Algorytmy ewolucyjne

Piotr Bielecki

June 2023

Streszczenie

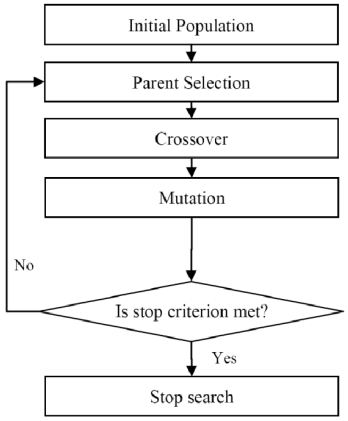
Niniejszy dokument zawiera omówienie wyników pracy z przedmiotu Metody inteligencji obliczeniowej w analizie danych na temat algorytmów ewolucyjnych.

Spis treści

1 wstęp teoretyczny

Algorytmy ewolucyjne to rodzaj metaheurystyki inspirowanej procesem ewolucji biologicznej. Są one stosowane do rozwiązywania problemów optymalizacyjnych, które są trudne do rozwiązania za pomocą tradycyjnych metod.

Algorytmy ewolucyjne działają na podobnej zasadzie jak proces ewolucji biologicznej. Algorytm rozpoczyna się od populacji początkowej, która składa się z pewnej liczby osobników (potencjalnych rozwiązań). Każdy osobnik jest reprezentowany jako zestaw cech, które stanowią potencjalne rozwiązanie problemu.



Rysunek 1: ogólna procedura algorytmu ewolucyjnego

Proces ewolucyjny składa się z kilku kroków:

- Selekcja: Wybierane są osobniki, które mają większe szanse na przetrwanie i rozmnażanie. Wybór jest oparty na miarach oceny, które określają jak dobre jest dane rozwiązanie w kontekście rozwiązywanego problemu.
- Krzyżowanie (crossover): Wybrane osobniki są łączone w celu stworzenia potomstwa. Jest to proces, w którym części cech (genów) od rodziców są łączone, aby stworzyć nowe rozwiązania.
- Mutacja: W pewnym stopniu, losowe zmiany są wprowadzane do potomstwa poprzez mutację genetyczną. Mutacja pozwala na wprowadzenie różnorodności genetycznej w populacji, co może prowadzić do odkrywania nowych, lepszych rozwiązań.

Proces selekcji, krzyżowania i mutacji jest powtarzany przez określoną liczbę generacji lub do momentu spełnienia pewnego warunku zakończenia, np. osiągnięcia zadowalającego rozwiązania.
W każdej implementacji algorytmu zastosowałem mechanizm elityzmu, polega

na tym, że z populacji rodziców grupa najlepszych osobników przechodzi do następnego pokolenia od razu, daje to algorytmowi możliwość zachowywania najlepszego rozwiązania.

2 podstawowy algorytm ewolucyjny

2.1 opis zadania

Zadanie polegało na zaprojektowaniu podstawowego algorytmu ewolucyjnego z mutacją gaussowską i krzyżowaniem jednopunktowym, do optymalizacji dane dwie różne funkcje.

2.2 sposób implementacji

Cechami osobników są w tym przypadku wartości kolejnych zmiennych, krzyżowanie polega na wybraniu części zmiennych z jednego rodzica, i dopełnieniu go wartościami brakującymi z drugiego rodzica, natomiast mutacja polega na dodaniu do każdej zmiennej pewnej losowej wartości z rozkładu normalnego.

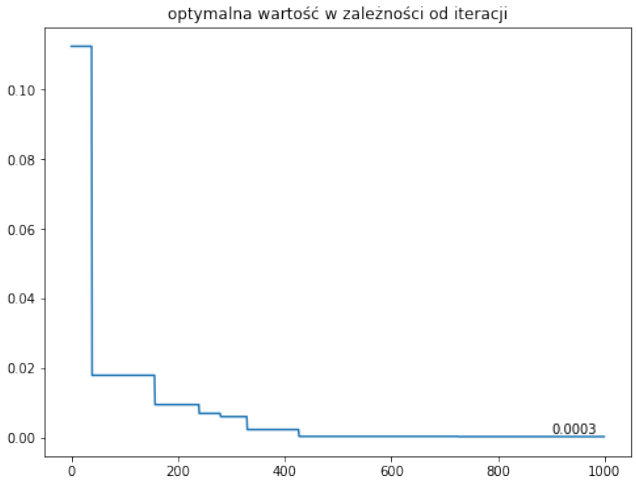
2.3 wyniki

2.3.1 trójwymiarowa funkcja kwadratowa

Jako pierwszą, przetestowałem funkcję kwadratową zadaną wzorem:

f(x) = $x^2 + y^2 + 2z^2$

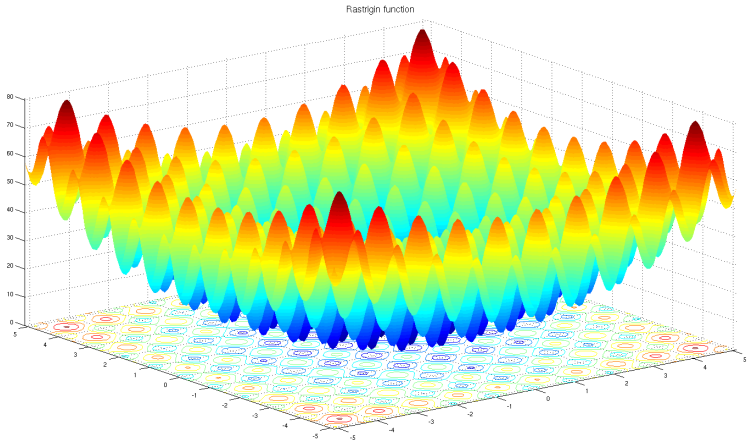
algorytm znajduje wartości coraz to bliższe minimum, co obrazuje poniższy wykres:



Rysunek 2: trójwymiarowa f. kwadratowa

2.3.2 pięciowymiarowa funkcja Rastrigina

Funkcja Rastrigina jest dobra do testowania algorytmów minimalizujących. Poniżej, wykres dwuwymiarowej funkcji Rastrigina w otoczeniu punktu (0,0) obrazujący ten fakt - wiele minimów lokalnych utrudnia znalezienie tego globalnego.

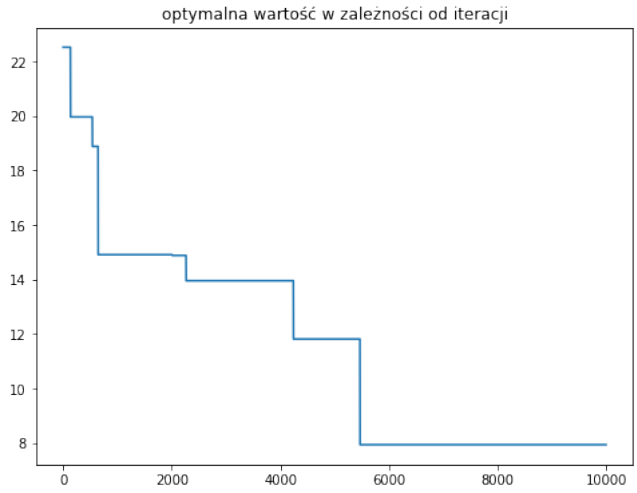


Rysunek 3: dwuwymiarowa funkcja rastrigina

ogólny wzór funkcji Rastrigina wygląda następująco:

$$\mathrm{f(x)} = \mathrm{An}+\Sigma_{i=1}^{n}[x_i^2 - Acos(2\pi x_i)]$$

Również i w tym przypadku, algorytm wykazuje zdolność do poprawiania najlepszego wyniku.



Rysunek 4: pięciowymiarowa funkcja Rastrigina

3 cutting stock problem

3.1 opis zadania

Mamy dane koło o promieniu r oraz zbiór dostępnych prostokątów zadanych przez trzy liczby: wysokość, szerokość i wartość.

