

Wpływ wykorzystania archiwum do mutacji różnicowej  
w algorytmie ewolucji różnicowej na jakość  
optymalizacji w przestrzeni ciągłej.

Andrzej Fiedukowicz

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>5</b>
1.1	Definicja problemu . . . . .	5
1.2	Motywacje . . . . .	7
1.3	Cel . . . . .	8
<b>2</b>	<b>Przegląd literatury</b>	<b>10</b>
2.1	Podejście heurystyczne do problemu optymalizacji . . . . .	10
2.2	Algorytmy ewolucyjne . . . . .	11
2.2.1	Ewolucja różnicowa . . . . .	13
2.3	Metodyka badań (opisywać tak szeroko?) . . . . .	14
2.3.1	Modelowanie statystyczne . . . . .	14
2.3.2	Testy statystyczne . . . . .	14
2.3.3	Testy parametryczne . . . . .	14
2.3.4	Testy nieparametryczne . . . . .	14
2.4	Benchmarki blackboxowe . . . . .	15
2.4.1	CEC . . . . .	15
<b>3</b>	<b>Algorytm DEArch</b>	<b>16</b>
3.1	Podejścia do wykorzystania archiwum . . . . .	16

---

3.1.1	Brak normalizacji . . . . .	16
3.1.2	Normalizacja . . . . .	16
3.1.3	Inne podejścia . . . . .	16
3.2	Definicja algorytmu . . . . .	16
<b>4</b>	<b>Badania</b>	<b>17</b>
4.1	Cel . . . . .	17
4.2	Plan eksperymentów . . . . .	17
4.3	Dobór parametrów . . . . .	17
4.4	Zbierane rezultaty . . . . .	17
4.4.1	Środek ciężkości populacji (DE/mid) . . . . .	17
4.5	Sposób analizy . . . . .	17
4.6	Wyniki (w załączniku?) . . . . .	17
4.7	Analiza wyników (wnioski) . . . . .	17
<b>5</b>	<b>Podsumowanie</b>	<b>18</b>

## Streszczenie

*Algorytmy ewolucji różnicowej (DE) występują w wielu wariantach i odmianach, których właściwości są cały czas badane. Jedną z proponowanych modyfikacji tego algorytmu stanowi wykorzystanie przy operatorze mutacji różnicowej osobników pochodzących z szerszego niż jednopokoleniowego okna historii [MS:PrzydałoBySięŹródło, jeśli nie – zmiana sformułowania].*

*Niniejsza praca opisuje proces opracowywania metody zastosowania archiwum osobników w celu poprawy jakości wyników uzyskiwanych w ewolucji różnicowej. W dalszej części praca skupia się na sprawdzeniu hipotezy o możliwości poprawy klasycznego schematu ewolucji różnicowej DE/rand/1/bin przez wykorzystanie archiwum. Badania skuteczności poszczególnych metod opracowane są w oparciu o zestaw funkcji testowych opracowanych w ramach konferencji IEEE Congress on Evolutionary Computation 2013 (CEC2013) [MS:Potrzebne Źródło].*

*Niniejsza praca odnosi się także do możliwości i skuteczności wykorzystania w procesie ewolucji różnicowej punktu środkowego populacji, badając jego wartość w czasie działania wszystkich testowanych wariantów algorytmów.*

## Słowa kluczowe

*Ewolucja Różnicowa, DE, Algorytmy Ewolucyjne, AE, Optymalizacja, Optymalizacja black0-box*

## Abstract

*There are many forms of differential evolution algorithms (DE), which properties are yet to be discovered. One of possible modifications of those algorithms is using historical points to perform differential mutation. [MS:PrzydałoBySięŹródło].*

*This thesis describes development process of method of using such archive to increase DE results quality. Hereafter this thesis is checking hipotesis about possibility of improving classic*

*DE/rand/1/bin by using archive. All experiments performed during this reasearch are based on benchmark functions set developed during the IEEE Congress of Evolutionary Computation 2013 conference (CEC2013) [MS:Potrzebne Źródło].*

*This thesis also refers to a possiblity and effectiveness of using middle point of population to improve quality of DE results by checking its value in all tested variants of DE.*

## **Keywords**

*Differential Evolution, DE, Evolutionary Algorithms, EA, Optimialization, Black-box optimization*

# Rozdział 1

## Wstęp

Problem optymalizacji jest jednym z najczęściej pojawiających się w praktycznych zastosowaniach problemów numerycznych. Sama powszechność podstawowego problemu, ale także możliwość sprowadzenia do problemu optymalizacji problemów klasyfikacji, regresji, grupowania czy też określając szerzej – uczenia maszynowego [7], sprawia, że przed algorytmami optymalizacji stawiane są coraz bardziej złożone i wymagające zadania.

