

Analizowany problem

Problem wymaga od algorytmu zaprojektowania płytki PCB spełniającej określone wymagania. Dane, które są na początku to:

- Ograniczony, spójny obszar płaszczyzny
- Uporządkowany zbiór punktów lutowniczych P
- Funkcja przyporządkowująca każdemu punktowi P jego pozycję na płycie
- Uporządkowany zbiór połączeń pomiędzy punktami

Problem polega na zaprojektowaniu sieci fizycznych połączeń w taki sposób, aby dowolne dwa punkty lutownicze zostały fizycznie połączone wtedy i tylko wtedy, jeżeli występowało między nimi połączenie strukturalne.

Warunki, które muszą zostać spełnione:

- Płytkę składa się z jednej warstwy w kształcie prostokąta, o ustalonych rozmiarach
- Punkty leżą na przecięciach siatki
- fizyczne połączenia mogą być prowadzone tylko wzdłuż nałożonej na powierzchnię płytki siatki, zbudowanej z kwadratów o boku równym 1.

W problemie należy tak utworzyć rozwiązanie, aby żadna ze ścieżek się nie przecinała i nie wychodziła poza płytkę. Dodatkowo, należy zminimalizować ich długość.

Badanie wpływu rozmiaru populacji na wyniki algorytmu genetycznego

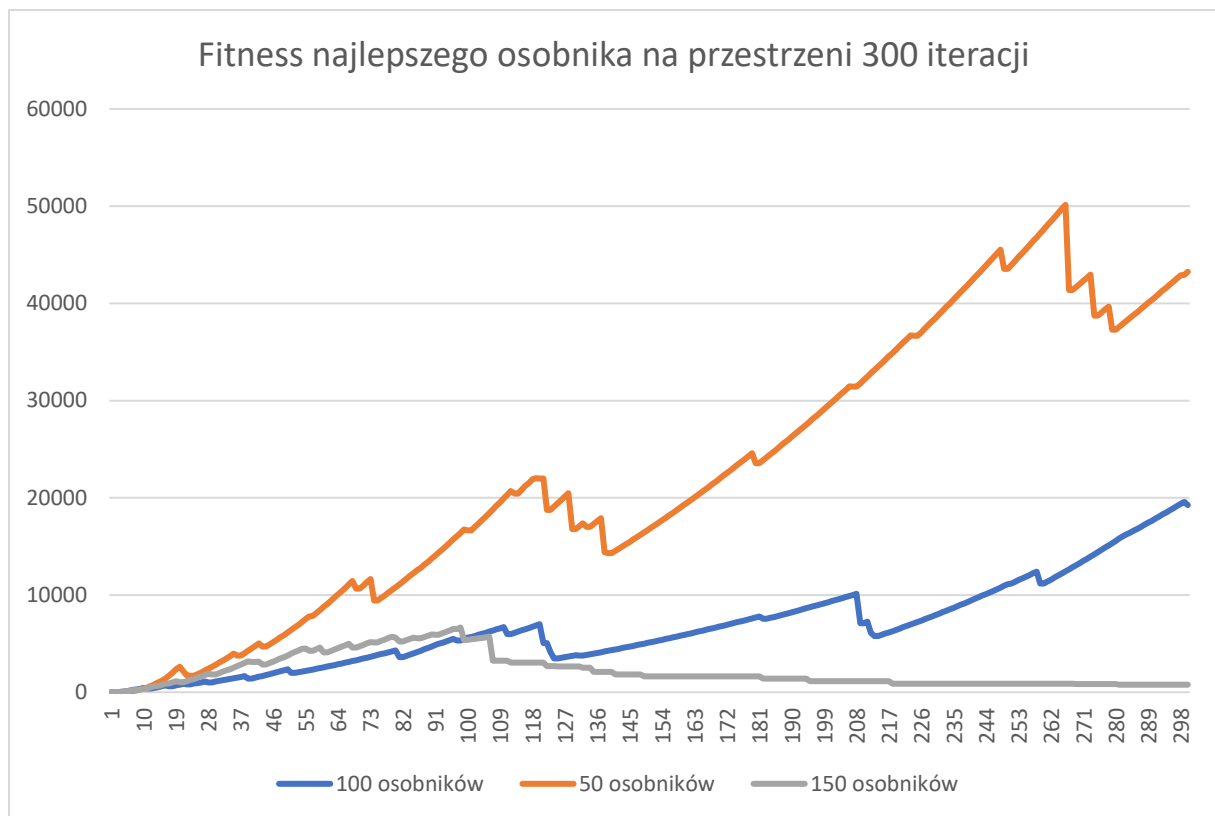
Testy przeprowadzono dla 300 iteracji oraz następujących rozmiarów populacji:

- 50
- 100
- 150

Algorytm wykorzystywał krokową adaptację wag, zatem musiałem przeskalować wyniki, aby móc porównać je z metodą losową. Prawdopodobieństwo mutacji wynosiło 0.3, a prawdopodobieństwo krzyżowania 0.2.

Zadanie 1

W przypadku zadania pierwszego większa ilość osobników pozytywnie wpływała na jakość rozwiązań, poniższy wykres prezentuje przykładową, bardzo powtarzalną sytuację. Widać tutaj również działanie krokowej adaptacji wag, która eksplatuje ekstremum lokalne, by następnie „skoczyć” dalej.



