

Sprawozdanie

Sztuczna inteligencja i inżynieria wiedzy

Ćwiczenie 1: Algorytmy genetyczne

Prowadzący: Dr inż. Michał Przewoźniczek

Wykonał: Aleksander Górka

1. Wstęp

Podczas testów były brane wyniki z 10 pomiarów, z których zostały policzone wartości średnie dopasowań oraz ich odchylenia standardowe. Wszystkie testy zostały przeprowadzone dla ziarna (seed) równego 123456789.

Słowniczek:

popSize – wielkość pojedynczej generacji,

Px - parametr krzyżowania (0-1),

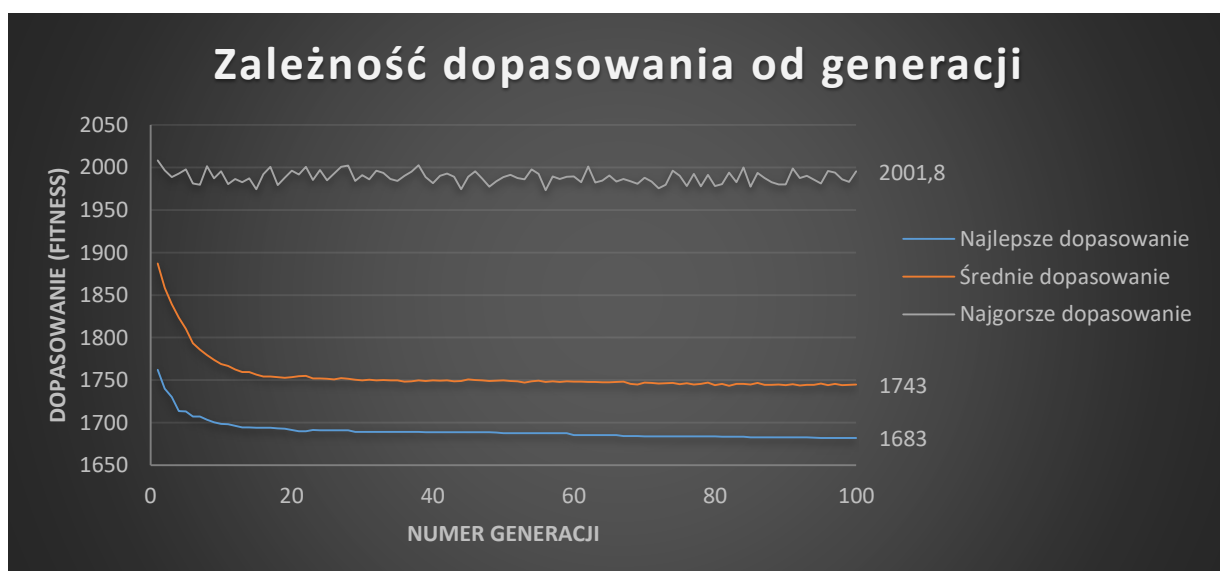
Pm – parametr mutacji(0-1),

Tour - ilość osobników wybieranych do turnieju (1-popSize).

2. Zbadanie poprawności działania algorytmu dla 5 plików testowych

Ilość generacji	popSize	Px	Pm	Tour	Selekcja
100	100	0,7	0,01	5	Turniej

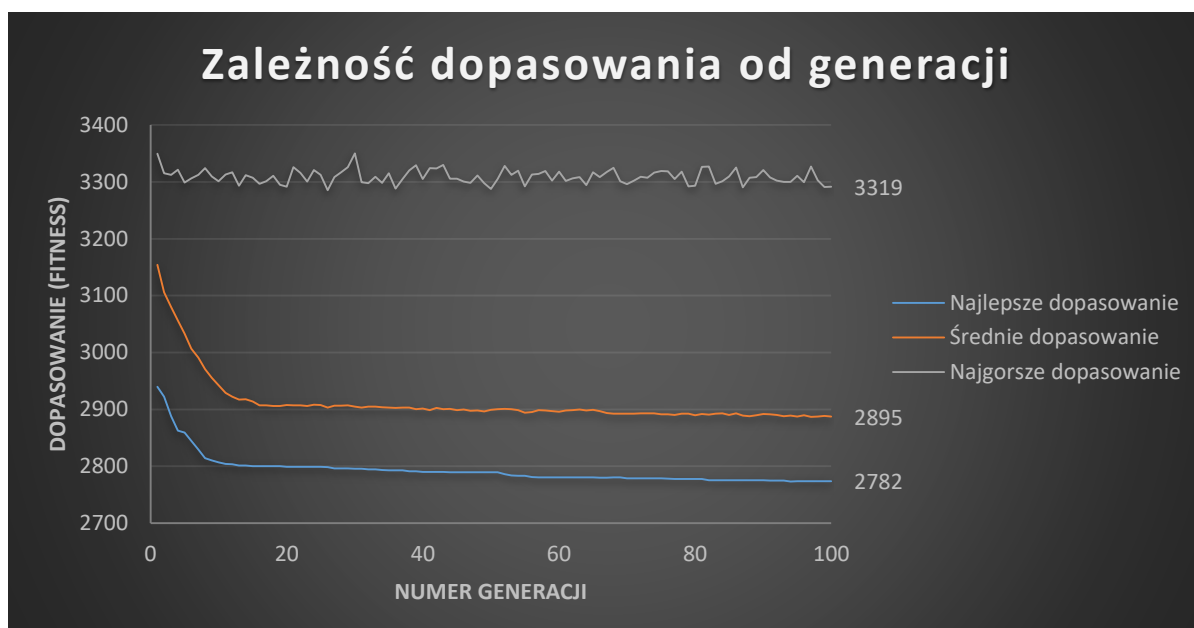
Tabela 1. Parametry algorytmu genetycznego, dla których zostały wykonane testy w bieżącym podpunkcie.



Wykres 1. Wyniki dla pliku „had12.dat” (najbardziej optymalna wartość: 1652).

Najlepsze dopasowanie	Średnie dopasowanie	Najgorsze dopasowanie
11,69	8,82	20,95

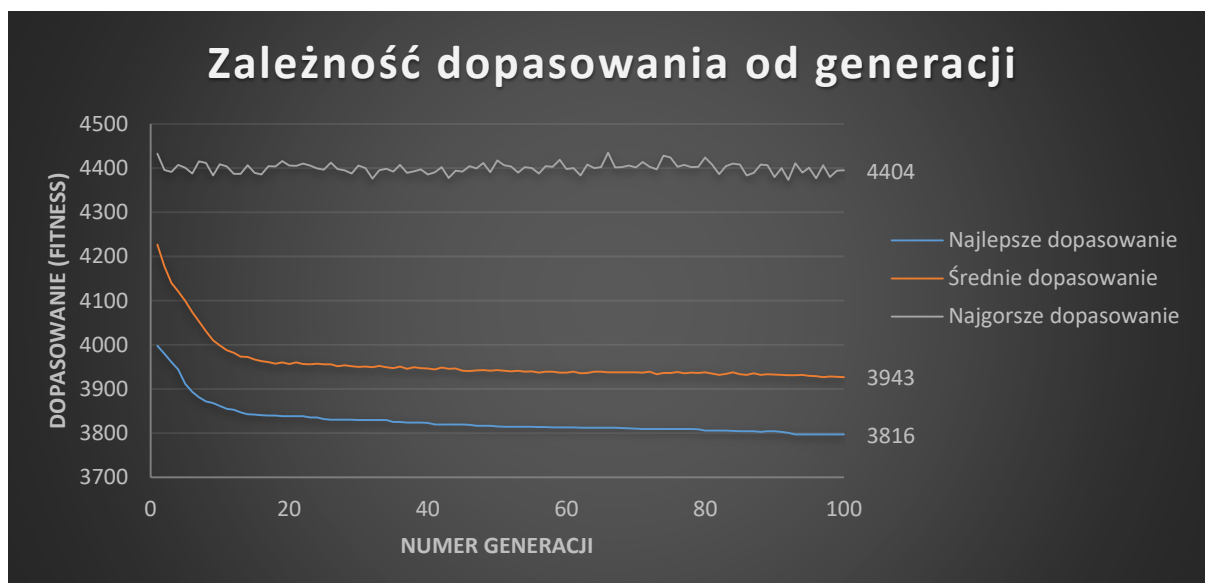
Tabela 2. Średnie odchylenie standardowe dla wykresu 1.



