

# **Algorytm roju** **cząstek** **POP - PROJEKT**

**Autorzy:**

**Damian Kolaska 300229**

**Mateusz Dryjański 304036**

# Spis treści

1. Opis projektu: .....	2
2. Opis algorytmu: .....	2
2.1. Wzór na prędkość cząstki: .....	2
2.2. Znaczenie głównych parametrów algorytmu: .....	3
2.3. Schemat działania algorytmu: .....	3
3. Badania: .....	4
3.1 Wartość współczynnika bezwładności: .....	4
3.2 Miary jakości: .....	4
3.3 Wybrane funkcje testowe: .....	5
3.3. Maszyny: .....	7
4. Stos technologiczny: .....	8
5. Źródła: .....	8

# 1. Opis projektu:

## **Algorytm roju cząstek z modyfikacjami dotyczącymi współczynnika bezwładności.**

Modyfikacja powinna polegać na opracowaniu np. propozycji dynamicznej wartości współczynnika. Zadanie polega na implementacji algorytmu roju cząstek z modyfikacją współczynnika bezwładności. Modyfikacja powinna polegać na zaproponowaniu dynamicznej wartości współczynnika (np. zależnego od czasu, zmieniającego się w czasie). Należy porównać wyniki ze standardowym algorytmem roju cząstek

# 2. Opis algorytmu:

Optymalizacja za pomocą roju cząstek (ang. Particle Swarm Optimization) to algorytm metaheurystyczny służący do rozwiązywania problemów optymalizacyjnych. "Rój" to określenie na zbiór, a "cząstki" to potencjalne rozwiązania problemu.

Jest to stochastyczny algorytm obliczeniowy naśladujący zachowanie przemieszczającego się stada ptaków. Jego ideą jest iteracyjne przeszukiwanie przestrzeni rozwiązań problemu poprzez poruszający się w przestrzeni rój cząstek. W każdej iteracji zapisywane jest najlepsze rozwiązanie znalezione dotychczas przez każdą z cząstek (rozwiązanie lokalne), a także najlepsze rozwiązanie z całego roju (rozwiązanie globalne). Prędkość ruchu poszczególnych cząstek jest zależna od położenia optymalnego globalnego i lokalnego rozwiązania oraz od prędkości w poprzednich iteracjach. W podobny sposób, w stadzie ptaków, każdy osobnik porusza się w prędkością którą określa na podstawie doświadczeń własnych oraz reszty stada. Jednocześnie dobiera on prędkość tak, aby zachowany był odpowiedni dystans od innych osobników.

## 2.1. Wzór na prędkość cząstki:

$$v = \omega v + \varphi_l r_l (l - x) + \varphi_g r_g (g - x)$$

Gdzie:

$v$  - prędkość cząstki

$\omega$  - współczynnik bezwładności (waga inercji)

$\varphi_l$  - współczynnik dążenia do najlepszego lokalnego rozwiązania (kognitywny współczynnik uczenia)

$\varphi_g$  - współczynnik dążenia do najlepszego globalnego rozwiązania (socjalny współczynnik uczenia)

$l$  - położenie najlepszego lokalnego rozwiązania

$g$  - położenie najlepszego globalnego rozwiązania

$x$  - położenie cząstki

$r_l, r_g$  - losowe wartości z przedziału  $[0, 1]$

## 2.2. Znaczenie głównych parametrów algorytmu:

- **Współczynnik bezwładności** określa wpływ prędkości w poprzednim kroku na jej aktualną wartość. Wraz ze wzrostem tego parametru, zwiększa się zdolność cząstek do przeszukiwania nowych rejonów przestrzeni rozwiązań
- Współczynnik dążenia do najlepszego lokalnego rozwiązania  $\varphi_l$  określa zdolność cząstki do oscylacji wokół swojej najlepszej pozycji. Im większa wartość współczynnika, tym większa zdolność do oscylacji
- Współczynnik dążenia do najlepszego globalnego rozwiązania  $\varphi_g$  określa tendencję do grupowania się cząstek wokół najlepszego globalnego rozwiązania. Im większa wartość współczynnika tym większa tendencja do grupowania się cząstek

