

Algorytm FIREFLY

Michał Romanowicz
Piotr Wasilewski



Struktura prezentacji

1. Twórca algorytmu
2. Inspiracja w przyrodzie
3. Algorytm
4. Zastosowania algorytmu
5. Krytyka algorytmu
6. Porównanie z PSO

Twórca algorytmu

- Xin-She Yang
- doktorat w Oxfordzie z matematyki stosowanej
- Middlesex University London
- znany z algorytmów:
 - firefly algorithm (2008)
 - cuckoo search (2009)
 - bat algorithm (2010)
 - flower pollination algorithm (2012)



Świetlikowate

- robaczki świętojańskie
- *lampyridae*
- rodzina z rzędu chrząszczy
- ok. 2000 gatunków
- żyją na obszarach leśnych
- bioluminescencja
- w Polsce żyją trzy gatunki:
 - iskrzyk
 - świeciuch
 - świetlik świętojański



Świetlikowate – bioluminescencja

- zimne źródło światła
- funkcje bioluminescencji:
 - przyciąganie innych świetlikowatych, potencjalnych partnerów (komunikacja)
 - przyciąganie potencjalnego pożywienia
 - mechanizm odstraszający
- generowane kolory:
 - żółte
 - zielone
 - jasnoczerwone
- różne gatunki mają swoje unikatowe wzorce świecenia
- samce i samice mają unikatowe wzorce świecenia
- mogą synchronizować swoje emitowanie światła tak by tworzyć bardziej złożone wzorce







Algorytm

- Tworzony jest rój robaczków świętojańskich
- Dany osobnik może być postrzegany jako rozwiązanie
- Imituje zachowanie roju robaczków świętojańskich (światło przyciąga kolejnych osobników)
- Jego zasady są następujące...



Algorytm – zasady

1. Założenie jedнопłciowości – jeden świetlik przyciąga inne świetliki
2. Atrakcyjność danego świetlika jest proporcjonalna do jasności emitowanego przezeń światła
3. Świetlik o mniejszej jasności jest przyciągany do osobnika o większej jasności
4. Jasność może się zmniejszać wraz ze zwiększającą się odległością
5. Jeśli nie ma żadnych świetlików o jaśniejszym świetle wtedy porusza się losowo
6. Jasność świetlika jest proporcjonalna do jego jakości w rozważanym problemie

Algorytm – I/O

wejście:

$f(x)$ - funkcja celu (funkcja kosztów), gdzie $x = [x_1 \dots x_n]$ - wektor parametrów

M - rój agentów

β_0 - maksymalna wartość atrakcyjności

γ - współczynnik absorpcji

u_{\min} , u_{\max} - określające wektor kierunku

wyjście:

x_i - gdzie $f(x_i) \rightarrow \min$

Algorytm – pseudokod

```
begin
/* wylosuj populację M świetlików */
repeat
  for i = 1 to M do
    for j = 1 to M do
      if  $f(x_j) > f(x_i)$  then
         $r_{ij} = \text{oblicz\_odległość}(x_i, x_j)$ 
         $\beta_i = \beta_0 \exp(-\gamma \cdot r_{ij}^2)$  /* modyfikujemy atrakcyjność */
         $u_i = \text{generuj\_losowy\_wektor}(u_{\min}, u_{\max})$ 
         $x_i = x_i + \beta_i (x_j - x_i) + u_i$  /* tworzymy nowe rozwiązanie */
      end
    end
  end
   $u_k = \text{generuj\_losowy\_wektor}(u_{\min}, u_{\max})$ 
   $x_k += u_k$  /* przesuwamy 'najlepszy' świetlik */
until warunek zakończenia niespełniony
end
```

Parametry algorytmu

Parametry szczegóły:

Maksimum atrakcyjności $\beta_0 \in [0, 1]$

Współczynnik absorpcji $\gamma \in [0.01, 100]$ (prop. przez Yanga w klas. schemacie)

