

Sprawozdanie

Analiza algorytmu genetycznego

Wykonał: Volodymyr Zakhovaiko, 251526

Content

Zbadanie wpływu parametrów:	4
Funkcja oceny	4
Użyte płytki do testowania	4
Badania	8
Porównanie operatorów selekcji „Ruletka” i „Turniej”	8
Wniosek	12
Zbadanie wpływu parametrów:	12
Krzyżowanie	12
Wniosek	17
Mutacja	18
Wniosek	21
Rozmiar populacji.....	21

Wagi oceny.....	24
Rozmiar turnieju	24
Porównanie algorytmu genetycznego z metodą losową	24
Wynik	27
Najlepsze rozwiązania.....	27

Zbadanie wpływu parametrów:

Parametry, które używam w algorytmie:

- Rozmiar populacji
- Liczba pokoleń
- Rozmiar turnieju
- Szansa krzyżowania
- Szansa mutacji
- Szansa mutacji (rozbicie na dwie ścieżki)
- Wagi dla oceny
- Inne, nie odnoszące się do GA

Znaczenia niebadanych parametrów:

- Max. ilość losowo generowanych segmentów: **4**
- Max. długość losowo generowanego segmentu: **3**
- Krzyżowanie: **jednopunktowe**

Funkcja oceny

Funkcja oceny została zrealizowana wg tej, która była omówiona w zadaniu ćwiczenia. Przy implementacji metody obliczenia oceny osobnika zostało użyte:

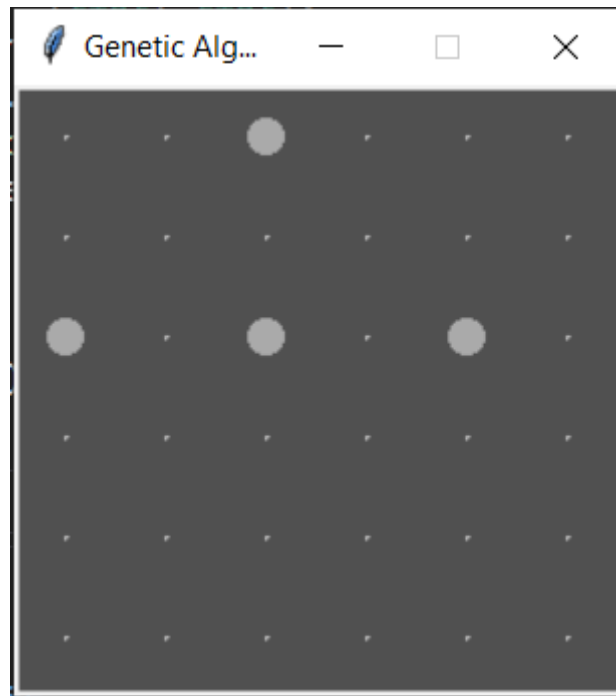
- Liczba przecięć
- Sumaryczna długość ścieżek
- Sumaryczna liczba segmentów tworzących ścieżki
- Liczba ścieżek poza płytką
- Sumaryczna długość części ścieżek poza obszarem płytki

Wygląd matematyczny funkcji:

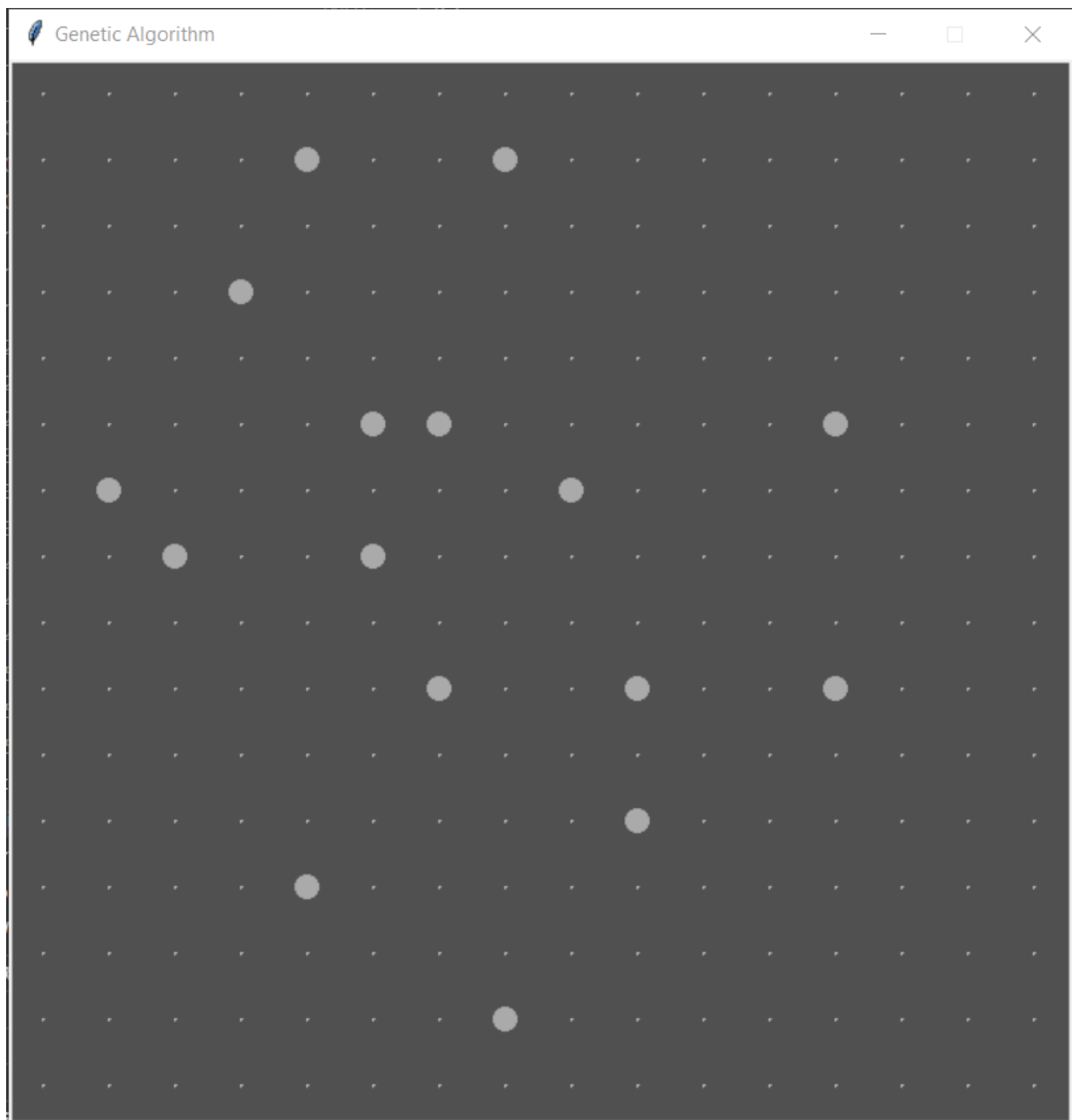
$$f = w1 * x1 + w2 * x2 + w3 * x3 + w4 * x4 + w5 * x5,$$

gdzie $w1..5$ – wagi dla oceny, $x1..5$ – wyżej ukazane liczby.

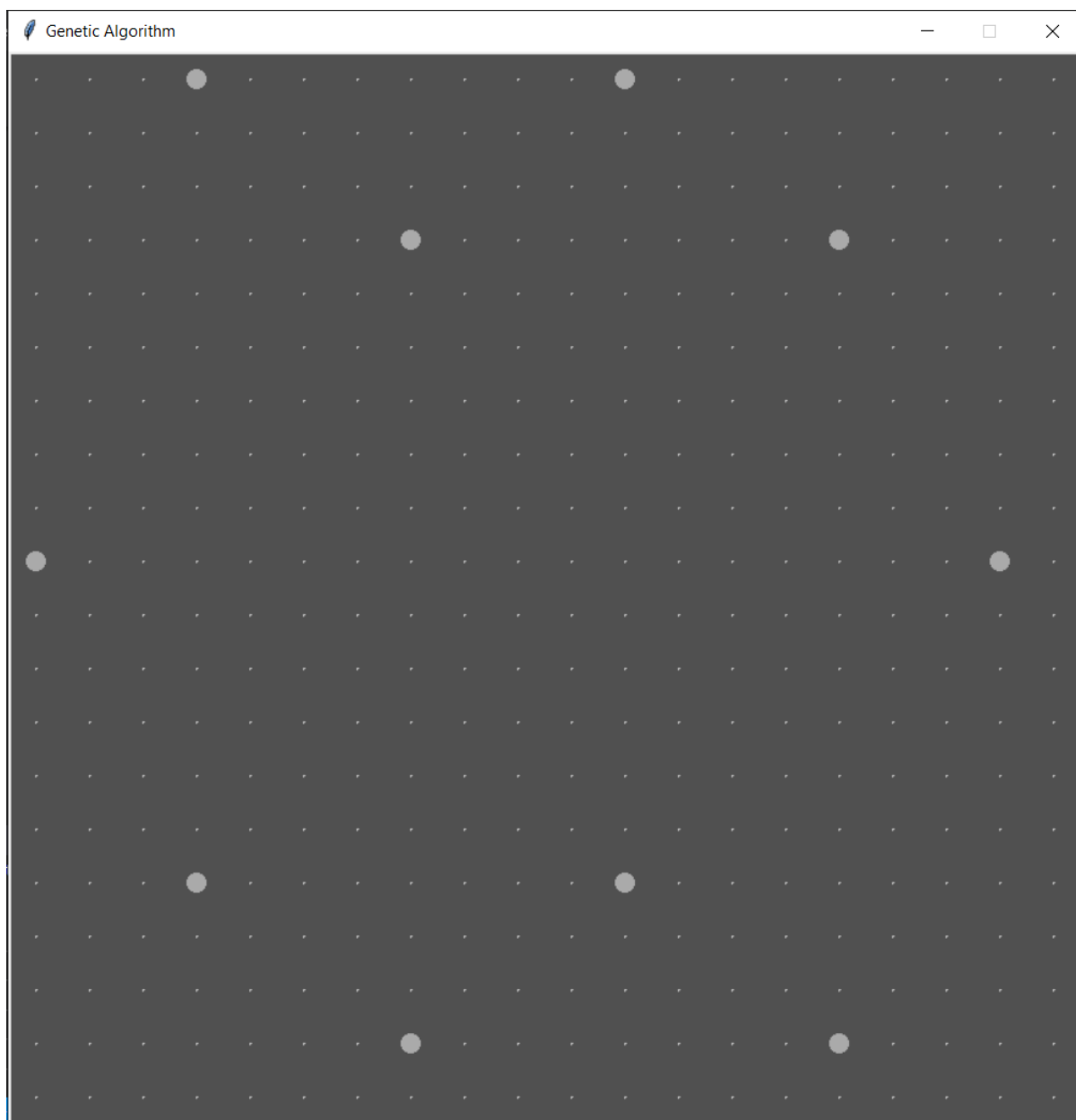
Użyte płytki do testowania



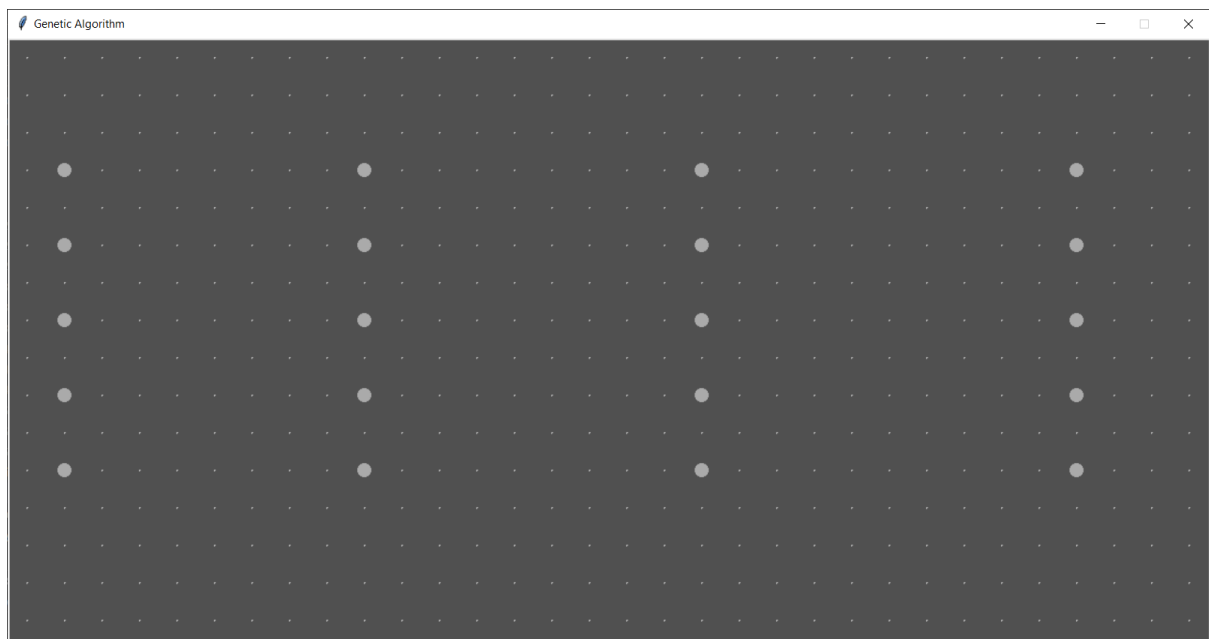
Rys. 1 – Plik zad0.txt



Rys. 2 – Plik zad1.txt



Rys. 3 – Plik zad2.txt



Rys. 4 – Plik zad3.txt

Badania

Niżej będziemy próbowali zmieniać parametry dla lepszych wyników.

