

# Sprawozdanie 3:

## Algorytmy Ewolucyjne

Autor:

Mateusz Stączek

Czerwiec, 2022

Nr indeksu:

305757

# Temat Laboratoriów

Własna implementacja prostego algorytmu genetycznego. Celem laboratoriów jest uzyskanie dobrych wyników przy różnych zadaniach, między innymi wypełniania koła prostokątami, a także wizualizacja efektów procesu uczenia.

## Wykonana praca i wyniki eksperymentów

Zaimplementowany został prosty algorytm genetyczny. Kodowanie genomu zmieniało się w zależności od problemu, natomiast wszędzie wykorzystana została selekcja turniejowa.

### Laboratorium 1

Cel: Minimalizacja wartości funkcji

Testowane funkcje:

- trójwymiarowego wielomianu drugiego stopnia
- pięciowymiarowej funkcji Rastrigina

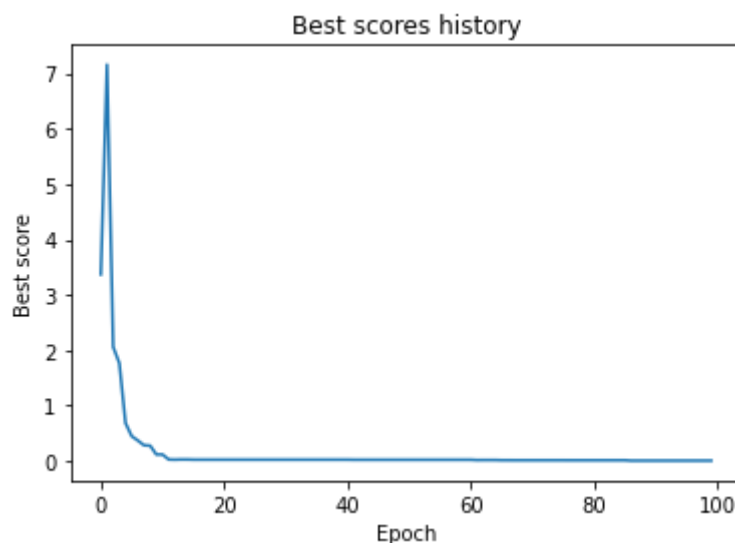
Rozwiązanie:

Genom zawiera  $n$  liczb ( $n=3$  lub  $n=5$ ) typu float, mutacja gaussowska, selekcja turniejowa oraz crossover jednopunktowy. Maksymalne wartości ograniczone do przedziału  $(-10, 10)$

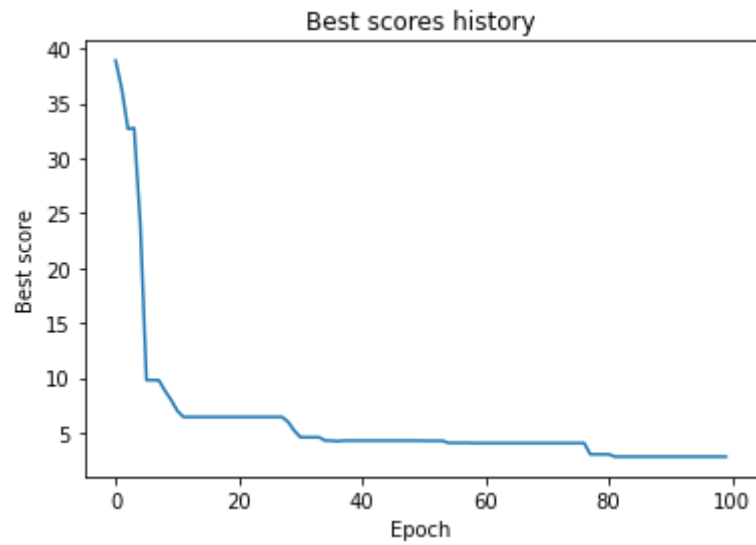
Rezultaty:

Dla obu funkcji minimum jest dla wszystkich zmiennych równych 0 i wynosi 0.

Dla pierwszej funkcji najlepszy osobnik miał wartości  $x, y$  i  $z$  jako  $[0.013 \ -0.077 \ -0.005]$  oraz osiągnął wartość minimalizowanej funkcji jako około 0.006.



Dla drugiej funkcji do minimalizacji, wyniki to  $[0.0003, -0.0204, 0.0294, -0.09, 0.9935]$ , a wartość minimalizowanej funkcji to około 2.81.



W obu przypadkach wyniki są bliskie 0, a minimalizowane funkcje przyjęły dość małe wartości. Liczby epok wynosiły 100, a populacje w pierwszej 100, a w drugiej funkcji 500.

## Laboratorium 2

Wykorzystane dane to listy rozmiarów i wartości prostokątów, które należało w najlepszy możliwy sposób ułożyć z powtórzeniami w kołach o promieniu, który znajdował się w nazwie pliku csv danego zbioru prostokątów. Wartości rozwiązań to suma wartości wewnątrz prostokątów.