## 2.3. Schemat działania algorytmu:

**Na początku:**

- Dla każdej cząstki ze zbioru:
  - Wylosuj pozycję początkową z przestrzeni rozwiązań
  - Zapisz aktualną pozycję cząstki jako najlepsze lokalne rozwiązanie
  - Jeśli rozwiązanie jest lepsze od najlepszego rozwiązania globalnego, zapisz je jako najlepsze
  - Wylosuj prędkość początkową

### **Dopóki nie zostanie spełniony warunek stopu:**

- Dla każdej cząstki ze zbioru:
  - Wybierz losowe wartości parametrów  $r_l, r_g$
  - Zaktualizuj prędkość cząstki (według wzoru z pkt 2.1)
  - Jeśli aktualne rozwiązanie jest lepsze od najlepszego rozwiązania aktualnego to zapisz je jako najlepsze lokalne rozwiązanie
  - Jeśli aktualne rozwiązanie jest lepsze od najlepszego rozwiązania globalnego to zapisz je jako najlepsze rozwiązanie globalne

## **3.Badania:**

Badania jakości zaimplementowanego rozwiązania (zmodyfikowanego algorytmu poprzez opracowaniu propozycji dynamicznej wartości współczynnika bezwładności) będą polegały na porównaniu jego skuteczności ze standardową implementacją algorytmu roju cząstek. Algorytmy będziemy porównywać dla zadania minimalizacji dla wybranych przez nas funkcji testowych (patrz punkt 3.3).

### **3.1 Wartość współczynnika bezwładności:**

Dynamiczną wartość współczynnika bezwładności będziemy wyznaczać następującymi sposobami:

- **odwrotnie zależna od numeru iteracji**
  - oczekiwana lepsza eksploatacja
- **losowa**
  - oczekiwana lepsza eksploracja, ale gorsza eksploatacja
- **wprost proporcjonalna do odległości od minimum globalnego**
  - lepsza eksploracja cząstek podążających za liderem

### **3.2 Miary jakości:**

Wersje algorytmu będziemy porównywać ze względu na jakość najlepszego rozwiązania, a także szybkość jego uzyskania.

### 3.3 Wybrane funkcje testowe:

Do badań zamierzamy użyć funkcji testowych z pakietu benchmark-functions [3]

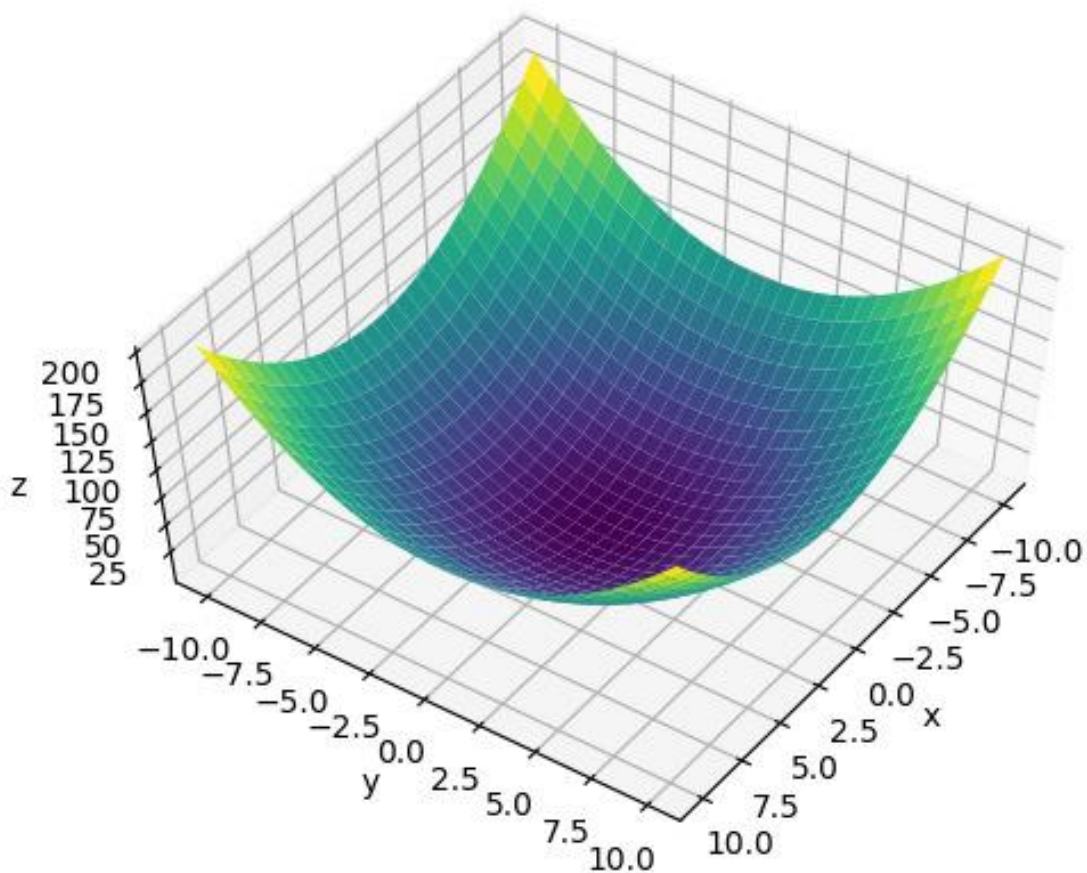
A dokładniej:

- **funkcji hipersfery:**

$$f(x) = \sum_{i=0}^{N-1} x_i^2$$

Funkcja kwadratowa w N wymiarach. Jedno minimum globalne.

**Trudności dla algorytmu:** brak. Prosty przypadek optymalizacji.



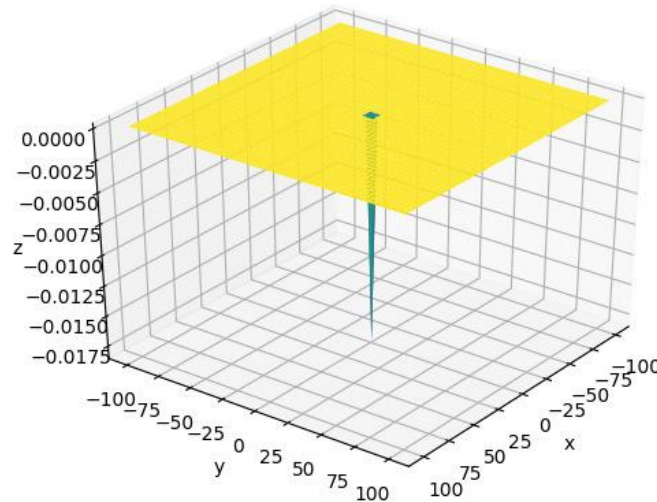
Wykres 1: Wykres 3D funkcji hipersfery

- funkcji Easoma:

$$f(x) = -\cos(x_0)\cos(x_1)\exp(-(x_0 - \pi)^2 - (x_1 - \pi)^2)$$

Funkcja jednomodalna z jednym minimum globalnym.

**Trudność dla algorytmu:** minimum znajduje się w stosunkowo małym obszarze



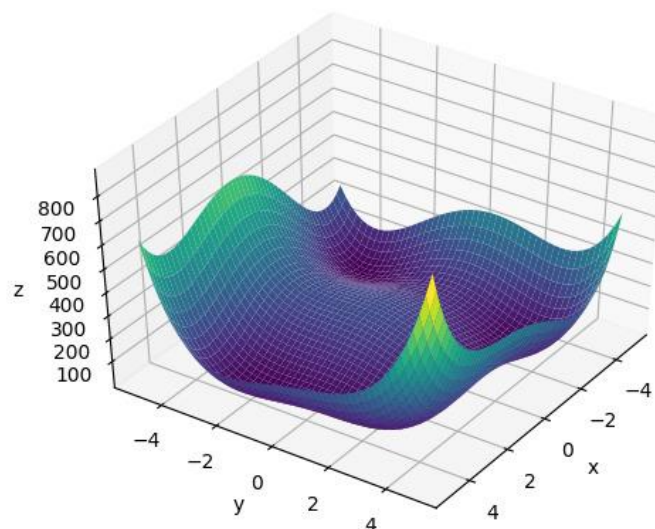
Wykres 2: Wykres 3D funkcji Easoma

- funkcji Himmelblau:

$$f(x) = (x_0^2 + x_1 - 11)^2 + (x_0 + x_1^2 - 7)^2$$

Funkcja wielomodalna z czterema minimami globalnymi

**Trudność dla algorytmu:** minima rozłożone są nierównomiernie w przestrzeni poszukiwań



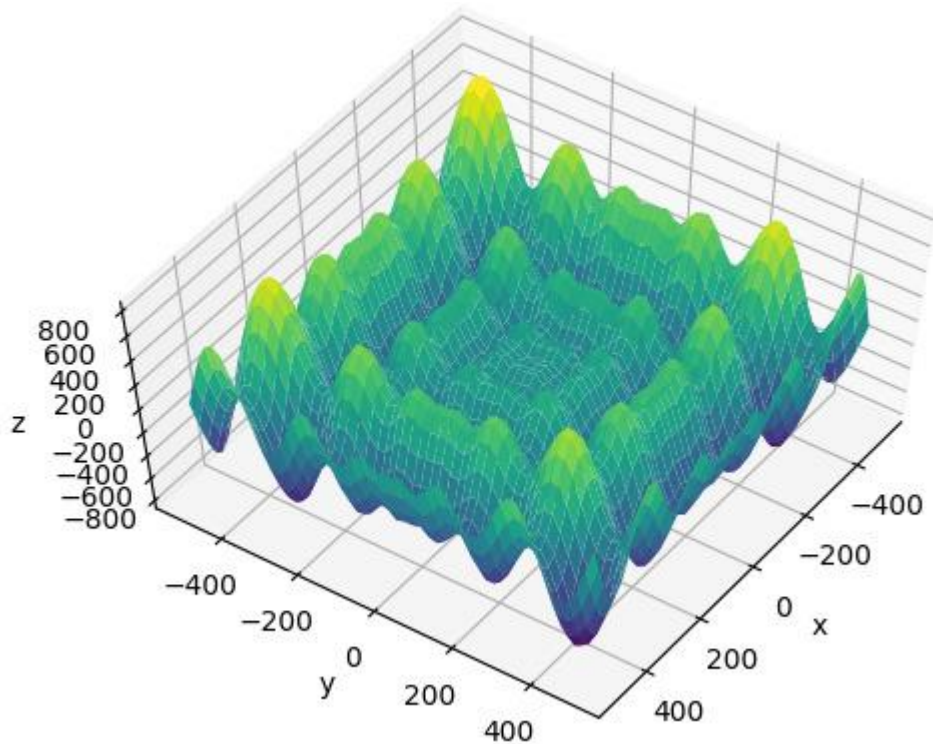
Wykres 3: Wykres 3D funkcji hipersfery

- funkcji Schwefela

$$f(x) = 418.9829N \sum_{i=0}^{N-1} x_i \sin(\sqrt{|x_i|})$$

Funkcja wielomodalna o złożonej strukturze.

**Trudność dla algorytmu:** najlepsze minima lokalne są od siebie sporo oddalone. Duże ryzyko utknięcia w jednym z nich.



Wykres 4: Wykres 3D funkcji Schwefela

### 3.3. Maszyny:

Badania nad algorytmem zostaną przeprowadzone na dwóch maszynach z procesorami:

Nazwa	Ilość rdzeni	Ilość wątków	RAM	Częstotliwość procesora
Ryzen 7 4800H	8	16	32GB	2,9-4,2Ghz
Intel Core i7-4810MQ	4	8	16GB	2,8-3,8Ghz



## 4.Stos technologiczny:

- **Język programowania:** Python w wersji 3.8
- **Edytor kodu:** Visual Studio Code
- **System operacyjny:** Linux Ubuntu 20.04

## 5.Źródła:

- [1] Opis algorytmu optymalizacji za pomocą roju cząstek:
  - <http://www.alife.pl/optymalizacja-rojem-czastek>
- [2] Praca naukowa na temat algorytmu optymalizacji rojem cząstek do znajdowania ekstremów globalnych:
  - [http://zeszyty-naukowe.wysi.edu.pl/zeszyty/zeszyt13/Zastosowanie\\_algorytmu\\_optymalizacji\\_rojem\\_czastek\\_do\\_znajdowania\\_ekstremow\\_globalnych\\_wybranych\\_funkcji\\_%20testowych.pdf](http://zeszyty-naukowe.wysi.edu.pl/zeszyty/zeszyt13/Zastosowanie_algorytmu_optymalizacji_rojem_czastek_do_znajdowania_ekstremow_globalnych_wybranych_funkcji_%20testowych.pdf)
- [3] Dokumentacja pakietu benchmark-functions:
  - <https://pypi.org/project/benchmark-functions/>
- [4] Opis algorytmu optymalizacji za pomocą roju cząstek wraz z przykładową implementacją:
  - <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/>