Zastosowania algorytmu

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

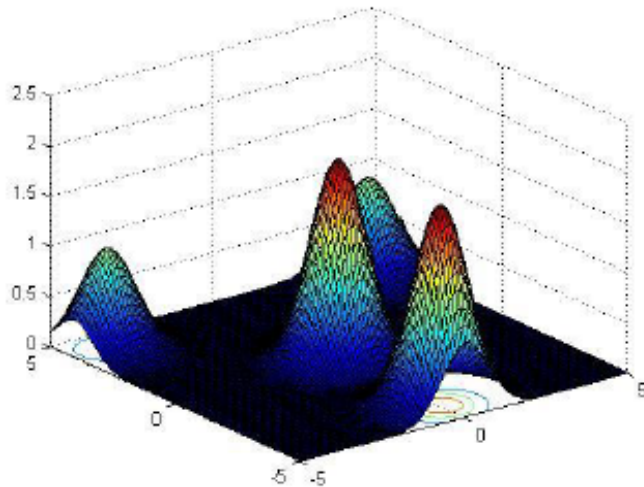
Krytyka algorytmu

- *„FA, on the other hand, has little to distinguish it from PSO, with the inverse-square law having a similar effect to crowding and fitness sharing in EAs, and the use of multi-swarms in PSO.“*
- *„For example, the differences between the particle swarm optimization metaheuristic and "novel" metaheuristics like the firefly algorithm, the fruit fly optimization algorithm, the fish swarm optimization algorithm or the cat swarm optimization algorithm seem negligible.“*

Porównanie z PSO

3.1. Four Peak function

$$f1(x,y) = e^{-(x-4)^2-(y-4)^2} + e^{-(x+4)^2-(y-4)^2} + 2[e^{-x^2-y^2} + e^{-x^2-(y+4)^2}]$$

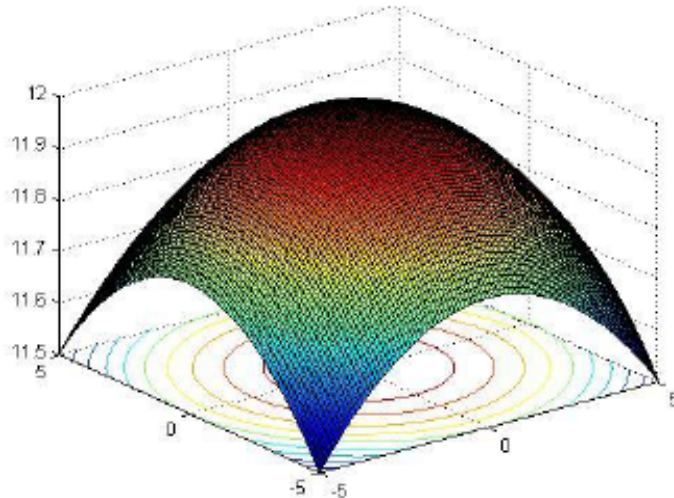


$f(x,y)$	N	PSO	FFA
$f1(x,y)$	15	1.5356	1.4840
	20	2.0135	1.9326
	25	2.4959	2.3652
	30	2.9367	2.8186
	35	3.4758	3.2951

Porównanie z PSO – ciąg dalszy

3.2. Parabolic Function

$$f2(x, y) = 12 - (x^2 + y^2)/100$$

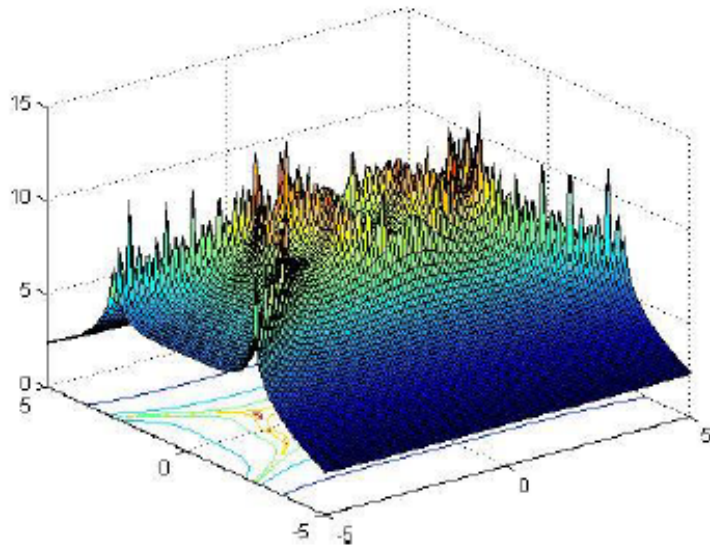


$f2(x, y)$	15	1.5482	1.5039
	20	2.0884	1.9296
	25	2.6466	2.3504
	30	2.9733	2.7769
	35	3.4039	3.2429

