla światełka  $i$  w chwili czasu  $t$

$\rho$  - stała rozkładu lucyferyny ( $0 < \rho < 1$ )

$\gamma$  - stała wzmocnienia lucyferyny

$J(\mathbf{x}_i(t))$  - wartość funkcji celu dla agenta  $i$  w chwili czasu  $t$

# Optymalizacja rojem światełków (GSO)

## *Faza ruchu*

- Prawdopodobieństwo przemieszczenia agenta  $i$  w kierunku sąsiada  $j$

$$p_{ij}(t) = \frac{\ell_j(t) - \ell_i(t)}{\sum_{k \in N_i(t)} \ell_k(t) - \ell_i(t)}$$

$$N_i(t) = \{j : d_{ij}(t) < r_d^i(t); \ell_i(t) < \ell_j(t)\}$$

gdzie

$N_i(t)$  - zbiór sąsiadów światełka  $i$

$d_{ij}(t)$  - odległość pomiędzy światełkiem  $i$  oraz  $j$

# Optymalizacja rojem świateł (GSO)

- Niech świateł  $i$  wybierze świateł  $j \in N_i(t)$  z prawdopodobieństwem  $p_{ij}(t)$
- Dyskretno-czasowy model przemieszczania świateł:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + s \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|}$$

gdzie

$\mathbf{x}_i \in \mathbf{R}^m$  - pozycja świateł  $i$

$s$  - rozmiar kroku przemieszczenia ( $>0$ )

# Optymalizacja rojem światełków (GSO)

## *Modyfikacja zakresu sąsiedztwa*

- Każdy agent posiada przypisane sąsiedztwo zdefiniowane przez dynamiczny radialny zasięg  $r_d^i$  ( $0 < r_d^i \leq r_s$ )
- Stały zakres sąsiedztwa nie jest używany
- Liczba lokalnych optimów jest pewną funkcją zasięgu sąsiedztwa



# Optymalizacja rojem świetlików (GSO)

- Jeżeli zakres percepcji agenta odnosi się o całej dziedzinie poszukiwań, wówczas wszyscy agenci zbiegają się do jednego globalnego punktu (localne optima są ignorowane)
- Informacja a priori o liczbie lokalnym optimów nie jest znana
- Trudno ustalić stałą wartość zakresu sąsiedztwa w przypadku stosowania algorytmu dla różnych zadań

# Optymalizacja rojem światełków (GSO)

- Wybrany zakres sąsiedztwa może dawać lepsze rezultaty, jeżeli minimalna odległość pomiędzy lokalnymi optimami jest większa niż  $r_d$
- GSO stosuje adaptacyjny zakres sąsiedztwa w celu wykrycia obecności wielu lokalnych optimów

# Optymalizacja rojem świateł (GSO)

- Niech  $r_0$  oznacza początkowy zakres otoczenia każdego świateł ( $r_d^i(0) = r_0, \forall i$ )
- Adaptacja zakresu sąsiedztwa

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\right\}\right\}$$

gdzie

$\beta$  - pewna stała (parameter)

$n_t$  - parametr używany do sterowania liczbą świateł w sąsiedztwie

# Optymalizacja rojem świateł (GSO)

- Wielkości  $\rho, \beta, \gamma, n_t, s, \ell_0$  są parametrami algorytmu ustalonymi na podstawie licznie przeprowadzonych eksperymentów

$\rho$	$\beta$	$\gamma$	$s$	$n_t$	$\ell_0$
0.4	0.08	0.6	0.03	5	5

# Optymalizacja rojem światełków (GSO)

## **Efekt „żabięgo skoku”**

- Zgodnie z mechanizmem GSO w danej iteracji światełko z maksymalną lucyferyną pozostaje nieruchome
- Powyższa własność może doprowadzić do zatrzymania poszukiwań, w której wszystkie światełki leżące w pobliżu lokalnego optimum zbiegają się do światełka znajdującego się najbliżej niego

# Optymalizacja rojem świetlików (GSO)

## **Efekt „żabiego skoku”**

- Ruch agenta jest ograniczony do regionu wypukłej powierzchni
- Wszystkie świetliki zbiegają się do świetlika, który osiągnął maksymalną ilość lucyferyny podczas jego ruchów na wypukłej powierzchni
- W rezultacie wszystkie świetliki są „uwięzione” do lokalnego optimum

# Optymalizacja rojem świetlików (GSO)

## Efekt „żabiego skoku”

- Dyskretny charakter zasad poruszania się agenta automatycznie zapobiega temu problemowi
- Podczas fazy ruchu każdy agent przesuwa się o określony skończoną liczbą kroków  $s$
- Kiedy odległość pomiędzy agentem  $i$  osiagającym sąsiada  $j$  jest mniejsza niż  $s$ , agent  $i$  wykonuje „żabi skok” przeskakując pozycję agenta  $j$  stając się liderem  $j$