Dla początku musimy ustawić seed `random.seed(1)`, żeby z każdym uruchomieniem algorytmu dostawać te same losowe liczby.

Teraz ustawimy takie parametry algorytmu genetycznego:

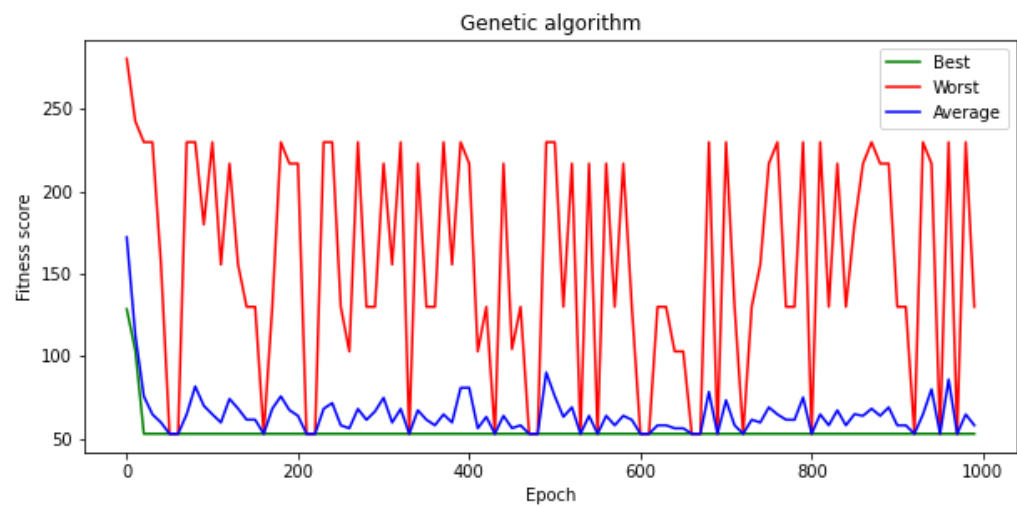
- Rozmiar populacji: **15**
- Szans krzyżowania: **0.8, 80%**
- Szans mutacji: **0.1, 10%**
- Ilość pokoleń: **1000**
- Rozmiar turnieju: **5**
- Wagi: **[25, 0, 0.7, 12, 17]**

Porównanie operatorów selekcji „Ruletka” i „Turniej”

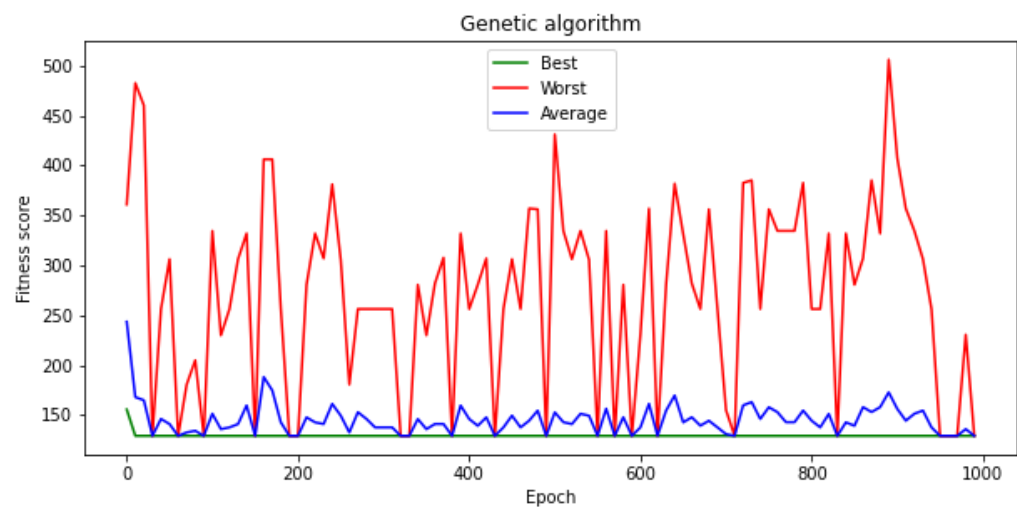
Na pierwszym zdjęciu - ustawiamy typ selekcji osobników na „ruletka” i badamy na podanego przykładu.

Na drugim zdjęciu – nie zmieniając parametrów, a tylko typ selekcji na „turniej” badamy te same pliki.