Wykres 2. Wyniki dla pliku „had14.dat” (najbardziej optymalna wartość: 2724).

Najlepsze dopasowanie	Średnie dopasowanie	Najgorsze dopasowanie
32,19	22,55	33,54

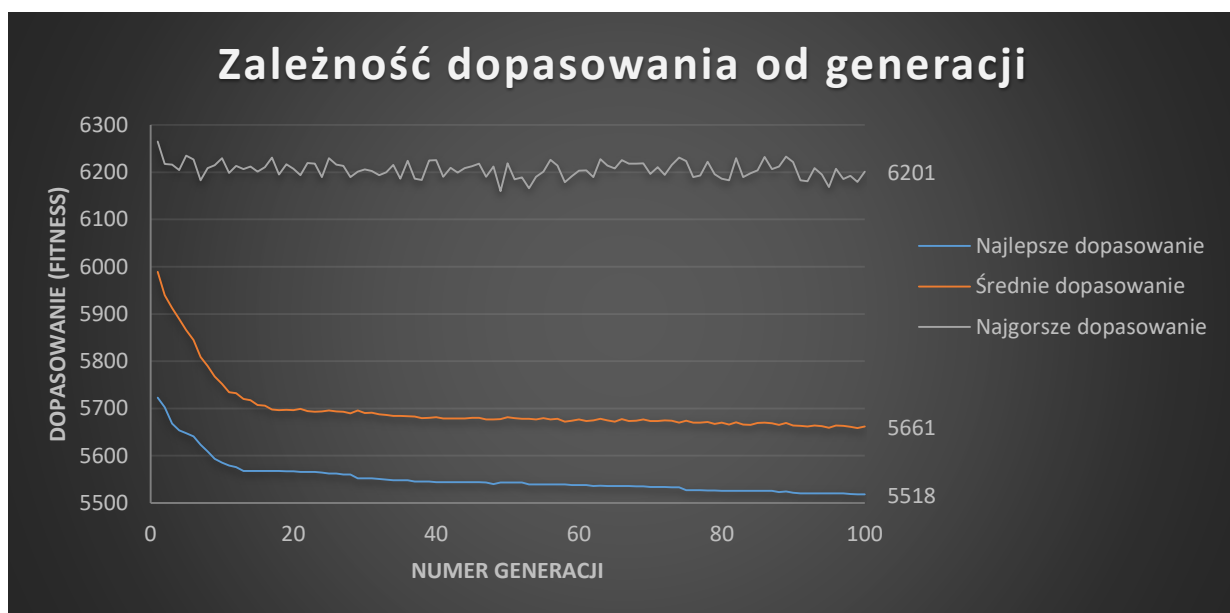
Tabela 3. Średnie odchylenie standardowe dla wykresu 2



Wykres 3. Wyniki dla pliku „had16.dat” (najbardziej optymalna wartość: 3720).

Najlepsze dopasowanie	Średnie dopasowanie	Najgorsze dopasowanie
52,55	36,92	36,45

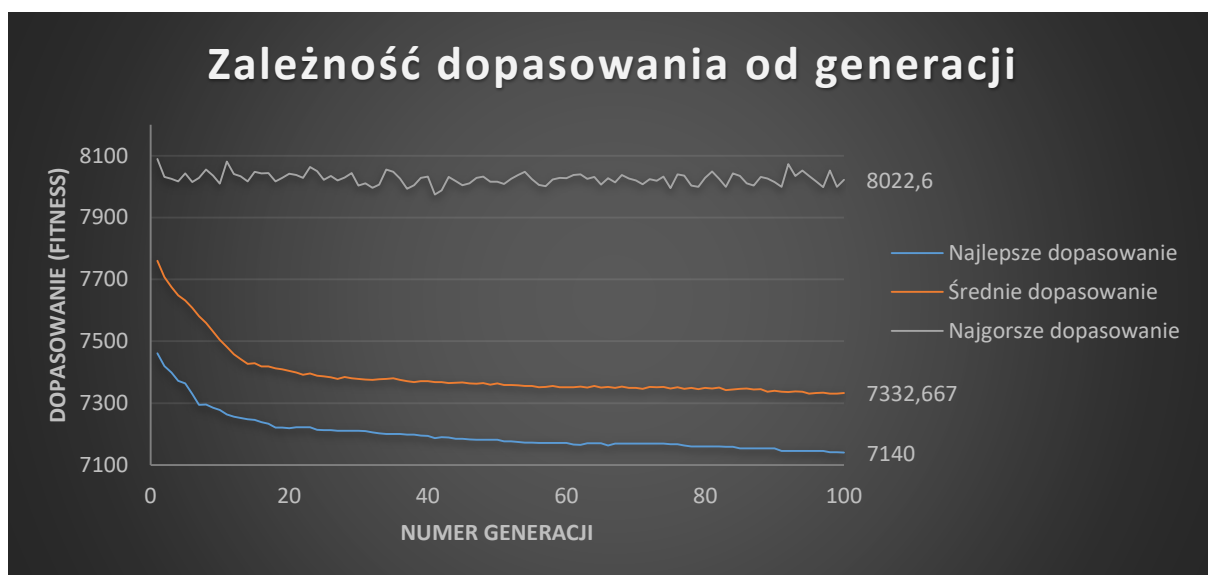
Tabela 4. Średnie odchylenie standardowe dla wykresu 3.



Wykres 4. Wyniki dla pliku „had18.dat” (najbardziej optymalna wartość: 5358).

Najlepsze dopasowanie	Średnie dopasowanie	Najgorsze dopasowanie
36,61	25,15	47,02

Tabela 5. Średnie odchylenie standardowe dla wykresu 4.



Wykres 5. Wyniki dla pliku „had20.dat” (najbardziej optymalna wartość: 6922).

Najlepsze dopasowanie	Średnie dopasowanie	Najgorsze dopasowanie
62,27	43,93	57,47

Tabela 6. Średnie odchylenie standardowe dla wykresu 5.

Wnioski:

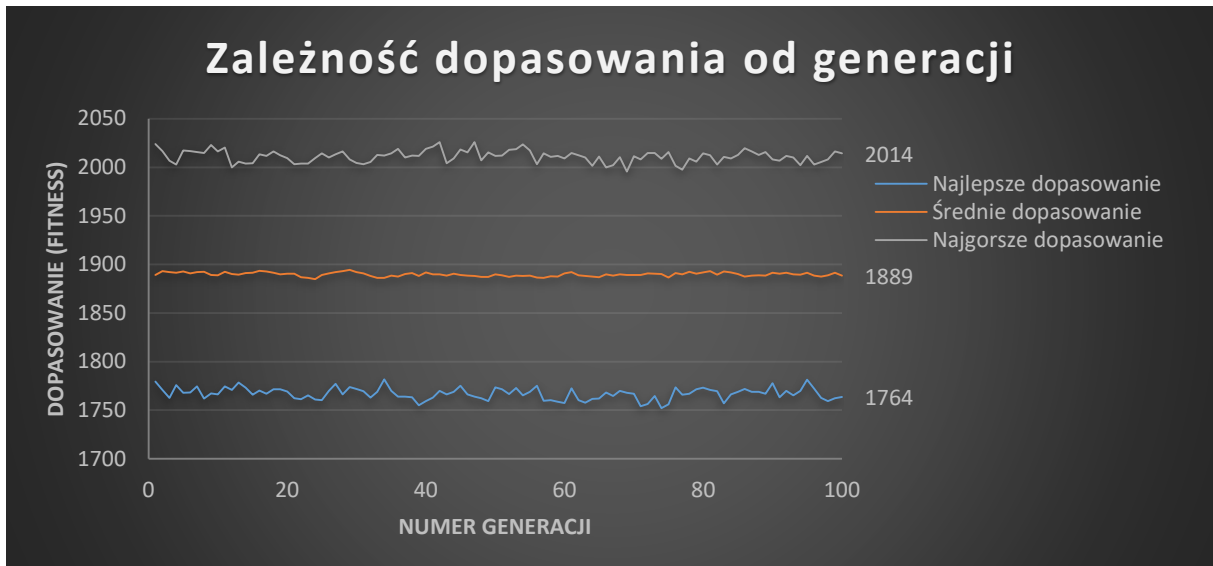
Otrzymane wyniki są bliskie najoptymalniejszego rozwiązania. Ponadto z wykresów można odczytać informację o tym, że z każdą kolejną generacją ilość „lepszych” osobników jest coraz większa. Zatem można stwierdzić, że napisany algorytm działa poprawnie.

