

*Zadanie projektowe nr 3.*

**Implementacja i analiza efektywności Alogrytmu  
Genetycznego dla problemu komiwojażera.**

Sprawozdanie z zadania projektowego przedmiotu „Projektowanie efektywnych algorytmów”.

Rok akademicki 2019/2020, kierunek Informatyka.

*Prowadzący:*

Dr inż. Zbigniew Buchalski

*Termin zajęć:*

Wtorek, 9.15

## 1. Wstęp teoretyczny

Celem wykonania trzeciego zadania projektowego była implementacja oraz analiza efektywności Algorytmu Genetycznego dla asymetrycznego problemu komiwojażera. Jest to algorytm heurystyczny a więc umożliwia on uzyskanie rozwiązania problemu bez gwarancji jego dokładności i optymalności. Jego zasada działania została zainspirowana zachodzącymi w biologii procesami ewolucji. Kluczowym elementem tego algorytmu jest osobnik, któremu dla problemu komiwojażera odpowiada pojedynczy punkt (czyli ścieżka) w przeszukiwanej przestrzeni rozwiązań, zwany również chromosomem lub genotypem. Zbiór osobników nazywany jest populacją, będącą podstawą dla procesu ewolucji odwzorowywanego przez Algorytm Genetyczny. Składa się on następujących elementów:

1. **Generowanie populacji początkowej** – najpierw wytwarzany jest zbiór osobników, będących losowymi rozwiązaniami znajdującymi się w przeszukiwanej przestrzeni.
2. **Ocena osobników** – na tym etapie, oceniana jest jakość osobników znajdujących się w populacji. Sposób w jaki jest ona przeprowadzana zależy od rozpatrywanego problemu. Dla problemu komiwojażera jest to całkowita długość/koszt ścieżki i ocena jej jest tym wyższa im niższy jest koszt jej przebycia.
3. **Selekcja** – jest to najważniejszy element całego algorytmu. Polega ona na wybieraniu z populacji osobników najlepiej przystosowanych, które znajdą się w następnej populacji (pokoleniu).
4. **Krzyżowanie** – proces ten polega na łączeniu ze sobą dwóch osobników celem wytworzenia potomków zastępujących ich w populacji.
5. **Mutowanie** – z bieżącej populacji wybierana jest pewna ilość osobników poddawanych losowym modyfikacjom. Zadaniem tego procesu jest wspomaganie eksploracji przestrzeni rozwiązań.
6. **Sprawdzenie warunku stopu** – jeżeli dobiegł końca okres czasu wyznaczony na wykonanie algorytmu lub wykonana została zadana ilość iteracji, to z populacji bieżącej zwracany zostaje najlepszy osobnik. W przeciwnym wypadku następuje powrót do punktu drugiego.

## 2. Implementacja

### 2.1. Selekcja

Podczas wykonywania zadania projektowego należało zaimplementować jedną ze strategii selekcji osobników. Wybrana została selekcja turniejowa. Polega ona na wyborze z populacji kilku osobników i wyłonienia spośród nich najlepiej przystosowanego, czyli takiego o najwyższej wartości funkcji oceny. Dobór osobników uczestniczących w turnieju odbywa się w sposób losowy. Proces selekcji kończy się po utworzeniu ze zwycięskich osobników nowej populacji.

### 2.2. Krzyżowanie

Dla implementowanego algorytmu zostały wykorzystane następujące schematy krzyżowania dwupunktowego osobników:

**Ordered Crossover, OX** – istotą tej odmiany krzyżowania jest tworzenie potomków składających się z odpowiadających sobie losowej długości fragmentów rodziców a następnie uzupełnianiu ich brakującymi elementami w takiej samej kolejności w jakiej występują one u drugiego rodzica (tego z którego nie został odziedziczony bazowy fragment).

Przykład:

- Wybór krzyżowanych osobników

$$R1 = ( 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 )$$
$$R2 = ( 5 \ 3 \ 6 \ 7 \ 8 \ 1 \ 2 \ 9 \ 4 )$$

- Wylosowanie fragmentu wspólnego

$$R1 = ( 1 \ 2 \ \color{red}{3 \ 4 \ 5 \ 6} \ 7 \ 8 \ 9 )$$
$$R2 = ( 5 \ 3 \ \color{red}{6 \ 7 \ 8 \ 1} \ 2 \ 9 \ 4 )$$

- Utworzenie potomków z bazowych fragmentów

$$P1 = ( \_ \_ \color{red}{3 \ 4 \ 5 \ 6} \_ \_ \_ )$$
$$P2 = ( \_ \_ \color{red}{6 \ 7 \ 8 \ 1} \_ \_ \_ )$$

- Uzupełnienie potomków o brakujące elementy w kolejności odpowiadającej drugiemu rodzicowi

P1 = ( 8 1 3 4 5 6 2 9 7 )

P2 = ( 4 5 6 7 8 1 9 2 3 )

**Partially Mapped Crossover, PMX** – analogicznie do poprzedniego schematu, podstawą tego krzyżowania jest dziedziczenie przez potomków odpowiadających sobie fragmentów rodziców. Inaczej natomiast przebiega proces ich uzupełniania o brakujące elementy.