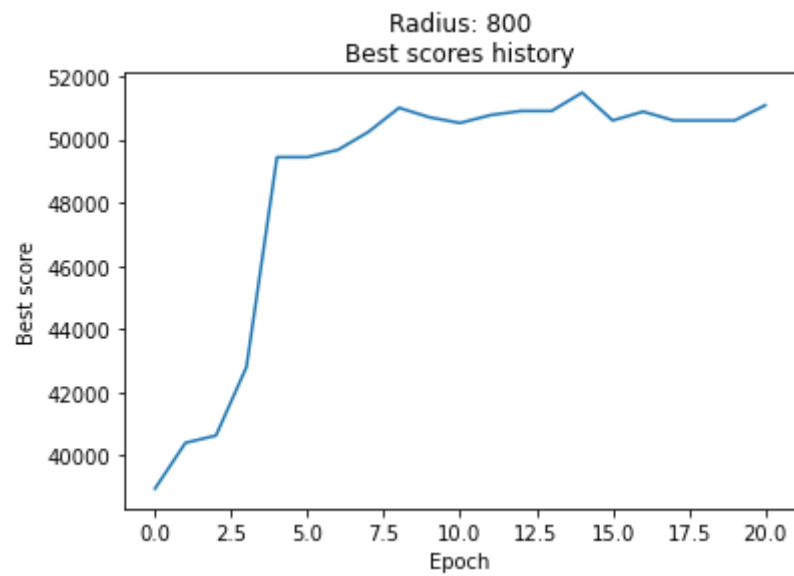
### Sposób rozwiązania

Koło będzie zawierać kolumny, z których każda będzie zawierać ten sam prostokąt tyle razy, ile tylko się zmieści. Kodowanie genomu to lista ID prostokątów, o zmiennej długości, bo liczba kolumn się może zmieniać. Funkcja fitness będzie sama znajdować liczbę prostokątów, które w każdej kolumnie się mieszczą, przy założeniu, że w pierwszej kolumnie znajduje się tylko 1 prostokąt.

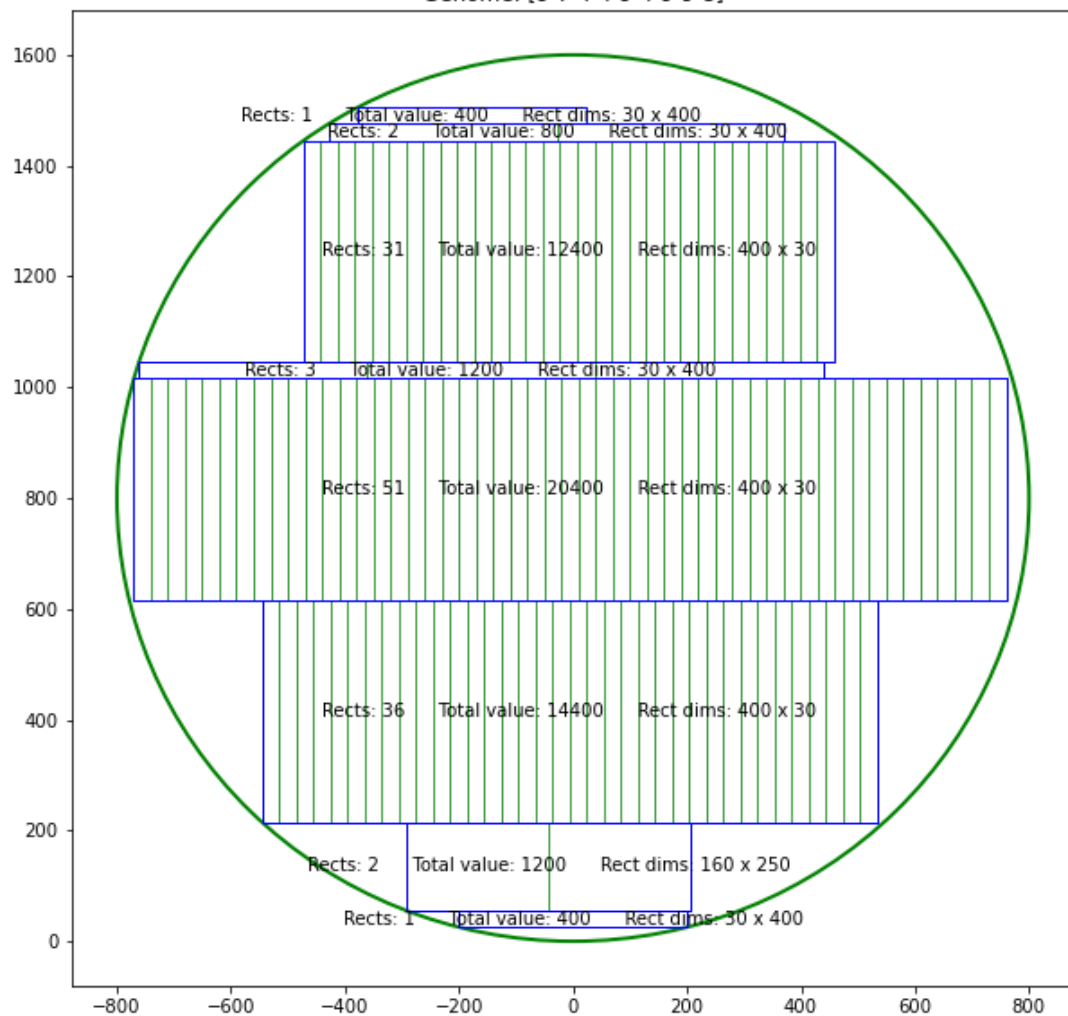
Mutacje to dodanie kolumny, usunięcie kolumny, zmiana prostokąta, który będzie w danej kolumnie. Crossover jednopunktowy, selekcja turniejowa. Gdy w danej kolumnie nie może być żaden prostokąt, od wyniku odejmowana jest suma wartości prostokątów, z których wystające kolumny miały się składać.

Dla każdego zbioru, algorytm genetyczny był uruchamiany przy populacji o liczności 100, przy 20 epokach