Stale zwiększający się poziom trudności problemów optymalizacyjnych jak i ciągła niedoskonałość dotychczas wytworzonych metod ich rozwiązywania powoduje duże zainteresowanie badaniami dotyczącymi tworzenia coraz doskonalszych, szybszych i dających lepsze rezultaty algorytmów owe problemy rozwiązujących [6].

### 1.1 Definicja problemu

Praca ta, umiejscowiona została w kontekście rozwiązywania problemów optymalizacji gdzie analityczne wyznaczenie ekstremum funkcji jest niemożliwe bądź w ogóle nie jest znana jej analityczna forma. Zakłada się bowiem, że zadanie jest zdefiniowane następująco [5]:

**Definition 1.** *Problem globalnej optymalizacji funkcji  $f : X \rightarrow \mathbb{R}$  polega na wyznaczeniu takiego*

$x_0 \in X$ , że:

$$\forall_{x \in X} f(x_0) \leq f(x)$$

Dla celów tej pracy przyjmuje się możliwość zbadania wartości funkcji  $f$  w dowolnym punkcie ze zbioru  $X$ , jednak brak możliwości bezpośredniego zbadania jakichkolwiek innych własności tejże funkcji. W tej formie problem ten można nazwać problemem optymalizacji funkcji zadanej w formie reaktywnej (*Black-box optimization*).

W ramach rozpatrywanych problemów istnieje także możliwość, że dziedzina funkcji  $f$  jest szersza niż zbiór rozwiązań dopuszczalnych. W takich wypadkach mówimy o problemie optymalizacji z ograniczeniami.

**Definition 2.** *Problem globalnej optymalizacji funkcji  $f : X \rightarrow \mathbb{R}$  z ograniczeni  $X_D \subset X$  polega na wyznaczeniu takiego  $x_0 \in X_D$ , że:*

$$\forall_{x \in X_D} f(x_0) \leq f(x)$$

Dla tego rodzaju problemów, możliwe jest wyznaczanie wartości funkcji w całej jej dziedzinie  $X$ , jednak poszukiwane rozwiązanie musi pochodzić z ograniczonego zbioru  $X_D$ . Problem globalnej optymalizacji funkcji z ograniczeniami stanowi uogólnienie problemu optymalizacji globalnej bez ograniczeń.

Szczególnym przypadkiem problemu optymalizacji globalnej z ograniczeniami jest przeszukiwanie przestrzeni ciągłej.

**Definition 3.** *Problem globalnej optymalizacji funkcji  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  z ograniczeniem  $X_D = \{x \in \mathbb{R}^n : g_1(x) \leq 0, g_2(x) \leq 0, \dots, g_m(x) \leq 0\}$  nazywamy problemem optymalizacji globalnej przestrzeni ciągłej z ograniczeniami nierównościami.*

W tym wariancie przyjmuje się, że  $X = \mathbb{R}^n$  gdzie  $n \in \mathbb{N}$ , a  $X_D$  zdefiniowane jest przez funkcje ograniczeń.

Przy tych założeniach, należy zauważyć, że dla każdego z powyższych problemów jeśli zbiór  $X$  jest nieskończenie liczny (nawet przeliczalny), nie jest możliwe zweryfikowanie, czy wyznaczony punkt jest rozwiązaniem zadanego problemu. Co więcej, w wielu przypadkach (brak istnienia hiperpłaszczyzny  $X$ , takiej, że w każdym jej punkcie wartość funkcji jest równa poszukiwanemu ekstremum) prawdopodobieństwo odnalezienia rozwiązania problemu (niezależnie od metody przeszukiwania przestrzeni) wynosi zero.

Mając to na względzie by problem można było uznać za rozwiązany, należy go przeformułować tak, że rozwiązaniem jest dowolny element  $x \in X_D$ , a jego jakość wyznaczana jest przez wartość funkcji  $f$  w tym punkcie. Dla tak zadanego problemu możliwe jest porównywanie różnych metod przeglądania przestrzeni  $X_D$  pod względem jakości otrzymanego rozwiązania.

Niniejsza praca będzie skupiała się na rozwiązywaniu tak przeformułowanego problemu optymalizacji globalnej funkcji w przestrzeni ciągłej z ograniczeniami nierównościowymi. Ze względu na zwięzłość zapisu w dalszej części pracy problem ten nazywany będzie **problemem optymalizacji** lub po prostu **problemem**.