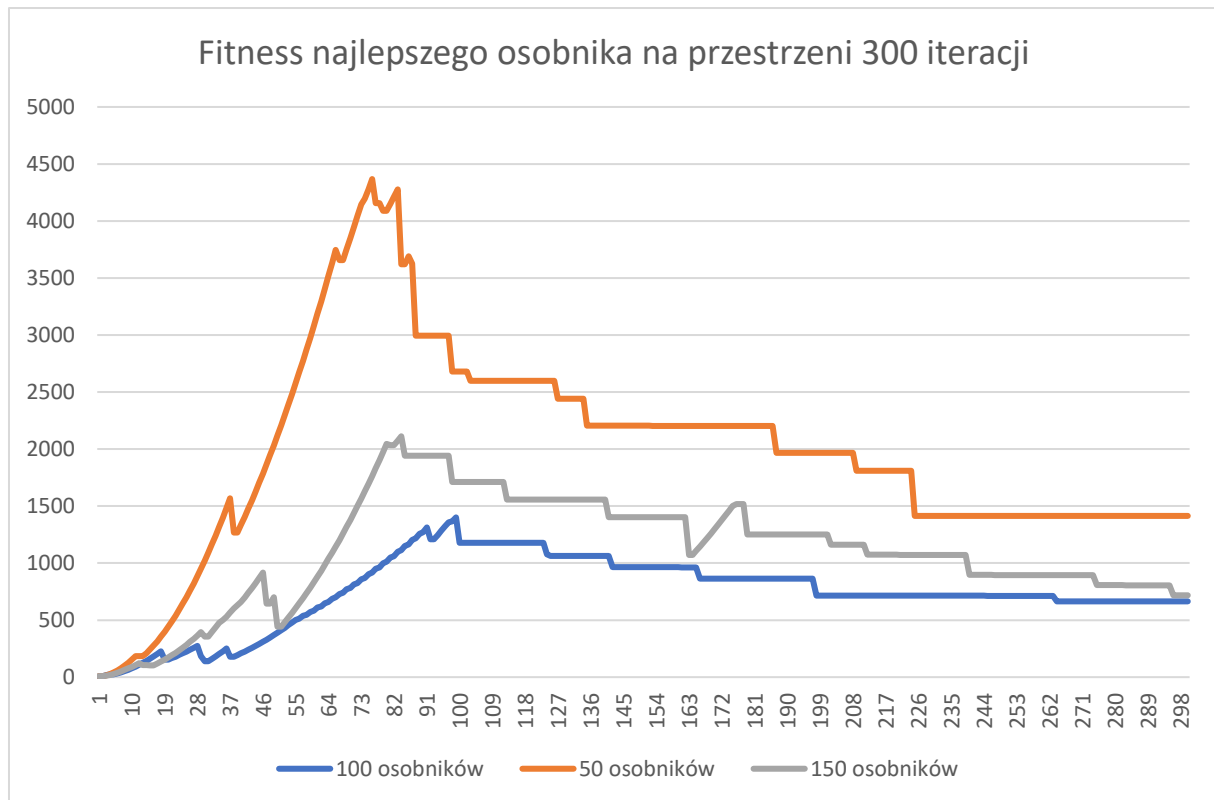
W tabeli znajdują się uśrednione wyniki z badania.

	50 osobników	100 osobników	150 osobników
Fitness	22627,55	19335,43	12046,71

Przy większej ilości osobników algorytm był w stanie dużo częściej znaleźć rozwiązanie problemu. Średnie z tabeli powyżej dobrze to obrazują. Mimo że rozbieżność nie jest tak duża, jak na przykładowym wykresie, algorytm zdecydowanie preferował populacje liczniejsze i bardzo często osiągał wyniki zobrazowane na wykresie.

Zadanie 2

W przypadku zadania nr 2 różnice nie były tak znaczące. Wyniki testów wykonanych na różnych rozmiarach populacji nie odbiegały od siebie bardzo. Widać to na przykładowym wykresie poniżej.



Zadanie 3

W przypadku zadania trzeciego sytuacja była analogiczna. Algorytm wykazuje się lepszymi wynikami dla większych rozmiarów populacji.

Wnioski

Z badań wynika, że większa ilość osobników ma pozytywny wpływ na działanie algorytmu genetycznego. Jest to spowodowane większą przestrzenią przeszukiwanych rozwiązań. Większa ilość osobników wpływa jednak negatywnie na czas wykonywania algorytmu, zatem należy znaleźć odpowiedni kompromis.