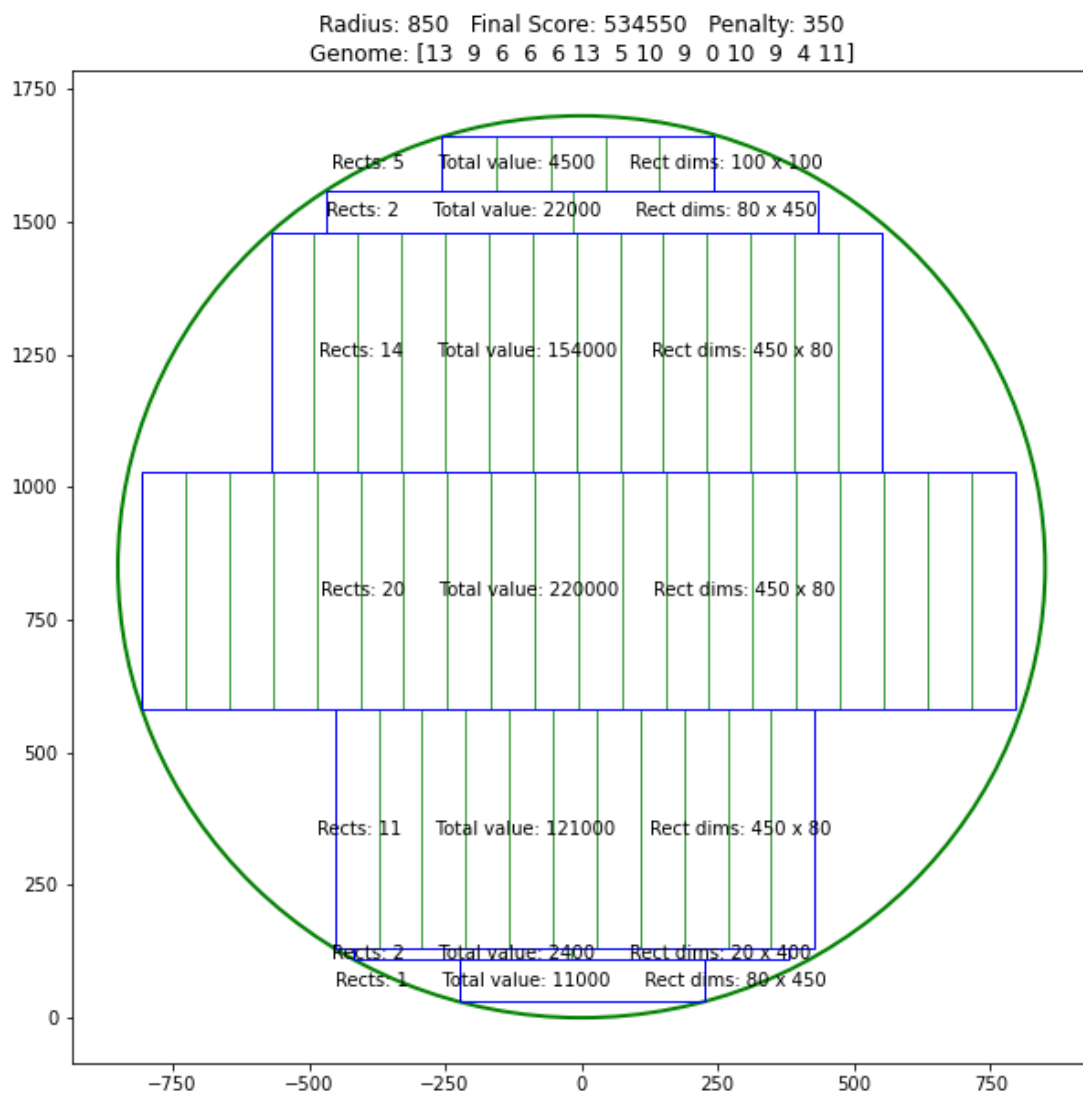
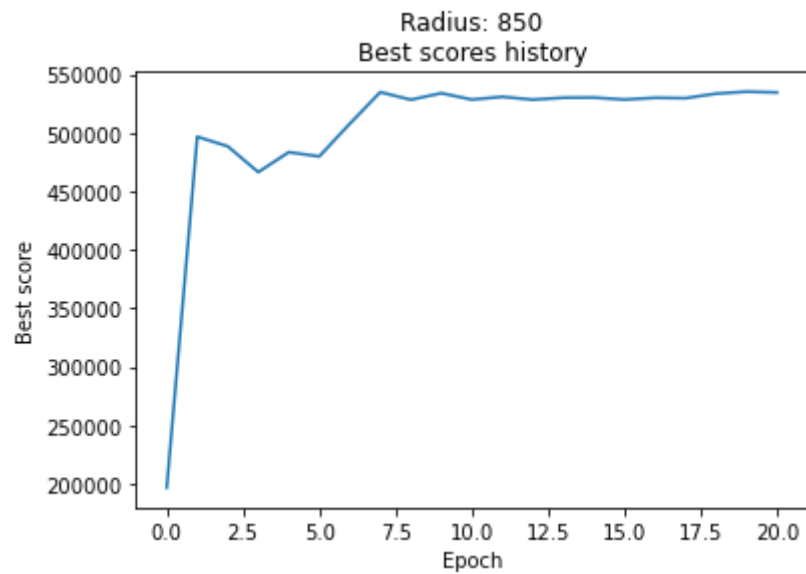
## Zbiór 1 - R800



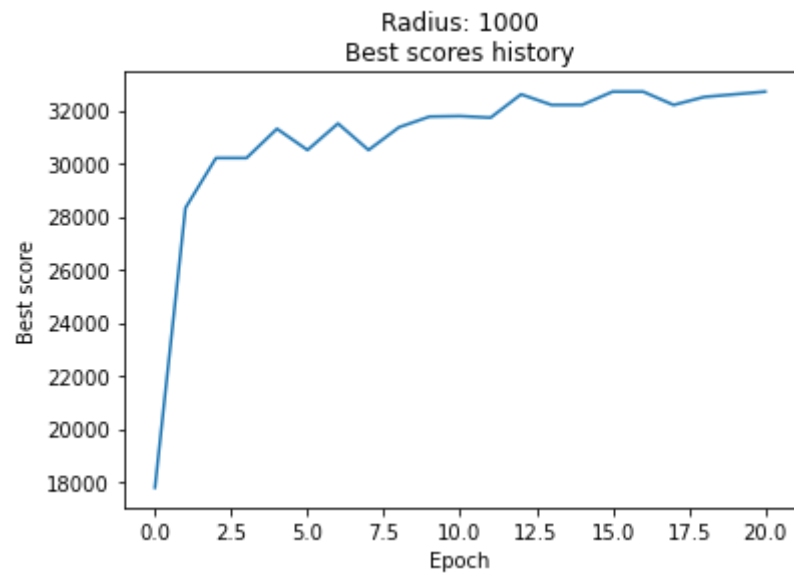
Radius: 800 Final Score: 51080 Penalty: 120  
Genome: [9 7 4 4 9 4 9 9 8]