1. Badanie dla pliku zad0.txt

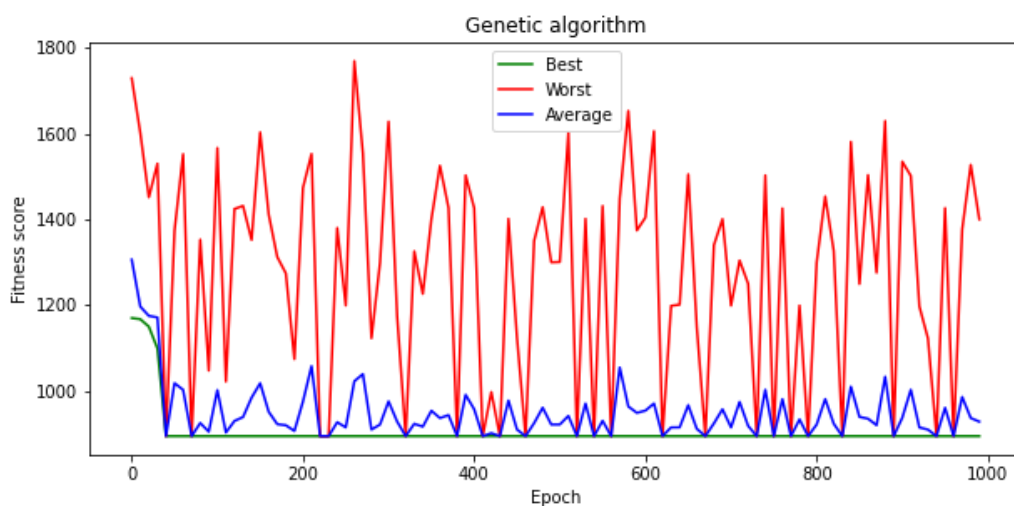


Rys. 5 – Ruletko zad0.txt

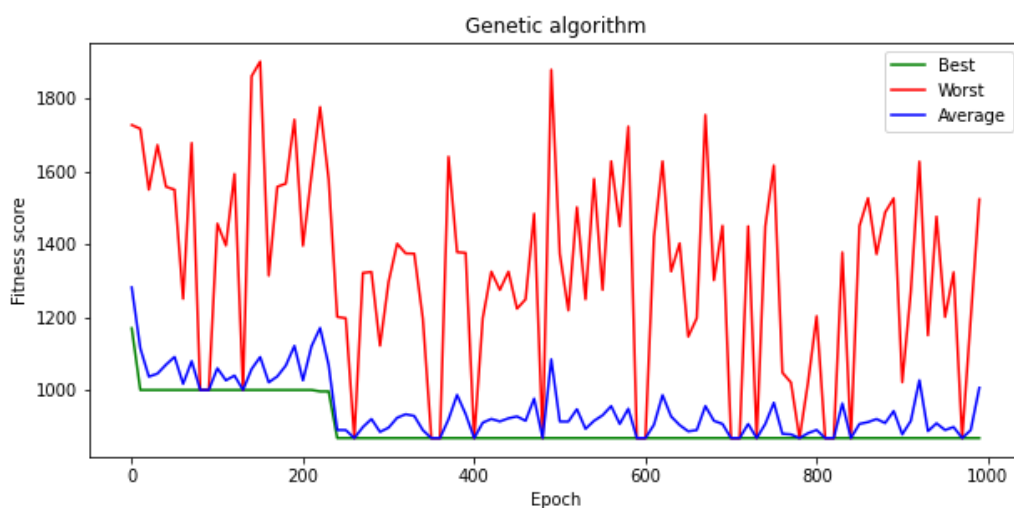


Rys. 6 – Turniej zad0.txt

2. Badanie dla pliku zad1.txt

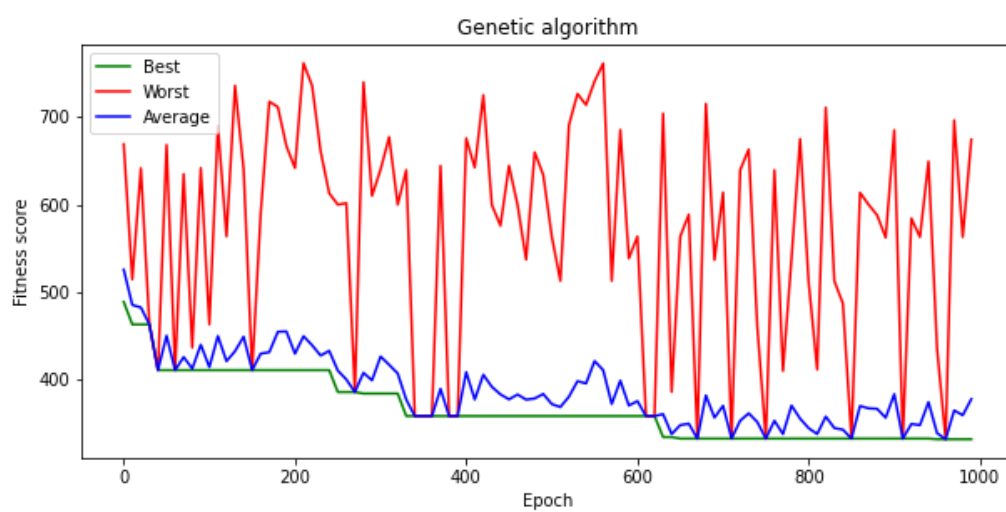


Rys. 7 – Ruletko zad1.txt

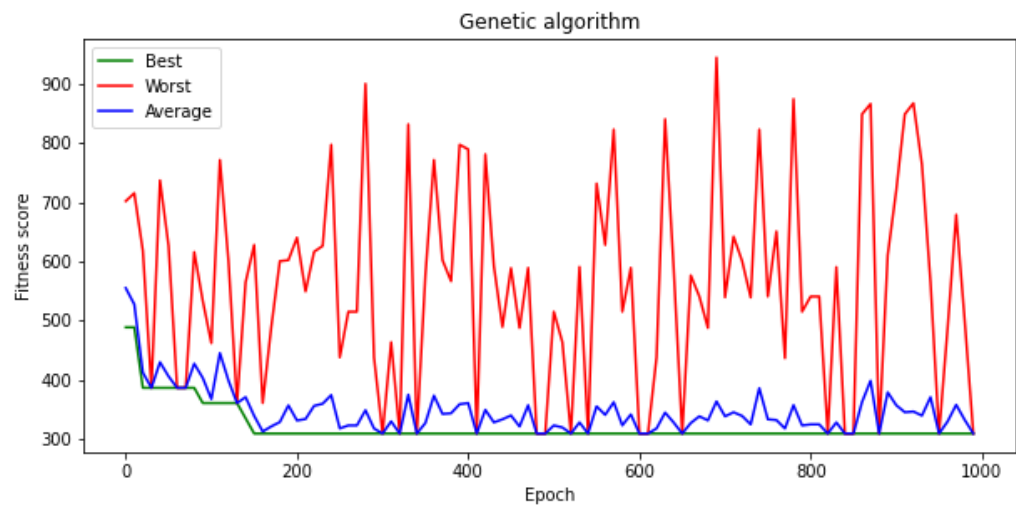


Rys. 8 – Turniej zad1.txt

3. Badanie dla pliku zad2.txt

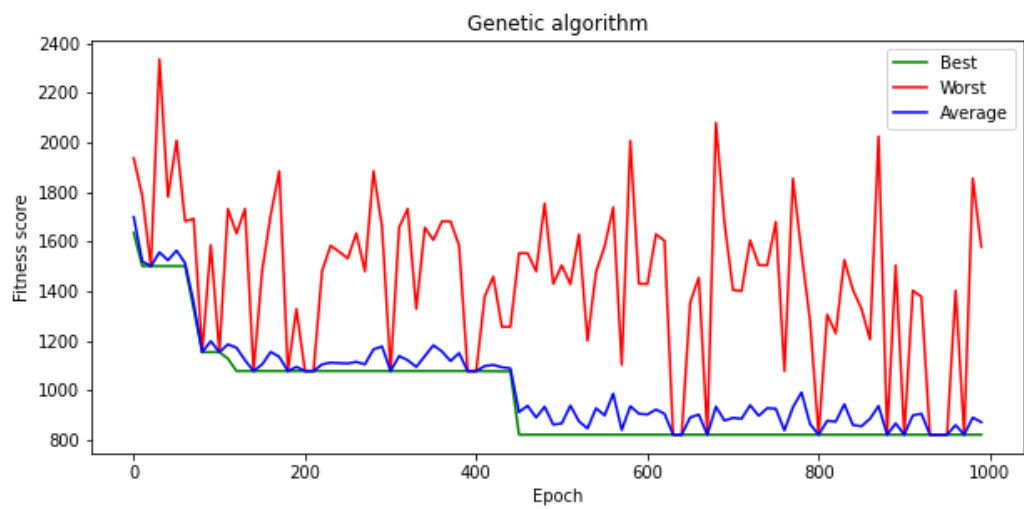


Rys. 9 – Ruletko zad2.txt

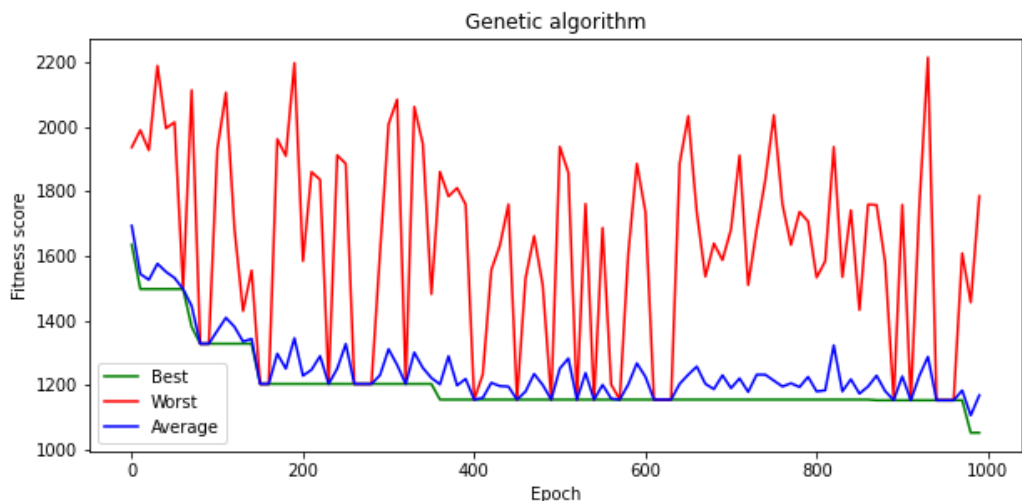


Rys. 10 – Turniej zad2.txt

4. Badanie dla pliku zad3.txt



Rys. 11 – Ruletko zad3.txt



Rys. 12 – Turniej zad3.txt

Wniosek

Jak widać z eksperymentów, najgorsza ocena przy użyciu selekcji „ruletką” zmniejsza się. Natomiast przy selekcji turniejowej odwrotnie, zwiększa się. Rzeczywiście jest, że ocena średnia przy użyciu selekcji turniejowej jest większa, za rachunek tego, że gorsze osobniki są wybierane częściej. To wszystko wychodzi z tego faktu, że przy selekcji „ruletką”, lepsze osobniki mają większe szanse na możliwość ich wybrania.

Ale. Jeżeli będziemy używali zawsze selekcję „ruletką”, to może nastąpić moment lokalnego minimum. Czyli moment, gdy algorytm będzie wciąż wybierał dobrych osobników, chociaż gorsze będą bardziej prawidłowymi. Metoda selekcji turniejowej daje szansę algorytmu genetycznemu na znalezienie rozwiązania w znacznie inny sposób, żeby uniknąć tego lokalnego minimum.

Oprócz tego, co było wyżej opowiadano, metody te musimy kombinować ze zmianą parametrów, żeby dostać jak najszybciej prawidłowy rezultat.

Zbadanie wpływu parametrów:

Dalej będziemy próbowali znaleźć rozwiązanie przez zmianę parametrów. Najpierw ustawimy `random.seed(10000)`.

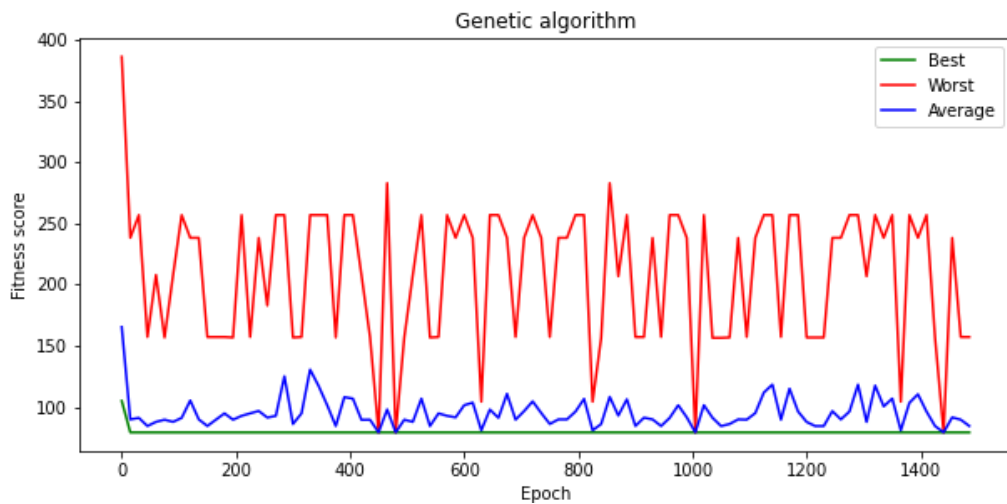
Zmienimy parametry na następujące:

- Rozmiar populacji: **15**
- Szans krzyżowania: **0.8, 80%**
- Szans mutacji: **0.14, 14%**
- Ilość pokoleń: **1500**
- Rozmiar turnieju: **5**
- Wagi: **[25, 0.2, 0.7, 12, 15]**
- Typ selekcji: **selekcja ruletką**

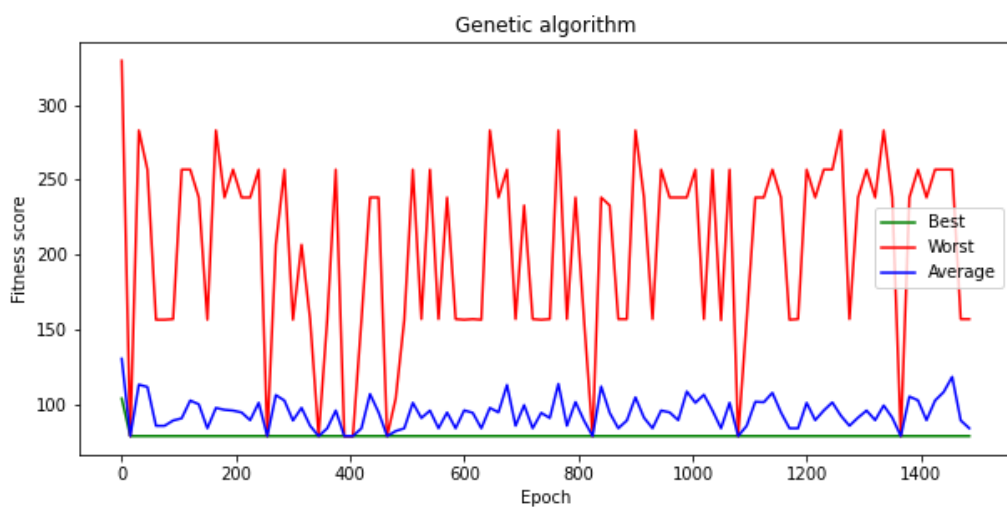
Krzyżowanie

Dla jednakowych parametrów będziemy zmieniali tylko szansę krzyżowania. Ćwiczyć będziemy na wszystkich płytkach testowych. Dla każdej płytki testujemy dla takich prawdopodobieństw: **0%, 25%, 75%, 100%**.

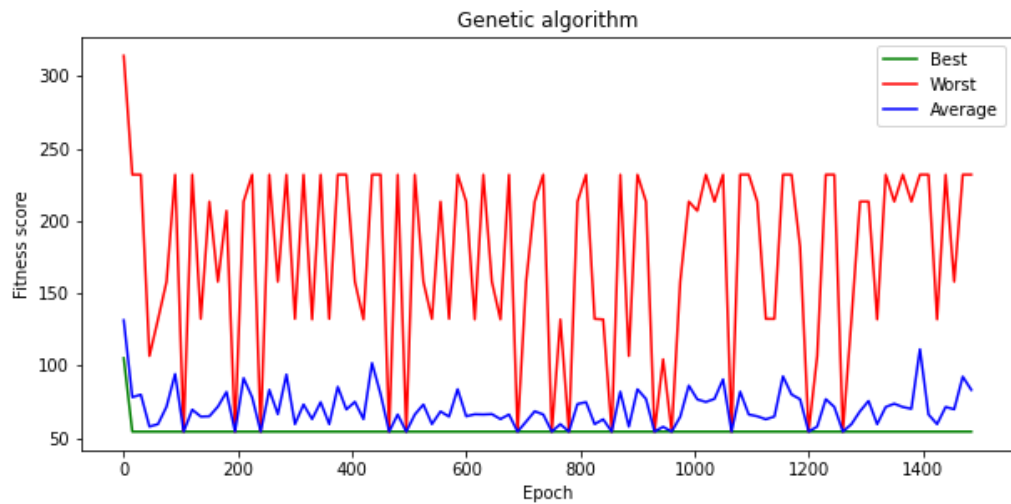
1. Badanie dla płytki zad0.txt



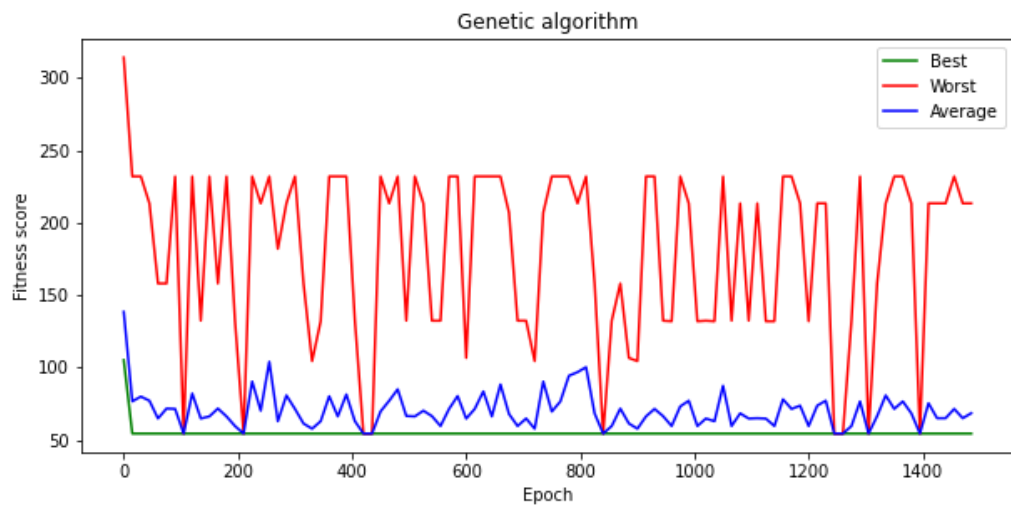
Rys. 13 zad0.txt, Szansa krzyżowania: 0%



