

Politechnika
Wrocławska,
Wydział Informatyki
i Telekomunikacji
Semestr V Rok III

Zadanie projektowe nr 3

Implementacja i analiza efektywności algorytmu
genetycznego (ewolucyjnego) dla problemu komiwojażera

Wykonał:

Patryk Ignasiak 263889

Gr. Piątek 11:15

Prowadzący:

dr inż. Jarosław Mierzwa

Przedmiot:

Projektowanie efektywnych
algorytmów – Projekt

Wrocław, 19 Styczeń 2024

Spis treści

1	Wstęp.....	4
1.1	Metody krzyżowania:.....	4
1.1.1	Order Crossover (OX).....	4
1.1.2	Partially Mapped Crossover (PMX).....	5
1.2	Metody mutacji	5
1.2.1	Swap mutation.....	5
2	Omówienie najważniejszych fragmentów kodu.....	6
2.1	Główna funkcja algorytmu.....	6
2.2	Generowanie osobników populacji.....	7
2.3	Obliczanie wartości fitness	8
2.4	Metoda order crossover.....	8
2.5	Metoda partially mapped crossover	9
2.6	Metoda mutacji	10
2.7	Metoda selekcji turniejowej	10
2.8	Pobieranie danych z pliku	11
3	Omówienie przeprowadzonych testów	12
3.1	Wstęp – omówienie sposobu testowania	12
3.2	Badanie wpływu wielkości populacji	12
3.2.1	Wstęp.....	12
3.2.2	Tabele z wynikami testów	12
3.2.3	Wykresy porównawcze dla testowanych wielkości populacji.....	21
3.2.4	Porównanie Tabu Search z algorytmem Genetycznym o populacji 500	22
3.2.5	Omówienie wyników testów	23
3.3	Badanie wpływu wielkości populacji na drugą metodę krzyżowania.....	23
3.3.1	Wstęp.....	23
3.3.2	Tabele z wynikami testów	24
3.3.3	Wykresy porównawcze dla testowanych wielkości populacji.....	30
3.3.4	Porównanie Tabu Search z algorytmem Genetycznym o populacji 500	31
3.3.5	Omówienie wyników testów	32
3.4	Wpływ wielkości populacji i metody krzyżowania	32
3.4.1	Plik ftv47.xml.....	32
3.4.2	Plik ftv170.xml.....	33
3.4.3	Plik rbg403.xml	33
3.4.4	Omówienie	33
3.5	Wpływ współczynnika mutacji.....	34
3.5.1	Wstęp.....	34
3.5.2	Tabele z wynikami testów	34

3.5.3	Wykresy porównawcze dla testowanych współczynników mutacji.....	44
3.5.4	Omówienie wyników	45
4	Podsumowanie projektu	45
5	Bibliografia	46

1 Wstęp

Podczas realizacji trzeciego projektu miała miejsce implementacja oraz analiza skuteczności algorytmu genetycznego (ewolucyjnego).

Badania zostały przeprowadzone w kontekście asymetrycznego problemu komiwojażera (ATSP). Ten specyficzny problem polega na znalezieniu minimalnego cyklu Hamiltona w grafie pełnym, skierowanym, gdzie wagi krawędzi są nieujemne. Minimalny cykl Hamiltona osiągany jest, gdy suma wag krawędzi w cyklu jest możliwie najmniejsza. Rozwiązanie tego problemu jest wyjątkowo trudne, zwłaszcza ze względu na liczbę krawędzi w grafie pełnym, skierowanym, która wynosi $V(V-1)$. Dodatkowo, liczba możliwych kombinacji tworzących cykl Hamiltona wynosi $(V-1)!$. Analiza efektywności algorytmów metaheurystycznych w tym kontekście ma kluczowe znaczenie, ponieważ dostarcza alternatywne rozwiązania w sytuacjach, gdzie metody klasyczne mogą być niewystarczające.

Algorytmy metaheurystyczne są technikami optymalizacyjnymi stosowanymi do rozwiązywania trudnych problemów, które przekraczają możliwości klasycznych metod. Omawiany algorytm posiada zdolność radzenia sobie z dużymi problemami optymalizacyjnymi, choć jego działanie jest obciążone pewnym stopniem losowości.

Algorytmy genetyczne są inspirowane procesem ewolucji biologicznej i wykorzystują mechanizmy selekcji naturalnej, krzyżowania (crossover) i mutacji do znalezienia lepszych rozwiązań w przestrzeni rozwiązań. W przypadku problemu komiwojażera, algorytm genetyczny może być zastosowany do generowania i ewoluowania populacji tras w poszukiwaniu trasy optymalnej lub zbliżonej do optymalnej.

Algorytm:

1. Inicjalizacja populacji
2. Obliczenie wartości fitness dla każdego osobnika
3. Krzyżowanie osobników
4. Mutacja osobników
5. Obliczenie wartości fitness
6. Selekcja metodą turniejową

Punkty 3-6 wykonywane są iteracyjnie aż do uzyskania warunku stopu, czyli przekroczeniu zadanego czasu.

1.1 Metody krzyżowania:

1.1.1 Order Crossover (OX)

Metoda ta polega na kopiowaniu fragmentu chromosomu znajdującego się pomiędzy dwoma punktami, a następnie uzupełnienie pozostałej części od drugiego rodzica.

Algorytm:

1. Losowy wybór dwóch punktów, które określą fragment
2. Kopiowanie fragmentu od jednego rodzica
3. Uzupełnianie pozostałej części od drugiego rodzica, zapamiętując wykorzystane wartości w celu uniknięcia duplikatów.

Przykład:

- Rodzice:
 - Parent1: {1, 3, 4, 5, 2, 8, 7, 9, 6, 0}
 - Parent2: {3, 2, 7, 5, 0, 1, 6, 9, 8, 4}
- Punkty:

- Start = 2
- End = 6
- Inicjalizacja potomka
 - Child: {-1, -1, -1, -1, -, 1, -1, -1, -, 1, -1}
- Kopiowanie fragmentu od Parent1
 - Child: {-1, -1, 4, 5, 2, 8, 7, -1, -1, -1}
- Wypełnienie pozostałej części od Parent2
 - Child: {3, 0, 4, 5, 2, 8, 7, 1, 6, 9}

1.1.2 Partially Mapped Crossover (PMX)

Metoda ta polega na inicjalizacji potomka danymi od jednego rodzica, a następnie kopiowaniu fragmentu chromosomu znajdującego się pomiędzy dwoma punktami od drugiego rodzica. Na koniec należy pozbyć się duplikatów

Algorytm:

1. Losowy wybór dwóch punktów, które określą fragment
2. Inicjalizacja potomka danymi od rodzica
3. Kopiowanie fragmentu od drugiego rodzica
4. Usuwanie duplikatów, zamieniać możemy tylko elementy które nie znajdują się w zadanym fragmencie

Przykład:

- Rodzice:
 - Parent1: {1, 3, 4, 5, 2, 8, 7, 9, 6, 0}
 - Parent2: {3, 2, 7, 5, 0, 1, 6, 9, 8, 4}
- Punkty:
 - Start = 2
 - End = 6
- Inicjalizacja potomka danymi od Parent1
 - Child: {1, 3, 4, 5, 2, 8, 7, 9, 6, 0}
- Kopiowanie fragmentu od Parent2
 - Child: {1, 3, 7, 5, 0, 1, 6, 9, 6, 0}
- Usuwanie duplikatów
 - Child: {1, 3, 7, 5, 0, 1, 6, 9, 6, 0}, znajdujemy zmapowaną wartość dla wartości 1, jeśli nie występuje wstawiamy, jeśli tak szukamy dalej
 - Child: {8, 3, 7, 5, 0, 1, 6, 9, 6, 0}
 - Child: {8, 3, 7, 5, 0, 1, 6, 9, 6, 2}
 - Child: {8, 3, 7, 5, 0, 1, 6, 9, 1, 2}
 - Child: {8, 3, 7, 5, 0, 1, 6, 9, 4, 2}, tutaj mieliśmy przypadek gdzie musieliśmy szukać dalej ponieważ wartość 7 już jest w Child.

1.2 Metody mutacji

1.2.1 Swap mutation

Metoda ta polega na wylosowaniu dwóch indeksów z zakresu $[0, \text{Size})$ i zamienienie wartości znajdujących się pod tymi indeksami.

Przykład:

- Rodzic:
 - Parent: {1, 3, 4, 5, 2, 8, 7, 9, 6, 0}
- Losujemy wartości:
 - I = 3

- J = 7
- Zamieniamy wartości:
 - Child: {1, 3, 4, 9, 2, 8, 7, 5, 6, 0}

2 Omówienie najważniejszych fragmentów kodu

2.1 Główna funkcja algorytmu

Funkcja genetic() realizuje główną strukturę algorytmu, to właśnie z niej wywoływane są inne, pomocnicze funkcje takie jak krzyżowania czy mutacje. Funkcja ta przyjmuje następujące parametry:

- macierz sąsiedztwa
- rozmiar sąsiedztwa
- warunek stopu (liczba sekund przez jakie ma działać algorytm)
- rozmiar populacji
- współczynnik mutacji (liczba typu double z zakresu 0.0 – 1.0)
- współczynnik krzyżowania ((liczba typu double z zakresu 0.0 – 1.0)
- wartość true/false (true – gdy wybrano Order Crossover, false – gdy wybrano Partially Mapped Crossover)

Na początku jest generowana populacja przy użyciu funkcji generatePath() omówionej w punkcie 2.2. Następnie przy użyciu funkcji calculateFitness() (pkt.2.3) obliczamy długości cykli dla każdego osobnika w populacji. Odbywa się to na początku przed pętlą ponieważ dla uproszczenia kodu zastosowałem funkcję do...while(), oraz delikatnie zmieniłem kolejność operacji. Wewnątrz pętli najpierw odbywa się krzyżowanie dwóch losowo wybranych osobników przy użyciu wybranej metody OX (orderCrossover() pkt.2.4) lub PMX(partiallyMappedCrossover() pkt.2.5). Następnie wykonywana jest mutacja (swapMutation() pkt.2.6). W kolejnym kroku obliczane są wartości osobników w populacji (calculateFitness() pkt.2.3), a na koniec odbywa się selekcja metodą turniejową (tournament() pkt.2.7). Podczas obliczania wartości fitness nowe najlepsze rozwiązania zapisywane są do zmiennych globalnych.

```
void Algorithm::genetic(const vector<vector<int>> matrix, int size, int stop, int populationSize, double
mutationRate, double crossRate, bool oX) {
    // Inicjalizacja populacji
    std::vector<std::vector<int>> population;
    for (size_t i = 0; i < populationSize; i++) {
        population.push_back(generatePath(matrix, size));
    }
    // Oblicz wartości fitness dla każdego osobnika
    std::vector<int> fitnessValues;
    for (const auto& individual : population) {
        int value = calculateFitness(individual, matrix, size);
        fitnessValues.push_back(value);
        if (bestResult > value) {
            bestResult = value;
            bestPath = individual;
        }
    }
    costs[0] = bestResult;
    int z = 1;
    time_t startTime = time(NULL);
    auto start = chrono::steady_clock::now();
```

```

do {
    // Krzyżowanie osobników
    for (int j = 0; j < populationSize; j++) {
        if (rand() / static_cast<double>(RAND_MAX) < crossRate) {
            int choose1, choose2;
            do {
                choose1 = rand() % populationSize;
                choose2 = rand() % populationSize;
            } while (choose1 == choose2);
            if (oX) {
                population.push_back(orderCrossover(population[choose1], population[choose2], size));
            }
            else {
                population.push_back(partiallyMappedCrossover(population[choose1], population[choose2],
size));
            }
        }
    }
    // Mutacja osobników
    for (int j = 0; j < populationSize; j++) {
        if (rand() / static_cast<double>(RAND_MAX) < mutationRate) {
            int choose1;
            choose1 = rand() % populationSize;
            population.push_back(swapMutation(population[choose1], size));
        }
    }
    // Oblicz wartości fitness dla każdego osobnika
    fitnessValues.clear();
    for (const auto& individual : population) {
        int value = calculateFitness(individual, matrix, size);
        fitnessValues.push_back(value);
        if (bestResult > value) {
            bestResult = value;
            bestPath = individual;

            auto end = chrono::steady_clock::now();
            auto duration = end - start;
            times[z] = chrono::duration_cast<chrono::microseconds>(duration).count();
            costs[z] = bestResult;
            z++;
        }
    }
    // Selekcja metodą turniejową
    population = tournament(population, fitnessValues, populationSize);
} while (time(NULL) - startTime < stop);
}

```

2.2 Generowanie osobników populacji

Generowanie osobników odbywa się metodą zachłanną z losowym wierzchołkiem startowym. Funkcja jako parametry przyjmuje macierz i rozmiar sąsiedztwa oraz zwraca wygenerowaną ścieżkę.

```

std::vector<int> Algorithm::generatePath(std::vector<std::vector<int>> matrix, int size) {
    vector<int> path(size);
    vector<int> tmp(size);
    bool* visited = new bool[size] {false}; // Tablica do śledzenia, czy miasto zostało odwiedzone
    // Losowy wybór startowego miasta
    int firstCity = rand() % size;

```

```

path[0] = firstCity;
visited[firstCity] = true;
// Wybór kolejnych miast
for (int i = 1; i < size; ++i) {
    int lastCity = path[i - 1];
    int nextCity = -1;
    int bCost = INT32_MAX;
    // Wybór najbliższego miasta spośród nieodwiedzonych
    for (int j = 0; j < size; ++j) {
        if (!visited[j] && matrix[lastCity][j] < bCost) {
            bCost = matrix[lastCity][j];
            nextCity = j;
        }
    }
    // Zaznaczenie miasta jako odwiedzonego
    visited[nextCity] = true;
    path[i] = nextCity;
}
return path;
}

```

2.3 Obliczanie wartości fitness

Obliczanie wartości fitness polega przeiterowaniu drogi oraz zsumowaniu długości kolejnych krawędzi. Funkcja zwraca obliczony dystans, a jako parametry przyjmuje drogę, macierz i rozmiar sąsiedztwa.

```

int Algorithm::calculateFitness(const std::vector<int>& route, const std::vector<std::vector<int>>& matrix,
int size) {
    int totalDistance = 0;
    for (int i = 0; i < size - 1; ++i) {
        if (i < size - 1 && route[i] < size && route[i + 1] < size) {
            totalDistance += matrix[route[i]][route[i + 1]];
        }
        else {
            cout << i << " " << route[i] << " " << route[i + 1] << endl;
        }
    }
    totalDistance += matrix[route.back()][route.front()];
    return totalDistance;
}

