Algorytmy Ewolucyjne

Raport
Indeks: 327295

Michał Matuszyk

Czerwiec 2025

Streszczenie

W pracy przedstawiono trzy warianty algorytmu genetycznego (AG) i oceniono ich skuteczność na różnych zadaniach optymalizacyjnych. W pierwszym etapie (AE1) zaimplementowano podstawowy AG z mutacją gaussowską i krzyżowaniem jednopunktowym, testując go na funkcji kwadratowej $f(x, y, z) = x^2 + y^2 + 2z^2$ oraz pięciowymiarowej funkcji Rastrigina. Wyniki dowiodły szybkości i stabilności zbieżności do minimum globalnego przy niewielkich populacjach. W drugim etapie (AE2) AG zastosowano do wariantu problemu *cutting stock* w otoczeniu kołowym, gdzie celem było maksymalizowanie wartości układu prostokątów. Pomimo poprawnej implementacji operatorów genetycznych, rezultaty były ograniczone przez złożoność przestrzeni rozwiązań. W trzecim etapie (AE3) zaproponowano ewolucyjną optymalizację wag i biasów wielowarstwowego perceptronu (MLP). Eksperymenty na zbiorach Iris, Multimodal-large oraz Auto-MPG wykazały doskonałą dokładność klasyfikacji (do 100%) oraz konkurencyjne wartości błędu średniokwadratowego (MSE). Ostateczne wnioski potwierdzają uniwersalność i elastyczność algorytmów genetycznych, a także wskazują na potencjał dalszego rozwoju poprzez hybrydyzację z metodami lokalnymi oraz automatyczne dostrajanie parametrów.

Spis treści

1 Wstęp

Algorytmy genetyczne (AG) to klasa metaheurystyk inspirowanych mechanizmami ewolucji biologicznej, w której populacja rozwiązań jest stopniowo "doskonalona"za pomocą operatorów selekcji, krzyżowania i mutacji. Dzięki elastyczności i zdolności do globalnego przeszukiwania przestrzeni rozwiązań, AG znajdują zastosowanie w szerokim spektrum problemów optymalizacyjnych, zarówno klasycznych funkcji testowych, jak i zadań rzeczywistych, takich jak problemy rozmieszczania prostokątów czy uczenie parametrów sieci neuronowych.

Celem niniejszej pracy jest przeprowadzenie serii eksperymentów z wykorzystaniem różnych wariantów algorytmu genetycznego:

- W części 2 (AE1) zaimplementowano podstawowy AG z mutacją gaussowską i krzyżowaniem jednopunktowym, testując go na funkcji kwadratowej $f(x, y, z) = x^2 + y^2 + 2z^2$ oraz pięciowymiarowej funkcji Rastrigina.
- W części 3 (AE2) zastosowano AG do rozwiązania wariantu problemu *cutting stock* w otoczeniu kołowym, gdzie celem było maksymalizowanie wartości układu prostokątów umieszczanych w kole.
- W części 4 (AE3) opracowano algorytm genetyczny do ewolucyjnej optymalizacji wag i biasów wielowarstwowego perceptronu (MLP), porównując wyniki na zbiorach Iris, Multimodal-large oraz Auto-MPG.

Struktura pracy jest następująca:

1. W części 2 przedstawiono implementację podstawowego AG oraz omówiono wyniki optymalizacji na dwóch funkcjach testowych.
2. W części 3 opisano kodowanie rozwiązania, operatorów genetycznych oraz przedstawiono przykładowe efekty zastosowania AG w problemie układania prostokątów (cutting stock problem).
3. W części 4 zademonstrowano rozszerzenie podejścia na zadania klasyfikacji i regresji z wykorzystaniem ewolucyjnej optymalizacji parametrów sieci neuronowej.

Podsumowanie uzyskanych wyników oraz wnioski końcowe zamieszczono w rozdziale 5.