Celem jest ułożenie prostokątów w kole tak, aby zmaksymalizować sumę ich wartości, spełniając następujące warunki:

- boki wszystkich prostokątów były równoległe do osi układu
- wnętrza prostokątów nie miały części wspólnej (intuicyjnie: prostokąty nie nachodzą na siebie, ale mogą się stykać bokami)
- każdy prostokąt można wstawić dowolnie wiele razy

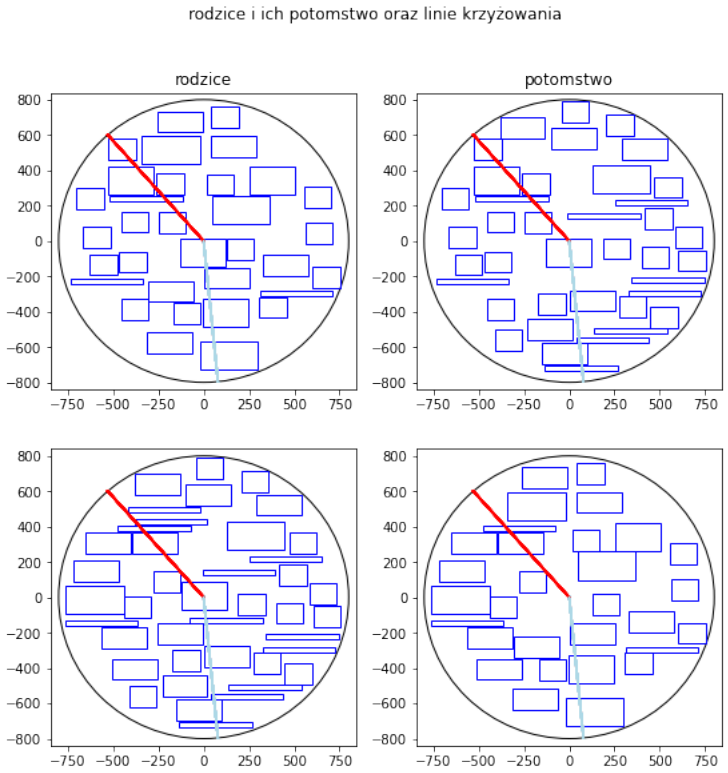
3.2 sposób implementacji

Jako że ten projekt jest znacznie większy od poprzedniego, zdecydowałem się w tej części poświęcić oddzielną sekcję na każdy element architektury algorytmu.

Za pojedynczego osobnika uznałem rozwiązanie zdefiniowane jako wypełniony okrąg, czyli listę prostokątów (ich wymiary, pozycję w okręgu, wartość)

W modelu proponowane krzyżowanie polega na wybraniu pary rodziców, a następnie pary kątów, które tną rodziców na dwa fragmenty. Powstaje dwóch potomków, z których każdy ma po jednym fragmencie od rodziców. Nadmiarowe prostokąty (takie, które nachodzą na inne prostokąty) są usuwane - tutaj w grę wchodzi mutacja, która takie "spawania"załata.

Poniżej prezentuję graficzną reprezentację krzyżowania (warto zwrócić uwagę, że progiem selekcji dla prostokątów był ich lewy dolny róg, a nie środek):



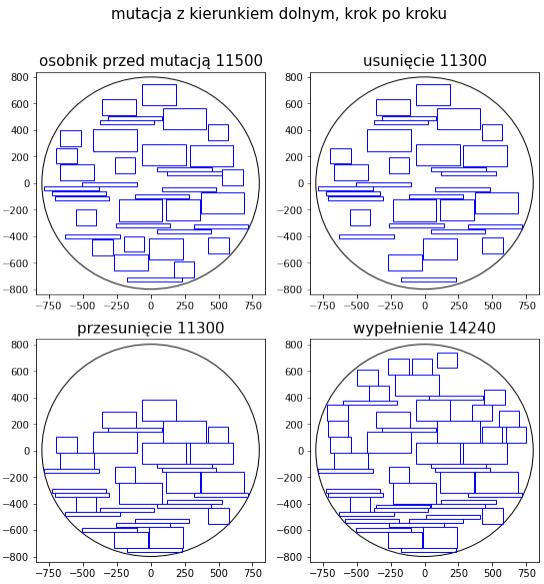
Rysunek 5: krzyżowanie - linie czerwona i niebieska reprezentują kąty podziału rodziców

Mutacja to zestaw operacji wykonanych jedna po drugiej:

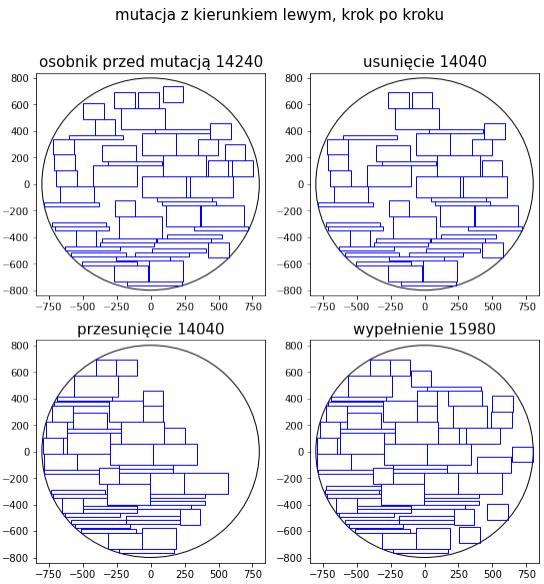
- usunięcie najmniej wartościowych prostokątów (wartość określana jest jako proporcja wagi prostokąta do jego powierzchni)
- przesunięcie wszystkich prostokątów wzdłuż wybranej osi (dół, lewo, prawo, aby zasymulować coś w rodzaju grawitacji w pudełku)
- wypełnienie wzdłuż wybranej osi okręgu prostokątami (gdy oś jest dół, prostokąty spadają z góry)

Poniżej prezentuję graficzną reprezentację mutacji, w tytułach każdego z obrazów podana wartość to suma wartości zawartych w okręgu prostokątów. Zaprezentowane są dwie mutacje, jedna po drugiej, najpierw w dół, potem w lewo. Warto zwrócić uwagę, że zastosowanie drugiej mutacji w innym niż wcześniejszy

kierunku, dało pozytywny efekt - prostokąty są ciaśniej spakowane, wartość w okręgu wzrosła.



Rysunek 6: mutacja w dół



Rysunek 7: mutacja w lewo

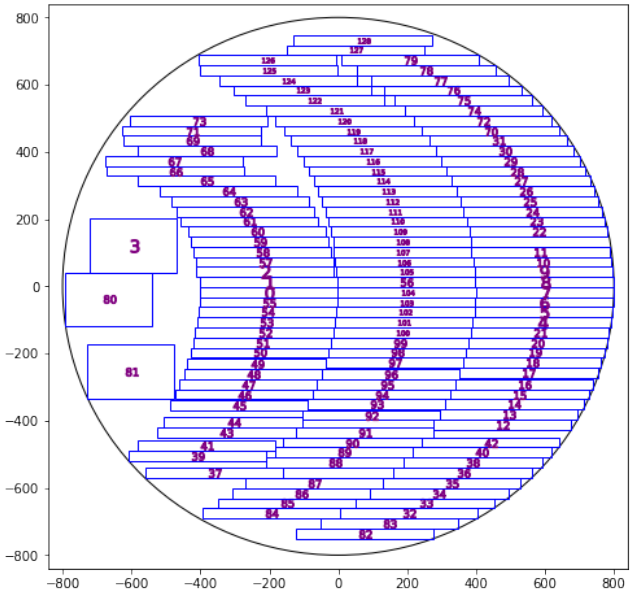
3.2.1 wyniki

Do zadanego problemu dołączone było kilka zestawów promień - prostokąty, poniżej przedstawiam opracowanie zadanych problemów.