```

2.4 Metoda order crossover

Funkcja implementuje metodę order crossover(pkt. 1.1.1), czyli przekopiuje określony fragment pierwszego rodzica, a pozostałą część uzupełnia danymi od drugiego. Funkcja zwraca nowo utworzoną ścieżkę, a jako parametry przyjmuje dwóch rodziców i rozmiar sąsiedztwa.

```

std::vector<int> Algorithm::orderCrossover(const std::vector<int>& parent1, const std::vector<int>& parent2,
int size) {
    size_t start = rand() % size;
    size_t end = rand() % size;
    if (start > end) {
        std::swap(start, end);
    }
    std::vector<int> child(size, -1);
    // Przekopiuj fragment trasy od rodzica1 do potomka
    for (size_t i = start; i <= end; i++) {
        child[i] = parent1[i];
    }
}

```



```

    }
    // Uzupełnij resztę trasy potomka używając miast z rodzica2
    size_t index = 0;
    for (size_t i = 0; i < size; i++) {
        if (child[i] == -1) {
            while (std::find(child.begin(), child.end(), parent2[index]) != child.end()) {
                index++;
            }
            child[i] = parent2[index];
        }
    }
    return child;
}

```

2.5 Metoda partially mapped crossover

Funkcja implementuje mutację metodą swap (pkt. 1.2.1), czyli przekopiuje dane pierwszego rodzica, następnie kopiuje określony fragment od drugiego rodzica i na koniec pozbywa się duplikatów. Funkcja zwraca nowo utworzoną ścieżkę, a jako parametry przyjmuje dwóch rodziców i rozmiar sąsiedztwa.

```

std::vector<int> Algorithm::partiallyMappedCrossover(const std::vector<int>& parent1, const
std::vector<int>& parent2, int size) {
    size_t start = rand() % size;
    size_t end = rand() % size;
    while (start == end) {
        end = rand() % size;
    }
    if (start > end) {
        std::swap(start, end);
    }
    // Zainicjuj potomka jako kopię jednego z rodziców
    std::vector<int> child(parent1.begin(), parent1.end());
    // Mapa do śledzenia przyporządkowań
    std::unordered_map<int, int> assignmentMap;
    // Skopiuj fragment między punktami krzyżowania z drugiego rodzica do potomka
    for (int i = start; i <= end; ++i) {
        assignmentMap[parent2[i]] = parent1[i];
        child[i] = parent2[i];
    }
    // Zastąp duplikaty w potomku przy użyciu przyporządkowań
    for (int i = 0; i < size; ++i) {
        if (i < start || i > end) {
            int currentValue = child[i];
            while (assignmentMap.find(currentValue) != assignmentMap.end()) {
                currentValue = assignmentMap[currentValue];
            }
            child[i] = currentValue;
        }
    }
    return child;
}

```

2.6 Metoda mutacji

Funkcja implementuje metodę partially mapped crossover (pkt. 1.1.2), czyli losuje dwa indeksy z zakresu $[0, \text{size})$, a następnie zamienia wartości zawarte pod nimi. Funkcja zwraca nowo utworzoną ścieżkę, a jako parametry przyjmuje rodzica i rozmiar sąsiedztwa.

```
std::vector<int> Algorithm::swapMutation(std::vector<int> route, int size) {
    size_t i, j;
    do {
        i = rand() % size;
        j = rand() % size;
    } while (i == j);
    std::swap(route[i], route[j]);
    return route;
}
```

2.7 Metoda selekcji turniejowej

Funkcja implementuje turniejową metodę selekcji, która polega na wylosowaniu określonej ilości uczestników turnieju, ja wybrałem rywalizację w parach, ponieważ generuje zadawalające wyniki jednocześnie nie spowalniając bardzo algorytmu. Następnie odbywa się turniej, czyli wybór najlepszego osobnika w moim wypadku wybór osobnika o najmniejszej wartości fitness. Funkcja zwraca nowo utworzoną populację, a jako parametry przyjmuje populację, wartości fitness oraz rozmiar populacji.

```
std::vector<std::vector<int>> Algorithm::tournament( std::vector<std::vector<int>>& population, const
std::vector<int>& fitnessValues, int populationSize) {
    std::vector<vector<int>> parents;
    int tournamentSize = 2;
    for (int i = 0; i < populationSize; i++) {
        // Wybór grupy turniejowej
        std::vector<int> tournamentGroup, indexes;
        for (int j = 0; j < tournamentSize; ++j) {
            int randomIndex = rand() % population.size();
            tournamentGroup.push_back(fitnessValues[randomIndex]);
            indexes.push_back(randomIndex);
        }
        // Wybór najlepszego kandydata
        auto bestIndex = std::min_element(tournamentGroup.begin(), tournamentGroup.end()) -
tournamentGroup.begin();
        parents.push_back(population[indexes[bestIndex]]);
    }
    population.clear();
    return parents;
}
```

2.8 Pobieranie danych z pliku

Aplikacja posiada funkcję, która umożliwia wczytywanie plików .xml z danymi.

```
void Data::readFile(string filename) {
    // Otwórz plik XML
    std::ifstream file(filename);
    if (!file.is_open()) {
        std::cerr << "Nie można otworzyć pliku XML." << std::endl;
        return;
    }
    // policz wierzchołki
    std::string line;
    size = 0;
    while (std::getline(file, line)) {
        if (line.find("<vertex>") != std::string::npos) {
            size++;
        }
    }
    file.close();
    // utwórz macierz
    std::vector<std::vector<int>> tmpMatrix(size, std::vector<int>(size, INT32_MAX));
    this->matrix = tmpMatrix;
    // załaduj dane
    std::ifstream f(filename);
    int numOfRow = -1;
    while (std::getline(f, line)) {
        if (line.find("<vertex>") != std::string::npos) {
            numOfRow++;
        }
        if (line.find("<edge>") != std::string::npos) {
            double edge;
            double power;
            int index;
            size_t startPos = line.find("\"") + 1;
            size_t endPos = line.find("\"", startPos);
            std::string numberStr = line.substr(startPos, endPos - startPos);
            std::istringstream iss(numberStr);
            iss >> edge;
            startPos = line.find("+") + 1;
            endPos = line.find("\"", startPos);
            numberStr = line.substr(startPos, endPos - startPos);
            numberStr = line.substr(startPos, endPos - startPos);
            std::istringstream iss2(numberStr);
            iss2 >> power;
            edge = edge * pow(10, power);
            startPos = line.find(">") + 1;
            endPos = line.find("<", startPos);
            numberStr = line.substr(startPos, endPos - startPos);
            numberStr = line.substr(startPos, endPos - startPos);
            std::istringstream iss3(numberStr);
            iss3 >> index;
            matrix[numOfRow][index] = edge;
        }
    }
    f.close();
    cout << "Dane zostały wczytane!!!" << endl;
}
```

3 Omówienie przeprowadzonych testów

3.1 Wstęp – omówienie sposobu testowania

Dla każdego z plików było wykonywane 10 powtórzeń dla każdego przypadku. Jako przypadek można rozumieć:

- Pkt 3.2, 3.3 – wielkość populacji
- Metoda krzyżowania

Warunek stopu w zależności od pliku:

- ftv47.xml – 30 sekund
- ftv170.xml – 60 sekund
- rbg403.xml – 90 sekund

3.2 Badanie wpływu wielkości populacji

3.2.1 Wstęp

W tym punkcie testowanym parametrem była wielkość populacji. Testy zostały przeprowadzone zgodnie z opisem w pkt. 3.1. Parametry testu:

- Metoda krzyżowania – Order Crossover
- Metoda mutacji – Swap Mutation
- Współczynnik krzyżowania – 0.8
- Współczynnik mutacji – 0.01

3.2.2 Tabele z wynikami testów

3.2.2.1 Plik ftv47.xml

Populacja 50		
Czas[ms]	Droga	Błąd względny [%]
0,00	2172	22,2973
0,40	2151	21,11486
1,25	2145	20,77703
1,39	2134	20,15766
69,61	2085	17,39865
109,771	2084	17,34234
110,724	2081	17,17342
224,216	2050	15,42793

Populacja 500		
Czas[ms]	Droga	Błąd względny [%]
0	2171	22,24099
4,413	2150	21,05856
4,419	2134	20,15766
4,786	2133	20,10135
35,16	2070	16,55405
42,183	2068	16,44144
104,141	2051	15,48423
132,347	2050	15,42793
237,716	2026	14,07658

Populacja 5000		
Czas[ms]	Droga	Błąd względny [%]
0	2171	22,24099
7,968	2149	21,00225
25,085	2122	19,48198
40,681	2110	18,80631
59,191	2108	18,69369
62,745	2089	17,62387
66,737	2072	16,66667
89,708	2070	16,55405
93,256	2051	15,48423
122,117	2048	15,31532
141,58	2046	15,2027
145,127	2027	14,13288
195,926	2026	14,07658
280,04	2006	12,95045

3.2.2.2 Plik ftv170.xml

Populacja 50		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,048799	3536	28,34846
0,078074	3515	27,58621
0,078967	3488	26,60617
2,983522	3469	25,91652
3,325455	3430	24,50091
4,134262	3415	23,95644
5,373224	3398	23,33938
6,349454	3384	22,83122

Populacja 5000		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,250881	3558	29,14701
0,279947	3532	28,20327
0,405844	3508	27,33212
0,419262	3486	26,53358
0,459613	3462	25,66243
0,499808	3454	25,37205
0,783944	3415	23,95644
2,979789	3379	22,64973
3,142912	3340	21,23412
4,964962	3326	20,72595

Populacja 500		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,051705	3544	28,63884
0,074535	3511	27,44102
0,190475	3481	26,35209
0,482607	3435	24,6824
0,660194	3434	24,6461
5,323038	3414	23,92015
5,830486	3404	23,55717
6,952452	3339	21,19782
10,27	3334	21,01633
11,79	3325	20,68966

3.2.2.3 Plik rbg403.xml

Populacja 50			Populacja 500			Populacja 5000		
Czas[s]	Droga	Błąd względny [%]	Czas[ms]	Droga	Błąd względny [%]	Czas[s]	Droga	Błąd względny [%]
0,00	3501	42,03	0,00	3497	41,87	0,00	3497	41,87
0,11	3499	41,95	0,20	3495	41,78	0,20	3495	41,78
0,13	3496	41,83	0,21	3488	41,50	0,59	3483	41,30
0,13	3495	41,78	0,22	3482	41,26	0,91	3481	41,22
0,14	3485	41,38	0,23	3479	41,14	1,10	3478	41,10
0,14	3470	40,77	0,25	3476	41,01	1,35	3474	40,93
0,15	3463	40,49	0,26	3466	40,61	1,81	3466	40,61
0,30	3453	40,08	0,31	3459	40,32	2,00	3465	40,57
0,31	3452	40,04	0,32	3458	40,28	2,06	3463	40,49
0,31	3449	39,92	0,33	3457	40,24	2,07	3462	40,45
0,34	3446	39,80	0,35	3456	40,20	2,07	3461	40,41
0,34	3445	39,76	0,44	3447	39,84	2,13	3454	40,12
0,47	3442	39,63	0,45	3445	39,76	2,25	3452	40,04
0,67	3438	39,47	0,50	3443	39,68	2,31	3446	39,80
0,67	3434	39,31	0,50	3440	39,55	2,56	3445	39,76

0,67	3433	39,27	0,50	3437	39,43	2,63	3443	39,68
0,68	3429	39,11	0,54	3436	39,39	2,63	3441	39,59
0,68	3423	38,86	0,54	3435	39,35	2,75	3428	39,07
0,69	3417	38,62	0,54	3434	39,31	3,07	3422	38,82
0,70	3409	38,30	0,55	3432	39,23	3,13	3421	38,78
0,71	3408	38,26	0,55	3431	39,19	3,19	3416	38,58
0,71	3407	38,22	0,56	3429	39,11	3,32	3415	38,54
0,78	3406	38,17	0,56	3428	39,07	3,39	3414	38,50
0,88	3401	37,97	0,61	3427	39,03	3,39	3412	38,42
0,95	3400	37,93	0,61	3423	38,86	3,46	3407	38,22
0,99	3397	37,81	0,62	3422	38,82	3,53	3403	38,05
0,99	3393	37,65	0,65	3420	38,74	3,60	3402	38,01
0,99	3386	37,36	0,65	3419	38,70	3,74	3398	37,85
1,06	3384	37,28	0,68	3418	38,66	3,81	3391	37,57
1,18	3379	37,08	0,68	3415	38,54	3,88	3390	37,53
1,24	3377	37,00	0,69	3413	38,46	3,95	3387	37,40
1,24	3376	36,96	0,70	3411	38,38	4,08	3385	37,32
1,26	3366	36,55	0,71	3410	38,34	4,09	3378	37,04
1,31	3364	36,47	0,76	3409	38,30	4,42	3374	36,88
1,31	3358	36,23	0,76	3407	38,22	4,54	3370	36,71
1,32	3357	36,19	0,77	3401	37,97	4,79	3368	36,63
1,32	3356	36,15	0,84	3394	37,69	4,79	3364	36,47
1,32	3352	35,98	0,89	3391	37,57	4,91	3363	36,43
1,40	3350	35,90	0,90	3383	37,24	4,91	3362	36,39
1,40	3345	35,70	0,93	3376	36,96	4,97	3357	36,19
1,48	3343	35,62	1,02	3374	36,88	4,97	3356	36,15
1,55	3342	35,58	1,21	3370	36,71	5,10	3352	35,98
1,58	3341	35,54	1,22	3367	36,59	5,16	3348	35,82
1,58	3339	35,46	1,24	3366	36,55	5,29	3341	35,54
1,94	3338	35,42	1,26	3365	36,51	5,41	3340	35,50
1,97	3332	35,17	1,36	3364	36,47	5,48	3335	35,29
2,15	3325	34,89	1,48	3363	36,43	5,60	3332	35,17
2,19	3322	34,77	1,49	3362	36,39	5,72	3328	35,01
2,24	3310	34,28	1,52	3360	36,31	5,72	3327	34,97
2,50	3309	34,24	1,54	3356	36,15	5,78	3324	34,85
2,59	3302	33,96	1,54	3355	36,11	5,85	3318	34,60
2,60	3301	33,91	1,55	3354	36,06	6,37	3316	34,52
2,60	3299	33,83	1,57	3353	36,02	6,43	3315	34,48
2,72	3295	33,67	1,58	3349	35,86	6,49	3314	34,44
2,73	3293	33,59	1,58	3343	35,62	6,55	3312	34,36
2,77	3291	33,51	1,67	3334	35,25	6,56	3311	34,32
2,80	3285	33,27	1,68	3333	35,21	6,62	3310	34,28
2,96	3283	33,18	1,72	3332	35,17	6,89	3308	34,20
3,01	3282	33,14	1,72	3331	35,13	7,03	3305	34,08
3,14	3276	32,90	1,72	3330	35,09	7,23	3304	34,04
3,20	3265	32,45	1,74	3329	35,05	7,23	3303	34,00