2 AE1

2.1 Cel eksperymentu

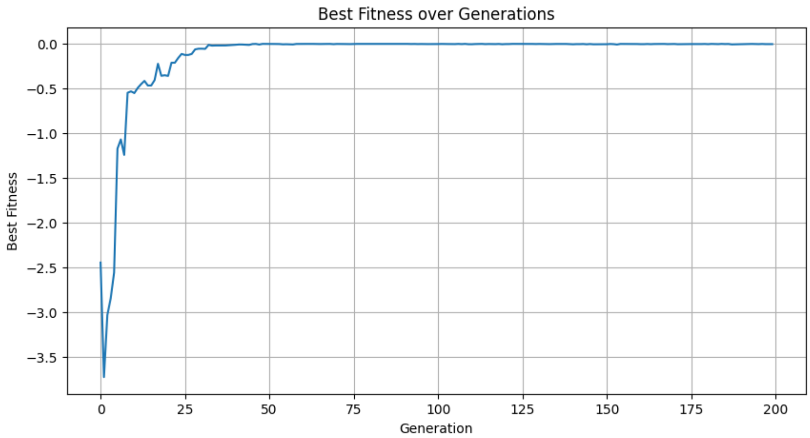
Celem tego zadania było zaimplementowanie podstawowego algorytmu genetycznego z:

- mutacją gaussowską,
- krzyżowaniem jednopunktowym.

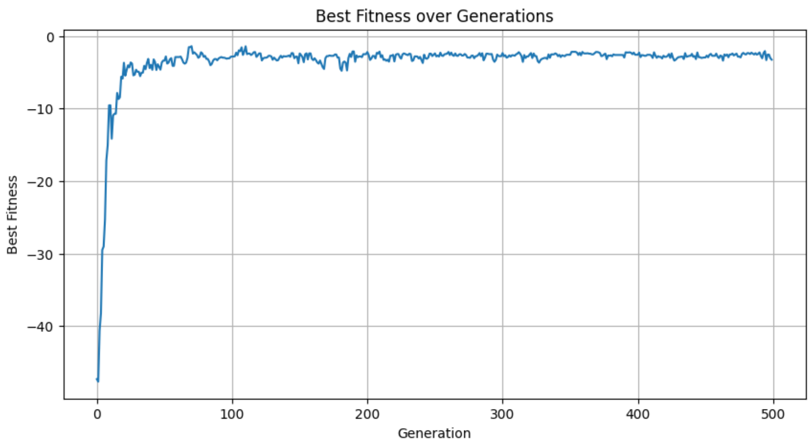
Algorytm przetestowano na dwóch funkcjach:

$$f(x, y, z) = x^2 + y^2 + 2z^2, \qquad g(\mathbf{x}) = \sum_{i=1}^{5} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$$

gdzie g to pięciowymiarowa funkcja Rastrigina.



Rysunek 1: Przebieg optymalizacji funkcji $x^2 + y^2 + 2z^2$: wartość funkcji celu w kolejnych epokach.



Rysunek 2: Przebieg optymalizacji pięciowymiarowej funkcji Rastrigina: wartość funkcji celu w kolejnych epokach.

