

# PODSTAWY SZTUCZNEJ INTELIGENCJI

## Grubsze ogony

Kostrzewa Łukasz, Napieralski Adam

23 kwietnia 2020

## 1 Wstęp

Celem projektu było zaimplementowanie algorytmu ewolucyjnego, korzystającego przy mutacji z rozkładu Cauchy'ego zamiast rozkładu normalnego i porównanie działania obu wersji algorytmu. Do formalnej analizy wykorzystano standardowe funkcje, dane testowe i metodę raportowania z CEC 2017 [1]. W tezie zakładamy, że mutacja z wykorzystaniem rozkładu Cauchy'ego pozwala na szybsze dojście do globalnego maksimum niż ta korzystająca z rozkładu normalnego.

### 1.1 Podział zadań

Kostrzewa Łukasz	Napieralski Adam
selekcja, krzyżowanie	mutacja, zastępowanie
Klasa algorytmu i klasa do jego analizy	Próbny przypadek testowy 2D i 3D
Funkcje z CEC2017 (3 pierwsze: implementacja i zastosowanie algorytmu)	Funkcje z CEC2017 (4 ostatnie: implementacja i zastosowanie algorytmu)
Dokumentacja: wykresy, algorytm, analiza, wnioski	Dokumentacja: wyniki tabelaryczne, analiza, wnioski

### 1.2 Wykorzystane narzędzia

Algorytm zaimplementowano w języku *Python*, opierając się przede wszystkim na bibliotece *numpy*. Analizę działania algorytmu z użyciem stworzonych klas realizowano w środowisku *Jupyter Notebook*. Do analizy wyników wykorzystano biblioteki *matplotlib* oraz *tabulate*.

## 2 Algorytm

Schemat wykorzystanego algorytmu ewolucyjnego.

- 
1. Inicjuj  $t := 0, P(0) := \{x_1, \dots, x_\mu\}$  losowe wartości początkowe
  2. for  $i = 1$  to  $\lambda$
  3.     if ( $a < p$ )
  4.          $O(t, i) := \text{mutation}(\text{crossover}(\text{select}(P(t), k)))$
  5.     else
  6.          $O(t, i) := \text{mutation}(\text{select}(P(t), 1))$
  7.  $P(t + 1) = \text{replacement}(P(t), O(t))$
  8. Jeśli warunek stopu jest spełniony zwróć populację  $P(t + 1)$ , w p.p.  $t = t + 1$  i wróć do kroku 2.
-

Gdzie:

- $P(k)$  – populacja w  $k$ -tej iteracji,
- $\mu$  – liczba osobników w populacji,
- $\lambda$  – liczba potomstwa,
- $a$  – zmienna losowa  $\mathcal{U}(0, 1)$ ,
- $p$  – liczba z przedziału  $(0, 1)$  określająca jaka część potomstwa powinna powstać z krzyżowania rodziców,
- $O(k)$  – potomstwo w  $k$ -tej iteracji.

Kroki algorytmu:

1. Selekcja - selekcja turniejowa o rozmiarze turnieju 2.
2. Krzyżowanie - metoda interpolacyjna z wagami losowanymi z  $\mathcal{U}(0, 1)$ .
3. Mutacja - mutacja parametrów z wykorzystaniem rozkładu normalnego (o odchyleniu standardowym równym 1) lub normalnego rozkładu Cauchy’ego [2].
4. Zastępowanie populacji - jako nowa populacja wybierana jest populacja potomna, jeśli najlepszy osobnik ze starej populacji jest lepszy od najgorszego z potomnej, zastępuje go.

### 3 Wyniki działania algorytmów

Oba algorytmy analizowano dla wybranych funkcji z CEC 2017 [1]. Parametry analiz dobrano wzorując się na tych z CEC 2017:

- Obszar przeszukiwania –  $(-50, 50)$  dla każdego wymiaru;
- Wymiary –  $D=2$  oraz  $D=10$ ;
- Maksymalna liczba iteracji –  $1000 \cdot D$ ;
- Warunek stopu – błąd bezwzględny  $\leq 10^{-7}$ ;
- Liczba powtórzeń dla każdego problemu – 51;
- Inicjalizacja – losowe wartości z  $\mathcal{U}(-50, 50)$ .

Końcowe wyniki działania algorytmów w postaci wartości funkcji błędów: minimalnej, maksymalnej, średniej i mediany ich wartości, a także odchylenia standardowego, przedstawiono w Tabelach 1 i 2.

