

# Projektowanie Efektywnych Algorytmów

Projekt

18.01.2022

256423 Maciej Radziszewski

( 6 ) Algorytm Genetyczny

Spis treści	strona
Sformułowanie zadania	2
Metoda	3
Algorytm	4
Dane testowe	6
Procedura badawcza	7
Wyniki	8
Analiza wyników i wnioski	16

## 1. Treść zadania

Zadanie polega na implementacji i zbadaniu efektywności algorytmu genetycznego rozwiązującego problem komiwojażera. Polega on na odnalezieniu minimalnego lub maksymalnego (w zależności od wymagań) cyklu Hamiltona w grafie. Graf, który jest brany pod uwagę musi być pełny, oraz jego krawędzie muszą mieć wagi.

## 2. Metoda

Metody ewoluujące, czyli nazwa zbiorcza dla metod takich jak:

- Algorytmy genetyczne,
- Strategie ewolucyjne,
- Programowanie genetyczne.

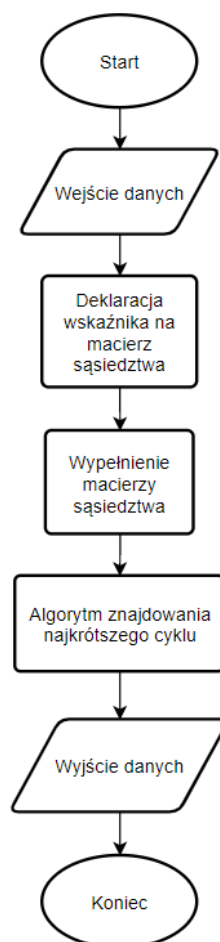
Są to metody reprezentujące odmienne podejście do poszukiwania rozwiązania. Wyróżniają się na tle innych metod poruszanych na przestrzeni projektu tym, że rozwiązanie tworzone jest na podstawie współpracy osobników, gdzie każdy posiada część wiedzy na temat przestrzeni rozwiązań. Osobniki te przekazują swoją wiedzę następnym pokoleniom za pomocą mechanizmów ewolucyjnych. Same algorytmy genetyczne mogą być realizowane sposobem ewolucji darwinowskiej, gdzie środowisko jest ograniczone wielkością populacji. Populacja zwiększa się z czasem, a jej wielkość regulowana jest warunkiem przystosowania. Najlepiej przystosowane osobniki z każdej populacji mają największą szansę na rozmnożenie. Oprócz zmian wymienionych wyżej mogą także wystąpić mutacje, które zmieniają przystosowanie osobników. W algorytmach genetycznych występują trzy podstawowe operatory genetyczne:

- Selekcja – wybór z populacji osobników, którzy zostaną poddani operacji krzyżowania. Wybór ten następuje na podstawie określonej metody selekcji, która wykorzystuje funkcje przystosowania. Dzięki temu wybierane są osobniki o największej wartości przystosowania
- Krzyżowanie – wymiana materiału genetycznego pomiędzy wybranymi w procesie selekcji osobnikami, co następuje według określonej metody krzyżowania. Potomstwo powstałe w wyniku krzyżowania powinno być lepiej przystosowane od swoich rodziców.
- Mutacja – zmiana wartości cechy osobnika, co zachodzi z prawdopodobieństwem  $p_m$ . Służy to na przykład do wyjścia z minimum lokalnego. Sama mutacja przebiega według określonej metody mutacji.