# Optymalizacja rojem świetlików (GSO)

## Efekt „żabiego skoku”

- W kolejnej iteracji, agent  $i$  staje się stacjonarnym rozwiązaniem z położeniem, które daje mu pozycję „lidera”
- Proces zamiany ról pomiędzy agentami  $i$  oraz  $j$  powtarza się powodując lokalne poszukiwania poprzez parę agentów jako pojedynczego kierunku wspinania się



# Optymalizacja rojem świetlików (GSO)

## **Efekt „żabięgo skoku”**

- A grupa świetlików używa tej samej zasady przy przeprowadzaniu ulepszonego lokalnego przeszukiwania i ewentualnego zbiegania się do optimów lokalnych
- Efekt „żabięgo skoku” jest obserwowany w licznych symulacjach

# Schemat GSO

## Procedura GSO

**begin**

*Generuj losowo populację  $n$  świetlików;*

**repeat**

*Zmodyfikuj lucyferynę każdego świetlika:  $\ell_i(t+1) = (1-\rho)\ell_i(t) + \gamma J(\mathbf{x}_i(t+1))$*

*Przemieść każdego świetlika:  $N_i(t) = \{j : d_{ij}(t) < r_d^i(t); \ell_i(t) < \ell_j(t)\}$*

**for** każdego świetlika  $j \in N_i(t)$  **do:**

$$p_{ij}(t) = \frac{\ell_j(t) - \ell_i(t)}{\sum_{k \in N_i(t)} \ell_k(t) - \ell_i(t)}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + s \frac{\mathbf{x}_j(t) - \mathbf{x}_i(t)}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|}$$

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\}$$

**end**

**until** waunek zakończenia nie jest spełniony

**end**

# Wielomodalne funkcje testowe - (GSO)

- Funkcja „peaks”:

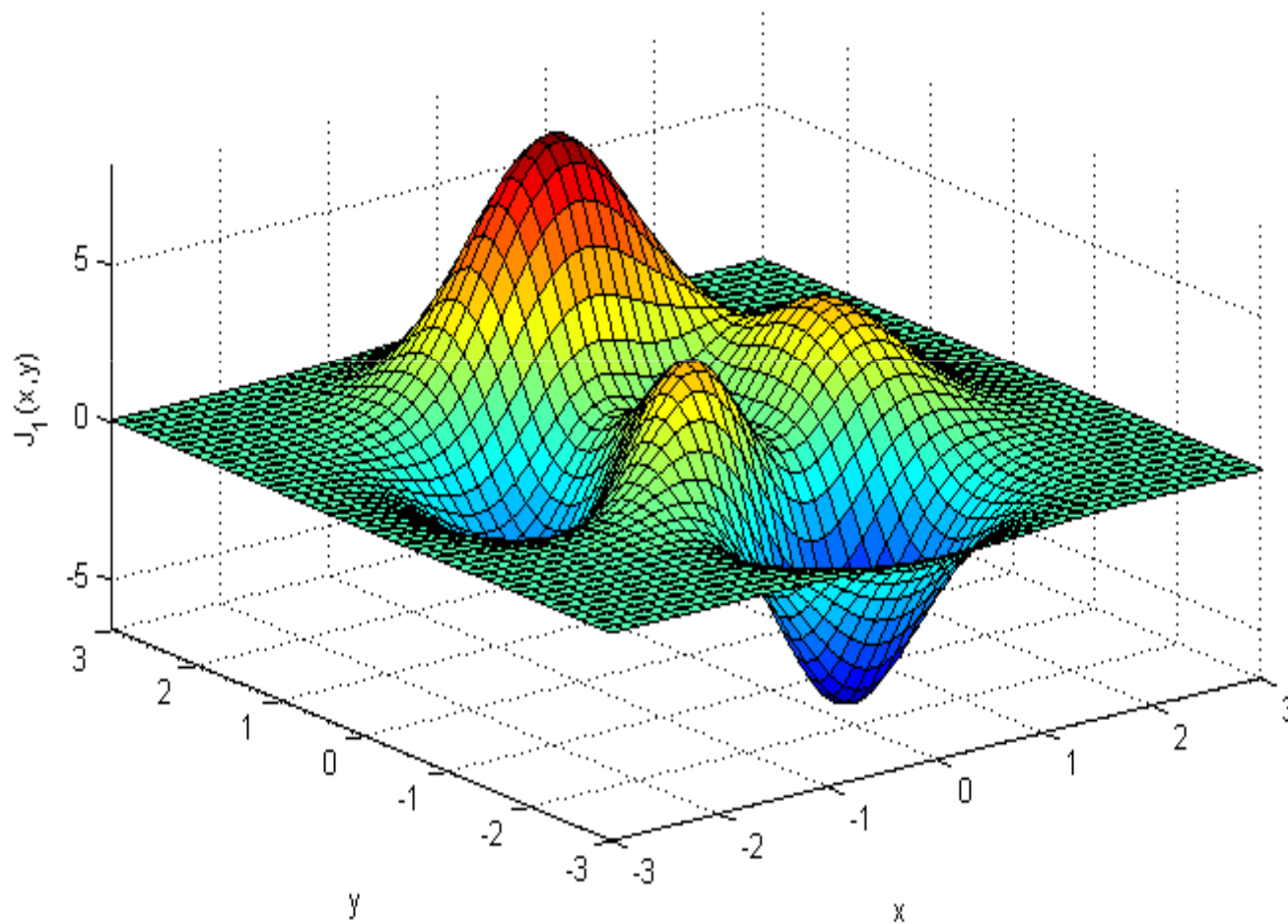
$$J(x, y) = 3(1-x)^2 e^{-[x^2+(y+1)^2]} + \\ -10\left(\frac{x}{5} - x^3 - y\right) e^{-(x^2+y^2)} - \frac{1}{3} e^{-[(x+1)^2+y^2]}$$