3,22	3263	32,37	1,75	3328	35,01	7,29	3301	33,91
3,88	3262	32,33	1,80	3327	34,97	7,54	3299	33,83
4,11	3261	32,29	1,95	3322	34,77	7,67	3298	33,79
4,11	3260	32,25	2,00	3311	34,32	7,73	3297	33,75
4,12	3259	32,21	2,08	3309	34,24	7,79	3295	33,67
4,21	3257	32,13	2,13	3308	34,20	7,79	3292	33,55
4,23	3255	32,05	2,21	3306	34,12	7,98	3291	33,51
4,30	3254	32,01	2,26	3304	34,04	7,98	3284	33,23
4,41	3251	31,89	2,29	3302	33,96	8,16	3283	33,18
4,46	3249	31,81	2,29	3300	33,87	8,29	3281	33,10
4,46	3246	31,68	2,29	3295	33,67	8,35	3280	33,06
4,57	3241	31,48	2,31	3292	33,55	8,48	3271	32,70
4,81	3234	31,20	2,36	3290	33,47	8,73	3266	32,49
4,82	3228	30,95	2,37	3288	33,39	8,80	3265	32,45
4,83	3226	30,87	2,43	3286	33,31	9,05	3263	32,37
5,05	3225	30,83	2,60	3276	32,90	9,24	3261	32,29
5,28	3223	30,75	2,66	3274	32,82	9,37	3256	32,09
5,30	3222	30,71	2,67	3271	32,70	9,80	3252	31,93
5,33	3220	30,63	2,69	3269	32,62	10,29	3250	31,85
5,34	3219	30,59	2,70	3267	32,54	10,29	3248	31,76
5,39	3217	30,51	2,70	3261	32,29	10,48	3246	31,68
5,41	3214	30,39	2,72	3256	32,09	10,67	3245	31,64
5,52	3210	30,22	2,78	3248	31,76	10,67	3242	31,52
5,65	3206	30,06	2,80	3245	31,64	10,98	3238	31,36
5,77	3205	30,02	2,80	3236	31,28	11,10	3235	31,24
5,84	3204	29,98	2,85	3233	31,16	11,23	3234	31,20
5,97	3201	29,86	2,86	3228	30,95	11,41	3232	31,12
6,63	3197	29,70	2,87	3223	30,75	11,60	3231	31,08
7,08	3193	29,53	2,90	3220	30,63	11,60	3230	31,03
7,08	3192	29,49	2,92	3219	30,59	11,72	3229	30,99
7,21	3190	29,41	2,96	3215	30,43	11,85	3227	30,91
7,33	3189	29,37	2,96	3214	30,39	11,91	3225	30,83
7,33	3185	29,21	2,96	3213	30,34	11,97	3223	30,75
7,45	3184	29,17	2,98	3212	30,30	12,16	3221	30,67
7,72	3182	29,09	3,01	3209	30,18	12,16	3219	30,59
7,97	3181	29,05	3,02	3207	30,10	12,16	3218	30,55
7,98	3178	28,92	3,02	3204	29,98	12,35	3214	30,39
8,98	3174	28,76	3,04	3202	29,90	12,48	3212	30,30
9,63	3167	28,48	3,17	3197	29,70	12,54	3211	30,26
9,88	3164	28,36	3,26	3193	29,53	12,60	3210	30,22
9,97	3161	28,24	3,27	3192	29,49	12,72	3209	30,18
10,05	3153	27,91	3,28	3190	29,41	12,72	3208	30,14
10,74	3148	27,71	3,29	3185	29,21	12,79	3207	30,10
10,88	3147	27,67	3,33	3182	29,09	12,85	3203	29,94
11,12	3146	27,63	3,37	3179	28,97	12,85	3202	29,90
11,55	3137	27,26	3,65	3177	28,88	12,91	3200	29,82

11,56	3136	27,22	3,84	3173	28,72	12,98	3196	29,66
11,65	3135	27,18	3,87	3165	28,40	13,30	3191	29,45
11,73	3125	26,77	3,91	3161	28,24	13,49	3186	29,25
12,37	3124	26,73	3,97	3160	28,19	13,69	3177	28,88
12,61	3115	26,37	4,03	3159	28,15	14,08	3176	28,84
12,62	3100	25,76	4,30	3158	28,11	14,08	3175	28,80
12,71	3098	25,68	4,34	3154	27,95	14,14	3171	28,64
12,72	3089	25,31	4,35	3149	27,75	14,27	3169	28,56
13,36	3085	25,15	4,41	3144	27,55	14,27	3167	28,48
13,37	3083	25,07	4,45	3143	27,51	14,45	3166	28,44
13,38	3073	24,67	4,46	3136	27,22	14,45	3164	28,36
13,54	3059	24,10	4,61	3134	27,14	14,58	3162	28,28
15,34	3054	23,89	4,82	3133	27,10	14,65	3159	28,15
15,35	3051	23,77	4,90	3123	26,69	14,71	3158	28,11
15,39	3050	23,73	5,00	3118	26,49	14,71	3156	28,03
15,39	3049	23,69	5,03	3112	26,25	14,84	3152	27,87
15,46	3044	23,49	5,32	3111	26,21	14,91	3147	27,67
16,19	3039	23,29	5,56	3107	26,04	15,22	3145	27,59
16,27	3037	23,20	5,59	3106	26,00	15,28	3144	27,55
16,34	3033	23,04	5,63	3097	25,64	15,28	3143	27,51
16,37	3025	22,72	5,65	3094	25,52	15,28	3141	27,42
16,68	3022	22,60	5,66	3091	25,40	15,60	3140	27,38
17,34	3021	22,56	5,68	3090	25,35	15,60	3139	27,34
17,63	3020	22,52	5,68	3089	25,31	15,66	3138	27,30
17,78	3015	22,31	5,69	3087	25,23	15,79	3137	27,26
18,47	3012	22,19	5,71	3085	25,15	15,79	3136	27,22
18,86	3011	22,15	5,73	3081	24,99	15,80	3134	27,14
19,30	3009	22,07	5,75	3075	24,75	15,99	3132	27,06
19,37	3003	21,83	5,81	3071	24,58	16,11	3131	27,02
20,06	3002	21,78	5,84	3070	24,54	16,18	3126	26,82
20,06	2997	21,58	5,85	3063	24,26	16,37	3125	26,77
20,48	2996	21,54	5,92	3062	24,22	16,50	3120	26,57
20,85	2990	21,30	5,93	3060	24,14	16,56	3117	26,45
21,08	2984	21,05	5,98	3058	24,06	16,62	3116	26,41
21,47	2983	21,01	6,36	3054	23,89	16,88	3114	26,33
21,59	2974	20,65	6,40	3050	23,73	17,01	3112	26,25
21,66	2973	20,61	6,42	3048	23,65	17,07	3108	26,09
22,64	2963	20,20	6,54	3044	23,49	17,39	3107	26,04
22,82	2957	19,96	6,70	3043	23,45	17,45	3106	26,00
22,90	2956	19,92	6,72	3035	23,12	17,52	3105	25,96
23,16	2954	19,84	6,76	3034	23,08	17,71	3103	25,88
23,61	2945	19,47	6,79	3032	23,00	17,77	3100	25,76
23,62	2941	19,31	6,81	3025	22,72	18,14	3095	25,56
23,70	2933	18,99	7,61	3024	22,68	18,33	3094	25,52
24,51	2932	18,95	7,76	3017	22,39	18,51	3093	25,48
24,65	2926	18,70	7,85	3015	22,31	18,51	3089	25,31

25,04	2922	18,54	7,86	3011	22,15	18,71	3088	25,27
25,05	2920	18,46	8,24	3007	21,99	18,83	3087	25,23
25,71	2915	18,26	8,31	3006	21,95	18,83	3085	25,15
26,84	2914	18,22	8,36	3005	21,91	18,89	3084	25,11
27,31	2911	18,09	8,42	3004	21,87	18,89	3080	24,95
29,19	2907	17,93	8,85	3001	21,74	19,01	3076	24,79
29,35	2906	17,89	9,31	2999	21,66	19,15	3072	24,62
29,97	2900	17,65	9,53	2998	21,62	19,29	3071	24,58
30,17	2893	17,36	9,58	2997	21,58	19,35	3068	24,46
30,57	2891	17,28	9,99	2996	21,54	19,55	3067	24,42
31,00	2889	17,20	10,17	2990	21,30	19,55	3066	24,38
31,96	2885	17,04	10,24	2989	21,26	19,55	3065	24,34
32,14	2875	16,63	10,24	2988	21,22	19,61	3062	24,22
32,39	2874	16,59	10,25	2983	21,01	19,73	3056	23,98
32,47	2872	16,51	10,25	2978	20,81	20,17	3049	23,69
32,48	2870	16,43	11,43	2972	20,57	20,49	3045	23,53
33,16	2860	16,02	11,67	2971	20,53	20,67	3041	23,37
33,91	2859	15,98	12,34	2967	20,37	20,74	3038	23,25
35,94	2856	15,86	12,97	2966	20,32	21,13	3037	23,20
36,46	2855	15,82	13,15	2964	20,24	21,38	3036	23,16
37,15	2851	15,66	13,46	2958	20,00	21,64	3022	22,60
37,15	2847	15,50	14,12	2955	19,88	22,76	3014	22,27
39,22	2845	15,42	14,20	2954	19,84	22,89	3013	22,23
39,22	2842	15,29	14,64	2952	19,76	23,14	3012	22,19
39,98	2837	15,09	15,01	2947	19,55	23,34	3011	22,15
40,16	2836	15,05	15,07	2941	19,31	23,34	3010	22,11
40,55	2835	15,01	17,09	2937	19,15	23,89	3008	22,03
41,28	2827	14,69	17,15	2936	19,11	24,07	3007	21,99
42,44	2826	14,65	17,33	2934	19,03	24,51	3005	21,91
42,79	2822	14,48	17,37	2929	18,82	24,51	3003	21,83
44,26	2821	14,44	18,20	2928	18,78	24,57	3001	21,74
44,40	2818	14,32	18,26	2926	18,70	24,63	2999	21,66
47,14	2814	14,16	20,08	2919	18,42	25,18	2997	21,58
47,15	2813	14,12	21,02	2917	18,34	25,67	2996	21,54
47,36	2806	13,83	21,16	2914	18,22	25,73	2994	21,46
50,89	2805	13,79	21,20	2912	18,13	25,91	2987	21,18
52,63	2799	13,55	21,20	2907	17,93	26,34	2986	21,14
52,70	2798	13,51	21,22	2903	17,77	26,59	2982	20,97
52,73	2796	13,43	21,36	2898	17,57	26,90	2978	20,81
54,88	2794	13,35	21,94	2894	17,40	26,96	2977	20,77
54,88	2793	13,31	21,96	2888	17,16	27,21	2976	20,73
55,79	2791	13,23	23,10	2885	17,04	27,27	2974	20,65
55,79	2789	13,14	23,17	2884	17,00	27,27	2972	20,57
58,12	2783	12,90	23,20	2883	16,96	27,33	2966	20,32
58,14	2777	12,66	24,74	2875	16,63	27,95	2964	20,24
64,67	2776	12,62	24,82	2874	16,59	28,13	2962	20,16

64,83	2768	12,29	24,84	2872	16,51	28,38	2959	20,04
72,49	2764	12,13	24,91	2871	16,47	28,38	2958	20,00
84,50	2763	12,09	24,96	2869	16,39	28,57	2954	19,84
84,80	2762	12,05	25,78	2858	15,94	28,76	2948	19,59
86,40	2759	11,93	25,90	2854	15,78	28,82	2946	19,51
86,66	2758	11,89	26,44	2845	15,42	29,06	2945	19,47
86,99	2755	11,76	27,08	2841	15,25	29,06	2942	19,35
			27,12	2840	15,21	29,12	2938	19,19
			27,12	2838	15,13	29,12	2937	19,15
			27,13	2833	14,93	29,25	2935	19,07
			27,23	2831	14,85	29,31	2934	19,03
			28,45	2827	14,69	29,31	2933	18,99
			28,96	2823	14,52	29,37	2932	18,95
			29,02	2819	14,36	29,43	2930	18,86
			30,21	2818	14,32	29,43	2929	18,82
			32,55	2810	14,00	29,62	2926	18,70
			34,37	2809	13,96	29,68	2925	18,66
			34,89	2807	13,87	29,92	2923	18,58
			34,97	2806	13,83	30,17	2918	18,38
			35,66	2804	13,75	30,42	2917	18,34
			35,94	2803	13,71	30,67	2916	18,30
			37,04	2802	13,67	30,73	2915	18,26
			37,12	2798	13,51	30,73	2914	18,22
			37,15	2796	13,43	30,91	2912	18,13
			37,22	2795	13,39	30,97	2906	17,89
			37,41	2791	13,23	31,04	2905	17,85
			37,46	2789	13,14	31,29	2904	17,81
			38,28	2788	13,10	31,35	2902	17,73
			40,40	2787	13,06	31,60	2900	17,65
			40,47	2786	13,02	31,60	2897	17,53
			40,49	2783	12,90	31,72	2896	17,48
			40,56	2778	12,70	32,10	2895	17,44
			45,00	2770	12,37	32,22	2894	17,40
			47,36	2769	12,33	32,29	2891	17,28
			47,42	2767	12,25	32,35	2890	17,24
			48,03	2765	12,17	32,53	2889	17,20
			49,21	2764	12,13	32,84	2887	17,12
			50,60	2763	12,09	32,91	2885	17,04
			50,61	2759	11,93	33,10	2884	17,00
			54,59	2756	11,81	33,16	2882	16,92
			54,66	2753	11,68	33,50	2881	16,88
			54,92	2751	11,60	33,57	2880	16,84
			56,73	2749	11,52	33,72	2877	16,71
			56,82	2748	11,48	34,03	2876	16,67
			56,93	2747	11,44	34,03	2873	16,55
			56,99	2738	11,08	34,45	2872	16,51

