

Rozwiązywanie Problemu Mobilnego Złodzieja za pomocą algorytmu genetycznego

Szymon Woźniak, 235040

22.03.2019

1 Wstęp teoretyczny

Rozwiązujący "Problem Mobilnego Złodzieja" jest złożeniem dwóch trudnych problemów optymalizacyjnych - plecakowego i komiwojażera.

1.1 Problem plecakowy

W problemie plecakowym mamy do dyspozycji plecak o zadanej pojemności C i zbiór N przedmiotów x_1, x_2, \dots, x_N . Każdy z nich posiada określoną wagę w_i i wartość p_i . Celem jest wybranie takiego podzbioru dostępnych przedmiotów, żeby zmaksymalizować zysk, jednocześnie nie przekraczając pojemności plecaka. Maksymalizowana funkcja ma zatem postać:

$$g(y) = \sum_{i=1}^N p_i y_i \quad (1)$$

przy ograniczeniu:

$$\sum_{i=1}^N w_i < C \quad (2)$$

Symbole użyte w równaniach 1 i 2 oznaczają odpowiednio:

- y - strategia wyboru przedmiotów,
- p_i - wartość i -tego przedmiotu,
- y_i - to czy dany przedmiot został zabrany czy nie (1 albo 0).

W przypadku "Problemu Mobilnego Złodzieja" każdy przedmiot ma dodatkowo przypisane miasto, z którego może on zostać zabrany.

1.2 Problem komiwojażera

Problem komiwojażera w przypadku tego zadania składa się z N miast i odległości między każdą parą. Celem jest wybranie takiej trasy, która przechodzi przez każde z miast dokładnie raz, przy minimalnym czasie podróży:

$$f(x) = \sum_{i=1}^{N-1} (t_{x_i, x_{i+1}}) + t_{x_N, x_1} \quad (3)$$

gdzie:

- x - wybrana trasa,
- $t_{x_i, x_{i+1}}$ - czas przejścia pomiędzy miastem i a miastem $i + 1$.

Czas przejścia pomiędzy dwoma miastami może być obliczony ze wzoru:

$$t_{x_i, x_{i+1}} = \frac{d_{x_i, x_{i+1}}}{v_{i, i+1}} \quad (4)$$

gdzie:

- $d_{x_i, x_{i+1}}$ - odległość pomiędzy miastem i a miastem $i + 1$,
- $v_{x_i, x_{i+1}}$ - prędkość na trasie pomiędzy miastem i a miastem $i + 1$.

1.3 Problem Mobilnego Złodzieja

Dwa powyższe problemy zostały ze sobą silnie powiązane poprzez sposób ich definicji. Wspomniana w równaniu 4 prędkość jest zależna od przedmiotów, które zostały zabrane z miast odwiedzonych na trasie przed miastem o indeksie i . Oznacza to, że znalezienie optimum każdego z podproblemów najprawdopodobniej nie daje rozwiązania globalnie optymalnego. Finałnym celem "Problemu Mobilnego Złodzieja" jest zatem maksymalizacja:

$$G(x, y) = g(y) - f(x, y) \quad (5)$$

czyli różnicy sumy wartości wybranych przedmiotów i czasu przejścia trasy przy danym wyborze przedmiotów.

1.4 Algorytm genetyczny

Algorytm genetyczny jest metaheurystyką stosowaną w optymalizacji, wzorowaną na biologicznej ewolucji. GA operuje na populacjach rozwiązań, z których do każdej kolejnej generacji wybierani są rodzice. Wyższe prawdopodobieństwo przekazania genów, mają osobniki o wyższym przystosowaniu. Dodatkowo wprowadza się tu pojęcia krzyżowania i mutacji. Pierwsze z nich określa sposób łączenia materiału genetycznego rodziców przy przekazywaniu ich potomkom. Drugie z kolei definiuje znaną z ewolucji przypadkową, występującą w wyniku błędu, zmianę informacji genetycznej. Może ona jednak potencjalnie wprowadzać do populacji pożądane cechy, zwiększające szanse danego osobnika na przeżycie.

Aby skorzystać z tej metaheurystyki, należy więc zdefiniować wszystkie wymienione wcześniej pojęcia w kontekście rozwiązywanego problemu. Potrzeba zdefiniować: potencjalne rozwiązanie, operacje selekcji, krzyżowania, mutacji i funkcję oceny osobnika.

2 Plan pracy

W pierwszej części pracy krótko opisane zostaną istotne cechy zaimplementowanego algorytmu genetycznego i modelu problemu. Następnie dla 5 wybranych przypadków testowych zostaną przeprowadzone badania działania metody. W następnej kolejności przebadany będzie wpływ poszczególnych jej parametrów i wybranego operatora selekcji, na jakość otrzymywanych wyników. W końcowej części skuteczność algorytmu genetycznego zostanie porównana ze skutecznością wybranych metod nieewolucyjnych.

3 Cechy algorytmu i modelu

W tej sekcji opisanych zostanie kilka, istotnych z punktu badań, cech zaimplementowanego modelu problemu i algorytmu genetycznego.

3.1 Algorytm genetyczny

3.1.1 Parametry algorytmu genetycznego

W zaimplementowanym algorytmie genetycznym wyróżnia się następujące parametry sterujące jego działaniem:

- *pop_size* - wielkość populacji,
- *gen* - liczba generacji przed zatrzymaniem działania,
- p_x - prawdopodobieństwo, że dwa osobniki zostaną skrzyżowane,
- p_m - prawdopodobieństwo mutacji, którego dokładna definicja zostanie podana w sekcji 3.1.3,
- *tour* - wielkość turnieju, dla badań wykorzystujących selekcję turniejową.

3.1.2 Operator krzyżowania

Z racji na specyfikę problemu, do przeprowadzenia badań zaimplementowany został wyspecjalizowany operator krzyżowania - OX (*Order Crossover*). Pierwsza część jego działania polega na wyborze losowej podsekwencji genów z jednego z rodziców i skopiowaniu ich do genotypu dziecka w niezmienionej kolejności. Następnie reszta pustych genów jest uzupełniana od lewej do prawej strony wartościami genów z drugiego z rodziców, które nie występują jeszcze w genotypie dziecka. W ten sposób zachowana zostaje poprawność permutacji.

3.1.3 Operator mutacji

Do rozwiązania problemu wybrany został operator mutacji, którego działanie polega na zamianie 2 losowo wybranych genów w rozwiązaniu. Dodatkowo zostały przeanalizowane 2 możliwe podejścia do tego typu mutacji.

Pierwsze z nich dla każdego osobnika, z pewnym prawdopodobieństwem wykonuje jedną losową zamianę 2 genów. Niestety to rozwiązanie okazało się nie dawać wystarczającej możliwości sterowania różnorodnością populacji. Wynika to bezpośrednio z faktu, że nawet przy $p_m = 1$ stosunek zmutowanych genów do wszystkich genów w populacji jest bardzo mały - $\frac{2 * pop_size}{G * pop_size} = \frac{2}{G}$ gdzie G jest liczbą genów osobnika. Dla przykładu może to być $\frac{2}{439}$ dla jednej z rozpatrywanych instancji problemu.

Z uwagi na to, w badaniach zastosowane zostało drugie podejście, które dla każdego genu każdego z osobników, z pewnym prawdopodobieństwem wykonuje zamianę z innym, losowo wybranym genem. W związku z tym p_m w pracy będzie oznaczało prawdopodobieństwo, że gen zostanie zamieniony z innym.

