

POLITECHNIKA WROCŁAWSKA

LABORATORIUM

INTELIGENCJA OBLICZENIOWA I JEJ ZASTOSOWANIA

---

# Algorytmy ewolucyjne i hybrydowe

---

*Authors:*

Rafał PIENIAŻEK  
Jakub POMYKAŁA

*Supervisor:*

prof. dr inż. Olgierd UNOLD

15 maja 2018

## Spis treści

## Spis rysunków

# 1 Wstęp

Celem laboratorium było przeprowadzenie optymalizacji globalnej dla wybranych funkcji z pakietu `globalOptTests`.

## 2 Zastosowany algorytm optymalizacji

W laboratorium zastosowano algorytmy genetyczne będące klasą algorytmów ewolucyjnych. Algorytmy ewolucyjne stanowią kierunek sztucznej inteligencji, która wykorzystuje i symuluje ewolucję biologiczną. Wszystkie algorytmy tej klasy symulują podstawowe zachowania w teorii ewolucji biologicznej - procesy selekcji, mutacji i reprodukcji. Zachowanie jednostek zależy od środowiska. Zbiór jednostek nazywa się populacją. Taka populacja ewoluuje zgodnie z regułami selekcji zgodnie z funkcją celu przypisaną do środowiska. Propagowane do kolejnych pokoleń są tylko najbardziej dopasowane osobniki.

### 2.1 Modyfikowane parametry

W celu badania wpływu zmiennych na zachowanie algorytmu genetycznego modyfikowano następujące parametry:

- **elitarność** - Procentowa wartość populacji, która zostaje przeniesiona do następnego pokolenia bez zmian. Niezerowa wartość gwarantuje, że jakość rozwiązania nie zmniejszy się z pokolenia na pokolenie.
- **mutacja** - Przypadkowa zmiana genomu w algorytmie genetycznym, analogicznie do mutacji biologicznej.
- **krzyżowanie** - Połączenie niektórych (wybranych losowo) genotypów w jeden. Kojarzenie ma sprawić, że potomek dwóch osobników rodzicielskich ma zespół cech, który jest kombinacją ich cech (może się zdarzyć, że tych najlepszych).
- **liczba iteracji** - Ilość pokoleń
- **rozmiar populacji** - Ilość osobników w jednym pokoleniu.

Tabela 1: Wartości modyfikowanych parametrów

Parametr	Wartości
populacja	50, 100, 150, 200, 250
liczba iteracji	50, 100, 150, 200, 250
p. krzyżowania	0, 0.25, 0.5, 0.75, 1
p. mutacji	0, 0.25, 0.5, 0.75, 1
p. populacji elitarniej	0, 0.25, 0.5, 0.75, 1

### 2.2 Zastosowane narzędzia implementacji

#### 2.2.1 Język R

R jest językiem programowania i środowiskiem programistycznym, używanym głównie do obliczeń statystycznych i wizualizacji danych, do sztucznej inteligencji a także do ekonomii i innych zagadnień wykorzystujących obliczenia numeryczne. Został stworzony przez Rossa Ihakę i Roberta Gentlemana na Uniwersytecie w Auckland w Nowej Zelandii.

### **2.2.2 Pakiet GA**

Pakiet GA zawiera zestaw funkcji ogólnego przeznaczenia do optymalizacji z wykorzystaniem algorytmów genetycznych. Dostępnych jest kilka operatorów genetycznych, których można łączyć w celu zbadania najlepszych ustawień dla bieżącego zadania.

### **2.2.3 Pakiet globalOpts**

Pakiet zawierający implementację funkcji przydatnych do przeprowadzania testów wydajnościowych algorytmów optymalizacji globalnej.

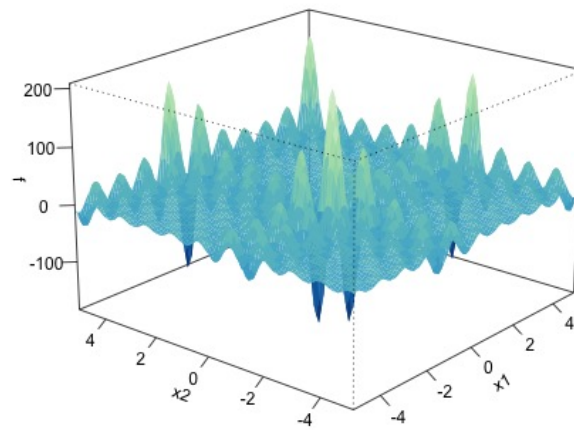
### 3 Funkcja Shuberta

#### 3.1 Wzór analityczny

$$f(\mathbf{x}) = \left( \sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left( \sum_{i=1}^5 i \cos((i+1)x_2 + i) \right)$$

