

# Algorytmy genetyczne

Tomasz Makowski

nr indeksu 291132

Czerwiec 2020

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Podstawowy algorytm genetyczny</b>	<b>2</b>
2.1	Opis . . . . .	2
2.2	Dane . . . . .	2
2.3	Eksperymenty . . . . .	2
2.4	Wyniki i wnioski . . . . .	2
<b>3</b>	<b>Cutting Stock Problem</b>	<b>3</b>
3.1	Problem . . . . .	3
3.2	Rozwiązanie . . . . .	3
3.3	Dane . . . . .	4
3.4	Eksperymenty . . . . .	4
3.5	Wyniki i wnioski . . . . .	5
<b>4</b>	<b>NEAT</b>	<b>5</b>
4.1	Opis . . . . .	5
4.2	Dane . . . . .	8
4.3	Eksperymenty . . . . .	9
4.4	Wyniki i wnioski . . . . .	9

## 1 Wstęp

Celem ostatniej części laboratoriów jest zbadanie działania algorytmów genetycznych oraz wpływu poszczególnych parametrów na jakość wyników.

W każdym problemie była rozważana jakaś wariacja algorytmu genetycznego, w którym można było ustawić wielkość populacji. Następnie algorytm genetyczny uczył się ustaloną z góry liczbę epok i zachowywał cały czas początkową wielkość populacji. Nauka sieci polegała na mutacji osobników (wartości ciągle były najczęściej zaburzane zgodnie z rozkładem normalnym), krzyżowaniu osobników (zawsze z 2 osobników powstawał 1 nowy a rodzice również byli zachowywani), ewaluacji i selekcji najlepszych n osobników. W zależności od rozważanego algorytmu kolejność operacji mogła być różna.

## 2 Podstawowy algorytm genetyczny

### 2.1 Opis

Podstawowy algorytm genetyczny został zaimplementowany z następującą kolejnością operacji w jednej epoce:

1. Ewaluacja funkcji
2. Selekcja 50% najlepszych osobników
3. Krzyżowanie pomiędzy najlepszymi osobnikami, tak żeby populacja była stała
4. Mutacja losowych  $mutation\_percentage$  osobników za pomocą dodania wartości losowej z rozkładu normalnego  $\cdot mutation\_coefficient$

### 2.2 Dane

Algorytm został przetestowany na 2 zbiorach danych:

- Funkcja 3-wymiarowa –  $f(x, y, z) = x^2 + y^2 + 2 \cdot z^2$
- 5-wymiarowa funkcja Rastrigina –  $f(x) = 10 \cdot n + \sum_{i=1}^n [x_i^2 - 10 \cdot \cos(2\pi x_i)]$ , dla  $n = 5$

### 2.3 Eksperymenty

Został zbadany wpływ 3 parametrów:

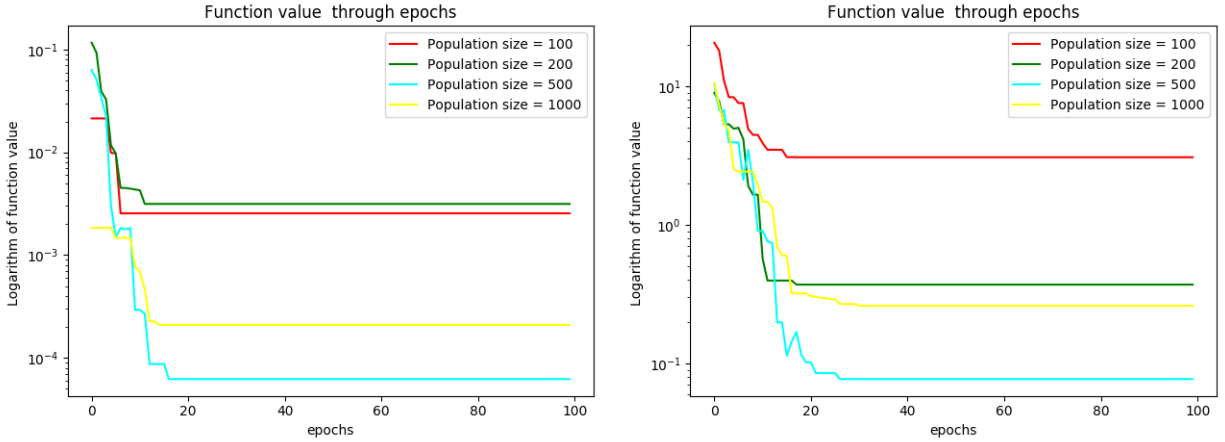
1. wielkość populacji – była ustawiona z listy [100, 200, 500, 1000], pozostałe parametry:  $mutation\_percentage = 0.2$ ,  $mutation\_coefficient = 1$
2.  $mutation\_percentage$  – ustawione z listy [0.05, 0.1, 0.2, 0.3], pozostałe parametry: wielkość populacji 500,  $mutation\_coefficient = 1$
3.  $mutation\_coefficient$  – porównanie pomiędzy [0.1, 0.2, 0.5, 1], pozostałe parametry: wielkość populacji 500,  $mutation\_percentage = 0.1$  dla funkcji 3-wymiarowej oraz  $mutation\_percentage = 0.3$  dla funkcji Rastrigina