3.2 Problem TTP

We wszystkich badaniach podejście do rozwiązania problemu TTP, opiera się o wstępne rozwiązanie podproblemu plecakowego algorytmem zachłannym i następną próbę znalezienia jak najlepszego rozwiązania problemu komiwojażera. Przy rozwiązywaniu problemu plecakowego algorytmem zachłannym można rozważyć następujące heurystyki:

- wybór przedmiotu o największej wartości,
- wybór przedmiotu o najmniejszej wadze,
- wybór przedmiotu o największym stosunku wartość/waga.

W tej pracy została wybrana trzecia opcja. Dodatkowo założono, że wszystkie przedmioty znane są z góry, więc strategia wyboru przedmiotów było opracowywana tylko raz i była wspólna dla wszystkich rozwiązań problemu TSP.

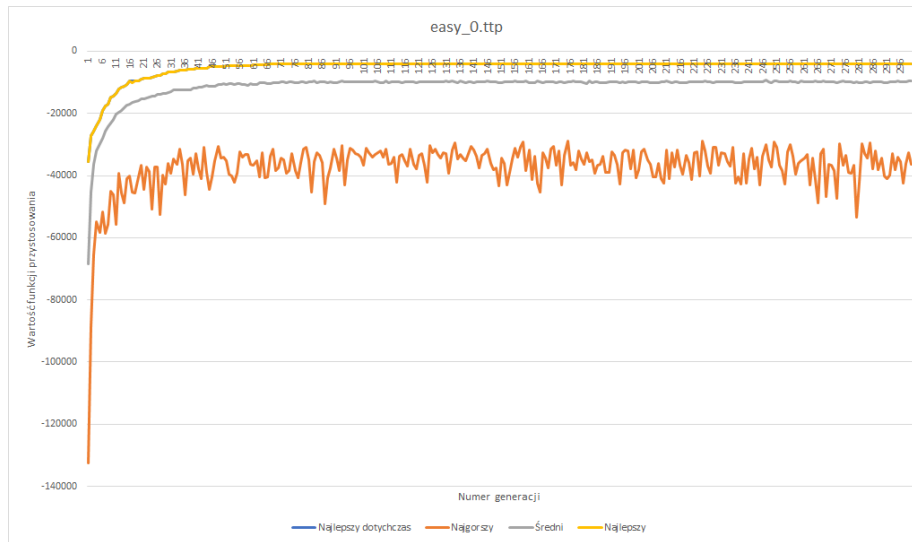
4 Badanie działania

W tej sekcji zostaną przedstawione wyniki badań działania algorytmu genetycznego dla 5 przypadków testowych. Dla każdej instancji problemu, przeprowadzonych zostało 10 uruchomień. Dla każdej instancji zaprezentowana zostanie średnia i odchylenie standardowe z najlepszych znalezionych w każdym z przebiegów rozwiązań, oraz wykresy przedstawiające zmianę funkcji przystosowania w czasie dla wybranego przebiegu przykładowe przebiegi. Wszystkie uruchomienia z wyjątkiem tych dla instancji `hard_0` zostały przeprowadzone z użyciem następujących parametrów metody:

- $gen = 300$,
- $pop_size = 1000$,
- $p_x = 0.8$,
- $p_m = 0.05$,
- $tour = 50$.

Dla problemu `hard_0`, w celu uzyskania wyników wyższej jakości, wartość parametru p_m została obniżona do 0.005.