Porównanie z PSO – ciąg dalszy

3.3. Camelback Function

$$f_3(x, y) = 10 - \log \left(x^2 - \left(4 - 2.1x^2 + \left(\frac{1}{3} \right) x^4 \right) + xy + 4y^2(y^2 - 1) \right)$$

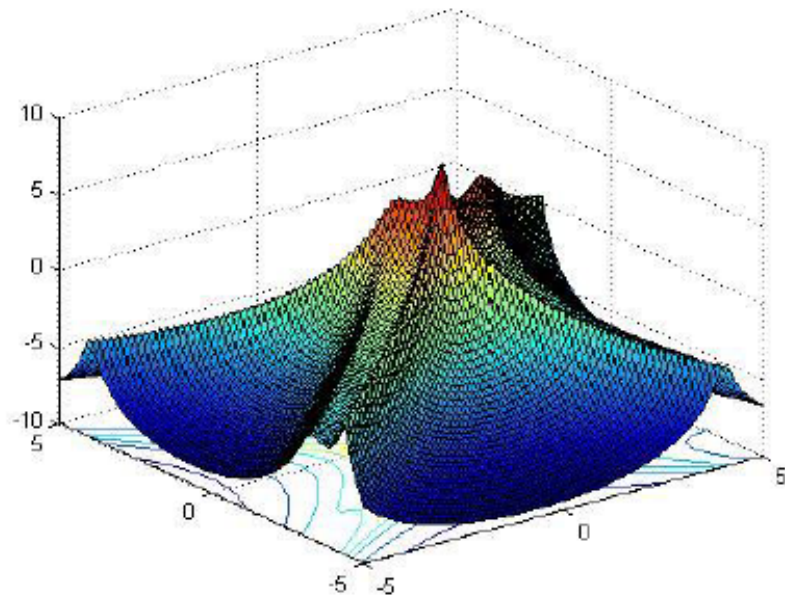


$f_3(x, y)$	15	3.3491	3.2372
	20	4.3162	4.2481
	25	5.3308	5.2477
	30	6.3998	6.3473
	35	7.4936	7.1793

Porównanie z PSO – ciąg dalszy

3.4. Goldstein-Price Function

$$f_4(x, y) = \frac{10 + \log\left[\frac{1}{\{(1+(1+x+y)^2(19-14x+3x^2-14y+6xy+3y^2))\}}\right]^*}{(30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2))}$$

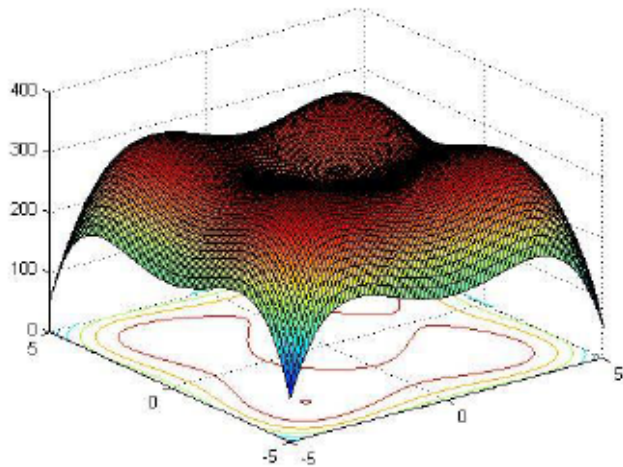


$f_4(x, y)$	15	1.7848	1.7189
	20	2.3342	2.2550
	25	2.8904	2.7896
	30	3.4392	3.3023
	35	3.9492	3.8121