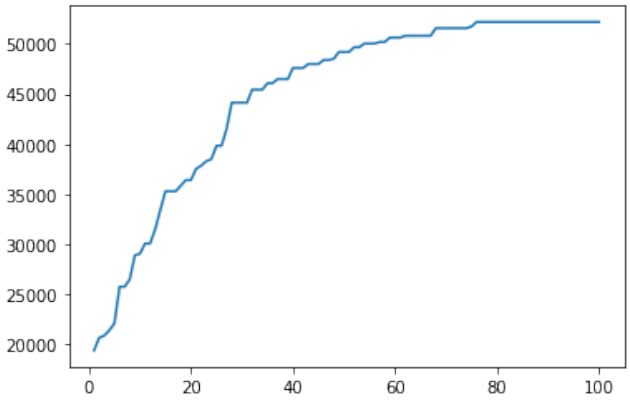
Najlepiej działanie algorytmu obrazuje wynik z pierwszego problemu, choć w każdym przypadku algorytm osiągał zdecydowanie wyższe wyniki niż to było wymagane. wizualnie, prostokąty są bardzo ciasno upakowane. Nie pokazuję tutaj kolejnych etapów dochodzenia do rozwiązania na przestrzeni procesu uczenia, ponieważ operacje które do poprawy tego rozwiązania prowadzą zostały już opisane i zwizualizowane, prezentuję natomiast ostateczne rozwiązanie problemu i wykres ilustrujący zdolność do poprawy i proces dochodzenia do ostatecznego wyniku, oraz ostateczne rozwiązanie dla każdego z zadanych problemów.

3.2.2 promień 800

ostateczna wartość wyniosła dla tego problemu 52200



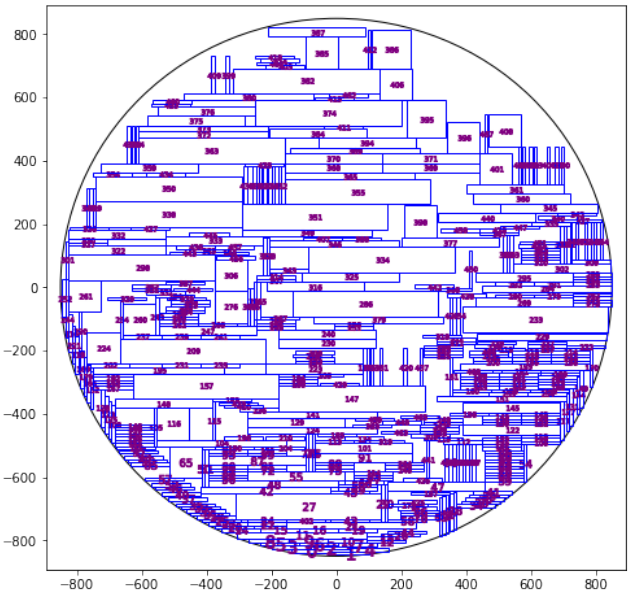
Rysunek 8: promień 800, najlepsze rozwiązanie



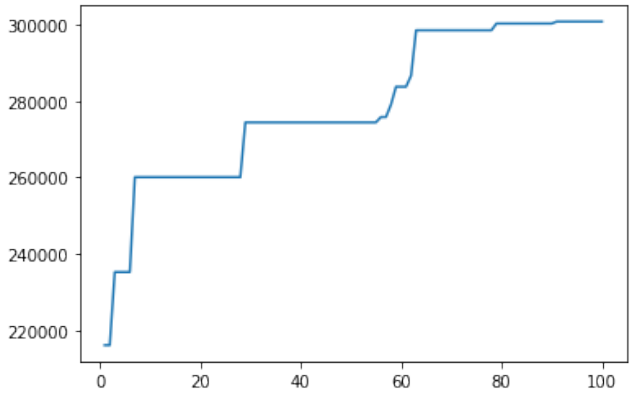
Rysunek 9: promień = 800, najlepszy wynik według iteracji

3.2.3 promień 850

ostateczna wartość wyniosła dla tego problemu 300780



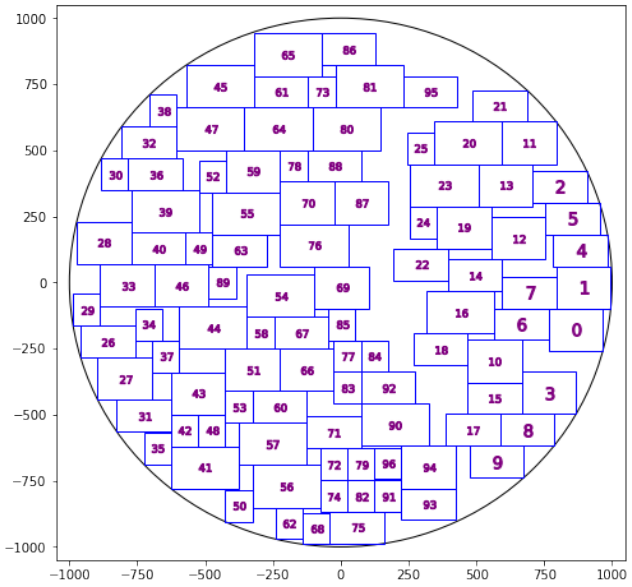
Rysunek 10: promień 850, najlepsze rozwiązanie



