

Mechanizm wymarcia w algorytmie CMA-ES

Dokumentacja końcowa

Szymon Stasiak

Przemysław Woźniakowski

28 września 2022

Streszczenie

Poniższy raport opisuje projekt, którego celem było przebadanie działania mechanizmu wymarcia populacji w algorytmie CMA-ES. W tym celu zaproponowany został sposób przeprowadzenia opartego na stagnacji mechanizmu wymarcia, wraz z dwoma jego wariantami. Zaproponowane warianty to wymieranie losowe, które polegało na odrzucaniu dowolnych punktów z populacji z pewnym prawdopodobieństwem, oraz wymieranie ukierunkowane, które wyłączało ze zbioru kandydatów do odrzucenia pewien odsetek najlepszych punktów. W ramach raportu opisane zostały eksperymenty, które miały na celu wyznaczyć optymalne parametry dla mechanizmu wymierania, czyli prawdopodobieństwo wymarcia oraz licznik stagnacji algorytmu, który decyduje kiedy dokonywane jest wymarcie. Dodatkowo opisane zostały eksperymenty na kilku przykładowych funkcjach ze zbioru CEC 2022. Pokazały one, że mechanizm ukierunkowanego wymarcia poprawił uśrednioną skuteczność dla funkcji takich jak funkcja Schaffera i Rastrigina, nie dał jednak żadnych pozytywnych efektów jeśli chodzi o funkcję Levy’ego.

Spis treści

1	Opis problemu	2
2	Opis algorytmu	2
2.1	Algorytm CMA-ES	2
2.2	Mechanizm wymarcia	3
2.2.1	Wymarcie losowe	3
2.2.2	Wymarcie ukierunkowane	3
2.3	Schemat algorytmu	3
2.4	Używane symbole i ich definicja	4
3	Technologia	4
3.1	Zbiory testowe	4
4	Eksperymenty	5
4.1	Wybór wielkości populacji	5
4.2	Wybór prawdopodobieństwa i progu wymarcia	8
5	Analiza wyników na pozostałych funkcjach ze zbioru CEC 2022	13
5.1	<i>Shifted and full Rotated Expanded Schaffer’s f6 function</i>	13
5.2	<i>Shifted and full Rotated Non-Continuous Rastrigin’s Function</i>	15
5.3	<i>Shifted and Rotated Levy Function</i>	16
6	Podsumowanie	18

1 Opis problemu

Tematem pracy jest połączenie dwóch konceptów stosowanych w algorytmach ewolucyjnych - algorytmu CMA-ES oraz mechanizmu wymarcia.

Nazwa algorytmu CMA-ES jest akronimem angielskiej nazwy Covariance Matrix Adaptation - Evolution Strategy. Na polski język można przetłumaczyć to jako strategię ewolucyjną bazującą na adaptacji macierzy kowariancji. Twórcami tego algorytmu są N. Hansen i A. Ostermaier, którzy w 2001 roku zamieścili jego opis w pracy *Completely derandomized self-adaptation in evolution strategies* [1].

Algorytm CMA-ES skupia się przede wszystkim na sposobie generowania osobników w kolejnych pokoleniach (etapie selekcji). Otwiera to drogę do modyfikowania innych etapów. Jedną z takich modyfikacji jest prezentowane w tej pracy rozszerzenie o mechanizm wymarcia.

Sam w sobie mechanizm wymarcia w algorytmach ewolucyjnych jest konceptem znanym, jednak często uznawanym za nieznaczący. Niemniej jednak istnieją badania pokazujące korzystny wpływ tego konceptu na osiągnięte rezultaty. Jedną z takich prac jest powstały w 2021 roku artykuł autorstwa Gan Zhen Ye oraz Dae-Ki Kang *Extended Evolutionary Algorithms with Stagnation-Based Extinction Protocol* [2]. Artykuł ten opisuje dwie implementacje oparte o stagnację mechanizmu wymarcia - wymarcie losowe oraz wymarcie ukierunkowane. Właśnie te dwie wersje zostały zaimplementowane i przeanalizowane w obrębie pracy. Głównym celem badania było zweryfikowanie, czy powszechna opinia o nieznaczącym wpływie na mechanizm wymarcia jest zasadna w przypadku CMA-ES, czy może jest to obiecujący kierunek rozwoju.

2 Opis algorytmu

Zaproponowany algorytm łączy w sobie implementację CMA-ES oraz proponowanych realizacji mechanizmów wymarcia. Dlatego też niezbędne jest zrozumienie obu tych części. Kolejne podrozdziały opisują właśnie wspomniane elementy a także używane w nich symbole (sekcja 2.4). Wreszcie, rozdział 2.3 zawiera pseudokod finalnego algorytmu, gdzie elementy wynikające z implementacji mechanizmu wymarcia zostały zaznaczone kolorem niebieskim.

2.1 Algorytm CMA-ES

Podobnie jak w wielu innych algorytmach ewolucyjnych, koncepcja CMA-ES opiera się na losowaniu kolejnych populacji z wielowymiarowym rozkładem normalnym. W każdej iteracji generowany jest tu jednak zupełnie nowy zestaw potomków. Proces ten dla generacji $t + 1$ przebiega według równania:

$$x_k^{(t+1)} \sim m^{(t)} + \sigma^{(t)} N(0, C^{(t)})$$

gdzie \sim oznacza generowanie wartości zmiennej losowej według danego rozkładu. Jak widać algorytm różni się od ogólnego schematu algorytmu ewolucyjnego tym, że każdy osobnik nie posiada jednoznacznie określonych rodziców.

Rozważając pewną określoną populację o λ osobnikach, w procesie selekcji wybrane zostaje μ najlepszych punktów pod względem wartości funkcji celu. Punkty te biorą udział w obliczaniu wartości oczekiwanej rozkładu kolejnej iteracji, zgodnie ze wzorem:

$$m^{(t+1)} = \sum_{i=1}^{\mu} w_i x_{i+\lambda}^{(t+1)}$$

Rozkład wag proponowany przez twórców algorytmu to: $w_i = \log(\mu + 1) - \log(i)$, który poddany zostaje normalizacji tak żeby wagi spełniały warunek:

$$\sum_{i=1}^{\mu} w_i = 1$$

Adaptacja macierzy kowariancji jest najważniejszym elementem algorytmu CMA-ES. Dzięki niej, każda kolejna iteracja używa rozkładu potencjalnie lepiej dostosowanego do otoczenia, niż iteracje poprzednie. Macierz kowariancji w przypadku bardzo dużej populacji i przy wykorzystaniu jako punkt odniesienia wartości oczekiwanej rozkładu z poprzedniej iteracji, przedstawiona jest wzorem:

$$C_{\mu}^{(t+1)} = \sum_{i=1}^{\mu} w_i (x_{i:\lambda}^{(t+1)} - m^{(t)}) (x_{i:\lambda}^{(t+1)} - m^{(t)})^T$$

Korzystanie z tego wzoru jest jednak problematyczne, gdyż znacznie korzystniejsze jest operowanie na mniej licznej populacji, ponieważ pozwala to na wykonanie znacznie większej liczby iteracji przy takiej samej liczbie

obliczeń funkcji celu. W związku z tym wykorzystywana jest ścieżka ewolucji p_c czyli ważona suma wektorów, łączących wartości oczekiwane rozkładów kolejnych generacji algorytmu. Wzór na macierz kowariancji wykorzystujący tę ścieżkę ma postać:

$$C^{(t+1)} = (1 - c_1 - c_\mu)C^{(t)} + c_1 p_c p_c^T + c_\mu \sum_{i=1}^\mu w_i y^{(t+1)} (y^{(t+1)})^T$$

gdzie wszystkie symbole opisane zostały w 2.4. Ostatnim kluczowym parametrem w algorytmie jest długość kroku δ . Pozwala ona dostosować zasięg mutacji do fazy w której znajduje się algorytm. Sposób jej obliczenia przedstawiony został w schemacie algorytmu w sekcji 2.3 w liniach 20 i 21 [3].

2.2 Mechanizm wymarcia

Mechanizm wymarcia opartego na stagnacji bazuje na odrzuceniu części populacji, jeżeli ta pozostaje w stagnacji przez zbyt dużą liczbę kolejnych pokoleń. W zaproponowanej implementacji, poprzez stagnację rozumie się bardzo zbliżone do siebie wartości funkcji celu dla najlepszych osobników w kolejnych pokoleniach. Do określenia bliskości wspomnianych wartości funkcji, wykorzystany jest stosunek wartości bezwzględnej ich różnicy do wartości funkcji w poprzednim pokoleniu. Taki sposób porównania pozwala uniezależnić tę część algorytmu od osiąganych wartości funkcji celu. Jeżeli opisany stosunek w dwóch kolejnych pokoleniach do będzie mniejszy od zadanej stałej C_e , zwiększany zostaje licznik. W przeciwnym wypadku, jest on zerowany. Cały warunek mówi więc „jeżeli w stosunku do poprzedniej generacji, najmniejsza wartość funkcji zmieniła się maksymalnie o C_e (procent), populacja jest w stagnacji”. Gdy ilość kolejnych populacji w stagnacji przekroczy określony próg K , część populacji zostaje odrzucona (wymiera), zgodnie z określonym algorytmem. Schemat mechanizmu wymarcia zawarty jest w liniach 13 - 18 pseudokodu w sekcji 2.3.

Algorytm wyboru osobników do wymarcia różni się w obu proponowanych podejściach.