### 2.4 Wyniki i wnioski

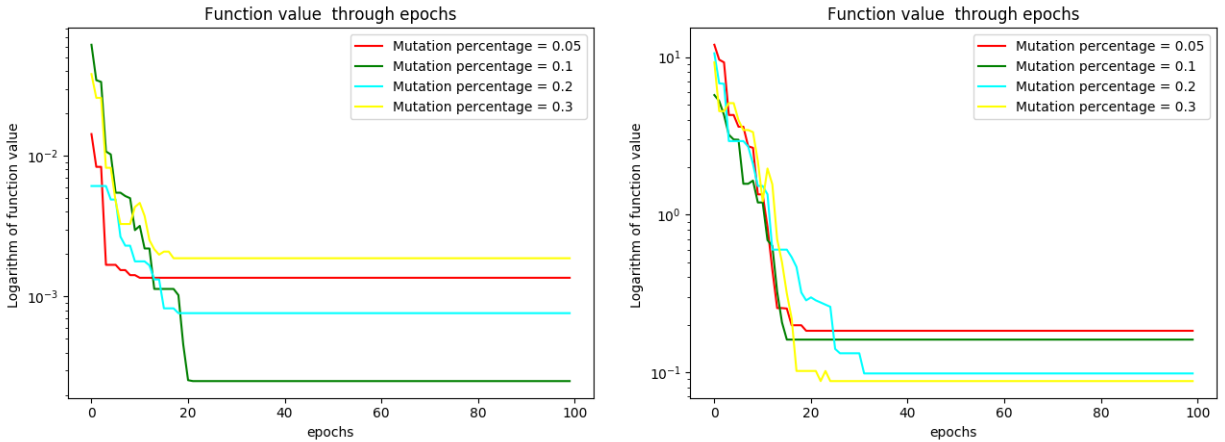
Na wykresie 1 został przedstawiony wpływ wielkości populacji w zależności od epoki na jakość wyników działania algorytmu. Im mniejsza wartość tym lepiej, ponieważ w przypadku obu funkcji należało znaleźć minimum, które było równe 0. W obu przypadkach najlepsza okazała się populacja licząca 500. Z tego powodu w następnym eksperymencie właśnie taka wielkość została użyta.

Na wykresie 2 został przedstawiony wpływ parametru  $mutation\_percentage$ , czyli prawdopodobieństwa mutacji każdego elementu populacji niezależnie, w zależności od epoki na jakość wyników działania algorytmu. Wyniki nie były niezależne od badanej funkcji i tak w przypadku funkcji 3-wymiarowej najlepsza wartość to 0.1 a najgorsza 0.3. Natomiast w przypadku funkcji Rastrigina najlepsze było 0.3 a najgorsze 0.05.

Na wykresie 3 został przedstawiony wpływ parametru  $mutation\_coefficient$ , czyli mnożnika zaburzenia z rozkładu normalnego o średniej 0 i wariancji 1, w zależności od epoki na jakość wyników działania algorytmu. W obu przypadkach okazało się, że mniejsza wartość tego współczynnika determinował lepszy wynik (z niewielkim odchyleniem od tej konkluzji).



Rysunek 1: Wpływ wielkości populacji na wartość funkcji najlepszego osobnika w populacji odpowiednio dla funkcji 3-wymiarowej (po lewej stronie) i funkcji Rastrigina (po prawej stronie)



Rysunek 2: Wpływ procentu mutowanych osobników na wartość funkcji najlepszego osobnika w populacji odpowiednio dla funkcji 3-wymiarowej (po lewej stronie) i funkcji Rastrigina (po prawej stronie)

### 3 Cutting Stock Problem

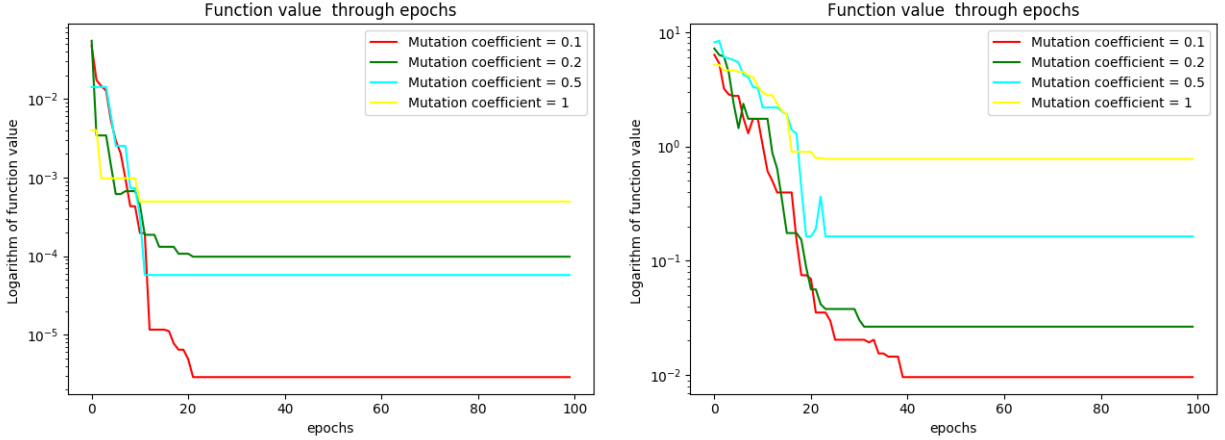
#### 3.1 Problem

Dane jest koło o promieniu  $r$  oraz kilka możliwych do wykorzystania prostokątów o podanych wymiarach (długość i szerokość) oraz wartości. Zadaniem jest ułożyć prostokąty wewnątrz koła tak, by uzyskać największą sumę wartości tych prostokątów. Prostokąty nie mogą się przecinać i boki mają być ułożone równoległe do 2 przecinających się prostopadłych prostych (nie mogą być układane pod skosem).

#### 3.2 Rozwiązanie

Rozwiązanie opiera się na podziale koła na poziome paski. Następnie paski wypełniane są zachłannie zaczynając od prostokąta, który da największy wzrost na jedną jednostkę szerokości paska.