- Lokalne maksima położone w punkcie (0, 1.58), (-0.46, -0.63), and (1.28, 0) o różnej wartości funkcji celu

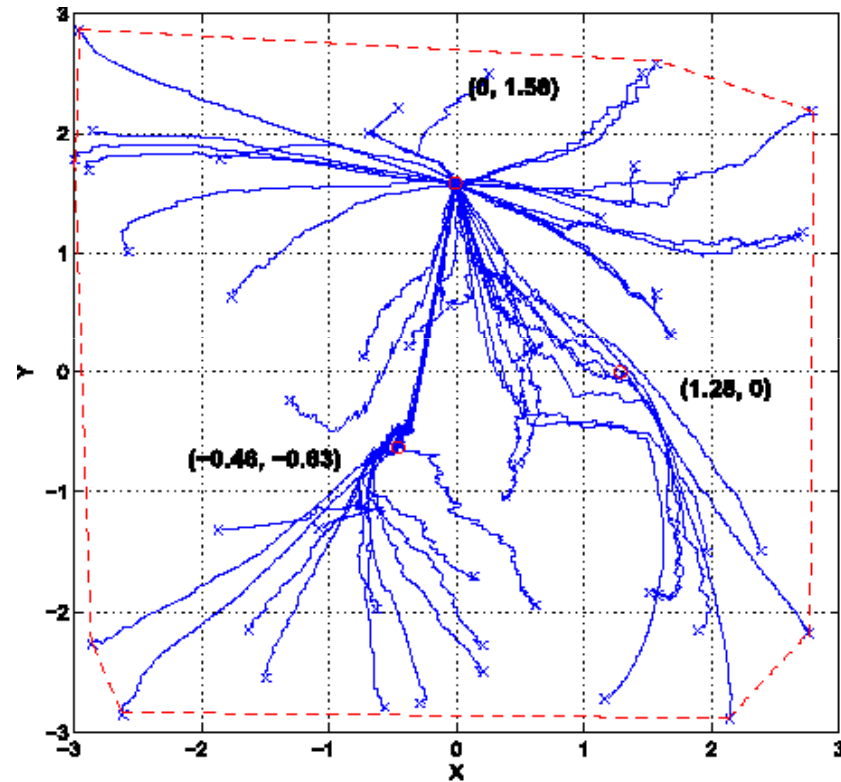
# Wielomodalne funkcje testowe - (GSO)

- Parametry symulacji:
  - Dziedzina parametrów:  $x, y \in [-3, 3]$
  - Liczba świetlików w populacji:  $n=50$
  - Radialne zakresy percepcji:  $r_s \in \{3, 2, 1.8, 1.5\}$
  - Liczba iteracji: 200

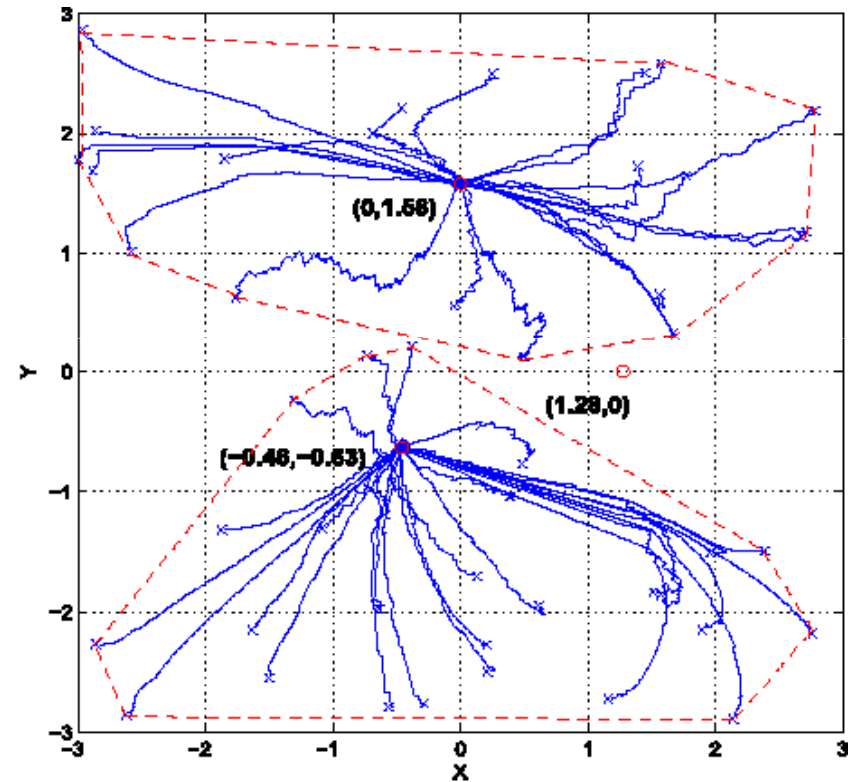
# Wielomodalne funkcje testowe - (GSO)



