

Algorytmy genetyczne i Sztuczne sieci neuronowe

Lista 5. - Modyfikacja algorytmu genetycznego

Celem niniejszej listy zadań jest zaproponowanie i przeanalizowanie modyfikacji algorytmu genetycznego na podstawie operatorów omówionych na wykładzie lub dostępnych w literaturze. Zmiany powinny obejmować różne elementy algorytmu, takie jak generowanie populacji, selekcja, krzyżowanie, mutacja oraz kryteria stopu. Należy również zaproponować metody ewaluacji wpływu zastosowanych modyfikacji.

Zadania do wykonania:

1. Modyfikacja procesu generowania populacji początkowej:

- Zaimplementować nowe podejście do inicjalizacji populacji, np. losowanie według określonych rozkładów (jednostajnego, normalnego) lub z uwzględnieniem heurystyk dopasowanych do rozwiązywanego problemu.
- Porównać skuteczność tradycyjnej, losowej inicjalizacji i zmodyfikowanego podejścia.

2. Implementacja nowych metod selekcji:

- Przetestować metody selekcji, **które nie były wcześniej zaimplementowane**, takie jak:
 - Turniejowa z różną liczbą uczestników.
 - Rankingowa z różnymi współczynnikami skali.
 - Ruletkowa z adaptacyjnym prawdopodobieństwem wyboru.
- Porównać wyniki nowych metod z metodami z poprzednich list zadań.

3. Zastosowanie alternatywnych operatorów krzyżowania:

- Zaimplementować operatory krzyżowania, **które nie były wcześniej zaimplementowane**.
- Określić szybkość znalezienia optymalnego rozwiązania dla każdego z rozważanych zbiorów danych.

4. Modyfikacja operatora mutacji:

- Zaimplementować nowe metody mutacji, np.:
 - Mutację gaussowską.
 - Mutację dynamiczną zmieniającą się w zależności od numeru pokolenia.
- Porównać wyniki z mutacją jednopunktową zastosowaną w poprzednich listach zadań.
- Określić szybkość znalezienia optymalnego rozwiązania dla każdego z rozważanych zbiorów danych.

5. Monitorowanie statystyk populacji:

- Obliczyć i analizować:
 - Wartość funkcji przystosowania dla najlepszego i najgorszego osobnika w każdym pokoleniu.
 - Średnią wartość funkcji przystosowania dla całej populacji.
 - Rozkład przystosowania w populacji.
- Przechowywać dane statystyczne dla każdej iteracji w celu późniejszej wizualizacji.

6. Przygotowanie wykresów i wizualizacji wyników:

- Opracować wykresy pokazujące:
 - Zmiany wartości funkcji przystosowania najlepszego i najgorszego osobnika na przestrzeni pokoleń.
 1. Oś X: numer pokolenia.
 2. Oś Y: wartość funkcji przystosowania dla najlepszego i najgorszego osobnika w danym pokoleniu.
 - Średnią wartość funkcji przystosowania w populacji.
 1. Oś X: numer pokolenia.
 2. Oś Y: średnia wartość funkcji przystosowania.
 - Porównanie liczby pokoleń wymaganych do osiągnięcia optymalnego rozwiązania dla różnych modyfikacji.
 1. Oś X: rodzaj modyfikacji.
 2. Oś Y: liczba pokoleń.

7. Sformułowanie wniosków:

- Przeanalizować wyniki eksperymentów.
- Wskazać, jakie modyfikacje algorytmu najbardziej wpływają na jego efektywność oraz w jakich sytuacjach ma to miejsce.
- Opisać, jak różne modyfikacje wpływają na tempo znalezienia optymalnego rozwiązania oraz jakość rozwiązań uzyskiwanych w poszczególnych pokoleniach.

8. Warunki opracowywanych rozwiązań:

- Opracowane rozwiązania powinny być implementowane w języku Python. Dopuszczalne jest wykorzystanie innego języka programowania pod warunkiem uzyskania zgody prowadzącego.
- Podczas implementacji wolno posługiwać się bibliotekami do obliczeń numerycznych ogólnego przeznaczenia (np. NumPy, SciPy) oraz przetwarzania danych (np. Pandas).
- Podczas implementacji nie wolno posługiwać się dedykowanymi bibliotekami do tworzenia algorytmów genetycznych (np. PyGAD, DEAP itp.) oraz architektur sieci neuronowych (np. scikit-learn, PyTorch, Tensorflow, JAX itp.).