2.2.1 Wymarcie losowe

Podejście to nie zwraca uwagi na wartość funkcji celu dla danych osobników. Każdy z nich jest traktowany sprawiedliwie i zostaje usunięty z populacji z zadaniem prawdopodobieństwem p_e w zależności od pewnej realizacji zmiennej losowej.

2.2.2 Wymarcie ukierunkowane

W tej metodzie, wymarcie ukierunkowane jest w stronę odrzucenia z populacji osobników mniej przystosowanych. W praktyce oznacza to wprowadzenie nowego parametru k , który informuje jak duża część populacji o najlepszych wynikach powinna zawsze pozostać niedotknięta wymarciem. W realizacji całego procesu najpierw wybierana jest wartość funkcji celu dla $[k \cdot \lambda]$ -tego najlepszego osobnika. Po oznaczeniu tej wartości przez T , osobniki dla których wartość funkcji $y_i < T$ usuwane są z populacji z zadaniem prawdopodobieństwem p_e .

2.3 Schemat algorytmu

Algorytm 1 CMA - ES z mechanizmem wymarcia

- 1: Inicjacja parametrów zgodnie z sekcją 2.4
- 2: $t \leftarrow 0$
- 3: **while** kryterium stopu nie spełnione **do**
- 4: Losowanie punktów dla $k = 1, \dots, \lambda$:
- 5: $z_k^{(t+1)} \sim N(0, C^{(t)})$
- 6: Obliczenie wartości punktów:
- 7: $y_k^{(t+1)} = BDz_k^{(t+1)}$
- 8: $x_k^{(t+1)} = m^{(t)} + \sigma^{(t)} y_k^{(t+1)}$
- 9: Wstępna selekcja i obliczenie nowej wartości oczekiwanej rozkładu:
- 10: $\langle y_k^{(t+1)} \rangle = \sum_{i=1}^\mu w_i y_{i:\lambda}$
- 11: $m^{(t+1)} \leftarrow m^{(t)} + \sigma^{(t)} \langle y \rangle$
- 12: **Wymarcie:**
- 13: **if** $\frac{|y_{1:\lambda}^{(t+1)} - y_{1:\lambda}^{(t)}|}{y_{1:\lambda}^{(t)}} \leq C_e$:
- 14: **if** $s_c + + \geq K$:
- 15: Odrzuć część populacji zgodnie z metodą wymarcia i zaktualizuj $y_k^{(t+1)}$ oraz $m^{(t+1)}$
- 16: Zaktualizuj parametry λ, μ oraz w dla zredukowanej populacji
- 17: **else:**
- 18: $s_c = 0$

19: Kontrola długości kroku:
20: $p_{\sigma}^{(t+1)} \leftarrow (1 - c_{\sigma})p_{\sigma}^{(t+1)} + \sqrt{c_{\sigma}(2 - c_{\sigma})\mu_{eff}}C^{(t)-\frac{1}{2}}\langle y^{(t+1)} \rangle$
21: $\sigma^{(t+1)} \leftarrow \sigma^{(t)} \times \left(c_{\sigma} \left(\frac{\|p_{\sigma}^{(t+1)}\|}{E\|N(0, I)\|} - 1 \right) \right)$
22: Adaptacja macierzy kowariancji:
23: $p_c^{(t+1)} \leftarrow (1 - c_c)p_c^{(t+1)} + \sqrt{c_c(2 - c_c)\mu_{eff}}\langle y^{(t+1)} \rangle$
24: $C^{(t+1)} \leftarrow (1 - c_1 - c_{\mu})C^{(t)} + c_1p_cp_c^T + c_{\mu}\sum_{i=1}^{\mu}w_iy^{(t+1)}(y^{(t+1)})^T$
25: $t \leftarrow t + 1$

2.4 Używane symbole i ich definicja

Symbol	Definicja	Wartość początkowa
n	Liczba wymiarów	-
t	Numer iteracji algorytmu	0
m	Wartość oczekiwana rozkładu	Losowa
C	Macierz kowariancji rozkładu	I
σ	Długość kroku	0.5
λ	Rozmiar populacji	$\lambda = 4 + 3 \ln n$?
μ	Liczba osobników podlegających selekcji	$\mu = \frac{\lambda}{2}$
p_C	Ścieżka ewolucji znormalizowana do macierzy kowariancji C	$p_C = 0$
p_{σ}	Ścieżka ewolucji znormalizowana do jednostkowej macierzy kowariancji C	$p_{\sigma} = 0$
w	Wektor wag o długości μ	$w_i = \log(\mu + 1) - \log(i)$ $w_i = \frac{w_i}{\sum_{i=1}^{\mu} w_i}$
μ_{eff}	Parametr charakteryzujący wagi selekcji	$\frac{1}{\sum_{i=1}^{\mu} w_i^2}, 1 \leq \mu_{eff} \leq \mu$
c_C	Stała ścieżki ewolucji p_c	$c_C = \frac{4}{n+4}$
c_{σ}	Stała długości kroku σ	$c_{\sigma} = 0.1$
x_i	i -ty osobnik z populacji	-
$x_{i:\lambda}$	i -ty najlepszy osobnik z populacji spośród λ wszystkich	-
y_i	Wektor łączący i -ty punkt populacji z wartością oczekiwaną rozkładu	-
$\langle y \rangle$	Zbiór wektorów łączących punkty populacji z wartością oczekiwaną rozkładu	-
C_e	Zmiana minimalnej wartości funkcji celu w kolejnych pokoleniach, określająca czy algorytm pozostaje w stagnacji	$C_e = 0.05 = 5\%$
s_c	Licznik od ilu pokoleń algorytm pozostaje w stagnacji	$s_c = 0$
K	Liczba pokoleń w stagnacji po których następuje wymarcie	?
p_e	Prawdopodobieństwo wymarcia	?
k	Parametr określający jaka część $k \in (0, 1)$ najlepszych osobników w populacji powinna zostać pominięta w procesie wymarcia ukierunkowanego	$k = 0.2$

3 Technologia

Analizowane algorytmy i testy zostały zaimplementowane w środowisku Matlab. Środowisko to pozwoliło na łatwe wykorzystanie i modyfikację algorytmu CMA-ES na podstawie źródeł udostępnionych przez autorów algorytmu [4]. Dodatkowo, dostępne funkcjonalności oraz bardzo duża dostępność przykładowych zbiorów testowych umożliwiły wygodną analizę i interpretację wyników.

3.1 Zbiory testowe

W ramach eksperymentów przebadane zostały funkcje wielowymiarowe, dla wymiarowości powyżej 10. Szczególną uwagę w raporcie poświęcono dwóm konkretnym wymiarowościom - 10 oraz 20. Podejście to zostało

wybrane przede wszystkim ze względu na wykorzystane funkcje ze zbioru CEC [5] dostępne właśnie w tych wariantach.

Funkcjami, które zostały przebadane są:

- Funkcja sferyczna - klasyczna funkcja okazała się dobrym początkiem, jednak nie stanowiła problemów dla żadnego wykorzystanego algorytmu.
- Funkcja Ackleya - podobnie jak w przypadku funkcji sferycznej, znalezienie minimum globalnego nie stanowiło problemów.
- Funkcja Rosenbrocka - zwana również „doliną Rosenbrocka” lub „funkcją bananową”, jej minimum globalne znajduje się w długim, parabolicznym wgłębieniu funkcji. W testach okazała się dobrym przykładem poprawnego działania algorytmów.
- Funkcja Rastrigina - funkcja o wielu minimach lokalnych, co okazało się dużym problemem dla prezentowanych algorytmów.
- Funkcje ze zbioru **CEC 2022** *Special Session and Competition on Single Objective Bound Constrained Numerical Optimization* [5] - wymagające funkcje, w dobry sposób uwidaczniające różnice między implementacjami.

4 Eksperymenty

Pierwszą częścią eksperymentów, przeprowadzoną jeszcze przed głównym porównaniem metod, było dobranie odpowiednich parametrów wymarcia. W tym celu obie implementacje zawierające ten mechanizm zostały najpierw kilkakrotnie wywołane dla tych samych zbiorów testowych, jednak z różnymi wartościami parametrów K oraz p_e . Następnie wyniki tych testów zostały ze sobą porównane w celu ustalenia najbardziej optymalnych parametrów dla każdego podejścia. Dodatkowo przeanalizowany został również wpływ wielkości populacji λ na osiągnięte wyniki.