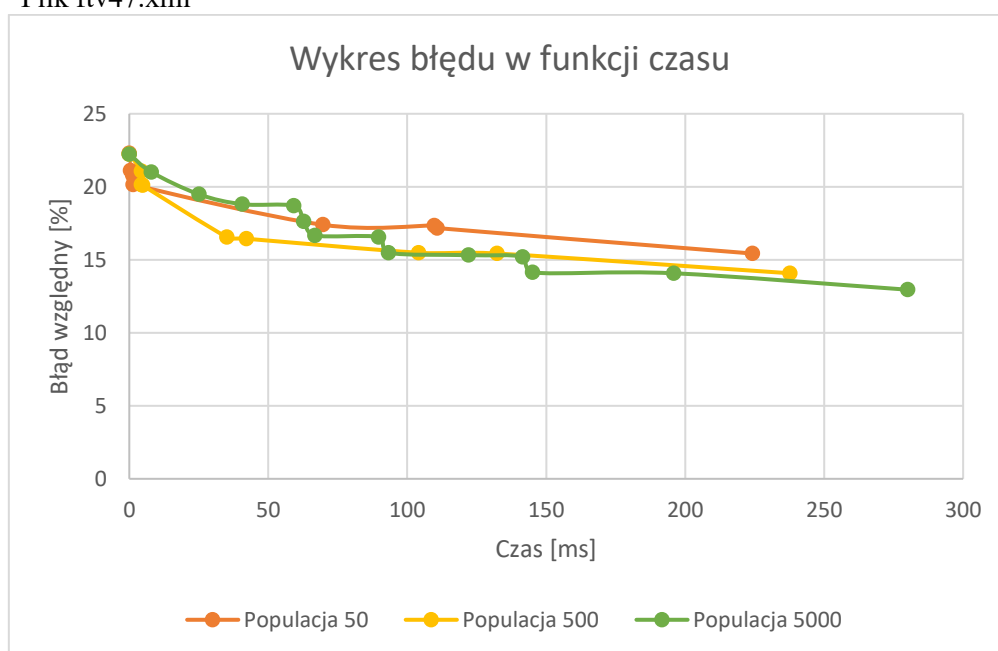
			58,79	2736	10,99	34,57	2868	16,35
			59,62	2733	10,87	34,84	2867	16,31
			61,70	2731	10,79	34,96	2866	16,27
			63,96	2728	10,67	35,02	2865	16,23
			66,81	2724	10,51	35,02	2864	16,19
			73,37	2719	10,30	35,09	2863	16,15
			75,22	2718	10,26	35,30	2862	16,11
			80,26	2717	10,22	35,37	2861	16,06
			80,80	2710	9,94	35,37	2858	15,94
						35,51	2855	15,82
						35,58	2854	15,78
						35,71	2853	15,74
						36,15	2846	15,46
						36,54	2845	15,42
						36,72	2844	15,38
						36,92	2843	15,33
						36,92	2840	15,21
						37,30	2839	15,17
						37,36	2838	15,13
						37,75	2834	14,97
						37,99	2833	14,93
						38,05	2831	14,85
						38,12	2827	14,69
						38,31	2826	14,65
						38,37	2825	14,60
						38,37	2823	14,52
						38,44	2822	14,48
						38,50	2821	14,44
						38,63	2820	14,40
						38,70	2818	14,32
						38,82	2817	14,28
						38,88	2814	14,16
						39,26	2812	14,08
						39,32	2810	14,00
						39,44	2808	13,91
						39,57	2807	13,87
						39,89	2805	13,79
						40,08	2804	13,75
						40,08	2803	13,71
						40,34	2801	13,63
						40,85	2800	13,59
						41,10	2799	13,55
						41,23	2795	13,39
						41,72	2794	13,35
						42,35	2792	13,27
						42,77	2789	13,14

						43,34	2788	13,10
						43,65	2784	12,94
						45,10	2783	12,90
						45,60	2782	12,86
						47,32	2777	12,66
						48,39	2774	12,54
						48,76	2771	12,41
						49,46	2770	12,37
						49,83	2769	12,33
						50,76	2765	12,17
						51,53	2764	12,13
						52,34	2762	12,05
						53,35	2759	11,93
						53,61	2753	11,68
						60,30	2751	11,60
						60,87	2750	11,56
						61,12	2747	11,44
						61,36	2746	11,40
						61,43	2741	11,20
						61,80	2738	11,08
						61,80	2737	11,03
						62,30	2736	10,99
						62,72	2734	10,91
						66,25	2724	10,51
						68,41	2722	10,43
						68,89	2721	10,39
						68,95	2718	10,26
						70,06	2717	10,22
						70,12	2715	10,14
						70,31	2705	9,74
						70,86	2704	9,70
						71,10	2703	9,66
						71,16	2697	9,41
						71,41	2696	9,37
						71,60	2695	9,33
						72,77	2693	9,25
						74,74	2690	9,13
						74,86	2689	9,09
						75,79	2687	9,01
						76,07	2686	8,97
						81,23	2685	8,92
						82,35	2683	8,84
						82,66	2678	8,64
						83,06	2676	8,56
						83,54	2674	8,48
						83,94	2673	8,44

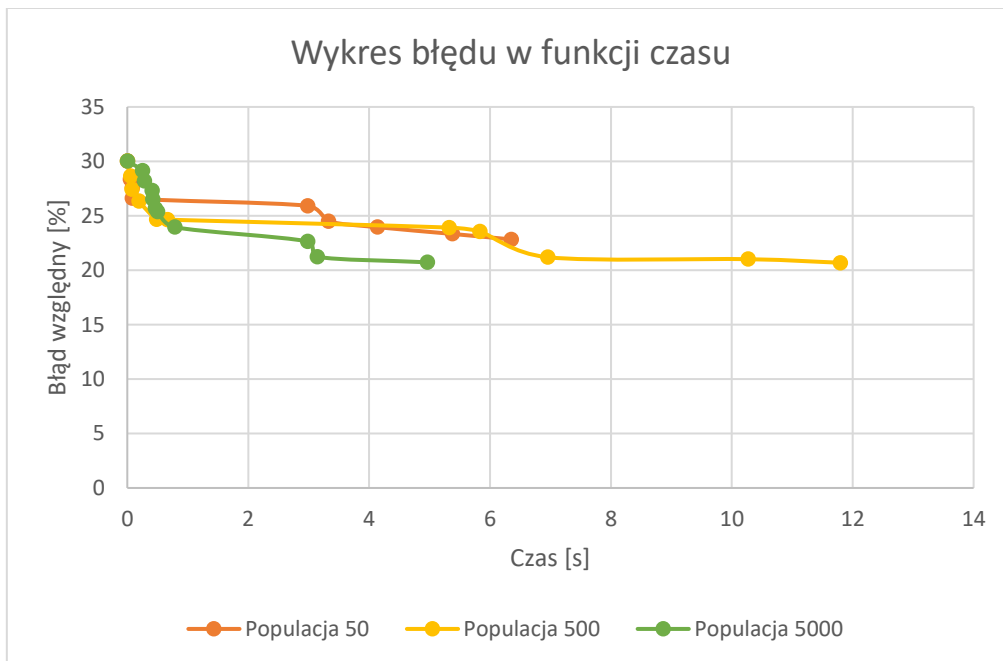
						84,00	2671	8,36
						84,26	2670	8,32
						84,78	2669	8,28
						84,90	2668	8,24
						84,96	2666	8,15
						85,03	2664	8,07
						85,09	2663	8,03
						85,15	2662	7,99
						85,27	2661	7,95
						85,58	2660	7,91
						85,71	2659	7,87

3.2.3 Wykresy porównawcze dla testowanych wielkości populacji

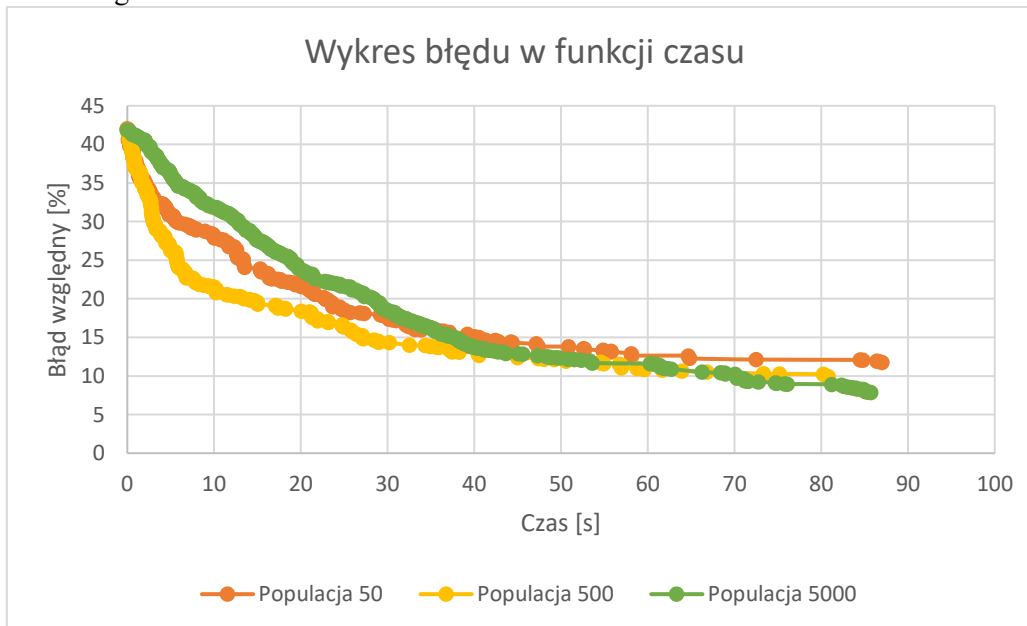
3.2.3.1 Plik ftv47.xml



3.2.3.2 Plik ftv170.xml



3.2.3.3 Plik rbg403.xml



3.2.4 Porównanie Tabu Search z algorytmem Genetycznym o populacji 500

3.2.4.1 Tabele

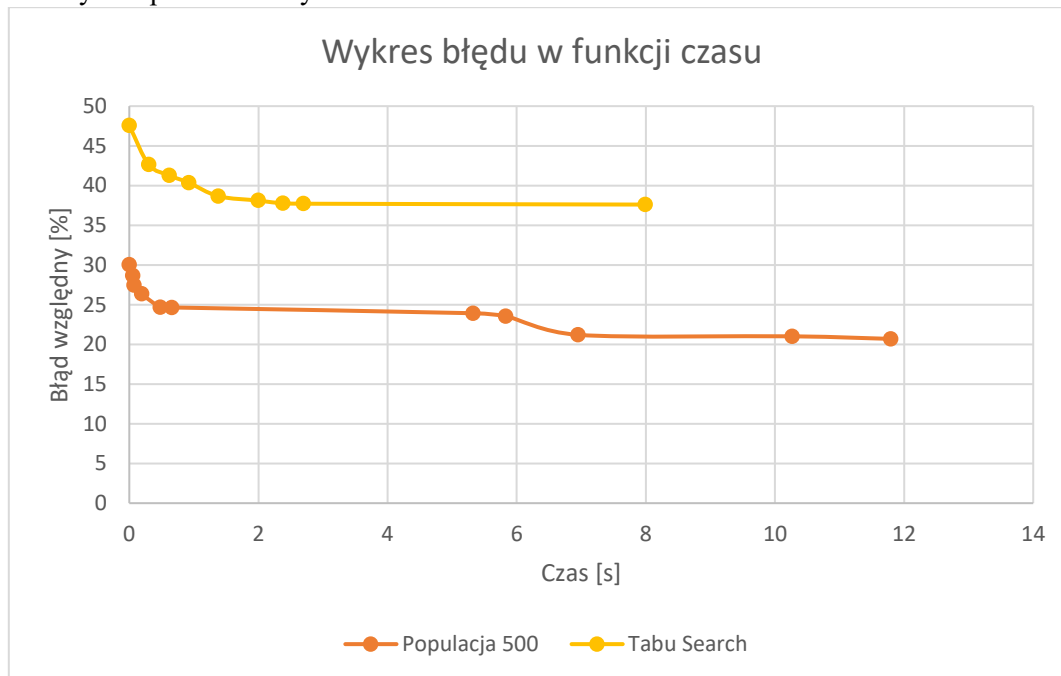
Tabu Search		
Czas[s]	Droga	Błąd względny [%]
0	4065	47,54991
0,300301	3930	42,64973
0,619732	3892	41,27042
0,91993	3867	40,36298
1,379878	3820	38,65699
1,996375	3805	38,11252
2,38037	3795	37,74955

Populacja 5000		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,051705	3544	28,63884
0,074535	3511	27,44102
0,190475	3481	26,35209
0,482607	3435	24,6824
0,660194	3434	24,6461
5,323038	3414	23,92015

2,694059	3794	37,71325
7,993527	3791	37,60436

5,830486	3404	23,55717
6,952452	3339	21,19782
10,26818	3334	21,01633
11,79429	3325	20,68966

3.2.4.2 Wykres porównawczy



3.2.5 Omówienie wyników testów

Algorytm genetyczny o populacji 500 okazał się zdecydowanie lepszy od algorytmu tabu search zaimplementowanego podczas poprzedniego projektu. Jednak wymieniając się informacjami o wynikach z innymi studentami, wiem, że moje algorytmy zarówno Tabu jak i wyżarzanie nie były znakomite. Więc pewnie z tego wziął się ten wynik porównania.

Jeśli chodzi o główny temat tych testów, to pragnę zauważyć, że z wynikiem 2:1 wygrała populacja o rozmiarze 5000, jednakże w tej jednej przegranej, czyli pliku ftv17.xml różnica była rzędu części setnych procenta (0.036 %). Warto również zauważyć, iż dla tych mniejszych plików zmiany dokonywały się z największą częstotliwością w pierwszych 10% czasu przeznaczonego na testy, jedynym plikiem dla którego zmiany odbywały się praktycznie przez cały okres testów okazał się rbg403.xml. Analizując szczegółowo powyższe wykresy możemy również zauważyć, iż populacja o rozmiarze 500 najbardziej nadaje się do bardzo krótkich testów. Tzn. w początkowej fazie trwania algorytmu najczęściej poprawiają się wyniki i są one znacząco lepsze od innych populacji.