3. Wpływ selekcji na efektywność algorytmu genetycznego

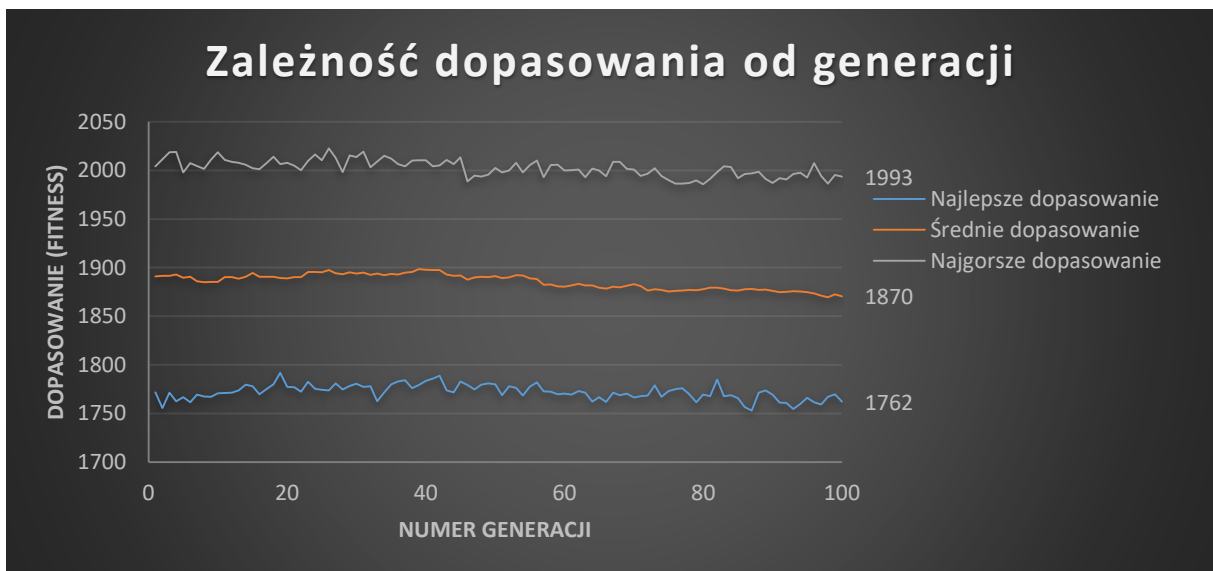
a) Ruletka standardowa

Tabela 7. Parametry algorytmu genetycznego, dla którego został wykonany test z bieżącego podpunktu.

Ilość generacji	popSize	Px	Pm	Selekcja
100	100	0,7	0,01	Ruletka



Wykres 6. Wyniki dla pliku „had12.dat” dla $P_x = 0,7$.



Wykres 7. Wyniki dla pliku „had12.dat” dla $P_x = 0,99$.

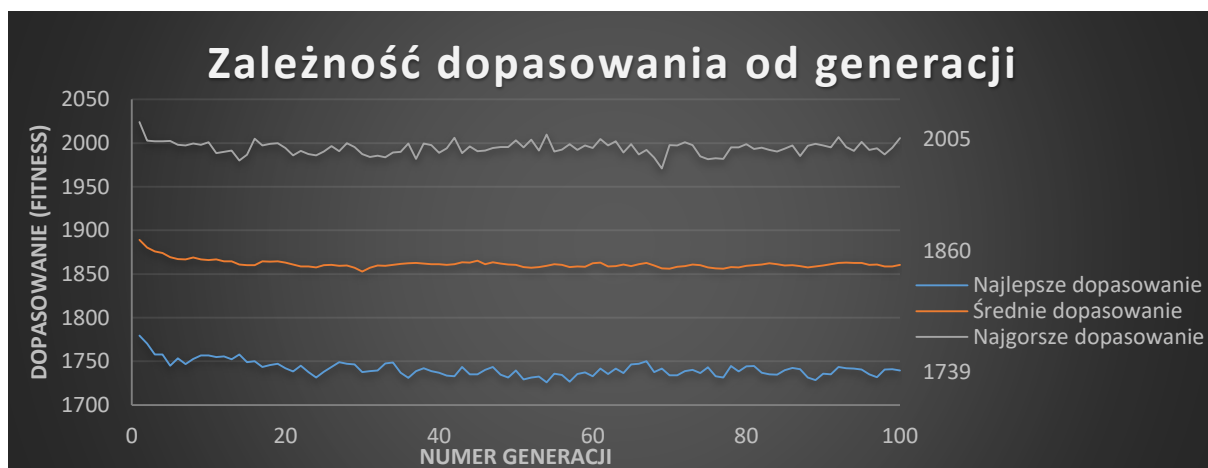
Wnioski:

Różnice między poszczególnymi dopasowaniami są zbyt małe, dlatego nawet ekstremalnie wysokie prawdopodobieństwo krzyżowania nie pomaga przetrwać najlepszym osobnikom.

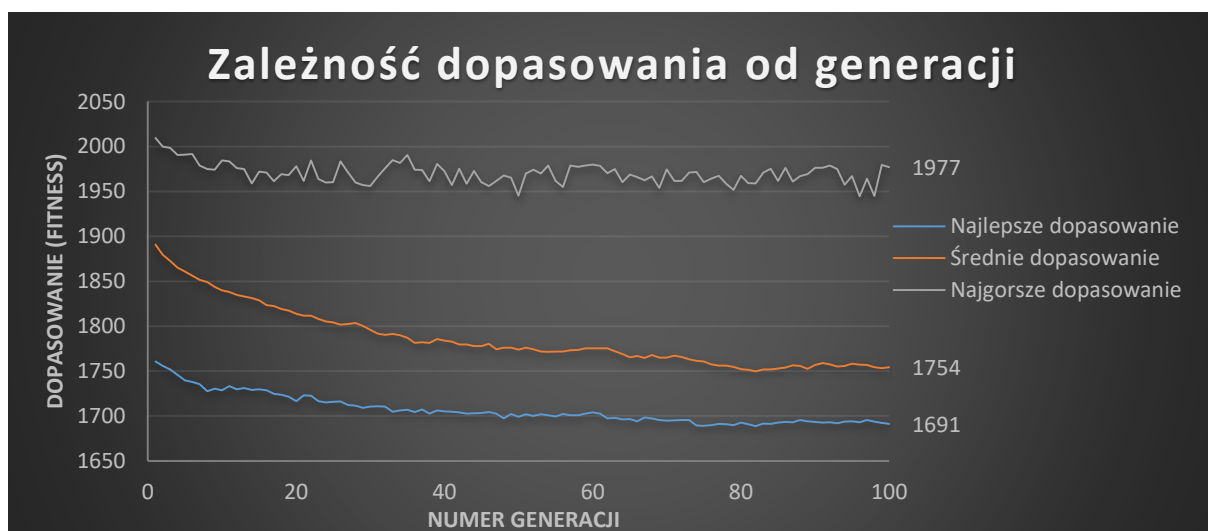
b) Ruletka zoptymalizowana

Ilość generacji	popSize	Pm	Selekcja
100	100	0,01	Ruletka zoptymalizowana

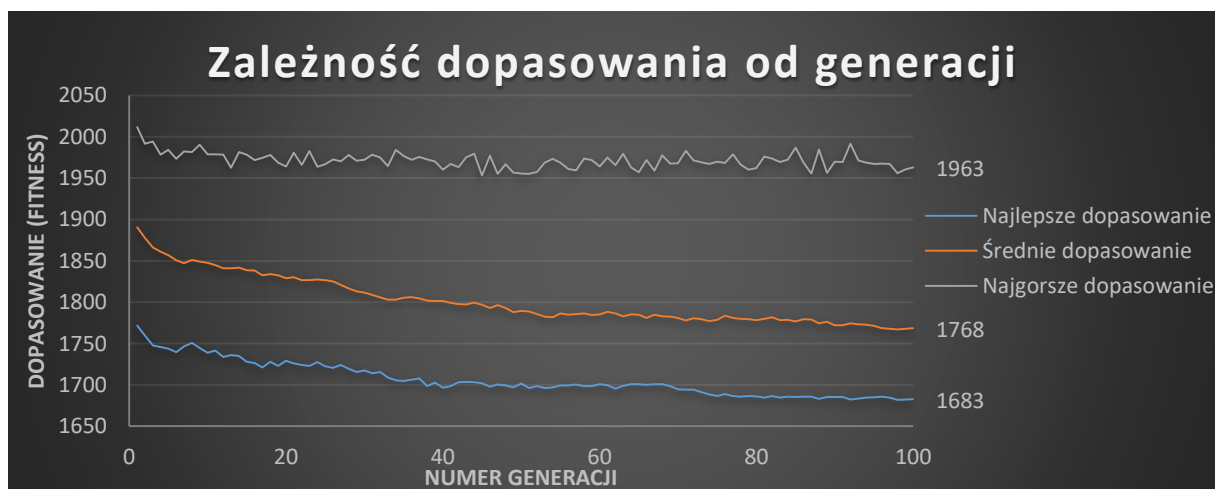
Tabela 8. Parametry algorytmu genetycznego dla ruletki zoptymalizowanej.