Gen ciągle koduje jak wysoko zaczyna się "pierwszy" górny pasek - jest to liczba od 0 do największego



Rysunek 3: Wpływ wielkości zaburzeń podczas mutacji na wartość funkcji najlepszego osobnika w populacji odpowiednio dla funkcji 3-wymiarowej (po lewej stronie) i funkcji Rastrigina (po prawej stronie)

wymiaru spośród dostępnych prostokątów (dla  $r=800$  jest to 400). Geny dyskretne kodują, jaką wysokość mają kolejne paski (paski stykają się ze sobą). Dla  $r=800$  dostępne wartości to 30, ..., 400, zatem 1 koduje wysokość 30, a 7 koduje wysokość 400. Pierwsze  $\lfloor \frac{r}{30} \rfloor$  koduje wielkości pasków na górnej części, a następne  $\lfloor \frac{r+400}{30} \rfloor$  koduje wielkości pasków w dolnej części. Oczywiście bardzo często ostatnie geny z obu "połówek" nie są używane ponieważ wymagają konkretnych wartości w innych genach.

Mutacja to dodanie liczb do genów z rozkładem normalnym. Dla genów dyskretnych jest to wartość całkowita zatem bardzo wiele genów nie mutuje - co jest raczej dobre (bo zmiany we wcześniejszych genach mają duży wpływ na ewaluację następnych genów).

Krzyżowanie jest to krzyżowanie jednopunktowe, gdzie gen ciągły jest brany z pierwszego wybranego osobnika.

### 3.3 Dane

Dane zawierają koła o promieniach 800, 850, 1000, 1100, 1200 oraz po 5 dostępnych prostokątów do układania.

### 3.4 Eksperymenty

Został zbadany wpływ 2 parametrów:

1. wielkość populacji – porównanie pomiędzy [500, 1000, 1500, 2000], przy  $mutation\_percentage = 0.2$
2.  $mutation\_percentage$  – porównanie pomiędzy [0.1, 0.2, 0.3, 0.5], podczas, gdy wielkość populacji wynosiła 1500

Wyniki zostały przedstawione jako *normalized\_score*, czyli wartość uzyskana jako suma punktów podzielona przez największą możliwą wartość możliwą do uzyskania z jednostki kwadratowej oraz podzieloną przez obszar koła. Zatem ta miara daje wyniki między 0 i 1, co oznacza, że jest dobra do porównywania wyników.

Ostatecznie zostały jeszcze raz obliczone wyniki dla populacji równej 1500 i  $mutation\_percentage = 0.3$ , żeby porównać wynik dla różnych wielkości okręgu oraz 2 inne miary:

- Possible area percent – procent koła, który był dostępny do użycia przez paski. Ponieważ paski były przycinane do prostokątów to w każdym pasku zostawały niewykorzystane skrawki i właśnie dopełnienie tej miary do 1 informuje o procentowej powierzchni skrawków.
- Used area percent – procent koła wykorzystany przez prostokąty. Jest to pole liczone jako długość wykorzystana przez prostokąty w pasku razy szerokość paska. A następnie suma po paskach jest dzielona przez powierzchnię koła.

### 3.5 Wyniki i wnioski

Na wykresie 4 został przedstawiony wpływ wielkości populacji w zależności od epoki na jakość wyników działania algorytmu. Generalizując po wszystkich  $r$  najlepsza okazała się wielkość populacji 1500 z racji na stabilne wyniki i często ostateczne znajdowanie maksimum jakie osiągnęły inne funkcje. Jedynym wyjątkiem jest  $R = 1100$ , ale gdyby implementacja zawsze zachowywała kopię najlepszego osobnika to algorytm miałby taką samą wartość jak pozostałe. Zatem wielkość populacji 1500 będzie używana dalej.

Na wykresie 5 został przedstawiony wpływ parametru *mutation\_percentage*, czyli prawdopodobieństwa mutacji każdego elementu populacji niezależnie, w zależności od epoki na jakość wyników działania algorytmu. Dla wszystkich promieni z wyjątkiem  $R = 850$  osiągane maksima były identyczne. Jedynie w przypadku  $R = 850$  wyraźnie lepiej poradził sobie algorytm z *mutation\_percentage* = 0.3 osiągając dużo większe maksimum jako pierwszy. Zatem ta wartość zostanie użyta w następnym porównaniu.

Na wykresie 6 zostało przedstawione porównanie pomiędzy konkretnymi  $R$ . Można zauważyć, że im większe  $R$  tym lepsze wyniki – jest to prawdopodobnie spowodowane mniejszymi wycinkami koła zostającymi poza obszarem użycia, na co wskazuje wykres possible area percent. Jedynym wyjątkiem od tej reguły jest  $R = 850$  dla którego possible area jest dużo większa w stosunku do wyniku.