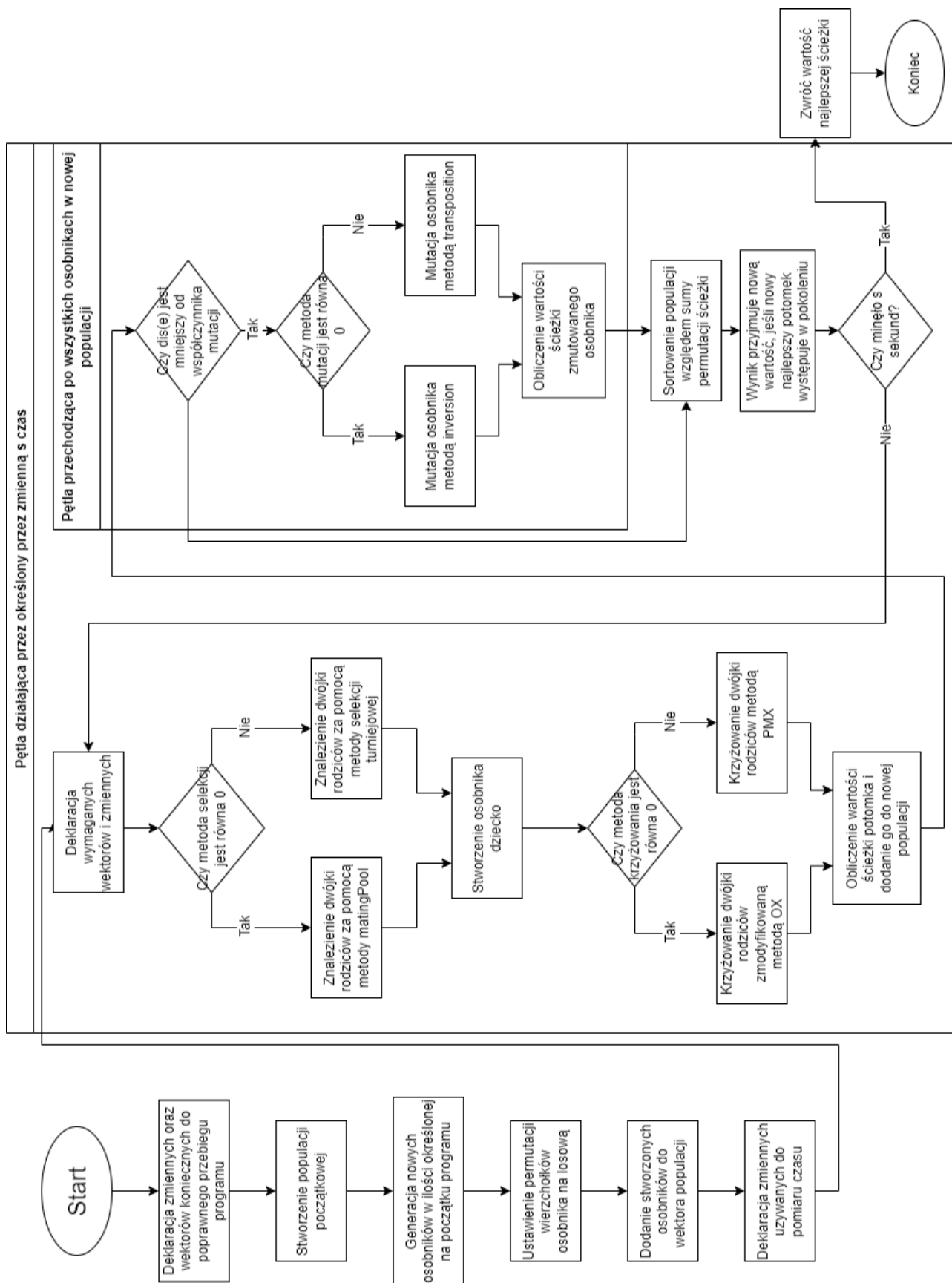
W założeniach im dłużej działa algorytm tym osobniki są coraz lepiej przystosowane, a co za tym idzie są coraz bliżej rozwiązania optymalnego.

### 3. Algorytm

W implementacji algorytmu użyto metod selekcji: turniejowej oraz ruletki. Metody krzyżowania to Partially Mapped Crossover (PMX) oraz zmodyfikowana Ordered Crossover (OX). Użyte metody mutacji to inversion oraz transposition. Algorytm rozpoczyna pracę od stworzenia populacji początkowej poprzez dobór osobników z losową permutacją wierzchołków. Następnie w zależności od metody następuje selekcja osobników do krzyżowania. Metoda selekcji turniejowej losuje zadaną ilość osobników, a następnie wybiera z nich najlepiej przystosowanego. Metoda ruletki dodaje do puli godowej osobników w ilości zależnej od ich funkcji przystosowania. Im większa jej wartość tym więcej takich osobników zostaje dodanych, gdzie najmniejsza ilość osobników dodanych to 1. Z puli są losowane pary osobników, które zostaną poddane krzyżowaniu. Następnie ma miejsce krzyżowanie w zależności od metody. Metoda PMX wybiera losowo początek i koniec podciągu, a następnie umieszcza podciąg w permutacjach potomków. Ostatnim krokiem jest uzupełnienie permutacji za pomocą tablicy mapowania. Metoda OX kopiuje do potomka pierwszą połowę jednego rodzica. Następnie permutacja potomka zostaje uzupełniona pasującymi wierzchołkami drugiego rodzica, oraz takimi, które jeszcze nie wystąpiły w permutacji potomka. Ostatnim krokiem jest mutacja która w zależności od metody, zamienia kolejność wierzchołków w wylosowanym podciągu (inversion), lub zamienia miejscami dwa wierzchołki w permutacji (transposition).