2.2 Wyniki dla funkcji $x^2 + y^2 + 2z^2$

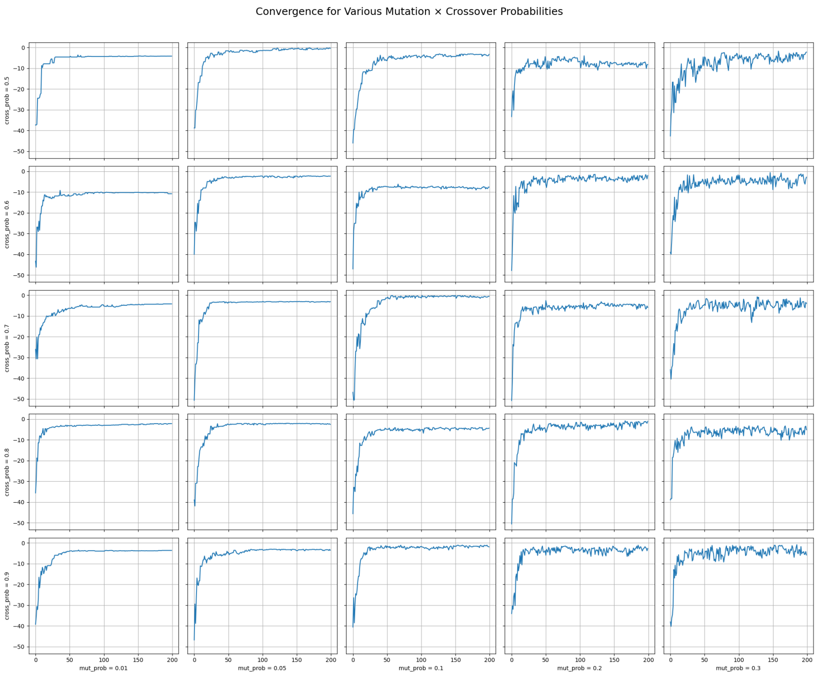
2.3 Wyniki dla funkcji Rastrigina

2.4 Eksperymenty własne

W ramach dodatkowych eksperymentów przeanalizowano wpływ dwóch parametrów:

1. prawdopodobieństwa mutacji,

2. wielkości populacji.



Rysunek 3: Wpływ prawdopodobieństwa mutacji na szybkość i jakość zbieżności przy stałym prawdopodobieństwie krzyżowania.

Przy rosnącej populacji, algorytm zbiega szybciej oraz z większym prawdopodobieństwem trafia do „mniejszego" minimum. Dla populacji równej 10 często obserwowano utknięcie w lokalnym minimum, natomiast przy populacjach ≥ 400 problem ten zanika.

Z powyższego wykresu wynika, że nawet niewielkie prawdopodobieństwo mutacji (już po 50 epokach) zapewnia relatywnie szybką zbieżność. Wyższe wartości prawdopodobieństwa krzyżowania zwiększają jednak oscylacje wokół minimum, co obniża stabilność wyników.

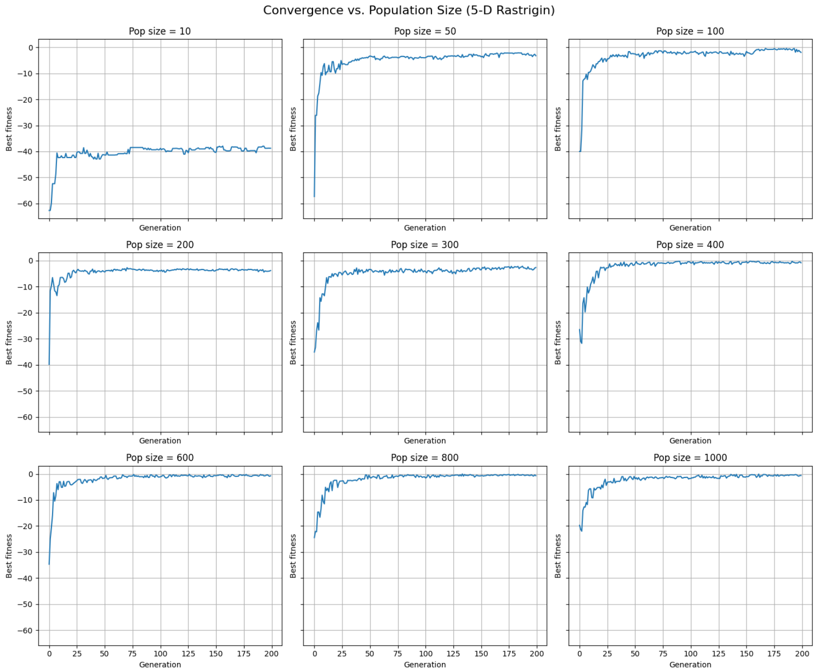
3 AE2

3.1 Cel

Rozwiązać wariant problemu *cutting stock* w otoczeniu kołowym. Mamy:

- koło o promieniu r,
- zbiór dostępnych prostokątów, każdy określony przez: szerokość w, wysokość h oraz wartość v.

