PODSTAWY SZTUCZNEJ INTELIGENCJI

Grubsze ogony

Kostrzewa Łukasz (285753), Napieralski Adam (285731)

21 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 Klasa algorytmu i klasa do jego analizy Funkcje z CEC2017 (3 pierwsze: implementacja i zastosowanie algorytmu) Dokumentacja: wykresy, algorytm, analiza, wnioski	mutacja, zastępowanie Próbny przypadek testowy 2D i 3D Funkcje z CEC2017 (4 ostatnie: implementacja i zastosowanie algorytmu) 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 $t := 0, P(0) := \{x_1, ...x_\mu\}$ *losowe wartości początkowe*
2. for i = 1 to λ
3.   if($a < p$)
4. $O(t, i) := mutation(crossover(select(P(t), k)))$
5.   else
6. $O(t, i) := mutation(select(P(t), 1))$
7. $P(t + 1) = replacement(P(t), O(t))$
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,
μ – liczba osobników w populacji,
λ – 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 liczna iteracji – 1000*D;
Warunek stopu – błąd bezwzględny $\leqslant 10^{-7}$;
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	$6{,}305 \cdot 10^{-5}$	0,151	0,035	0,027	0,033
	Normal.	$1{,}592 \cdot 10^{-4}$	0,022	0,007	0,004	0,007
Zakharova	Cauchy	$1{,}881 \cdot 10^{-6}$	$9{,}945 \cdot 10^{-4}$	$1{,}785 \cdot 10^{-4}$	$1{,}175 \cdot 10^{-4}$	$2{,}003 \cdot 10^{-4}$
	Normal.	$8{,}978 \cdot 10^{-8}$	$2{,}284 \cdot 10^{-4}$	$4{,}446 \cdot 10^{-5}$	$3{,}153 \cdot 10^{-5}$	$4{,}61 \cdot 10^{-5}$
Rosenbrocka	Cauchy	0,02	0,02	0,02	0,02	0
	Normal	0,02	0,02	0,02	0,02	0
Rastrigina	Cauchy	0,001	0,209	0,047	0,03	0,044
	Normal.	$1{,}955 \cdot 10^{-5}$	0,045	0,012	0,01	0,012
Shaffera	Cauchy	$2{,}439 \cdot 10^{-5}$	0,009	$9{,}453 \cdot 10^{-4}$	$3{,}797 \cdot 10^{-4}$	0,002
	Normal.	$9{,}717 \cdot 10^{-6}$	$9{,}014 \cdot 10^{-4}$	$2{,}165 \cdot 10^{-4}$	$1{,}547 \cdot 10^{-4}$	$2{,}076 \cdot 10^{-4}$
Levy'ego	Cauchy	$2{,}53 \cdot 10^{-9}$	$4{,}58 \cdot 10^{-7}$	$1{,}039 \cdot 10^{-7}$	$8{,}108 \cdot 10^{-8}$	$9{,}977 \cdot 10^{-8}$
	Normal.	$5{,}259 \cdot 10^{-9}$	$1{,}21 \cdot 10^{-7}$	$4{,}98 \cdot 10^{-8}$	$5{,}09 \cdot 10^{-8}$	$2{,}575 \cdot 10^{-8}$
Schwefela	Cauchy	$3{,}751 \cdot 10^{-5}$	0,014	0,002	0,001	0,002
	Normal.	$1{,}955 \cdot 10^{-5}$	217,14	4,2581	$3{,}95 \cdot 10^{-4}$	30,106

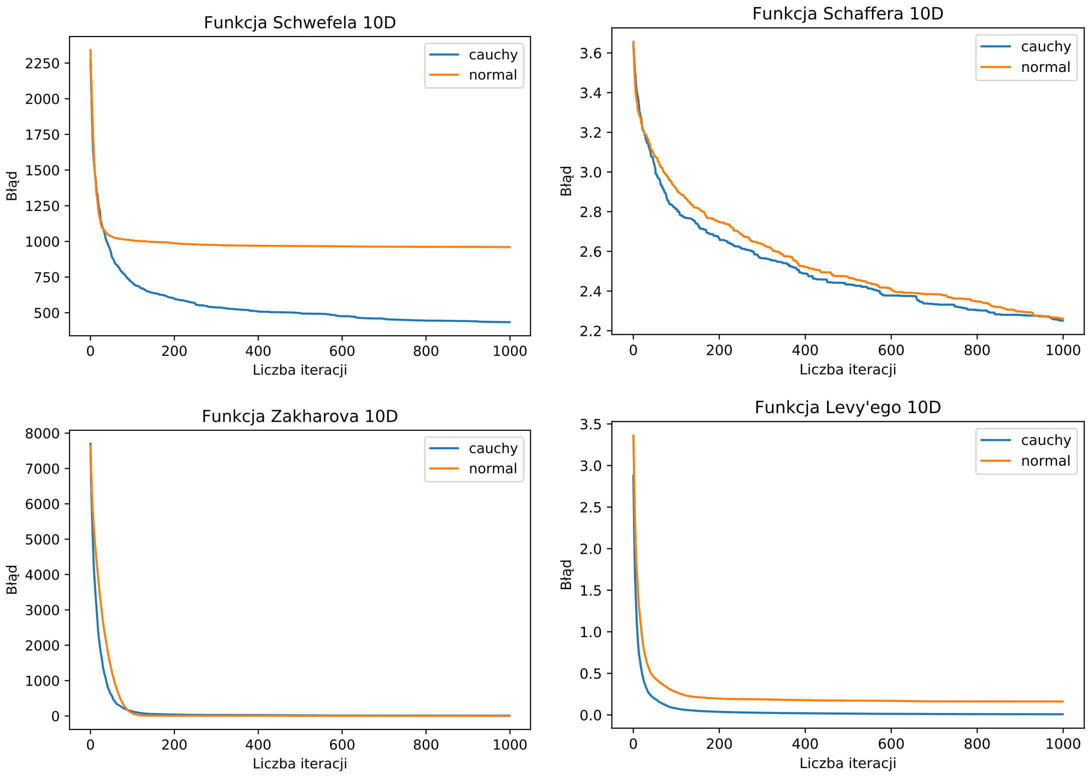
Tabela 1: Podsumowanie uzyskanych wyników dla 2 wymiarów