3.3 Badanie wpływu wielkości populacji na drugą metodę krzyżowania

3.3.1 Wstęp

W tym punkcie testowanym parametrem również była wielkość populacji, ale dla innej metody krzyżowania. Dodatkowo została przeprowadzona analiza porównawcza metod krzyżowania. Testy zostały przeprowadzone zgodnie z opisem w pkt. 3.1. Parametry testu:

- Metoda krzyżowania – Partially Mapped Crossover
- Metoda mutacji – Swap Mutation
- Współczynnik krzyżowania – 0.8
- Współczynnik mutacji – 0.01

3.3.2 Tabele z wynikami testów

3.3.2.1 Plik ftv47.xml

Populacja 50		
Czas[ms]	Droga	Błąd względny [%]
0	2172	22,2973
23,805	2108	18,69369
837,161	2051	15,48423
865,358	2050	15,42793

Populacja 500		
Czas[ms]	Droga	Błąd względny [%]
0	2171	22,24099
10,052	2133	20,10135
14,735	2113	18,97523
39,505	2104	18,46847
378,127	2031	14,35811
477,121	2030	14,3018

Populacja 5000		
Czas[ms]	Droga	Błąd względny [%]
0	2171	22,24099
26,202	2151	21,11486
79,832	2145	20,77703
79,837	2144	20,72072
86,9	2133	20,10135
133,52	2113	18,97523
338,869	2050	15,42793
430,078	2031	14,35811
500,483	2030	14,3018

3.3.2.2 Plik ftv170.xml

Populacja 50		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,432509	3559	29,1833
2,538509	3536	28,34846
5,285208	3522	27,84029
7,138028	3476	26,1706
8,781752	3454	25,37205
10,65959	3442	24,93648
15,11125	3403	23,52087
17,81992	3381	22,72232

Populacja 500		
Czas[s]	Droga	Błąd względny [%]
0	3689	33,902
0,297427	3675	33,39383
0,555119	3659	32,81307
1,142276	3569	29,54628
7,368979	3545	28,67514
7,532475	3499	27,00544
8,585145	3468	25,88022
10,07609	3434	24,6461
43,22518	3381	22,72232

Populacja 5000		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,467371	3536	28,34846
1,505061	3532	28,20327
1,616605	3531	28,16697
1,892572	3501	27,07804
2,45964	3486	26,53358
2,869157	3473	26,06171
3,404575	3451	25,26316
3,599325	3434	24,6461
3,617561	3412	23,84755
6,490934	3381	22,72232
10,60991	3362	22,03267
18,36881	3331	20,90744

3.3.2.3 Plik rbg403.xml

Populacja 50			Populacja 500			Populacja 5000		
Czas[s]	Droga	Błąd względny [%]	Czas[s]	Droga	Błąd względny [%]	Czas[s]	Droga	Błąd względny [%]
0,00	3502	42,07	0,00	3497	41,87	0,00	3497	41,87
0,28	3501	42,03	0,08	3496	41,83	0,10	3487	41,46
0,29	3496	41,83	0,08	3488	41,50	0,74	3478	41,10
0,35	3495	41,78	0,09	3485	41,38	0,84	3474	40,93
0,39	3492	41,66	0,11	3474	40,93	0,89	3472	40,85
0,40	3490	41,58	0,21	3472	40,85	1,11	3469	40,73
0,58	3479	41,14	0,22	3468	40,69	1,20	3463	40,49
0,69	3476	41,01	0,23	3467	40,65	1,24	3461	40,41
0,79	3473	40,89	0,23	3459	40,32	1,29	3457	40,24
0,79	3468	40,69	0,25	3456	40,20	1,33	3455	40,16
1,04	3467	40,65	0,27	3455	40,16	1,38	3452	40,04
1,05	3465	40,57	0,27	3452	40,04	1,47	3446	39,80
1,21	3464	40,53	0,28	3445	39,76	1,51	3443	39,68
1,25	3453	40,08	0,30	3444	39,72	1,56	3439	39,51
1,39	3450	39,96	0,30	3441	39,59	1,69	3438	39,47
1,41	3444	39,72	0,32	3440	39,55	1,73	3433	39,27
1,62	3438	39,47	0,36	3438	39,47	1,82	3429	39,11
1,71	3435	39,35	0,42	3435	39,35	1,82	3426	38,99
1,72	3434	39,31	0,58	3428	39,07	1,95	3422	38,82
1,85	3433	39,27	0,63	3427	39,03	2,09	3420	38,74
1,87	3420	38,74	0,63	3426	38,99	2,09	3415	38,54
2,18	3419	38,70	0,66	3425	38,95	2,26	3414	38,50
2,26	3416	38,58	0,67	3422	38,82	2,35	3413	38,46
2,35	3413	38,46	0,67	3420	38,74	2,62	3411	38,38
2,44	3408	38,26	0,69	3408	38,26	2,70	3410	38,34
2,45	3407	38,22	0,73	3407	38,22	2,75	3404	38,09
2,57	3400	37,93	0,76	3405	38,13	3,02	3401	37,97
2,69	3398	37,85	0,88	3403	38,05	3,41	3400	37,93
2,79	3397	37,81	0,95	3394	37,69	3,51	3397	37,81
2,93	3389	37,48	1,03	3390	37,53	3,59	3395	37,73
3,20	3387	37,40	1,08	3386	37,36	3,68	3394	37,69
3,27	3385	37,32	1,09	3383	37,24	3,77	3393	37,65
3,32	3378	37,04	1,11	3380	37,12	3,86	3392	37,61
3,54	3377	37,00	1,14	3379	37,08	3,90	3391	37,57
3,95	3373	36,84	1,18	3377	37,00	4,22	3390	37,53
4,03	3368	36,63	1,28	3376	36,96	4,26	3389	37,48
4,17	3366	36,55	1,35	3369	36,67	4,31	3381	37,16
4,33	3365	36,51	1,41	3363	36,43	4,35	3380	37,12
4,35	3355	36,11	1,74	3360	36,31	4,40	3378	37,04
4,42	3349	35,86	1,80	3357	36,19	4,53	3377	37,00
4,42	3347	35,78	1,84	3354	36,06	4,66	3375	36,92
4,63	3342	35,58	1,99	3352	35,98	4,71	3374	36,88
5,01	3341	35,54	2,13	3348	35,82	4,76	3372	36,80

5,08	3340	35,50	2,25	3347	35,78	5,02	3371	36,75
5,11	3333	35,21	2,30	3346	35,74	5,29	3370	36,71
5,24	3330	35,09	2,36	3344	35,66	5,52	3369	36,67
5,29	3328	35,01	2,37	3337	35,38	5,61	3368	36,63
5,38	3327	34,97	2,50	3328	35,01	5,74	3367	36,59
5,93	3323	34,81	2,89	3326	34,93	5,74	3365	36,51
6,25	3322	34,77	2,91	3324	34,85	5,74	3364	36,47
6,48	3321	34,73	2,94	3322	34,77	5,92	3363	36,43
6,51	3319	34,65	2,97	3320	34,69	5,92	3362	36,39
6,76	3318	34,60	3,00	3319	34,65	5,97	3361	36,35
6,84	3314	34,44	3,16	3318	34,60	6,10	3360	36,31
7,18	3309	34,24	3,34	3317	34,56	6,15	3359	36,27
7,80	3303	34,00	3,35	3314	34,44	6,15	3356	36,15
7,96	3300	33,87	3,38	3312	34,36	6,34	3353	36,02
7,99	3298	33,79	3,43	3310	34,28	6,43	3351	35,94
8,01	3294	33,63	3,56	3309	34,24	6,52	3350	35,90
8,75	3293	33,59	3,58	3306	34,12	6,56	3347	35,78
8,93	3292	33,55	3,58	3305	34,08	6,70	3346	35,74
9,13	3282	33,14	3,66	3303	34,00	6,70	3345	35,70
9,15	3280	33,06	3,93	3299	33,83	6,74	3344	35,66
9,35	3278	32,98	4,15	3295	33,67	6,75	3343	35,62
9,39	3277	32,94	4,22	3286	33,31	6,88	3342	35,58
9,44	3271	32,70	5,55	3285	33,27	6,97	3341	35,54
9,59	3270	32,66	5,98	3284	33,23	7,06	3337	35,38
9,59	3269	32,62	5,98	3281	33,10	7,19	3328	35,01
9,65	3262	32,33	6,28	3274	32,82	7,37	3327	34,97
9,65	3259	32,21	6,32	3271	32,70	7,68	3326	34,93
9,74	3254	32,01	6,53	3264	32,41	7,82	3325	34,89
9,85	3253	31,97	6,64	3260	32,25	7,82	3322	34,77
9,93	3250	31,85	6,74	3256	32,09	7,96	3321	34,73
10,30	3246	31,68	7,74	3255	32,05	8,09	3320	34,69
11,14	3244	31,60	8,05	3254	32,01	8,18	3319	34,65
11,38	3241	31,48	8,41	3252	31,93	8,23	3313	34,40
11,64	3240	31,44	8,43	3245	31,64	8,32	3309	34,24
11,87	3239	31,40	8,75	3242	31,52	8,36	3308	34,20
12,13	3228	30,95	9,58	3241	31,48	8,50	3307	34,16
12,95	3220	30,63	9,67	3239	31,40	8,50	3306	34,12
13,18	3215	30,43	9,90	3237	31,32	8,50	3303	34,00
13,31	3213	30,34	9,92	3234	31,20	8,59	3301	33,91
13,48	3212	30,30	10,09	3227	30,91	8,60	3300	33,87
13,70	3211	30,26	10,29	3225	30,83	8,78	3299	33,83
13,82	3209	30,18	10,53	3222	30,71	9,30	3298	33,79
13,96	3207	30,10	10,60	3220	30,63	9,30	3290	33,47
14,00	3205	30,02	10,81	3218	30,55	9,52	3287	33,35
14,01	3202	29,90	10,84	3215	30,43	9,76	3278	32,98
14,08	3199	29,78	10,87	3214	30,39	9,95	3277	32,94

14,09	3196	29,66	10,88	3212	30,30	10,04	3270	32,66
14,21	3195	29,61	10,88	3211	30,26	10,09	3269	32,62
14,45	3194	29,57	11,06	3209	30,18	10,41	3266	32,49
14,61	3193	29,53	11,64	3204	29,98	10,67	3265	32,45
14,70	3181	29,05	12,05	3200	29,82	10,99	3256	32,09
14,79	3179	28,97	12,14	3198	29,74	11,08	3251	31,89
15,11	3178	28,92	12,15	3191	29,45	12,31	3250	31,85
15,27	3175	28,80	12,17	3189	29,37	12,44	3238	31,36
15,35	3165	28,40	12,25	3188	29,33	13,06	3237	31,32
15,46	3159	28,15	12,29	3182	29,09	13,37	3236	31,28
15,50	3155	27,99	12,89	3180	29,01	13,42	3234	31,20
15,50	3152	27,87	12,90	3178	28,92	13,55	3226	30,87
15,50	3148	27,71	12,99	3175	28,80	14,00	3215	30,43
15,52	3147	27,67	13,47	3168	28,52	14,27	3214	30,39
15,55	3136	27,22	13,67	3156	28,03	14,86	3208	30,14
16,27	3132	27,06	14,10	3155	27,99	15,08	3207	30,10
16,28	3130	26,98	14,18	3144	27,55	15,32	3206	30,06
16,40	3129	26,94	14,32	3143	27,51	15,37	3204	29,98
16,44	3123	26,69	14,72	3141	27,42	15,60	3203	29,94
16,45	3120	26,57	15,07	3136	27,22	15,69	3201	29,86
16,59	3112	26,25	15,45	3132	27,06	15,78	3200	29,82
16,68	3107	26,04	15,56	3124	26,73	15,83	3199	29,78
17,78	3106	26,00	15,76	3118	26,49	16,00	3197	29,70
18,03	3105	25,96	16,06	3117	26,45	16,00	3196	29,66
18,24	3103	25,88	16,20	3114	26,33	16,22	3193	29,53
18,54	3102	25,84	16,46	3106	26,00	16,40	3192	29,49
18,74	3101	25,80	16,53	3099	25,72	16,58	3190	29,41
18,84	3098	25,68	16,71	3096	25,60	16,63	3180	29,01
19,05	3095	25,56	17,28	3095	25,56	16,80	3174	28,76
19,94	3094	25,52	18,13	3094	25,52	17,07	3173	28,72
20,14	3089	25,31	18,91	3091	25,40	17,16	3170	28,60
20,51	3087	25,23	19,03	3086	25,19	17,56	3167	28,48
20,55	3086	25,19	20,17	3072	24,62	18,05	3157	28,07
20,59	3081	24,99	21,52	3064	24,30	18,64	3149	27,75
20,80	3080	24,95	23,15	3057	24,02	19,09	3148	27,71
20,94	3079	24,91	23,33	3056	23,98	19,37	3140	27,38
21,13	3078	24,87	23,39	3051	23,77	19,61	3139	27,34
21,28	3077	24,83	23,65	3047	23,61	20,19	3130	26,98
21,32	3076	24,79	23,81	3042	23,41	20,63	3125	26,77
21,58	3075	24,75	25,15	3036	23,16	20,94	3122	26,65
21,60	3074	24,71	25,17	3035	23,12	20,94	3121	26,61
21,98	3073	24,67	25,94	3034	23,08	21,13	3117	26,45
22,03	3071	24,58	26,78	3027	22,80	21,27	3116	26,41
22,05	3070	24,54	27,10	3026	22,76	21,45	3114	26,33
22,35	3069	24,50	27,19	3020	22,52	21,80	3113	26,29
22,40	3058	24,06	27,95	3017	22,39	22,58	3112	26,25

