POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: Informatyka

SPECJANOŚĆ: Inżynieria Systemów Informatycznych (INS)

Inteligencja Obliczeniowa i jej Zastosowania

Sprawozdanie z laboratorium 3 - 5

AUTORZY:

Piotr Chorościn

Dawid Mikowski

PROWADZĄCY:

Dr hab. inż. Olgierd Unold,

prof. uczelni

OCENA PRACY:

WROCŁAW 2020

Spis treści

[Spis ilustracji 3](#_Toc41468891)

[Spis tabel 4](#_Toc41468892)

[Spis listingów 4](#_Toc41468893)

[1 Cel ćwiczenia 5](#_Toc41468894)

[2 Badanie efektywności własnych funkcji 6](#_Toc41468895)

[2.1 Funkcja Schuberta 6](#_Toc41468896)

[2.2 Funkcja Hosaki 8](#_Toc41468897)

[3 Problem komiwojażera 10](#_Toc41468898)

[4 Program genetyczny hybrydowy 11](#_Toc41468899)

[5 Tabele i wykresy z doświadczeń 11](#_Toc41468900)

[5.1 Badanie wpływu rozmiaru populacji 13](#_Toc41468901)

[5.1.1 Badanie wpływu rozmiaru populacji na przystosowanie dla funkcji Schuberta 13](#_Toc41468902)

[5.1.2 Badanie wpływu rozmiaru populacji na przystosowanie dla funkcji Hosaki 13](#_Toc41468903)

[5.1.3 Obserwacje nt. wpływu liczności populacji na jej średnie przystosowanie oraz przystosowanie najlepszego osobnika 14](#_Toc41468904)

[5.2 Badanie wpływu liczby pokoleń 15](#_Toc41468905)

[5.2.1 Badanie wpływu liczby pokoleń na przystosowanie dla funkcji Schuberta 15](#_Toc41468906)

[5.2.2 Badanie wpływu liczby pokoleń na przystosowanie dla funkcji Hosaki 16](#_Toc41468907)

[5.2.3 Obserwacje nt. wpływu liczby pokoleń na średnie przystosowanie populacji oraz przystosowanie najlepszego osobnika 17](#_Toc41468908)

[5.3 Badanie wpływu prawdopodobieństwa mutacji 18](#_Toc41468909)

[5.3.1 Badanie wpływu prawdopodobieństwa mutacji na przystosowanie dla funkcji Schuberta 18](#_Toc41468910)

[5.3.2 Badanie wpływu prawdopodobieństwa mutacji na przystosowanie dla funkcji Hosaki 19](#_Toc41468911)

[5.3.3 Obserwacje nt. wpływu prawdopodobieństwa mutacji na średnie przystosowanie populacji oraz przystosowanie najlepszego osobnika 20](#_Toc41468912)

[5.4 Badanie wpływu prawdopodobieństwa krzyżowania 21](#_Toc41468913)

[5.4.1 Badanie wpływu prawdopodobieństwa krzyżowania na wartość funkcji Schuberta 21](#_Toc41468914)

[5.4.2 Badanie wpływu prawdopodobieństwa mutacji na przystosowanie dla funkcji Hosaki 22](#_Toc41468915)

[5.4.3 Obserwacje nt. wpływu prawdopodobieństwa krzyżowania na średnie przystosowanie populacji oraz przystosowanie najlepszego osobnika 23](#_Toc41468916)

[5.5 Badanie wpływu prawdopodobieństwa krzyżowania i mutacji 24](#_Toc41468917)

[5.6 Badanie wpływu poziomu elitaryzmu 26](#_Toc41468918)

[5.6.1 Badanie wpływu elitaryzmu na przystosowanie do funkcji Schuberta 26](#_Toc41468919)

[5.6.2 Badanie wpływu elitaryzmu na przystosowanie do funkcji Hosaki 26](#_Toc41468920)

[5.6.3 Obserwacje nt. wpływu elitaryzmu na średnie przystosowanie populacji oraz przystosowanie najlepszego osobnika 27](#_Toc41468921)