Celem jest ułożenie prostokątów (z możliwością wielokrotnego użycia każdego typu) wewnątrz koła tak, aby:



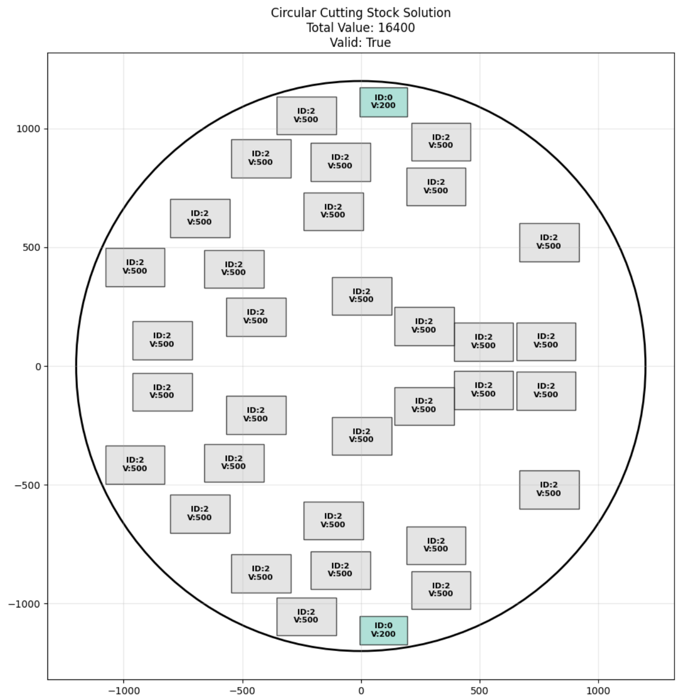
Rysunek 4: Wpływ wielkości populacji na szybkość zbieżności algorytmu.

1. boki prostokątów były równoległe do osi układu współrzędnych,
2. wnętrza prostokątów nie nachodziły na siebie (stykanie bokami jest dozwolone),
3. zmaksymalizować sumę wartości wszystkich umieszczonych prostokątów.

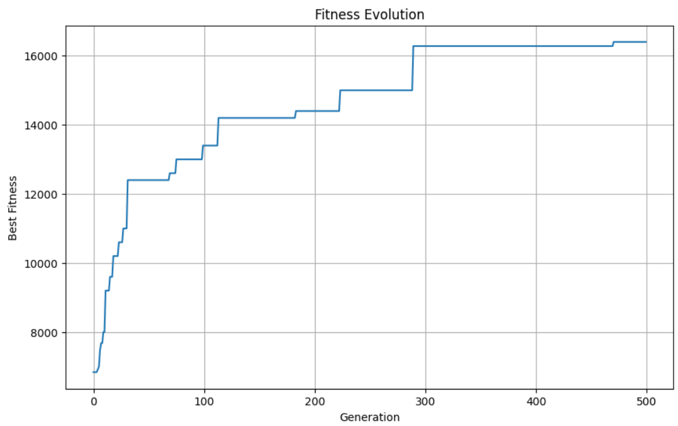
3.2 Opis rozwiązania

- **Kodowanie (genotyp):** każdy *Gen* to krotka $(x, y, \text{rect\_id})$, gdzie (x, y) to środek prostokąta, a rect_id – indeks typu prostokąta.
  Symetrię wzdłuż osi x uzyskuje się przez lustrzane odbicie genów o dodatnich y.
- **Inicjalizacja:** generacja losowych osobników z losową liczbą prostokątów, umieszczanych przypadkowo w górnej połowie koła, tylko jeśli mieszczą się w obszarze.
- **Ocena przystosowania:**
  1. Sprawdzenie, czy każdy prostokąt wraz z odbiciem mieści się wewnątrz koła.
  2. Sprawdzenie braku nakładania się żadnych par prostokątów.
  3. Jeśli osobnik jest ważny, fitness = suma wartości prostokątów (z uwzględnieniem podwójnej wartości za odbicie).
- **Selekcja:** turniejowa ($k = 3$), czyli losujemy 3 osobniki i wybieramy najlepszego.
- **Krzyżowanie:** nieklasyczne *jednopunktowe*-tworzymy pulę genów obu rodziców, a następnie losowo wybieramy próbkę generatorów do dziecka.
- **Mutacja:**
  1. Z losowym prawdopodobieństwem przesuwamy istniejące geny w nowe losowe pozycje.
  2. Z prawdopodobieństwem p dodajemy nowy gen (jeśli nie przekroczono limitu).
  3. Z prawdopodobieństwem p usuwamy losowy gen (jeśli pozostało co najmniej 1).