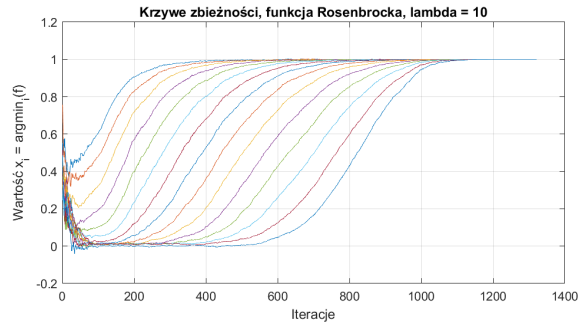
4.1 Wybór wielkości populacji

Jedną z głównych zalet algorytmu CMA-ES, poza osiąganymi rezultatami, jest brak parametrów wymagających dokładnej konfiguracji. Zgodnie ze stanowiskiem autorów publikacji [1], jedynym parametrem, którego modyfikację można potencjalnie rozważyć, jest wielkość populacji. Autorzy sugerują dobranie tej wartości zgodnie ze wzorem

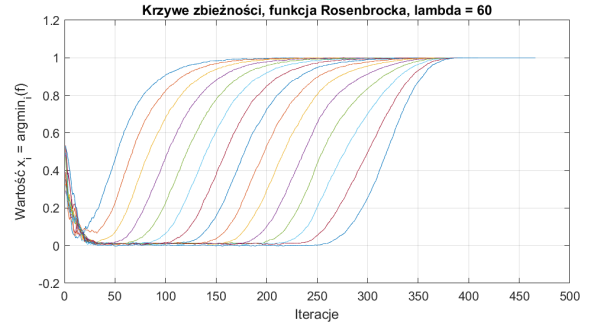
$$\lambda = 4 + \lfloor 3\log(N) \rfloor,$$

gdzie N oznacza wymiarowość problemu. W przypadku zadań o wymiarowości poniżej 20 (jak na przykład w zbiorze zadań CEC [5]), daje to wielkość populacji nieprzekraczającą 13. Ograniczenie to zdaje się być bardzo małe, w szczególności przy rozważaniu mechanizmu wymierania, który dodatkowo będzie redukował wielkość populacji. Z tego też powodu, eksperymenty rozpoczęto od dostosowania podstawowej wersji algorytmu CMA-ES.

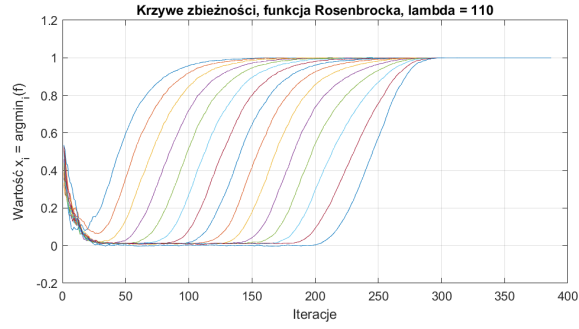
W tym celu, algorytm w wersji podstawowej został uruchomiony dla zmiennej wielkości populacji, z przedziału $\lambda \in \{10, 60, 110, \dots, 410\}$, czyli 9 różnych wartości, rozpoczynając od wartości zbliżonej sugerowanej, z krokiem między porównywanymi rozmiarami równym 50. Badanie przeprowadzono dla 15-wymiarowych funkcji Rosenbrocka oraz Rastrigina, ograniczając działanie algorytmu maksymalną liczbą ewaluacji funkcji zgodnie z sugestiami twórców algorytmu, czyli $stopeval = N^2 * 10^3 = 4 * 10^5$. Dla każdej funkcji, badania przeprowadzono dla 15 różnych ziaren generatora liczb pseudolosowych, wybierając przy tym średnią z uzyskanych wyników. Krzywe zbieżności dla wspomnianych funkcji przedstawiono na rysunkach 1 oraz 2



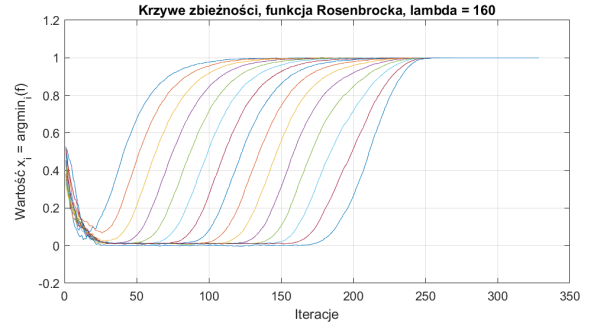
(a) $\lambda = 10$



(b) $\lambda = 60$



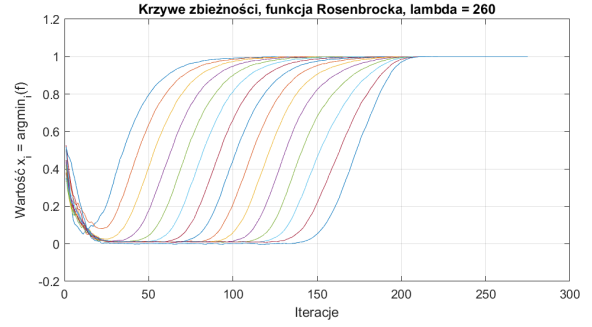
(c) $\lambda = 110$



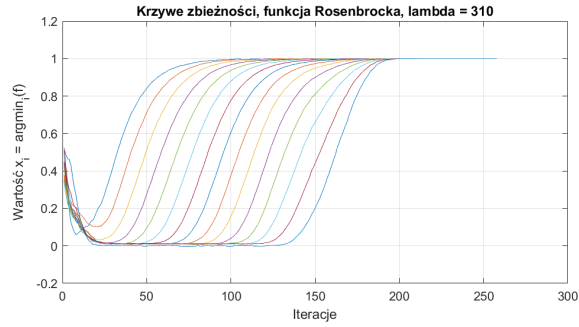
(d) $\lambda = 160$



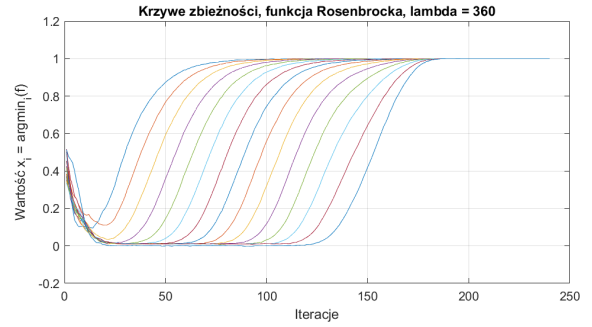
(e) $\lambda = 210$



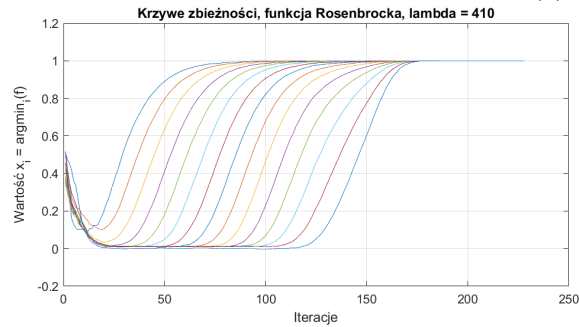
(f) $\lambda = 260$



(g) $\lambda = 310$

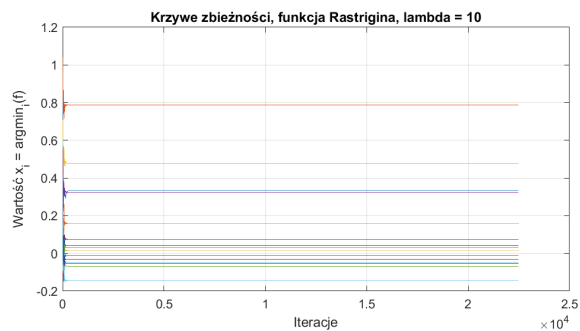


(h) $\lambda = 360$

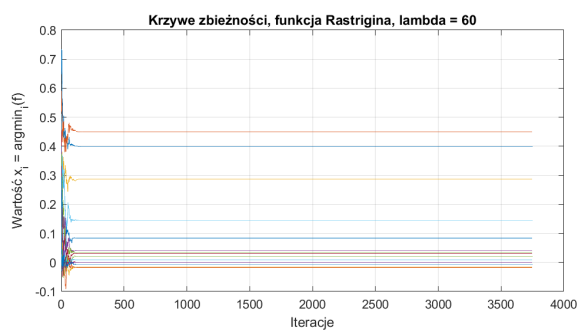


(i) $\lambda = 410$

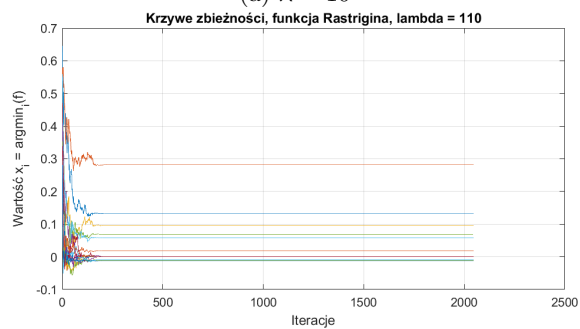
Rysunek 1: Krzywe zbieżności algorytmu CMA-ES dla funkcji Rosenbrocka przy różnych wielkościach populacji