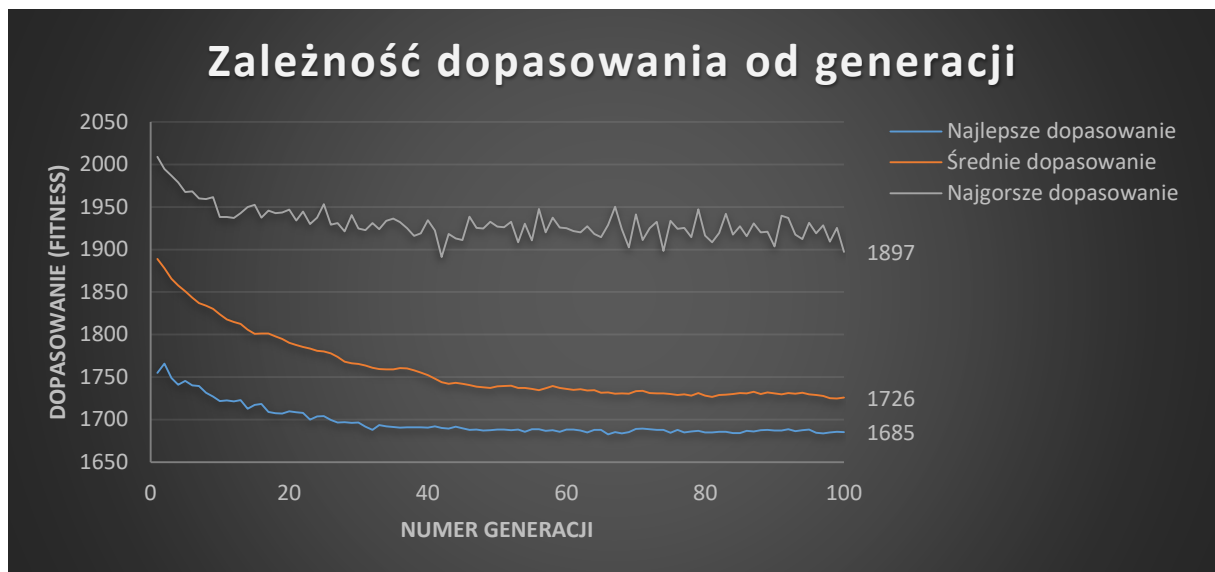
Wykres 8. Wyniki dla pliku „had12.dat” dla $P_x = 0,7$.



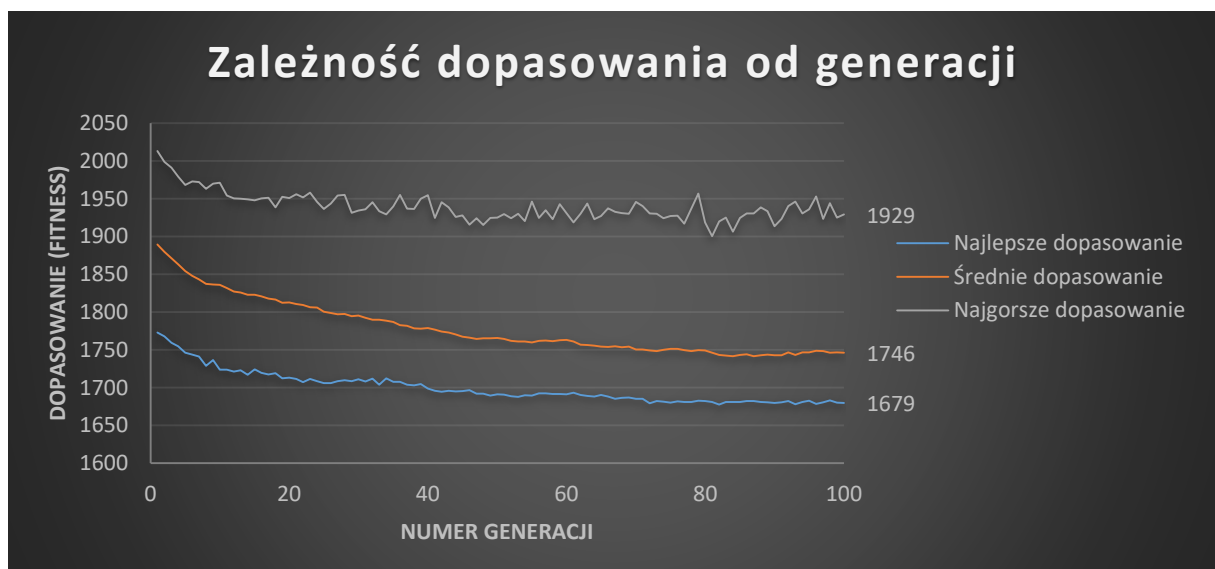
Wykres 9. Wyniki dla pliku „had12.dat” dla $P_x = 0,9$.



Wykres 10. Wyniki dla pliku „had12.dat” dla $P_x = 0,9$ i $P_m = 0.1$.



Wykres 11. Wyniki dla pliku „had12.dat” dla $P_x = 0,99$ i $P_m = 0,1$.



Wykres 12. Wyniki dla pliku „had12.dat” dla $P_x = 0,99$ i $P_m = 0,2$.

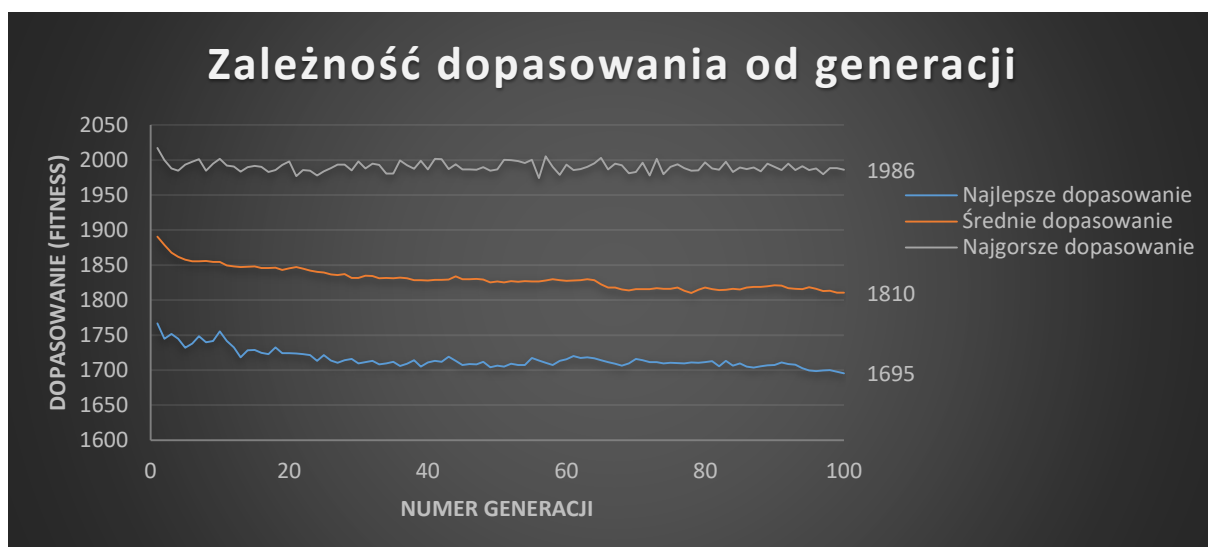
Wnioski:

Selekcja za pomocą ruletki zoptymalizowanej działa poprawnie tylko dla wysokiego współczynnika krzyżowania P_x . W celu zwiększenia skuteczności algorytmu należało zatem zwiększyć współczynnik mutacji, aby wprowadzić do algorytmu większą różnorodność osobników.