Warto zwrócić uwagę, że pomimo iż między różnymi rozwiązaniami problemu istnieje relacja częściowego porządku generowana przez wartość funkcji  $f$ , to jednak przy przyjętych założeniach wartość (rozumiana jako liczba rzeczywista) tej funkcji nie niesie żadnej dodatkowej informacji. Z tego też powodu, wszelkie badania porównawcze przeprowadzane na poszczególnych metodologiach rozwiązywania problemu muszą być oparte na tej właśnie relacji porządku.

## 1.2 Motywacje

Jednym z podejść do zdefiniowanego powyżej problemu optymalizacji jest wykorzystanie algorytmów ewolucji różnicowej (DE) [4]. Algorytm ten zaproponowany w 1995r. od wielu lat rozwijany w ramach licznych prac naukowych [3] cieszy się zwykle bardzo dobrymi rezultatami



w testach jakościowych [2].

Typową procedurą przy heurystycznych algorytmach optymalizacyjnych jest wykorzystywanie do generowania kolejnych punktów w przestrzeni przeszukiwań punktów z okna historii o zmiennej szerokości [MS:MożeJakiśŹródło?]. W przypadku klasycznego algorytmu ewolucji różnicowej, okno historii ma rozmiar równy liczności populacji – starsze punkty nie są wykorzystywane w kolejnych pokoleniach. Choć w przeszłości podejmowano próby wykorzystywania szerszego okna historii w tym algorytmie [MS:Gdzie?], temat ten zwykle występował przy okazji innych badań i nie był osobno zbadany /Note:AleCzyNaPewno?/. Praca ta będzie starała się dokładnie zbadać możliwości wykorzystania tego rodzaju strategii i określić w jaki sposób wykorzystanie tej strategii wpłynie na jakość uzyskiwanych przy pomocy ewolucji różnicowej wyników.

### 1.3 Cel

Celem pracy jest zbadanie zmodyfikowanego klasycznego algorytmu ewolucji różnicowej [4] tak by wektory mutacji różnicowej były budowane w oparciu o osobniki z parametryzowanej liczby ostatnich pokoleń.

Przebadane zostaną możliwe podejścia do wykorzystania tych osobników i wynikające z nich zmiany jakości działania algorytmu. W wyniku tych działań zdefiniowany zostanie możliwie najskuteczniejszy algorytm parametryzowany ilością pokoleń uwzględnianych w oknie historii (zwany dalej *DEArch*).

Cel pracy stanowi także zbadanie gotowego algorytmu *DEArch* przy różnych wartościach nowego parametru i ustalenie jak zmiana parametru wpływa na jakość wyników. Otrzymane rezultaty zostaną następnie zestawione z klasycznym algorytmem ewolucji różnicowej a także przedstawione w kontekście zmodyfikowanych wariantów ewolucji różnicowej przebadanych w

ramach konferencji CEC2013 /Note: Długa nazwa?/.

W ramach pracy zostanie także zbadana teza o możliwości poprawy jakości otrzymywanych przez ewolucję różnicą wyników przy ograniczonym budżecie zapytań przy wykorzystaniu punktu środkowego populacji.

Wszelkie badania jakościowe w ramach pracy będą opracowane przy użyciu testowych funkcji wyspecyfikowanych w ramach konferencji CEC2013 i porównania rezultatów otrzymanych w określonym w ramach konferencji budżecie zapytań o wartości funkcji [MS: GdzieJestWyspecyfikowanyBudżet?].

## Rozdział 2

# Przegląd literatury

### 2.1 Podejście heurystyczne do problemu optymalizacji

Do rozwiązywania problemów optymalizacji ze względu na ich złożoność i (w wielu przypadkach) brak możliwości odnalezienia precyzyjnego rozwiązania stosuje się algorytmy o charakterze heurystycznym. Algorytmy te mają za zadanie rozwiązywać problemy w sposób przybliżony tj. z założenia nie muszą podać rozwiązania dokładnego. Są one stosowane zamiast algorytmów dokładnych tam, gdzie: nie jest możliwe stworzenie precyzyjnego algorytmu, nie jest znany precyzyjny algorytm, precyzyjny algorytm jest zbyt kosztowny w wykonaniu. Algorytmy tej klasy bywają stosowane jako rozwiązanie kompromisowe pomiędzy jakością rozwiązania a czasem jego poszukiwania.