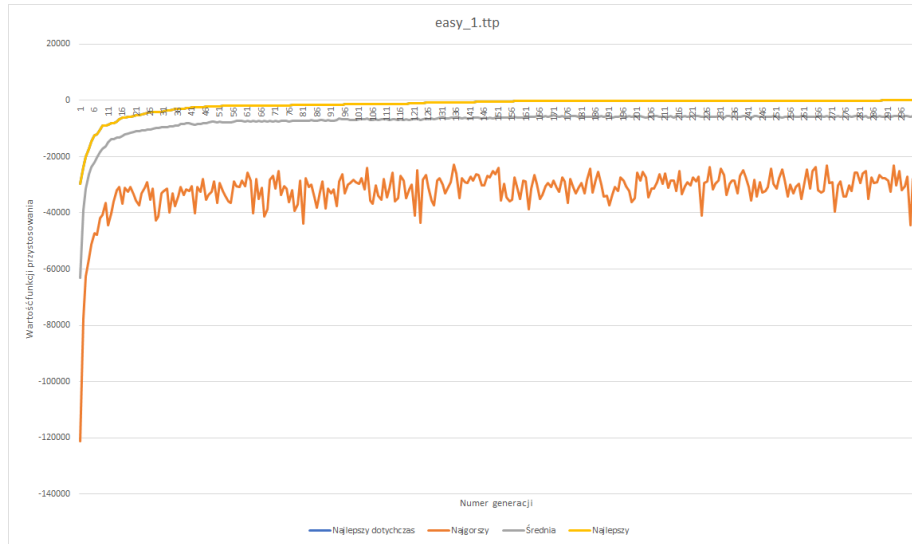
4.1 Problem easy_0.ttp



Rysunek 1: Przykładowy przebieg metody dla instancji easy_0

$$\mu = -5168.9$$
$$\sigma = 719.7$$

4.2 Problem easy_1.ttp



Rysunek 2: Przykładowy przebieg metody dla instancji easy_1

$$\mu = 613.4$$
$$\sigma = 794.64$$

4.3 Problem medium_0.ttp

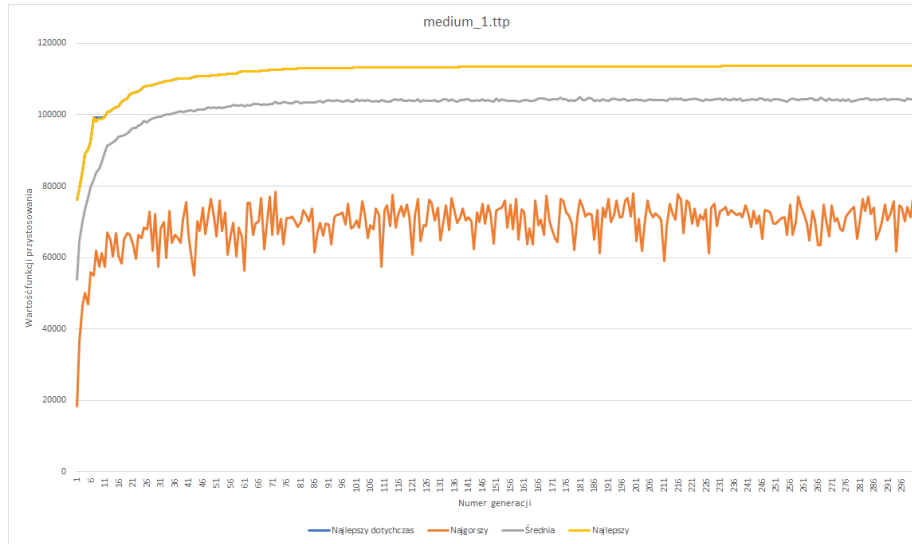


Rysunek 3: Przykładowy przebieg metody dla instancji medium_0

$$\mu = 57284.8$$

$$\sigma = 1046.99$$

4.4 Problem medium_1.ttp

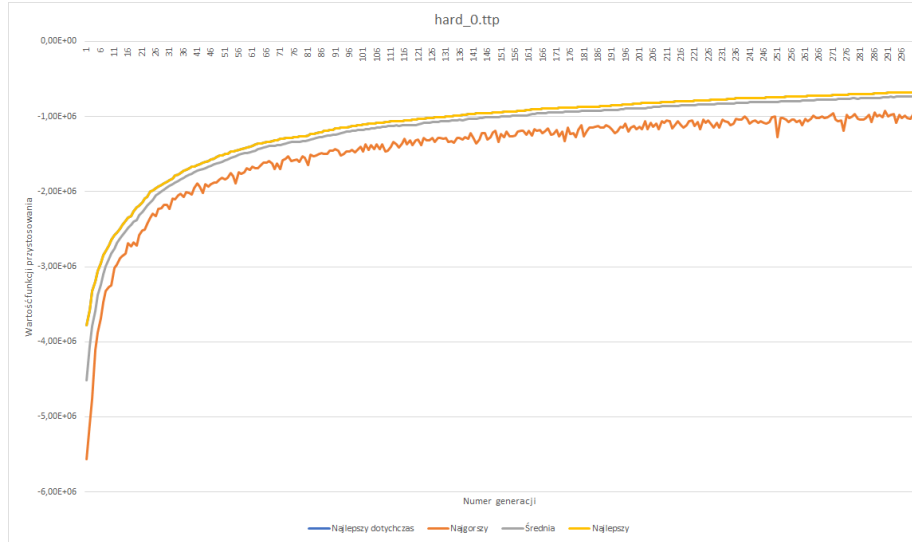


Rysunek 4: Przykładowy przebieg metody dla instancji medium_1

$$\mu = 113726.1$$

$$\sigma = 1692.03$$

4.5 Problem hard_0.ttp



Rysunek 5: Przykładowy przebieg metody dla instancji hard_0

$$\mu = -713010.9$$

$$\sigma = 43227.52$$

4.6 Wnioski

Z powyższych wykresów można wnioskować, że algorytm genetyczny działa poprawnie. Widać że wartość średnia jest oddalona od najlepszego rozwiązania, a wartość przystosowania najgorszego osobnika w populacji jest znacząco oddalona od średniej, co może oznaczać, że przestrzeń rozwiązań jest poprawnie eksplorowana. Z drugiej strony można zauważyć, że dla czterech pierwszych instancji problemu TTP, algorytm zdaje się utykać i nie znajdować lepszych rozwiązań po mniejszej liczbie iteracji niż została przewidziana. Być może lepiej byłoby zatem zmniejszyć ten parametr i uruchomić metodę więcej razy lub inaczej dobrać wartości innych parametrów. Z kolei na wykresie przebiegu dla instancji hard_0 widać, że nawet po upływie 300 generacji, dalej zauważalna jest tendencja wzrostowa, co oznacza że dobrze byłoby próbować szukać rozwiązania dłużej.

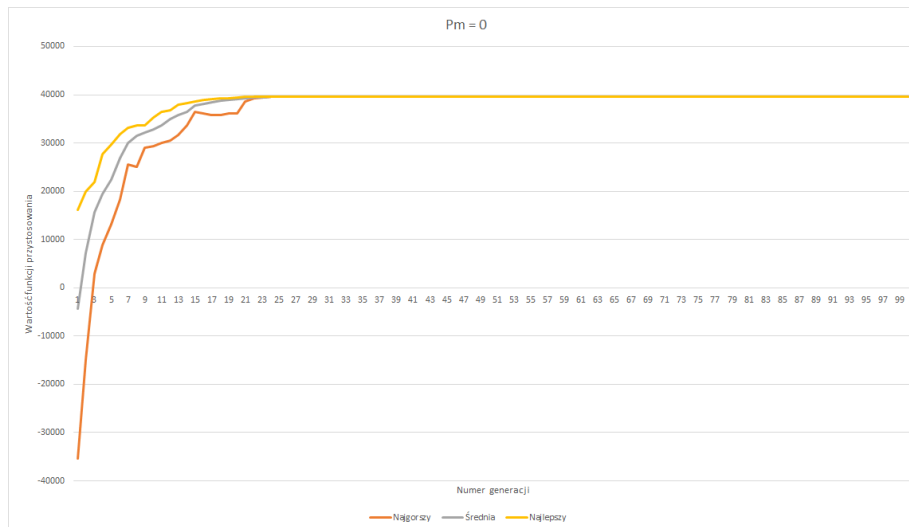
5 Badanie wpływu parametrów metody na wyniki działania

W tej sekcji przedstawione zostaną wyniki badań dotyczących wpływu wartości poszczególnych parametrów na wyniki działania algorytmu genetycznego. Dla każdej wartości parametru zostało wykonane 10 uruchomień.

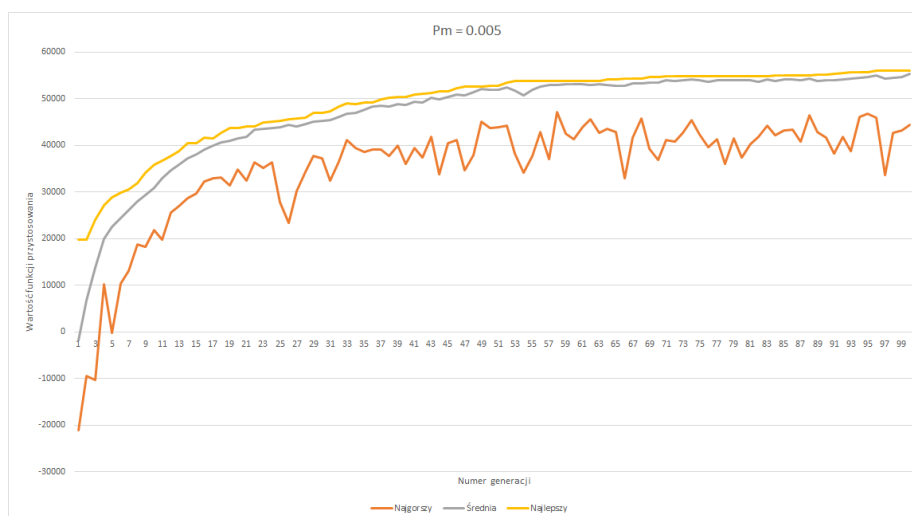
5.1 Wpływ prawdopodobieństwa mutacji

Pierwszy parametr, który zostanie omówiony to prawdopodobieństwo mutacji - p_m . Badanie zostało przeprowadzone dla problemu medium_0.ttp, a wartości innych parametrów zostały ustawione następująco:

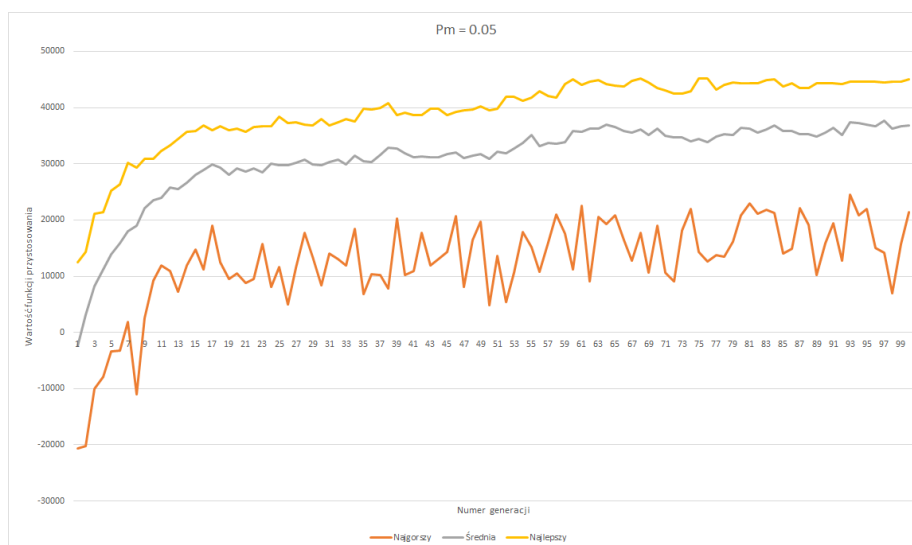
- $gen = 100$,
- $pop_size = 100$,
- $p_x = 0.8$,
- $tour = 10$.



