

Projekt 1 – implementacja klasycznego algorytmu genetycznego

1. Dobierz się w grupy 2-3 osobowe do realizacji projektów w ramach przedmiotu obliczenia ewolucyjne.
2. Zaimplementuj algorytm genetyczny celem rozwiązywania problemu optymalizacji (maksymalizacji oraz minimalizacji funkcji wielu zmiennych). Implementacja powinna dać możliwość wyboru dowolnej liczby zmiennych w konfiguracji (to znaczy powinna być możliwość przetestowania algorytmu np. dla funkcji 5, 10, 20 czy 27 zmiennych).
3. Rekomenduje się wykonanie projektu w języku Python.
4. Zawartość projektu:
 - a) Implementacja binarnej reprezentacji chromosomu + konfiguracja dokładności
 - b) Implementacja konfiguracji wielkości populacji
 - c) Implementacja konfiguracji liczby epok
 - d) Implementacja metod selekcji najlepszych, kołem ruletki, selekcji turniejowej + konfiguracje parametrów
 - e) Implementacja krzyżowania jednopunktowego, dwupunktowego, krzyżowania jednorodnego, krzyżowania ziarnistego + konfiguracja prawdopodobieństwa krzyżowania.
 - f) Implementacji mutacji brzegowej, jedno oraz dwupunktowej + konfiguracja prawdopodobieństwa mutacji
 - g) Implementacja operatora inwersji + konfiguracja prawdopodobieństwa jego wystąpienia
 - h) Implementacja strategii elitarniej + konfiguracja % lub liczby osobników przechodzącej do kolejnej populacji
5. W ramach zespołu proszę wybrać jedną funkcję do testów z pliku FunkcjeTestowe.pdf (łatwiejsze funkcje, gdzie znalezienie optimum nie powinno być trudne) oraz jedną funkcję z pliku Cec2014.pdf bądź dowolną inną funkcję z konferencji CEC (trudniejsze funkcje, może nie udać się znaleźć optimum, ale warto poeksperymentować by znaleźć rozwiązanie jak najlepsze). Wybrane funkcje powinny być niepowtarzające się na roku, ale zachęcam by zaimplementować sobie ich więcej celem testów i eksperymentów. Proszę wybrać też takie funkcje, które mają możliwość ustawienia różnej liczby zmiennych.

Proszę też nie implementować tych funkcji samodzielnie tylko skorzystać z gotowych implementacji:

https://gitlab.com/luca.baronti/python_benchmark_functions

https://github.com/thieu1995/opfunu/tree/master/opfunu/cec_based

6. Proszę wykonać aplikację w formie graficznej (nie console application).
7. Proszę dać możliwość konfiguracji aplikacji z poziomu gui.
8. Proszę wyświetlić w aplikacji czas wykonywanych obliczeń.
9. Proszę przygotować w aplikacji możliwość wygenerowania następujących wykresów:
 - a) Wartości funkcji od kolejnej iteracji
 - b) Średniej wartości funkcji, odchylenia standardowego od kolejnej iteracji
10. Proszę zapisać do pliku lub bazy danych (np. sqlite) kolejne wyniki algorytmu w poszczególnych iteracji.
11. Przygotuj sprawozdanie do projektu. Powinno ono zawierać:
 - a) Informacje o wykorzystywanych technologiach do wykonania projektu
 - b) Wymaganiach środowiska do uruchomienia aplikacji
 - c) Wybrane funkcje, które będą testowane wraz z rysunkiem funkcji oraz wartościami optimum wraz z argumentami dla których to optimum jest osiągnięte. Jeżeli w sprawozdaniu testują Państwo funkcje 10, 20, 30 zmiennych to proszę zamieścić informacje o optimach i argumentach dla których ono zachodzi dla 10, 20 i 30 zmiennych, tak by można było łatwo porównać wyniki które Państwo osiągnęli ewolucyjną optymalizacją z rzeczywistymi wartościami optimum
 - d) Wykresy zależności wartości funkcji celu od kolejnej iteracji
 - e) Średniej wartości funkcji celu oraz odchylenia standardowego w kolejnej iteracji
 - f) Porównanie osiągniętych wyników przy różnych konfiguracjach algorytmu + porównanie czasu obliczeń. Każda konfiguracja którą uruchomicie powinna być powtórzona przynajmniej 10 razy – tak by wyniki zostały odpowiednio uśrednione i wolne od efektu losowości. W sprawozdaniu proszę zamieścić średnie wyników z 10 uruchomień, najlepszy wynik oraz najgorszy wynik. Wykresy z podpunktów d oraz e proszę zamieścić tylko z najlepszego uruchomienia.

- g) W podsumowaniu zamieść tabelkę zawierającą nazwę optymalizowanej funkcji, liczbę zmiennych, rzeczywista wartość optimum, wartość optimum którą udało się uzyskać z wykorzystaniem algorytmu genetycznego oraz błąd – czyli różnicę rzeczywistej wartości z osiągniętą.
12. Proszę zamodelować aplikację z wykorzystaniem obiektowego paradygmatu programowania z uwzględnieniem wszelkich dobrych praktyk wytwarzania oprogramowania.
13. Nagraj krótkie video (około minuty, dwóch) przedstawiające działanie projektu w praktyce. Nagranie dołącz do sprawozdania.
14. Projekt i sprawozdanie wgraj na platformę Delta – 1 członek zespołu. Proszę wgrać zarówno kod źródłowy, sprawozdanie jak i nagranie video w postaci archiwum ZIP o nazwie P1_Nazwisko1Imie1_Nazwisko2Imie2_Nazwisko3Imie3.zip