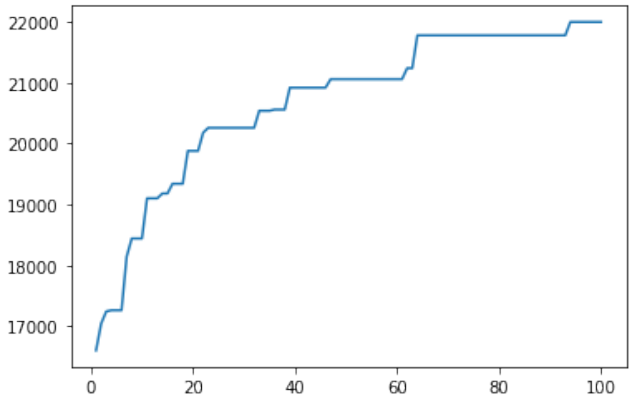
Rysunek 11: promień = 850, najlepszy wynik według iteracji

3.2.4 promień 1000

ostateczna wartość wyniosła dla tego problemu 22000



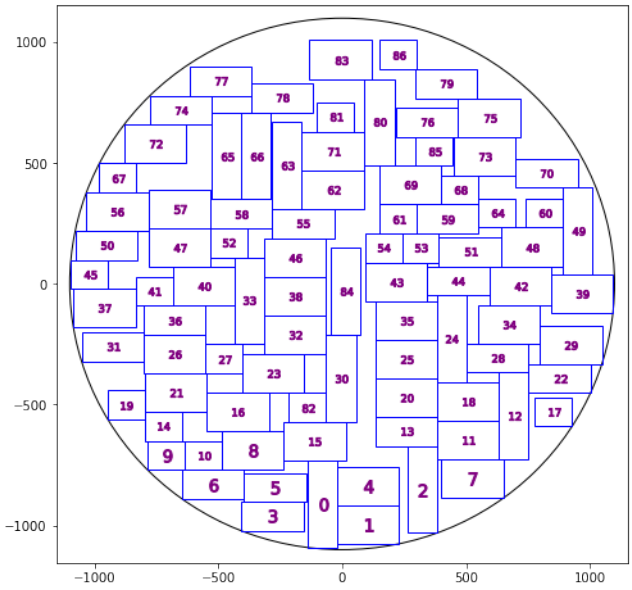
Rysunek 12: promień 1000, najlepsze rozwiązanie



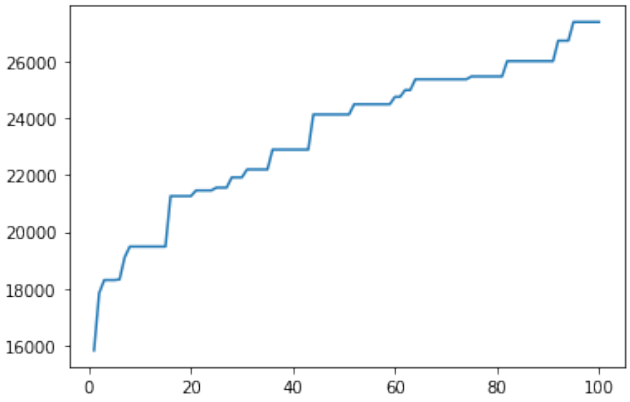
Rysunek 13: promień = 1000, najlepszy wynik według iteracji

