

# Dokumentacja algorytmu

## Optymalizacja rojem cząstek (PSO)

*Anastasiia Bzova 66617*

### Opis wybranego algorytmu:

**Algorytm optymalizacji rojem cząstek (PSO)** to metoda inteligencji obliczeniowej inspirowana zachowaniem zbiorowym takich systemów jak stada ptaków czy ławice ryb. Został zaproponowany przez Jamesa Kennedy'ego i Russella Eberharta w 1995 roku. Jest metodą optymalizacji ciągłych nieliniowych funkcji.

#### **Opierała się na dwóch metodach :**

1) Sztuczne życie [artificial life]

(stada ptaków, ławice ryb, teoria roju)

2) Ewolucja obliczeniowa [evolutionary computation]

(algorytmy genetyczne, strategie ewolucyjne, programowanie genetyczne)

W PSO grupa "cząstek" (rozwiązań) porusza się w przestrzeni poszukiwań, kierując się zarówno własnym doświadczeniem, jak i doświadczeniem innych cząstek. Każda cząstka posiada **pozycję**, **prędkość** oraz **pamięta najlepszą pozycję**, jaką kiedykolwiek osiągnęła. Ruch odbywa się poprzez aktualizację prędkości i pozycji na podstawie trzech czynników:

- **Inercja** – tendencja do utrzymania dotychczasowego kierunku ruchu;
- **Komponenta poznawcza** – dążenie do własnego najlepszego rozwiązania;
- **Komponenta społeczna** – dążenie do najlepszego rozwiązania znalezionego przez cały rój.

### Złożoność obliczeniowa

Mamy takie dane:

$n$  — liczba cząstek (rozwiązań),

$d$  — wymiar przestrzeni,

$t$  — liczba iteracji.

### Złożoność PSO:

**Złożoność czasowa:**  $O(n * d * t)$  — w każdej iteracji każda cząstka aktualizuje swoją pozycję i ocenia funkcję dopasowania.

**Złożoność pamięciowa:**  $O(n * d)$  — przechowywanie pozycji, prędkości i najlepszych wyników każdej cząstki.

## Zastosowanie algorytmu PSO

PSO mają wiele zastosowań w syntezie logicznej, logice, kryptografii. Znajduje szerokie zastosowanie w ogólnie zadaniach optymalizacyjnych:

- Optymalizacja funkcji matematycznych (np. Rastrigina, Rosenbrocka, Ackleya),
- Uczenie maszynowe – strojenie hiperparametrów modeli,
- Przetwarzanie sygnałów i obrazów – filtrowanie, segmentacja,
- Energetyka – optymalne zarządzanie zasobami i rozkład obciążenia,
- Robotyka – planowanie trajektorii i koordynacja ruchu,
- Inżynieria – optymalizacja projektów i konstrukcji.

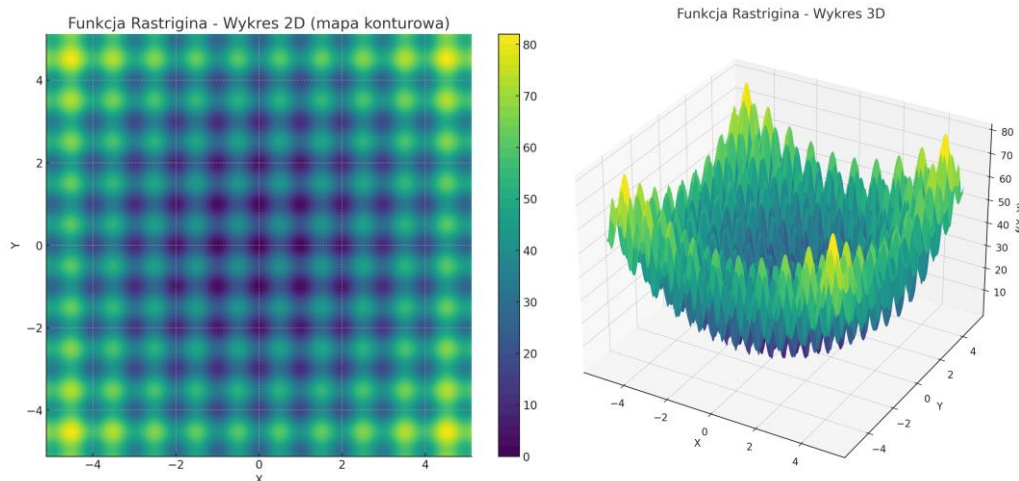
*więcej o algorytmie Rastrigina*

**Funkcja Rastrigina** to popularna funkcja testowa używana w optymalizacji i uczeniu maszynowym - do oceny wydajności algorytmów optymalizacyjnych. Opracowana przez L.A. Rastrigina. Funkcja ma wiele minimów lokalnych, co czyni ją trudną dla algorytmów poszukujących globalnego minimum.