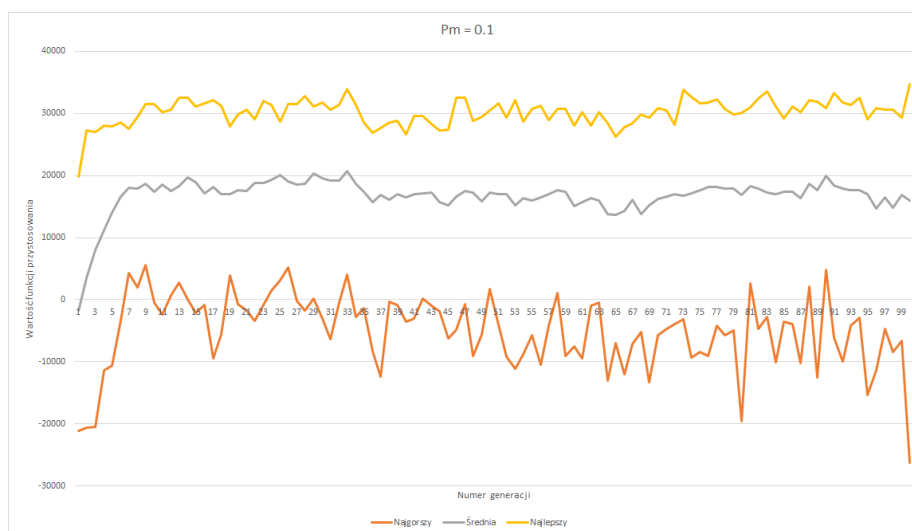
Rysunek 6: Przykładowy przebieg dla $p_m = 0$



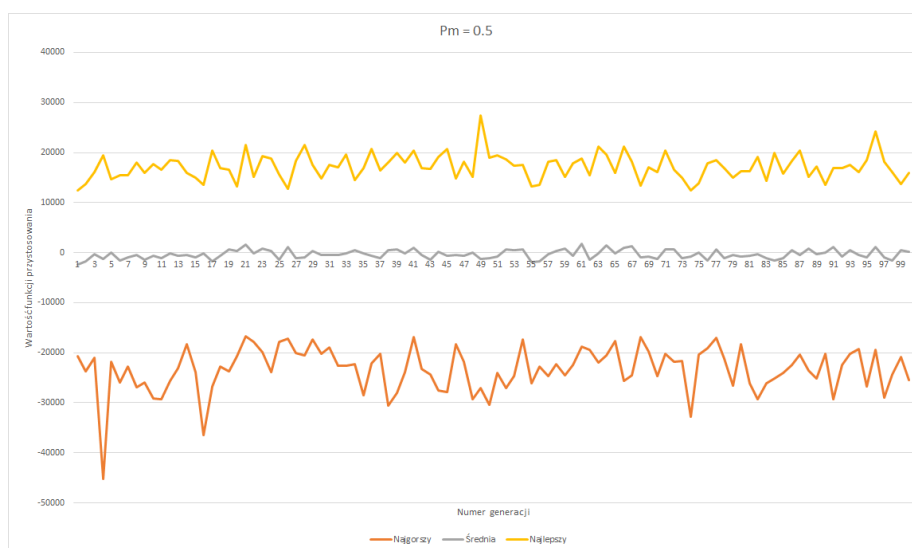
Rysunek 7: Przykładowy przebieg dla $p_m = 0.005$



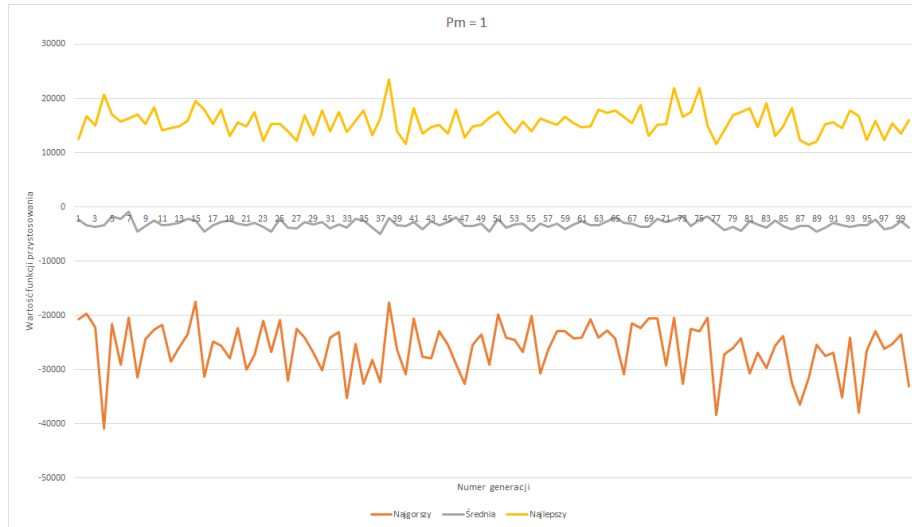
Rysunek 8: Przykładowy przebieg dla $p_m = 0.05$



Rysunek 9: Przykładowy przebieg dla $p_m = 0.1$



Rysunek 10: Przykładowy przebieg dla $p_m = 0.5$



Rysunek 11: Przykładowy przebieg dla $p_m = 1$

p_m	μ	σ
0	38837.3	4054.36
0.005	53087.7	1900.79
0.05	45279.2	1644.11
0.1	36578.1	1425.52
0.5	25012.4	1342.05
1	23004.6	1202.17

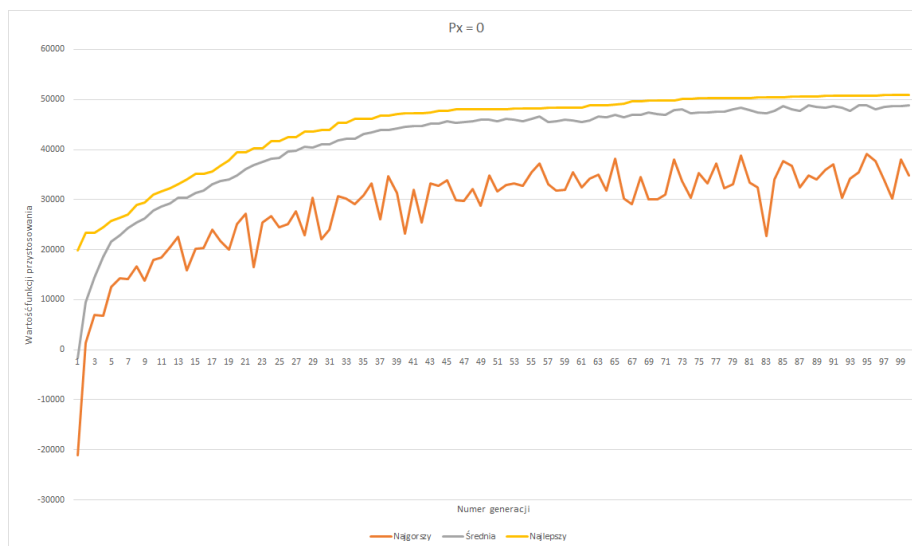
Tabela 1: Średnia wartość i odchylenie standardowe funkcji przystosowania najlepszego znalezionej osobnika, dla różnych wartości parametru p_m .

