

# Sprawozdanie – Projekt 1

Kamil Urbanowski

Michał Szymacha

Jakub Święs

Paweł Styka

# 1. Wykorzystane technologie i wymagania środowiska

Do projektu został użyty język Python z następującymi bibliotekami głównymi:

- NumPy: Służy do operacji matematycznych i obsługi wielowymiarowych tablic, co jest kluczowe przy obliczeniach optymalizacyjnych.
- Matplotlib: Umożliwia tworzenie wykresów, które ilustrują przebieg algorytmu (np. wartość funkcji celu w kolejnych iteracjach).
- Pandas: Ułatwia manipulację i analizę danych, np. przy zapisywaniu wyników do bazy danych lub plików.
- Opfunu oraz benchmark-functions: Specjalistyczne biblioteki, które zawierają gotowe implementacje funkcji testowych wykorzystywanych do oceny działania algorytmu genetycznego. Warto przytoczyć linki do repozytoriów lub dokumentacji tych narzędzi.
- Tkinter lub inne narzędzia GUI: Jeżeli projekt został oparty na graficznym interfejsie użytkownika, należy opisać używany framework oraz uzasadnić wybór tej technologii.

## 2. Wybrane funkcje

Funkcja Ackleya jest powszechnie stosowanym benchmarkiem w dziedzinie optymalizacji ewolucyjnej ze względu na swój skomplikowany krajobraz poszukiwań. Charakteryzuje się ona licznymi lokalnymi minimami, co czyni ją trudną do optymalizacji, szczególnie dla algorytmów gradientowych. Funkcja ta jest wykorzystywana do testowania zdolności algorytmów do unikania pułapek lokalnych minimów oraz do oceny szybkości zbieżności do globalnego optimum.

$$f(x) = -a * \exp\left(-b \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(cx_i)\right) + a + e$$

Gdzie:

- $x = (x_1, x_2, \dots, x_D)$  – wektor zmiennych,
- $D$  – liczba wymiarów,
- typowe wartości parametrów to:  $a = 20, b = 0.2, c = 2\pi$
- $e$  – podstawa logarytmu naturalnego (ok. 2.7182818285)

### Charakterystyka funkcji:

- Sugerowany zakres poszukiwań:  $[-32.768, 32.768]$  dla każdej zmiennej.
- Globalne minimum: Funkcja osiąga globalne minimum w punkcie  $x^* = (0, 0, \dots, 0)$  przy czym wartość funkcji jest bardzo bliska 0 (w praktyce może wystąpić niewielkie odchylenie numeryczne, np. rzędu  $10^{-16}$ ).
- **Wielomodalność:** Funkcja posiada wiele lokalnych minimów, co stawia duże wymagania przed algorytmami optymalizacyjnymi, zmuszając je do eksploracji przestrzeni poszukiwań, a nie tylko eksploatacji obszaru w pobliżu początkowego rozwiązania.

Modyfikacja klasycznej funkcji Ackleya poprzez przesunięcie i obrót przestrzeni poszukiwań znacząco podnosi stopień trudności problemu optymalizacyjnego. W tej wersji funkcji globalne minimum nie znajduje się w naturalnym środku przestrzeni (tj. w punkcie  $(0, 0, \dots, 0)$ ) lecz w punkcie określonym przez wektor przesunięcia  $o$ . Dodatkowo, zastosowanie macierzy rotacji  $M$  powoduje, że zmienne funkcji stają się nieseparowalne – współzależne, co utrudnia tradycyjnym metodom optymalizacji wykrycie kierunku poprawy.

### Wzór funkcji:

Zakładając, że przekształcony wektor  $z$  definiuje się jako

$$z = M * (x - o)$$

Gdzie:

- $M$  – macierz rotacji,
- $o$  – wektor przesunięcia (shift wektor),

Wzór funkcji przyjmuje postać:

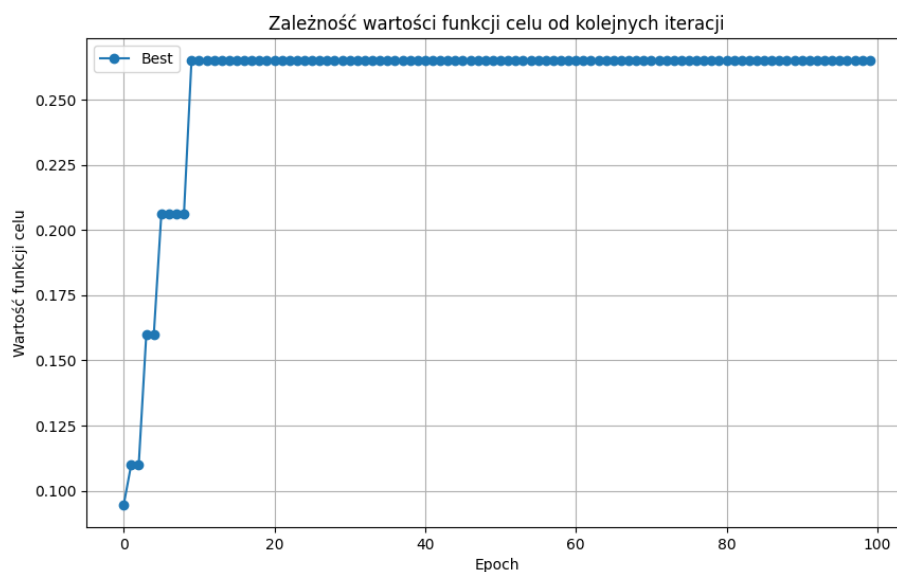
$$f(x) = -a * \exp\left(-b \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(cz_i)\right) + a + e$$

Przy czym standardowe wartości parametrów pozostają takie same ( $a = 20, b = 0.2, c = 2\pi$ )

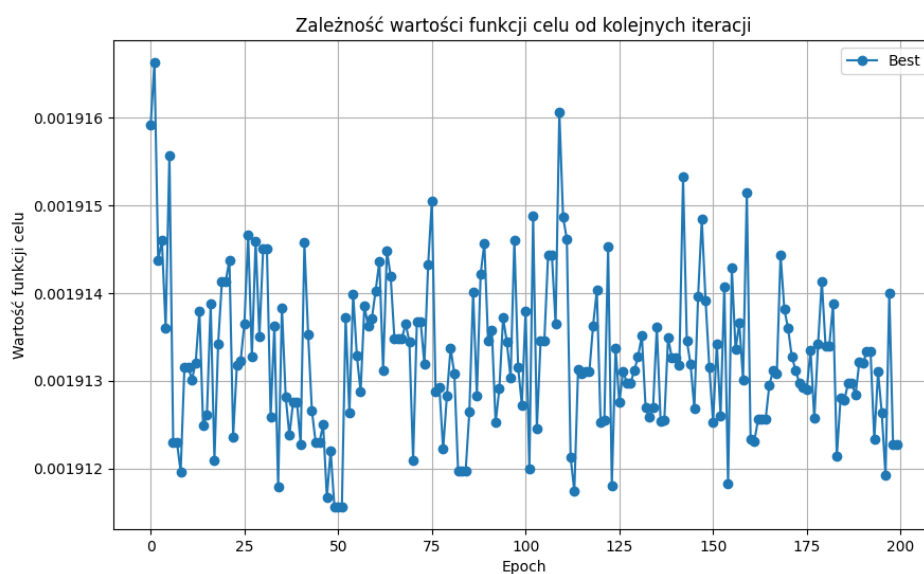
### Charakterystyka funkcji:

- **Przesunięcie globalnego optimum:** Dzięki wektorowi  $o$  globalne minimum funkcji przesunięte jest z punktu  $(0, 0, \dots, 0)$  do innego miejsca w przestrzeni, co symuluje realne problemy, gdzie optymalne rozwiązanie nie znajduje się w centrum zadanej przestrzeni.
- **Rotacja przestrzeni:** Zastosowanie macierzy  $M$  wprowadza nieliniowe zależności pomiędzy zmiennymi, czyniąc funkcję nieseparowalną oraz jeszcze bardziej złożoną pod kątem poszukiwania globalnego minimum.
- **Wielomodalność:** Podobnie jak klasyczna funkcja Ackleya, wersja shift-rotated charakteryzuje się licznymi lokalnymi minimami, jednak dodatkowe transformacje (przesunięcie i obrót) sprawiają, że krajobraz funkcji jest bardziej skomplikowany, co stanowi dodatkowe wyzwanie dla algorytmów optymalizacyjnych.