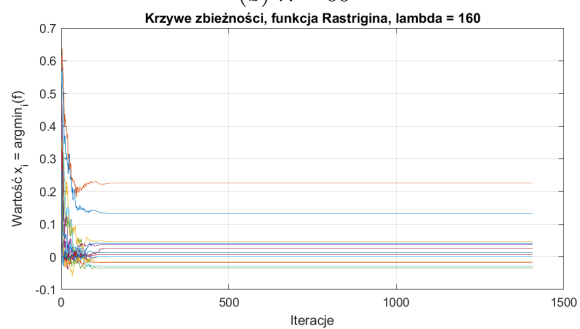
(a) $\lambda = 10$



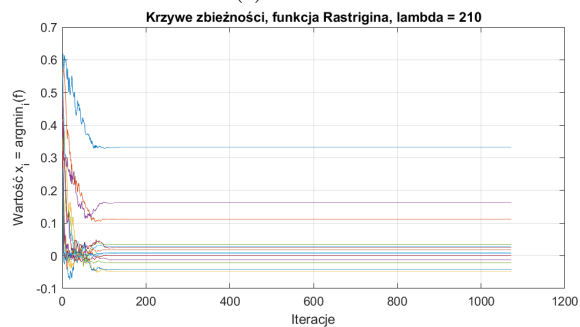
(b) $\lambda = 60$



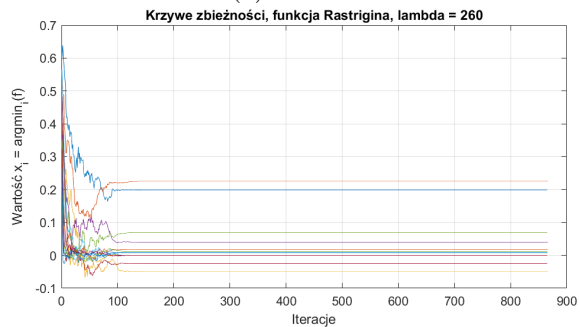
(c) $\lambda = 110$



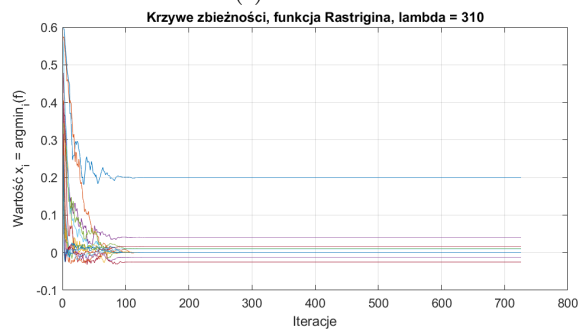
(d) $\lambda = 160$



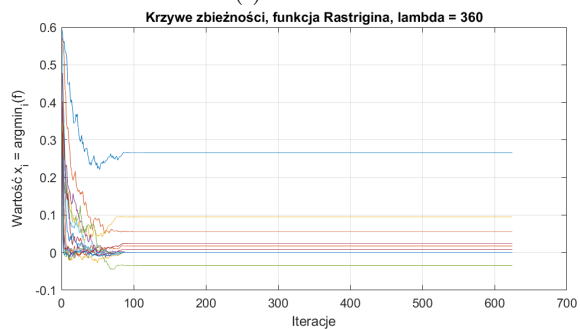
(e) $\lambda = 210$



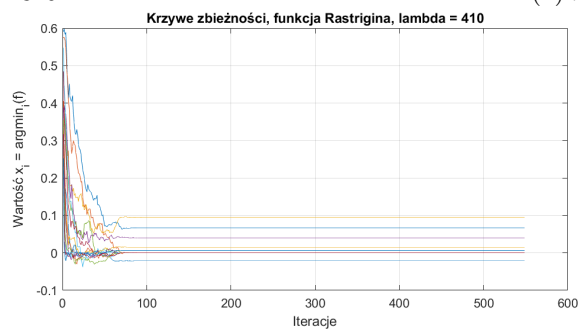
(f) $\lambda = 260$



(g) $\lambda = 310$



(h) $\lambda = 360$



(i) $\lambda = 410$

Rysunek 2: Krzywe zbieżności algorytmu CMA-ES dla funkcji Rastrigina przy różnych wielkościach populacji

Jak widać, pierwsza z funkcji, zbiegała do faktycznego minimum w punkcie o każdej współrzędnej równej 1, dla wszystkich sprawdzanych wartości λ . Sytuacja prezentuje się jednak zupełnie inaczej dla funkcji Rastrigina. W jej przypadku, globalne minimum (w punkcie o współrzędnych równych 0), nie jest osiągnięte dla żadnej wartości parametru. Jednak dla trzech ostatnich wartości parametru $\lambda \in \{310, 360, 410\}$, wynik był zdecydowanie bliższy. W celu wybrania najlepszej spośród tych wartości, przeprowadzono również analizę dla algorytmu z wymierzaniem oraz analizę dystrybucji uzyskiwanych rezultatów. Jednak jako, iż nie był to główny temat pracy, podjęto decyzję o pominięciu wykresów w opracowaniu. Niemniej, okazało się, że najlepsze efekty spośród testowanych wartości, były osiągane dla wartości $\lambda = 310$.

Ponownie, zdając sobie sprawę z niedokładności, ale również z drugorzędnego charakteru tego badania, porzeczano na przyjęciu $\lambda = 310$ do dalszych testów.

4.2 Wybór prawdopodobieństwa i progu wymarcia

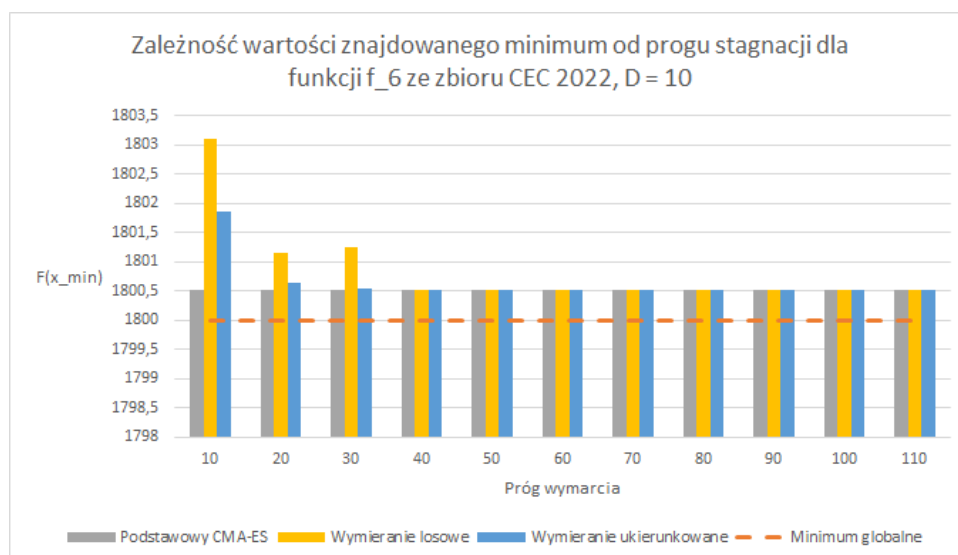
Kolejnymi badanymi parametrami były już parametry dotyczące bezpośrednio mechanizmu wymarcia, czyli prawdopodobieństwo oraz próg wymarcia. Prawdopodobieństwo wymarcia określa szansę z którą dany osobnik populacji zostaje z niej usunięty w procesie wymarcia losowego lub ukierunkowanego. Jednocześnie, w tym drugim przypadku odnosi się to wyłącznie do osobników niespełniających kryterium k najlepszych wyników. Na potrzeby testów, przyjęto sugerowaną w artykule [2] wartość $k = 0.2$, gdyż okazała się ona dawać optymalne wyniki. Progu wymarcia, to z kolei liczba pokoleń przez ile algorytm musi pozostawać w stagnacji, aby uruchomić mechanizm wymarcia.

Zgodnie z intuicją oraz wskazówkami z artykułu [2], można przypuszczać wzajemną zależność obu tych parametrów. Z tego powodu, testy zostały przeprowadzone dla różnych kombinacji jednocześnie. Tak też badania zostały przeprowadzone dla $K \in \{10, 20, 30, \dots, 110\}$ oraz $p_e \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Nie lada zadaniem było dobranie funkcji, dla których różne wartości parametrów powodowały widoczne, nielosowe zmiany. Z tego powodu, po przeanalizowaniu wszystkich funkcji ze zbioru CEC [5], uznano za najciekawsze rezultaty uzyskane w przypadku analizy funkcji numer 6 ze wspomnianego zbioru (dla wymiarowości równej 10 i 20), jak też 20-wymiarowej funkcji Rastrigina. Ponownie, podobnie jak w przypadku poprzedniej części testów, eksperymenty powtarzano dla 15 różnych ziaren generatora liczb pseudolosowych, uśredniając uzyskane wyniki. Uzyskane wyniki prezentują rysunki 3 i 4, a także tabele 1, 2 oraz 3. W przypadku ostatniej funkcji pominięto zaznaczenie na wykresie wartości faktycznego minimum globalnego, gdyż ma ono wartość 0, co zaburzyłoby skalę i czytelność wykresów.