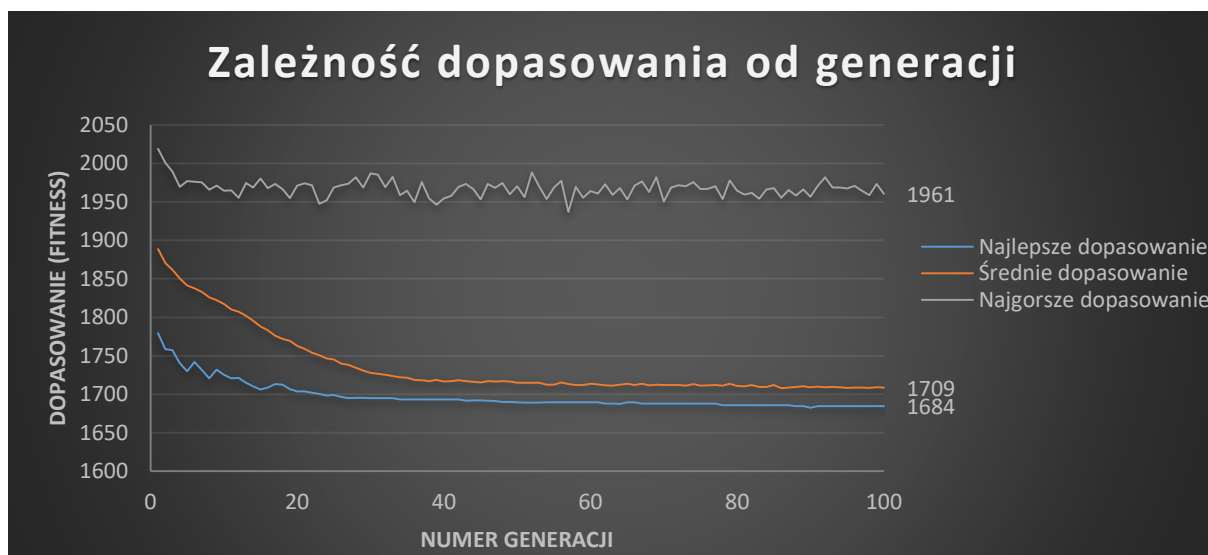
c) Turniej

Ilość generacji	popSize	P_x	P_m	Selekcja
100	100	0,7	0,01	Turniej

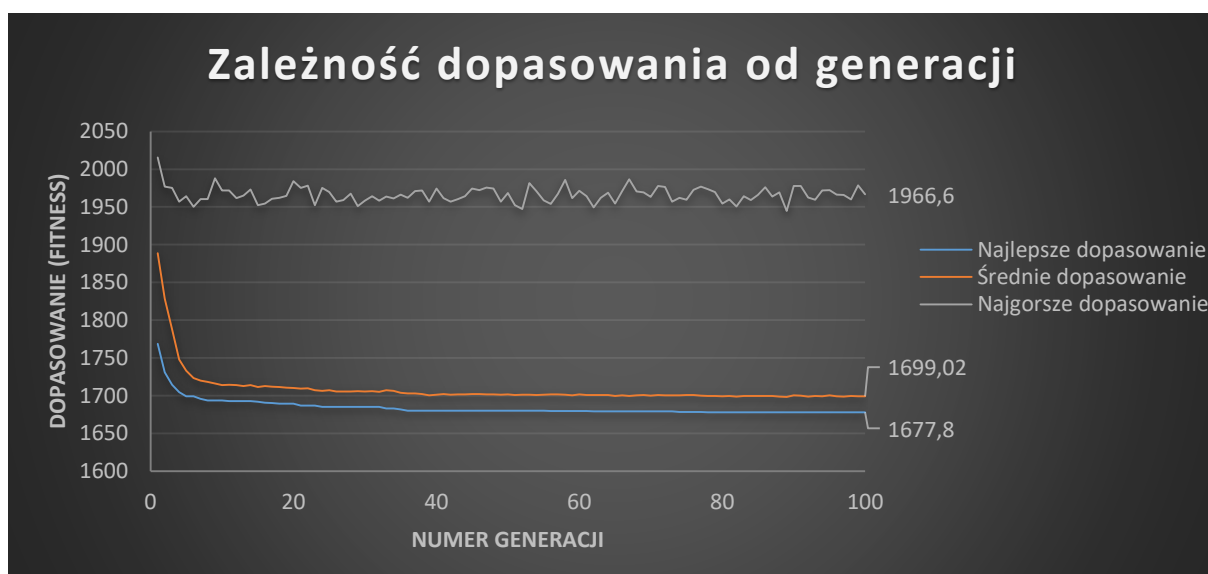
Tabela 8. Parametry algorytmu genetycznego dla turnieju.



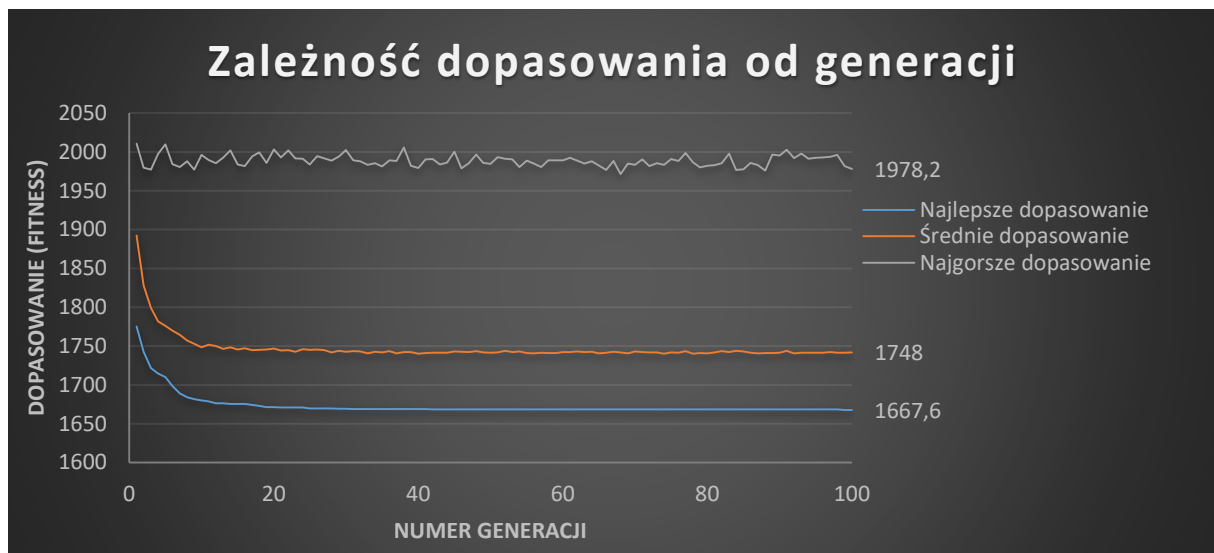
Wykres 13. Wyniki dla pliku „had12.dat” dla Tour = 2 i Px = 0,7



Wykres 14. Wyniki dla pliku „had12.dat” dla Tour = 2 i Px = 0,9



Wykres 15. Wyniki dla pliku „had12.dat” dla Tour = 20 i Px = 0,9



Wykres 16. Wyniki dla pliku „had12.dat” dla Tour = 100, Px = 0,7 i Pm = 2.