Badanie wpływu liczby pokoleń na wyniki algorytmu genetycznego

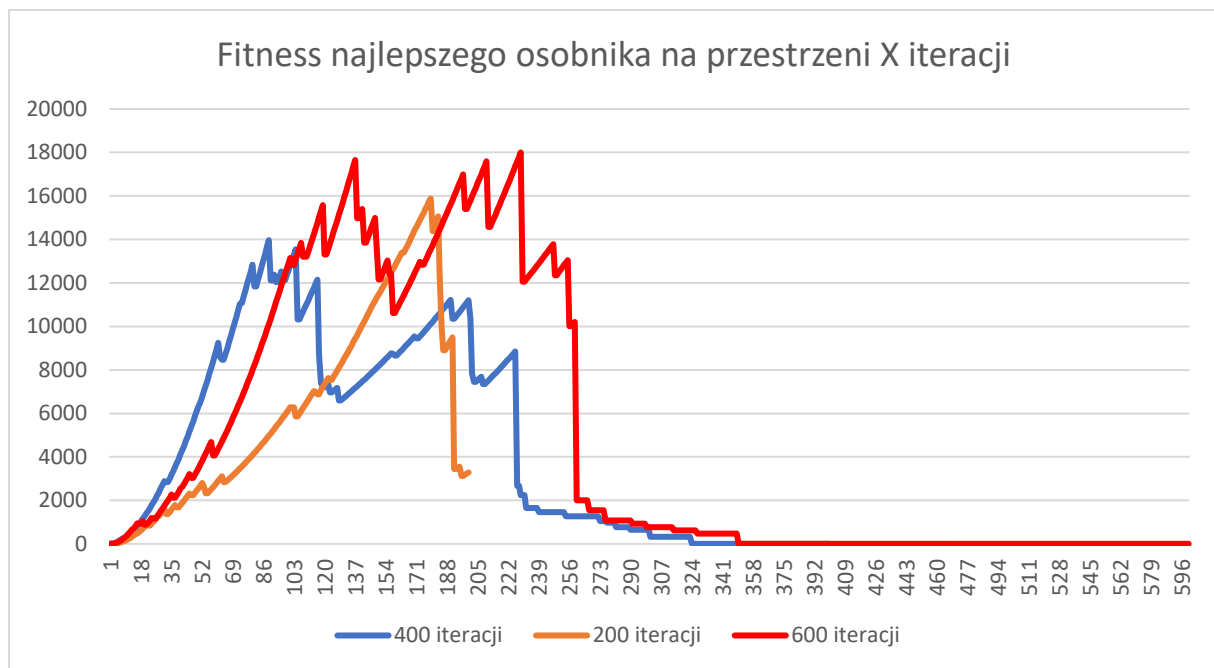
Testy przeprowadzono dla 100 osobników oraz następujących liczb pokoleń:

- 200
- 400
- 600

Prawdopodobieństwo mutacji wynosiło 0.3, a prawdopodobieństwo krzyżowania 0.2.

Zadanie 1

Jak widać na poniższym przykładowym wykresie, większa liczba pokoleń pozytywnie wpływa na wyniki algorytmu genetycznego. Większa liczba iteracji pozwala algorytmowi na dłuższą optymalizację i zwiększa prawdopodobieństwo znalezienia optymalnego rozwiązania.



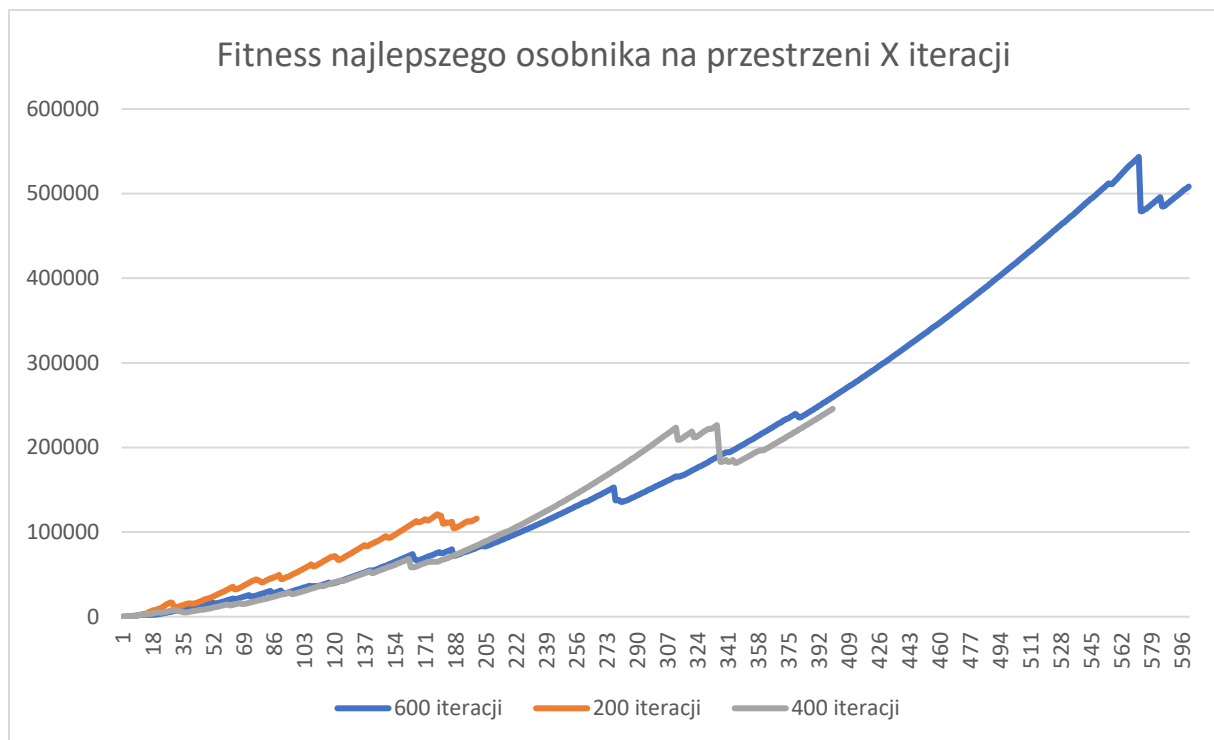
Zadanie 2

W przypadku zadania nr 2 zachodzi taka sama sytuacja. Większa ilość pokoleń prowadzi do poprawy jakości rozwiązania. Poniższa tabela ukazuje średnią wartość adaptacyjnej funkcji przystosowania dl

	200 iteracji	400 iteracji	600 iteracji
Fitness	1320,05	768,69	216,29

Zadanie 3

Zadanie trzecie jest przypadkiem szczególnym – 600 iteracji nie wystarczyło do znalezienia rozwiązania bliskiego optymalnemu, zatem w tym przypadku fitness osobnika najlepszego jest gorszy, niż fitness osobnika najlepszego po 200 iteracjach. Zwiększając jednak liczbę pokoleń algorytm ostatecznie dotarłby do „skoku” i odnalazł odpowiednie rozwiązanie.