Rys. 1. – Schemat blokowy programu



Rys. 2. – Schemat blokowy algorytmu

## 4. Dane testowe

Do sprawdzenia poprawności działania algorytmu wybrano następujący zestaw instancji:

- 1.fri26.txt, war. optymalna: 937; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 2.ftv33.txt, war. optymalna: 1286; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 3.ftv35.txt, war. optymalna: 1473; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 4.ftv38.txt, war. optymalna: 1530; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 5.dantzig42, war. optymalna: 699; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 6.ftv44.txt, war. optymalna: 1520; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 7.burma14.txt, war. optymalna: 3323; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 8.gr17.txt, war. optymalna: 2085; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 9.gr21.txt, war. optymalna: 2707; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 10.gr24.txt, war. optymalna: 1272; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 11.bays29.txt, war. optymalna: 2020; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 12.ft53.txt, war. optymalna: 6905; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 13.ftv70.txt, war. optymalna: 1950; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 14.ch150.txt, war. optymalna: 6528; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 15.ftv170.txt, war. optymalna: 2755; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 16.gr202.txt, war. optymalna: 40160; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 17.rbg323.txt, war. optymalna: 1326; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 18.pcb442.txt, war. optymalna: 50778; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>
- 19.rbg443.txt, war. optymalna: 2720; <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>

## 5. Procedura badawcza

W przypadku rozwiązywania problemu komiwojażera za pomocą genetycznego pojawiają się parametry, których zmiany wpływają na efektywność działania programu. Są nimi:

- Wielkość populacji
- Prawdopodobieństwo mutacji
- Prawdopodobieństwo selekcji
- Prawdopodobieństwo krzyżowania
- Metoda selekcji
- Metoda krzyżowania
- Metoda mutacja
- Czas działania

W przypadku implementacji opisanej wcześniej zastosowano dwie metody selekcji: Turniejowa i Ruletki, krzyżowania: OX i PMX, oraz mutacji: inversion i trasposition. Zależność wielkości populacji była badana dla wartości  $n$ ,  $n \cdot 2$ ,  $n : 2$ . Prawdopodobieństwo selekcji jest zależne od wybranej metody. Prawdopodobieństwo mutacji przyjmowało wartości 0.1, 0.15 i 0.2. Czas działania w testach przyjmował wartości odpowiednio 5, 10 i 20 sekund.

Pomiary zostały wykonane na komputerze ze specyfikacją: Procesor AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz. Zainstalowana pamięć RAM 8,00 GB (dostępne: 7,37 GB). Typ systemu 64-bitowy system operacyjny, procesor x64. Do pomiaru czasu została użyta biblioteka <chrono>, wraz z narzędziem high\_resolution\_clock. Fragment kodu w języku C++ wykonujący pomiar czasu:

```
auto start = high_resolution_clock::now();  
  
auto stop = high_resolution_clock::now();  
  
auto duration = duration_cast<nanoseconds>(stop - start);  
  
float time = (duration.count())/(1e+9);  
  
csvFile << "Time [ms] = ;" << std::fixed << std::setprecision(3) << time*1000 << ";" << blad;
```

## 6. Wyniki

Wyniki zgromadzone zostały w pliku: Wykres.xlsx. Wyniki pomiarów dla różnych rozmiarów populacji zostały wykonane za pomocą metody selekcji Ruletki, krzyżowania PMX oraz mutacji transposition. Czas wykonywania algorytmu to 5 sekund. Prawdopodobieństwo mutacji to 1.5.

- $n$

Nazwa pliku	Błąd [%]
fri26.txt	9.534
ftv33.txt	11.612
ftv35.txt	11.767
ftv38.txt	7.821
dantzig42.txt	26.323
ftv44.txt	23.662

- $n \cdot 2$

Nazwa pliku	Błąd [%]
fri26.txt	2.028
ftv33.txt	12.390
ftv35.txt	10.093
ftv38.txt	10.959
dantzig42.txt	14.974
ftv44.txt	23.223

- $n : 2$

Nazwa pliku	Błąd [%]
fri26.txt	16.827
ftv33.txt	28.849
ftv35.txt	25.572
ftv38.txt	28.867
dantzig42.txt	39.056
ftv44.txt	47.960

Choć zaprezentowane wyżej wyniki jednoznacznie pokazują, że im większy rozmiar populacji, tym wyniki są lepsze, jednak wyjątkiem od tej reguły są instancje większe od 170, ponieważ w takich przypadkach lepiej sprawdzają się rozmiary populacji mniejsze niż  $n$ . Wynika to z faktu, że dzięki takiemu doborowi możliwe jest przejście przez algorytm po większej ilości generacji.