Funkcja	Mutacja	Min	Max	Średnia	Mediana	Odchylenie standardowe
Bent Cigar	Cauchy	$6,31 \cdot 10^{-5}$	0,15	0,03	0,03	0,03
	Normal.	$1,59 \cdot 10^{-4}$	0,02	0,01	$4,21 \cdot 10^{-3}$	0,01
Zakharova	Cauchy	$1,88 \cdot 10^{-6}$	$9,95 \cdot 10^{-4}$	$1,79 \cdot 10^{-4}$	$1,18 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
	Normal.	$8,98 \cdot 10^{-8}$	$2,28 \cdot 10^{-4}$	$4,45 \cdot 10^{-5}$	$3,15 \cdot 10^{-5}$	$4,61 \cdot 10^{-5}$
Rosenbrocka	Cauchy	0,02	0,02	0,02	0,02	0
	Normal	0,02	0,02	0,02	0,02	0
Rastrigina	Cauchy	$1,02 \cdot 10^{-3}$	0,21	0,05	0,03	0,04
	Normal.	$1,96 \cdot 10^{-5}$	0,05	0,01	0,01	0,01
Shaffera	Cauchy	$2,44 \cdot 10^{-5}$	0,01	$9,45 \cdot 10^{-4}$	$3,797 \cdot 10^{-4}$	$1,74 \cdot 10^{-3}$
	Normal.	$9,72 \cdot 10^{-6}$	$9,01 \cdot 10^{-4}$	$2,17 \cdot 10^{-4}$	$1,55 \cdot 10^{-4}$	$2,08 \cdot 10^{-4}$
Levy’ego	Cauchy	$2,53 \cdot 10^{-9}$	$4,58 \cdot 10^{-7}$	$1,04 \cdot 10^{-7}$	$8,11 \cdot 10^{-8}$	$9,98 \cdot 10^{-8}$
	Normal.	$5,26 \cdot 10^{-9}$	$1,21 \cdot 10^{-7}$	$4,98 \cdot 10^{-8}$	$5,09 \cdot 10^{-8}$	$2,58 \cdot 10^{-8}$
Schwefela	Cauchy	$3,75 \cdot 10^{-5}$	0,01	$1,17 \cdot 10^{-3}$	$1,12 \cdot 10^{-3}$	$2,14 \cdot 10^{-3}$
	Normal.	$1,96 \cdot 10^{-5}$	217,14	4,26	$3,95 \cdot 10^{-4}$	30,11

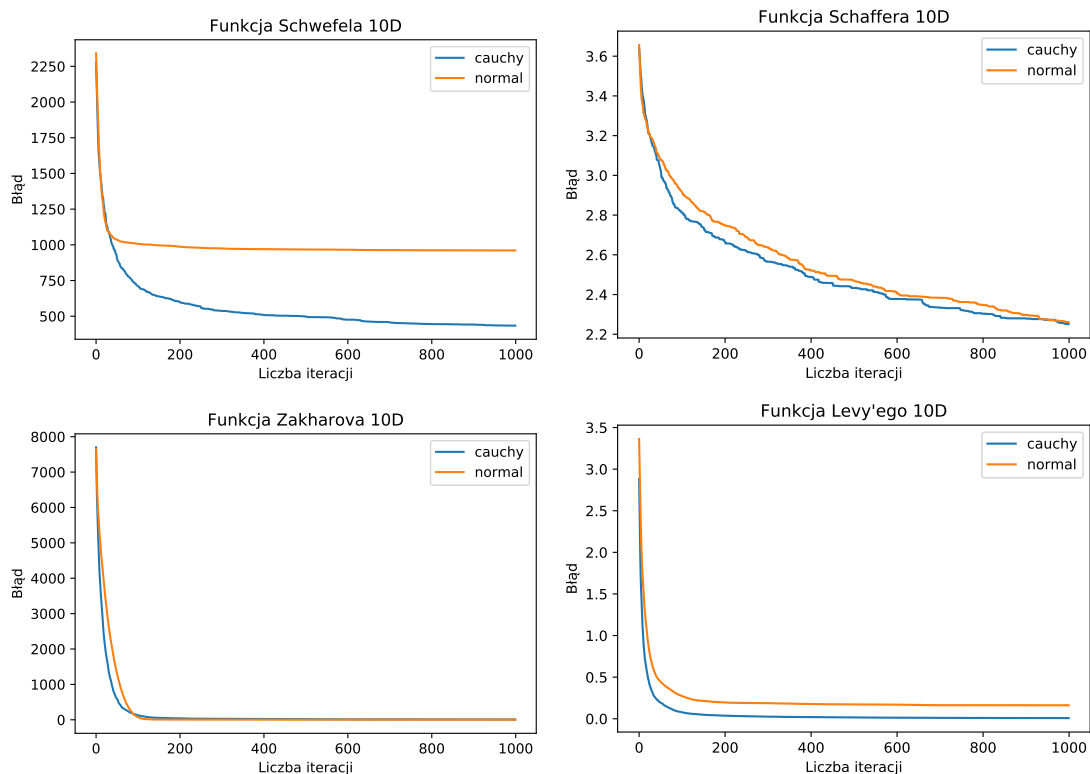
Tabela 1: Podsumowanie uzyskanych wyników dla 2 wymiarów