Niestety uzyskano słabe wyniki:



Rysunek 5: Przykładowy wynik



Rysunek 6: Ewolucja CS

4 AE3

W ramach ćwiczenia AE3 zaimplementowano algorytm genetyczny do uczenia wielowarstwowego perceptrona (MLP). Zastąpiono standardowy algorytm wstecznej propagacji błędu ewolucyjną optymalizacją wag i biasów sieci neuronowej.

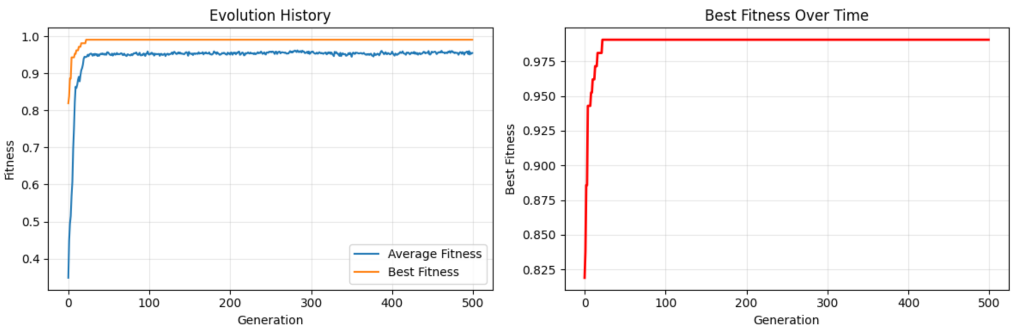
Zaimplementowano standardowe operatory genetyczne: selekcję turniejową, krzyżowanie jednopunktowe oraz mutację gaussowską z mechanizmem elitaryzmu. Funkcja fitness została zdefiniowana jako dokładność klasyfikacji dla zadań klasyfikacyjnych oraz ujemny błąd średniokwadratowy dla regresji.

Przeprowadzono eksperymenty na trzech zbiorach danych:

- Iris – klasyfikacja wieloklasowa (dokładność na zbiorze testowym)
- Multimodal-large – regresja funkcji wielomodalnej (błąd MSE)
- Auto-MPG – predykcja zużycia paliwa (błąd MSE w skali oryginalnej)

Algorytm genetyczny wykazał zdolność do skutecznej optymalizacji parametrów sieci.

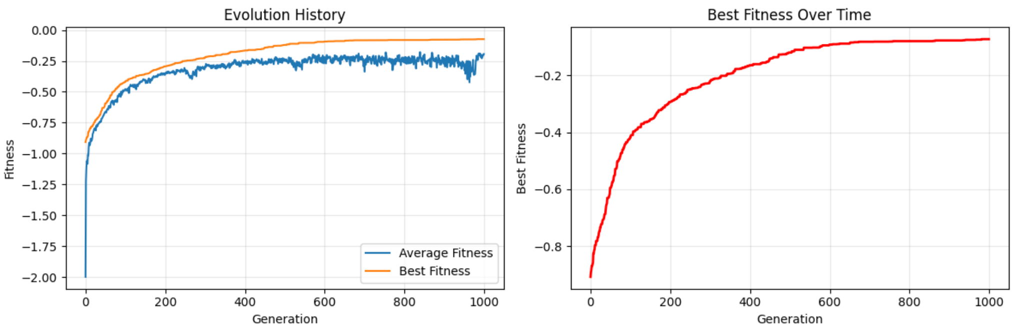
4.1 Iris



Rysunek 7: Przebieg ewolucji oraz wyniki klasyfikacji na zbiorze Iris.

Na zbiorze `Iris` algorytm zakończył ewolucję po 500 pokoleniach z najwyższą średnią przystosowania avg_fitness = 0,9546 oraz najlepszym przystosowaniem best_fitness = 0,9905. Osiągnięto najlepszą dokładność na zbiorze treningowym równą 99,05%, natomiast na zbiorze testowym uzyskano perfekcyjną dokładność 100,00%.