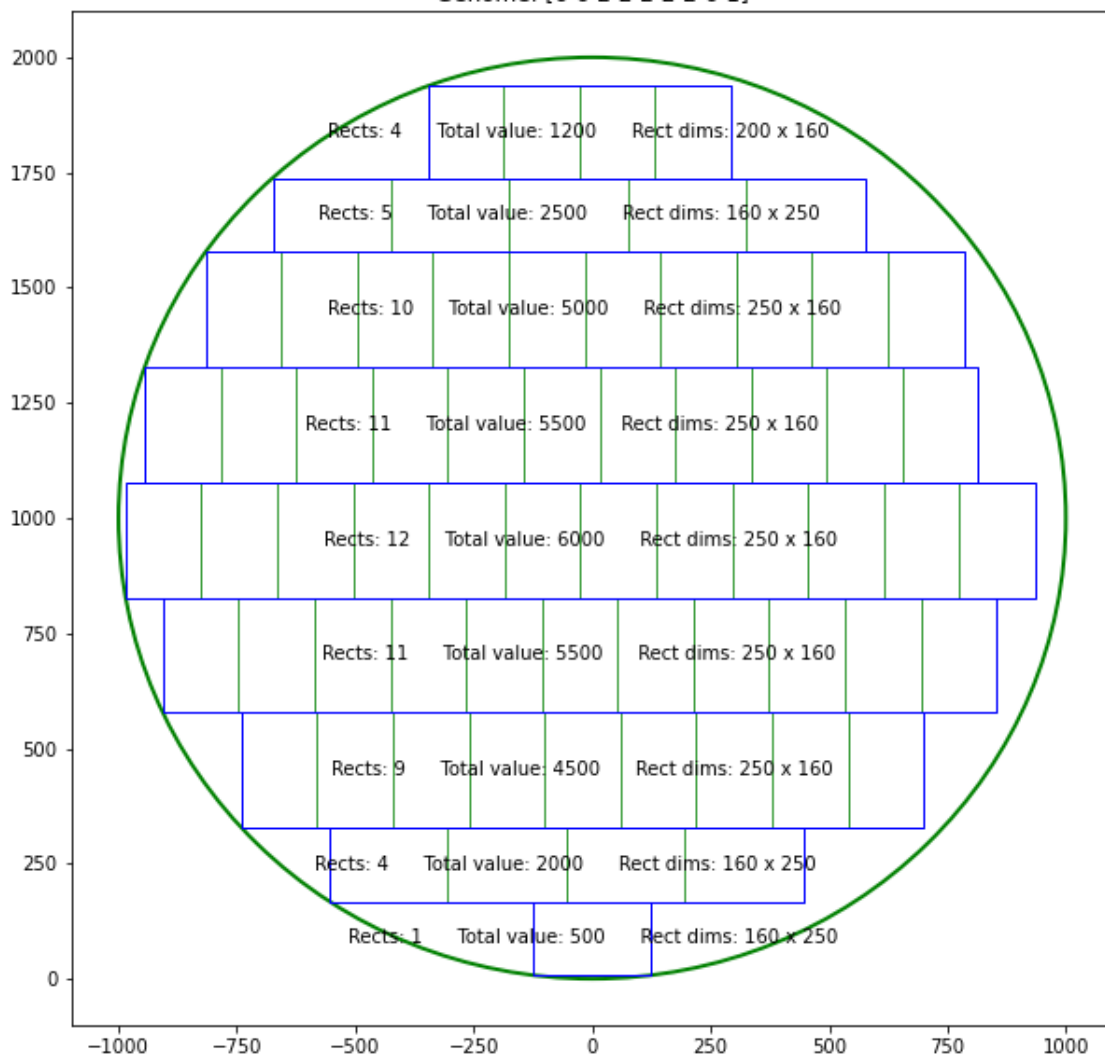
## Zbiór 2 - R850



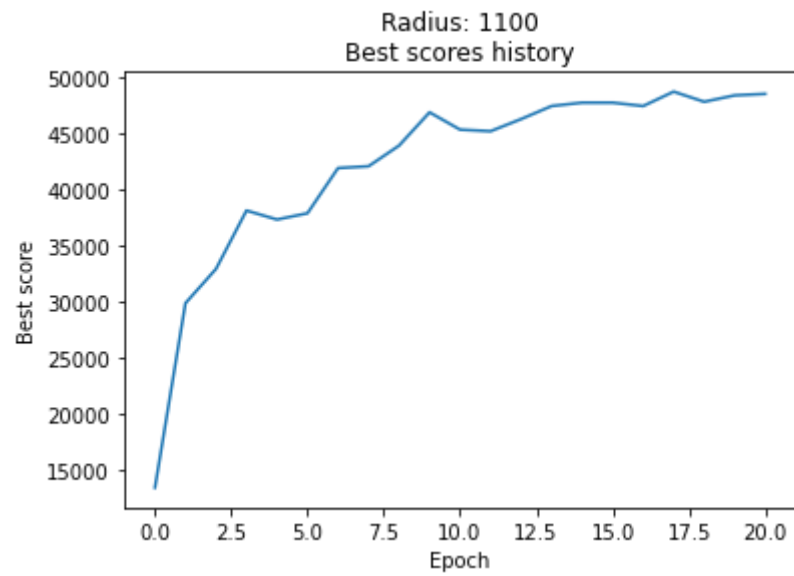