- Po określeniu losowej sekcji dopasowania, dla każdego z potomków tworzona jest tablica odwzorowań elementów

R1 = ( 1 2 3 4 5 6 7 8 9 )

R2 = ( 5 3 6 7 8 1 2 9 4 )

Tablica odwzorowań dla pierwszego potomka: [3 => 6][4 => 7][5 => 8][6 => 1]

- Następnie dodawane zostają brakujące elementy z drugiego rodzica które nie powodują konfliktów

P1 = ( \_ \_ 3 4 5 6 2 9 \_ )

- Dodawanie elementów powodujących konflikty odbywa się na podstawie tablicy odwzorowań. Elementami powodującymi konflikt jest 4, 5 oraz 3.

- 4 przechodzi na 7

P1 = ( \_ \_ 3 4 5 6 2 9 7 )

- 5 przechodzi na 8

P1 = ( 8 \_ 3 4 5 6 2 9 7 )

- 3 przechodzi na 6 i 6 przechodzi na 1

P1 = ( 8 1 3 4 5 6 2 9 7 )

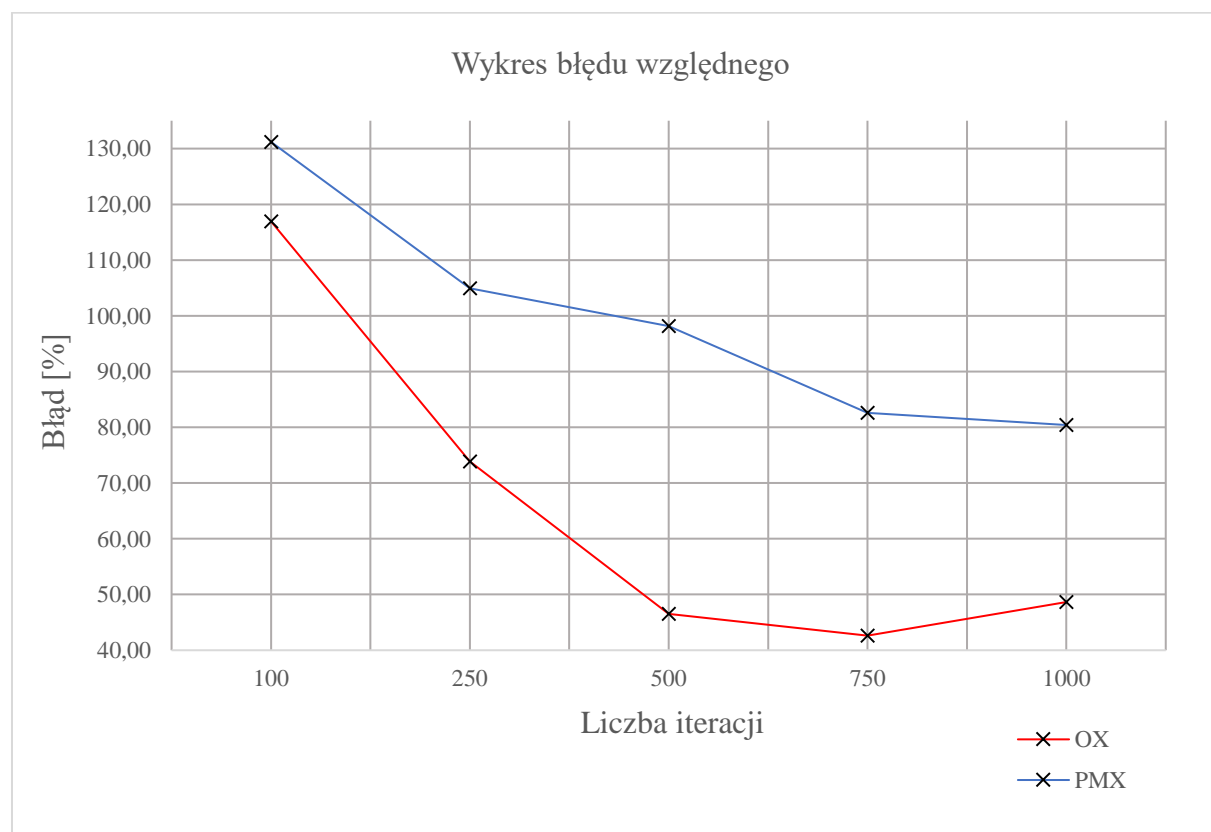
Drugi potomek jest generowany w analogiczny sposób.