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	0,184	5,297	2,995	2,921	0,892
	Normal.	0,54	1,862	1,167	1,185	0,282
Rosenbrocka	Cauchy	5,234	9,681	6,93	7,045	1,133
	Normal	5,202	5,252	5,225	5,224	0,0112
Rastrigina	Cauchy	0,001	0,209	0,047	0,03	0,044
	Normal.	$1{,}955 \cdot 10^{-5}$	0,045	0,012	0,01	0,012
Shaffera	Cauchy	37,613	73,213	53,118	53,163	7,202
	Normal.	22,215	41,186	33,509	33,702	3,917
Levy'ego	Cauchy	$7{,}612 \cdot 10^{-4}$	0,005	0,003	0,003	$7{,}646 \cdot 10^{-4}$
	Normal.	$6{,}067 \cdot 10^{-4}$	1,73	0,157	0,001	0,35
Schwefela	Cauchy	22,913	658,417	260,908	254,094	142,454
	Normal.	227,428	1404,74	909,376	979,908	301,141

Tabela 2: Podsumowanie uzyskanych wyników dla 10 wymiarów

4 Wykresy przebiegu funkcji błędu

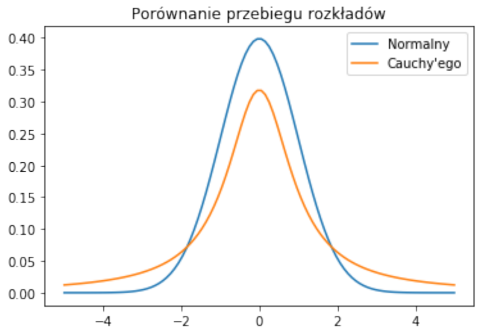
Poniżej przedstawiono wybrane wykresy funkcji błędu od liczby iteracji.



5 Analiza wyników i wnioski

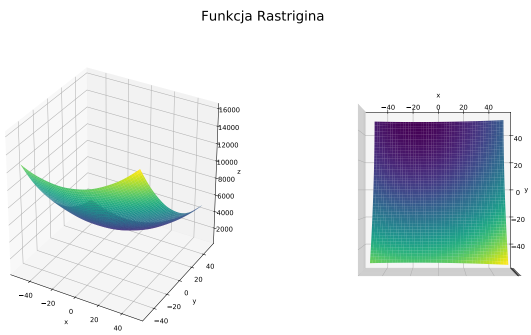
Na podstawie wykresów przebiegu funkcji błędu dla funkcji testowych, można zauważyć, że algorytm korzystający z rozkładu Cauchy'ego na etapie mutacji szybciej odnajduje rozwiązanie. Dodatkowo z danych tabelarycznych wynika, że w niektórych przypadkach użycie przyjętego rozkładu normalnego nie pozwala na znalezienie globalnego minimum i zatrzymuje się na lokalnym, podczas gdy algorytmowi z rozkładem Cauchy'ego udaje się przez nie przejść i znaleźć oczekiwane rozwiązanie.

Wyjaśnieniem
jest najpewniej charakter rozkładu Cauchy'ego (różniący go 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.

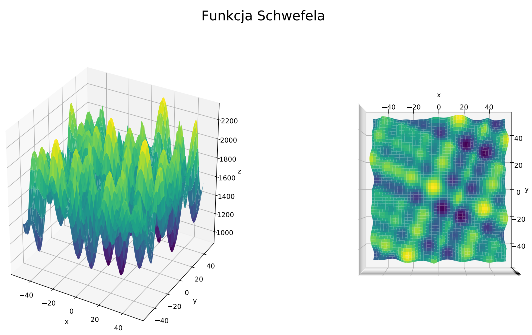


Rys. 1: Porównanie przebiegu rozważanych rozkładów

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.



Rys. 2: Wykres funkcji Rastrigina



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-BoundContrained`

[2] `https://numpy.org/devdocs/reference/random/generated/numpy.random.standard_cauchy.html`

[3] `https://staff.elka.pw.edu.pl/~jarabas/ALHE/wyklad6.pdf`