3.2.5 promień 1100

ostateczna wartość wyniosła dla tego problemu 27400



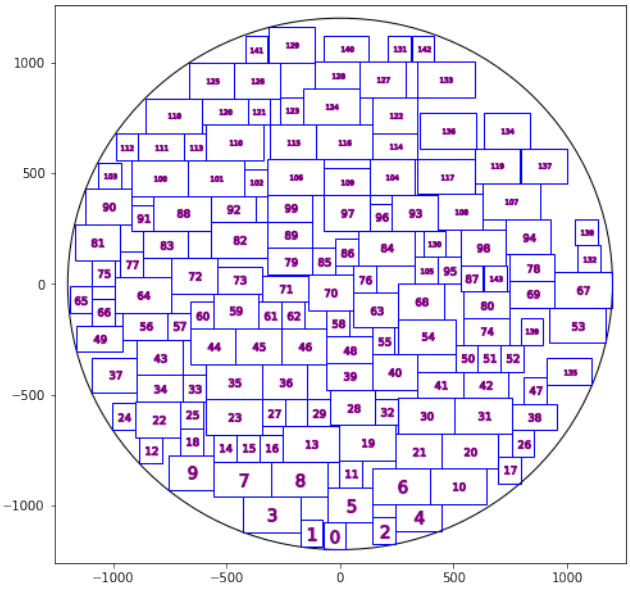
Rysunek 14: promień 1100, najlepsze rozwiązanie



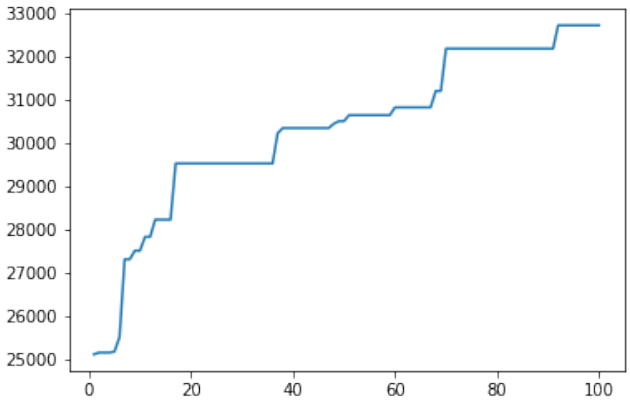
Rysunek 15: promień = 1100, najlepszy wynik według iteracji

3.2.6 promień 1200

ostateczna wartość wyniosła dla tego problemu 32720



Rysunek 16: promień 1200, najlepsze rozwiązanie



Rysunek 17: promień = 1200, najlepszy wynik według iteracji

4 optymalizacja wag sieci MLP

4.1 opis zadania

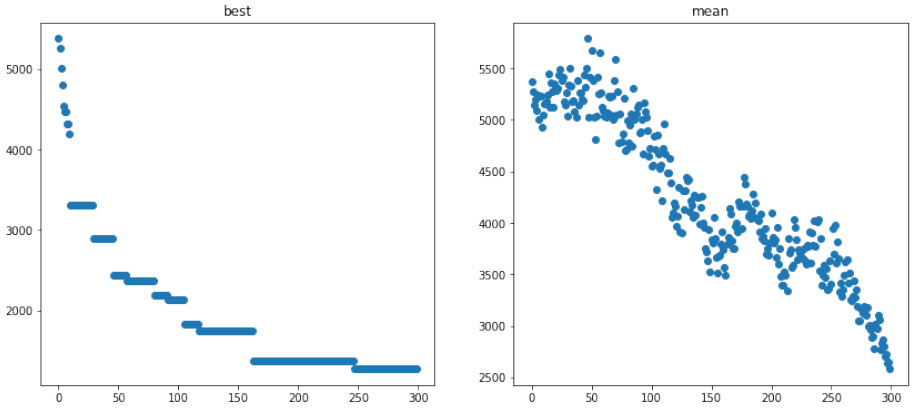
Należało zaimplementować proste uczenie MLP z użyciem algorytmu genetycznego. Na wejściu struktura sieci neuronowej i dane uczące. Optymalizowana funkcja to funkcja przekształcająca wektor wag sieci na błąd na zbiorze uczącym. Zastosować standardowe operatory krzyżowania i mutacji.
Do zadania dołączone były zbiory, na których należało przeprowadzić uczenie.