### 3. Plan eksperymentu.

Analiza efektywności działania algorytmu została przeprowadzana dla trzech, różnej wielkości instancji problemu komiwojażera. Dla każdego z zadanych plików należało obliczyć błąd dostarczonego przez algorytm rozwiązania w zależności od ilości wykonanych iteracji, wielkości populacji oraz schematu krzyżowania. Było to 100, 250, 500, 750 oraz 1000 iteracji dla wielkości populacji równych 100, 500 oraz 1000 osobników. Podczas badań, wartość współczynnika krzyżowania została ustawiona na 0.8 a współczynnik mutacji na wartość 0.01. Najlepsze rozwiązania uzyskane za pomocą Algorytmu Genetycznego zostały porównane z najlepszymi rozwiązaniami uzyskanymi metodą Tabu Search.

### 4. Analiza danych pomiarowych.

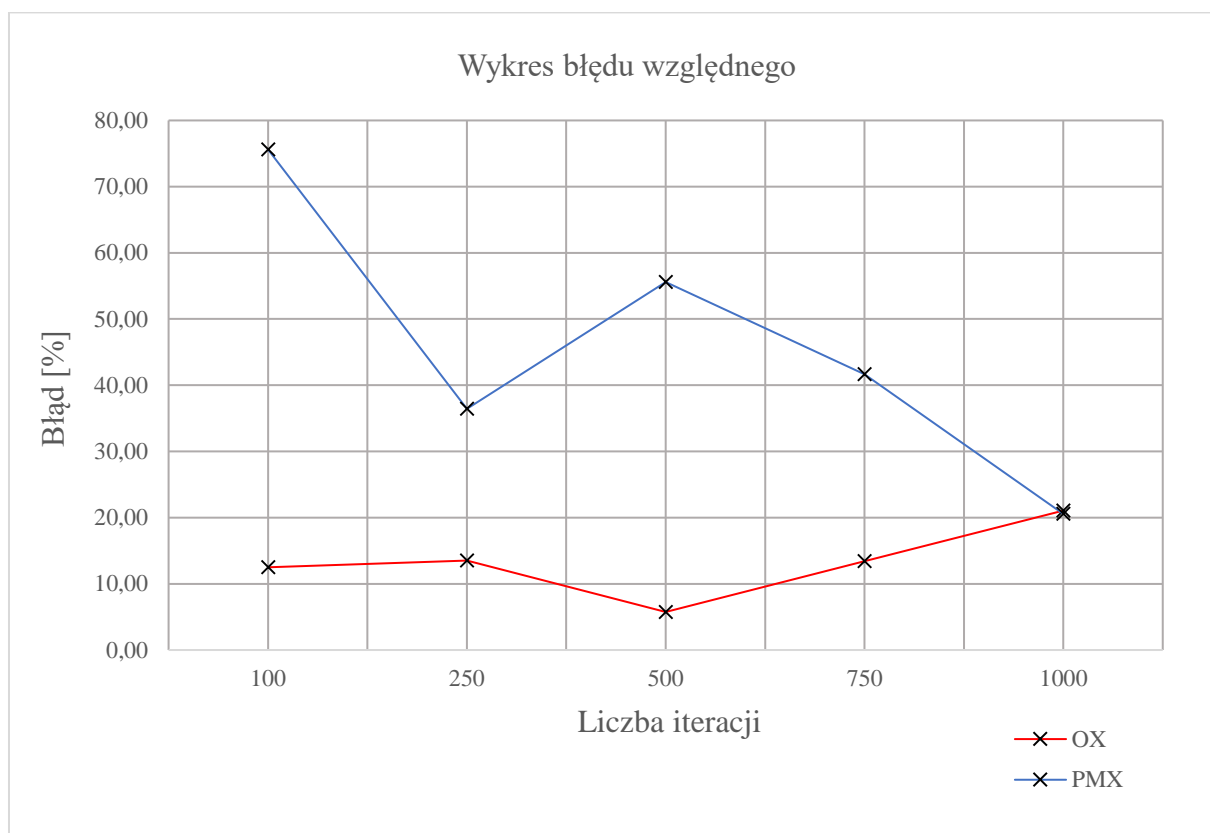
Plik		Wielkość populacji		
ftv47.atsp		100		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	3853	116,95	4106	131,19
250	3088	73,87	3640	104,95
500	2602	46,51	3519	98,14
750	2533	42,62	3243	82,60
1000	2640	48,65	3204	80,41



Plik		Wielkość populacji		
ftv47.atsp		500		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	2338	31,64	2814	58,45
250	2073	16,72	2541	43,07
500	2083	17,29	2402	35,25
750	2012	13,29	2513	41,50
1000	2050	15,43	2710	52,59



Plik		Wielkość populacji		
ftv47.atsp		1000		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	1998	12,50	3119	75,62
250	2016	13,51	2423	36,43
500	1878	5,74	2763	55,57
750	2014	13,40	2516	41,67
1000	2150	21,06	2141	20,55