Wnioski:

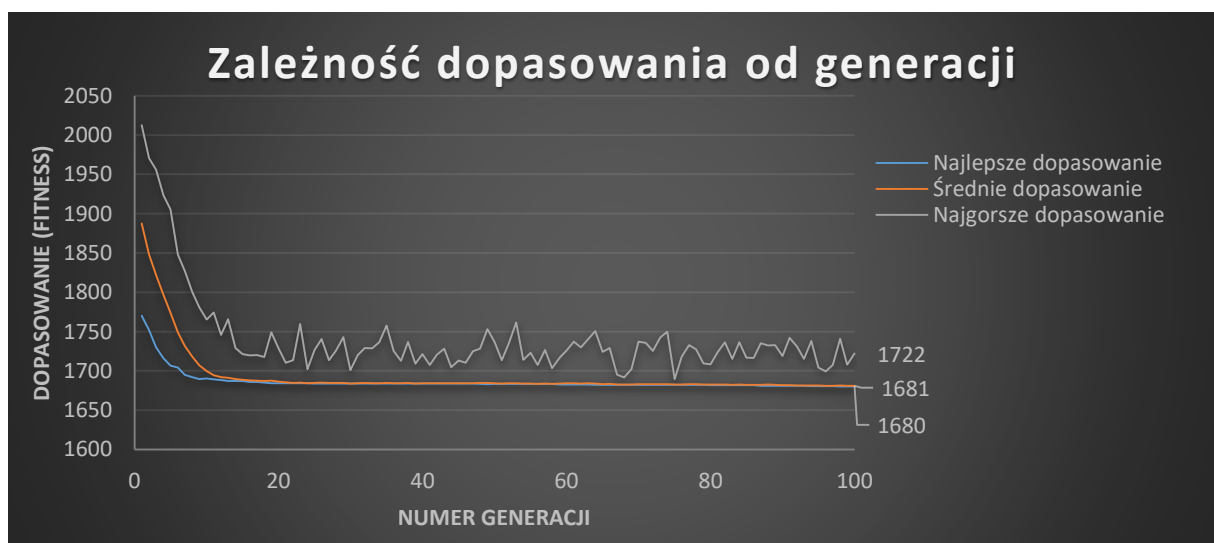
Selekcja za pomocą turnieju działa poprawnie jeżeli parametry Tour i Px mają odpowiedni stosunek. Jeżeli prawdopodobieństwo krzyżowania jest małe ilość wybieranych osobników do turnieju musi być większa. W innym wypadku będzie małe prawdopodobieństwo przetrwania najlepszych osobników. Jeżeli parametr Tour będzie równy ilości osobników w populacji to przez cały czas będzie krzyżował się ten sam najlepszy osobnik.

Zostały pokazane tylko wykresy z analizy jednego pliku ponieważ wyniki dla reszty plików były bardzo podobne.

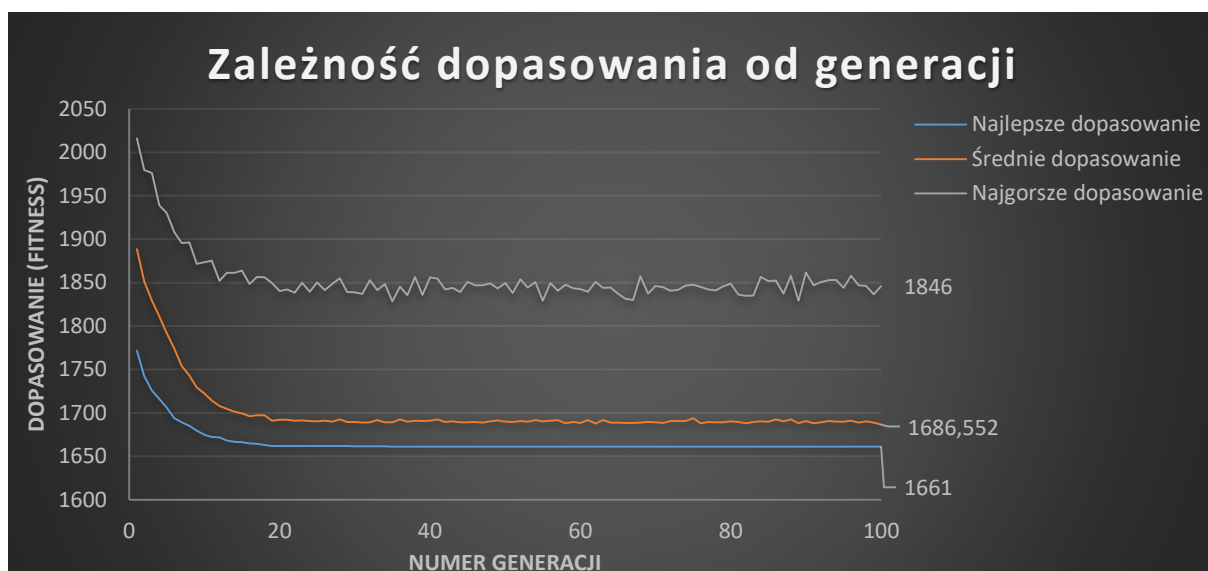
4. Wpływ krzyżowania i mutacji na efektywność algorytmu genetycznego.

Ilość generacji	popSize	Tour	Selekcja
100	100	5	Turniej

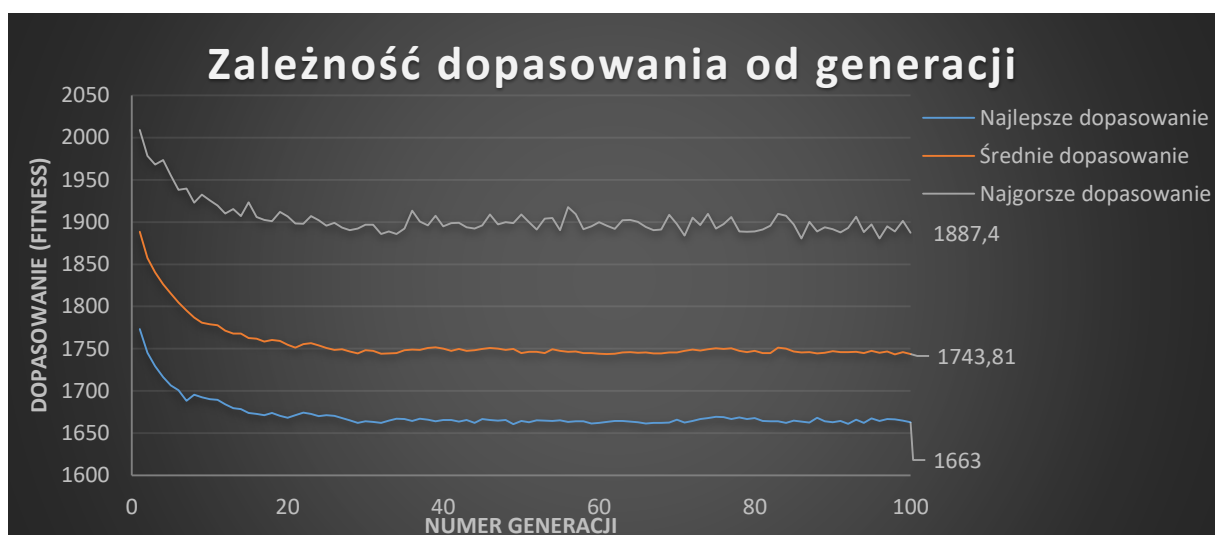
Tabela 9. Wstępne parametry algorytmu genetycznego, dla których zostały wykonane testy w bieżącym podpunkcie.



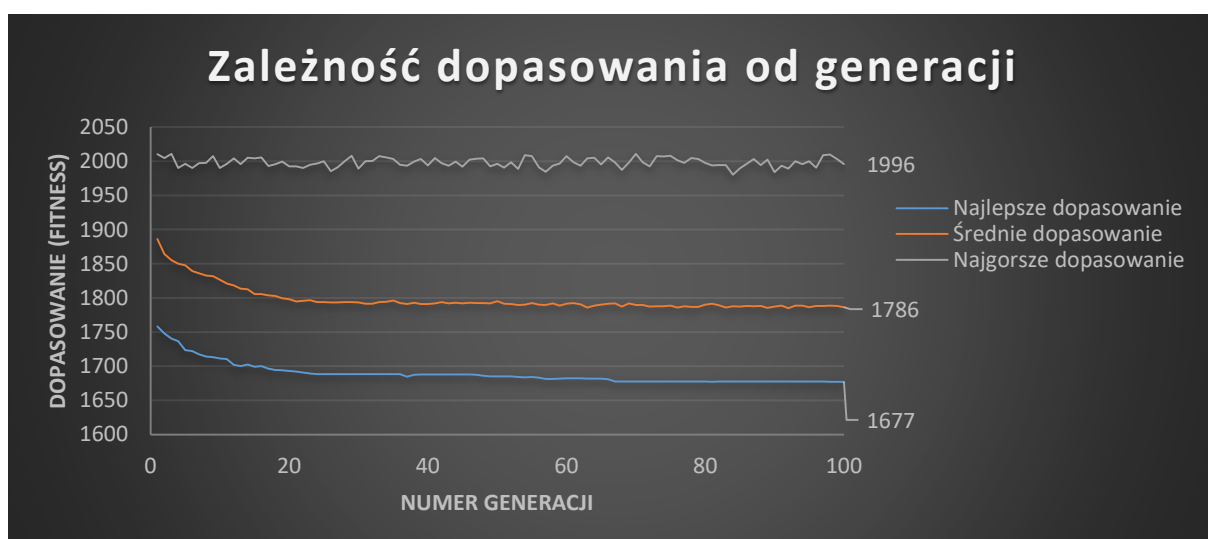
Wykres 17. Wyniki dla pliku „had12.dat” dla Px = 1 i Pm = 0,01.