[6 Wnioski 28](#_Toc41468922)

[6.1 Wnioski nt. własnych implementacji 28](#_Toc41468923)

[6.2 Wnioski nt. implementacji i języka R 29](#_Toc41468924)

[7 Kod z komentarzem 30](#_Toc41468925)

[7.1 Przegląd najważniejszych fragmentów kodu 30](#_Toc41468926)

[7.2 Pełny kod programu 33](#_Toc41468927)

[8 Literatura 37](#_Toc41468928)

# Spis ilustracji

[Rysunek 6.1Przebieg zmienności wartości funkcji dopasowania najlepszego osobnika i średniej dla całej populacji w zależności od pokolenia dla przykładowego uruchomienia algorytmu genetycznego dla funkcji Hosaki. 12](#_Toc41001908)

[Rysunek 6.2 Przebieg dla innego uruchomienia (też dla funkcji Hosaki). 12](#_Toc41001909)

[Rysunek 6.3 Zależność wartości funkcji celu Schuberta od liczebności populacji 13](#_Toc41001910)

[Rysunek 6.4 Zależność wartości funkcji celu Hosaki od liczebności populacji 13](#_Toc41001911)

[Rysunek 6.5 Zależność wartości funkcji celu Schuberta dla najlepszego osobnika z populacji oraz średniej wartości dla calej populacji w zależności od liczby pokoleń 15](#_Toc41001912)

[Rysunek 6.6 Powiększenie wykresu dla najlepszego osobnika 15](#_Toc41001913)

[Rysunek 6.7 Powiększenie wykresu dla średniej z całej populacji 15](#_Toc41001914)

[Rysunek 6.8 Zależność wartości funkcji celu Hosaki dla najlepszego osobnika z populacji oraz średniej wartości dla calej populacji w zależności od liczby pokoleń 16](#_Toc41001915)

[Rysunek 6.9 Powiększenie wykresu dla najlepszego osobnika 16](#_Toc41001916)

[Rysunek 6.10 Powiększenie wykresu dla średniej z całej populacji 16](#_Toc41001917)

[Rysunek 6.11 Zależność wartości funkcji celu Schuberta dla najlepszego osobnika z populacji oraz średniej wartości dla całej populacji w zależności od prawdopodobieństwa mutacji. 18](#_Toc41001918)

[Rysunek 6.12 Powiększenie wykresu dla najlepszego osobnika 18](#_Toc41001919)

[Rysunek 6.13 Powiększenie wykresu dla średniej z całej populacji 18](#_Toc41001920)

[Rysunek 6.14 Zależność wartości funkcji celu Schuberta dla najlepszego osobnika z populacji oraz średniej wartości dla całej populacji w zależności od prawdopodobieństwa mutacji. 19](#_Toc41001921)

[Rysunek 6.15 Powiększenie wykresu dla najlepszego osobnika 19](#_Toc41001922)

[Rysunek 6.16 Powiększenie wykresu dla średniej z całej populacji 19](#_Toc41001923)

[Rysunek 6.17 Zależność wartości funkcji celu Schuberta dla najlepszego osobnika z populacji oraz średniej wartości dla całej populacji w zależności od prawdopodobieństwa krzyżowania 21](#_Toc41001924)

[Rysunek 6.18 Powiększenie wykresu dla najlepszego osobnika 21](#_Toc41001925)

[Rysunek 6.19 Powiększenie wykresu dla średniej z całej populacji 22](#_Toc41001926)

[Rysunek 6.20 Zależność wartości funkcji celu Hosaki dla najlepszego osobnika z populacji oraz średniej wartości dla całej populacji w zależności od prawdopodobieństwa krzyżowania 22](#_Toc41001927)

[Rysunek 6.21 Powiększenie wykresu dla najlepszego osobnika 23](#_Toc41001928)

[Rysunek 6.22 Zależność wartości funkcji celu Schuberta dla najlepszego osobnika z populacji oraz średniej wartości dla całej populacji w zależności od wartości parametru „elitism” 26](#_Toc41001929)