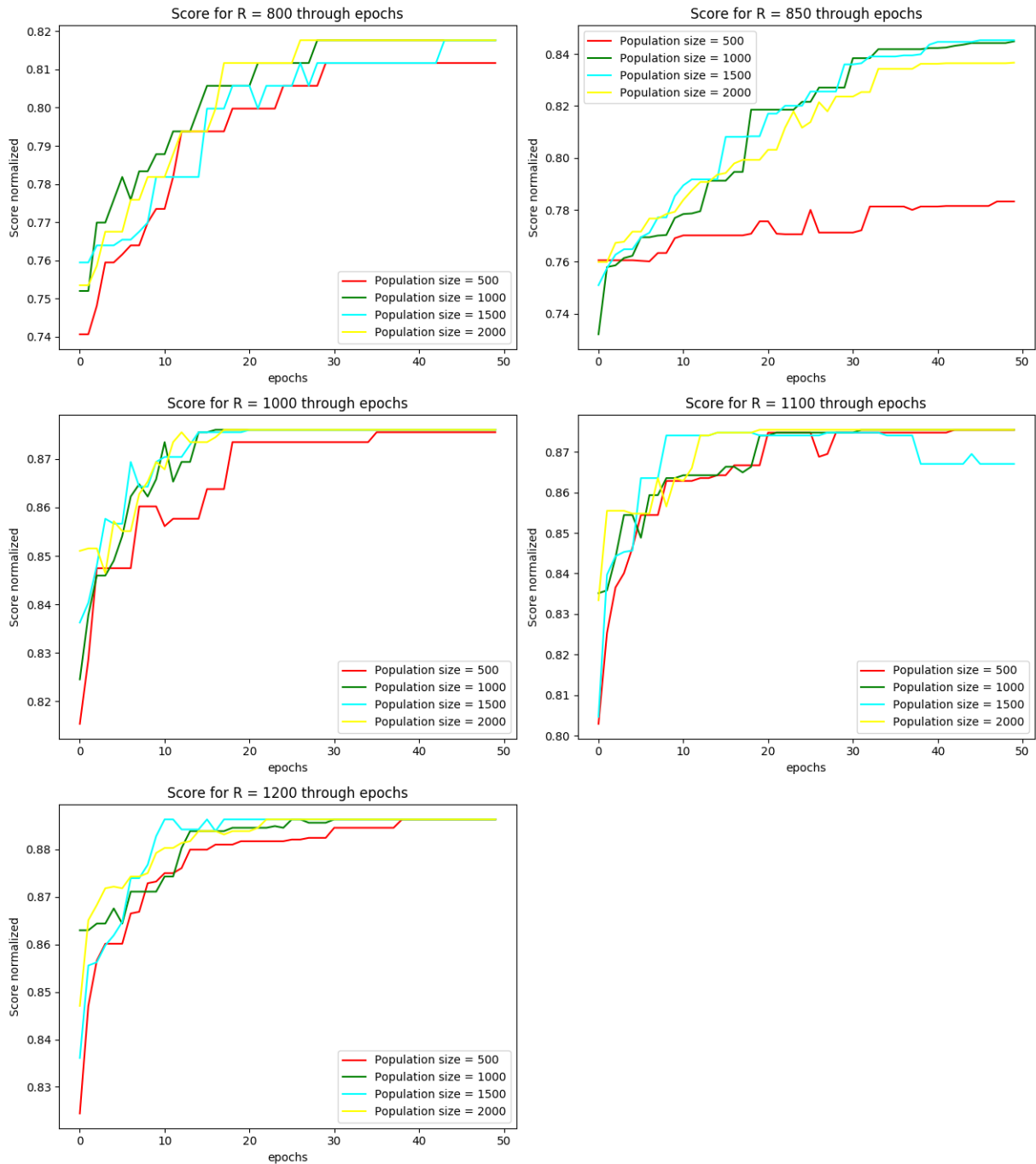
## 4 NEAT

### 4.1 Opis

NEAT jest to algorytm ewolucyjny, który steruje zarówno architekturą sieci neuronowej (neurony i połączenia) jak i samymi wagami takiej sieci.

Podczas ewaluacji jednej epoki wykonywane są następujące operacje:

1. Podział na gatunki (jest to opcjonalne i wpływ tego elementu będzie pokazany dalej)
2. Krzyżowanie – prawdopodobieństwo krzyżowania to 75% (jeśli podział na gatunki był aktywny to krzyżowanie odbywa się wewnątrz gatunku)
3. Mutacje:
  - Mutacja wagi – 80% na mutowanie wag danej sieci, przy czym podczas mutowania na 90% była dodawana liczba z rozkładu normalnego, a na 10% nadawana nowa waga
  - Dodanie połączenia – prawdopodobieństwo 95% na dodanie nowego połączenia między neuronami, które jeszcze nie były połączone, było możliwe tworzenie cykli.
  - Dodawanie neuronu – prawdopodobieństwo 97% na dodanie nowego neuronu na jednym z istniejących połączeń. W takim przypadku jedna waga jest przepisywana, a druga waga dostaje wartość 1.