## Zbiór 3 - R1000



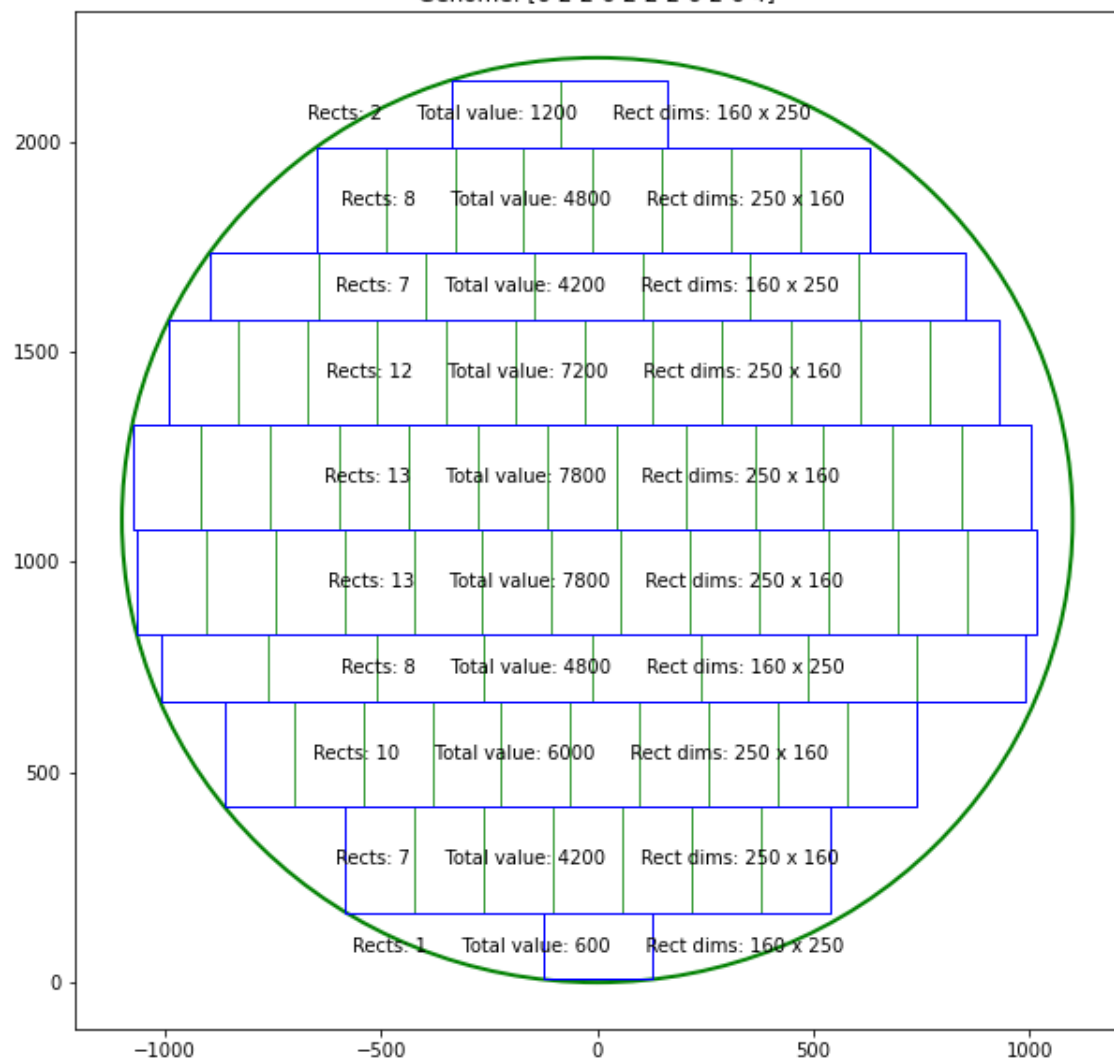
Radius: 1000 Final Score: 32700 Penalty: 0  
Genome: [6 6 2 2 2 2 2 6 1]