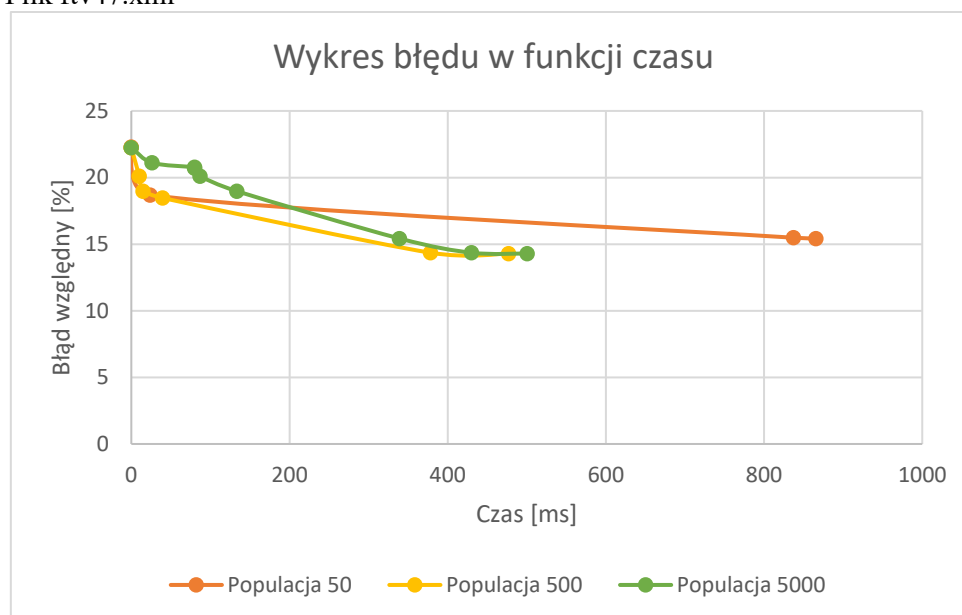
23,07	3056	23,98	28,01	3016	22,35	22,75	3111	26,21
23,63	3055	23,94	28,06	3015	22,31	23,44	3110	26,17
23,66	3053	23,85	28,28	3014	22,27	23,75	3109	26,13
23,95	3051	23,77	28,49	3012	22,19	23,75	3102	25,84
24,42	3046	23,57	29,34	3008	22,03	24,51	3098	25,68
24,59	3045	23,53	30,09	3003	21,83	25,42	3097	25,64
24,70	3039	23,29	30,30	3002	21,78	26,04	3095	25,56
24,85	3036	23,16	30,82	3001	21,74	26,13	3092	25,44
24,94	3034	23,08	31,14	2999	21,66	26,85	3090	25,35
25,28	3027	22,80	31,51	2997	21,58	26,99	3089	25,31
25,40	3024	22,68	31,95	2992	21,38	27,35	3084	25,11
25,72	3021	22,56	32,00	2991	21,34	27,40	3079	24,91
26,53	3020	22,52	32,34	2990	21,30	27,66	3073	24,67
26,75	3018	22,43	32,34	2980	20,89	27,97	3072	24,62
27,08	3009	22,07	33,38	2979	20,85	28,10	3057	24,02
27,32	3008	22,03	34,67	2973	20,61	28,45	3056	23,98
27,52	3007	21,99	34,77	2970	20,49	28,54	3055	23,94
27,68	3005	21,91	34,97	2969	20,45	28,59	3054	23,89
29,18	3003	21,83	35,17	2967	20,37	28,72	3049	23,69
29,44	2996	21,54	35,47	2965	20,28	29,65	3045	23,53
29,46	2995	21,50	35,96	2964	20,24	30,93	3043	23,45
29,84	2988	21,22	36,05	2959	20,04	31,52	3042	23,41
29,85	2984	21,05	36,38	2958	20,00	31,60	3041	23,37
31,43	2982	20,97	37,01	2953	19,80	31,65	3034	23,08
31,55	2972	20,57	37,28	2952	19,76	31,96	3033	23,04
34,22	2970	20,49	37,60	2951	19,72	31,96	3032	23,00
34,30	2966	20,32	38,38	2950	19,68	32,09	3031	22,96
35,03	2965	20,28	38,69	2949	19,63	32,46	3026	22,76
35,15	2963	20,20	38,86	2948	19,59	32,59	3025	22,72
35,30	2959	20,04	39,21	2940	19,27	32,90	3023	22,64
35,44	2954	19,84	39,55	2930	18,86	33,03	3017	22,39
35,90	2944	19,43	43,56	2928	18,78	33,57	3016	22,35
37,51	2934	19,03	44,77	2921	18,50	33,79	3008	22,03
37,70	2928	18,78	46,81	2919	18,42	33,93	3007	21,99
39,35	2920	18,46	47,17	2918	18,38	34,75	3006	21,95
39,50	2917	18,34	48,79	2917	18,34	35,07	3003	21,83
39,61	2910	18,05	50,57	2914	18,22	35,69	3002	21,78
40,74	2909	18,01	51,78	2913	18,17	35,96	2994	21,46
41,17	2908	17,97	53,47	2910	18,05	36,65	2990	21,30
41,42	2907	17,93	54,66	2908	17,97	37,72	2989	21,26
41,69	2906	17,89	55,32	2905	17,85	38,04	2979	20,85
41,69	2898	17,57	55,48	2904	17,81	39,15	2976	20,73
42,54	2894	17,40	56,00	2901	17,69	44,55	2966	20,32
42,62	2890	17,24	56,03	2899	17,61	47,31	2965	20,28
43,60	2887	17,12	56,16	2897	17,53	47,75	2963	20,20
43,60	2886	17,08	56,29	2896	17,48	48,39	2962	20,16

43,64	2885	17,04	56,98	2886	17,08	48,86	2961	20,12
43,85	2881	16,88	57,05	2884	17,00	48,91	2956	19,92
44,42	2879	16,80	57,12	2882	16,92	49,75	2950	19,68
45,58	2872	16,51	57,27	2879	16,80	50,44	2945	19,47
45,89	2864	16,19	57,36	2873	16,55	50,95	2942	19,35
51,19	2858	15,94	57,58	2871	16,47	51,27	2938	19,19
51,89	2851	15,66	58,09	2859	15,98	54,97	2937	19,15
52,74	2849	15,58	58,48	2852	15,70	55,42	2936	19,11
54,18	2847	15,50	58,53	2849	15,58	56,54	2933	18,99
55,48	2846	15,46	60,17	2846	15,46	56,63	2925	18,66
56,31	2843	15,33	60,81	2844	15,38	56,89	2922	18,54
56,34	2840	15,21	61,88	2843	15,33	57,07	2919	18,42
56,46	2835	15,01	62,64	2842	15,29	57,29	2913	18,17
57,88	2834	14,97	63,35	2835	15,01	57,66	2911	18,09
58,24	2833	14,93	63,67	2834	14,97	57,97	2903	17,77
58,85	2828	14,73	65,30	2833	14,93	58,28	2898	17,57
59,77	2825	14,60	65,67	2828	14,73	58,86	2890	17,24
60,18	2822	14,48	65,79	2820	14,40	59,34	2888	17,16
63,27	2819	14,36	66,85	2816	14,24	59,97	2884	17,00
64,00	2818	14,32	67,11	2814	14,16	60,87	2883	16,96
64,50	2817	14,28	68,02	2809	13,96	61,45	2882	16,92
65,08	2814	14,16	68,15	2808	13,91	61,94	2879	16,80
67,28	2807	13,87	68,38	2807	13,87	62,03	2873	16,55
69,35	2806	13,83	69,03	2806	13,83	62,36	2869	16,39
69,55	2805	13,79	69,46	2799	13,55	62,78	2866	16,27
70,51	2804	13,75	69,89	2798	13,51	63,69	2864	16,19
71,01	2803	13,71	71,09	2790	13,18	64,34	2863	16,15
71,12	2798	13,51	72,23	2781	12,82	65,24	2861	16,06
72,47	2797	13,47	77,80	2775	12,58	65,83	2856	15,86
73,62	2793	13,31	77,93	2774	12,54	68,08	2855	15,82
74,70	2791	13,23	78,22	2768	12,29	68,35	2847	15,50
85,30	2788	13,10	79,13	2759	11,93	69,20	2846	15,46
86,82	2786	13,02	80,39	2758	11,89	70,67	2841	15,25
89,39	2773	12,49	80,66	2756	11,81	71,21	2839	15,17
			81,54	2748	11,48	72,81	2831	14,85
			83,17	2744	11,32	75,69	2830	14,81
			83,88	2743	11,28	75,74	2829	14,77
			88,94	2735	10,95	75,79	2827	14,69
			89,70	2729	10,71	76,91	2822	14,48
			89,83	2725	10,55	80,10	2819	14,36
						80,28	2818	14,32
						81,88	2814	14,16
						82,73	2811	14,04
						83,66	2810	14,00
						83,97	2806	13,83
						84,82	2797	13,47

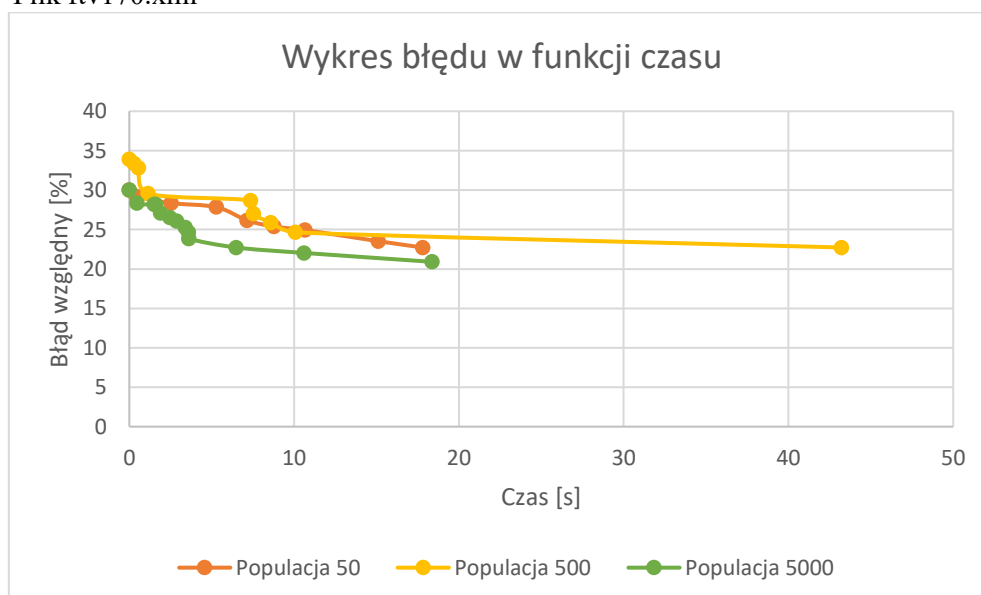
						84,90	2796	13,43
						85,39	2793	13,31
						85,44	2792	13,27
						85,88	2788	13,10
						88,70	2785	12,98

3.3.3 Wykresy porównawcze dla testowanych wielkości populacji

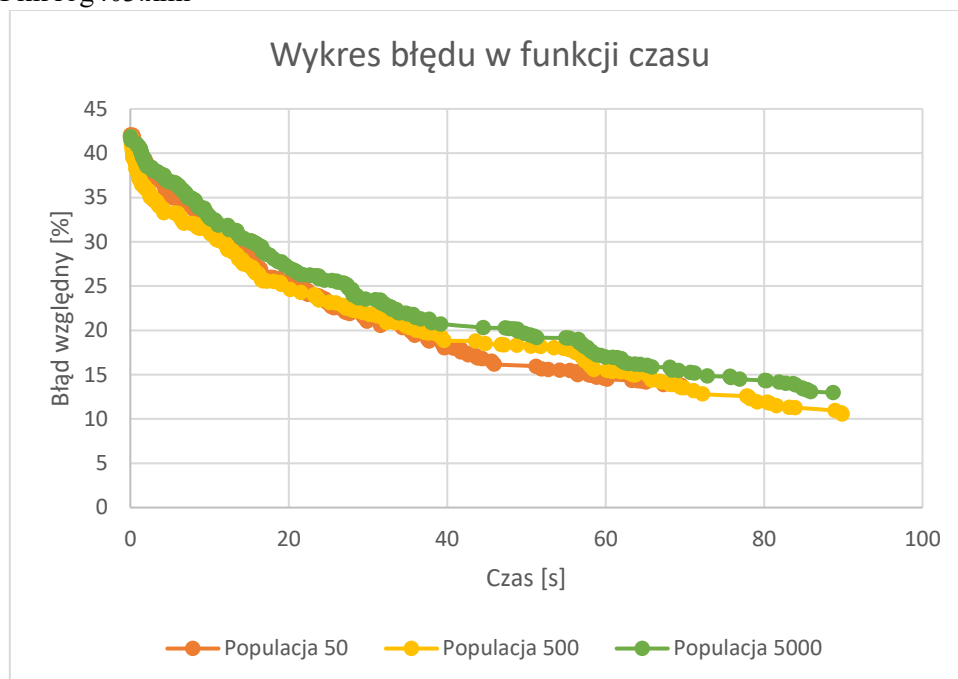
3.3.3.1 Plik ftv47.xml



3.3.3.2 Plik ftv170.xml



3.3.3.3 Plik rbg403.xml



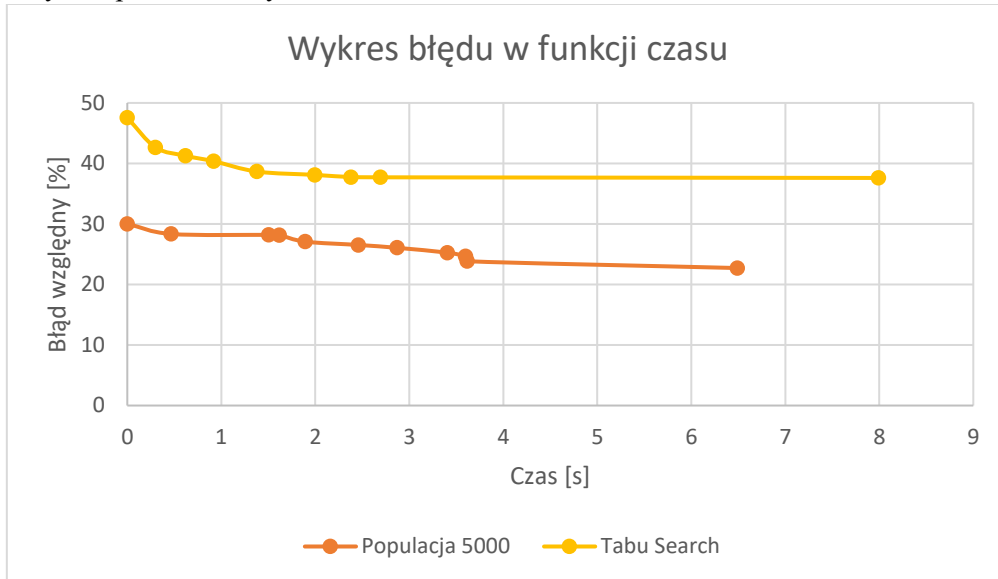
3.3.4 Porównanie Tabu Search z algorytmem Genetycznym o populacji 500

3.3.4.1 Tabele

Tabu Search		
Czas[s]	Droga	Błąd względny [%]
0	4065	47,54991
0,300301	3930	42,64973
0,619732	3892	41,27042
0,91993	3867	40,36298
1,379878	3820	38,65699
1,996375	3805	38,11252
2,38037	3795	37,74955
2,694059	3794	37,71325
7,993527	3791	37,60436

Populacja 5000		
Czas[s]	Droga	Błąd względny [%]
0	3582	30,01815
0,467371	3536	28,34846
1,505061	3532	28,20327
1,616605	3531	28,16697
1,892572	3501	27,07804
2,45964	3486	26,53358
2,869157	3473	26,06171
3,404575	3451	25,26316
3,599325	3434	24,6461
3,617561	3412	23,84755
6,490934	3381	22,72232
10,60991	3362	22,03267
18,36881	3331	20,90744