Rys. 14 zad0.txt, Szansa krzyżowania: 25%



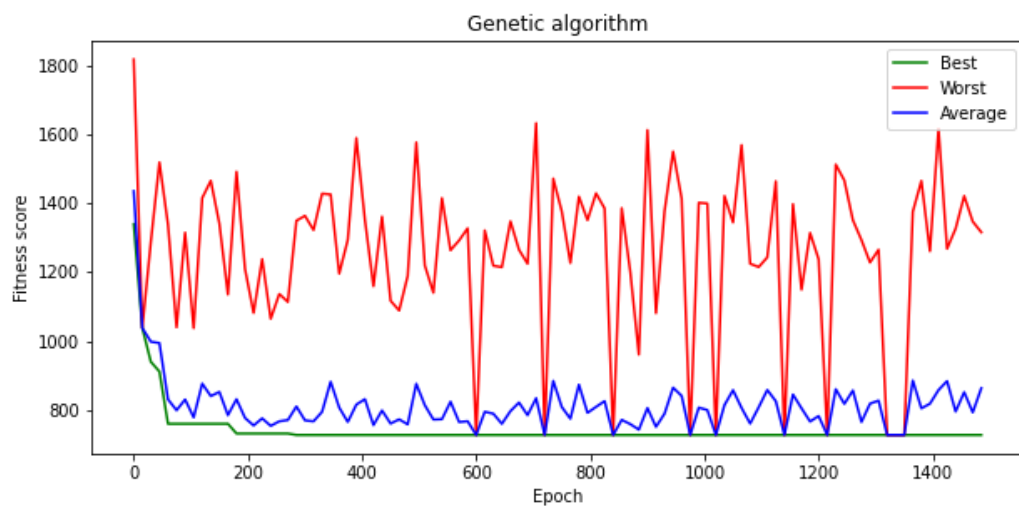
Rys. 15 zad0.txt, Szansa krzyżowania: 75%



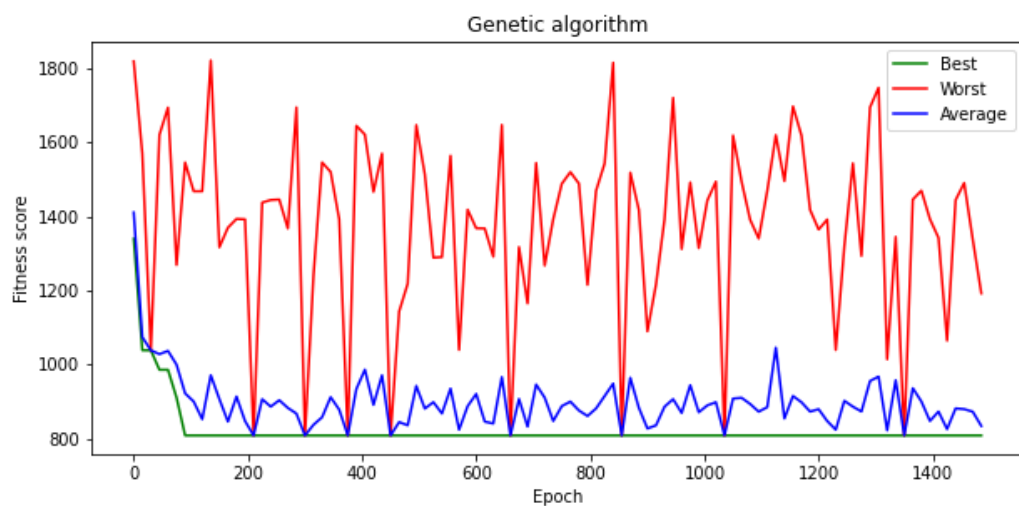
Rys. 16 zad0.txt, Szansa krzyżowania: 100%

* ta płytka nie jest dobrym przykładem dla danego badania, bo ma za mało segmentów do mutacji i krzyżowania.

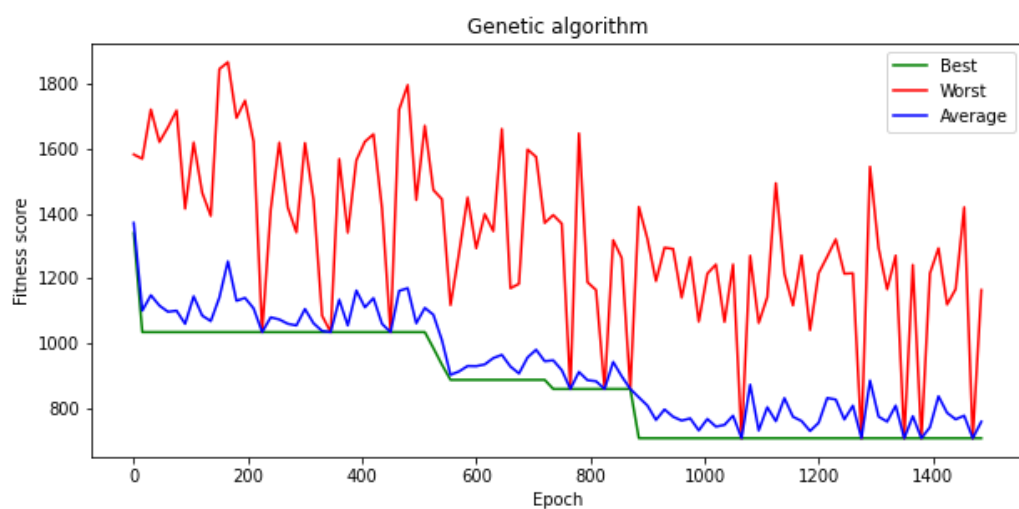
2. Badanie na płytce zad1.txt



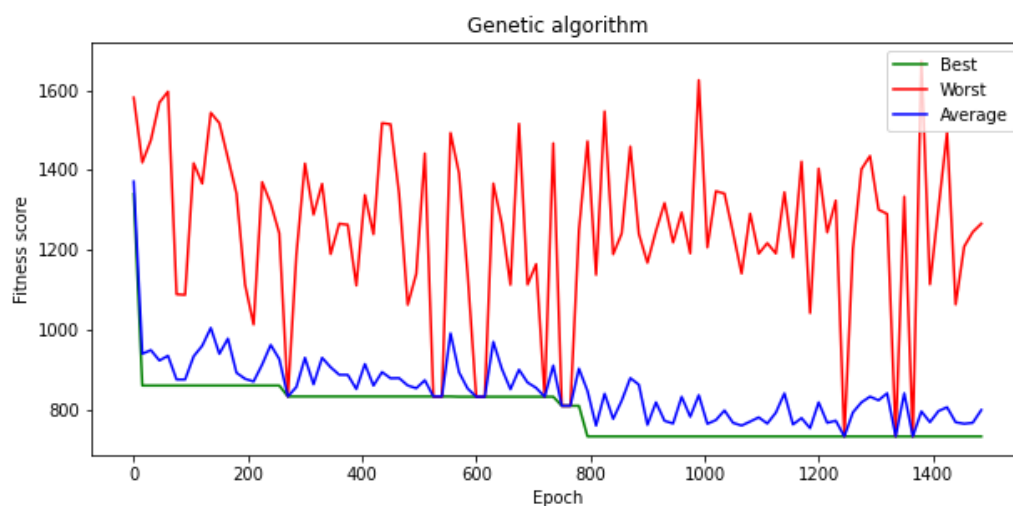
Rys. 17 zad1.txt, Szansa krzyżowania: 0%