## Zbiór 4 - R1100

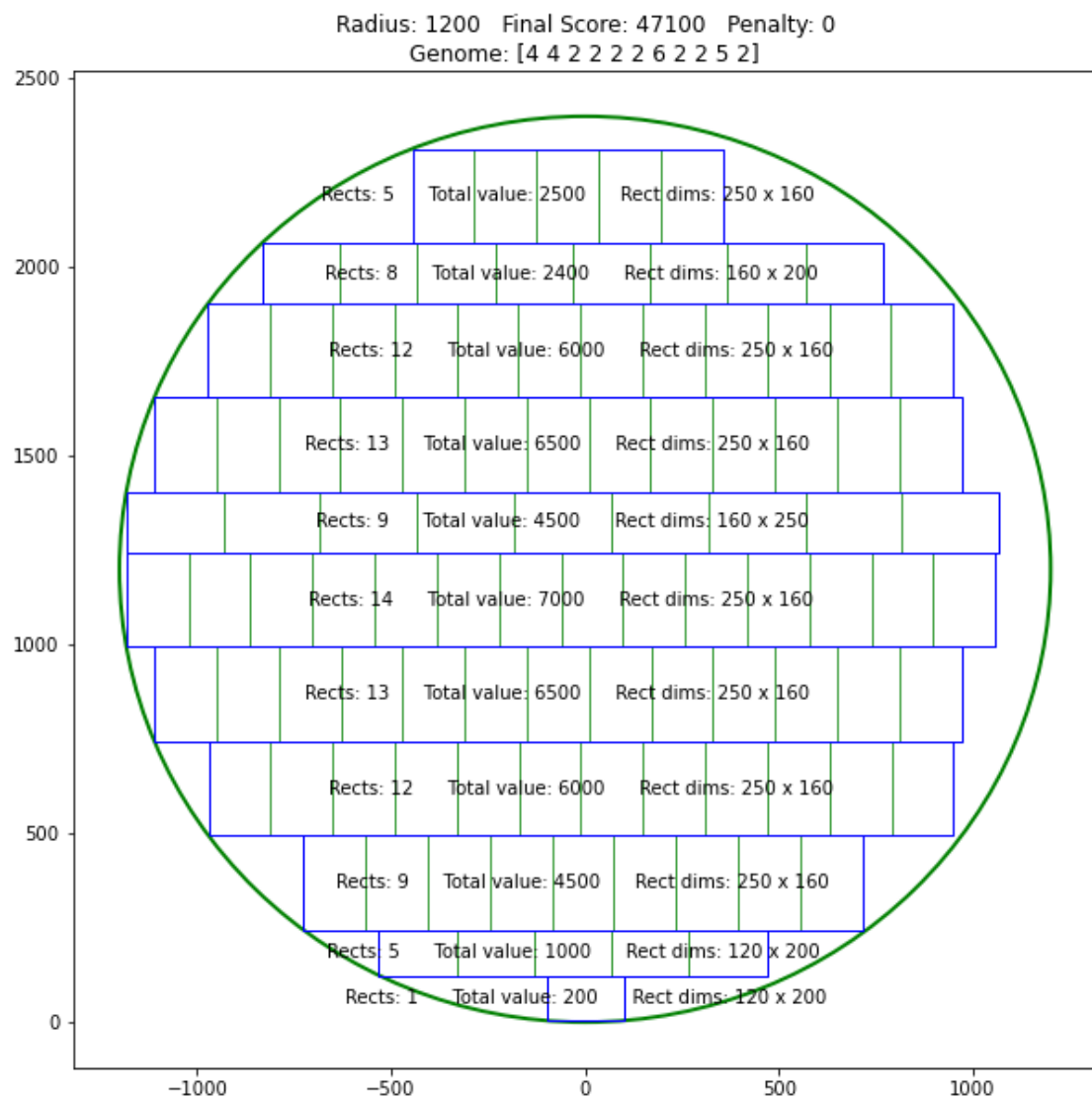
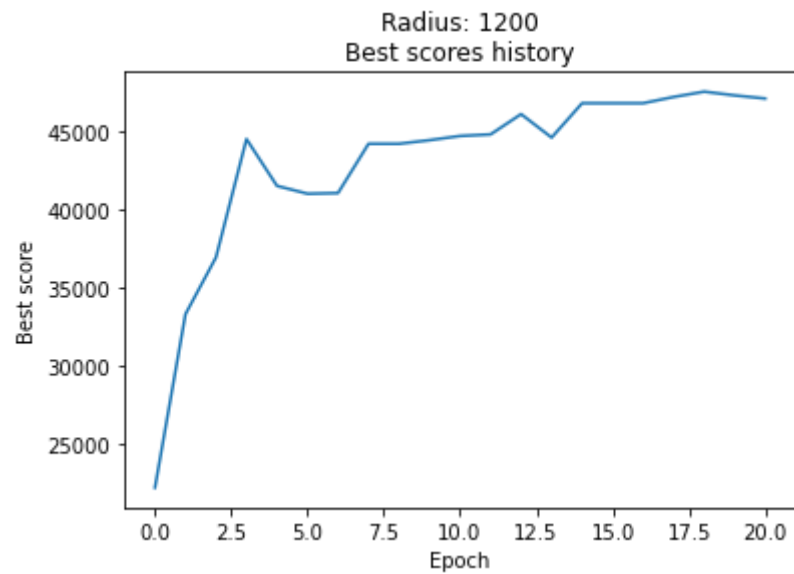


Radius: 1100 Final Score: 48480 Penalty: 120  
Genome: [6 2 2 6 2 2 2 6 2 6 4]





## Zbiór 5 - R1200



## Rezultaty lab 2

Otrzymane wyniki dla poszczególnych promieni okręgów spełniają wymagania z treści. Powyżej, każdy zbiór ma przedstawione właśnie najlepsze znalezione rozwiązanie.

Wyniki (zaokrąglone do tysięcy):

Promień	800	850	1000	1100	1200
Wynik	51000	535000	33000	48000	47000

## Laboratorium 3

Wykorzystane dane to iris, multimodal large oraz auto mpg. Algorytm ewolucyjny został wykorzystany do znalezienia optymalnych wag sieci neuronowych dla każdego z problemów.

### Sposób rozwiązania

Algorytm na wejściu otrzymuje informację o liczbie neuronów w każdej warstwie i funkcjach aktywacji. Ponadto otrzymuje dane uczące, czyli wszystkie dane dostępne dla danego problemu. Na tych danych będzie wykonywane liczenie miary błędu w trakcie i po skończeniu

Genom:

- lista macierzy wag,
- Każda macierz odpowiada za wagi pomiędzy kolejnymi warstwami
- Jest dodany bias.

Mutacja:

- Losowana jest warstwa i waga z tej warstwy i dodawana jest liczba z rozkładu jednostajnego na  $-0.5, 0.5$ ,
- Takie dodawanie jest wykonywane z prawdopodobieństwem 0.2,
- Powyższe 2 kroki są powtarzane tyle razy, ile jest wszystkich wag w sieci.

Crossover:

- Dla każdego osobnika w populacji losowany jest osobnik.
- Następnie zachodzi procedura jak w mutacji, tylko z prawdopodobieństwem 0.4 oraz zamiast dodawania losowej liczby, to w kopii pierwszego rodzica dana waga jest podmieniana wagą drugiego rodzica, po czym nowy osobnik jest zwracany.

Fitness:

- MSE, dla klasyfikacji oraz dla regresji.

Selekcja:

- Turniejowa, o rozmiarze 3.