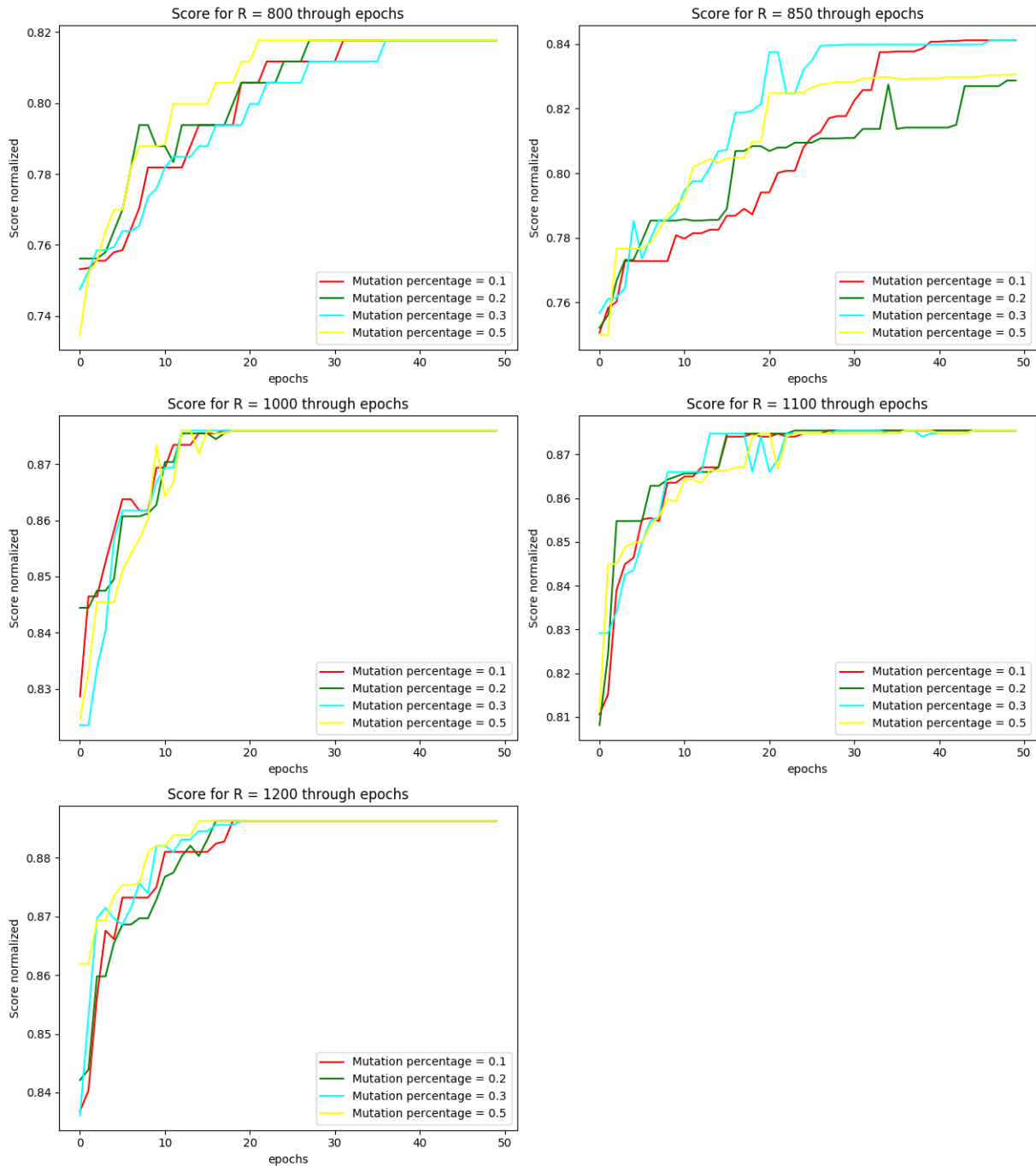


Rysunek 4: Wpływ wielkości populacji na znormalizowany wynik najlepszego osobnika w populacji

- Aktywacja/deaktywacja krawędzi – prawdopodobieństwo 97% na wybór jednej z losowych krawędzi sieci i zmianę jej stanu aktywacji na przeciwny.

4. Ewaluacja – podczas ewaluacji robiony był forward z funkcją aktywacji sigmoid oraz funkcją aktywacji softmax na ostatniej warstwie. Jako miarę jakości danej sieci została wykorzystana jedna z 2 miar:

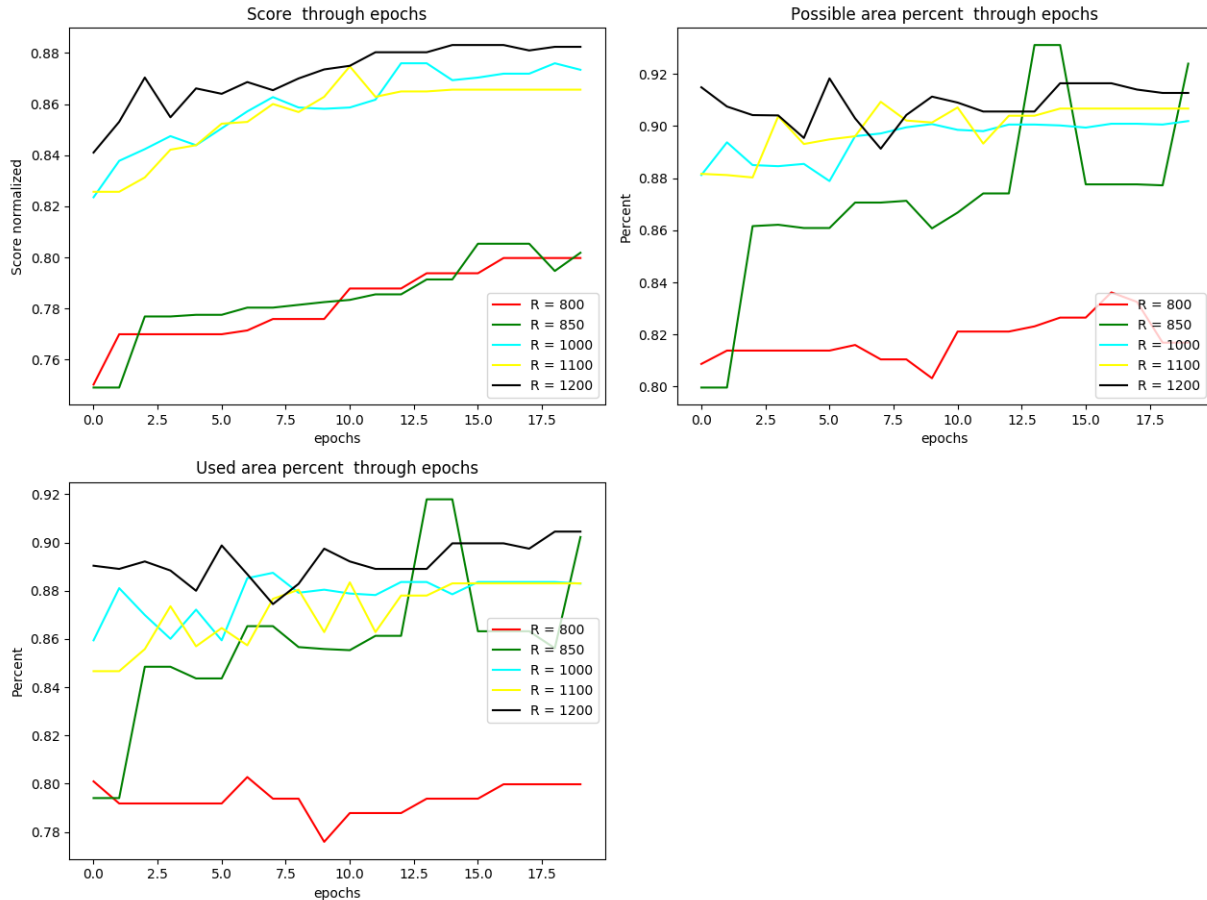
- Accuracy – czyli procent poprawnie przyporządkowanych odpowiedzi