Wzór:

$$f(x) = A \cdot n + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$$

- $A=10$  (standardowo),
- $x_i$  to  $i$ -ty wymiar wektora wejściowego,
- $n$  to liczba wymiarów.



**Wykres 2D** — mapa konturowa pokazująca doliny i wzgórza funkcji.

**Wykres 3D** — pokazujący kształt funkcji przestrzennie: widać wiele lokalnych minimów.

## Co zostało zaimplementowane

Klasa “PSOVisualization”, która zarządza animacją i rysowaniem.

Klasa “PSO”, zawierająca logikę algorytmu PSO.

Klasa “Particle,” reprezentująca pojedynczą cząstkę.

Klasa pomocnicza “MathTools” do funkcji matematycznych (m.in. funkcja Rastrigina, mapowanie wartości).

**Skorzystano z klasycznego PSO** (model Clerc-Kennedy) który wygląda tak:

$$v_j(t + 1) = w \cdot v_j(t) + c_1 \cdot r_1 \cdot (pbest_j - x_j(t)) + c_2 \cdot r_2 \cdot (gbest_j - x_j(t))$$

- $v_j(t)$  — prędkość cząstki w wymiarze  $j$  w czasie  $t$ ,
- $x_j(t)$  — pozycja cząstki w wymiarze  $j$  w czasie  $t$ ,
- $pbest_j$  — najlepsza osobista pozycja cząstki w wymiarze  $j$ ,
- $gbest_j$  — najlepsza globalna pozycja (znaleziona przez cały rój) w wymiarze  $j$ ,
- $w$  — **waga bezwładności** (*InertiaWeight*),
- $c_1$  — **współczynnik kognitywny** (*CognitiveWeight*),
- $c_2$  — **współczynnik społeczny** (*SocialWeight*),
- $r_1, r_2$  — losowe liczby z przedziału  $[0, 1]$ .

**Działanie programu :**

Program rysuje cząstki w przestrzeni 2D ograniczonej zakresem funkcji Rastrigina [-5.12, 5.12] (jest to domyślne znaczenia).

Cząstki są wizualizowane jako kółka: czerwone (aktywnie szukające) i zielone (blisko najlepszego rozwiązania).

W miarę postępu iteracji, cząstki zbliżają się do najlepszego znalezionej minimum globalnego.

Użytkownik może sterować prędkością animacji za pomocą suwaka.

Do płynności animacji wykorzystano podwójnego buforowania formularzy i obrazu + [Interpolację](#) (Buforowanie podwójne używa buforu pamięci do rozwiązywania problemów z migotaniem skojarzonych z wieloma operacjami malowania. )

## Optymalizacja i przyspieszenie algorytmu

1. **Ograniczanie prędkości:** aby zapobiec zbyt szybkiemu "uciekaniu" cząstek poza przestrzeń poszukiwań.
2. **Topologia lokalna:** ograniczenie interakcji do sąsiedztwa, zmniejszające liczbę obliczeń.
3. **Równoległe przetwarzanie:** aktualizacja cząstek może być wykonywana niezależnie, co umożliwia wykorzystanie wielordzeniowych procesorów lub GPU.

## Możliwe ulepszenia

- Obsługa więcej niż 2 wymiarów przez projekcję.
- Dodanie różnych funkcji celu (np. Sphere, Rosenbrock).
- Lepsza kontrola nad parametrami algorytmu w rzeczywistym czasie (GUI).
- Zapisywanie wyniku jako na przykład plik.

## Alternatywne algorytmy:

Ponieważ PSO rozwiązuje zadania globalnej optymalizacji, można go porównać do następujących algorytmów:

<b><u>Algorytm</u></b>	<b><u>Cechy charakterystyczne</u></b>
<b>Algorytmy genetyczne (GA)</b>	Wykorzystują selekcję, krzyżowanie i mutację. Często są wolniejsze.
<b>Differential Evolution (DE)</b>	Dobry balans eksploracji i eksploatacji. Prosta implementacja.
<b>Symulowane wyżarzanie (SA)</b>	Inspiracja procesem hartowania. Skuteczny przy złożonych krajobrazach funkcji.
<b>Ant Colony Optimization (ACO)</b>	Bardzo dobre dla problemów trasowania i optymalizacji dyskretnej.
<b>Gradient Descent / Adam</b>	Skuteczne dla funkcji różniczkowalnych. Szybkie, ale mogą utknąć w minimum lokalnym.

### ~Podsumowanie~

Algorytm PSO jest potężnym i elastycznym narzędziem oraz jest łatwy do zaimplementowania – stosowany do globalnej optymalizacji. Jego siłą jest prostota równoległego przetwarzania oraz skuteczność w szerokim zakresie problemów. Dzięki możliwościom adaptacji do różnych typów zadań, PSO znajduje zastosowanie w wielu dziedzinach nauki i techniki.