Wnioski Jak widać na rysunku 6 przy prawdopodobieństwie mutacji $= 0$, po pewnej liczbie iteracji, cała populacja zbiega do jednego punktu. Brakuje źródła różnorodności. Z kolei już na rysunku 8 i w tabeli 1 widać, że prawdopodobieństwo mutacji większe od 0.05 zaczyna stopniowo pogarszać jakość wyników. Przy zbyt dużych wartościach tego prawdopodobieństwa przeszukiwanie staje się zbyt losowe. Przy tak zdefiniowanym operatorze mutacji, dla problemu medium_0, wartość około pomiędzy 0.005, a 0.015 zdaje się być odpowiednia.

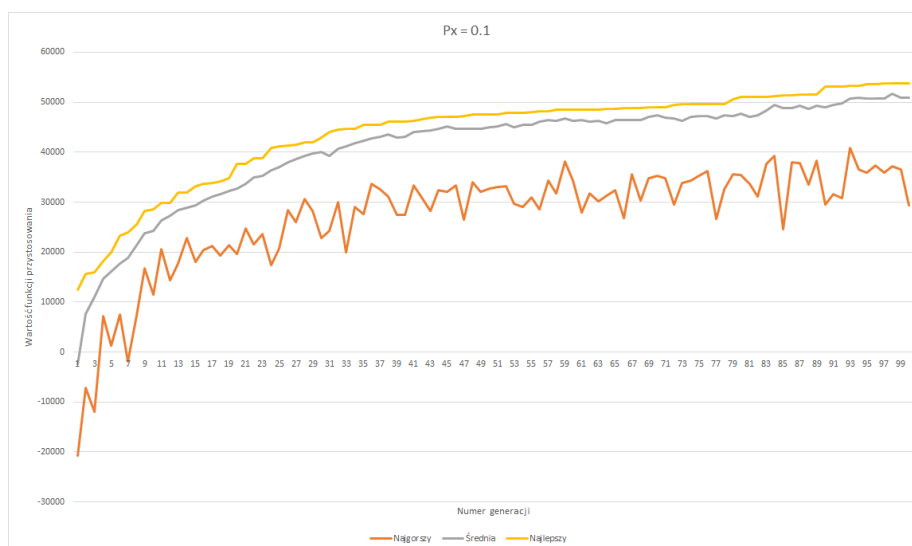
5.2 Wpływ prawdopodobieństwa krzyżowania

Kolejny parametr, który zostanie omówiony to prawdopodobieństwo krzyżowania p_x . Badanie również zostało przeprowadzone na problemie `medium_0`, a ustalone wartości parametrów wyglądały następująco:

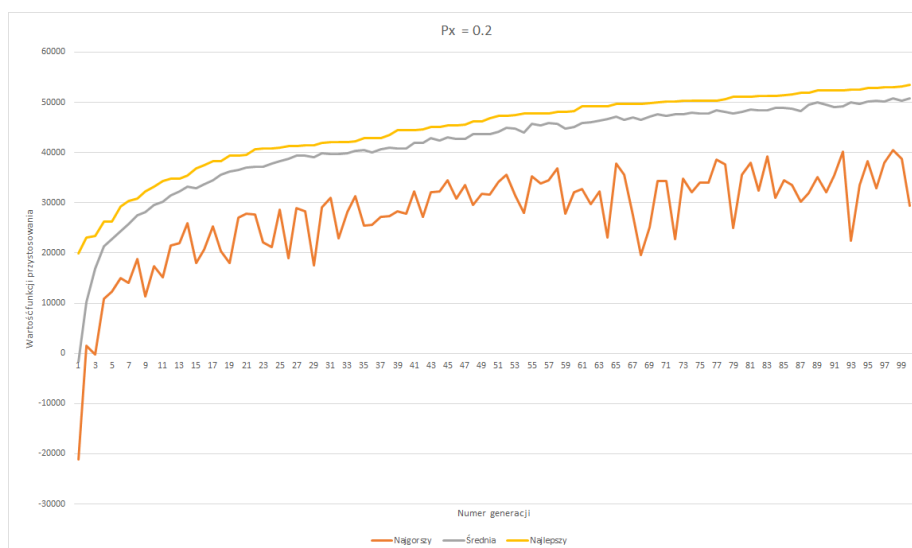
- $gen = 100$,
- $pop_size = 100$,
- $p_m = 0.015$,
- $tour = 10$.



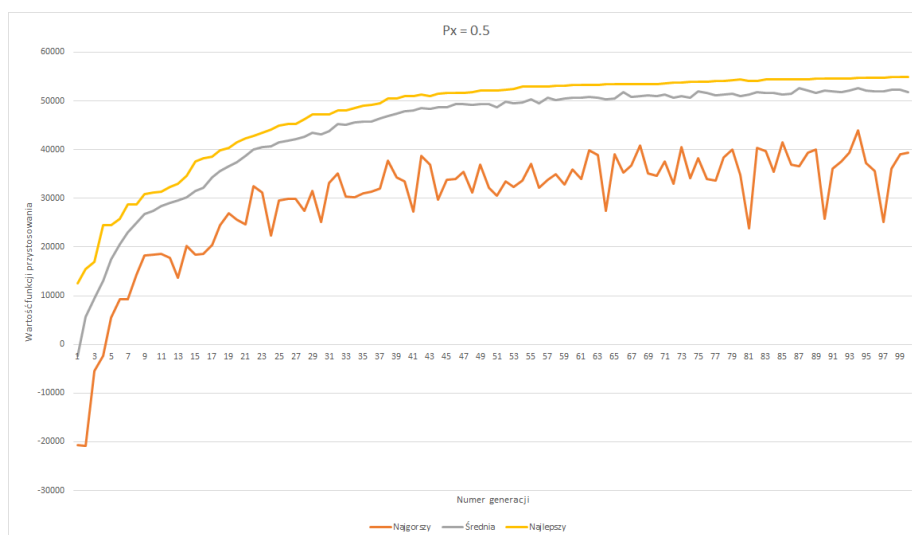
Rysunek 12: Przykładowy przebieg dla $p_x = 0$



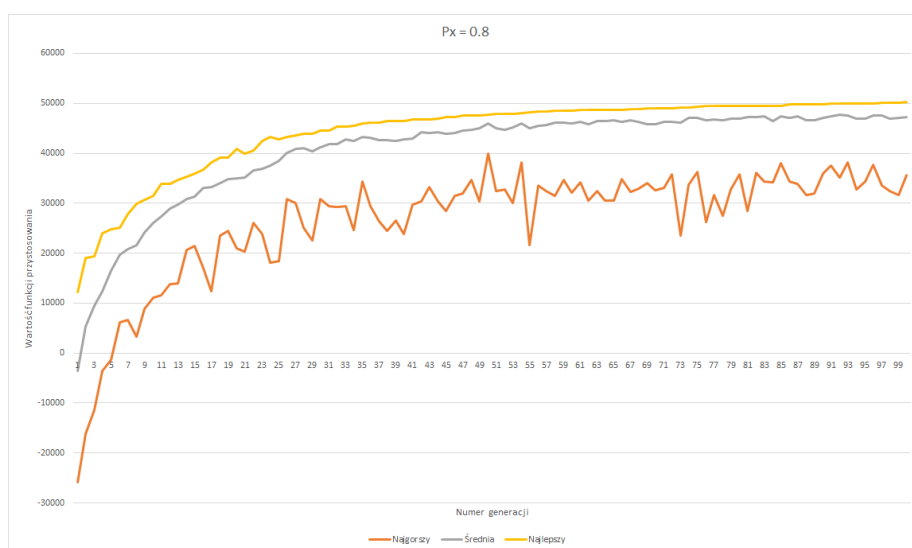
Rysunek 13: Przykładowy przebieg dla $p_x = 0.1$