Wnioski

Liczba pokoleń ma znaczący wpływ na jakość znalezionego rozwiązania. Po przekroczeniu pewnej liczby iteracji, indywidualnej dla danego problemu, algorytm odpowiednio optymalizuje wyniki.

Badanie wpływu rozmiaru turnieju na wyniki algorytmu genetycznego

Badania przeprowadzono na populacji liczącej 100 osobników, 300 pokoleniach oraz następujących wartościach okna turnieju:

- 5
- 25
- 50
- Prawdopodobieństwo krzyżowania wynosiło 0,2, prawdopodobieństwo mutacji – 0,3.

Zadanie 1

W przypadku zadania pierwszego turniej o rozmiarze okna 5 wykazał się największą skutecznością w znajdowaniu optymalnych rozwiązań, ponieważ znalazł ich najwięcej. Widać to również po średniej wartości fitnessu na przestrzeni 10 iteracji, zamieszczonej w poniższej tabeli

	Turniej okno 5	Turniej okno 25	Turniej okno 50
Fitness	8321,093	18476,82	15024,1

Zadanie 2

W zadaniu drugim najlepiej wypadł turniej o rozmiarze okna 50, jednak różnice w średnich fitnessach są tak niewielkie, że ciężko jednoznacznie stwierdzić wyższość jednego okna nad drugim.

	Turniej okno 5	Turniej okno 25	Turniej okno 50
Fitness	211,8533	324,8	169,24

Zadanie 3

W zadaniu trzecim zaobserwowano podobną zależność, co w zadaniu 2 – różnice pomiędzy średnimi wynikami są bardzo niewielkie.

	Turniej okno 5	Turniej okno 25	Turniej okno 50
Fitness	174952	100308,2	87468,17

Wnioski

Na podstawie przeprowadzonych badań nie udało się stwierdzić znaczących różnic pomiędzy różnymi oknami turnieju. Można jednak przypuszczać, że mniejsze okna turnieju radzą sobie znacznie lepiej z wyszukiwaniem optimów lokalnych, które algorytm może zbadać.

Porównanie operatorów selekcji – ruletki oraz turnieju

Testy przeprowadzono na 100 osobnikach, 300 iteracjach algorytmu oraz turnieju o oknach wynoszących 5, 25 oraz 50.

Prawdopodobieństwo krzyżowania wynosiło 0,2, prawdopodobieństwo mutacji 0,3.

Zadanie 1

W przypadku zadania pierwszego widać zauważalną różnicę pomiędzy średnimi wynikami turnieju o różnych oknach oraz ruletki – ruletka osiąga zdecydowanie gorsze wyniki. Dodatkowo, algorytm, podczas wykorzystywania operatora turnieju, znacznie częściej znajdował poprawne rozwiązania, niż w przypadku operatora ruletki.

	Ruletka	Turniej okno 5	Turniej okno 25	Turniej okno 50
Fitness	32288,99	8321,093	18476,82	15024,1

Zadanie 2

W zadaniu drugim turniej zupełnie zdeklasował ruletkę.

	Ruletka	Turniej okno 5	Turniej okno 25	Turniej okno 50
Fitness	1028,427	211,8533	324,8	169,24

Zadanie 3

W zadaniu 3 zaszła analogiczna sytuacja, co w dwóch poprzednich.

	Ruletka	Turniej okno 5	Turniej okno 25	Turniej okno 50
Fitness	174952	100308,2	87468,17	84836,1

Wnioski

Selekcja turniejowa przy małych oknach osiąga znacznie lepsze wyniki niż selekcja ruletki.