# Wielomodalne funkcje testowe - (GSO)

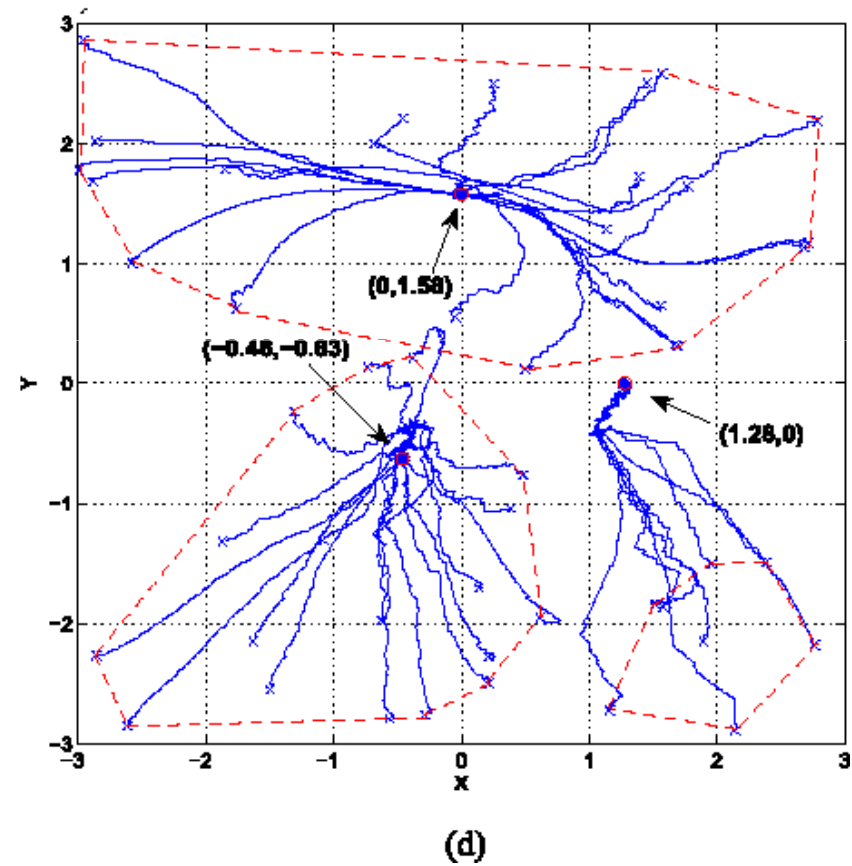
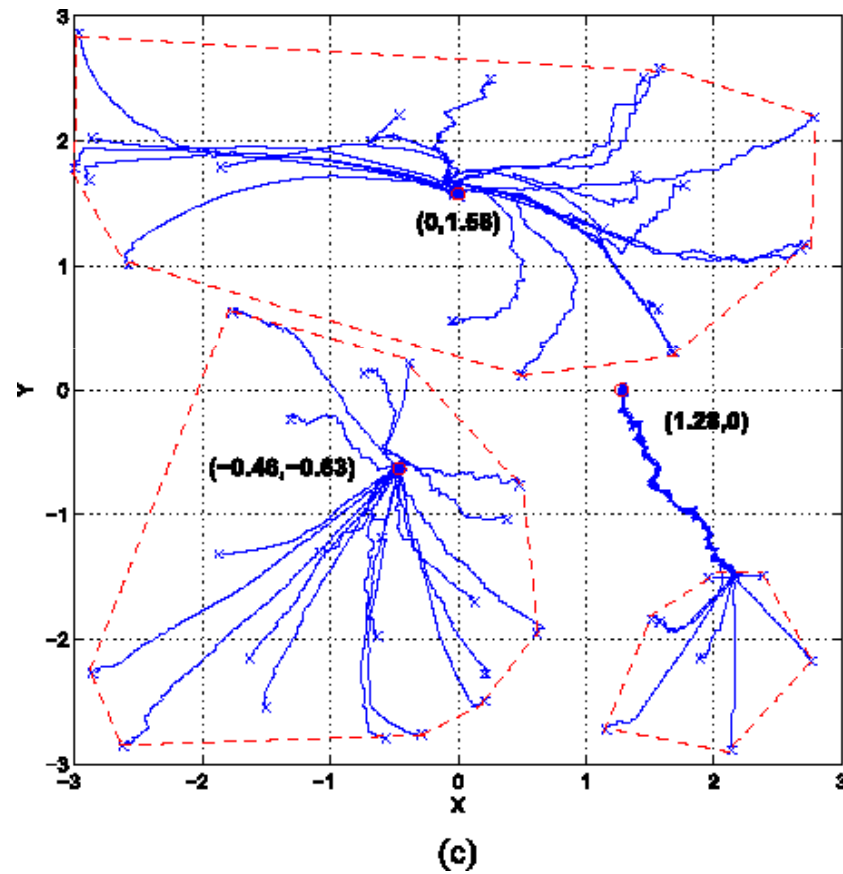