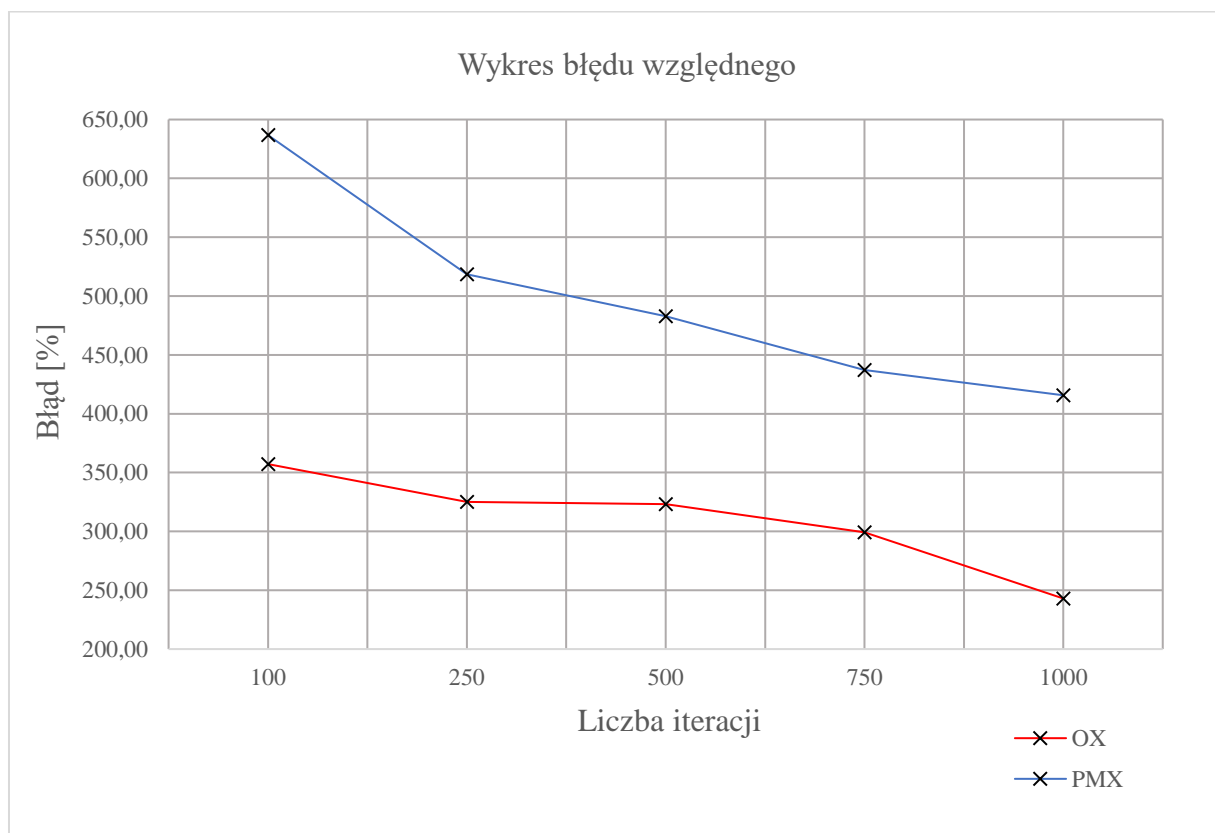


Najlepsze znane rozwiązanie: **1776**

Najlepsze rozwiązanie uzyskane przez Algorytm Genetyczny: **1878**

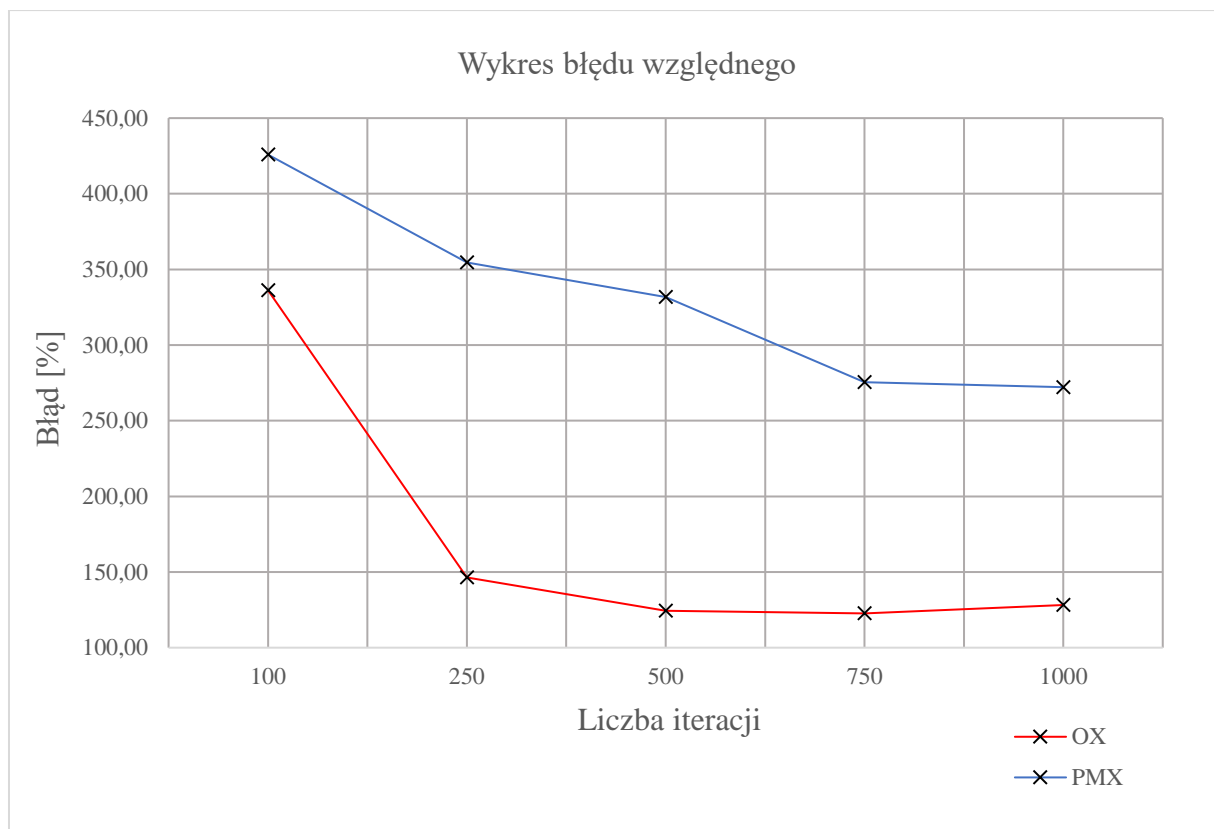
Najlepsze rozwiązanie uzyskane przez Tabu Search: **1886**

Plik		Wielkość populacji		
ftv170.atasp		100		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	12596	357,21	20297	636,73
250	11707	324,94	17037	518,40
500	11658	323,16	16057	482,83
750	11000	299,27	14799	437,17
1000	9449	242,98	14207	415,68