Rysunek 1: Wzór analityczny funkcji Schuberta

#### 3.2 Wykres w ustalonym przedziale zmiennych

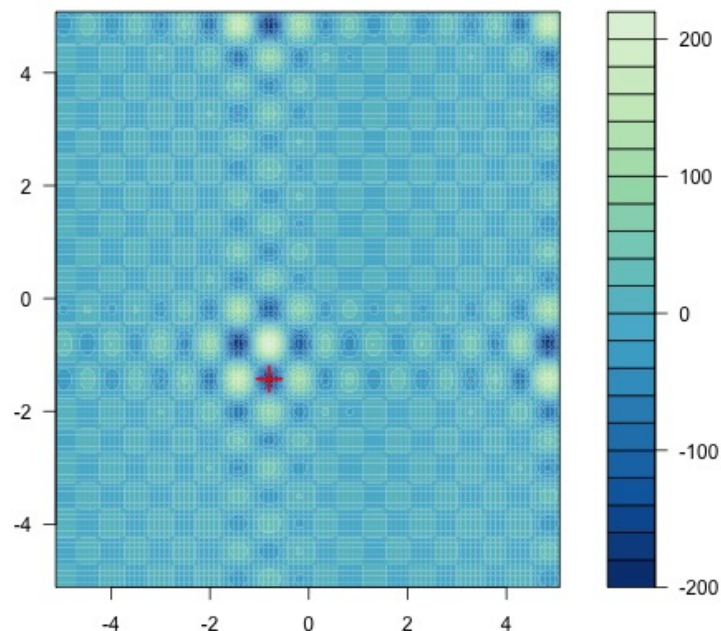


Rysunek 2: Wykres funkcji Schuberta

#### 3.3 Ekstremum globalne

$$f(\mathbf{x}^*) = -186.7309$$

Rysunek 3: Minimum globalne dla funkcji Schuberta



Rysunek 4: Minimum globalne dla funkcji Schuberta

### 3.4 Własne operatory krzyżowania i mutacji

#### 3.4.1 Zmiana funkcji krzyżowania

## 4 Wnioski

Badanie wpływu paramterów na jakość rozwiązań optymalnych poszukiwanych przez algorytm genetyczny jest zadaniem nietrywialnym, ale możliwym do wykonania. Zastosowanie pakietu GA dla języka R pozwoliło skupić się bardziej na testowaniu parametrów, a nie na implementacji algorytmu.

Warto zauważyć, że na jakość rozwiązań ma wpływ nie tylko wartość parametru, ale również funkcja poddawana testom. Dla tych samych zestawów danych wykresy testów różnią się pomiędzy testowanymi funkcjami. W przypadku poszukiwania ekstremum globalnego dla funkcji z wieloma ekstremami lokalnymi i niekorzystnym doborze parametrów algorytm może nie znaleźć poprawnego rozwiązania. Jest to spowodowane stochastycznym doбором populacji początkowej.

## 5 Literatura

1. Artur Suchwałko, "Wprowadzenie do R dla programistów innych języków", <https://cran.r-project.org/doc/contrib/R-dla-programistow-innych-jezykow.pdf>, 2014-02-23
2. Luca Scrucca, "On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution", <https://arxiv.org/pdf/1605.01931.pdf>, 2016-05-09

3. dr inż. Julian Sienkiewicz, "Pakiet R w analizie układów złożonych", <http://www.if.pw.edu.pl/~julas/CSAR/csar11.html>, 2017
4. W. N. Venables, D. M. Smith, R Core Team, "An Introduction to R", <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>, 2018-04-23
5. Luca Scrucca, "Package 'GA'", <ftp://cran.r-project.org/pub/R/web/packages/GA/GA.pdf>, 2016-09-29
6. Katharine Mullen, "Package 'globalOptTests'", <https://cran.r-project.org/web/packages/globalOptTests/globalOptTests.pdf>, 2015-02-15
7. Abdal-Rahman Hedar, "Global Optimization Test Problems", [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/TestGO.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm), dostęp online: 2018-05-04

## 6 Kod źródłowy