### 3. Wykres zależności wartości funkcji celu od kolejnych iteracji

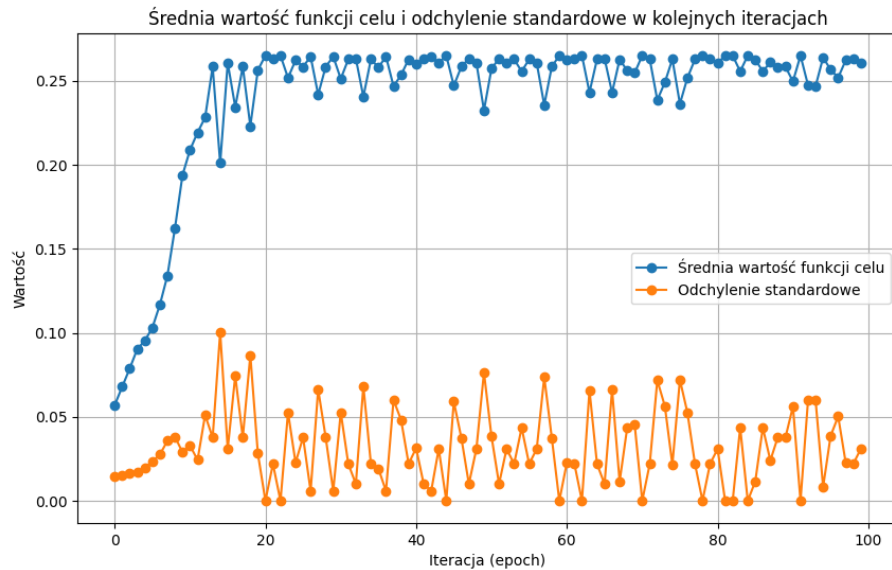


Rysunek 1 Funkcja Ackley

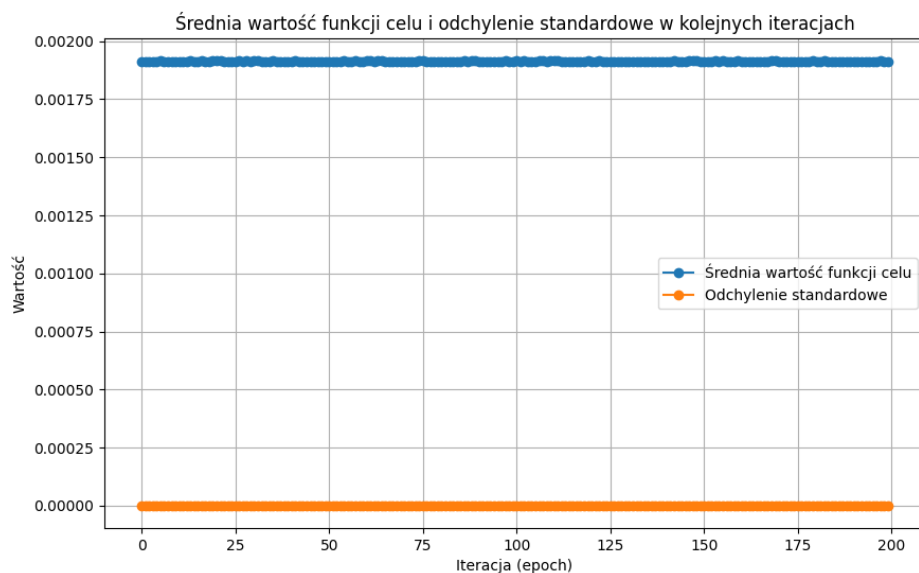


Rysunek 2 Funkcja F5 - Shifted and rotated Ackley

## 4. Wykres średniej wartości funkcji celu oraz odchylenia standardowego



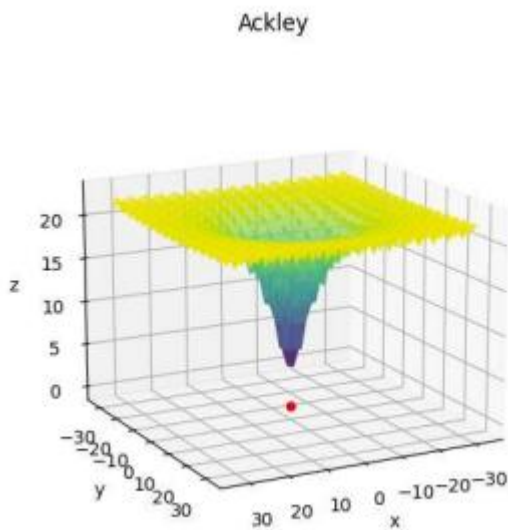
Rysunek 3 STD i Mean - Ackley



Rysunek 4 STD i Mean - Shifted and rotated Ackley

## 5. Omówienie wyników

### Ackley



Globalne minima:  $4.440892098500626e-16$ ,  $[0.0, 0.0]$

Wiele konfiguracji osiągało wynik równy 0,265022461 przy wektorze o wartościach  $[0.25801574803149663, 0.25801574803149663]$  lub  $[0.25801574803149663, -0.25801574803149663]$ , co mimo pewnego błędu względem globalnego optimum (0) oznacza, że rozwiązania są stosunkowo bliskie oczekiwanemu wektorowi  $[0, 0]$  (odległość euklidesowa około 0,365). Z kolei konfiguracje dające wynik funkcji bliższy 0 (np. około 0,043) charakteryzowały się złym wektorem - Best Vector o wartościach rzędu kilkunastu lub kilkudziesięciu, co wskazuje, że mimo lepszej wartości funkcji, rozwiązania te są znacznie oddalone od optimum.

Spośród 108 przetestowanych konfiguracji, jedna wyróżniła się:

```
{
  "var_bounds": [
    -32.768,
    32.768
  ],
  "precision": 8.0,
  "vars_number": 2,
  "pop_size": 100,
  "epochs": 100,
  "selection_method": "best",
  "selection_percentage": 0.5,
  "tournament_size": 3,
```

```
"crossover_method": "two_point",
"crossover_prob": 0.5,
"mutation_method": "boundary",
"mutation_prob": 0.01,
"inversion_prob": 0.015,
"elite_percentage": 0.15,