Porównanie z PSO – ciąg dalszy

3.5. *Styblinski Function*

$$f_5(x, y) = 275 - \left[\left(\frac{x^4 - 16x^2 + 5x}{2} \right) + \left(\frac{y^4 - 16y^2 + 5y}{2} \right) + 3 \right]$$

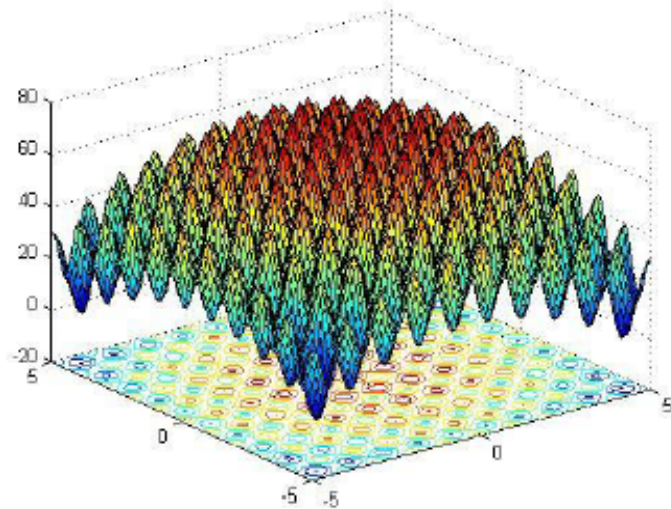


$f_5(x, y)$	15	1.6444	1.5478
	20	2.1504	2.0725
	25	2.6144	2.5323
	30	3.1201	3.0196
	35	3.5502	3.4115

Porównanie z PSO – ciąg dalszy

3.6. Rastrigin Function

$$f_6(x, y) = 80 - [20 + x^2 + y^2 - 10(\cos(2\pi x) + \cos(2\pi y))]$$

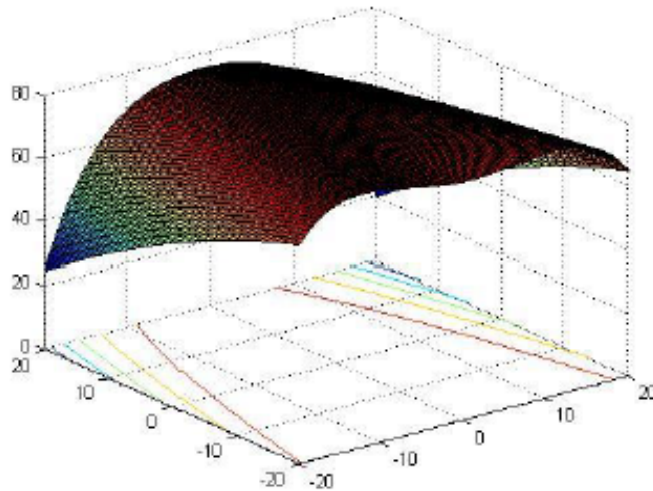


$f_6(x, y)$	15	9.6761	9.5298
	20	12.6412	12.5404
	25	15.6878	15.5457
	30	18.6878	18.4993
	35	21.6923	21.4953

Porównanie z PSO – ciąg dalszy

3.7. Rosenbrock Function

$$f_7(x, y) = 70 \left[\frac{\left[\left[20 - \left\{ \left(1 - \frac{x}{-7} \right)^2 + \left(\frac{y}{6} + \left(\frac{x}{-7} \right)^2 \right)^2 \right\} \right] + 150 \right]}{170} \right] + 10$$



$f_7(x, y)$	15	1.3076	1.2575
	20	1.6808	1.6187
	25	2.0679	2.0029
	30	2.4794	2.3223
	35	2.8054	2.6684

Porównanie z PSO – ciąg dalszy

- *„This implies that FFA is potentially more powerful in solving noisy non-linear optimization problems. The FFA seems to be favorable optimization tool in part due to the effect of the attractiveness function which is a unique to the firefly behavior. The FFA not only includes the self improving process with the current space, but it also includes the improvement among its own space from the previous stages.“*

Koniec

Dziękujemy za uwagę