Badanie wpływu parametru krzyżowania na wyniki algorytmu genetycznego

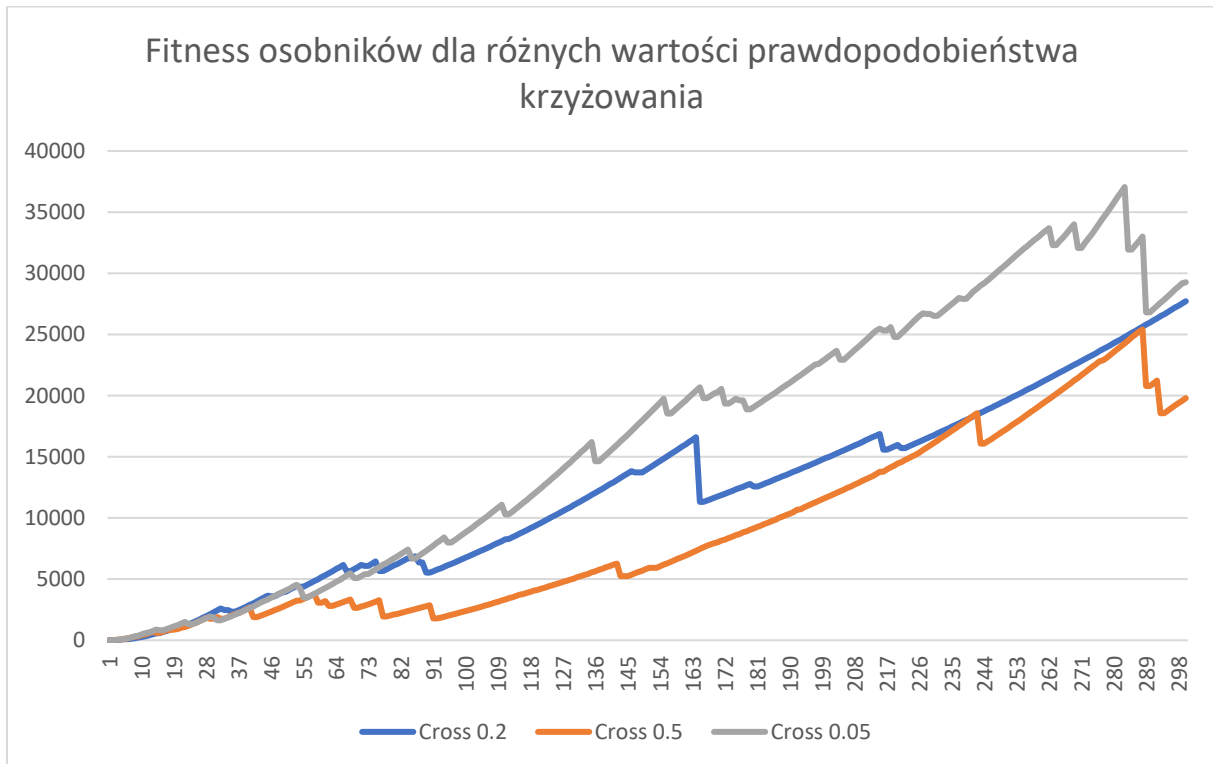
Testy przeprowadzono na 100 osobnikach, 300 iteracjach algorytmu oraz następujących wielkościach parametru krzyżowania:

- 0,2
- 0,5
- 0,05

Prawdopodobieństwo mutacji wynosiło 0,3.

Zadanie 1

Nie udało mi się zaobserwować żadnej znaczącej różnicy w działaniu algorytmu dla różnych wartości prawdopodobieństwa krzyżowania. Widać to na poniższym, przykładowym wykresie. Zmiana prawdopodobieństwa krzyżowania nie zmodyfikowała w żaden sposób trendu.



Brak znaczących zmian widać również na poniższej tabeli, która zawiera średnią wartość adaptacyjnego przystosowania najlepszych osobników na koniec działania algorytmu.

	Cross 0.05	Cross 0.2	Cross 0.5
Fitness	36863,61	38600,027	29026,78

Zad 2

W zadaniu 2, analogicznie do zadania 1, nie widać szczególnej różnicy po zmianie wartości prawdopodobieństwa krzyżowania.

	Cross 0.05	Cross 0.2	Cross 0.5
Fitness	1357,63	1047,93	1027,13

Zad 3

W zadaniu 3 najgorzej wypadło prawdopodobieństwo krzyżowania o wartości 0.05, co widać w poniższej tabeli. Nie są to jednak na tyle duże różnice, aby móc stwierdzić jednoznaczny wpływ tego parametru na algorytm.

	Cross 0.05	Cross 0.2	Cross 0.5
Fitness	174185,4	94935,22	129346,6

Wnioski

Badania nie wykazały znaczącego wpływu prawdopodobieństwa krzyżowania na jakość produkowanych rozwiązań.

Badanie wpływu parametru mutacji na wyniki algorytmu genetycznego

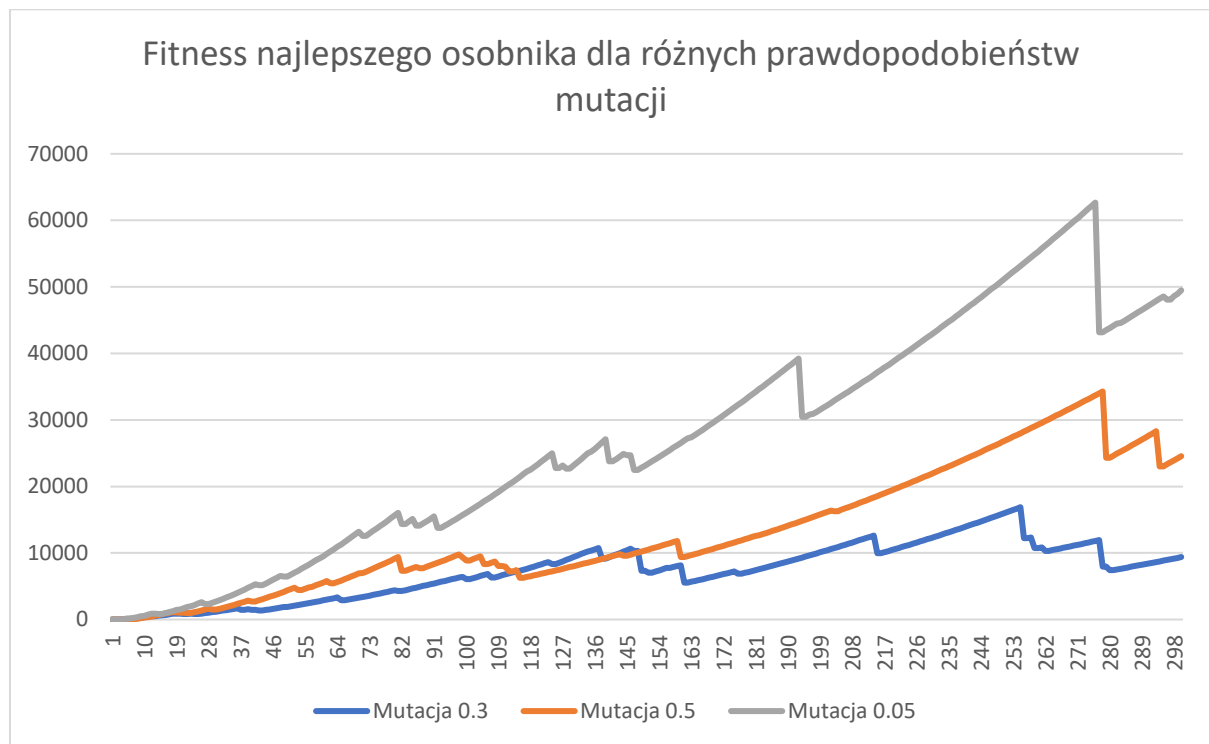
Testy przeprowadzono na 100 osobnikach, 300 iteracjach algorytmu oraz następujących wielkościach parametru mutacji:

- 0,3
- 0,5
- 0,05

Prawdopodobieństwo krzyżowania wynosiło 0,2.

Zadanie 1

Wyniki badań prawdopodobieństwa mutacji dla zadania pierwszego wykazały, że wartości skrajne, tj. zbyt duże oraz zbyt małe (przy czym wartość zbytniego niedostosowania uzależniona jest od wielu parametrów, m. in. problemu czy wielkości populacji), wykazują dużo gorsze wyniki niż wartość uśredniona. Dobrze obrazuje to poniższy poglądowy wykres.



Instancja	Algorytm ewolucyjny				Metoda losowa			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
Mut. 0.3	46,94	202,91	60,16	12,78				
Mut. 0.5	20,79	75,97	31,16	12,61				
Mut. 0.05	12,39	67,82	23,02	11,15				

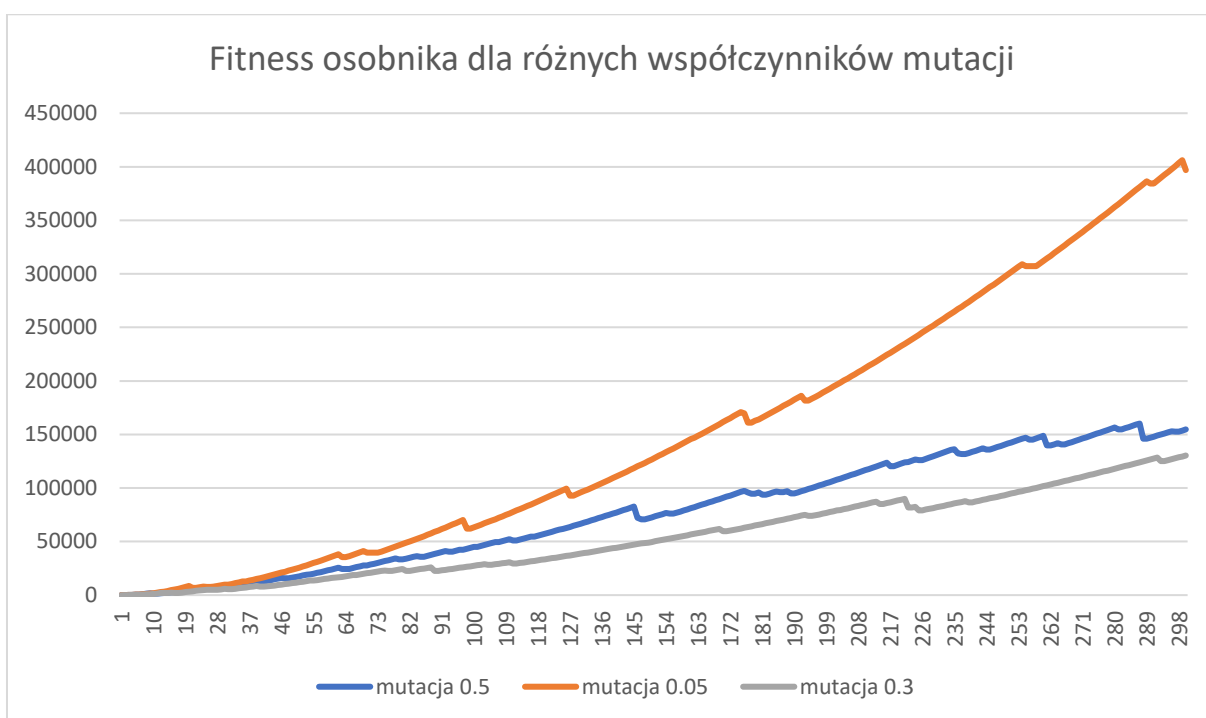
Zadanie 2

Uzyskane wyniki są analogiczne do wyników uzyskanych w zadaniu pierwszym. Algorytm przy prawdopodobieństwie mutacji równym 0.05 nie był w stanie odnaleźć odpowiednio dobrego rozwiązania. W tym przypadku jednak prawdopodobieństwo mutacji równe 0.5 wykazało się nieco lepszą skutecznością, niż w przypadku zadania pierwszego. Widać to w poniższej tabeli.