Wyniki pomiarów dla różnych prawdopodobieństw mutacji zostały wykonane za pomocą metody selekcji ruletki, krzyżowania PMX oraz mutacji transposition. Czas wykonywania algorytmu to 5 sekund. Wielkość populacji to  $n \cdot 3$ .

- 0.1

Nazwa pliku	Błąd [%]
fri26.txt	4.198
ftv33.txt	13.427
ftv35.txt	13.487
ftv38.txt	13.421
dantzig42.txt	16.509
ftv44.txt	33.114

- 0.15

Nazwa pliku	Błąd [%]
fri26.txt	3.380
ftv33.txt	15.474
ftv35.txt	13.668
ftv38.txt	19.150
dantzig42.txt	14.211
ftv44.txt	24.562

- 0.2

Nazwa pliku	Błąd [%]
fri26.txt	11.088
ftv33.txt	13.842
ftv35.txt	18.760
ftv38.txt	11.983
dantzig42.txt	13.257
ftv44.txt	30.592

Wyniki pomiarów dla różnych czasów zostały wykonane za pomocą metody selekcji ruletki, krzyżowania PMX oraz mutacji transposition. Wielkość populacji to  $n \cdot 3$ . Współczynnik prawdopodobieństwa mutacji to 0.15.

- 5 sekund

Nazwa pliku	Błąd [%]
fri26.txt	3.380
ftv33.txt	15.474
ftv35.txt	13.668
ftv38.txt	19.150
dantzig42.txt	14.211
ftv44.txt	24.562

- 10 sekund

Nazwa pliku	Błąd [%]
fri26.txt	3.202
ftv33.txt	12.074
ftv35.txt	12.288
ftv38.txt	10.806
dantzig42.txt	5.007
ftv44.txt	27.237

- 20 sekund

Nazwa pliku	Błąd [%]
fri26.txt	0.640
ftv33.txt	7.138
ftv35.txt	9.331
ftv38.txt	9.316
dantzig42.txt	3.639
ftv44.txt	24.913

Porównanie wyników pomiarów dla różnych metod selekcji zostały wykonane za pomocą metody krzyżowania PMX oraz mutacji transposition. Czas wykonywania algorytmu to 10 sekund. Wielkość populacji dobierana była różnie w zależności od wielkości instancji. Współczynnik prawdopodobieństwa mutacji to 0.15.