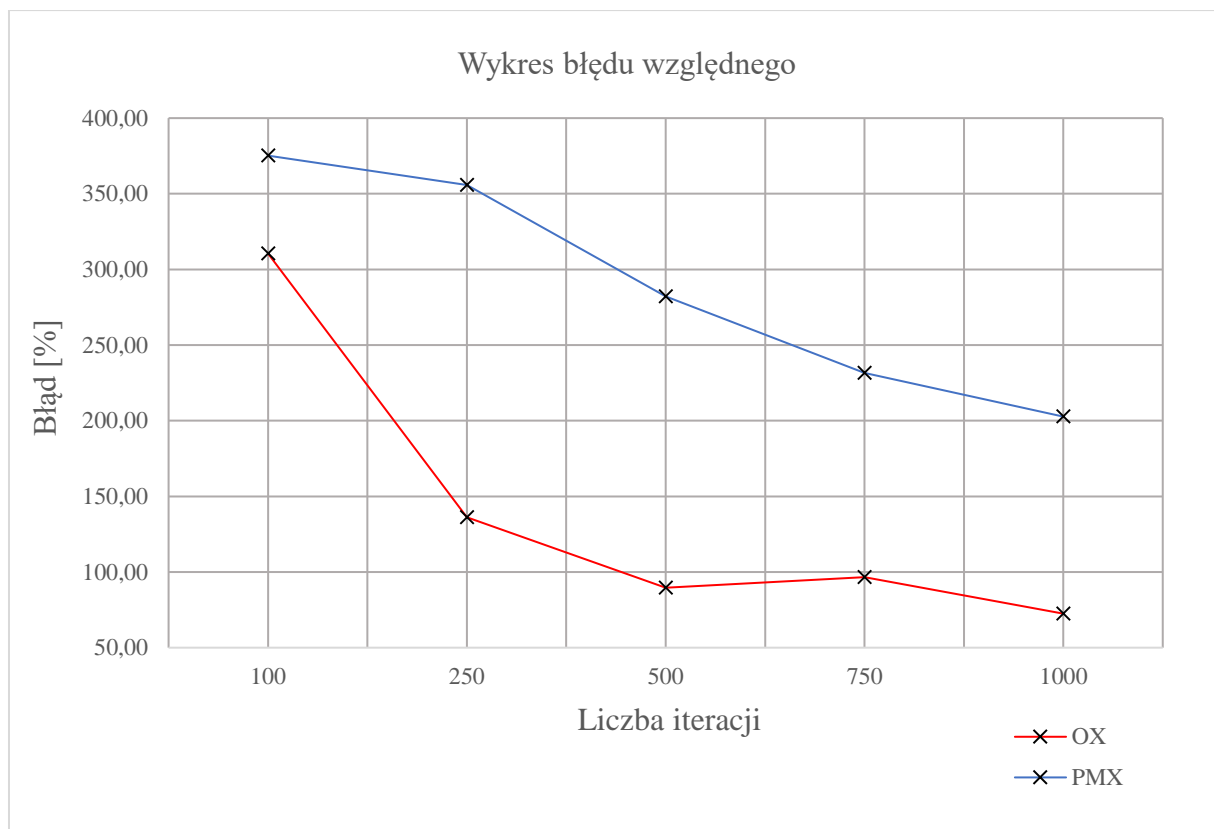




Plik		Wielkość populacji		
ftv170.atsp		500		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	12018	336,23	14488	425,88
250	6790	146,46	12522	354,52
500	6182	124,39	11896	331,80
750	6134	122,65	10343	275,43
1000	6287	128,20	10253	272,16



Plik		Wielkość populacji		
ftv170.atsp		1000		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	11310	310,53	13092	375,21
250	6507	136,19	12556	355,75
500	5223	89,58	10530	282,21
750	5414	96,52	9137	231,65
1000	4752	72,49	8340	202,72