## Parametry wywołań

W poniższych rezultatach algorytmy ewolucyjne zostały wywołane z następującymi parametrami:

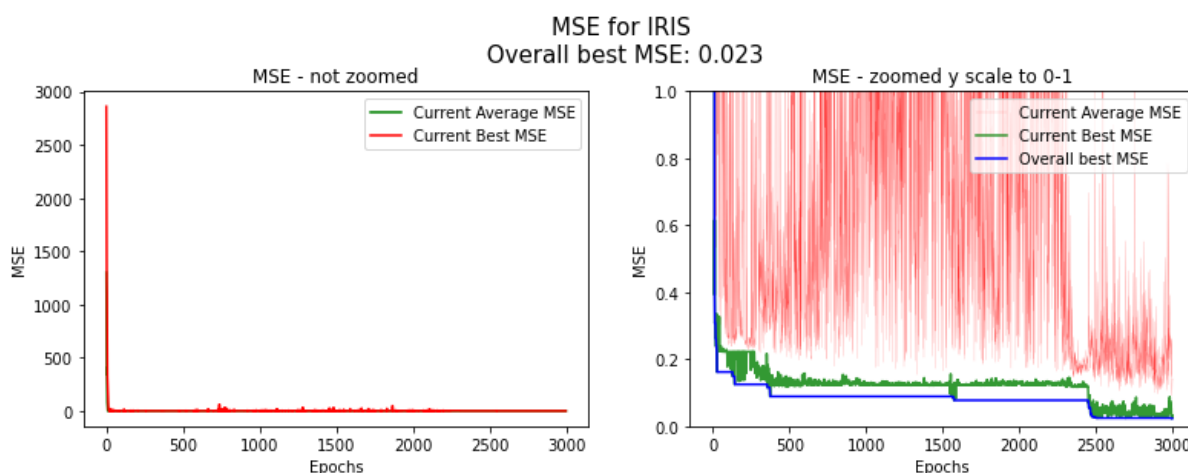
Dane	iris	multimodal	Auto mpg
Rozmiar populacji	30	20	50
Liczba epok	3000	3000	1500
Rozmiar sieci	[4,6,5,3]	[1,10,7,1]	[7,6,5,1]
Funkcje aktywacji	ReLu + ReLu output	Tanh + linear output	Tanh + linear output
Czas wywołania	80 s	110 s	70 s

## Rezultaty

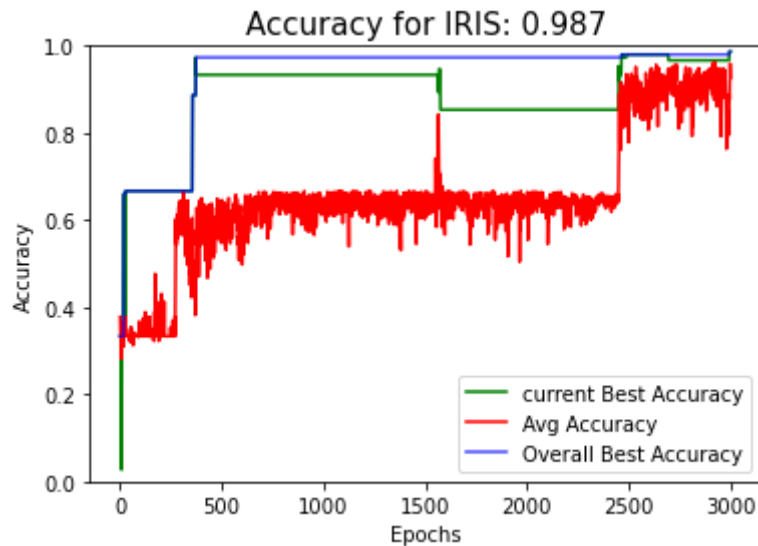
Wykresy uczenia zawierają na wykresie MSE średnie dla obecnej epoki (czerwone), najlepsze dla obecnej epoki (zielone) oraz najlepsze z dotychczasowych epok (niebieskie). W przypadku, jeśli pełny wykres MSE od epoki jest nieczytelny, dodawany jest drugi wykres ze zmienioną skalą na osi y.

### Iris

W przypadku pierwszego zbioru, zazwyczaj znajdowane było rozwiązanie na poziomie 0.95 accuracy, aczkolwiek czasami algorytm zatrzymywał się w 0.67 i nie był w stanie poruszyć się dalej.



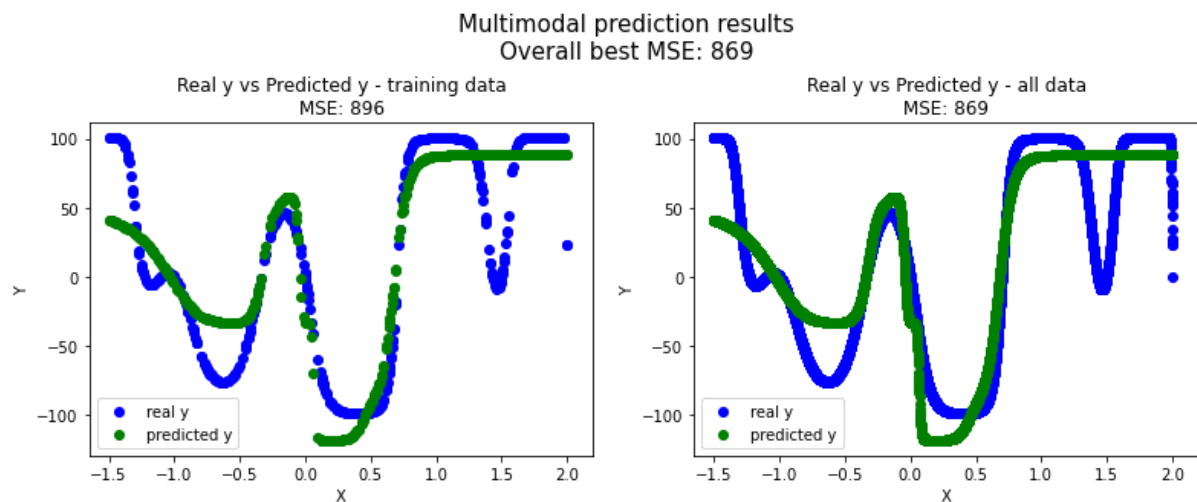
Aby rozwiązać wątpliwości odnośnie poniższego wykresu, w algorytmie genetycznym selekcja następowała po MSE, więc czasami zdarzało się, że spadek MSE obniżał accuracy, jak około epoki 1500.