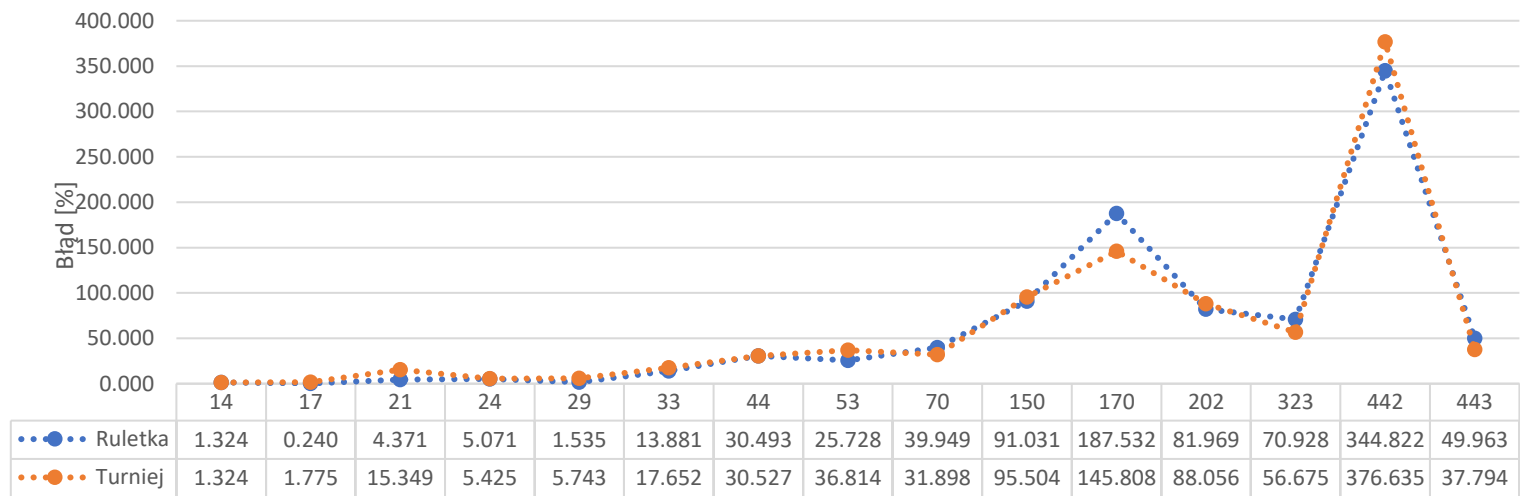
- Metoda selekcji ruletki

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	0.24
gr21.txt	4.371
gr24.txt	5.071
bays29.txt	1.534
ftv33.txt	13.880
ftv44.txt	30.493
ftv53.txt	25.727
ftv70.txt	39.949
ch150.txt	91.031
ftv170.txt	187.532
gr202.txt	81.968
rbg323.txt	70.928
pcb442.txt	344.821
rbg443.txt	49.963

- Metoda selekcji Turniejowa

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	1.774
gr21.txt	15.349
gr24.txt	5.424
bays29.txt	5.743
ftv33.txt	17.652
ftv44.txt	30.526
ftv53.txt	36.814
ftv70.txt	31.897
ch150.txt	95.504
ftv170.txt	145.808
gr202.txt	88.055
rbg323.txt	56.674
pcb442.txt	376.635
rbg443.txt	37.794

Błąd przy różnych metodach selekcji



Rys 3. – Wykres porównujący procent błędu dla 2 sposobów selekcji

Po analizie wyników można zauważyć, że różnice w zależności od metody selekcji nie są znaczące. Najłatwiej można dostrzec różnicę dla instancji 170. Wielkość turnieju to  $n : 2$ . Przy poprzednich badaniach jednak można było dostrzec drobną przewagę na korzyść metody ruletki, dlatego zostanie ona wykorzystana w poprzednich testach.

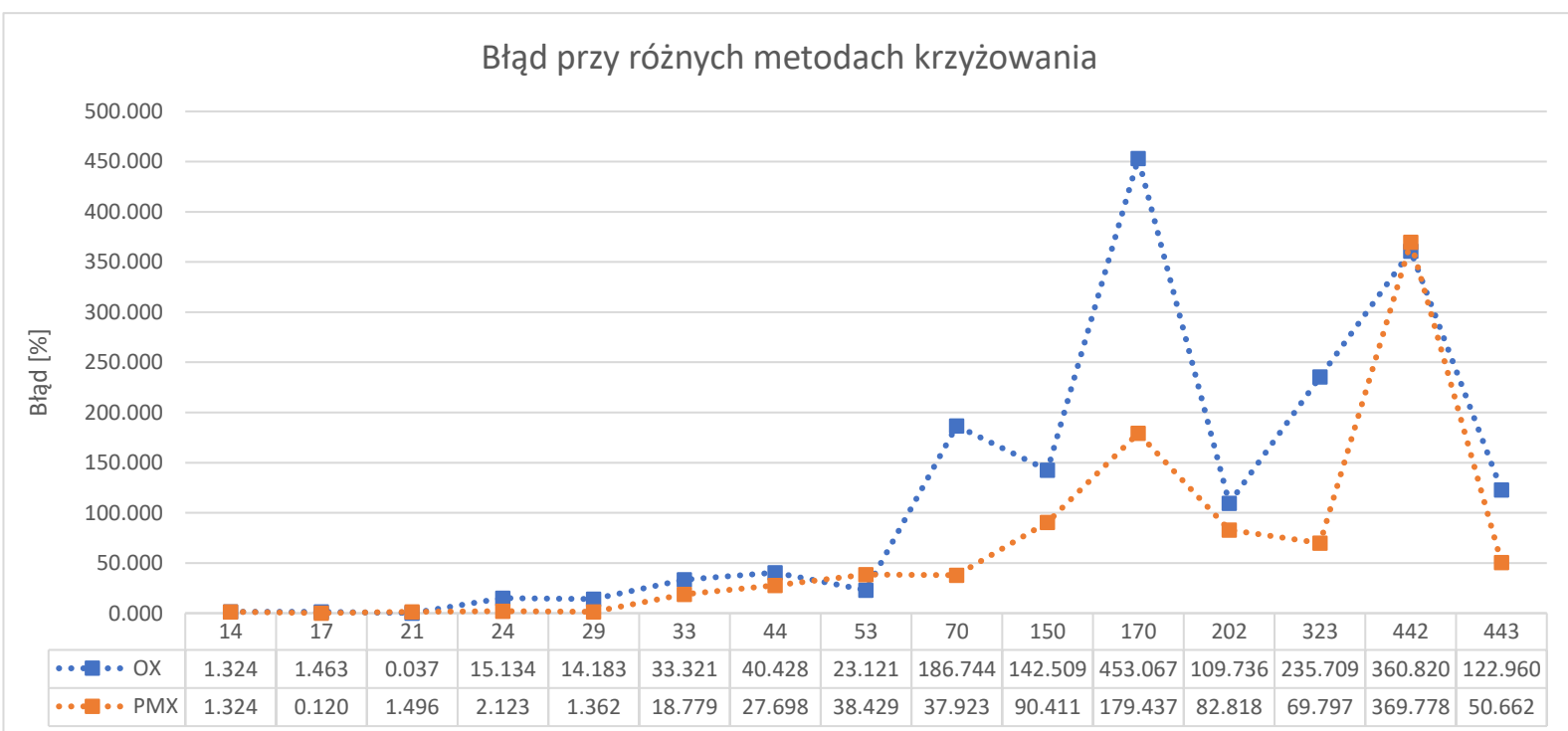
Porównanie wyników pomiarów dla różnych metod krzyżowania zostały wykonane za pomocą metody selekcji ruletki oraz mutacji transposition. Czas wykonywania algorytmu to 10 sekund. Wielkość populacji dobierana była różnie w zależności od wielkości instancji. Współczynnik prawdopodobieństwa mutacji to 0.15.