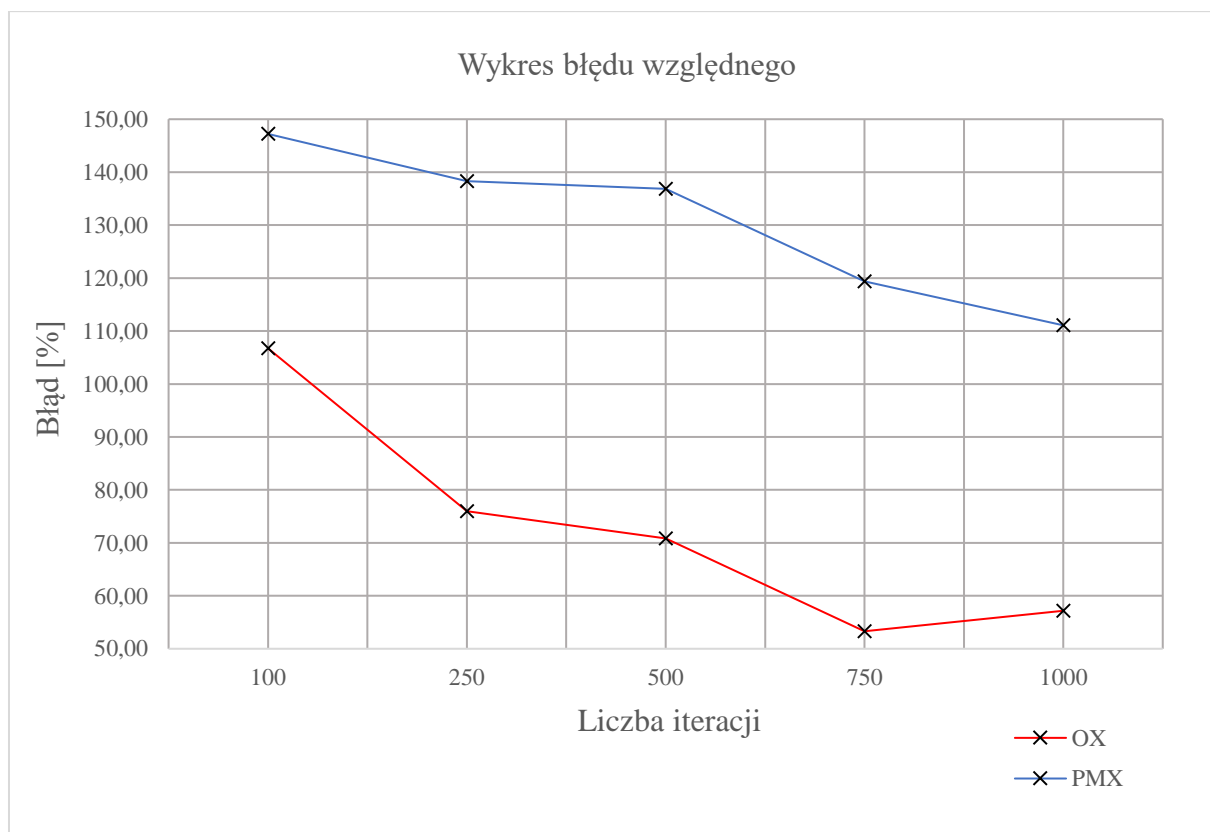


Najlepsze znane rozwiązanie: **2755**

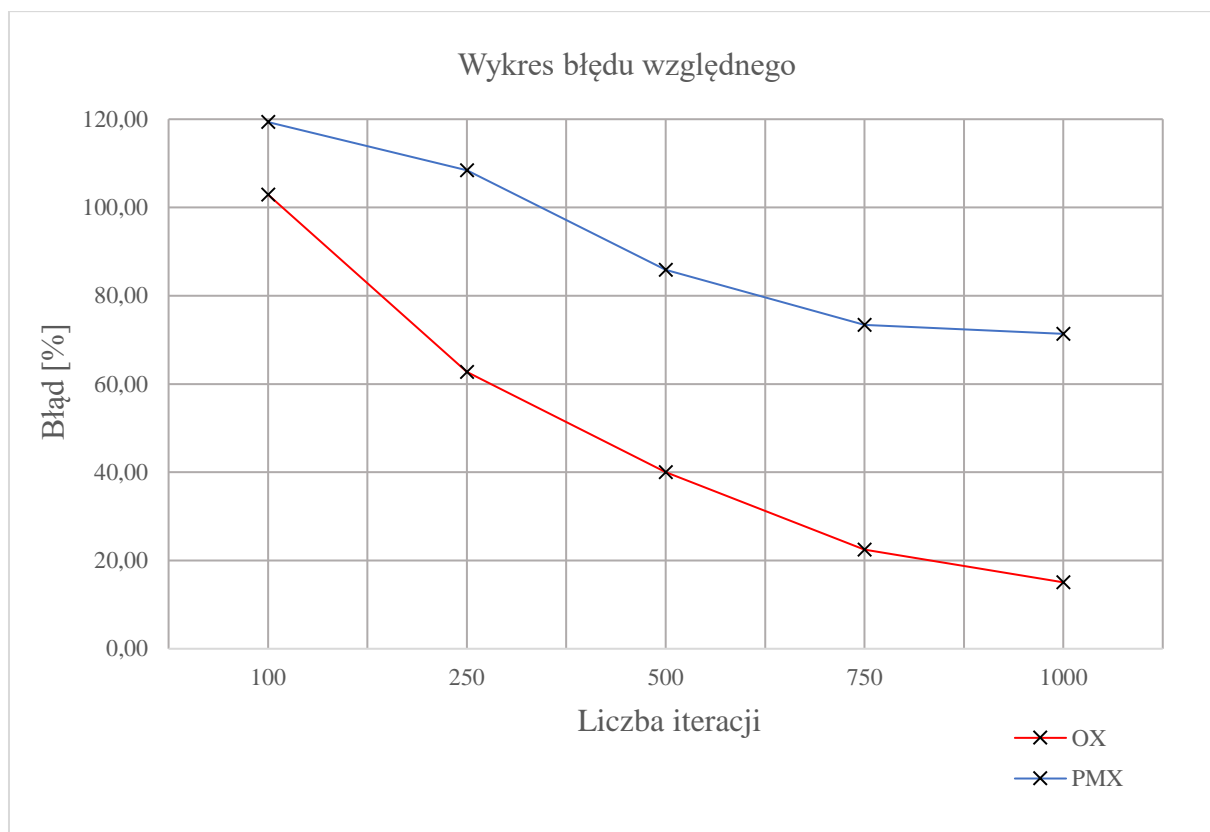
Najlepsze rozwiązanie uzyskane przez Algorytm Genetyczny: **4752**