Podejście tego rodzaju niesą za sobą ciekawe konsekwencje. Warto zauważyć, że ponieważ z definicji prowadzą one do rozwiązania nie koniecznie najlepszego, to jeśli istnieje metoda porównywania jakości wyników, można także mówić o porównywaniu jakości działania różnych algorytmów w kontekście danego zadania. Mimo, iż wiele z algorytmów heurystycznych ma charakter niedeterministyczny (tj. dla tego samego wejścia mogą wygenerować dwa różne wyjścia) posługując się narzędziami statystyki i probabilistyki można wysnuwać wnioski dt. jakości algo-

rymtów w kontekście problemu. /Note: A w sekcji o benchmarkach - funkcje są skonstruowane tak, by dawać możliwy obraz jakości algorytmu jako takiego pozwalając go oceniać w oderwaniu od problemu!/

Wartym wspomnienia w kontekście algorytmów heurystycznych jest sposób ich konstruowania. Często dyktowany jest on intuicją, a podstawy teoretyczne działania tego rodzaju algorytmów bywają badane dużo później [3]. Tego rodzaju podejście pozwoliło na stworzenie wielu dobrze sprawujących się algorytmów, lecz zarazem ich analityczne badanie pozwala na dokładniejsze zrozumienie problematyki, uogólnienie lub uproszczenie reguł ich działania [3]. /Note: I co z tą myślą? W pracy postaramy się chociaż trochę zwrócić uwagę na teorię?/

## 2.2 Algorytmy ewolucyjne

Przykładem algorytmów heurystycznych stosowanych do rozwiązywania problemu optymalizacyjnego są algorytmy należące do grupy algorytmów ewolucyjnych (ang. *Evolutionary Algorithms* – AE) [1, 7]. Algorytmy te, wzorowane na ewolucji naturalnej operują pojęciami takimi jak osobnik, środowisko czy przystosowanie w celu adaptacyjnego przeszukiwania przestrzeni optymalizacyjnej.

Ogólna idea stojąca za algorytmami ewolucyjnymi, polega na opisanu genotypu osobników, z których następnie można odczytać fenotyp (stanowiący punkt przestrzeni przeszukiwań) oceniany przez środowisko (funkcję optymalizowaną). Osobniki – podobnie jak osobniki w naturze – podlegają krzyżowaniu oraz niewielkim mutacjom, walcząc między sobą o zasoby naturalne. Zgodnie z teorią doboru naturalnego, osobniki gorzej dostosowane do środowiska są – w szerokiej perspektywie czasowej i przy odpowiednio dużej populacji – wypierane przez te dostosowane lepiej [MS: MożeJakieśŹródłoNaDobórNaturalny? Darwin?].

/Note:Obrazek ze schematem EA?/

W związku z powyższą analogią w ramach algorytmów ewolucyjnych wyróżnia się następujące operatory:

**Krzyżowanie** – to operator opisujący w jaki sposób na podstawie genotypu dwóch lub więcej osobników wytworzyć genotyp jednego lub więcej osobnika potomnego.

**Mutacja** – to operator wprowadzający genotypie wybranego osobnika losowe (nieznaczące) zaburzenie.

**Selekcja** – to operator odpowiadający selekcji naturalnej, który oceniając osobniki na podstawie przystosowania ich fenotypu do środowiska odrzuca część z nich preferując pozostawienie w populacji osobników lepiej dostosowanych.

**Dobór do krzyżowania** – to operator wybierający zestawy osobników podlegające krzyżowaniu. Operator ten opisuje znany z natury mechanizm preferencji (fenotypowej) organizmów przy rozmnażaniu.

**Dobór do mutacji** – to operator wybierający, które spośród osobników w populacji należy poddać mutacji. Odpowiada on zmiennemu w zależności od fenotypu organizmu prawdopodobieństwu wystąpienia u niego mutacji.

**Inicjalizacja** – to operator nieco oderwany od teorii ewolucji (gdyż ta nie opisuje powstawania pierwszych organizmów) opisujący sposób wygenerowania populacji początkowej spośród wszystkich możliwych genotypów.

