

Politechnika Krakowska

Wydział Informatyki i Telekomunikacji

Studia Stacjonarne

Sprawozdanie z projektu 1:

**Implementacja klasycznego algorytmu genetycznego**

Wykonali:

Dawid Wilgucki

Magdalena Skwarczek

Mateusz Stec

Mikołaj Wnuk

**1. Wprowadzenie**

Projekt polega na implementacji klasycznego algorytmu genetycznego w języku Python. Celem jest rozwiązanie problemu optymalizacji funkcji wielu zmiennych poprzez maksymalizację oraz minimalizację wartości funkcji testowych.

**2. Technologie wykorzystane w projekcie**

* **Język programowania:** Python
* **Biblioteki:** numpy, matplotlib, tkinter, sqlite3
* **Środowisko programistyczne:** PyCharm / VS Code
* **System kontroli wersji:** Git (GitHub / GitLab)

**3. Wymagania środowiska**

Aby uruchomić aplikację, należy mieć zainstalowane:

* Python w wersji 3.8+
* Poniższe biblioteki:

pip install numpy matplotlib tkinter sqlite3

* System operacyjny: Windows/Linux/MacOS

**4. Opis implementacji algorytmu genetycznego**

Algorytm składa się z następujących etapów:

1. Inicjalizacja populacji chromosomów w reprezentacji binarnej.
2. Ocena populacji na podstawie funkcji celu.
3. Selekcja najlepszych osobników metodą turniejową lub ruletki.
4. Krzyżowanie: jednopunktowe, dwupunktowe, jednorodne, ziarniste.
5. Mutacja: bit flip, brzegowa, dwupunktowa.
6. Inwersja z określonym prawdopodobieństwem.
7. Strategia elitarna zapewniająca przejście najlepszych osobników do następnej populacji.

**5. Opis testowanych funkcji optymalizacyjnych**

Do testów wybrano dwie funkcje:

1. **Łatwa funkcja testowa:** Rastrigin
   * Wzór funkcji:
   * Rzeczywiste optimum:
2. **Trudna funkcja testowa:** Hyperellipsoid
   * Wzór funkcji:
   * Rzeczywiste optimum:

Testy przeprowadzono dla 10, 20 i 30 zmiennych.

**6. Wyniki eksperymentów**

**6.1 Wykres wartości funkcji od iteracji**

(Wstawione wykresy przedstawiające zmiany wartości funkcji w kolejnych iteracjach)

**6.2 Średnia wartość funkcji i odchylenie standardowe**

(Wstawione wykresy średnich wartości funkcji i ich odchylenia standardowego)

**6.3 Porównanie wyników dla różnych konfiguracji**

**6.4 Porównanie czasu obliczeń dla różnych konfiguracji**

**7. Podsumowanie**

Algorytm genetyczny pozwolił znaleźć optymalne lub bliskie optymalnym wartości funkcji testowych. Wyniki wykazały, że większa liczba generacji oraz większa populacja zwiększają dokładność wyników, ale kosztem czasu obliczeń.

**8. Załączniki**

1. **Kod źródłowy** – zawarty
2. **Nagranie wideo** – demonstracja działania aplikacji.
3. **Pliki wynikowe** – zapisane w bazie SQLite (result.db) oraz plikach CSV.