Rys. 18 zad1.txt, Szansa krzyżowania: 25%

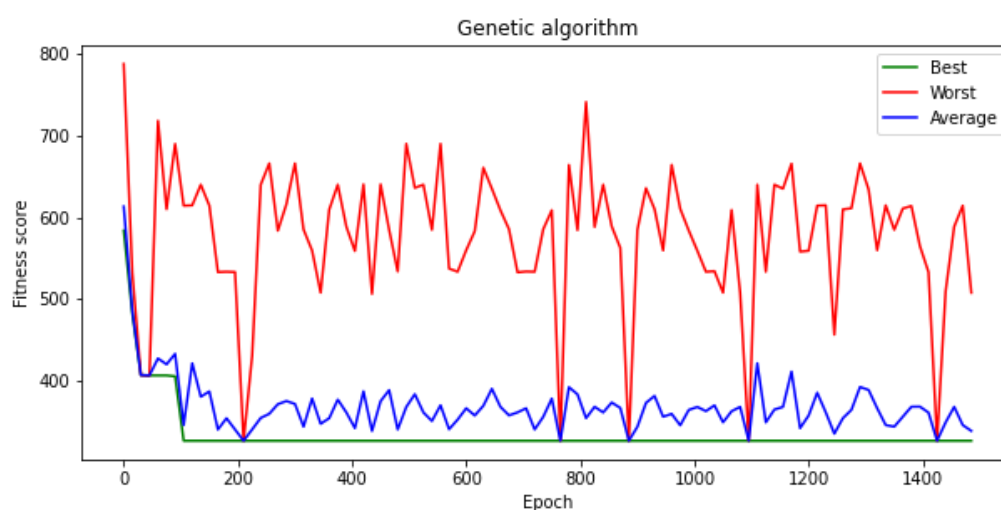


Rys. 19 zad1.txt, Szansa krzyżowania: 75%

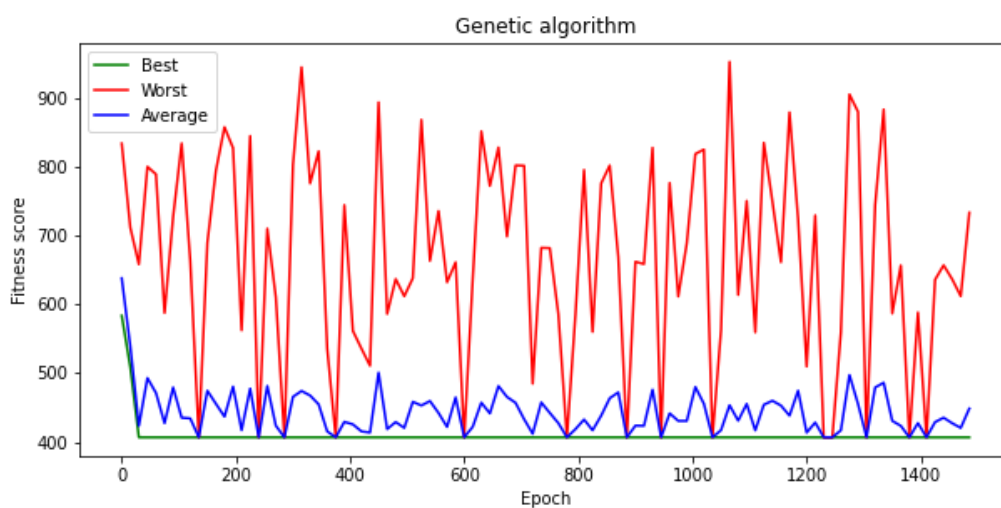


Rys. 20 zad1.txt, Szansa krzyżowania: 100%

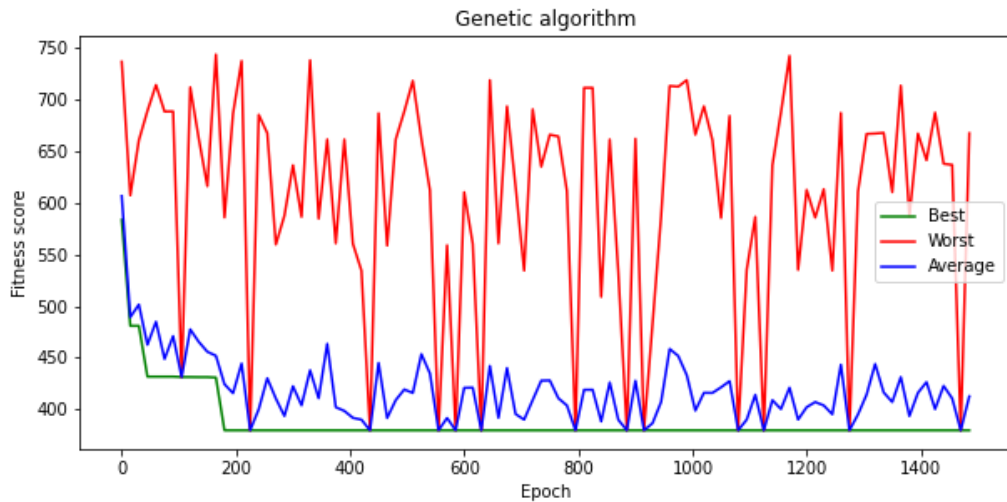
3. Badania na płycie zad2.txt



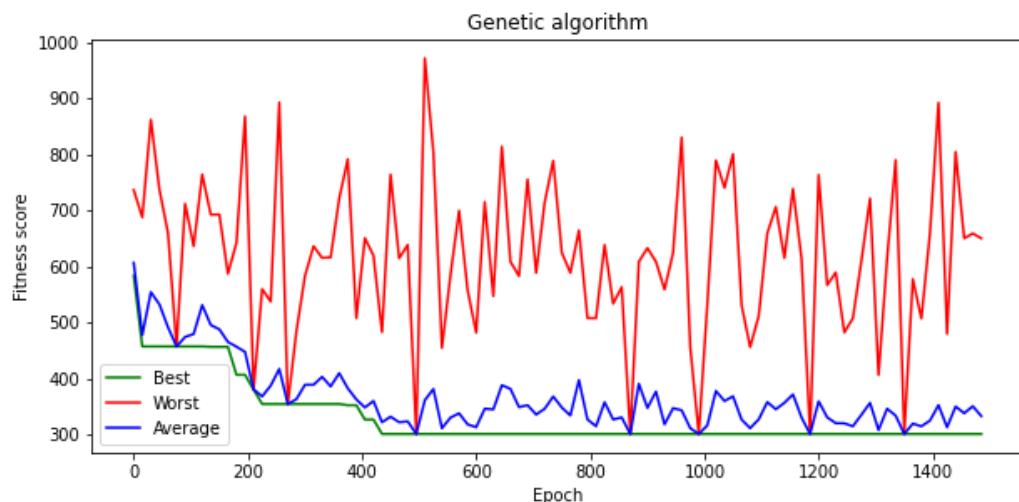
Rys. 21 zad2.txt, Szansa krzyżowania: 0%



Rys. 22 zad2.txt, Szansa krzyżowania: 25%



Rys. 23 zad2.txt, Szansa krzyżowania: 75%



Rys. 24 zad2.txt, Szansa krzyżowania: 100%

Wniosek

Jak widać z wyników, przy małym prawdopodobieństwie krzyżowania ocena nie spiesz się zmniejszać. Przy szansie krzyżowania równym 0 ocena osobnika będzie zależała tylko od mutacji. Zajmie to trochę czasu do losowego wyboru potrzebnej prawidłowej ścieżki.

Przy szansie krzyżowania większym o 50% jest większa szansa, że osobniki z nieprawidłowymi ścieżkami będą skrzyżowane z osobnikami, które mają część prawidłowych.

Ależ, gdy zostawimy szansę krzyżowania o 100%, będzie to dla nas gorzej, bo wtedy osobniki nie będą mieli szansę zostać się do następnych populacji. Może się wydarzyć tak, że jakiś osobnik był już

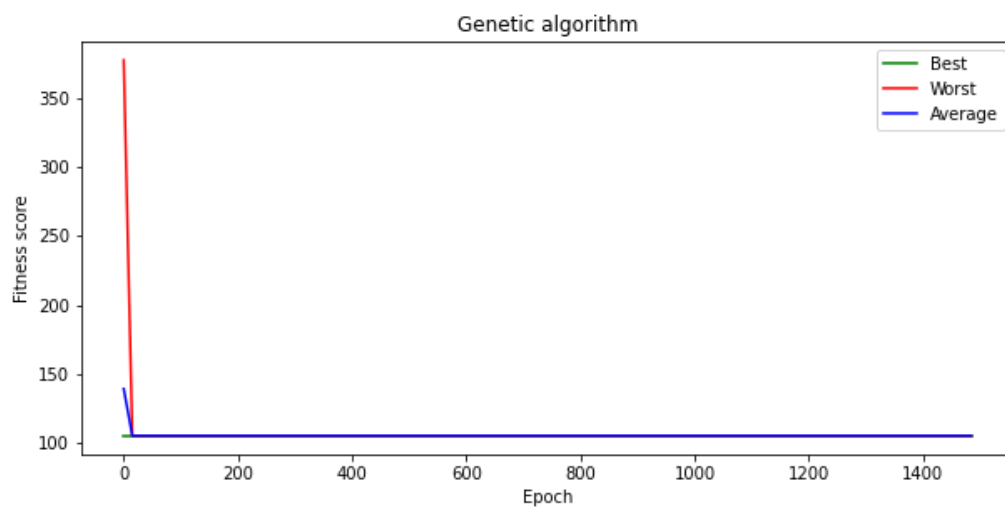
dobrze natrenowany, ale został skrzyżowany z innym i tym samym zmniejszył szanse znalezienia prawidłowych (dobrych) osobników.

Mutacja

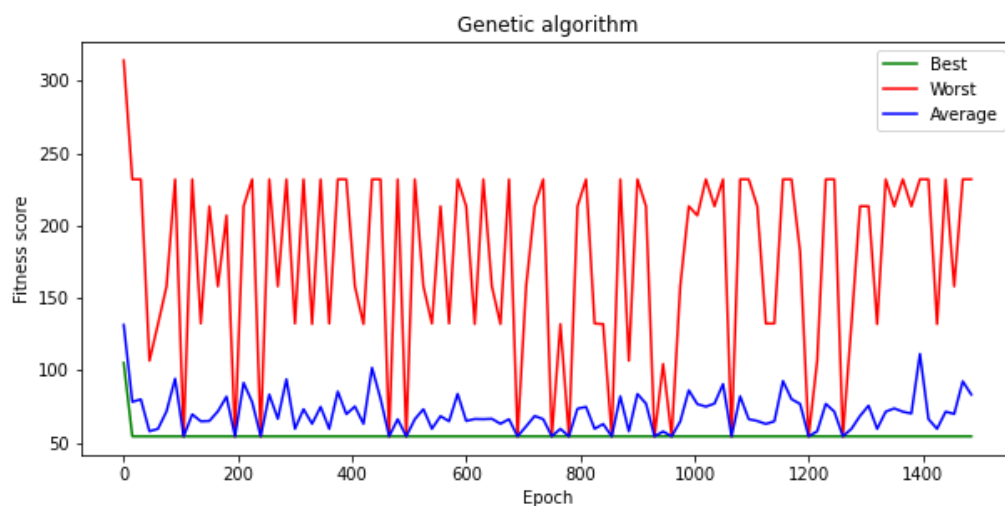
Zostawiamy wszystkie parametry podobne do poprzednich. Ustawiamy szansę krzyżowania o 75%. Teraz zbadajmy szansę mutacji.

Będziemy testowali takie znaczenia: **0%, 14%, 50%, 100%**.