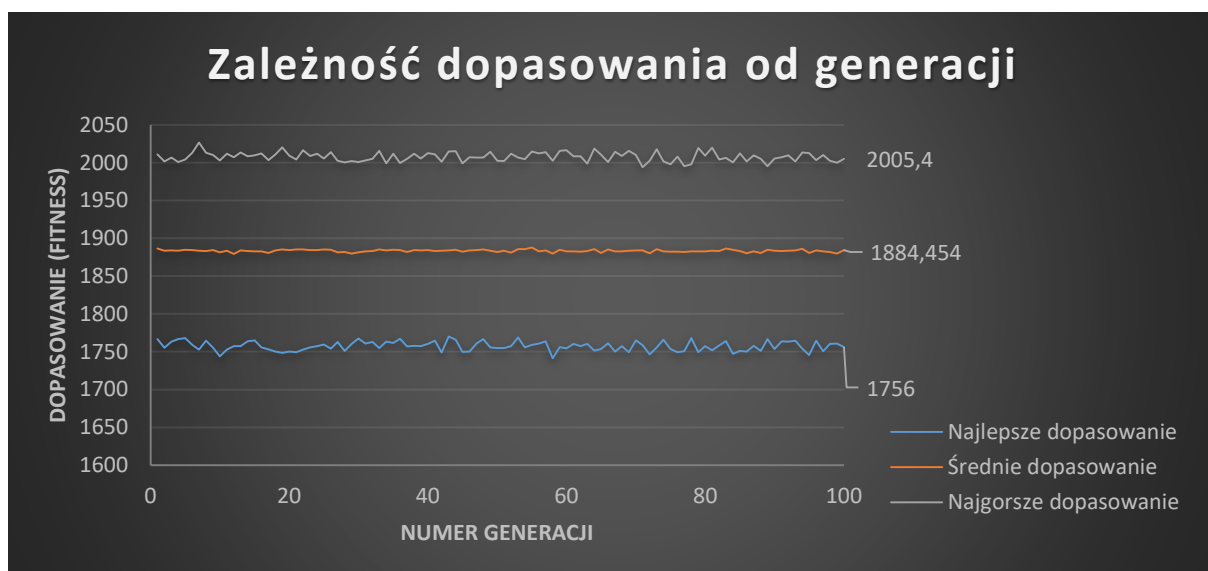
Wykres 18. Wyniki dla pliku „had12.dat” dla $P_x = 1$ i $P_m = 0,5$.



Wykres 19. Wyniki dla pliku „had12.dat” dla $P_x = 1$ i $P_m = 1$.



Wykres 20. Wyniki dla pliku „had12.dat” dla $P_x = 0,5$ i $P_m = 0,1$.



Wykres 21. Wyniki dla pliku „had12.dat” dla $P_x = 0,1$ i $P_m = 0,01$.

Wnioski:

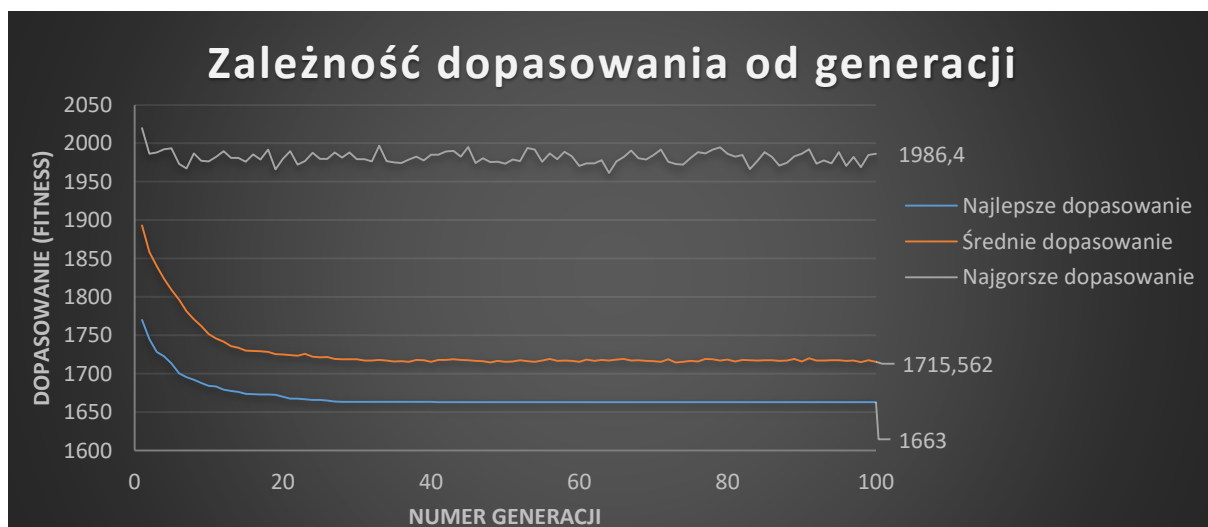
Jeżeli zbyt mało osobników się krzyżuje to maleje prawdopodobieństwo przekazania „dobrych” genów dalej. Zatem mogą przepaść osobniki z najlepszym dopasowaniem co będzie skutkowało oddaleniem się od znalezienia pożądanej najoptymalniejszej wartości.

Z drugiej strony jeżeli zbyt dużo osobników jest mutowanych to istnieje duże prawdopodobieństwo tego, że chwilowe znalezione najoptymalniejsze rozwiązanie zostanie zmutowane na gorsze.

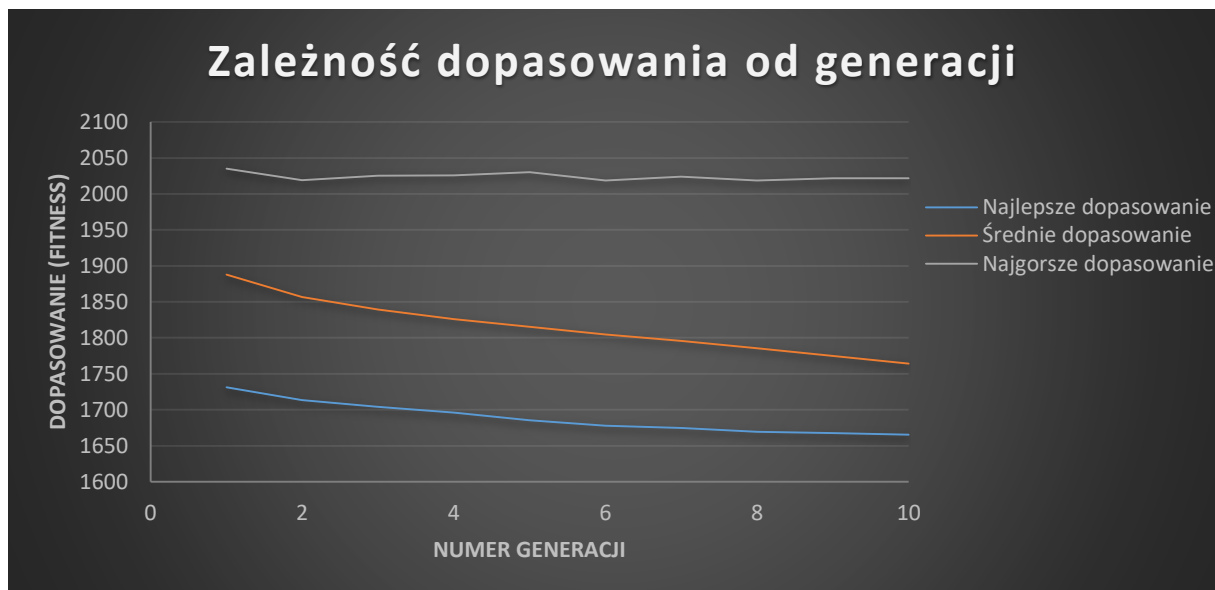
5. Wpływ liczby osobników i rozmiaru populacji na algorytm genetyczny.