Funkcja	Mutacja	Min	Max	Średnia	Mediana	Odchylenie standardowe
Bent Cigar	Cauchy	4300,19	28388,8	17331	16784,3	5160,71
	Normal.	3146,44	11916,6	7335,12	7341,45	1984,13
Zakharova	Cauchy	0,18	5,3	3	2,92	0,89
	Normal.	0,54	1,86	1,17	1,19	0,28
Rosenbrocka	Cauchy	5,23	9,68	6,93	7,05	1,13
	Normal	5,2	5,25	5,23	5,22	0,01
Rastrigina	Cauchy	37,61	73,21	53,12	53,16	7,2
	Normal.	22,21	41,19	33,51	33,7	3,92
Shaffera	Cauchy	0,9	2,26	1,64	1,63	0,34
	Normal.	0,49	2,97	1,72	1,73	0,52
Levy'ego	Cauchy	$7,61 \cdot 10^{-4}$	$4,57 \cdot 10^{-3}$	$2,68 \cdot 10^{-3}$	$2,81 \cdot 10^{-3}$	$7,65 \cdot 10^{-4}$
	Normal.	$6,07 \cdot 10^{-4}$	1,73	0,16	$1,13 \cdot 10^{-3}$	0,35
Schwefela	Cauchy	22,91	658,42	260,91	254,09	142,45
	Normal.	227,43	1404,74	909,38	979,91	301,14

Tabela 2: Podsumowanie uzyskanych wyników dla 10 wymiarów

## 4 Wykresy przebiegu funkcji błędu

Poniżej przedstawiono wybrane wykresy funkcji błędu od liczby iteracji.



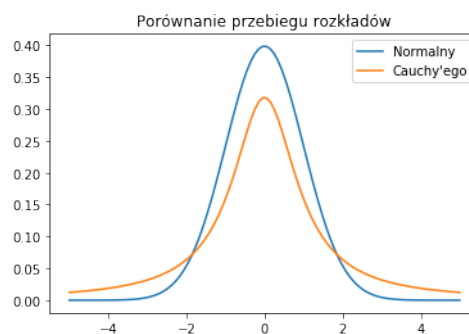
## 5 Analiza wyników i wnioski

Na podstawie przedstawionych danych nie można jednoznacznie określić który algorytm jest uniwersalnie lepszy. W sytuacjach, gdy funkcja przystosowania posiada niewiele ekstremów lokalnych (np. funkcja Rastrigina – Rys. 2), algorytm z rozkładem normalnym jest bardziej odpowiedni, ponieważ jego wyniki są dokładniejsze i bardziej spójne — otrzymane wartości odchylenia standardowego są w niemal wszystkich przypadkach mniejsze niż dla rozkładu

Cauchy'ego. W przypadkach, gdzie funkcja przystosowania posiada wiele ekstremów lokalnych (np. funkcja Schwefela – Rys. 3) algorytm z rozkładem normalnym często zatrzymuje się na ekstremum lokalnym i daje wyniki zdecydowanie gorsze niż algorytm korzystający z rozkładu Cauchy'ego, przez co ten drugi lepiej nadaje się do takich przypadków.

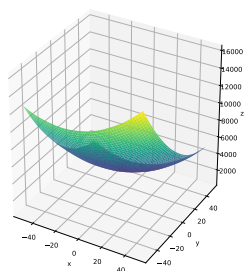
Na podstawie wykresów przebiegu funkcji błędu dla funkcji testowych, można zauważyć, że algorytm korzystający z rozkładu Cauchy'ego na etapie mutacji szybciej odnajduje rozwiązanie.

Algorytm z rozkładem Cauchy'ego lepiej radzi sobie z lokalnymi ekstremami z powodu charakteru tego rozkładu (różnego od normalnego – Rys. 1) w postaci tytułowych "grubszych ogonów". Argumenty bardziej oddalone od przyjętej średniej rozkładu mają nadal dużo większą wartość gęstości prawdopodobieństwa niż te same argumenty w przypadku rozkładu normalnego. Prawdopodobieństwo wylosowania wartości bardziej oddalonych od średniej rozkładu jest dużo większe niż w przypadku rozkładu normalnego. Umożliwia to częstsze takie mutacje osobników, by znalazły się one dalej od rodziców, a co za tym idzie - w przypadku "utknięcia" w lokalnym ekstremum, miały większe szanse wyjścia poza nie i znalezienia interesującego globalnego ekstremum.



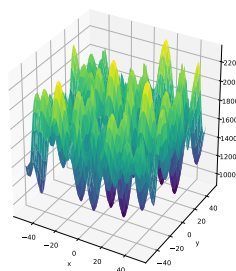
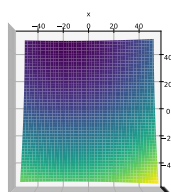
Rys. 1: Porównanie przebiegu rozważanych rozkładów

Funkcja Rastrigina



Rys. 2: Wykres funkcji Rastrigina

Funkcja Schwefela



Rys. 3: Wykres funkcji Schwefela

Realizacja projektu pomogła nam zrozumieć i poznać od praktycznej strony szczegóły każdego z kroków algorytmu ewolucyjnego, a w szczególności mutacji.

## Literatura

- [1] <https://github.com/P-N-Suganthan/CEC2017-BoundConstrained>
- [2] [https://numpy.org/devdocs/reference/random/generated/numpy.random.standard\\_cauchy.html](https://numpy.org/devdocs/reference/random/generated/numpy.random.standard_cauchy.html)
- [3] <https://staff.elka.pw.edu.pl/~jarabas/ALHE/wyklad6.pdf>