[Rysunek 6.23 Zależność wartości funkcji celu Hosaki dla najlepszego osobnika z populacji oraz średniej wartości dla całej populacji w zależności od wartości parametru „elitism” 26](#_Toc41001930)

[Rysunek 6.24 Powiększenie wykresu dla najlepszego osobnika. 27](#_Toc41001931)

# Spis tabel

[Tabela 6.1 Wartości parametrów i zmiennych przyjęte w badaniach 11](#_Toc41004859)

[Tabela 6.2 Poszczególne osobniki populacji w 1. i 60. pokoleniu w zależności od wartości prawdopodobieństw: mutacji i krzyżowania 24](#_Toc41004860)

# Spis listingów

[Listing 8.1 Kod funkcji objective.fun.of 30](#_Toc41004929)

[Listing 8.2 Kod funkcji objective.fun.get 30](#_Toc41004930)

[Listing 8.3 Kod funkcji objective.fun.plot 31](#_Toc41004931)

[Listing 8.4 Kod funkcji GA.run.iterations 31](#_Toc41004932)

[Listing 8.5 Kod funkcji GA.run.experiment 31](#_Toc41004933)

[Listing 8.6 Kod funkcji GA.run.experiment.list 32](#_Toc41004934)

[Listing 8.7 Wywołanie funkcji GA.run.experiment 32](#_Toc41004935)

[Listing 8.8 Funkcja GA.run.once 33](#_Toc41004936)

[Listing 8.9 Skrypt global\_opt.R 33](#_Toc41004937)

[Listing 8.10 Skrypt global\_opt.R 35](#_Toc41004938)

[Listing 8.11 Skrypt main.R 36](#_Toc41004939)

# Cel ćwiczenia

Celem ćwiczenia jest:

* sprawdzenie efektywności funkcji wielomodalnych z zaimplementowanymi własnymi funkcjami krzyżowania oraz mutacji a następnie porównanie jakości działania algorytmu z funkcjami domyślnymi,
* wykonanie wcześniejszych zadań dla 3 wybranych parametrów dla problemu komiwojażera,
* wykonanie z wykorzystaniem programu genetycznego hybrydowego zadania rozwiązania problemu komiwojażera.

# Badanie efektywności własnych funkcji

## badana funkcja

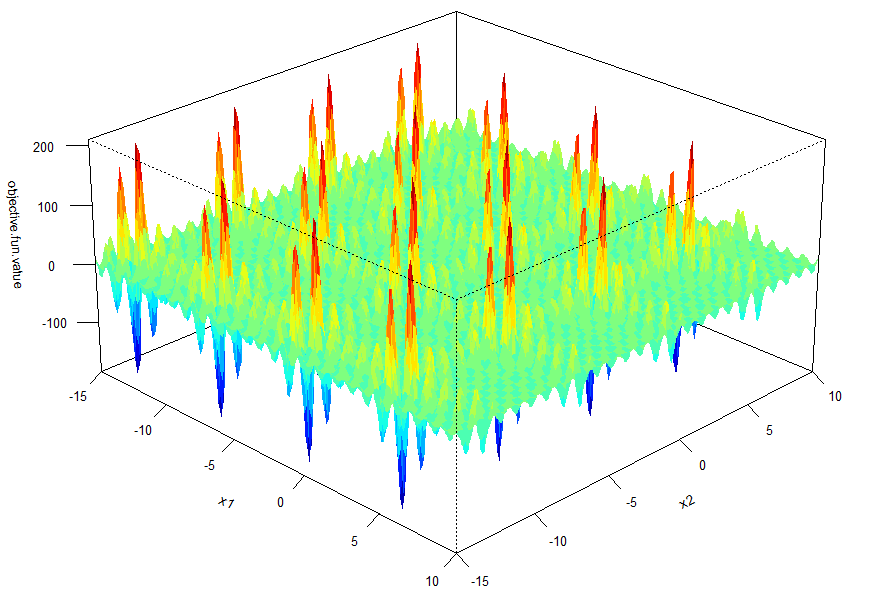
Do testowania efektywności własnych funkcji krzyżowania i mutacji potrzebne było wybranie jednej funkcji wielomodalnej na której sprawdzane będą wyniki. Została do tego wyznaczona funkcja, która była testowana także w poprzednim laboratorium a mianowicie funkcja Schuberta.