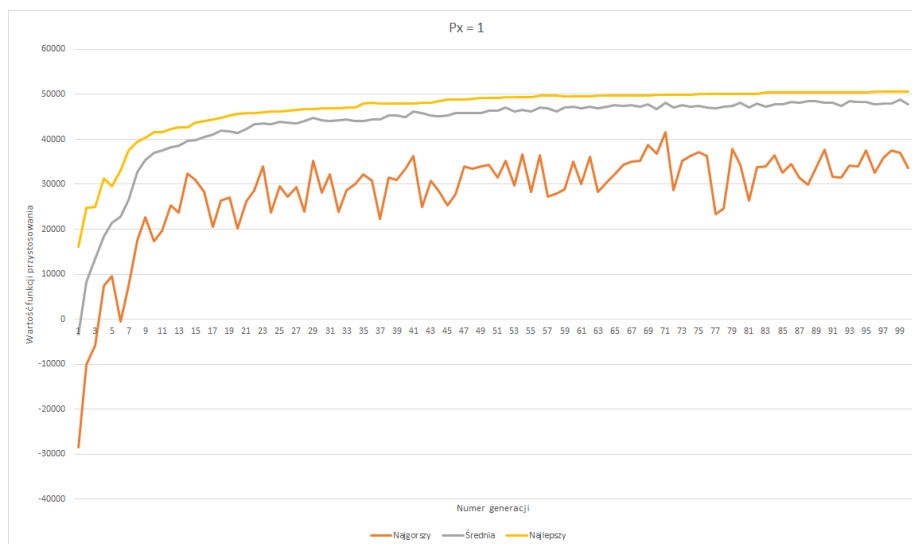
Rysunek 14: Przykładowy przebieg dla $p_x = 0.2$



Rysunek 15: Przykładowy przebieg dla $p_x = 0.5$



Rysunek 16: Przykładowy przebieg dla $p_x = 0.8$



Rysunek 17: Przykładowy przebieg dla $p_x = 1$

p_x	μ	σ
0	51535.7	984.14
0.1	52316.9	1982.64
0.2	51933.6	1498.61
0.5	52560.9	1942.67
0.8	52271.5	1604.25
1	53329.6	1942.71

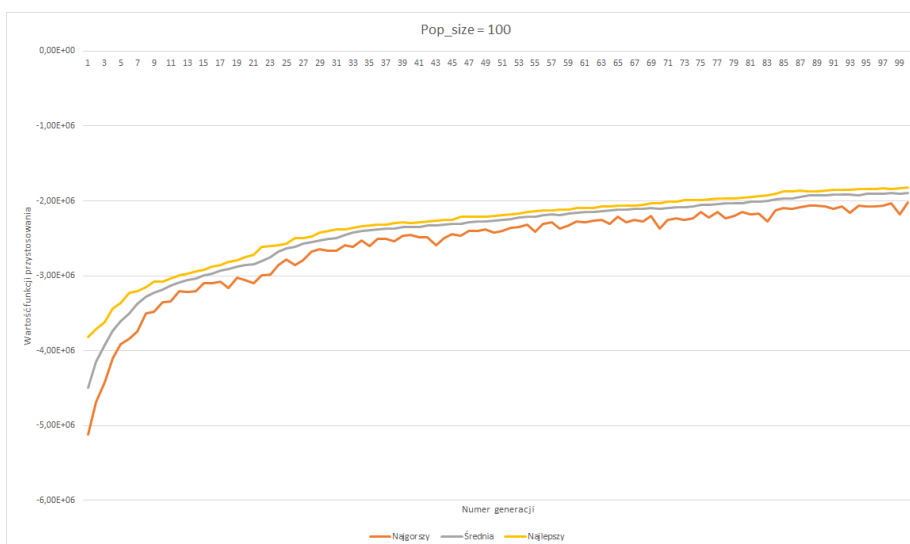
Tabela 2: Średnia wartość i odchylenie standardowe funkcji przystosowania najlepszego znalezionej osobnika, dla różnych wartości parametru p_x .

Wnioski Wnioskowanie na temat wpływu prawdopodobieństwa krzyżowania na jakość znajdowanych rozwiązań, nie jest w przypadku tego problemu tak oczywista jak w przypadku operatora mutacji. Na podstawie uzyskanych wyników nie da się łatwo stwierdzić jaka powinna być ta wartość, aby otrzymywane wyniki były jak najlepsze.

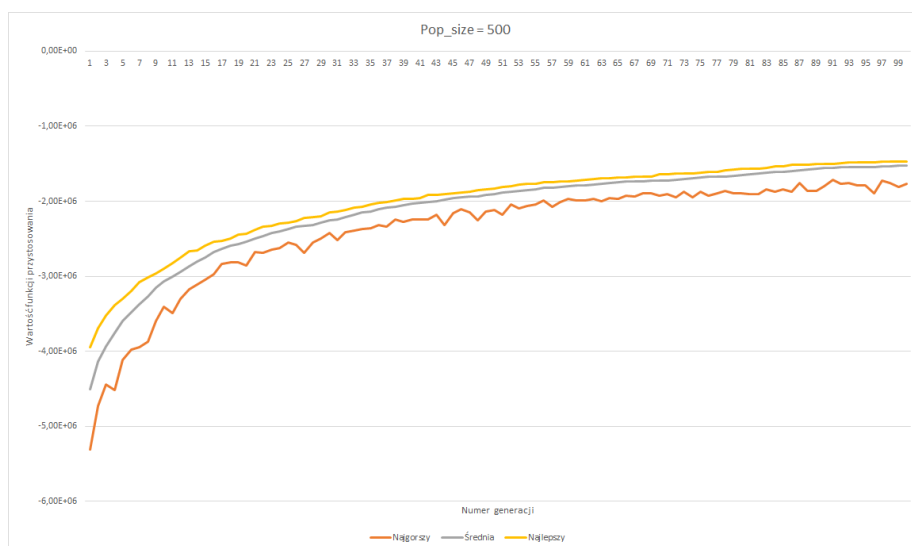
5.3 Wpływ rozmiaru populacji

W następnej kolejności zostaną omówione wyniki badań nad wpływem rozmiaru populacji na jakość znajdowanych rozwiązań. To badanie zostało przeprowadzone na instancji `hard_0`. Ze względu na silną korelację parametrów `pop_size` i `tour`, w tym badaniu były zmieniane wartości ich obu. Wynika to z faktu że stosunek tych dwóch wartości steruje presją selekcyjną. Wartości reszty parametrów zostały ustalone następująco:

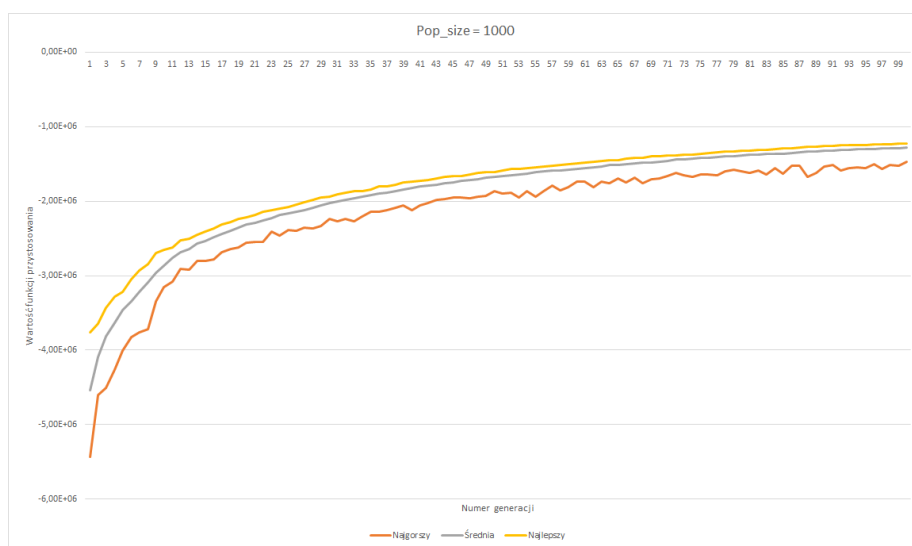
- $gen = 100$,
- $p_m = 0.005$,
- $p_x = 0.8$.