Rysunek 5: Wpływ prawdopodobieństwa mutacji osobników na znormalizowany wynik najlepszego osobnika w populacji

- Score – czyli rodzaj - MSE (minus po to, żeby maksymalizować), polegał na sumie po wszystkich klasach i obserwacjach kwadratu różnicy pomiędzy spodziewanym wynikiem (1 lub 0 - w zależności, czy dana klasa była poprawna czy nie) oraz wynikiem przewidzianym przez sieć (pomiędzy 0 a 1).

5. Selekcja – selekcja rankingowa tylu osobników, by wielkość populacji była stała. W przypadku imple-



Rysunek 6: Porównanie wyników dla różnych rozmiarów koła

mentacji z gatunkami dodatkowo gatunki mogły wymierać po 15 epokach bez poprawy oraz z każdego gatunku powyżej 5 osobników był zachowywany jeden osobnik, który miał największą wartość funkcji przystosowania.

## 4.2 Dane

Algorytm został przetestowany na 4 zbiorach danych:

- Zbiór easy – jest to liniowo separowalny zbiór służący głównie do sprawdzenia poprawności implementacji. Nie był używany do eksperymentów, ponieważ łatwo było znaleźć sieć ze skutecznością 97%
- Zbiór rings3-regular – zawiera 3 klasy będące fragmentami pierścienia i otaczającej przestrzeni. Jest to zbiór wymagający zaawansowanej architektury sieci.
- Zbiór Iris – jest to zbiór o 3 gatunkach irysów zawierający informacje liczbowe o konkretnych kwiatach takie jak długość i szerokość słupków oraz pręcików. Jedna klasa jest łatwo separowalna, a pozostałe 2 stykają się ze sobą.
- Zbiór Adult – jest to zbiór zawierający informacje o danych ludzi z USA (płeć, wykształcenie, rasa, itp.) oraz informacja czy dana osoba zarabia więcej niż 50k \$ rocznie czy nie, jest to wartość przewidywana



### 4.3 Eksperymenty

Zostały przeprowadzone 2 eksperymenty:

1. Czy używanie podziału na gatunki oraz miara jakości do ewaluacji wyników (accuracy / score bazujący na MSE) ma wpływ na wyniki, w przypadku używania gatunków został użyty  $\text{threshold} = 1.3$ , sprawdzone na zbiorach iris i rings3.
2. Dla miary jakości accuracy i używania gatunków sprawdzenie, która wartość thresholdu do powstania nowego gatunku z pośród  $[1, 1.3, 1.5]$  daje najlepsze wyniki. Sprawdzone na zbiorze iris.

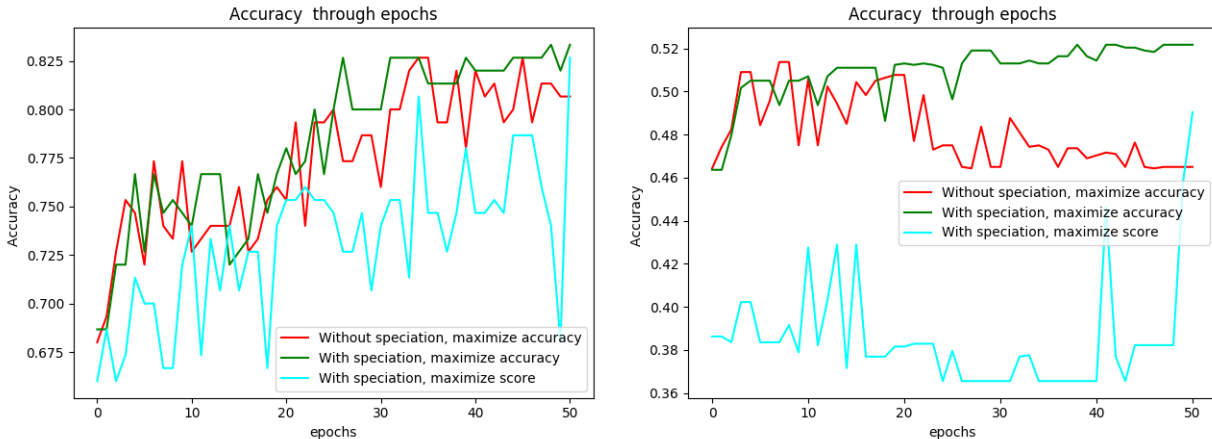
Dodatkowo została przeprowadzona jedna ewaluacja na zbiorze adult.