(a)



(b)

# Wielomodalne funkcje testowe - (GSO)



# Wielomodalne funkcje testowe - (GSO)

- Rastrigin's function:

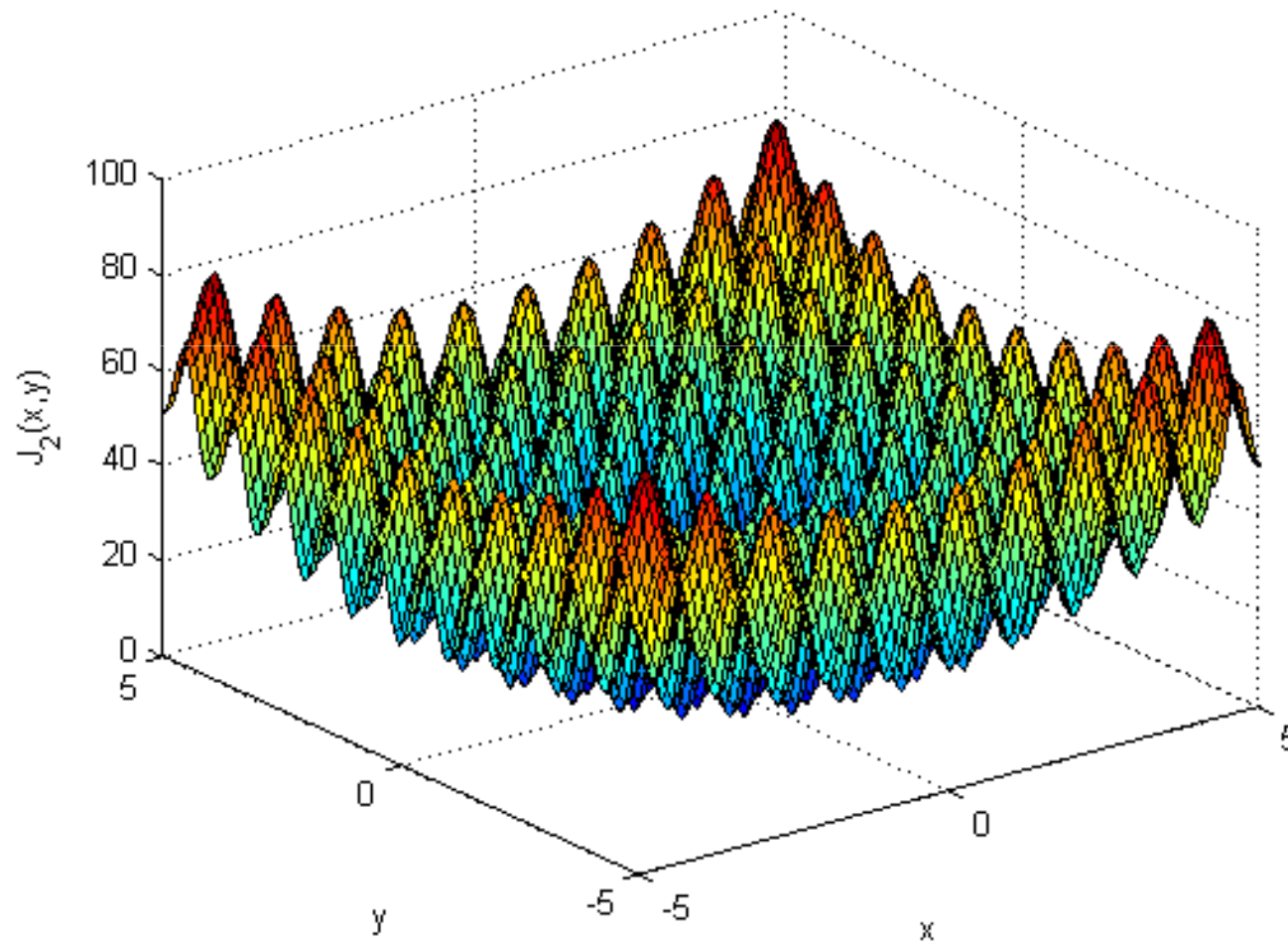
$$J(x, y) = 20 + (x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y))$$