"objective_function": "<benchmark_functions.benchmark_functions.Ackley
object at 0x000001A5AAC33140>",
"maximize": false,
"population": "<genetic_algorithm.Population.Population object at
0x0000001A5ABAC8150>"
}
```

Osiągnęła w miarę dokładny wynik (0,265022461 przy wektorze [0.25801574803149663, 0.25801574803149663]) o czasie 0,38531208 sekundy. To ustawienie zostało użyte do przedstawienia wykresów z punktów d) i e).

### Shifted and Rotated Ackley

Na podstawie uzyskanych wyników, najlepsze wartości funkcji (najniższy Best Result) osiągają konfiguracje cec2014\_f5\_5ad55d93 (Best Result = 0,001911638) oraz cec2014\_f5\_1747bd75 (Best Result = 0,001911827). Obie konfiguracje charakteryzują się bardzo krótkim czasem obliczeń – odpowiednio około 1,65 sekundy i 1,68 sekundy, co stanowi istotną zaletę przy porównaniu z innymi, gdzie czas wykonania był znacznie dłuższy.

Warto jednak zauważyć, że mimo bardzo zbliżonych i niskich wartości funkcji, wyznaczone przez te konfiguracje wektory rozwiązań znacznie odbiegają od oryginalnego shift data wektora, który dla funkcji Shifted and Rotated Ackley’a wynosi:

[-12.3383, -64.9233, 15.7638, -4.65212, 31.3519, 31.9821, 22.1649, -74.6234, -68.9910, -28.8640].

Oznacza to, że choć funkcja celu jest bardzo bliska optymalnej wartości, pożądane rozwiązanie – czyli wektor odpowiadający przesunięciu – nie zostało uzyskane.



## 6. Tabela porównawcza

Nazwa funkcji	Liczba zmiennych	Rzeczywista wartość optimum	Uzyskane optimum	Błąd
Ackley	2	0	0,265022461	0,265022461
Shifted and rotated Ackley	10	0	~0,001911562	~0,001911562