Rysunek 18: Przykładowy przebieg dla `pop_size = 100`



Rysunek 19: Przykładowy przebieg dla $pop_size = 500$



Rysunek 20: Przykładowy przebieg dla $pop_size = 1000$

<i>pop_size</i>	μ	σ
100	-1843672	66598,53
500	-1401129	49426,29
1000	-1194559	17964,83

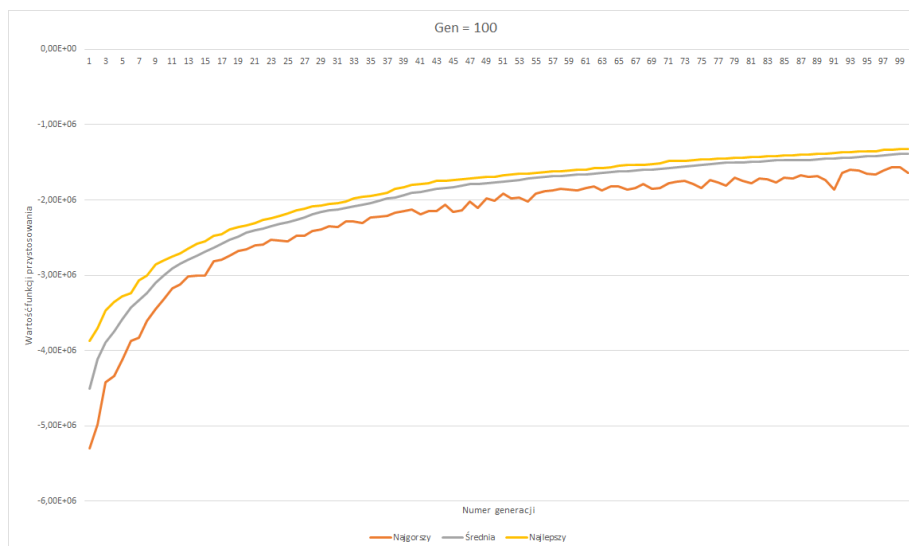
Tabela 3: Średnia wartość i odchylenie standardowe funkcji przystosowania najlepszego znalezionej osobnika, dla różnych wartości parametru *pop_size*.

Wnioski Z tabeli 3 dość dobrze widać, jak istotnym parametrem jest wielkość populacji. Dla problemu *hard_0*, przy zwiększaniu wartości tego parametru znacząco poprawia się zarówno średnia jak i odchylenie standardowe. Wielkość populacji ma bezpośredni wpływ na liczbę przeglądanych przez algorytm rozwiązań.

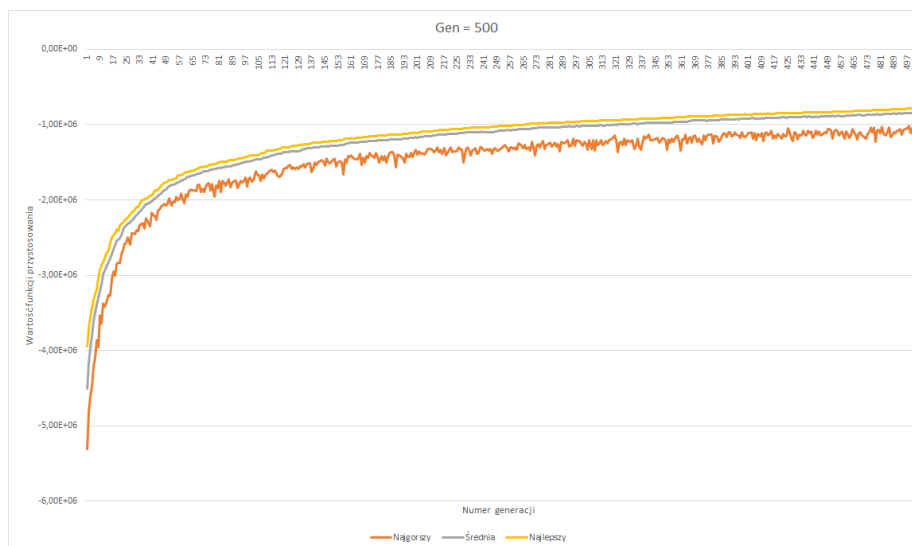
5.4 Wpływ liczby pokoleń

Ostatnim omawianym parametrem będzie liczba pokoleń. To badanie, tak jak poprzednie, zostało przeprowadzone na problemie `hard_0`. Wartości parametrów zostały ustalone następująco:

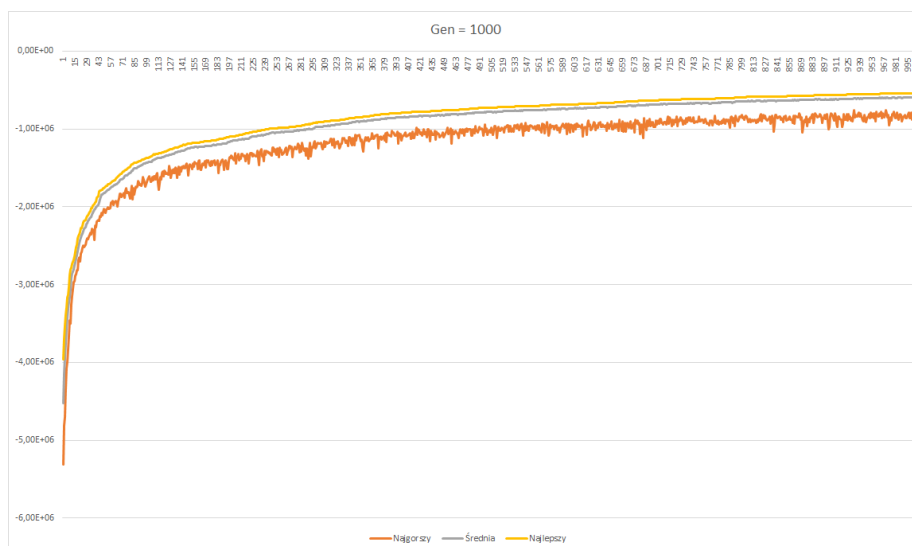
- $pop_size = 500$
- $p_m = 0.005$,
- $p_x = 0.8$,
- $tour = 25$.