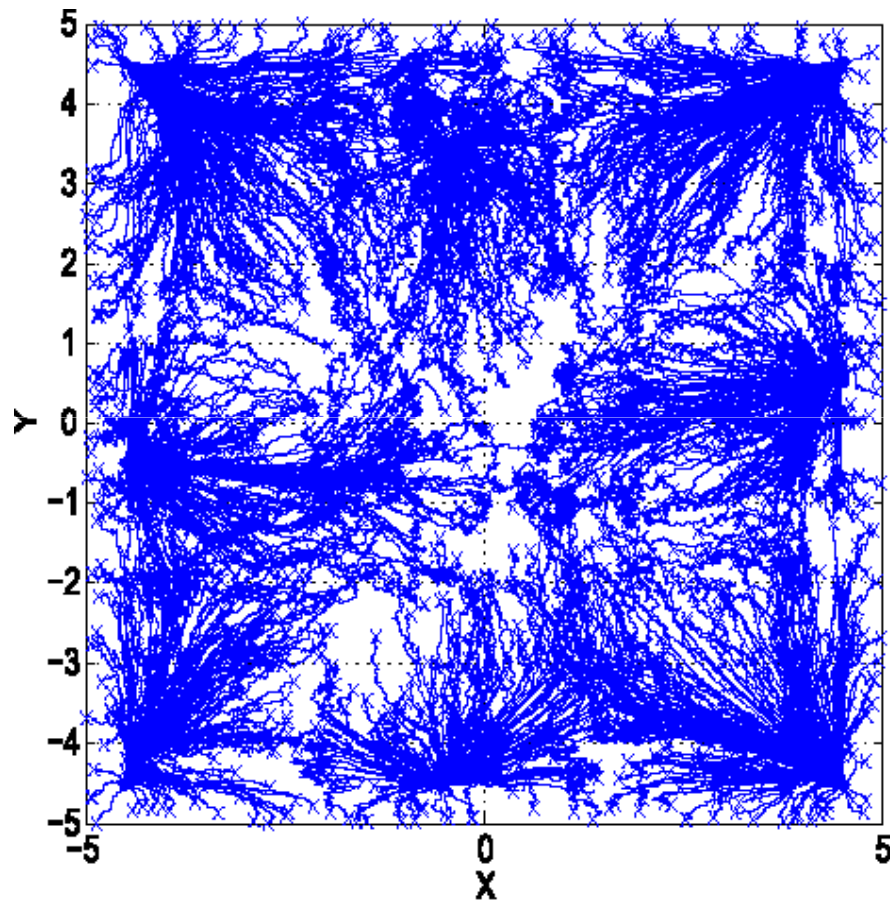
# Wielomodalne funkcje testowe - (GSO)

- Parametry symulacji:
  - Dziedzina parametrów:  $x, y \in [-5, 5]$
  - Liczba światełek w populacji:  $n=1500$
  - Radialny zakres percepcji  $r_s = 2$
  - Number of the iteration: 500

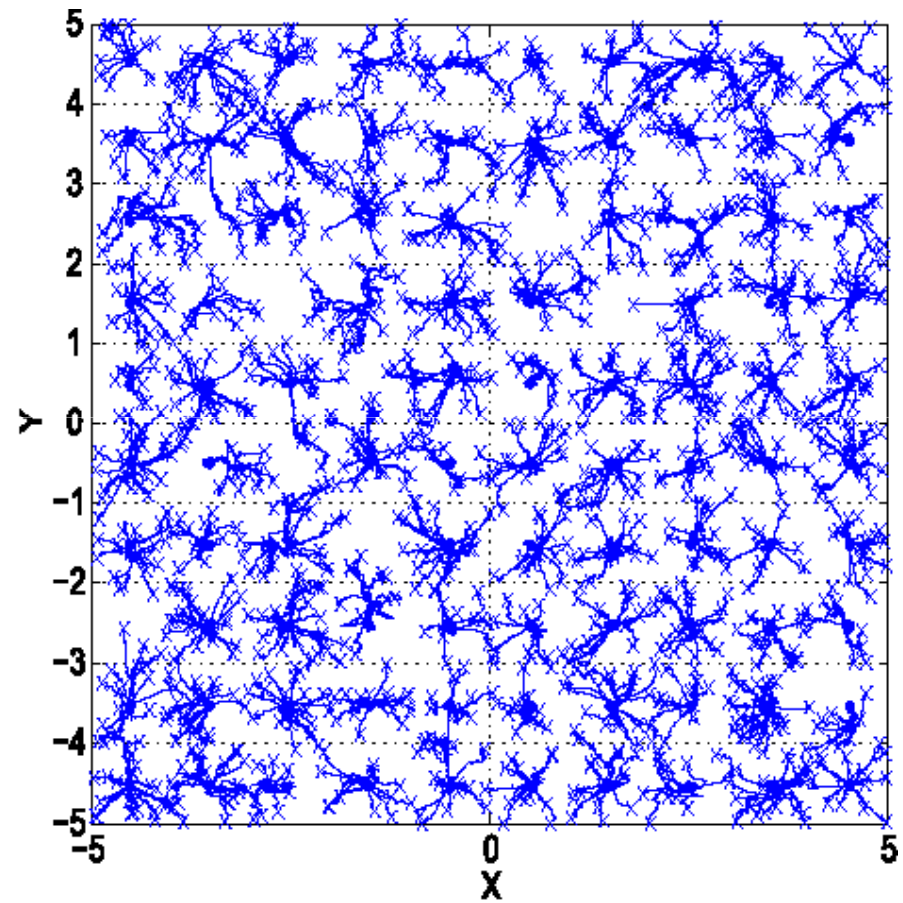
# Wielomodalne funkcje testowe - (GSO)