### 4.4 Wyniki i wnioski

Na wykresie 7 zostały przedstawione wyniki accuracy w zależności od epoki. Można zauważyć, że specjacja (podział na gatunki) daje niewiele lepsze wyniki niż jej brak. Dużo większy skok jakości jest przy użyciu funkcji accuracy zamiast score.

Jeżeli chodzi o wyniki przedstawione jako podział punktów na 8 można zauważyć, że wszystkie 3 podejścia miały problem znaleźć poprawny podział klas 1 i 2. Wykres był zrobiony poprzez użycie PCA - stąd może być uzasadnione tak mocne pomieszanie klas.

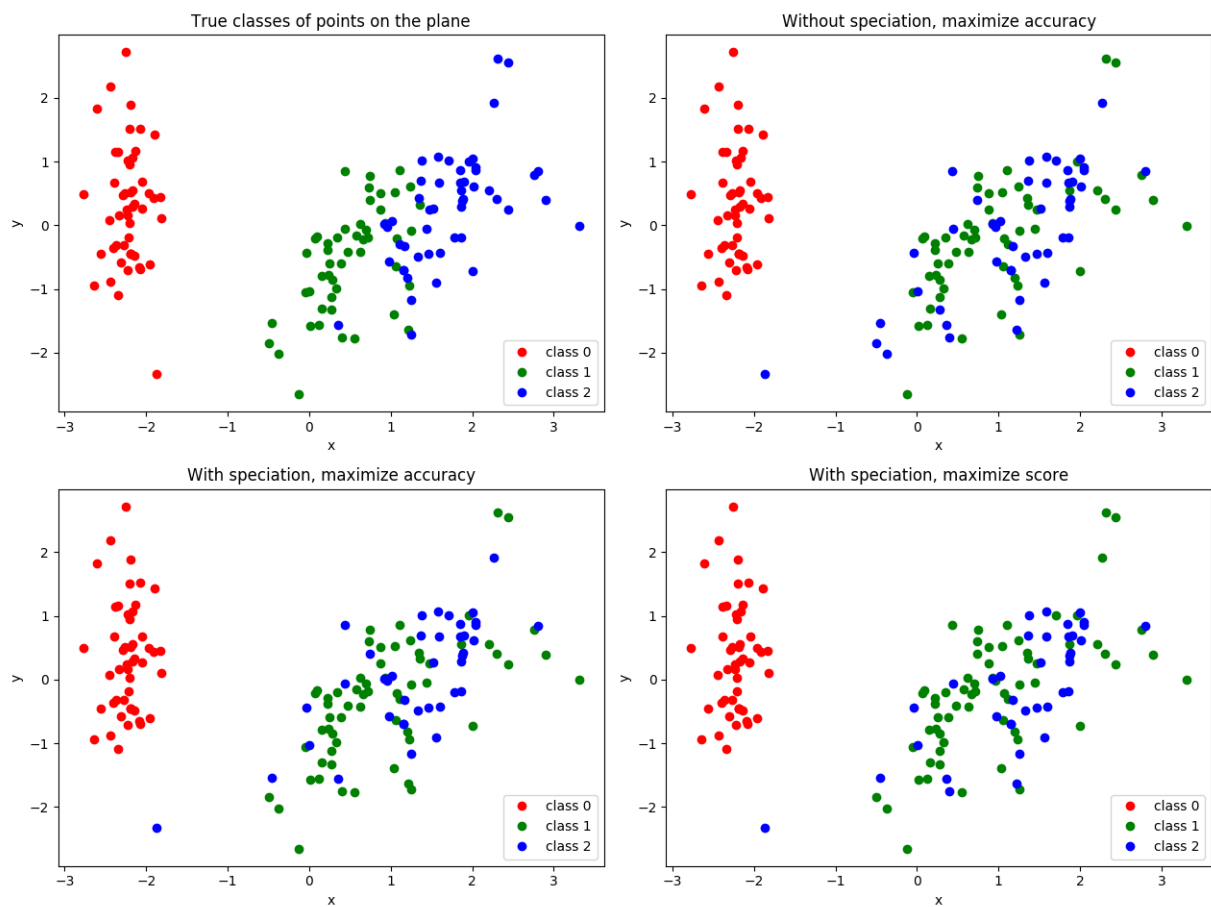
W przypadku podziału na klasy dla zbioru rings3 na 9 można zaobserwować, że sieć nie była w stanie się nauczyć żadnych kształtów. Jedyna różnica jest taka, że podział na gatunki nie był w stanie wykryć żadnego elementu klasy 2 (niebieskiej).