3.3.4.2 Wykres porównawczy

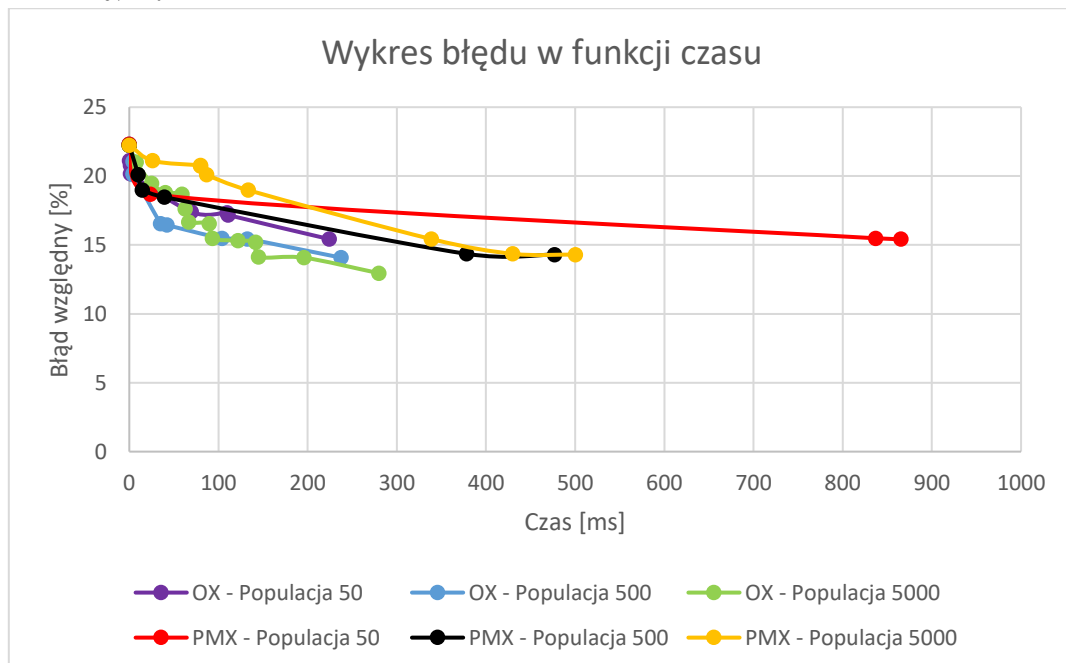


3.3.5 Omówienie wyników testów

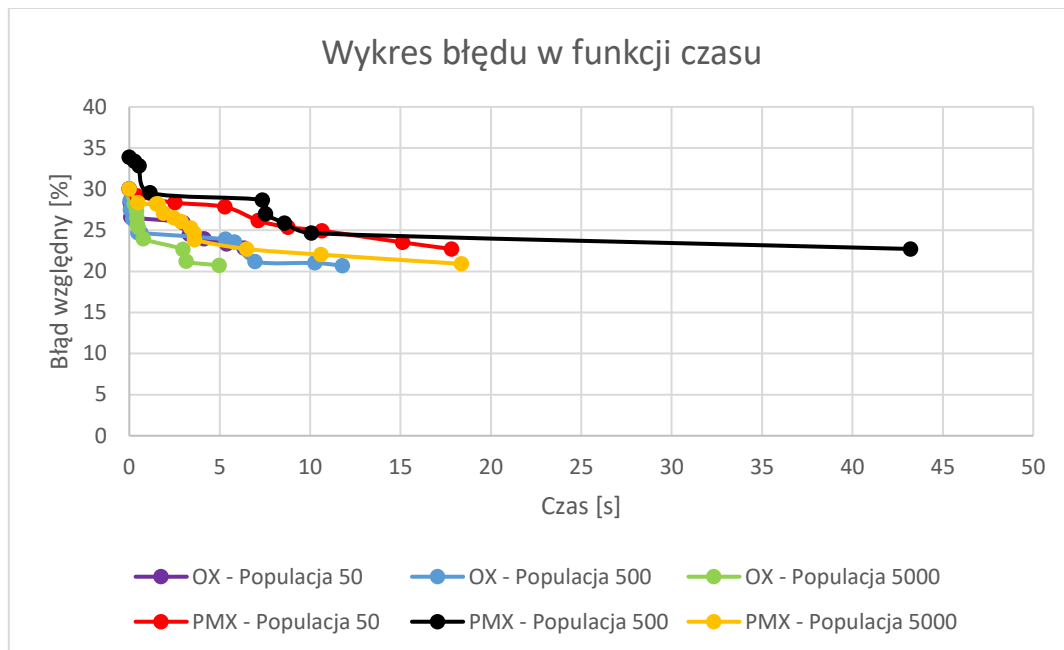
Podobnie jak poprzednio algorytm genetyczny okazał się znacznie lepszy od tabu search, wiąże się to z tym co opisałem w punkcie 3.2.5. Natomiast jeśli chodzi o wpływ rozmiaru populacji to mamy tutaj remis pomiędzy rozmiarami 500 i 5000. Dlatego remis, że mimo nie parzystej liczby testów, dla pliku ftv47.xml oba rozmiary uzyskały jednakowy wynik. Najważniejszą zmianę można zauważyć dla pliku ftv170.xml gdzie tym razem znacząco przegrał rozmiar 500.

3.4 Wpływ wielkości populacji i metody krzyżowania

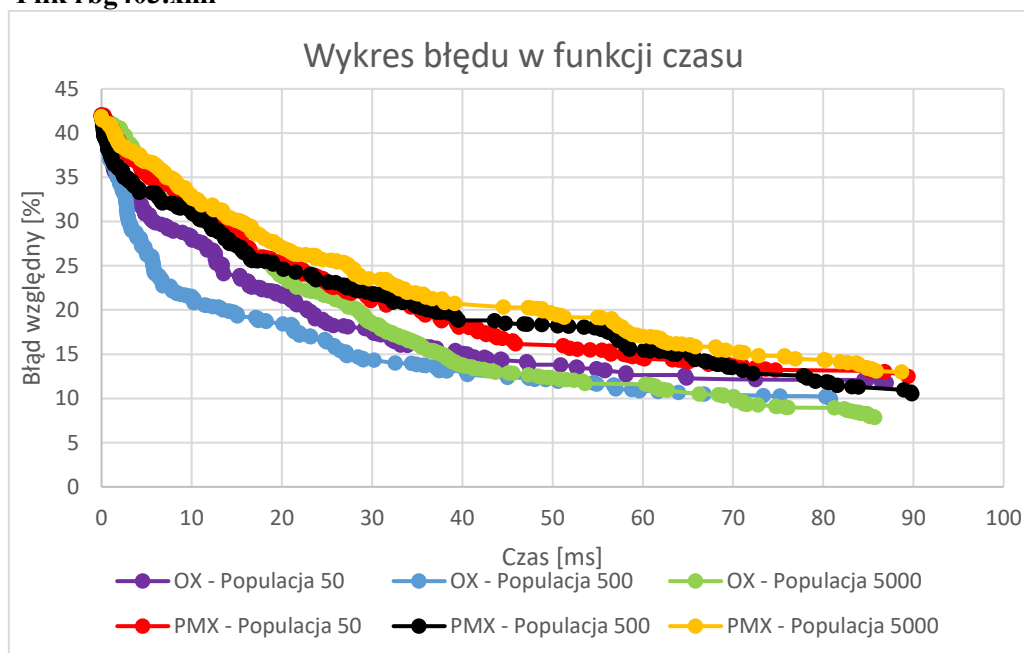
3.4.1 Plik ftv47.xml



3.4.2 Plik ftv170.xml



3.4.3 Plik rgb403.xml



3.4.4 Omówienie

Analizując powyższe przypadki możemy zauważyć, że w większości przypadków Order Crossover wypada lepiej od Partially Mapped Crossover. Dodatkowo można zauważyć, iż dla większości przypadków populacja o rozmiarze 5000 wypada najlepiej. Biorąc pod uwagę uzyskane wyniki, do kolejnych testów zostały wybrane następujące kombinacje:

- plik ftv47.xml – Order Crossover – populacja 5000
- plik ftv170.xml – Order Crossover – populacja 500
- plik rgb403.xml – Order Crossover – populacja 5000

Omówienie kolejnych testów znajduje się poniżej.

3.5 Wpływ współczynnika mutacji

3.5.1 Wstęp

W tym punkcie testowanym parametrem był współczynnik mutacji (0.02, 0.05, 0.1). Testy zostały przeprowadzone zgodnie z opisem w pkt. 3.1. Parametry testu dobrane zostały po przeanalizowaniu poprzednich testów i opisane w punkcie 3.4.4. Wyniki będą porównywane względem tych wybranych parametrów dla których współczynnik mutacji wynosił 0.01.

3.5.2 Tabele z wynikami testów

3.5.2.1 Plik ftv47.xml

Wsp. Mutacji 0.02		
Czas[ms]	Droga	Błąd względny [%]
0,00	2171	22,24099
22,65	2154	21,28378
30,21	2151	21,11486
66,87	2134	20,15766
92,54	2133	20,10135
92,565	2108	18,69369
123,2	2070	16,55405
154,136	2050	15,42793
209,233	2048	15,31532
255,03	2031	14,35811
363,869	2030	14,3018

Wsp. Mutacji 0.1		
Czas[ms]	Droga	Błąd względny [%]
0	2171	22,24099
14,644	2154	21,28378
30,191	2151	21,11486
46,129	2150	21,05856
53,592	2133	20,10135
96,839	2123	19,53829
135,076	2111	18,86261
135,233	2108	18,69369
154,477	2091	17,73649
154,66	2070	16,55405
192,95	2050	15,42793
231,17	2031	14,35811
283,58	2030	14,3018

Wsp. Mutacji 0.05		
Czas[ms]	Droga	Błąd względny [%]
0	2171	22,24099
22,649	2154	21,28378
30,207	2151	21,11486
66,869	2134	20,15766
92,544	2133	20,10135
92,565	2108	18,69369
123,2	2070	16,55405
154,136	2050	15,42793
209,233	2048	15,31532
255,03	2031	14,35811
363,869	2030	14,3018

3.5.2.2 Plik ftv170.xml

Wsp. Mutacji 0.02		
Czas[ms]	Droga	Błąd względny [%]
0,00	3582	30,01815
0,44	3569	29,54628
2,12	3519	27,7314
2,96	3501	27,07804
3,60	3451	25,26316
8,58547	3420	24,13793
9,919626	3412	23,84755
13,44828	3381	22,72232

Wsp. Mutacji 0.1		
Czas[ms]	Droga	Błąd względny [%]
0	3582	30,01815
0,388754	3572	29,65517
0,421216	3544	28,63884
0,550806	3531	28,16697
0,6608	3530	28,13067
0,714798	3505	27,22323
0,932124	3500	27,04174
0,937408	3474	26,098

Wsp. Mutacji 0.05		
Czas[ms]	Droga	Błąd względny [%]
0	3582	30,01815
1,166875	3560	29,2196
1,179269	3538	28,42105
1,474817	3485	26,49728
1,696494	3446	25,08167
3,763931	3420	24,13793
3,96308	3381	22,72232

1,100467	3456	25,44465
1,35	3434	24,6461
2,25	3381	22,72232
16,80	3379	22,64973

3.5.2.3 Plik rbg403.xml

Wsp. Mutacji 0.02			Wsp. Mutacji 0.05			Wsp. Mutacji 0.1		
Czas[s]	Droga	Błąd względny [%]	Czas[s]	Droga	Błąd względny [%]	Czas[s]	Droga	Błąd względny [%]
0,00	3497	41,87	0,00	3497	41,87	0,00	3497	41,87
0,04	3491	41,62	0,10	3496	41,83	0,23	3496	41,83
0,32	3486	41,42	0,15	3495	41,78	0,32	3492	41,66
0,51	3485	41,38	0,33	3494	41,74	0,37	3489	41,54
0,64	3484	41,34	0,52	3489	41,54	0,60	3488	41,50
0,73	3475	40,97	0,70	3487	41,46	0,64	3487	41,46
1,04	3473	40,89	0,75	3484	41,34	0,69	3485	41,38
1,13	3472	40,85	0,79	3479	41,14	0,82	3484	41,34
1,17	3471	40,81	0,84	3478	41,10	0,82	3482	41,26
1,17	3464	40,53	0,93	3475	40,97	0,91	3480	41,18
1,34	3463	40,49	1,10	3460	40,37	0,96	3478	41,10
1,38	3462	40,45	1,36	3458	40,28	1,01	3475	40,97
1,38	3461	40,41	1,41	3457	40,24	1,05	3465	40,57
1,42	3460	40,37	1,41	3443	39,68	1,27	3462	40,45
1,42	3457	40,24	1,53	3440	39,55	1,40	3461	40,41
1,46	3455	40,16	1,71	3430	39,15	1,45	3460	40,37
1,46	3452	40,04	1,75	3429	39,11	1,45	3459	40,32
1,51	3448	39,88	1,79	3427	39,03	1,45	3458	40,28
1,55	3444	39,72	1,83	3424	38,90	1,49	3455	40,16
1,55	3439	39,51	1,88	3422	38,82	1,58	3451	40,00
1,71	3438	39,47	1,96	3421	38,78	1,58	3450	39,96
1,71	3435	39,35	1,96	3419	38,70	1,62	3448	39,88
1,76	3431	39,19	2,01	3417	38,62	1,66	3445	39,76
1,84	3427	39,03	2,05	3412	38,42	1,66	3441	39,59
1,93	3426	38,99	2,09	3410	38,34	1,70	3439	39,51
1,97	3423	38,86	2,13	3408	38,26	1,75	3437	39,43
2,01	3420	38,74	2,17	3407	38,22	1,79	3436	39,39
2,18	3419	38,70	2,17	3402	38,01	1,83	3427	39,03
2,26	3418	38,66	2,22	3401	37,97	1,88	3425	38,95
2,26	3416	38,58	2,22	3398	37,85	1,92	3420	38,74
2,31	3414	38,50	2,26	3397	37,81	1,92	3419	38,70
2,39	3409	38,30	2,26	3396	37,77	2,00	3417	38,62