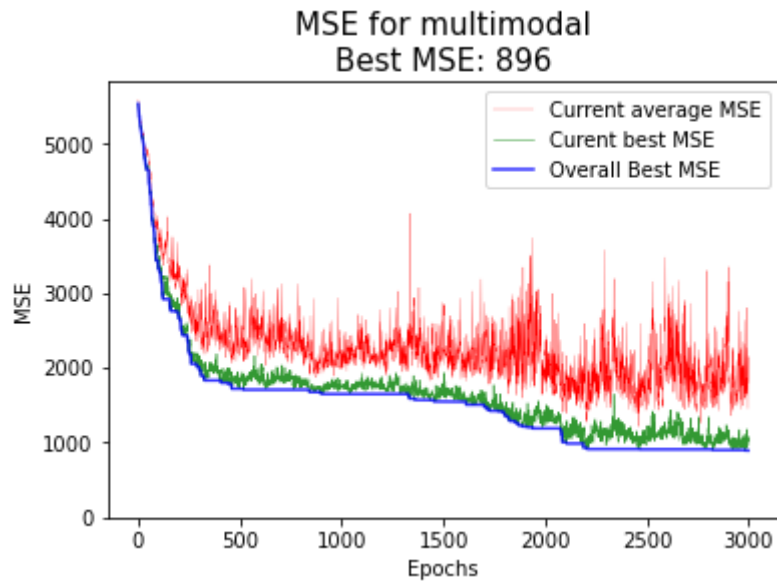
Przedstawione powyżej accuracy zostało wyznaczone dla całego zbioru iris, na którym także odbyło się sprawdzanie accuracy w trakcie szukania jej najlepszych wag algorytmem genetycznym.

## Multimodal

Z uwagi na problemy wydajnościowe, algorytm genetyczny używał do walidacji sieci jedynie subsetu 500 losowo wybranych obserwacji. Końcowo najlepsza sieć została sprawdzona na pełnych danych, a obie walidacje przedstawiają poniższe wykresy.



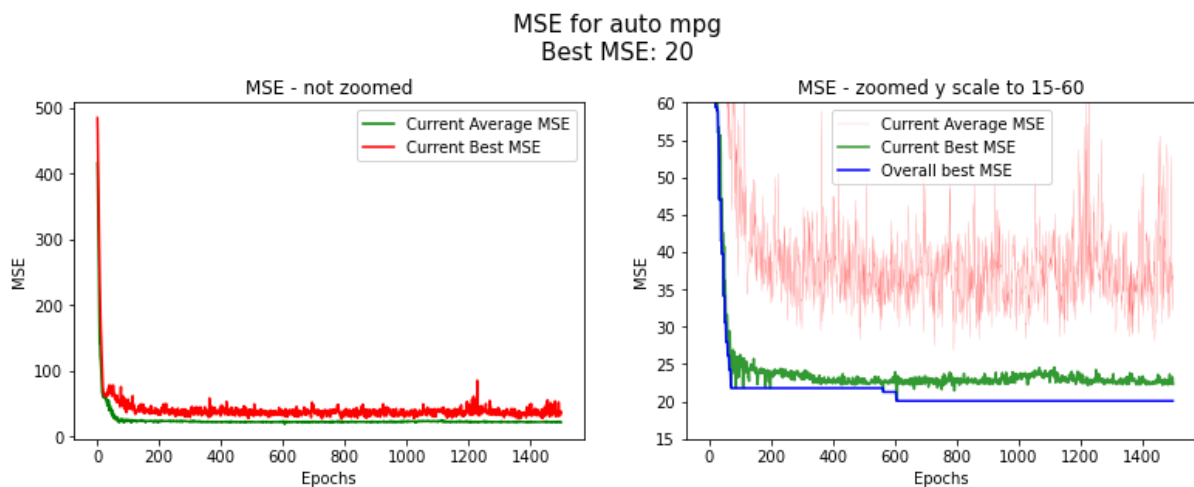
Wykres MSE pokazuje spore wahania średniego MSE. Podobny trend jest widoczny, ale słabiej, w MSE najlepszym dla danej epoki.



Trenowanie na jedynie 500 obserwacji jest w zupełności wystarczające. Alternatywnie można byłoby trenować na pełnych danych, pomieszanych, dzieląc dane na porcje (batches).

## Auto-mpg

Algorytm genetyczny szybko zbiega do poziomemu MSE około 20 i na takim poziomie się utrzymuje pomimo średniego MSE znacząco większego (około 35).



## Wyniki lab3 w tabelce

Dane	Iris	multimodal	auto-mpg
Wynik	Accuracy 0.987	MSE 870	MSE 20

# Podsumowanie

Algorytmy ewolucyjne są w stanie rozwiązywać skomplikowane problemy takie, jak znajdowanie minimum funkcji Rastrigina lub maksymalizacja wartości prostokątów umieszczonych w okręgu, a nawet znajdowanie wag do sieci neuronowej.

W przedstawionych wyżej problemach osiągnięte wyniki spełniają wymagania przedstawione w treści poleceń lub osiągając satysfakcjonujący poziom w przypadku ich braku.

Z obserwacji, podczas trenowania sieci neuronowych zauważyć można było, że w przypadku zbyt małej populacji model potrafił "utknąć" na pewnym poziomie i mieć trudności z niego wyjść.

Inną obserwacją, której dokonałem, było spostrzeżenie, że jeżeli mutacje oraz crossover dzieją się zbyt agresywnie, to sieć nie będzie w stanie się uczyć, a wręcz będzie się oduczać. Miało to miejsce najlepiej widoczne w przypadku sieci neuronowej, jeżeli mutacja lub crossover polegały na zamianie całej macierzy wag między dwoma losowymi warstwami.

Ostatnim wnioskiem jest, że im więcej epok, tym lepsze będą rezultaty, także w przypadku małych populacji. Natomiast w przypadku dużych populacji, jest większa różnorodność, więc zdarza się, że potrzebna będzie mniejsza liczba epok, aż któryś osobnik osiągnie znacznie lepsze wyniki.

W skrócie: dużo epok i populacji rozwiązać może praktycznie każdy problem.