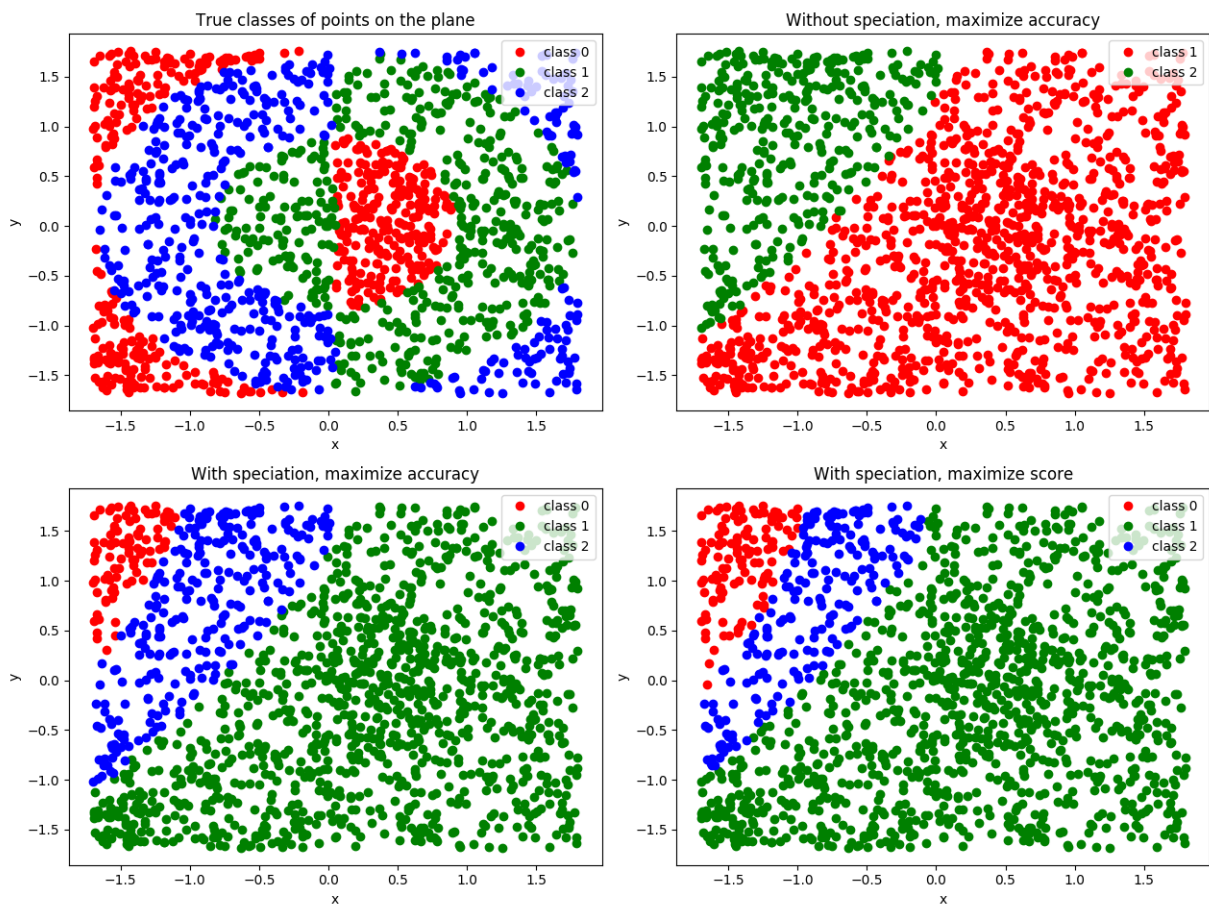
Rysunek 7: Wpływ używania podziału na gatunki oraz miary jakości do ewaluacji wyników na accuracy najlepszego osobnika w populacji odpowiednio dla zbioru iris (po lewej stronie) i ring3-regular (po prawej stronie)

Na wykresie 10 został przedstawiony wpływ thresholdu dla nowego gatunku. Różnice pomiędzy thresholdami są niewielkie i raczej nieznaczące. Co prawda  $\text{threshold} = 1.3$  dawał dobre wyniki przy najniższej epoce, ale mogła to być kwestia ziarna losowego.

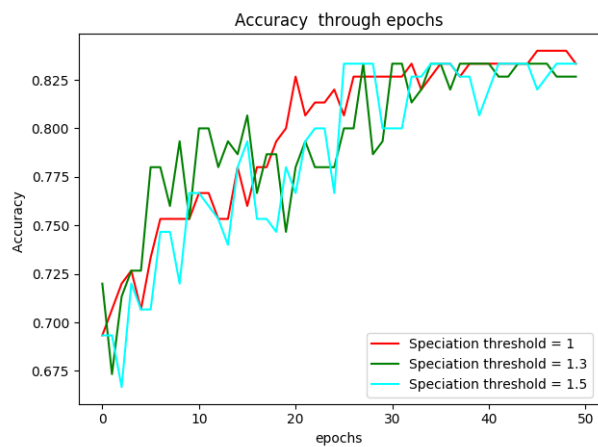
Na wykresie 11 zostały przedstawione wyniki dla zbioru adult. Jedyne co da się wyczytać, to że im więcej epok tym lepsze wyniki. Poprawa względem epok jest mniejsza niż w przypadku zbioru iris, ale większa niż dla zbioru rings3.



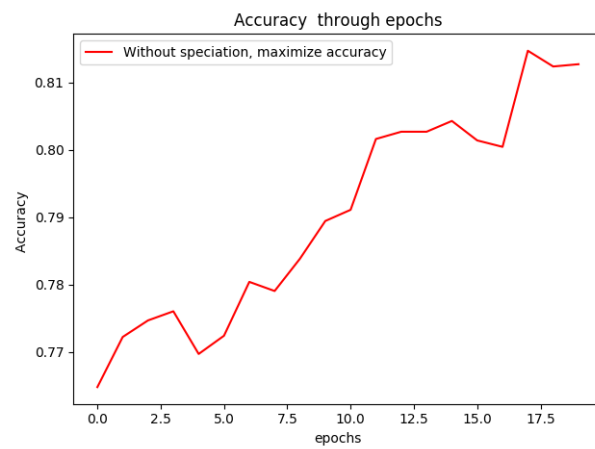
Rysunek 8: Wpływ używania podziału na gatunki oraz miary jakości do ewaluacji wyników na wizualny podział zbioru danych dla najlepszego osobnika w populacji dla zbioru iris



Rysunek 9: Wpływ używania podziału na gatunki oraz miary jakości do ewaluacji wyników na wizualny podział zbioru danych dla najlepszego osobnika w populacji dla zbioru ring3-regular



Rysunek 10: Wpływ thresholdu do ustanawiania nowego gatunku na accuracy najlepszego osobnika w populacji dla zbioru iris



Rysunek 11: Wyniki dla zbioru adult