Funkcja Schuberta określona jest poniższym wzorem [1]:

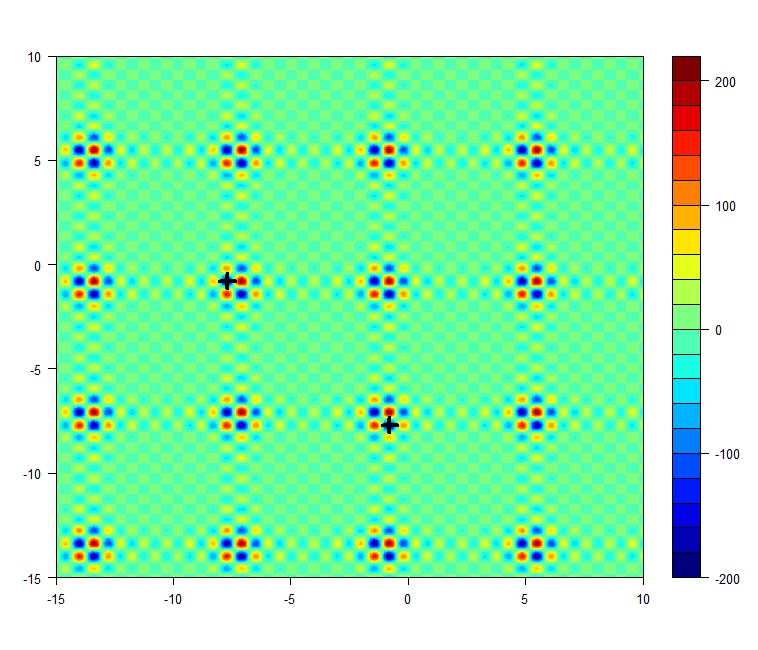
Wykres funkcji Schuberta, wygenerowany dla następujących wartości zmiennych:

* z krokiem próbkowania 0,1,
* z krokiem próbkowania 0,1,

Są to domyślne zakresy zmiennych z pakietu globalOptTest dla funkcji Schuberta.



Powyższy trójwymiarowy wykres przedstawiony na płaszczyźnie poprzez rzut z góry przedstawiono na poniższym rysunku.



Zgodnie z informacjami z pakietu GlobalOptTest, funkcja Schuberta w minimum globalnym przyjmuje wartość **.**

## Sposób przeprowadzenia badań

Dla funkcji Schuberta przyjęto zakres wartości zmiennych odczytany za pomocą funkcji *getDefaultBounds* z pakietu globalOptTests oraz wartość ekstremum globalnego odczytaną za pomocą funkcji getGlobalOpt. Funkcję poddano optymalizacji globalnej stosując gotowy algorytm genetyczny zaimplementowany w pakiecie do języka R o nazwie „GA”, który umożliwia również hybrydyzację. Podczas badania algorytmu hybrydowego zostały ustawione domyślne wartości wszystkich parametrów. Aby zbadać wpływ danego operatora, wszystkie operatory poza badanym miały ustawioną stałą wartość domyślną. W poniższej tabeli zawarte są wszystkie badane parametry oraz wartości domyślne.

|  |  |  |
| --- | --- | --- |
| Operator | Badane parametry | Wartość domyślna |
| Krzyżowania | la, blx, laplace, sp, wa | la |
| Mutacji | ra, pow, rs | ra |
| Selekcji | ls, lr, nlr, rw, sigma, tour | ls |

Domyślne wartości dla parametrów algorytmu genetycznego prezentują się następująco:

|  |  |
| --- | --- |
| Type | Real-valued |
| Population size | 50 |
| Number of generations | 100 |
| Elitism | 5 |
| Crossover possibility | 0.8 |
| Mutation propability | 0.1 |

## Funkcja krzyżowania

Własna implementacja funkcji krzyżowania:

Aby poprawnie wykonać badania należało także zastąpić funkcję w algorytmie genetycznym w następujący sposób:

## Funkcja mutacji

Własna implementacja funkcji selekcji populacji:

Wykonane zostało porównanie jakości wyników dla poszczególnych parametrów selekcji przy zachowaniu domyślnych parametrów algorytmu genetycznego:

## Wnioski

# Problem komiwojażera

## Instancje

Do badania algorytmu genetycznego dla problemu komiwojażera zdecydowano się wykorzystać dwie instancje ze zbioru TSPlib.

|  |  |  |
| --- | --- | --- |
| Nazwa instancji | Liczba miast | Długość najkrótszej trasy |
|  |  |  |
|  |  |  |

Wyniki algorytmu genetycznego zostały przeprowadzone na trzech instancjach z których każda ma inną liczbę wierzchołków i przedstawia wersję symetryczną badanego problemu. Instancje zostały wybrane biorąc pod uwagę czasochłonność wykonywanych obliczeń. Na wykresy naniesiono najkrótsze trasy dla każdej instancji, aby można było w prosty graficzny sposób uzyskać odniesienie co do jakości uzyskanych rozwiązań. Jak widać na wykresach jakość rozwiązań dla domyślnych wartości parametrów oraz domyślnych operatorów, znacznie odbiega od rozwiązania optymalnego.

# Program genetyczny hybrydowy

Wykorzystano następujące paczki języka R:

* GA – zawierająca implementację algorytmu genetycznego,
* GlobalOptTest – zawierająca funkcje celu wraz z wartościami ich ekstremów.
* TSPlib – zawierająca funkcje pozwalające na wczytanie oraz rozwiązanie plików związanych z problemem komiwojażera

Środowisko implementacyjne:

* Interpreter języka R w wersji 3.6.1 z pakietu Anaconda,
* Środowisko VS code z wtyczką do języka R

# Tabele i wykresy z doświadczeń

# Wnioski

## Wnioski nt. własnych implementacji

## Wnioski nt. implementacji i języka R

* język R stanowi bardzo ciekawą alternatywę do innych języków programowania,
* szczególnie przydatne są funkcyjne aspekty tego języka, które umożliwiły implementację skryptów bez zbędnego narzutu programistycznego (np. zwracanie funkcji jako wynik funkcji, która jako argument przyjmuje nazwę funkcji z pakietu globalOptTest),
* jeden z autorów sprawozdania miał okazję wcześniej implementować już podobne zadanie (z własną implementacją algorytmu genetycznego) w języku Kotlin. Ilość kodu potrzebna np. na przeprowadzenie badań (pomijając implementację samego algorytmu, której w tym przypadku nie było – skorzystano z gotowej paczki GA) w przypadku języka R jest znacząco niższa. Prostsze jest również generowanie wykresów,
* język R zawiera jednak kilka „puapek”. Jedną z nich jest problem, na który spowodował opóźnienie w oddaniu listy: podając nazwę elementu listy można bowiem używać składni nazwa.listy$nazwa.elementu lub nazwa.listy[[nazwa.elementu]]. Okazało się że jedynie ten drugi sposób umożliwia przekazanie nazwy elementu jako zmiennej typu string. Początkowo użyta składnia nazwa.listy$nazwa.elementu (gdzie nazwa.elementu to zmienna typu string) nie powodowała błędu, ale tworzyła w liście nowy element o nazwie „nazwa.elementu”. Stąd znalezienie tego błędu, znacząco wydłużyło czas implementacji.

# Kod z komentarzem

Komentarze umieszczono w kodzie w języku angielskim, dla zachowania spójności z nazewnictwem metod.

## Przegląd najważniejszych fragmentów kodu

# Literatura

[1] <http://infinity77.net/global_optimization/test_functions_nd_H.html>

[2] <https://www.sfu.ca/~ssurjano/shubert.html>

[3] <https://pl.wikipedia.org/wiki/Algorytm_genetyczny>

[4] <https://cran.r-project.org/web/packages/GA/vignettes/GA.html>