- Metoda krzyżowania OX

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	1.463
gr21.txt	0.037
gr24.txt	15.134
bays29.txt	14.183
ftv33.txt	33.321
ftv44.txt	40.428
ft53.txt	23.121
ftv70.txt	186.744
ch150.txt	142.509
ftv170.txt	453.067
gr202.txt	109.736
rbg323.txt	235.709
pcb442.txt	360.820
rbg443.txt	122.960

- Metoda krzyżowania PMX

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	0.120
gr21.txt	1.496
gr24.txt	2.123
bays29.txt	1.362
ftv33.txt	18.779
ftv44.txt	27.698
ft53.txt	38.429
ftv70.txt	37.923
ch150.txt	90.411
ftv170.txt	179.437
gr202.txt	82.818
rbg323.txt	69.797
pcb442.txt	369.778
rbg443.txt	50.662



Rys 4. – Wykres porównujący procent błędu dla 2 sposobów krzyżowania

Na wykresie można dostrzec, że metoda krzyżowania OX pomimo tego, że dla mniejszych instancji pozwala osiągnąć podobną dokładność co PMX, to w przypadku większych niż 53 wierzchołki różnica jest diametralna z korzyścią dla PMX, dlatego dla pozostałych pomiarów zostanie użyta ta metoda.