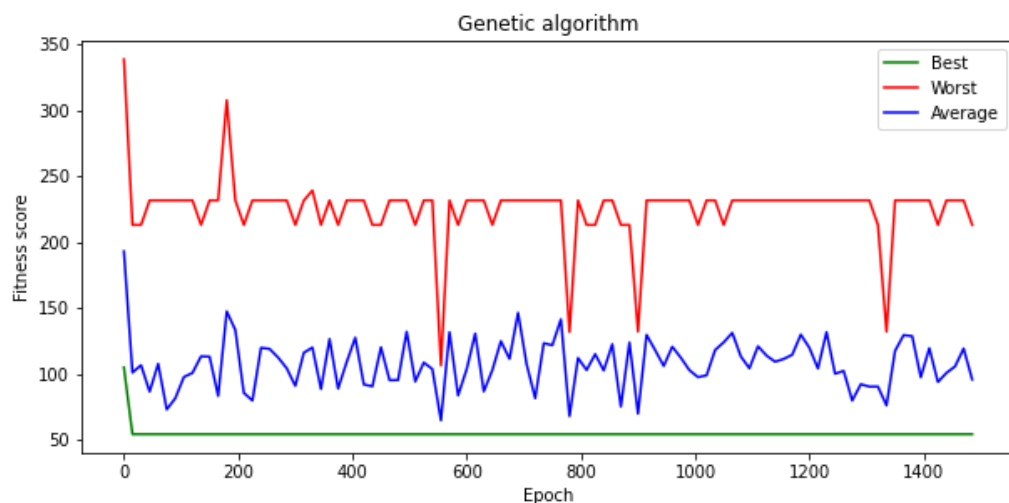
1. Badanie płytki zad0.txt



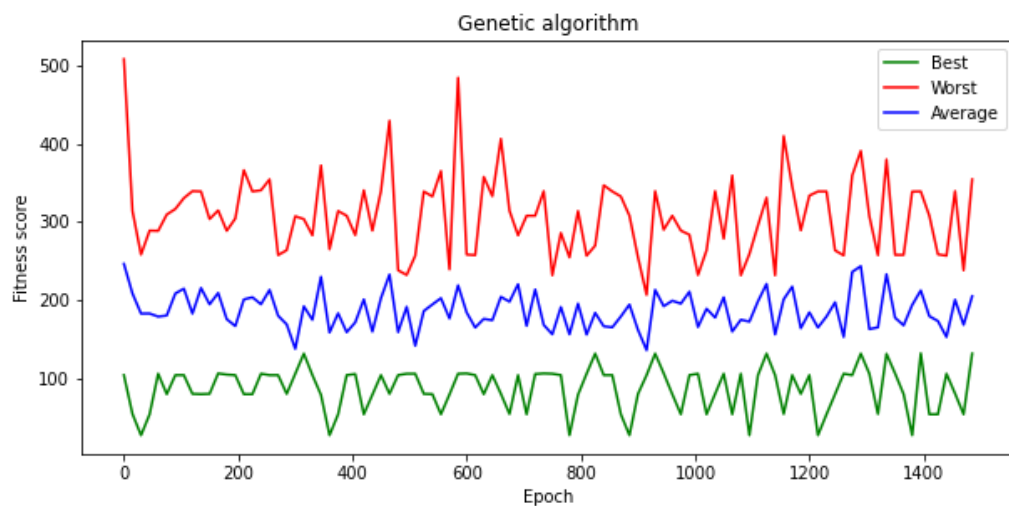
Rys. 25 zad0.txt – Szansa mutacji: 0%



Rys. 26 zad0.txt – Szansa mutacji: 14%

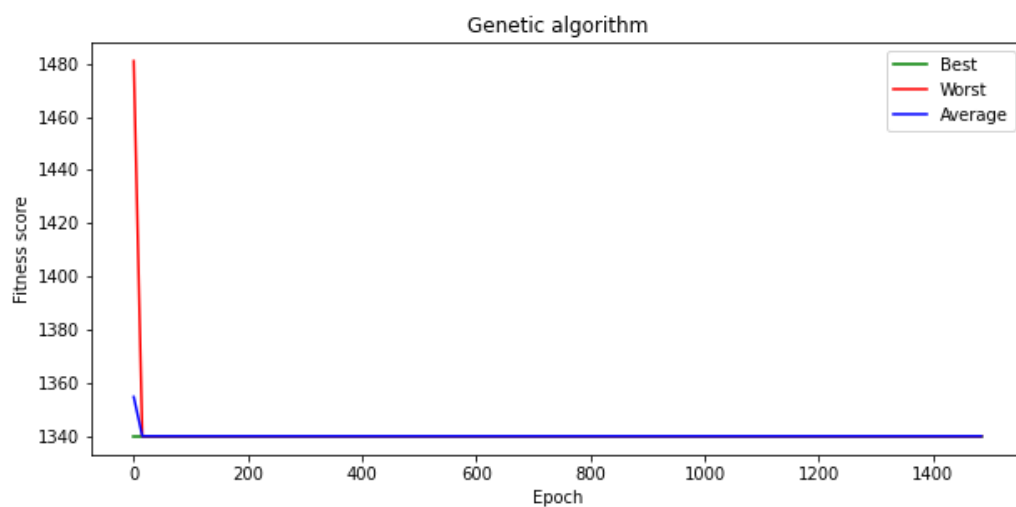


Rys. 27 zad0.txt - Szansa mutacji: 50%

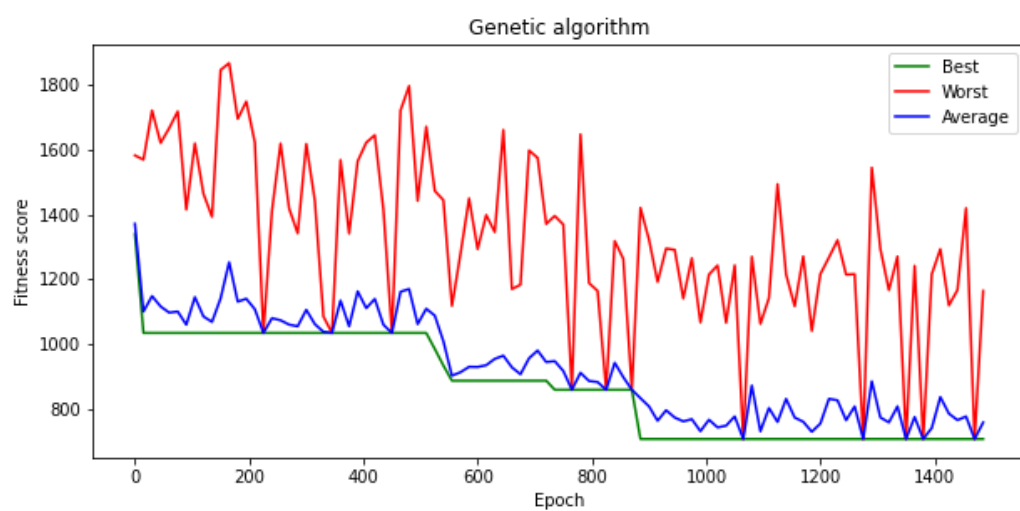


Rys. 28 zad0.txt – Szansa mutacji: 100%

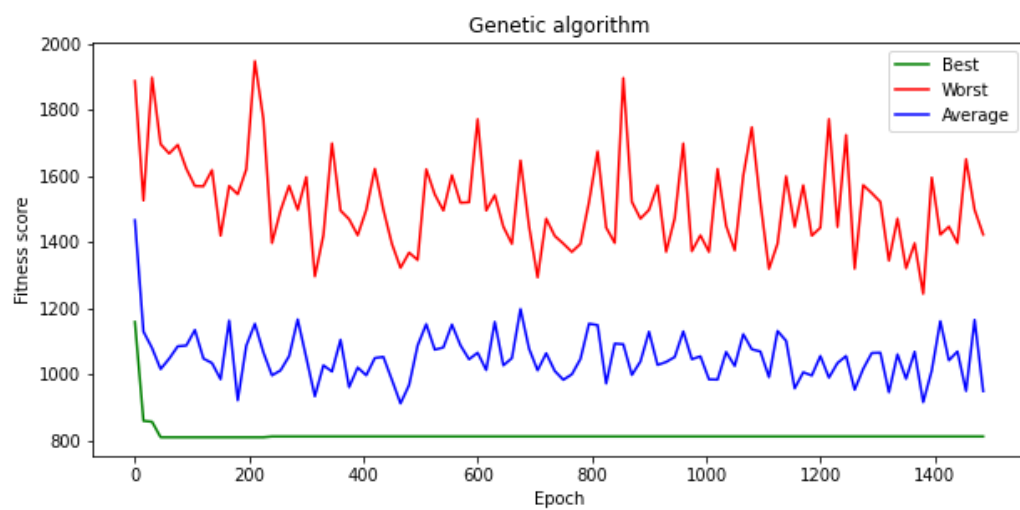
2. Badanie płytki zad1.txt



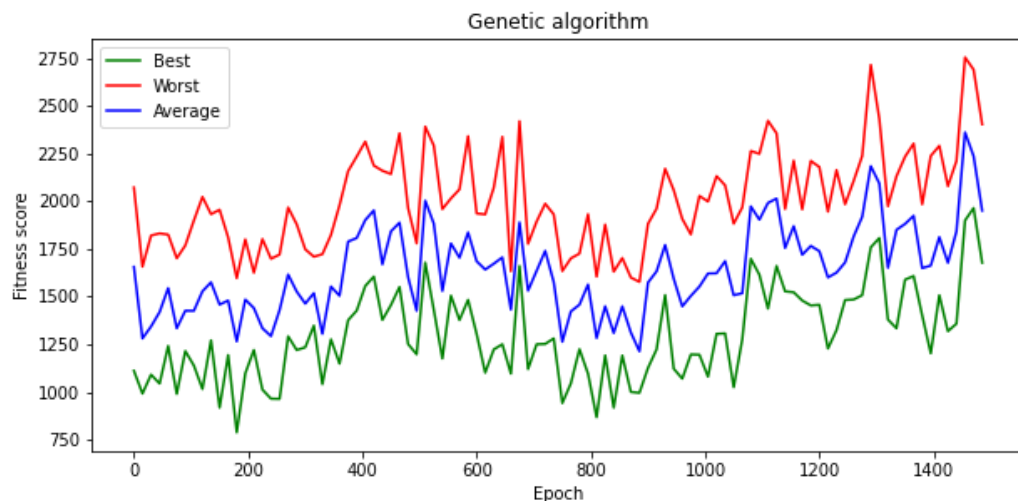
Rys. 29 zad1.txt – Szansa mutacji: 0%



Rys. 30 zad1.txt – Szansa mutacji: 14%



Rys. 31 zad1.txt – Szansa mutacji: 50%



Rys. 32 zad1.txt – Szansa mutacji: 100%

Wniosek

Więc, dla analizy wystarczy nam dwóch płytek. Od razu powiem, że przy szansie mutacji równemu 0, mamy powtarzającą się ocenę dla każdego pokolenia. Wynika to z tego, że nie wnosimy żadnych zmian do segmentów ścieżki. Widać to na rysunkach 29 i 25.

Dla szansy mutacji równej 100%, algorytm może zacząć generować zbyt nieprawidłowe dane lub nie mieć żadnego wpływu. Zwiększenie oceny powoduje ten fakt, że osobnik każde nowe pokolenie będzie zmieniał segmenty ścieżki. Przy tym, może wydarzyć się tak, że te zmiany będą gorsze od poprzednich, a więc stracimy dobrego osobnika.

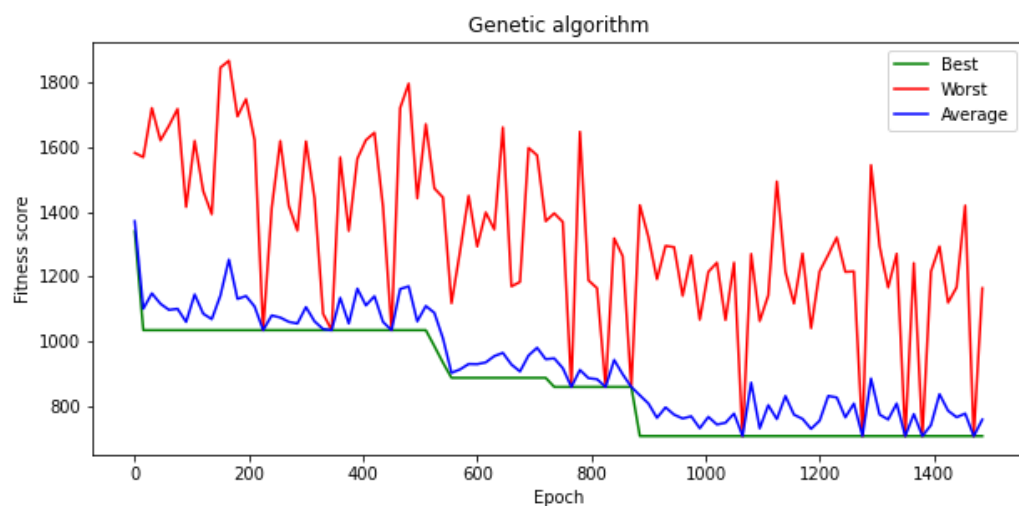
Zbyt duża częstotliwość mutacji też może powodować takie przechowywanie algorytmu, że osobniki będą zmieniony zbyt dużo i potrzebne będzie znacznie więcej czasu, żeby dojść do sensownego wyniku.

Przy szansie mutacji około 10-15%, algorytm krokowo dochodzi do najlepszych wyników, przez to, że nie zmieniamy segmenty zbyt dużo razy – nie tracimy ewentualnie dobrych osobników. Także szansa mutacji zależy od rozmiaru płytek. Jeżeli płytka jest bardzo mała, to jest sens w zmniejszeniu szansy i odwrotnie.

Rozmiar populacji

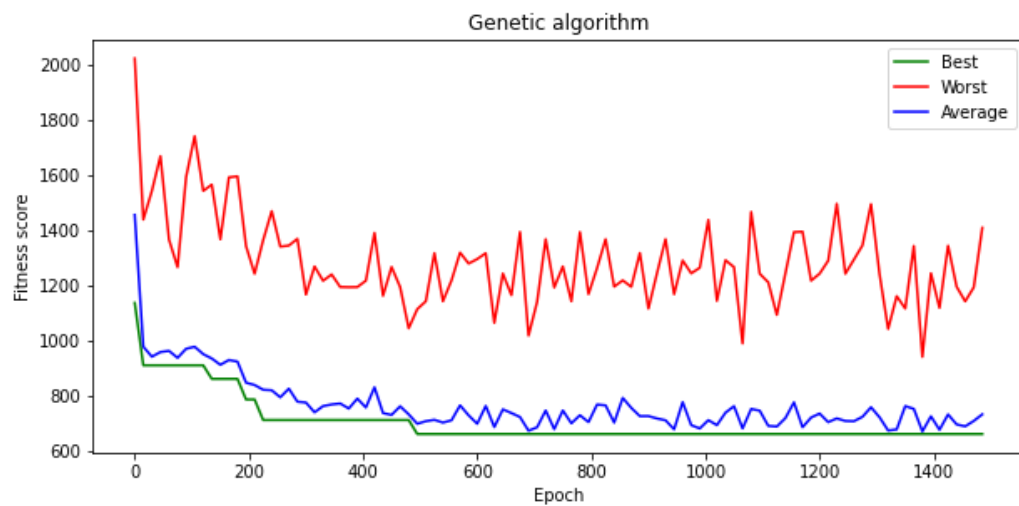
Zbadajmy dla płytki **zad1.txt** rozmiar populacji. Czy przyspieszy to trening czy nie. Teraz ustawimy szansę mutacji o 14%. I wybierzmy selekcję typu turniej o wyborze w 5 osobników.

Dla początku zaczniemy o populacji o rozmiarze **15**.



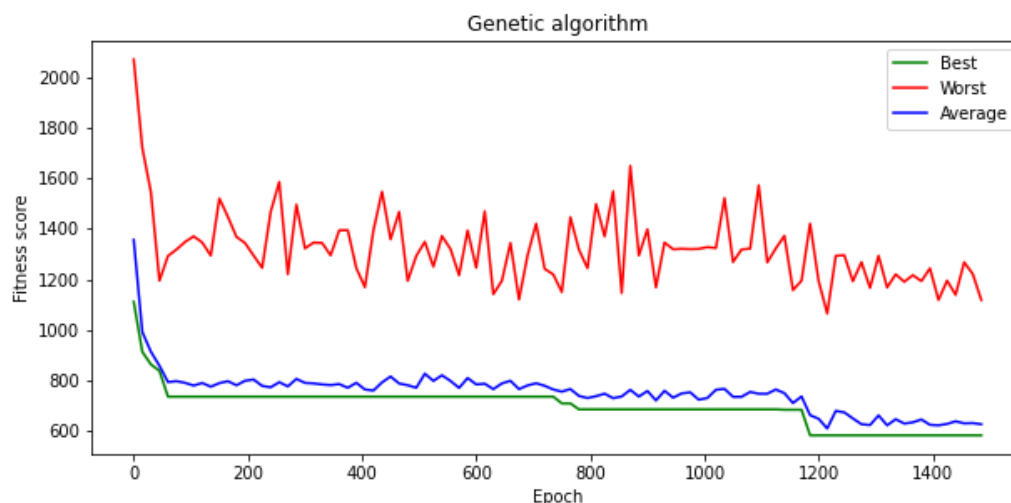
Rys. 33 zad1.txt – 15 osobników w populacji

A teraz zbadajmy dla populacji o rozmiarze **30**.