Ogólny schemat działania algorytmu ewolucyjnego opiera się o iteracyjną adaptację populacji aż do ustalonego warunku stopu, według następujących kroków:

```
P = Inicjalizacja(N)
```

```
while (!Stop) {
```

```
    P += Krzyzowanie(DoborDoKryzowania(P))
```

```
P += Mutacja ( DoborDoMutacji (P) )  
  
P = Selekcja (P)  
  
}  
  
// Note: Uzyc symboli matematycznych zbiorow zamiast +=
```

W tym miejscu warto nadmienić, że choć w powyższym opisie algorytmów ewolucyjnych jest wprost mowa o podziale na genotyp i fenotyp, to na potrzeby zdefiniowanego problemu pojęcie fenotypu będzie wystarczające. Algorytmy opierające się o operacje na zakodowanym genotypie oderwanym od jego semantyki stanowią odrębną gałąź algorytmów ewolucyjnych i są nazywane algorytmami genetycznymi. Algorytmy te, ze względu na specyfikę swojego działania wymagają odpowiedniej reprezentacji danych i odpowiednio dobranych do reprezentacji operatorów. W przypadku rozważanego w pracy problemu optymalizacji, operowanie bezpośrednio na fenotypie ocenianym przez środowisko (a więc wektorach liczb rzeczywistych dla których wyznaczane są wartości funkcji optymalizowanej) ze względu na właściwości i możliwości samych liczb rzeczywistych wydaje się być w zupełności wystarczające. Zagadnienie podziału na genotyp i fenotyp będzie w dalszej części pracy zaniedbane.

(Odmiany i schematy?)

(Parametry)

(Problemy, także te przy badaniu właściwości)

(Eksploracja vs. Ekspłatacja)

### 2.2.1 Ewolucja różnicowa

(Ogólny opis)

(Czym się różni od klasycznych algorytmów ewolucyjnych)

(Zalety i wady w kontekście problemu optymalizacji)

(Parametry)

(Wyniki w kontekście innych metod)

(Idea punktu środkowego)

(Metody uwzględniania ograniczeń)

### **Podejście do archiwum**

(Odniesienia w literaturze)

(Dotychczasowe wnioski)

## **2.3 Metodyka badań (opisywać tak szeroko?)**

### **2.3.1 Modelowanie statystyczne**

(Czemu służy, dlaczego potrzebne do tych badań)

### **2.3.2 Testy statystyczne**

(Ogólna koncepcja, p-value)

### **2.3.3 Testy parametryczne**

(Co to jest i kiedy się stosuje)

#### **Test T-studenta**

(Jak działa, ZAŁOŻENIA)

(Wady zalety).

### **2.3.4 Testy nieparametryczne**

(Co to jest i kiedy się stosuje)

**Test wilcoxona**

(Opis, działanie, ZAŁOŻENIA)

(Wady zalety)

**2.4 Benchmarki blackboxowe**

(Czemu służą)

(Dlaczego są dobrym odniesieniem)

**2.4.1 CEC**

(Opis benchmarka, funkcje testowe)

(Implementacja w R)



## Rozdział 3

# Algorytm DEArch

### 3.1 Podejścia do wykorzystania archiwum

#### 3.1.1 Brak normalizacji

#### 3.1.2 Normalizacja

#### 3.1.3 Inne podejścia

### 3.2 Definicja algorytmu

## Rozdział 4

# Badania

### 4.1 Cel

### 4.2 Plan eksperymentów

### 4.3 Dobór parametrów

### 4.4 Zbierane rezultaty

#### 4.4.1 Środek ciężkości populacji (DE/mid)

### 4.5 Sposób analizy

### 4.6 Wyniki (w załączniku?)

### 4.7 Analiza wyników (wnioski)

## Rozdział 5

## Podsumowanie

# Bibliografia

- [1] Jarosław Arabas. *Wykłady z algorytmów ewolucyjnych*. Wydawnictwo WNT, Warszawa, 2004.
- [2] Tianjun Liao Ilya Loshchilov, Thomas Stuetzle. Ranking results of cec'13 special session & competition on real-parameter single objective optimization. 2013.
- [3] Karol Opara. *Analiza algorytmu ewolucji różnicowej i jego zastosowanie w wyznaczaniu zależności statystycznych*. Praca doktorska, Instytut Badań Systemowych Polskiej Akademii Nauk, 2014.
- [4] Kenneth Price Rainer Storn. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, 1995.
- [5] Andrzej Strojnowski. *Matematyka Stosowana - Optymalizacja II*. Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki, 2014.
- [6] Ponnuthurai Nagaratnam Suganthan Swagatam Das. Differential evolution: A survey of the state-of-the-art. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*,, 15(1):4–31, feb 2011.
- [7] Mitsuo Gen Xinjie Yu. *Introduction to Evolutionary Algorithms*. Decision Engineering. Springer, 2010.