# Wielomodalne funkcje testowe - (GSO)



(a)



(b)

# Algorytm Pszczół (BA)

- Populacyjny algorytm poszukiwań opracowany przez Pham DT. w 2005r.
- Naśladuje zachowanie żerowania rojów miodnych pszczoł
- Algorytm wykonuje przeszukiwania lokalne w połączeniu z mechanizmami losowymi
- Stosowany w obu rodzajach optymalizacji: kombinatorycznej oraz funkcjonalnej

# Biologiczne tło algorytmu BA

- Kolonia pszczół miodnych:
  - może poruszać się na duże odległości (> 14 km) w wielu kierunkach jednocześnie oraz korzystać z wielu źródeł żywności
  - rozwija się, gromadząc żywność
- Tereny kwiatów zawierające znaczne ilości nektaru i pyłku gromadzone są przy mniejszym wysiłku wielu pszczół (tereny o mniejszej ilości pożywienia są eksploatowane przez kilka pszczół)

# Biologiczne tło algorytmu BA

- Proces poszukiwania pożywienia rozpoczyna się od wystania z kolonii pszczół zwanych „zwiadowcami” w obiecujące tereny kwiatowe
- Pszczoły zwiadowcy poruszają się losowo z jednego miejsca na drugie
- Podczas sezonu zbierania pożywienia, kolonia pszczół kontynuuje eksplorację utrzymując pewien procent osobników do przeprowadzania zwiadu

# Biologiczne tło algorytmu BA

- Po powrocie do ula, pszczoły wykonujące zwiad są oceniane
- Powyżej pewnego progu jakości (mierzonej jako połączenie niektórych składników, takich jak zawartość cukru) i złożeniu nektaru lub pyłku przechodzą do tzw. tańca znanego jako „waggle dance”

# Biologiczne tło algorytmu BA

- Taniec pszczół (mowa pszczół) - konieczny dla komunikacji w kolonii i zawiera trzy informacje:
  - kierunek
  - odległość od ula
  - ocena jakości źródła pożywienia
- Te informacje pomagają kolonii na wysłanie swoich pszczół w dokładne położenie kwiatów (bez przewodników i map)



# Biologiczne tło algorytmu BA

- Wiedza o środowisku zewnętrznym jest czerpana wyłącznie od tańca „waggle dance”
- Taniec ten pozwala kolonii ocenić względne zalet różnych źródeł pożywienia i ilość energii potrzebnej do jej zbioru
- Po wykonanym tańcu (pszczółka zwiadowca) powraca z powrotem na tereny kwiatowe z pszczołami (tzw. rekrutami) czekającymi w ulu

# Biologiczne tło algorytmu BA

- Większa liczba pszczół naśladowczą zwiadowcę daje możliwość efektywniejszego zebrania pożywienia
- Podczas zbioru pożywienia, pszczoły monitorują jego poziom
- Jest to niezbędna informacja do podjęcia decyzji następnego tańca waggles po powrocie do ula
- Pożywienie - wystarczająco dobre - nadal „reklamowane” w tańcu waggles

# Algorytm BA

- Algorytm wymaga założenia wartości poszczególnym parametrom:
  - Liczba pszczół zwiadowców ( $n$ ),
  - Liczba miejsc wybranych  $n$  z odwiedzonych ( $m$ ),
  - Liczba najlepszych obszarów dla  $m$  odwiedzanych miejsc ( $e$ ),
  - Liczba rekrutów dla najlepszych  $e$  obszarów ( $n_{ep}$ ),
  - Liczba rekrutów dla innych ( $m-e$ ) wybranych obszarów ( $n_{sp}$ )
  - początkowy rozmiar obszarów ( $n_{gh}$ ) oraz sąsiedztwa
  - kryterium stopu

# Schemat algorytmu BA

## **Procedura BA**

### **begin**

Generuj losowo początkową populację rozwiązań

Oceń jakość rozwiązań

### **repeat**

Wybierz miejsca do przeszukiwania sąsiedztwa

Wybierz rekrutów dla wybranych miejsc (większa liczba pszczoł dla najlepszych  $e$  miejsc) i oceń ich

Wybierz najlepszą pszczołę z każdego obszaru

Przypisz pozostałym pszczołom przemieszczenia losowe i ocen je

**until** kryterium końca BA niespełnione

### **end**

# Zastosowania algorytmu BA

- Uczenie sieci neuronowych do rozpoznawania wzorców
- Tworzenie komórek produkcji
- Harmonogramowanie zadań linii produkcyjnej
- Rozwiązywanie inżynierskich problemów ciągłej optymalizacji
- Znajdowanie wielu rozwiązań dopuszczalnych (wielomodalność)