Rysunek 21: Przykładowy przebieg dla $gen = 100$



Rysunek 22: Przykładowy przebieg dla $gen = 500$



Rysunek 23: Przykładowy przebieg dla $gen = 1000$

<i>pop_size</i>	μ	σ
100	-1390237,0	57115,43
500	-741901,7	44242,51
1000	-527889,9	47903,73

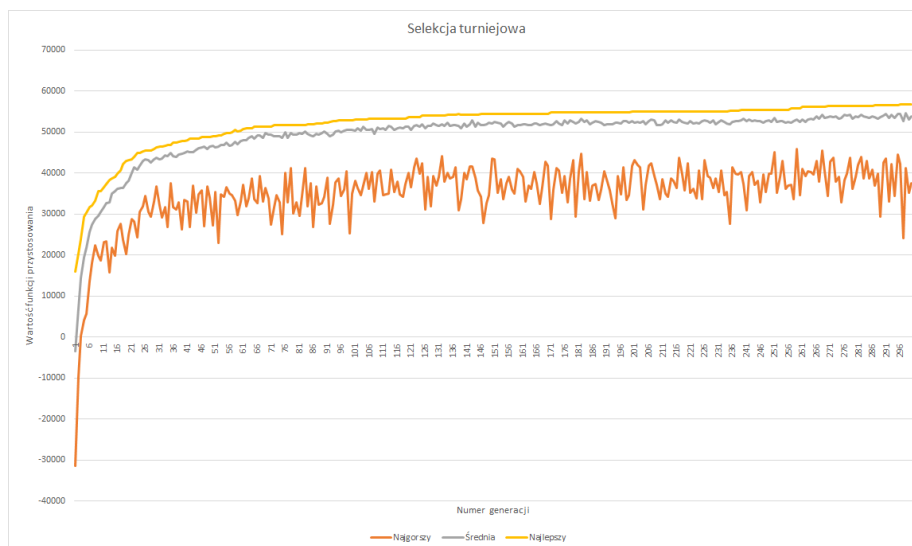
Tabela 4: Średnia wartość i odchylenie standardowe funkcji przystosowania najlepszego znalezionej osobnika, dla różnych wartości parametru *gen*.

Wnioski Z otrzymanych wyników dobrze widać, że tak jak wielkość populacji, liczba generacji ma bardzo duży wpływ na jakość znalezionych rozwiązań.

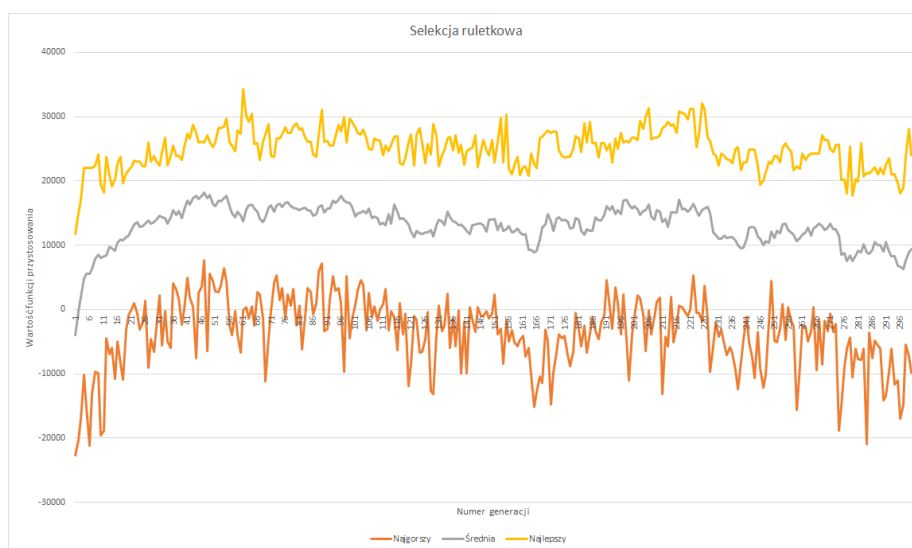
6 Badanie wpływu selekcji na skuteczność algorytmu genetycznego

W tej sekcji omówione zostaną wyniki badań wpływu dwóch rodzajów selekcji na skuteczność algorytmu genetycznego. Oba zostaną przeprowadzone na problemie *medium.0* z parametrami:

- *gen* = 300
- *pop_size* = 100
- $p_m = 0.015$,
- $p_x = 0.8$,
- *tour* = 10.



Rysunek 24: Przykładowy przebieg dla selekcji turniejowej



Rysunek 25: Przykładowy przebieg dla selekcji ruletkowej

Rodzaj selekcji	μ	σ
turniej	56370.1	1264.43
ruletka	32256.8	988.66

Tabela 5: Średnia wartość i odchylenie standardowe funkcji przystosowania najlepszego znalezionej osobnika, dla różnych rodzajów selekcji.

Wnioski Z rysunków 24 i 25 widać, że selekcja turniejowa daje zdecydowanie lepsze rezultaty niż selekcja ruletkowa. Ruletka bez żadnych modyfikacji zdaje się być zbyt losowa i nie dawać tak dobrze ukierunkowanego przeszukiwania jak turniej.

7 Porównanie skuteczności algorytmu genetycznego z wynikami metod nieewolucyjnych

Do porównania zostały wybrane następujące algorytmy nieewolucyjne:

- przeszukiwanie losowe,
- algorytm wspinaczkowy (*Hill Climber*),
- algorytm zachłanny.

Algorytm	easy_0	easy_1	medium_0	medium_1	hard_0
GA	-4015	2265	58877	115762	-713010
RS	-28520.7	-22798.6	26485.6	83114.4	-3581490
HC	-3659.34	1864.12	58884.7	116226	-
Zachłanny	-18206	-19556.3	47730.8	105551	61718.2

Tabela 6: Zestawienie jakości rozwiązań problemów znalezionych przy pomocy poszczególnych algorytmów.

Algorytm	easy_0	easy_1	medium_0	medium_1
GA	87741	105581	262211	129206
HC	$6.24 \cdot 10^6$	$7.88 \cdot 10^6$	$2.23 \cdot 10^6$	$5.19 \cdot 10^6$
Zachłanny	51	52	52	52

Tabela 7: Liczba ewaluacji funkcji celu do znalezienia podanego w tabeli 6 rozwiązania.

Wnioski Algorytm przeszukiwania losowego w żadnym stopniu nie sprawdza się do rozwiązywania takich problemów jak ten. Algorytm wspinaczkowy z kolei, znalazł najlepsze rozwiązania dla 3 z 5 problemów. Posiada on jednak jedną zasadniczą wadę - wymaga ogromnej liczby ewaluacji funkcji celu. Warte uwagi jest również to, że algorytm zachłanny znalazł najlepsze rozwiązanie problemu hard_0. Jego dużą zaletą jest to, że do problemu TSP wymaga tylko N ewaluacji funkcji celu, gdzie N jest liczbą możliwych miast początkowych.

Bardzo istotny jest również fakt, że algorytm genetyczny znalazł rozwiązania niewiele gorsze niż algorytm wspinaczkowy, a osiągnął to dużo mniejszym kosztem. Prawdopodobnie gdyby dobrać lepsze operatory genetyczne i przeprowadzić dokładniejsze strojenie parametrów, algorytm genetyczny poradziłby sobie najlepiej we wszystkich problemach.

8 Podsumowanie

Algorytm genetyczny jest bardzo dobrą metodą optymalizacji do problemów, w których występują duże przestrzenie rozwiązań. Pozwala on znajdować dobre jakościowo rozwiązania dużo mniejszym kosztem niż inne metody optymalizacji. Jego największą wadą jest prawdopodobnie duża liczba silnie powiązanych parametrów, które do każdego problemu należy dobrać osobno, tak aby algorytm działał dobrze.