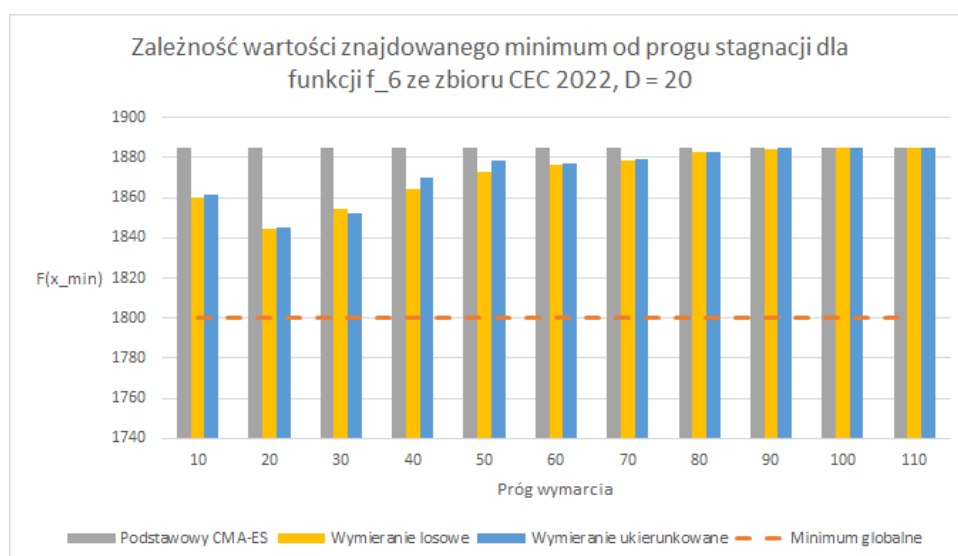
Analizując wyłącznie wyniki uzyskane dla różnych progów wymarcia (uśrednioną dla wszystkich sprawdzanych wartości prawdopodobieństwa wymarcia), warto zwrócić uwagę na nieprzewidywalne zachowanie algorytmów przy niskiej wartości progu. W tych przypadkach algorytm bardzo szybko podejmował decyzję o uruchomieniu mechanizmu wymarcia, co prowadziło do znacznych modyfikacji w dalszym rozwoju populacji. Niestety, ciężko jednoznacznie określić kierunek tych zmian. Dla małego progu wymarcia, dla 10-wymiarowej funkcji ze zbioru CEC [5], wyniki były gorsze niż podstawowego algorytmu CMA-ES, dla 20-wymiarowej - lepsze, zaś dla funkcji Rastrigina zależało to od typu wymierzania. Jednocześnie, wartości progu powyżej 60 praktycznie nie wpływały na zachowanie algorytmu. W tych przypadkach, stagnacja osiągnięta była już w momencie gdy praktycznie cała populacja grupowała się wokół znalezionej minimum (zazwyczaj lokalnego) i usuwanie części osobników nie zmieniało dynamiki całej grupy.

Z drugiej strony, skupiając się wyłącznie na wartościach prawdopodobieństwa, również ciężko wskazać jednoznaczną tendencję poprawy jakości rozwiązań. Dla funkcji ze zbioru CEC [5], przy większej wymiarowości, wymierzanie poprawiło nieznacznie uzyskany rezultat we wszystkich przypadkach. Dla mniejszej wymiarowości z kolei, uzyskane rezultaty były gorsze. Przy funkcji Rastrigina, ponownie, wszystko zależało to również w dużym stopniu od stosowanego typu wymierzania.

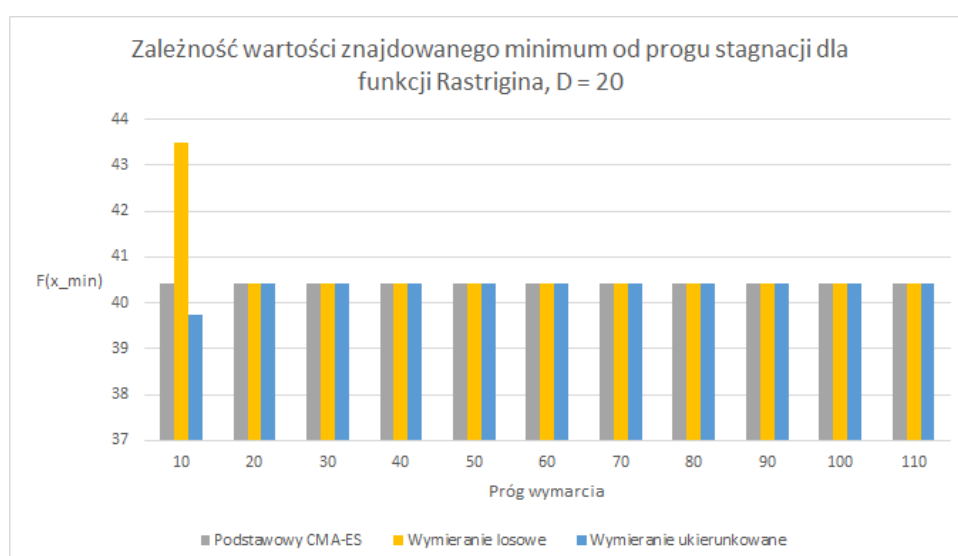
Bardziej reprezentatywne w tym przypadku okazało się zestawienie wyników dla obu parametrów, co przedstawiono na wspomnianych już tabelach 1, 2 oraz 3. Dla czytelności pogrubione zostały w nich najlepsze uzyskane wartości oraz najlepsze średnie. Najlepszą w każdej tabeli dodatkowo podkreślono.



(a) 10-wymiarowa funkcja z CEC

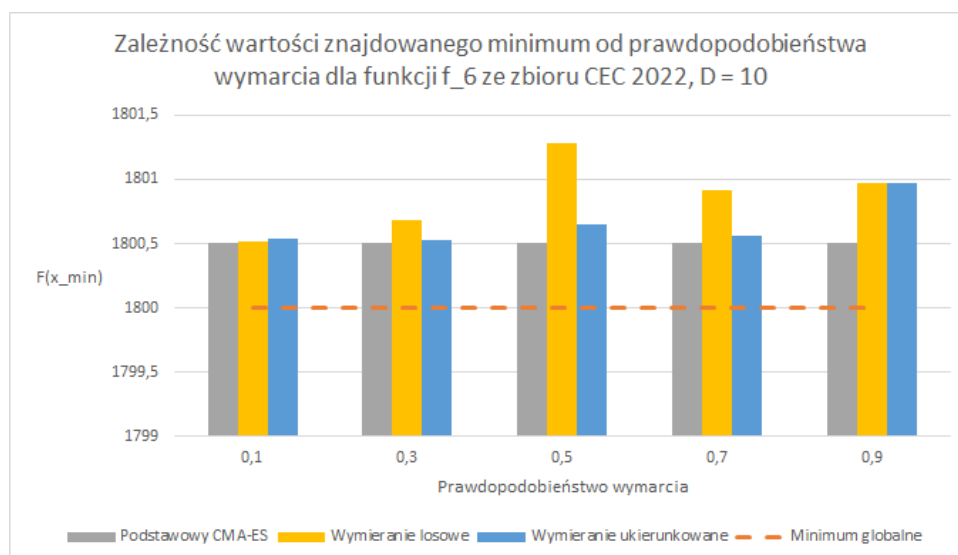


(b) 20-wymiarowa funkcja z CEC

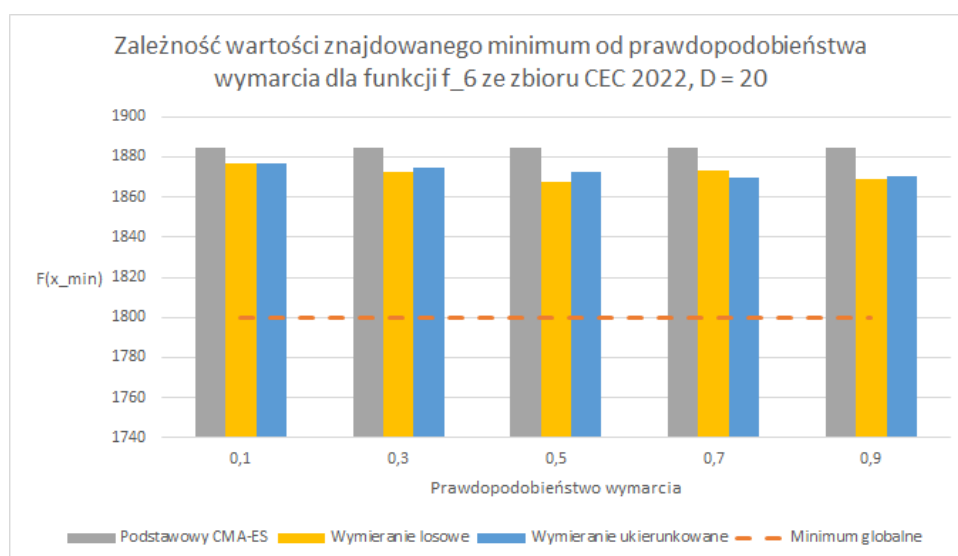


(c) 20-wymiarowa funkcja Rastrigina

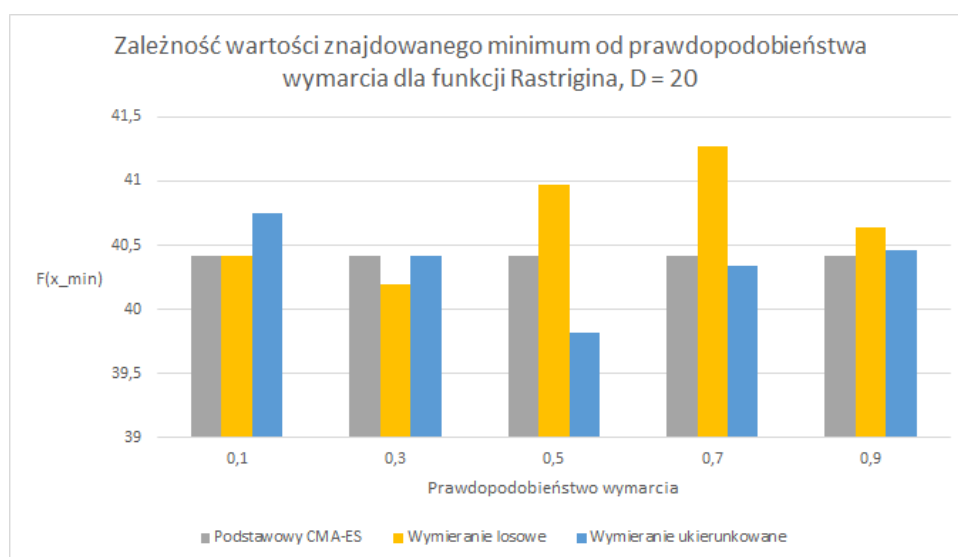
Rysunek 3: Wartości znajdowanego minimum w zależności od algorytmu i progu wymarcia