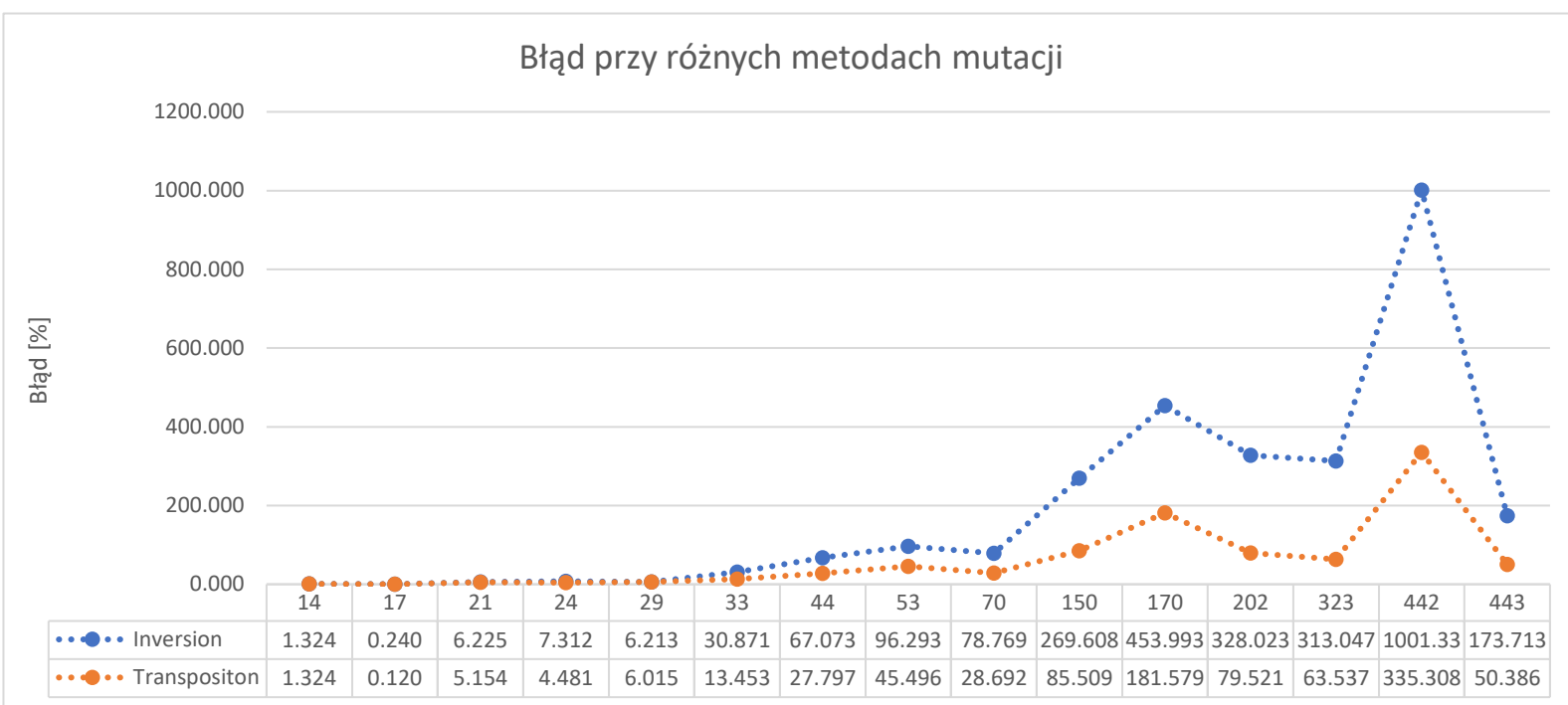
Porównanie wyników pomiarów dla różnych metod mutacji zostały wykonane za pomocą metody selekcji ruletki oraz krzyżowania PMX. Czas wykonywania algorytmu to 10 sekund. Wielkość populacji dobierana była różnie w zależności od wielkości instancji. Współczynnik prawdopodobieństwa mutacji to 0.15.

- Metoda mutacji inversion

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	0.240
gr21.txt	6.225
gr24.txt	7.312
bays29.txt	6.213
ftv33.txt	30.871
ftv44.txt	67.073
ft53.txt	96.293
ftv70.txt	78.769
ch150.txt	269.608
ftv170.txt	453.993
gr202.txt	328.023
rbg323.txt	313.047
pcb442.txt	1001.330
rbg443.txt	173.713

- Metoda mutacji transposition

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	0.120
gr21.txt	5.154
gr24.txt	4.481
bays29.txt	6.015
ftv33.txt	13.453
ftv44.txt	27.797
ft53.txt	45.496
ftv70.txt	28.692
ch150.txt	85.509
ftv170.txt	181.579
gr202.txt	79.521
rbg323.txt	63.537
pcb442.txt	335.308
rbg443.txt	50.386