	Mutacja 0.05	Mutacja 0.3	Mutacja 0.5
Fitness	3763, 36	653,68	401,68

Zadanie 3

Zadanie trzecie nie wykazało zbyt dużych rozbieżności w porównaniu do dwóch poprzednich. Wysokie wartości fitnessu spowodowane są zbyt niską liczbą pokoleń – zadanie to wymaga, w przypadku algorytmu napisanego przeze mnie, stosunkowo dużej ilości iteracji, aby znaleźć rozwiązanie.



	Mutacja 0.05	Mutacja 0.3	Mutacja 0.5
Fitness	358576,2	194733,3	175118,6

Wnioski

Prawdopodobieństwo mutacji ma bardzo duży wpływ na zachowanie algorytmu genetycznego. Zbyt duża lub zbyt mała wartość znacząco utrudnia, bądź uniemożliwia odnalezienie zadowalającego rozwiązania. Ciężko tutaj jednak o podanie najlepszej wartości – uzależnione jest to od rodzaju problemu, wielkości populacji oraz innych czynników.

Porównanie algorytmu genetycznego z metodą naiwną

Porównanie algorytmu genetycznego z metodą naiwną wykonane zostało na problemie nr 2. Dobrane parametry to:

- Ilość pokoleń równa 300, 500 oraz 700
- Wielkość populacji równa 50, 100 oraz 150

- Prawdopodobieństwo mutacji równe 0.3
- Prawdopodobieństwo krzyżowania równe 0.2
- Wielkość okna turnieju równa 15

Jak widać na poniższych tabelach, wyniki algorytmu ewolucyjnego są nieporównywalnie lepsze od wyników uzyskanych przez metodę losową. Dodatkowo, metoda losowa oscyluje wokół pewnej grupy rozwiązań, która powtarza się bardzo często. W poniższych tabelach zestawione zostały wyniki z wywołań algorytmu oraz metody losowej.

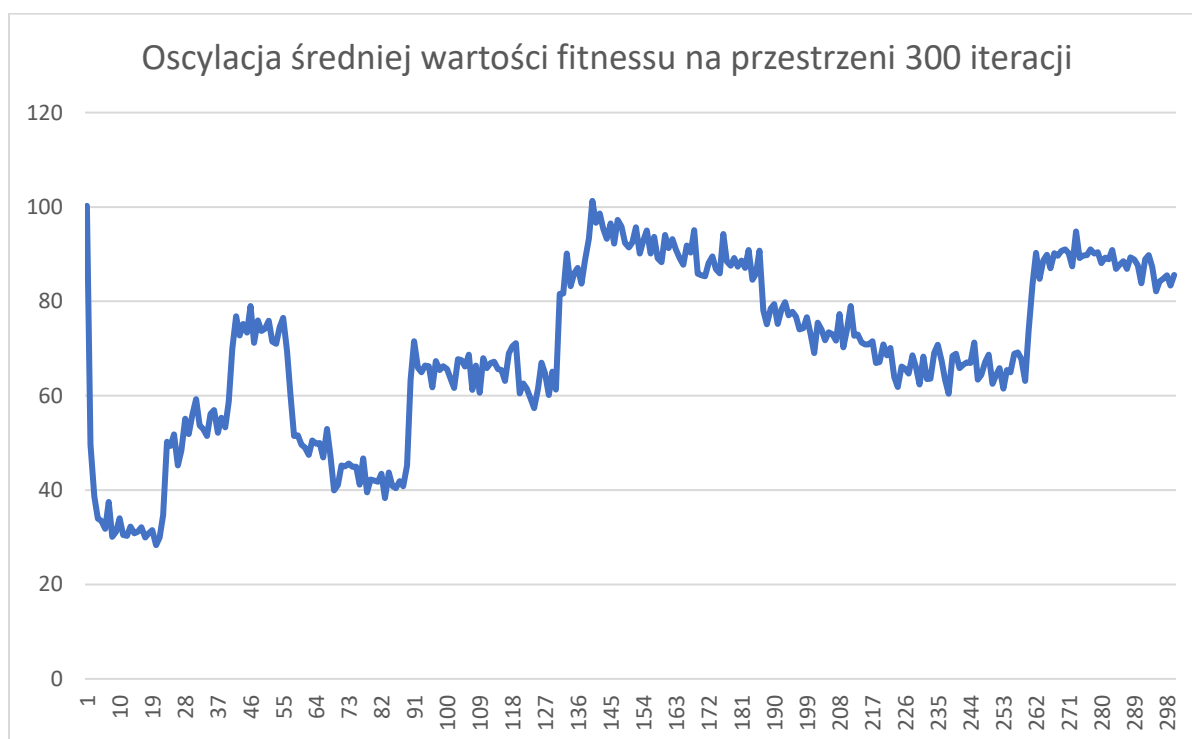
Instancja	Algorytm ewolucyjny				Metoda losowa			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
Populacja 100 Pokolenia 500 Mutacja 0.3	16,2674	56,75	20,09	10,11	36, 9	149,22	33,83	16,93