(a) 10-wymiarowa funkcja z CEC



(b) 20-wymiarowa funkcja z CEC



(c) 20-wymiarowa funkcja Rastrigina

Rysunek 4: Wartości znajdowanego minimum w zależności od algorytmu i prawdopodobieństwa wymarcia

Tablica 1: Wartość znalezione minimum w zależności od progu i prawdopodobieństwa wymarcia dla 10 wymiarowej funkcji ze zbioru CEC 2022

K	p = 0,1	p = 0,3	p = 0,5	p = 0,7	p = 0,9	Średnia
10	1800,592	1800,604	1807,871	1803,032	1803,383	1803,097
20	1800,538	1801,408	1800,725	1801,551	1801,579	1801,160
30	1800,511	1801,397	1801,388	1801,389	1801,554	1801,248
40	1800,507	1800,503	1800,501	1800,500	1800,558	1800,514
50	1800,507	1800,508	1800,510	1800,508	1800,507	1800,508
60	1800,508	1800,508	1800,508	1800,508	1800,508	1800,508
70	1800,508	1800,509	1800,508	1800,509	1800,509	1800,508
80	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
90	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
100	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
110	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
Średnia	1800,519	1800,679	1801,277	1800,912	1800,967	1800,871

(a) Wymieranie losowe

K	p = 0,1	p = 0,3	p = 0,5	p = 0,7	p = 0,9	Średnia
10	1800,616	1800,543	1801,916	1800,942	1805,235	1801,850
20	1800,650	1800,603	1800,548	1800,640	1800,814	1800,651
30	1800,540	1800,576	1800,536	1800,534	1800,547	1800,547
40	1800,508	1800,511	1800,532	1800,534	1800,528	1800,522
50	1800,509	1800,508	1800,509	1800,509	1800,509	1800,509
60	1800,508	1800,509	1800,509	1800,509	1800,508	1800,508
70	1800,509	1800,509	1800,509	1800,509	1800,508	1800,509
80	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
90	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
100	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
110	1800,509	1800,509	1800,509	1800,509	1800,509	1800,509
Średnia	1800,534	1800,527	1800,645	1800,565	1800,971	1800,648

(b) Wymieranie ukierunkowane

Tablica 2: Wartość znalezione minimum w zależności od progu i prawdopodobieństwa wymarcia dla 20 wymiarowej funkcji ze zbioru CEC 2022

K	p = 0,1	p = 0,3	p = 0,5	p = 0,7	p = 0,9	Średnia
10	1859,010	1854,897	1844,083	1878,535	1863,387	1859,982
20	1851,389	1842,943	1833,831	1847,564	1847,352	1844,616
30	1875,891	1857,056	1851,551	1844,331	1843,214	1854,408
40	1881,413	1868,000	1861,623	1860,547	1850,434	1864,403
50	1878,521	1876,269	1869,386	1876,250	1865,103	1873,106
60	1878,281	1880,609	1869,578	1878,433	1875,149	1876,410
70	1878,684	1878,805	1874,743	1881,048	1879,210	1878,498
80	1884,647	1884,629	1882,457	1881,002	1881,073	1882,762
90	1884,762	1884,822	1884,810	1884,731	1882,383	1884,302
100	1884,744	1884,766	1884,749	1884,819	1884,938	1884,803
110	1884,705	1884,712	1884,717	1884,790	1884,731	1884,731
Średnia	1876,550	1872,501	1867,412	1872,913	1868,816	1871,638

(a) Wymieranie losowe

K	p = 0,1	p = 0,3	p = 0,5	p = 0,7	p = 0,9	Średnia
10	1859,205	1843,709	1867,304	1864,366	1873,636	1861,644
20	1857,667	1855,907	1836,825	1827,946	1849,260	1845,521
30	1873,642	1857,945	1856,964	1843,233	1830,299	1852,417
40	1878,097	1881,629	1871,690	1863,200	1857,023	1870,328
50	1878,087	1882,724	1875,589	1879,141	1876,375	1878,383
60	1880,741	1878,555	1869,852	1878,828	1875,892	1876,774
70	1880,892	1884,739	1877,575	1877,466	1875,636	1879,262
80	1884,575	1884,682	1884,849	1880,835	1878,170	1882,622
90	1884,724	1884,730	1884,744	1884,960	1884,910	1884,814
100	1884,779	1884,738	1884,876	1884,714	1884,717	1884,765
110	1884,729	1884,696	1884,720	1884,708	1884,712	1884,713
Średnia	1877,013	1874,914	1872,272	1869,945	1870,057	1872,840

(b) Wymieranie ukierunkowane

Tablica 3: Wartość znalezionej minimum w zależności od progu i prawdopodobieństwa wymarcia dla 20 wymiarowej funkcji Rastrigina

K	p = 0,1	p = 0,3	p = 0,5	p = 0,7	p = 0,9	Średnia
10	40,418	37,984	46,447	49,742	42,866	43,491
20	40,418	40,418	40,418	40,418	40,418	40,418
30	40,418	40,418	40,418	40,418	40,418	40,418
40	40,418	40,418	40,418	40,418	40,418	40,418
50	40,418	40,418	40,418	40,418	40,418	40,418
60	40,418	40,418	40,418	40,418	40,418	40,418
70	40,418	40,418	40,418	40,418	40,418	40,418
80	40,418	40,418	40,418	40,418	40,418	40,418
90	40,418	40,418	40,418	40,418	40,418	40,418
100	40,418	40,418	40,418	40,418	40,418	40,418
110	40,418	40,418	40,418	40,418	40,418	40,418
Średnia	40,418	40,197	40,966	41,266	40,641	40,698

(a) Wymieranie losowe

K	p = 0,1	p = 0,3	p = 0,5	p = 0,7	p = 0,9	Średnia
10	44,013	40,431	33,841	39,558	40,864	39,742
20	40,418	40,418	40,418	40,418	40,418	40,418
30	40,418	40,418	40,418	40,418	40,418	40,418
40	40,418	40,418	40,418	40,418	40,418	40,418
50	40,418	40,418	40,418	40,418	40,418	40,418
60	40,418	40,418	40,418	40,418	40,418	40,418
70	40,418	40,418	40,418	40,418	40,418	40,418
80	40,418	40,418	40,418	40,418	40,418	40,418
90	40,418	40,418	40,418	40,418	40,418	40,418
100	40,418	40,418	40,418	40,418	40,418	40,418
110	40,418	40,418	40,418	40,418	40,418	40,418
Średnia	40,745	40,419	39,820	40,340	40,459	40,357

(b) Wymieranie ukierunkowane

Analizując przedstawione tabele a także mając na uwadze duże wahania skrajnych wartości, uznano za najbardziej reprezentatywne wartości parametrów $p_e = 0,5$ oraz $K = 20$. Prawdopodobieństwo to dawało najczęściej wyniki najlepsze lub od nich nieodległe. Jednocześnie, dla właśnie tej wartości prawdopodobieństwa, najbardziej optymalne rezultaty były osiągnięte przy progu równym 20, a żaden z nich nie odbiegał znacznie od najlepszych.

5 Analiza wyników na pozostałych funkcjach ze zbioru CEC 2022

W sekcji tej skupiono się na tych funkcjach dla których odnotowano obiecujące wyniki dla wersji algorytmu z zastosowanym mechanizmem wymarcia.

5.1 *Shifted and full Rotated Expanded Schaffer's f6 function*

Przesunięta i obrócona, rozszerzona funkcja Schaffera ma wzór:

$$F_3(x) = f_3(M(\frac{0.5(x-o)}{100}) + F_3^*, F_3^* = 600$$

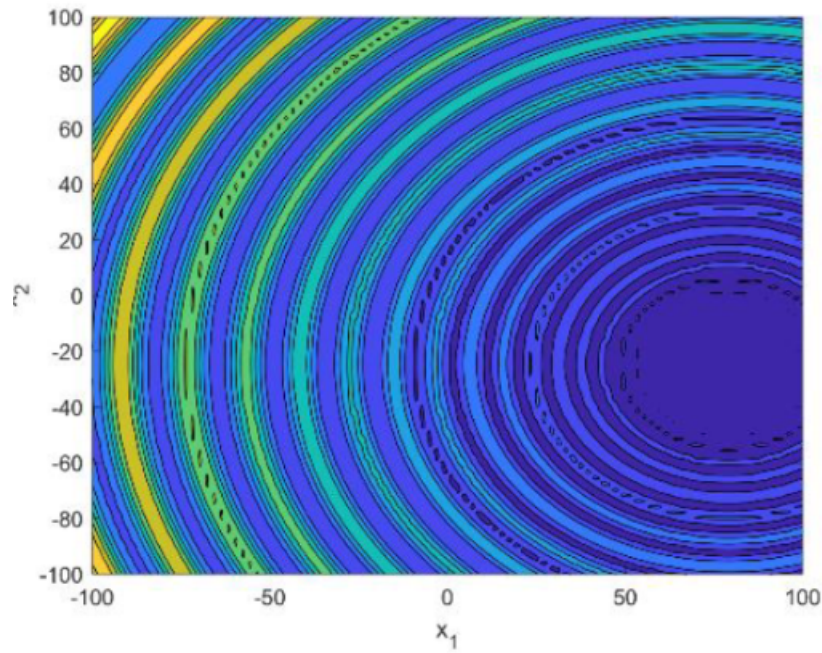
gdzie M to macierz obrotu, o to przesunięcie globalnego minimum, a $f_3(x)$ to rozszerzona funkcja Schaffera, która ma wzór:

$$f_3(x) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$$

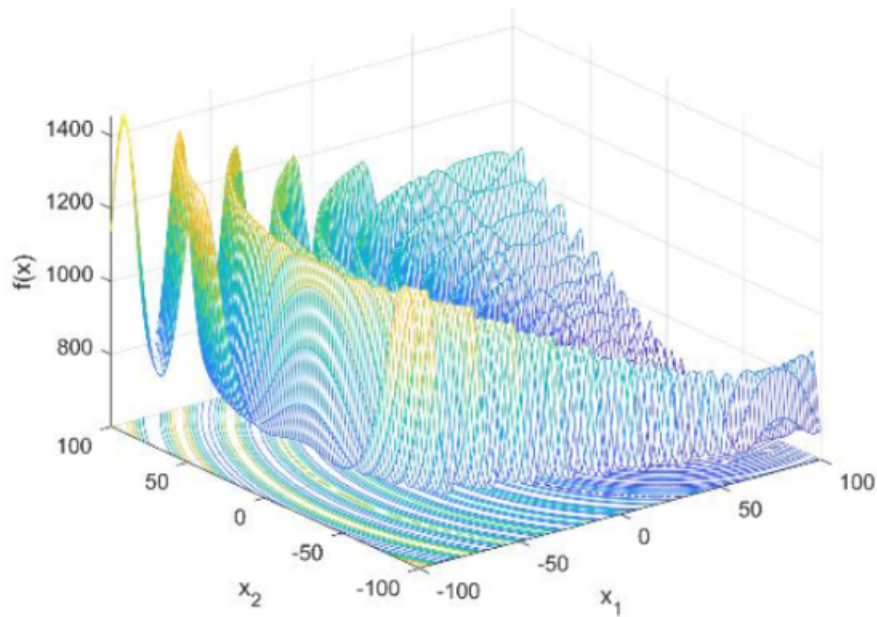
D to wymiar zadania, a $g(x, y)$ to funkcja Schaffera o wzorze:

$$g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$$

Ze względu na wartość F_3^* wartość w minimum globalnym to 600. Wykres funkcji konturowej dla dwóch wymiarów przypomina tafle wody po której roznoszą się okrągłe fale. Funkcja ta posiada wiele minimów lokalnych, oraz jest niesymetryczna.



Rysunek 5: Konturowy wykres przesuniętej i obróconej, rozszerzonej funkcji Schaffera.



Rysunek 6: Wykres 3D przesuniętej i obróconej, rozszerzonej funkcji Schaffera dla 2 wymiarów.

Przeprowadzono eksperyment polegający na znalezieniu minimum tej funkcji dla 20 wymiarów, za pomocą podstawowej wersji algorytmu CMA-ES jak i wersji z wymieraniem ukierunkowanym i losowym. Przyjęto wyznaczoną w poprzednich testach wielkość populacji 310, próg stagnacji wymagany do wymarcia 20, prawdopodobieństwo wymarcia 0.5, a odsetek populacji nie podlegający wymarciu w wersji ukierunkowanej przyjęto 0.2. Wyniki osiągnięte przez trzy wersje algorytmu, uśrednione dla 15 uruchomień, zostały przedstawione w tabeli 4. Widać po nich, że wersja algorytmu CMA-ES z wymieraniem ukierunkowanym osiągnęła średnio lepszy wynik niż podstawowa wersja, gdyż znalazła ona punkt minimum o wartości 666.2754, podczas gdy podstawowa wersja znalazła średnio punkt o wartości 670.3158. Ponadto odchylenie standardowe dla znalezionej minimum było najmniejsze właśnie dla wymierania ukierunkowanego. Jednakże dla wszystkich wersji, średnia znaleziona wartość była dość odległa od faktycznego minimum.

Wersja algorytmu	Znalezione minimum	Odchylenie standardowe
Podstawowy CMA-ES	670.3158	13.4620
Z wymieraniem ukierunkowanym	666.2754	8.8949
Z wymieraniem losowym	767.4047	15.9639
Minimum globalne	600	-

Tablica 4: Wyniki dla funkcji f_3 dla populacji o rozmiarze 310 osobników.

5.2 *Shifted and full Rotated Non-Continous Rastrigin's Function*

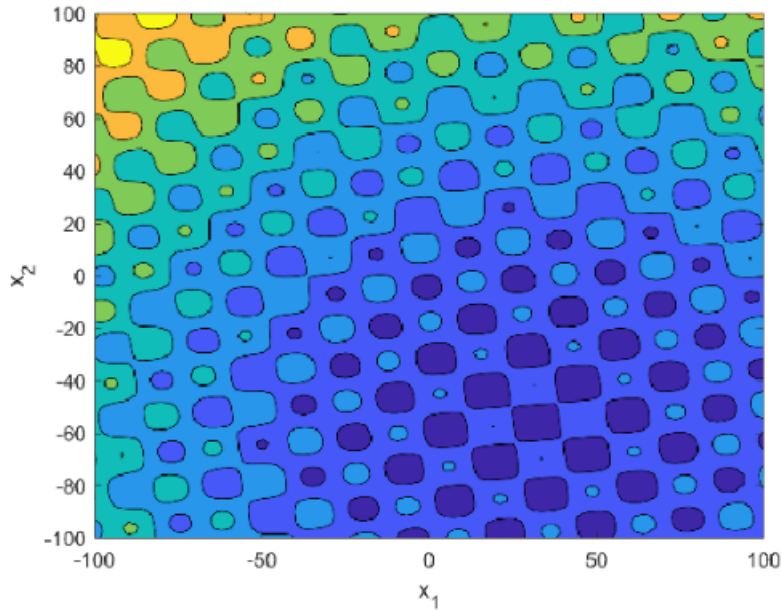
Przesunięta i obrócona, nieciągła funkcja Rastrigina ze zbioru CEC 2022 ma wzór:

$$F_4(x) = f_4(M(\frac{5.12(x - o)}{100}) + F_4^*, F_4^* = 800$$

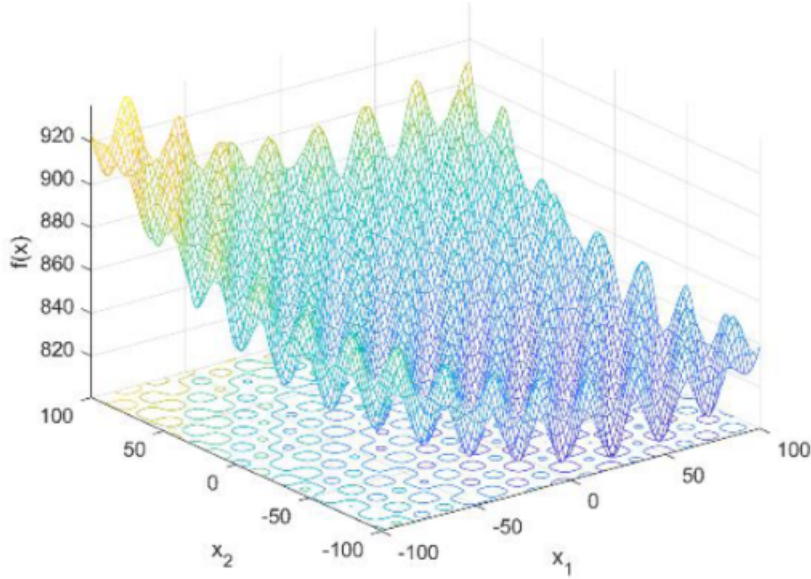
gdzie M to macierz obrotu, o to przesunięcie globalnego minimum, a $f_4(x)$ to funkcja Rastrigina, która ma wzór:

$$f_4(x) = \sum_{i=1}^D (x_u^2 - 10 \cos(2\pi x_i) + 10)$$

Ze względu na wartość F_4^* wartość w minimum globalnym to 800. Wykres funkcji konturowej dla dwóch wymiarów przypomina „wygiętą” wytłoczkę do jajek. Funkcja ta posiada wiele minimów lokalnych, oraz jest niesymetryczna.



Rysunek 7: Konturowy wykres przesuniętej i obróconej, nieciągłej funkcji Rastrigina.



Rysunek 8: Wykres 3D przesuniętej i obróconej, nieciągłej funkcji Rastrigina dla 2 wymiarów.

Wykonano taki sam eksperyment jak w przypadku funkcji F_3 . Uśrednione wyniki przedstawiono w tabeli 5. Pokazują one, że podobnie jak dla funkcji Schaffera, najlepsze wyniki otrzymano dla wersji z ukierunkowanym wymieraniem. Także odchylenie standardowe było najmniejsze dla tej wersji. Co ciekawe, także losowa wersja osiągnęła lepszy średni wynik niż podstawowa wersja CMA-ES.