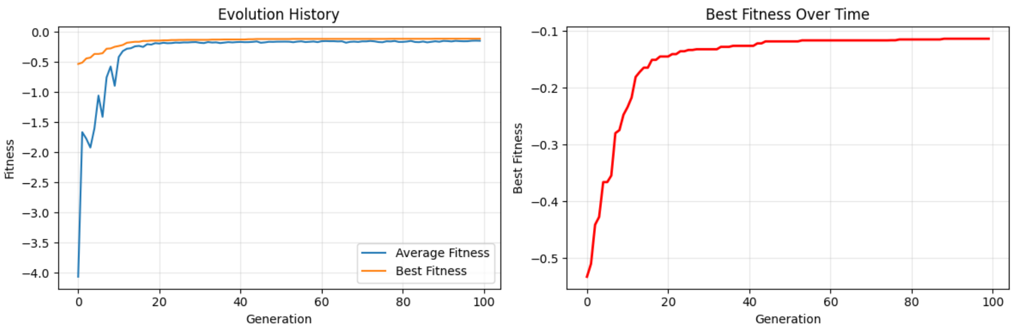
4.2 Multimodal



Rysunek 8: Przebieg ewolucji oraz wyniki regresji na zbiorze Multimodal.

W eksperymencie na zbiorze `Multimodal` ewolucja trwała 1000 pokoleń, osiągając średnie przystosowanie avg_fitness = $-0{,}1938$ oraz najlepsze przystosowanie best_fitness = $-0{,}0732$. Ostateczny błąd średniokwadratowy na zbiorze testowym wyniósł MSE = 872,1054.

4.3 MPG



Rysunek 9: Przebieg ewolucji oraz wyniki regresji na zbiorze AutoMPG.

Dla zbioru `AutoMPG` algorytm ewoluował przez 100 pokoleń, osiągając średnie przystosowanie avg_fitness $= -0{,}1472$ i najlepsze best_fitness $= -0{,}1135$. Na zbiorze testowym błąd średniokwadratowy wyniósł 0,1273 w skali znormalizowanej, co odpowiada wartości 7,7359 w skali oryginalnej.

Dla wszystkich trzech zbiorów danych uzyskane wyniki można uznać za bardzo dobre, co świadczy o skuteczności zastosowanego podejścia ewolucyjnego.

5 Wnioski

Przeprowadzone eksperymenty potwierdziły, że algorytmy genetyczne stanowią elastyczne i uniwersalne narzędzie optymalizacyjne, zdolne do adaptacji w bardzo różnych dziedzinach – od klasycznych benchmarków matematycznych, poprzez trudne problemy rozmieszczania figur, aż po uczenie parametrów sieci neuronowych. Na podstawie wyników z części 2–4 można sformułować następujące kluczowe wnioski:

1. **Prosty AG z mutacją gaussowską i krzyżowaniem jednopunktowym (AE1)** osiąga satysfakcjonujące wyniki na gładkich funkcjach testowych przy niewielkich populacjach, co potwierdza, że nawet podstawowe warianty ewolucyjne potrafią skutecznie odnajdywać minimum globalne.

2. **Złożone, dyskretno-ciągłe przestrzenie rozwiązań (AE2)** wymagają bogatszych operatorów (np. krzyżowania geometrycznego, mutacji heurystycznej) i starannego doboru parametrów. Uzyskane słabsze rezultaty wskazują, że sam dobór reprezentacji i konwencjonalne operatory mogą być niewystarczające bez dodatkowego wsparcia heurystycznego.

3. **Ewolucyjna optymalizacja wag MLP (AE3)** pokazała, że AG mogą być realną alternatywą dla gradientowych metod uczenia – szczególnie w sytuacjach, gdy pochodne są trudne do obliczenia lub funkcja celu jest niestandardowa. Perfekcyjna dokładność na zbiorze Iris i konkurencyjne MSE w regresji dowodzą praktycznej wartości podejścia.