Rys. 34 zad1.txt – 30 osobników w populacji

Zwiększymy tą liczbę do **100**.



Rys. 35 zad1.txt – 100 osobników w populacji

Możemy uruchomić algorytm dla liczby pokoleń równej 10 i zobaczyć wyniki w tabeli:

Instancja	Alg. Elowucyjny [10x]				Metoda losowa [N]			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
zad1.txt	1140	1674	1229	167	1730	3081	2344	388
zad2.txt	724	1121	782	124	686	1386	1090	216
zad3.txt	2040	2695	2093	167	1733	3775	2791	579

Tab 1. Wyniki z populacją o rozmiarze 15

Instancja	Alg. Elowucyjny [10x]				Metoda losowa [N]			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
zad1.txt	1306	1482	1344	39	1445	2934	2185	373
zad2.txt	507	777	538	71	665	1710	1143	264
zad3.txt	1375	2159	1471	223	1955	3661	2883	466

Tab 2. Wyniki z populacją o rozmiarze 30

Instancja	Alg. Elowucyjny [10x]				Metoda losowa [N]			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
zad1.txt	1016	1874	1118	164	1457	3260	2233	320
zad2.txt	380	699	408	74	533	1682	1086	210
zad3.txt	1289	2127	1342	172	2234	3856	2898	381

Tab 3. Wyniki z populacją o rozmiarze 100

Wyniki także zależą od metody generacji losowych osobników. Możemy zmienić ilość generowanych losowo segmentów i dostać inne wyniki.

Jak widać z rysunków większa populacja daje więcej osobników z minimalną oceną. Ale też to zmniejsza szanse na wybór dobrych osobników w selekcji turniejowej. Dlatego warto wybierać populację o nie bardzo dużej i bardzo małej ilości osobników.

Wagi oceny

Dla wag oceny warto zaznaczyć, że największe znaczenia muszą mieć przecięcia i liczby obliczania ilości ścieżek poza obszarem płytki.

Także ważnym momentem jest to, że w niektórych płytka warto ustawić wagę o wartości 0 przy obliczeniu długości całkowitej ścieżek. Bo może się wydarzyć tak, że algorytm znajdzie w minimum lokalny i nie będzie obniżał ocenę. Dlatego wagę tą warto ustawiać o 10-20 razy mniej, niż inne.

Dla przykładu waga długości ścieżek ma 4, a wartości przecięć 15. Po obliczeniu oceny wydarza się tak, że $4 * 5 > 15 * 1$, wtedy najlepszym rozwiązaniem będzie te, które ma przecięcie. A to nie jest możliwe i nie jest prawidłowo.

Także warto zaznaczyć, że wartości takie jak: długość sumaryczna ścieżek, długość sumaryczna ścieżek poza obszarem płytki muszą mieć znaczenia o kilku razy mniejsze od innych. Dlatego, że średnia długość ścieżek może być nawet 30-40 jednostek, a ilość przecięć lub ilość segmentów o znacznie wielu razy mniej. I spowoduje to problem, gdzie ścieżki będą krótsze, ale będą mieli wiele przecięć i td.

Rozmiar turnieju

Rozmiar turnieju ma znaczniejszy wpływ, gdy typem selekcji jest wybrana selekcja turniejowa. Wtedy przy wybieraniu losowych, przypuśćmy, pięciu osobników z 25, jest większe prawdopodobieństwo wybrania z gorszą oceną. Co za kilki pokoleń da możliwość niepopadania w minimum lokalny.

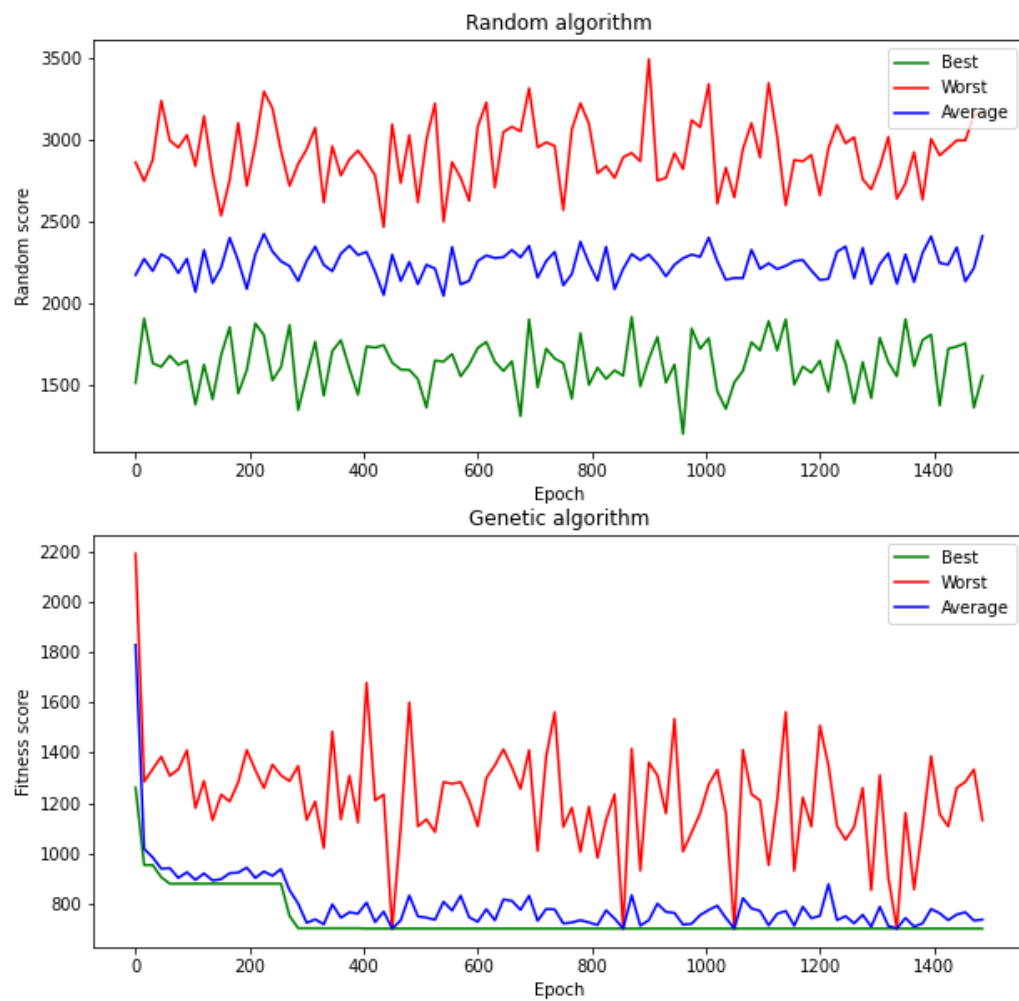
Porównanie algorytmu genetycznego z metodą losową

Weźmiemy dla przykładu płytkę **zad1.txt**, **zad2.txt**, **zad3.txt**.

Parametry są ustalone takie:

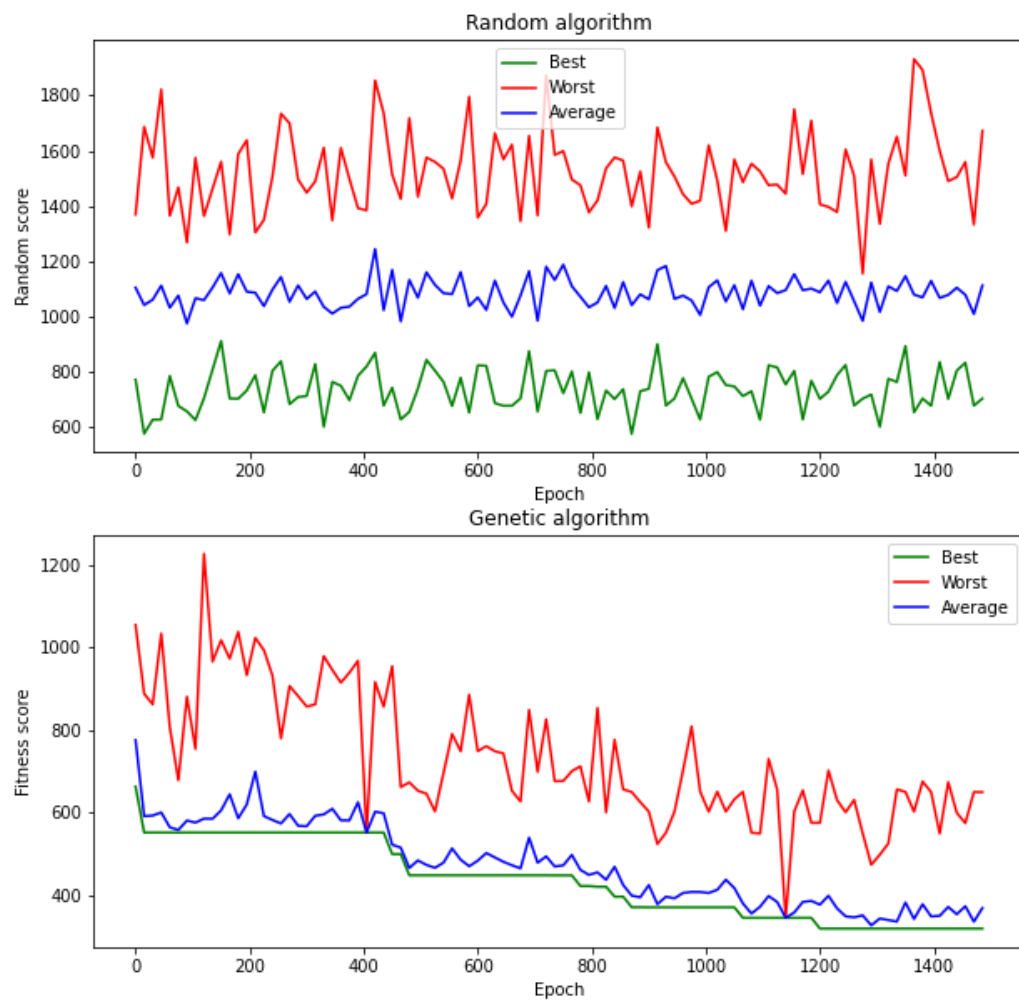
- Selekcja typu: **ruletka**
- Rozmiar populacji: **20**
- Szansa krzyżowania: **75%**
- Szansa mutacji: **14%**
- Rozmiar turnieju: **5**
- Liczba pokoleń: **1500**
- Generowanie losowych segmentów na początku: **4 +- 2**