Wersja algorytmu	Znalezione minimum	Odchylenie standardowe
Podstawowy CMA-ES	893.3401	6.5186
Z wymieraniem ukierunkowanym	887.6355	5.4866
Z wymieraniem losowym	890.1483	6.1983
Minimum globalne	800	-

Tablica 5: Wyniki dla funkcji f_4 dla populacji o rozmiarze 310 osobników.

5.3 *Shifted and Rotated Levy Function*

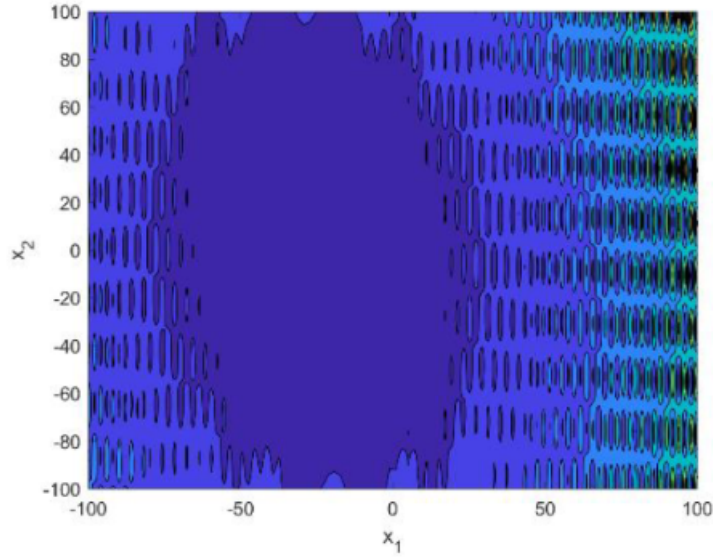
Przesunięta i obrócona Funkcja Levy’ego ze zbioru CEC [5] ma wzór:

$$F_5(x) = f_5(M(\frac{5.12(x-o)}{100}) + F_{5*}, F_{5*} = 900$$

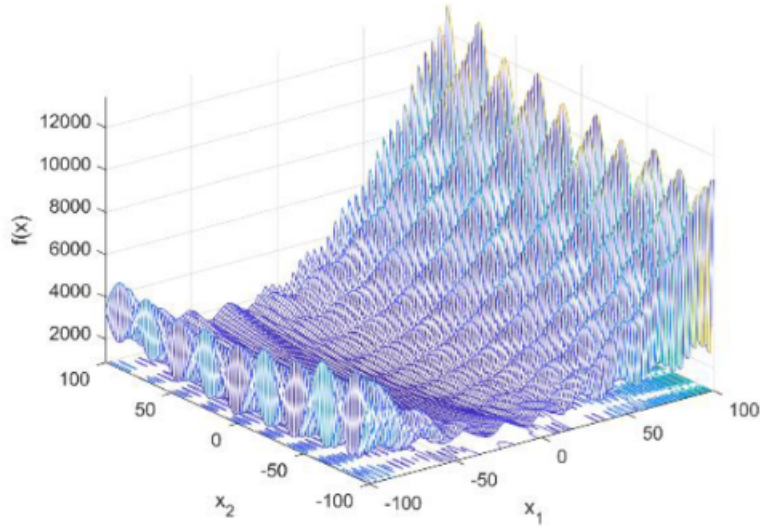
gdzie M to macierz obrotu, o to przesunięcie globalnego minimum, a $f_5(x)$ to funkcja Levy’ego, która ma wzór:

$$f_5(x) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i - 1)] + (w_D - 1)^2 [1 + \sin^2(2\pi w_D)],$$

gdzie $w_i = 1 + \frac{x_i - 1}{4}, \forall i = 1, \dots, D$. Ze względu na wartość F_{5*} jej minimum ma wartość 900. Funkcja ta podobnie jak F_4 i F_3 posiada wiele minimum lokalnych.



Rysunek 9: Konturowy wykres przesuniętej i obróconej funkcji Levy'ego.



Rysunek 10: Wykres 3D przesuniętej i obróconej funkcji Levy'ego dla 2 wymiarów.

Dla funkcje Levy'ego wykonano ten sam eksperyment jak dla funkcji F_3 i F_4 . Wyniki przedstawiono w tabeli 6. Widać, że dla tej funkcji algorytm CMA-ES nie jest w stanie znaleźć dobrej wartości. Średnia znaleziona wartość dla podstawowej metody wynosi 3306.6331, czyli znacznie różni się od faktycznego minimum globalnego, które ma wartość 900. Warto tu zauważyć, że wartości dla wszystkich wersji algorytmu są zbliżone. Pomimo iż dla wymienionych wyżej funkcji F_3 i F_4 , wersja z ukierunkowanym wymieraniem była w stanie znaleźć lepsze rozwiązania, to dla tej funkcji, średnia wartość rozwiązania wyznaczonego za pomocą tej modyfikacji, była gorsza. Ponadto pozwala to zauważyć, że często wartości znajdowane za pomocą wymierania są zbliżone do wartości znalezionych przez podstawowy algorytm CMA-ES i jedynie nieznacznie od nich odbiegają (czasami dając poprawę jak w przypadku F_3 i F_4 , a czasami pogorszenie jak dla funkcji Levy'ego).

Wersja algorytmu	Znalezione minimum	Odchylenie standardowe
Podstawowy CMA-ES	3306.6331	80.8529
Z wymieraniem ukierunkowanym	3308.1225	81.2973
Z wymieraniem losowym	3309.2155	81.2182
Minimum globalne	900	-

Tablica 6: Wyniki dla funkcji f_4 dla populacji o rozmiarze 310 osobników.

6 Podsumowanie

Niestety, w wyniku przeprowadzonych eksperymentów nie udało się jednoznacznie ustalić, czy modyfikacja algorytmu CMA-ES poprzez wprowadzenie opartego na stagnacji mechanizmu wymarcia wpływa pozytywnie na skuteczność metody. W niektórych przypadkach, dodanie wymarcia pozwala co prawda uzyskiwać wyniki bliższe minimum globalnemu, jednak często przy wymagających funkcjach algorytm potrafią zostać „zwiedzione” przez minima lokalne. Podkreśla to moc podstawowej wersji CMA-ES, gdyż pozornie istotna poprawka okazuje się równie często pogarszać działanie wersji podstawowej.

Jednocześnie, osiągniętych rezultatów nie można uznać za nieciekawe. W toku przeprowadzonych eksperymentów, zauważono tendencję wszystkich analizowanych algorytmów do bardzo krótkich okresów stagnacji przed osiągnięciem zbieżności do pewnego minimum lokalnego. Niewątpliwie zmniejszyło to potencjalną skuteczność wymarcia opartego właśnie na stagnacji. Jednocześnie, być może wprowadzenie mechanizmu wymarcia niezależnego od stagnacji, który działałby również na wczesnym etapie przebiegu algorytmu, mogłoby dać ciekawe rezultaty.

Kolejnym dużym problemem była maksymalizacja efektów działania poprzez dobór uniwersalnego progu wymarcia i tolerancji określającej stagnację. W zależności od funkcji, różne wartości parametrów, dla różnych wymiarowości, czy nawet wywołań, były w stanie dawać wyraźnie różne wyniki. Być może, przyjmując konkretną specyfikę przeszukiwanej przestrzeni bądź funkcji celu, dobór tych parametrów byłby łatwiejszy i bardziej efektywny.

Innym interesującym wnioskiem jest duża zależność jakości działania metody od wielkości wybranej populacji. Wartość tego parametru, mimo iż uznana przez autorów algorytmu za mało istotną, okazała się mieć największy wpływ ze wszystkich sprawdzanych parametrów.

Poza wielkością populacji, olbrzymi wpływ na wyniki miała również wartość ziarna generatora liczb pseudolosowych. Analizując wywołania dla różnych wartości, wyniki często diametralnie się od siebie różniły. Nasuwa to kolejny bardzo ciekawy potencjalny kierunek dalszych badań. Dla algorytmów o czasie działania ograniczonym z góry przez liczbę wywołań funkcji celu, ciekawym ulepszeniem może okazać się redukcja liczby populacji, a następnie dodawanie w ich miejsce innych, zwiększając w ten sposób zdolność do eksploracji, przy stosunkowo stałej wielkości populacji.

Bibliografia

- [1] Nikolaus Hansen i Andreas Ostermeier. „Completely Derandomized Self-Adaptation in Evolution Strategies”. W: *Evolutionary Computation* 9.2 (2001), s. 159–195. DOI: 10.1162/106365601750190398.
- [2] Gan Zhen Ye i Dae-Ki Kang. „Extended Evolutionary Algorithms with Stagnation-Based Extinction Protocol”. W: *Applied Sciences* 11.8 (2021). ISSN: 2076-3417. DOI: 10.3390/app11083461. URL: <https://www.mdpi.com/2076-3417/11/8/3461>.
- [3] J. Arabas M. Bobowski. *Roznicowa implementacja algorytmu CMAES*. 2015. URL: <https://repo.pw.edu.pl/info/master/WUT0cb7aa5615d34f83a75b55f5d38f595a/>.
- [4] Nikolaus Hansen. *CMA-ES Source Code*. 2011. URL: https://cma-es.github.io/cmaes_sourcecode_page.html#matlab (term. wiz. 19.06.2022).
- [5] Abhishek Kumar Kenneth V. Price Ali Wagdy Mohamed Anas A. Hadi P. N. Suganthan. *Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*. 2021. URL: <https://github.com/ewarchul/cec/blob/main/doc/cec2022-specification.pdf> (term. wiz. 19.06.2022).