*Rys 5. – Wykres porównujący procent błędu dla 2 sposobów mutacji*

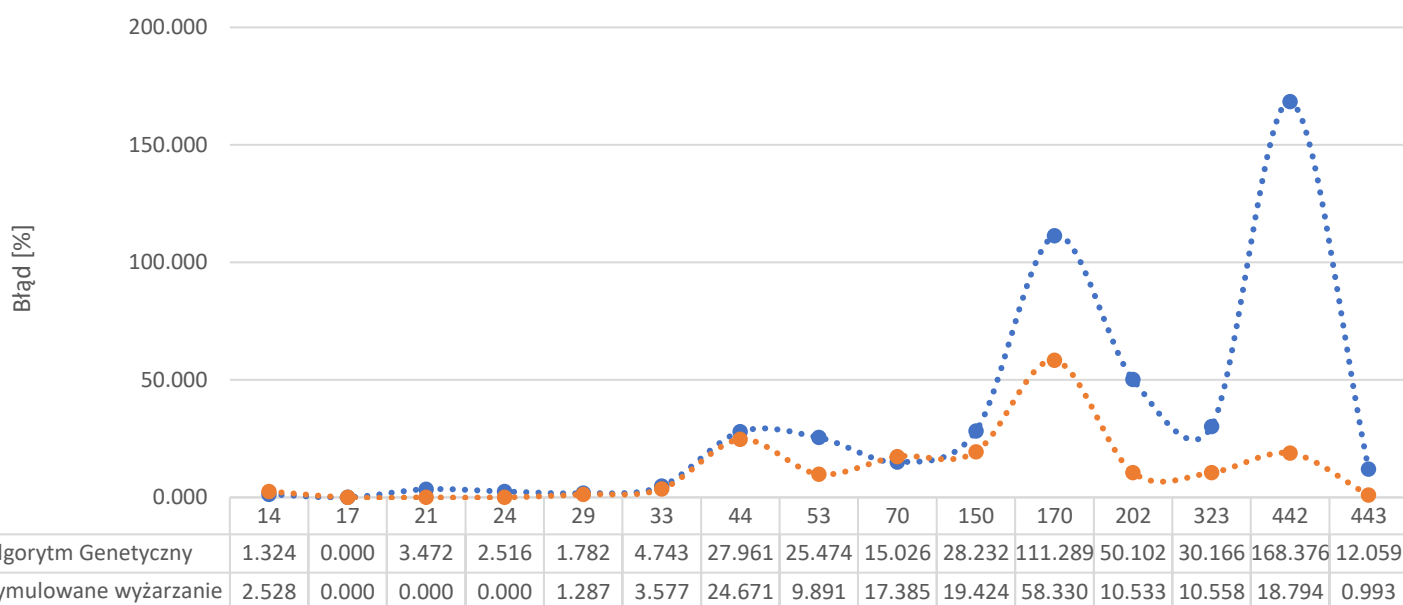
Na wykresie można zauważyć różnicę w procencie błędu dla metody inversion i transposition. W dokładności druga metoda deklasuje pierwszą, zwłaszcza w przypadku instancji większych niż 33. Dlatego w pozostałych pomiarach zostanie użyta metoda transposition.

## 7. Analiza wyników i wnioski

W ostatnim teście użyto parametrów oraz metod, które najlepiej sprawdzały się w poprzednich podpunktach z czasem trwania algorytmu ustawionym na 10 minut.

Nazwa pliku	Błąd [%]
burma14.txt	1.324
gr17.txt	0.000
gr21.txt	3.472
gr24.txt	2.516
bays29.txt	1.782
ftv33.txt	4.743
ftv44.txt	27.961
ft53.txt	25.474
ftv70.txt	15.026
ch150.txt	28.232
ftv170.txt	111.289
gr202.txt	50.102
rbg323.txt	30.166
pcb442.txt	168.376
rbg443.txt	12.059

Procent błędu dla algorytmu symulowanego wyżarzania i algorytmu genetycznego



Rys 6. – Wykres porównujący czasy wykonywania algorytmów w zależności od wielkości instancji



Na wykresie dostrzec można znaczne różnice w dokładności z korzyścią dla algorytmu symulowanego wyżarzania, szczególnie w instancjach większych niż 44 i to w czasie lepszym niż algorytm genetyczny. Nie zmienia to faktu, że gdyby ustawić czas działania algorytmu genetycznego mógłby on po czasie dojść do optimum. W zastosowanej implementacji algorytmu genetycznego najlepiej sprawdzały się metoda selekcji ruletki, metoda krzyżowania PMX i metoda mutacji transposition. Wartości parametrów były dobierane w zależności od instancji. W miejscu doboru parametrów mogłyby mieć miejsce największe usprawnienia i poprawa błędu. Współczynnik prawdopodobieństwa oraz rozmiar populacji powinny być dobierane za pomocą funkcji, która dopasuje takie wartości tych zmiennych, aby jak najlepiej zbliżyć się do optimum. Złożoność przejścia jednej generacji to  $wP \cdot n$ , gdzie  $wP$  to wielkość populacji, a  $n$  to rozmiar instancji. Aby uzyskać złożoność całego algorytmu należy powyższy wynik pomnożyć przez ilość generacji, które zdołają się wykonać w zadany czas. Same metody ewolucyjne są interesujące, a ich zastosowanie w uczeniu maszynowym może być przyszłością w nauce.