

# Algorytmy genetyczne i Sztuczne sieci neuronowe

## Lista 3. - Operatory selekcji i mutacji

1. Zapoznać się z podstawowymi zasadami działania selekcji i mutacji w algorytmach genetycznych.
2. Opracować algorytm selekcji turniejowej:
  - a. Zaimplementować proces losowego wyboru osobników do turniejów.
  - b. Zaprojektować mechanizm wyboru najlepszego osobnika z każdego turnieju.
3. Opracować algorytm selekcji proporcjonalnej (metoda ruletki):
  - a. Obliczyć prawdopodobieństwa selekcji na podstawie wartości funkcji przystosowania.
  - b. Zaimplementować mechanizm wyboru osobników na podstawie tych prawdopodobieństw.
4. Zaprojektować operator mutacji punktowej:
  - a. Określić mechanizm losowego wyboru genów do mutacji.
  - b. Zdefiniować zmiany w wybranym genie zgodnie z założeniami problemu.
5. Porównać efektywność opracowanych metod selekcji:
  - a. Uruchomić testowe iteracje algorytmu genetycznego z różnymi operatorami selekcji.
  - b. Przeanalizować wpływ na szybkość i jakość otrzymywanych rozwiązań.
6. Zbadać wpływ operatora mutacji na różnorodność populacji:
  - a. Zmierzyć zmienność genetyczną w populacji po kilku iteracjach.
  - b. Opracować wnioski dotyczące wpływu parametrów mutacji na proces optymalizacji.

## **7. Warunki opracowywanych rozwiązań:**

- a. Opracowane rozwiązania powinny być implementowane w języku Python. Dopuszczalne jest wykorzystanie innego języka programowania pod warunkiem uzyskania zgody prowadzącego.
- b. Podczas implementacji wolno posługiwać się bibliotekami do obliczeń numerycznych ogólnego przeznaczenia (np. NumPy, SciPy) oraz przetwarzania danych (np. Pandas).
- c. 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.).