1. Badanie dla **zad1.txt**



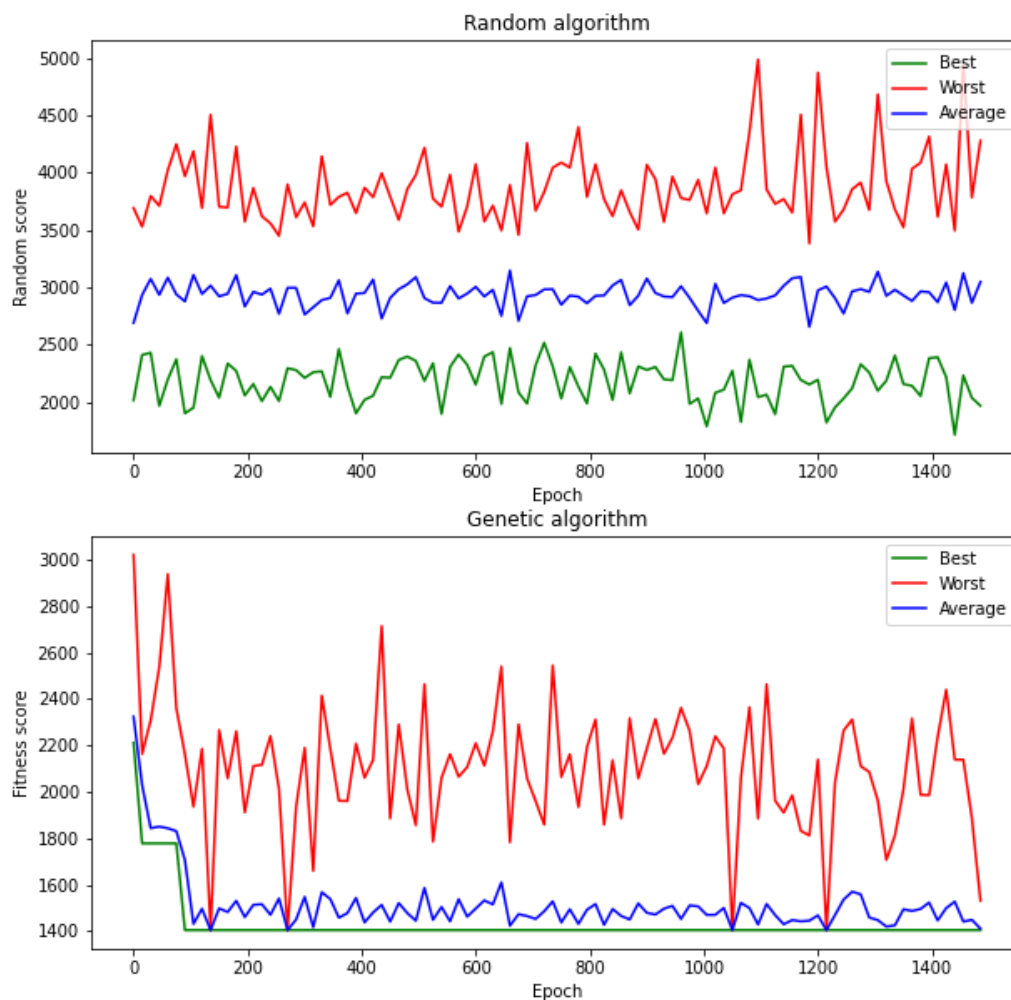
Rys. 35 zad1.txt – Porównanie metody losowej i algorytmu genetycznego

2. Badanie dla zad2.txt



Rys. 36 zad2.txt – Porównanie metody losowej i algorytmu genetycznego

3. Badanie dla zad3.txt



Rys. 37 zad3.txt – Porównanie metody losowej i algorytmu genetycznego

Instancja	Alg. Ewolucyjny [10x]				Metoda losowa [N]			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
zad1.txt	1350	1561	1361	47	1399	3007	2224	477
zad2.txt	638	915	671	84	610	1550	1069	217
zad3.txt	1803	2538	1857	178	2248	3613	2798	366

Wynik

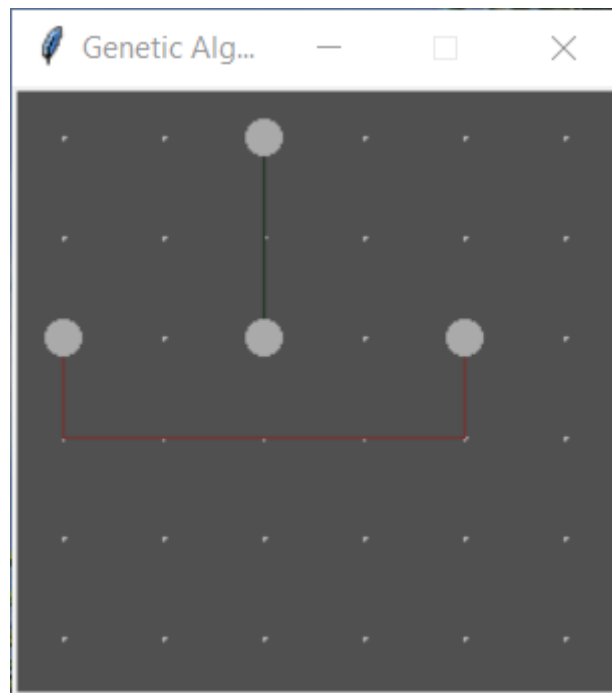
Jak widać z obrazków wyżej, metoda losowa generuje osobników, oceny których jest w pewnym przedziale. Ocena najlepszego osobnika nie dąży do 0, przez to, że nie robimy żadnych zmian na segmentach, czyli nie mutujemy je.

Możemy z pewnością powiedzieć, że algorytm genetyczny lepiej dopasowuje osobników do problemu, niż metoda losowa.

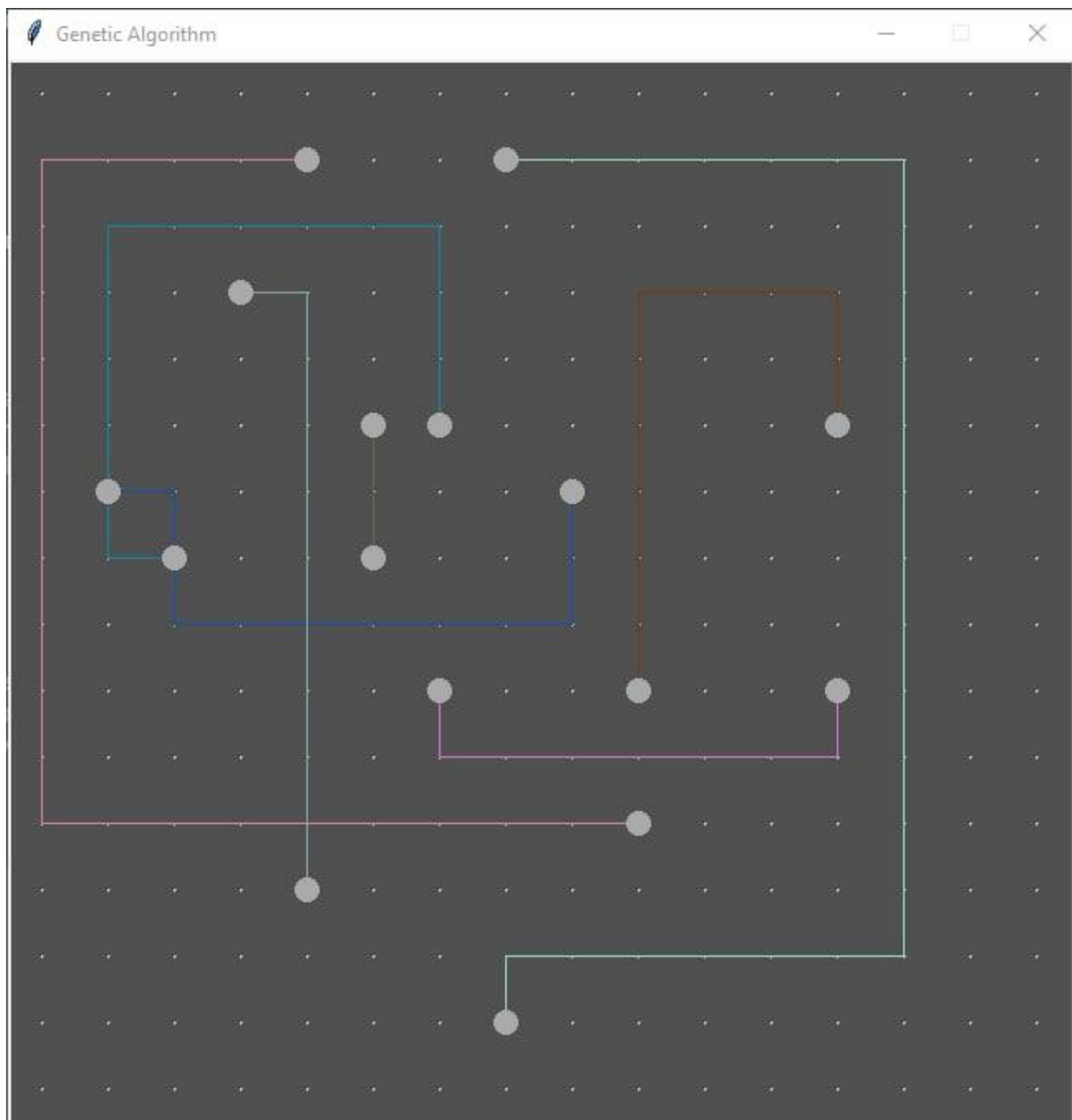
Najlepsze rozwiązania

Parametry:

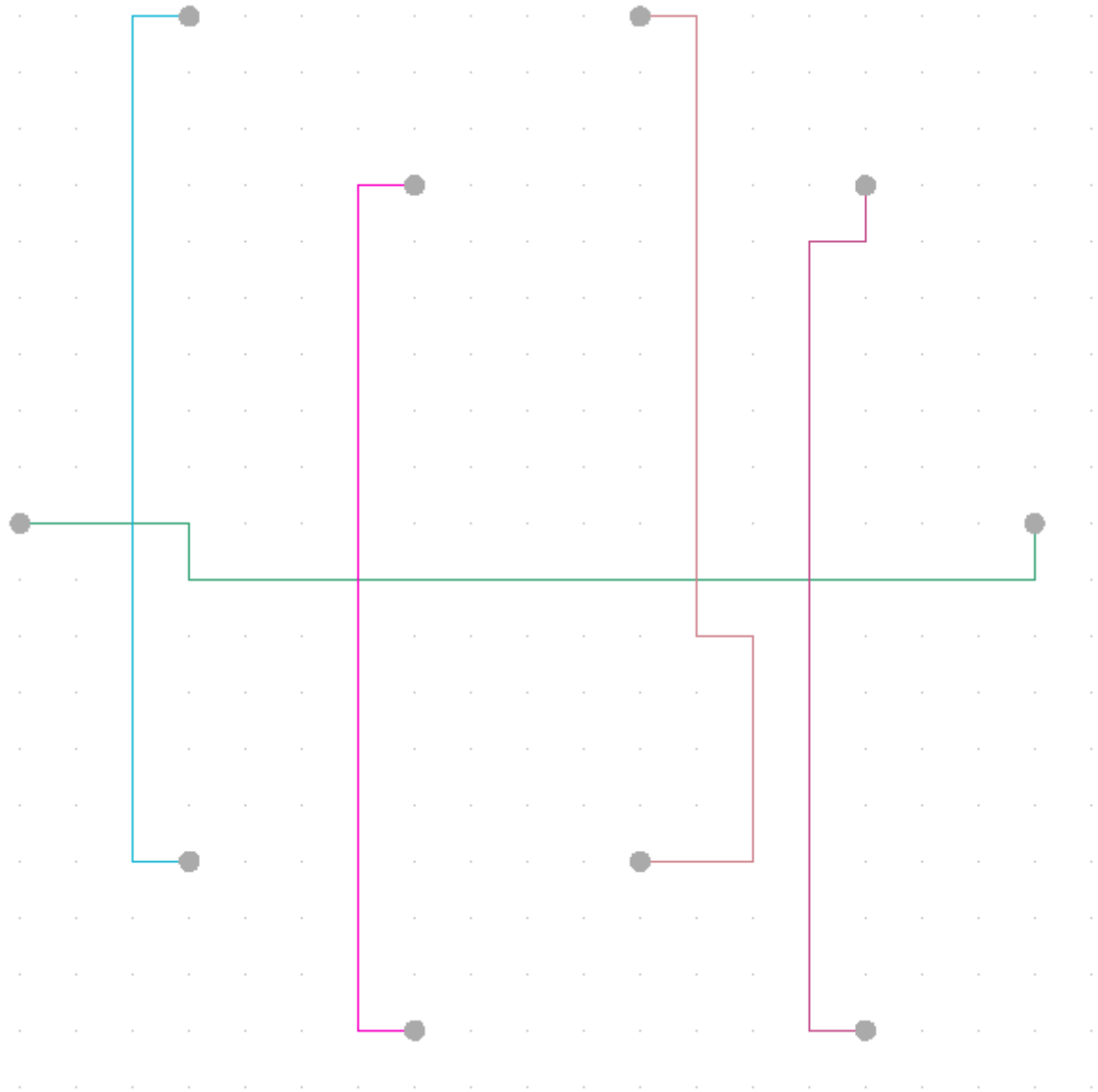
- Ziarno: **random.seed(100)**
- Rozmiar populacji: **20**
- Szansa krzyżowania: **70%**
- Szansa mutacji: **12%**
- Rozmiar selekcji: **5**
- Ilość pokoleń: **1600**
- Wagi: **[25, 0, 0.5, 15, 15]**
- Szansa mutacji metodą B: **25%**



Rys. 1 - Najlepsze rozwiązanie dla zad0.txt



Rys. 2 - Najlepsze rozwiązanie dla zad1.txt



Rys. 3 - Najlepsze rozwiązanie dla zad2.txt