# Zastosowania algorytmu BA

- Grupowanie danych
- Optymalizacja projektowania elementów mechanicznych
- Wielokryterialna optymalizacja
- Strojenie rozmytych regulatorów dla robotów
- Grafika komputerowa i analiza obrazów

# Algorytm koloni sztucznych pszczół (ABC)

- ABC algorytm zaproponowany przez Karaboga (2005) do problemów optymalizacji
- ABC symuluje inteligentne zachowanie zbierania pożywienia przez rój pszczół
- W algorytmie ABC, kolonia pszczół składa się z dwóch grup:
  - **pszczoły robotnice**
  - **pszczoły bezrobotne**  
( **obserwatorzy i zwiadowcy**)

# Algorytm koloni sztucznych pszczoł (ABC)

- Pszczoły zwiadowcy losowo przeszukują otoczenie środowiska ula dla znalezienia źródeł pożywienia
- Zachowanie takie ma charakter fluktuacji i są istotne dla samoorganizacji
- Obserwatorzy czekają w ulu na pożywienie by ocenić informację o pożywieniu dostarczone przez robotnice
- Średnia liczba zwiadowców jest na poziomie od 5% –10% osobników



# Algorytm koloni sztucznych pszczół (ABC)

- Pszczoły robotnice którym źródło żywności zostało wyczerpane stają się zwiadowcami
- Miejsce źródła pokarmu stanowi możliwe rozwiązanie problemu optymalizacji
- Ilość nektaru ze źródła pożywienia odpowiada jakości rozwiązania
- Liczba pracujących pszczół jest równa liczbie źródeł żywności
- Każda z robotnic reprezentuje aktualny eksploatowany obszar

# Techniczne szczegóły ABC algorithmu

- Utworzenie nowego rozwiązania  
(położenie źródła pożywienia)

$\mathbf{v}_i$  jest w sąsiedztwie  $\mathbf{x}_i$  pszczoły robotnicy

$$\mathbf{v}_i = \mathbf{x}_i + \varphi_i(\mathbf{x}_i - \mathbf{x}_k) \quad (*)$$

gdzie

$\mathbf{x}_k$  – losowo wybrany wektor ( $k \neq i$ )

$\varphi_i$  – losowa liczba z zakresu  $[-1,1]$

# Techniczne szczegóły ABC algorithmu

- Sztuczna pszczoła obserwator wybiera źródło pożywienia na podstawie prawdopodobieństwa

$$p_i = \frac{f_i}{\sum_{k=1}^n f_k} \quad (**)$$

gdzie

$f_i$  – wartość przystosowania  $i$

(proporcjonalna do ilości nektaru w położeniu  $i$ )

$n$  – liczba źródeł pożywienia (równa liczbie pszczoł robotnic)

# Techniczne szczegóły ABC algorithmu

- Usuwane rozwiązane ze skautem

$$x_i^j = x_{\min}^j + \varepsilon \cdot (x_{\max}^j - x_{\min}^j) \quad (***)$$

gdzie

$x_{\min, \max}^j$  - dolny/górny zakres  $j$  współrzędnej  
parametru  $x_i$

$\varepsilon$  - losowa liczba z zakresu  $[0,1]$

# Schemat algorytmu ABC

## Procedura ABC

**begin**

Inicjalizuj populację  $n$  s osobników  $x_i$

Oceń populację

**repeat**

Utwórz nowe rozwiązanie  $v_i$  dla bezrobotnych pszczół (\*)

Zastosuj chciwą metodę selekcji dla pracujących pszczół

Oblicz prawdopodobieństwo  $p_i$  dla rozwiązania  $x_i$  przez (\*\*)

Utwórz nowe rozwiązania dla obserwatorów  $x$  w zależności od  $p_i$

Zastosuj chciwą metodę selekcji dla pracujących pszczół

Określ rozwiązania odrzucane, jeśli można i losowo wygeneruj nowe  $x_i$  (\*\*\*)

Zapamiętaj rozwiązanie najlepsze dotychczas znalezione

**until** warunek zakończenia niespełniony

**end**

# Literatura

1. Lukasik, S.; Zak, S. (2009). Firefly algorithm for continuous constrained optimization task. 5796. pp. 97–100.
2. [http://en.wikipedia.org/wiki/Firefly\\_algorithm](http://en.wikipedia.org/wiki/Firefly_algorithm)
3. Sayadi, M. K.; Ramezani, R.; Ghaffari-Nasab, N. (2010). "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems". *Int. J. of Industrial Engineering Computations* 1: 1–10.
4. K.N. Krishnanand and D. Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intelligence*, Vol. 3, No. 2, pp. 87–124, June 2009.
5. [http://en.wikipedia.org/wiki/Glowworm\\_swarm\\_optimization](http://en.wikipedia.org/wiki/Glowworm_swarm_optimization)
6. <http://www.bees-algorithm.com/>
7. <http://mf.erciyes.edu.tr/abc/>