Ilość generacji	popSize	P_x	P_m	Tour	Selekcja
100	100	0,8	0,2	5	Turniej

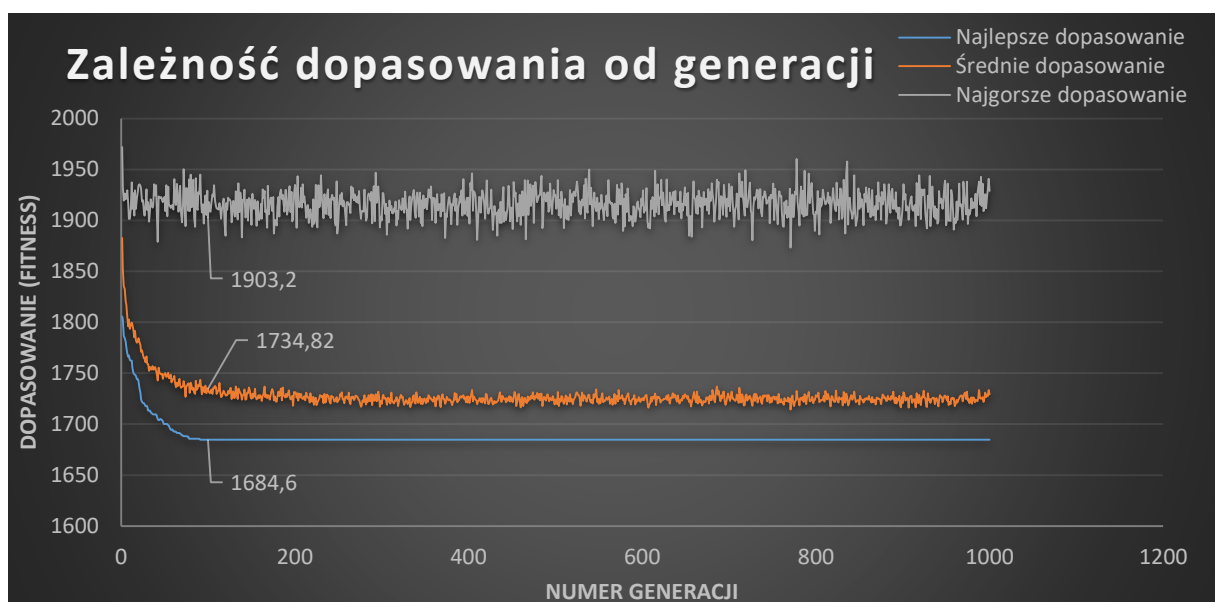
Tabela 10. Wstępne parametry algorytmu genetycznego, dla których zostały wykonane testy w bieżącym podpunkcie.



Wykres 22. Wyniki dla pliku „had12.dat” dla ilości gen = 100 i liczby urodzeń = 80.



Wykres 23. Wyniki dla pliku „had12.dat” dla ilości gen = 10 i liczby urodzeń = 800.



Wykres 24. Wyniki dla pliku „had12.dat” dla ilości gen = 1000 i liczby urodzeń = 8.

Jeżeli ilość generacji jest zbyt duża to od pewnego momentu nie wpływa ona praktycznie w żaden sposób na wynik końcowy. Szczególnie jest to widoczne na wykresie 24 gdzie liczba urodzeń jest zbyt mała w stosunku do wielkości parametru mutacji oraz do wielkości turnieju i zbyt duża w stosunku do wielkości populacji. Przez to tworzy się zbyt mało nowych osobników co prowadzi w końcu do tego, że cały czas są powielane te same.

6. Porównanie algorytmu genetycznego z innymi algorytmami optymalizacyjnymi

Nazwa pliku	„had12”	„had14”	„had16”	„had18”	„had20”	„had30”
RandomSearch (100000 osobników)	1698	2804	3846	5554	7222	16819
GreedyAlgorithm (1000 osobników)	1698	2800	3856	5516	7254	16578
GreedyAlgorithm (10000 osobników)	1672	2796	3814	5516	7210	16578
GeneticAlgorithm ($P_x=0,8$, $P_m=0,2$, Tour = 5)	1664	2734	3733	5400	6974	15639
Max optimization	1652	2724	3720	5358	6922	-

Tabela 11. Porównanie wyników dla algorytmów niegenetycznych.

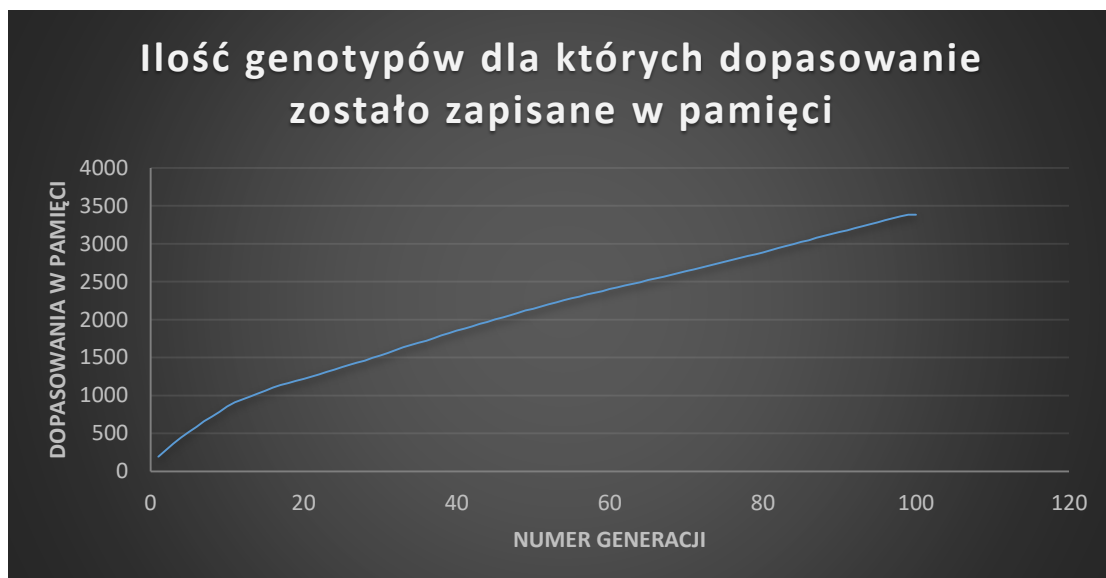
Wnioski:

Algorytmy niegenetyczne mogą rozwiązywać proste problemy osiągając przy nich zadawalające wyniki. Natomiast nie są one w stanie podać dobrego rozwiązania dla złożonych problemów niezależnie od ilości podanych na start losowych osobników.

7. Optymalizacja

a) „Najdroższa” metoda w kodzie.

Najdroższą metodą w kodzie jest liczenie „fitnessu” dla poszczególnego genotypu. Spowodowane jest to wyciąganiem zmiennych z tablicy dwuwymiarowej. W celu optymalizacji tej metody został stworzony słownik, który przechowuje obliczone dopasowania.



Wykres 25. Ilość zapisanych w słowniku dopasowań.

Jak widać z wykresu po kilku generacjach do słownika jest zapisywanych mniej więcej tyle genotypów ile tworzy się całkowicie nowych osobników. Zatem jest to około $[\text{popSize} - (\text{popSize} * P_x)]$. Pozwoliło to przyspieszyć tę funkcję o około procent krzyżujących się osobników.

8. Problem dobrania odpowiednich parametrów

Dobranie najbardziej optymalnych parametrów algorytmu genetycznego może stanowić problem. Jego rozwiązaniem może być powtórne zastosowanie algorytmu genetycznego, ale tym razem na parametrach. Należałoby stworzyć osobnika, który jako klucz genotypu będzie miał zapisane typy parametrów, a jako wartość genotypu wartość parametru (jak w tabeli 12).

GeneKey	Ilość generacji	popSize	Px	Pm	Tour	Selekcja
GeneValue	100	100	0,8	0,2	5	Turniej

Tabela 12. Przykładowy osobnik.