Najlepsze rozwiązanie uzyskane przez Tabu Search: **8539**

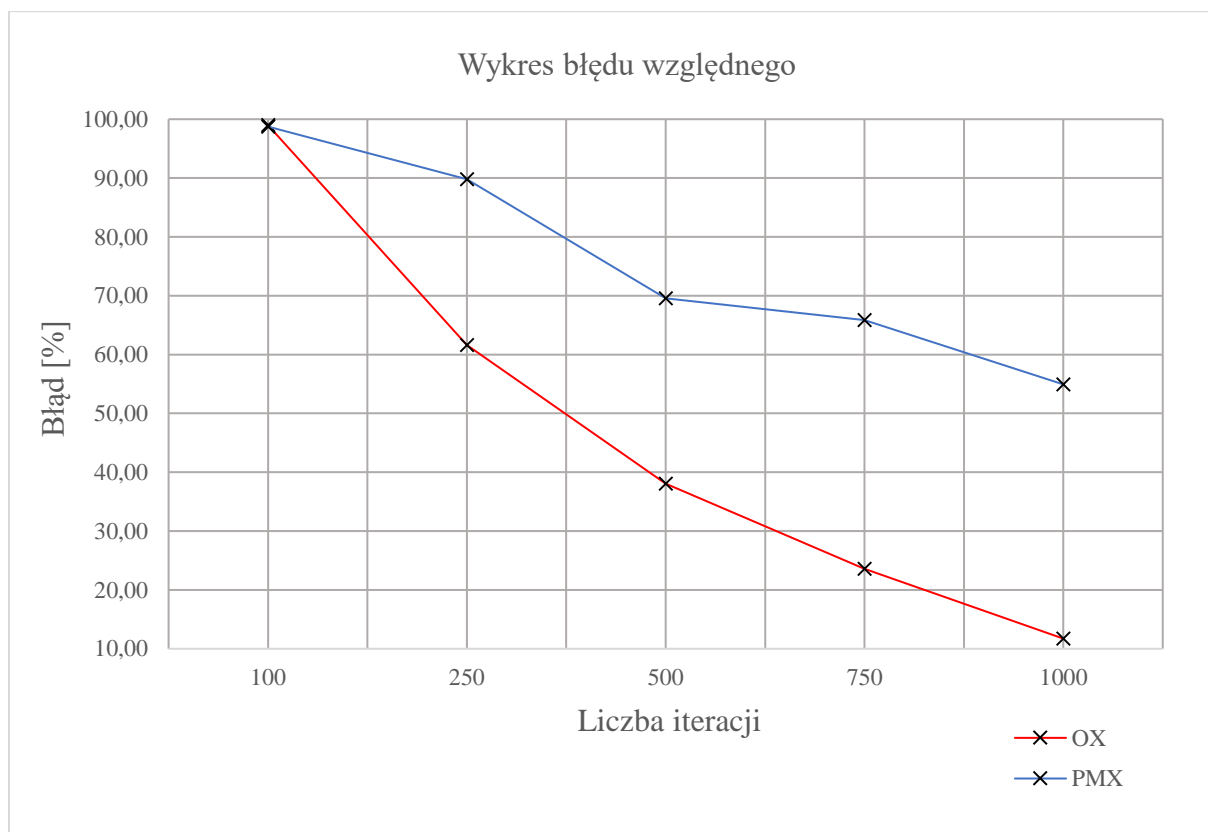
Plik		Wielkość populacji		
rbg403.atsp		100		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	5096	106,73	6094	147,22
250	4338	75,98	5874	138,30
500	4211	70,83	5839	136,88
750	3779	53,31	5408	119,39
1000	3874	57,16	5203	111,08



Plik		Wielkość populacji		
rbg403.atsp		500		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	5002	102,92	5407	119,35
250	4012	62,76	5138	108,44
500	3451	40,00	4582	85,88
750	3019	22,47	4274	73,39
1000	2836	15,05	4224	71,36



Plik		Wielkość populacji		
rbg403.atsp		1000		
	Ordered Crossover		Partially Mapped Crossover	
Ilość iteracji	Rozwiązanie	Błąd [%]	Rozwiązanie	Błąd [%]
100	4905	98,99	4899	98,74
250	3984	61,62	4679	89,82
500	3403	38,05	4179	69,53
750	3046	23,57	4088	65,84
1000	2754	11,72	3819	54,93



Najlepsze znane rozwiązanie: **2465**

Najlepsze rozwiązanie uzyskane przez Algorytm Genetyczny: **2745**

Najlepsze rozwiązanie uzyskane przez Tabu Search: **6678**

## 5. Wnioski.

Analiza wykresów sporządzonych na podstawie zebranych w trakcie badań wyników pozwala zauważyć znaczącą korelację jakości uzyskanego rozwiązania z liczbą iteracji algorytmu oraz wielkością populacji osobników. W prawie każdym przypadku można zauważyć spadek wielkości błędu względem rosnącej liczby iteracji algorytmu oraz rosnącej wielkości populacji. Wyjątek stanowi tutaj zbiór wyników dla małej instancji problemu, wykazujący stabilizowanie się wielkości błędu już dla wielkości populacji niemniejszych od 500. Tendencja ta jest tym silniejsza im większy jest rozmiar instancji problemu. Ponadto, porównana została również efektywność zaimplementowanych schematów krzyżowania. Dla każdej instancji problemu oraz wielkości populacji, znacząco mniejszy błąd był generowany przy krzyżowaniu typu OX. Okazało się ono efektywniejsze od krzyżowania typu PMX. Porównanie uzyskanych podczas badań wyników z wynikami uzyskanymi metodą Tabu Search wskazuje na wyższą efektywność Algorytmu Genetycznego dla odpowiednio dużych populacji oraz ilości iteracji.

## 6. Bibliografia.

- <http://www.alife.pl/gp/p/AGelem.html>
- [http://www.zio.iiar.pwr.wroc.pl/pea/w9\\_ga\\_tsp.pdf](http://www.zio.iiar.pwr.wroc.pl/pea/w9_ga_tsp.pdf)
- [https://pl.wikipedia.org/wiki/Algorytm\\_genetyczny](https://pl.wikipedia.org/wiki/Algorytm_genetyczny)
- [http://home.agh.edu.pl/~vlsi/AI/gen\\_t/](http://home.agh.edu.pl/~vlsi/AI/gen_t/)
- <https://www.youtube.com/watch?v=Pg4HP6Ayijs>