4. **Parametry algorytmu silnie wpływają na stabilność i tempo konwergencji.** Eksperymenty własne z prawdopodobieństwem mutacji oraz rozmiarem populacji (rys. 3, 4) jednoznacznie wskazują, że zbyt małe populacje prowadzą do przedwczesnej zbieżności, natomiast nadmierne wartości mutacji wywołują oscylacje i spadek stabilności wyników.

Rekomendacje na przyszłość

- **Hybrydyzacja z lokalnym przeszukiwaniem**: połączenie AG z metodami gradientowymi (np. Adam) lub heurystykami lokalnymi (hill-climbing, 2-opt) może znacząco przyspieszyć konwergencję.

- **Adaptacyjne operatory i samostrojenie**: techniki takie jak dynamiczna regulacja zakresu mutacji lub auto-adjustment wielkości populacji mogą ograniczyć potrzebę ręcznej kalibracji parametrów.

- **Paralelizacja i implementacje GPU**: z uwagi na niezależność osobników, AG szczególnie dobrze skalują się w środowiskach równoległych, co jest istotne przy dużych populacjach lub kosztownych funkcjach celu.

- **Badania nad wielokryterialnością**: rozszerzenie algorytmu o rankingi Pareto (NSGA-II, SPEA2) pozwoliłoby jednocześnie optymalizować, np. dokładność i złożoność modelu.

Podsumowując, algorytmy genetyczne – po odpowiednim dostrojeniu i ewentualnym wzbogaceniu o dodatkowe heurystyki – pozostają wartościowym podejściem wszędzie tam, gdzie klasyczne metody zawodzą lub wymagają kosztownych obliczeń gradientu. Zaproponowane kierunki dalszych badań mogą uczynić AG jeszcze bardziej konkurencyjnymi w praktycznych zastosowaniach.

6 Podsumowanie

W pracy zaprezentowano trzy warianty algorytmu genetycznego, różniące się sposobem reprezentacji rozwiązań i zastosowanymi operatorami ewolucyjnymi, oraz oceniono ich skuteczność na zadaniach testowych i praktycznych:

- W części 2 zaimplementowano podstawowy AG z mutacją gaussowską i krzyżowaniem jednopunktowym. Testy na funkcji kwadratowej $f(x, y, z) = x^2 + y^2 + 2z^2$ oraz pięciowymiarowej funkcji Rastrigina wykazały, że algorytm szybko i stabilnie znajduje wartości bliskie minimum globalnemu, przy relatywnie niewielkich rozmiarach populacji.
- W części 3 algorytm zastosowano do wariantu problemu *cutting stock* w otoczeniu kołowym. Pomimo poprawnej realizacji kodowania genotypu i operatorów genetycznych, uzyskane wyniki okazały się mało satysfakcjonujące – prawdopodobnie ze względu na trudność wielowymiarowej, dyskretno-ciągłej przestrzeni rozwiązań.
- W części 4 zaproponowano ewolucyjną optymalizację wag i biasów MLP. Na zbiorach Iris, Multimodal-large oraz Auto-MPG algorytm osiągnął doskonałe wyniki klasyfikacji (do 100% dokładności) i konkurencyjne wartości błędu MSE, potwierdzając przydatność podejścia ewolucyjnego w treningu sieci neuronowych.

Podsumowując, praca dowiodła:

1. Uniwersalności i elastyczności algorytmów genetycznych w optymalizacji zarówno funkcji ciągłych, jak i problemów rzeczywistych,
2. Wysokiej efektywności prostego AG na problemach o gładkiej powierzchni celu,
3. Konieczności dalszej analizy i dostrojenia parametrów (np. strategii mutacji, doboru populacji) w trudniejszych zadaniach mieszanych.

Jako kierunki dalszych badań proponuje się:

- Badanie hybrydowych podejść łączących AG z lokalnymi metodami optymalizacji,
- Eksperymenty na większych i bardziej złożonych zbiorach rzeczywistych danych,
- Automatyczne dostrajanie parametrów algorytmu (metaoptymalizacja).

Końcowe wnioski potwierdzają, że algorytmy genetyczne są wartościowym narzędziem w obszarze optymalizacji, zwłaszcza tam, gdzie przestrzeń rozwiązań jest wielowymiarowa i ma charakter nieliniowy.