4.2 sposób implementacji

Wykorzystałem przygotowaną przeze mnie na wcześniejszym etapie zajęć z przedmiotu sieć MLP. Zadane zbiory danych do przetestowania to dane iris(problem klasyfikacji) oraz dane mpg i multimodal-large (problem regresji). W problemach klasyfikacji minimalizowaną funkcją jest błąd średniokwadratowy, natomiast w przypadku problemu klasyfikacji próbuję maksymalizować accuracy.
krzyżowanie i mutacja działają identycznie jak w przypadku podstawowej implementacji sieci, z tym, że teraz cechami nie są wartości kolejnych zmiennych, a wagi kolejnych połączeń w sieci.

4.2.1 zbiór multimodal-large

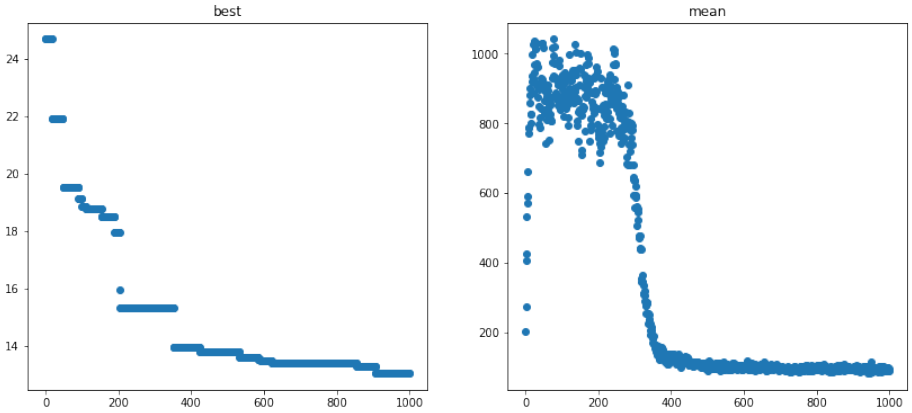
Poniżej najlepszy i średni wynik według iteracji, prezentujący zdolność algorytmu do optymalizacji i w tym przypadku.



Rysunek 18: MSE - zbiór multimodal-large

4.2.2 zbiór mpg

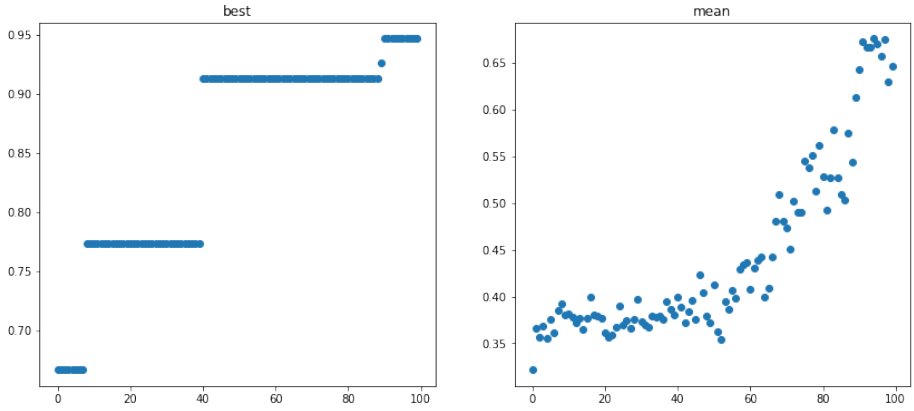
W tym zadaniu należało przewidywać wartość mpg, co również udało się zrobić, choć w tym przypadku sieć miała pewne problemy, które rozwiązałem zwiększając parametr elityzmu.



Rysunek 19: MSE - zbiór mpg

4.2.3 zbiór iris

W tym zadaniu celem była klasyfikacja trzech typów irysów, na podstawie danych takich jak wielkość i długość płatków kwiatowych. Poniżej prezentuję wyniki uczenia



Rysunek 20: MSE - zbiór mpg

5 podsumowanie

Zaproponowane algorytmy prowadzą do pewnej optymalizacji rozwiązań, choć funkcja rastrigina i zbiór mpg przysporzyły kłopotów w trakcie rozwijania algorytmów. Każdy z zadanych zbiorów został przetestowany i na każdym algorytmy wykazały zdolność do uczenia.