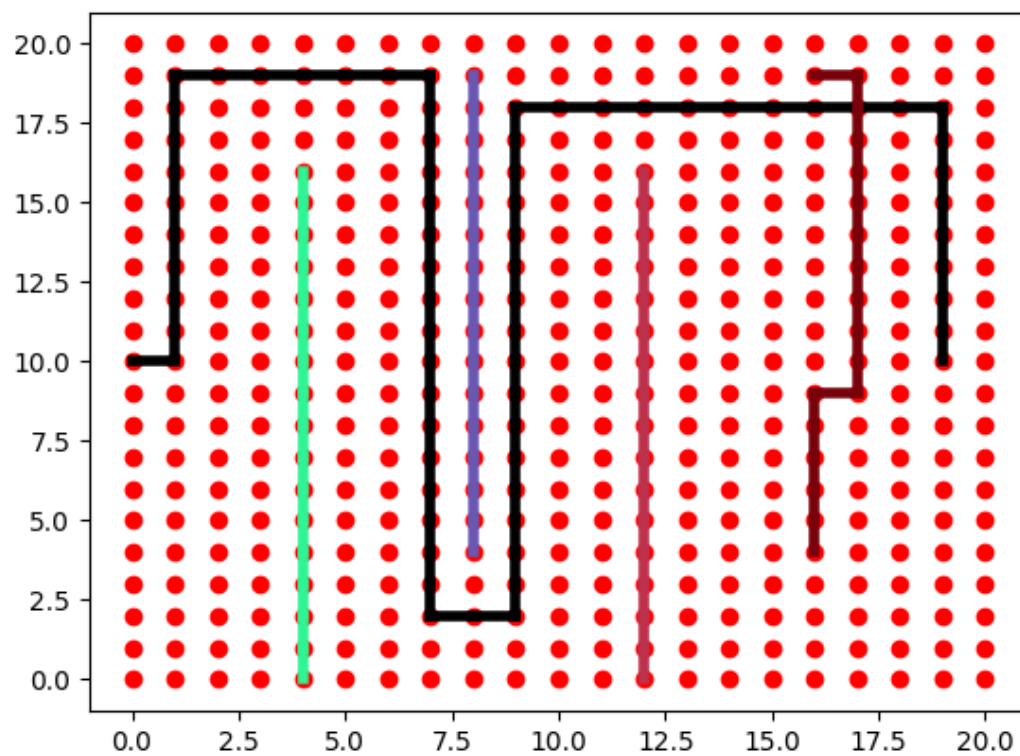
Instancja	Algorytm ewolucyjny				Metoda losowa			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
Populacja 50 Pokolenia 300 Mutacja 0.3	18,94	58,54	23,17	9,85	34,98	160,62	33,87	17,01

Instancja	Algorytm ewolucyjny				Metoda losowa			
	Best	Worst	Avg	Std	Best	Worst	Avg	Std
Populacja 150 Pokolenia 700 Mutacja 0.3	17	60,49	21,24	9,84	24,02	165,3	33,74	16,87

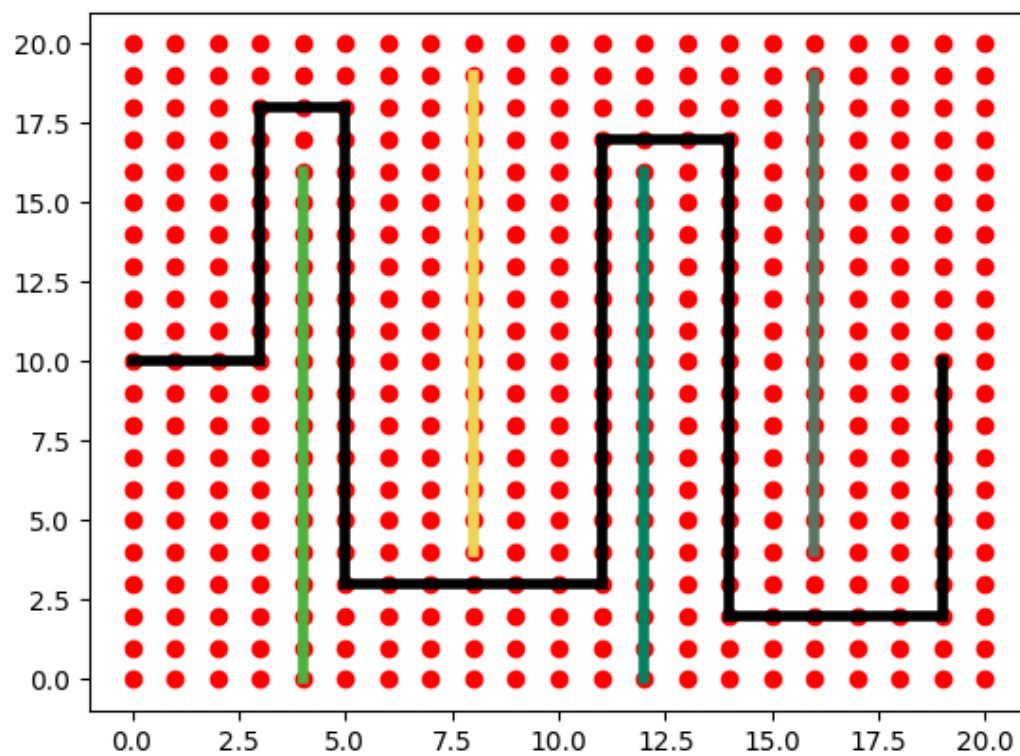
Wnioski

Algorytm osiąga nieporównywalnie lepsze wyniki niż metoda losowa. Metoda losowa zawsze oscyluje w okolicach jednego rozwiązania, natomiast algorytm genetyczny, jest w stanie odnaleźć poprawne, choć nie zawsze optymalne, wielorakie rozwiązania danego problemu.





Zrzut 1 - Przykładowe rozwiązanie problemu przez algorytm (700 iteracji, populacja 150)



Zrzut 2 - Przykładowe rozwiązanie problemu przez algorytm (500 iteracji, populacja 100)