2,52	3406	38,17	2,30	3388	37,44	2,00	3415	38,54
2,56	3404	38,09	2,39	3384	37,28	2,00	3410	38,34
2,64	3403	38,05	2,47	3380	37,12	2,00	3407	38,22
2,64	3400	37,93	2,51	3375	36,92	2,09	3403	38,05
2,77	3399	37,89	2,52	3373	36,84	2,13	3400	37,93
2,81	3397	37,81	2,64	3369	36,67	2,13	3396	37,77
2,81	3395	37,73	2,77	3367	36,59	2,13	3395	37,73
2,90	3393	37,65	2,77	3360	36,31	2,18	3392	37,61
2,94	3392	37,61	2,77	3357	36,19	2,22	3391	37,57
2,94	3389	37,48	2,90	3355	36,11	2,22	3388	37,44
2,98	3388	37,44	3,03	3354	36,06	2,26	3380	37,12
3,02	3385	37,32	3,03	3353	36,02	2,31	3376	36,96
3,06	3384	37,28	3,03	3351	35,94	2,39	3374	36,88
3,06	3382	37,20	3,07	3350	35,90	2,43	3365	36,51
3,15	3381	37,16	3,11	3349	35,86	2,52	3359	36,27
3,15	3378	37,04	3,11	3343	35,62	2,56	3355	36,11
3,23	3375	36,92	3,15	3337	35,38	2,61	3354	36,06
3,28	3373	36,84	3,32	3330	35,09	2,65	3353	36,02
3,41	3372	36,80	3,41	3321	34,73	2,65	3350	35,90
3,45	3370	36,71	3,58	3320	34,69	2,69	3349	35,86
3,45	3369	36,67	3,58	3317	34,56	2,69	3346	35,74
3,45	3368	36,63	3,62	3314	34,44	2,73	3336	35,33
3,49	3364	36,47	3,62	3313	34,40	2,82	3331	35,13
3,49	3363	36,43	3,66	3309	34,24	2,86	3326	34,93
3,53	3362	36,39	3,71	3308	34,20	2,95	3325	34,89
3,62	3360	36,31	3,71	3303	34,00	3,03	3323	34,81
3,70	3359	36,27	3,79	3299	33,83	3,08	3321	34,73
3,83	3357	36,19	3,87	3290	33,47	3,12	3320	34,69
3,83	3354	36,06	3,96	3289	33,43	3,16	3318	34,60
3,91	3353	36,02	4,05	3287	33,35	3,16	3316	34,52
3,91	3351	35,94	4,09	3286	33,31	3,20	3313	34,40
4,00	3350	35,90	4,17	3285	33,27	3,25	3311	34,32
4,08	3348	35,82	4,26	3284	33,23	3,29	3307	34,16
4,08	3347	35,78	4,26	3283	33,18	3,33	3305	34,08
4,12	3345	35,70	4,34	3282	33,14	3,34	3301	33,91
4,29	3342	35,58	4,43	3281	33,10	3,38	3300	33,87
4,29	3340	35,50	4,56	3280	33,06	3,46	3296	33,71
4,33	3336	35,33	4,56	3279	33,02	3,50	3295	33,67
4,55	3335	35,29	4,60	3276	32,90	3,50	3293	33,59
4,55	3329	35,05	4,73	3275	32,86	3,59	3287	33,35
4,67	3328	35,01	4,77	3272	32,74	3,68	3285	33,27
4,67	3327	34,97	4,89	3270	32,66	3,68	3279	33,02
4,71	3326	34,93	4,94	3269	32,62	3,72	3270	32,66
4,76	3318	34,60	5,02	3267	32,54	3,89	3269	32,62
4,84	3317	34,56	5,07	3263	32,37	3,97	3266	32,49
5,01	3316	34,52	5,15	3262	32,33	3,97	3257	32,13

5,05	3312	34,36	5,15	3261	32,29	4,19	3255	32,05
5,09	3308	34,20	5,15	3259	32,21	4,19	3253	31,97
5,13	3307	34,16	5,23	3258	32,17	4,23	3248	31,76
5,34	3303	34,00	5,28	3255	32,05	4,36	3247	31,72
5,43	3298	33,79	5,41	3254	32,01	4,44	3244	31,60
5,47	3297	33,75	5,49	3253	31,97	4,44	3243	31,56
5,56	3288	33,39	5,49	3251	31,89	4,49	3242	31,52
5,85	3286	33,31	5,58	3248	31,76	4,53	3241	31,48
5,94	3284	33,23	5,62	3247	31,72	4,57	3238	31,36
6,02	3282	33,14	5,62	3240	31,44	4,61	3236	31,28
6,06	3281	33,10	5,83	3238	31,36	4,70	3234	31,20
6,23	3280	33,06	5,92	3237	31,32	4,70	3233	31,16
6,27	3278	32,98	5,96	3235	31,24	4,74	3232	31,12
6,36	3275	32,86	5,96	3234	31,20	4,78	3230	31,03
6,40	3273	32,78	6,00	3233	31,16	4,78	3229	30,99
6,40	3271	32,70	6,04	3231	31,08	4,83	3228	30,95
6,40	3266	32,49	6,05	3229	30,99	4,87	3226	30,87
6,74	3265	32,45	6,05	3226	30,87	4,91	3225	30,83
6,74	3264	32,41	6,13	3219	30,59	4,95	3223	30,75
6,78	3263	32,37	6,22	3216	30,47	4,95	3220	30,63
6,91	3262	32,33	6,26	3213	30,34	4,95	3218	30,55
6,91	3258	32,17	6,26	3209	30,18	5,04	3217	30,51
7,37	3254	32,01	6,39	3208	30,14	5,04	3213	30,34
7,41	3251	31,89	6,56	3207	30,10	5,12	3212	30,30
7,45	3250	31,85	6,64	3201	29,86	5,12	3210	30,22
7,70	3249	31,81	6,77	3199	29,78	5,12	3206	30,06
7,70	3248	31,76	6,77	3198	29,74	5,17	3203	29,94
7,70	3244	31,60	6,86	3197	29,70	5,21	3202	29,90
7,79	3242	31,52	6,90	3196	29,66	5,25	3201	29,86
7,87	3241	31,48	6,90	3195	29,61	5,25	3200	29,82
7,91	3239	31,40	6,98	3194	29,57	5,25	3199	29,78
8,12	3237	31,32	7,07	3193	29,53	5,29	3197	29,70
8,17	3232	31,12	7,07	3191	29,45	5,34	3193	29,53
8,63	3231	31,08	7,15	3189	29,37	5,38	3191	29,45
8,67	3230	31,03	7,15	3182	29,09	5,46	3186	29,25
8,71	3227	30,91	7,24	3180	29,01	5,51	3185	29,21
8,76	3225	30,83	7,40	3177	28,88	5,55	3183	29,13
8,76	3224	30,79	7,49	3176	28,84	5,55	3182	29,09
8,80	3223	30,75	7,49	3174	28,76	5,59	3181	29,05
8,80	3221	30,67	7,53	3171	28,64	5,64	3180	29,01
8,89	3220	30,63	7,62	3170	28,60	5,64	3179	28,97
9,05	3214	30,39	7,62	3165	28,40	5,64	3178	28,92
9,10	3202	29,90	7,70	3164	28,36	5,72	3176	28,84
9,44	3197	29,70	7,78	3159	28,15	5,76	3174	28,76
9,44	3195	29,61	7,83	3156	28,03	5,77	3170	28,60
9,52	3194	29,57	7,87	3155	27,99	5,89	3169	28,56

9,56	3188	29,33	7,87	3150	27,79	5,98	3168	28,52
9,78	3187	29,29	7,95	3145	27,59	5,98	3165	28,40
9,95	3184	29,17	8,04	3144	27,55	6,02	3161	28,24
9,99	3182	29,09	8,08	3140	27,38	6,10	3157	28,07
10,24	3179	28,97	8,21	3138	27,30	6,19	3153	27,91
10,50	3178	28,92	8,29	3134	27,14	6,32	3150	27,79
10,62	3166	28,44	8,42	3132	27,06	6,36	3149	27,75
10,83	3165	28,40	8,46	3130	26,98	6,45	3148	27,71
10,96	3163	28,32	8,59	3126	26,82	6,45	3147	27,67
11,00	3162	28,28	8,63	3124	26,73	6,53	3146	27,63
11,13	3161	28,24	8,75	3122	26,65	6,57	3145	27,59
11,21	3158	28,11	8,80	3120	26,57	6,62	3143	27,51
11,34	3156	28,03	8,84	3117	26,45	6,66	3141	27,42
11,42	3152	27,87	8,88	3115	26,37	6,70	3138	27,30
11,59	3150	27,79	8,92	3113	26,29	6,70	3137	27,26
11,59	3148	27,71	9,05	3111	26,21	6,79	3136	27,22
11,64	3146	27,63	9,09	3108	26,09	6,79	3135	27,18
11,76	3144	27,55	9,09	3106	26,00	6,83	3130	26,98
11,80	3142	27,46	9,18	3104	25,92	6,87	3128	26,90
11,85	3141	27,42	9,30	3102	25,84	6,96	3127	26,86
11,85	3140	27,38	9,35	3100	25,76	7,00	3126	26,82
11,97	3138	27,30	9,43	3098	25,68	7,00	3124	26,73
12,01	3134	27,14	9,47	3096	25,60	7,04	3123	26,69
12,01	3132	27,06	9,51	3094	25,52	7,04	3121	26,61
12,14	3127	26,86	9,56	3093	25,48	7,04	3118	26,49
12,27	3123	26,69	9,56	3091	25,40	7,09	3116	26,41
12,40	3122	26,65	9,64	3089	25,31	7,09	3114	26,33
12,52	3120	26,57	9,73	3087	25,23	7,13	3111	26,21
12,65	3119	26,53	9,73	3081	24,99	7,17	3110	26,17
12,78	3116	26,41	9,77	3076	24,79	7,22	3109	26,13
12,86	3115	26,37	9,94	3075	24,75	7,26	3107	26,04
12,90	3111	26,21	9,98	3074	24,71	7,30	3105	25,96
13,11	3106	26,00	10,02	3072	24,62	7,39	3104	25,92
13,37	3101	25,80	10,07	3066	24,38	7,39	3101	25,80
13,58	3100	25,76	10,11	3065	24,34	7,56	3100	25,76
13,62	3099	25,72	10,20	3064	24,30	7,60	3097	25,64
13,66	3097	25,64	10,24	3063	24,26	7,64	3096	25,60
13,79	3096	25,60	10,28	3061	24,18	7,73	3094	25,52
13,83	3095	25,56	10,36	3057	24,02	7,73	3093	25,48
13,95	3090	25,35	10,70	3047	23,61	7,81	3090	25,35
14,08	3087	25,23	10,87	3046	23,57	7,90	3087	25,23
14,21	3085	25,15	11,03	3044	23,49	7,90	3086	25,19
14,25	3084	25,11	11,16	3043	23,45	7,98	3084	25,11
14,25	3083	25,07	11,20	3041	23,37	8,07	3081	24,99
14,29	3081	24,99	11,33	3040	23,33	8,20	3080	24,95
14,46	3077	24,83	11,33	3039	23,29	8,28	3079	24,91

14,75	3074	24,71	11,33	3036	23,16	8,32	3077	24,83
14,92	3073	24,67	11,50	3032	23,00	8,37	3074	24,71
15,01	3071	24,58	11,58	3030	22,92	8,45	3072	24,62
15,09	3070	24,54	11,67	3028	22,84	8,45	3070	24,54
15,21	3067	24,42	11,72	3024	22,68	8,62	3069	24,50
15,21	3066	24,38	11,84	3021	22,56	8,71	3060	24,14
15,34	3063	24,26	11,92	3017	22,39	8,88	3059	24,10
15,47	3061	24,18	12,05	3015	22,31	8,88	3058	24,06
15,64	3058	24,06	12,18	3014	22,27	8,92	3057	24,02
15,76	3056	23,98	12,26	3013	22,23	8,96	3056	23,98
15,76	3053	23,85	12,27	3011	22,15	8,96	3055	23,94
15,80	3051	23,77	12,31	3010	22,11	9,00	3052	23,81
16,01	3050	23,73	12,56	3009	22,07	9,00	3051	23,77
16,06	3049	23,69	12,60	3008	22,03	9,05	3047	23,61
16,14	3048	23,65	12,73	3005	21,91	9,13	3046	23,57
16,78	3047	23,61	12,77	3004	21,87	9,17	3045	23,53
16,78	3037	23,20	12,81	3003	21,83	9,17	3041	23,37
17,11	3036	23,16	12,85	3002	21,78	9,22	3039	23,29
17,28	3035	23,12	12,89	3001	21,74	9,26	3037	23,20
17,28	3026	22,76	12,94	2999	21,66	9,30	3035	23,12
17,45	3025	22,72	12,98	2997	21,58	9,34	3033	23,04
17,49	3023	22,64	13,02	2996	21,54	9,51	3029	22,88
17,61	3022	22,60	13,06	2995	21,50	9,56	3027	22,80
17,61	3019	22,47	13,11	2994	21,46	9,60	3023	22,64
17,66	3018	22,43	13,15	2993	21,42	9,81	3021	22,56
17,74	3017	22,39	13,19	2992	21,38	9,85	3020	22,52
17,74	3016	22,35	13,27	2991	21,34	9,90	3015	22,31
17,82	3015	22,31	13,31	2990	21,30	9,90	3011	22,15
17,91	3014	22,27	13,31	2989	21,26	10,03	3007	21,99
18,16	3013	22,23	13,36	2987	21,18	10,07	3004	21,87
18,21	3011	22,15	13,36	2984	21,05	10,07	3002	21,78
18,33	3010	22,11	13,44	2983	21,01	10,19	2997	21,58
18,33	3009	22,07	13,49	2982	20,97	10,28	2995	21,50
18,54	3006	21,95	13,53	2981	20,93	10,37	2994	21,46
18,75	3005	21,91	13,57	2980	20,89	10,37	2991	21,34
18,79	2998	21,62	13,61	2979	20,85	10,45	2990	21,30
19,51	2991	21,34	13,65	2978	20,81	10,58	2987	21,18
19,55	2985	21,10	13,74	2977	20,77	10,62	2983	21,01
20,51	2982	20,97	13,86	2976	20,73	10,66	2982	20,97
20,72	2980	20,89	13,86	2975	20,69	10,70	2980	20,89
20,97	2979	20,85	13,87	2974	20,65	10,79	2978	20,81
21,14	2978	20,81	13,91	2972	20,57	10,87	2974	20,65
21,27	2977	20,77	13,95	2971	20,53	10,96	2973	20,61
21,77	2974	20,65	14,03	2970	20,49	10,96	2972	20,57
21,81	2967	20,37	14,03	2969	20,45	11,00	2971	20,53
22,53	2966	20,32	14,12	2968	20,41	11,04	2970	20,49

22,53	2962	20,16	14,16	2967	20,37	11,13	2968	20,41
22,78	2961	20,12	14,25	2960	20,08	11,25	2967	20,37
22,94	2960	20,08	14,63	2957	19,96	11,38	2966	20,32
22,94	2959	20,04	14,80	2954	19,84	11,38	2964	20,24
22,94	2958	20,00	14,92	2951	19,72	11,47	2959	20,04
22,99	2956	19,92	15,05	2949	19,63	11,64	2958	20,00
23,11	2955	19,88	15,13	2948	19,59	11,72	2956	19,92
23,28	2951	19,72	15,18	2946	19,51	11,72	2955	19,88
23,32	2947	19,55	15,30	2945	19,47	11,81	2954	19,84
23,49	2945	19,47	15,34	2944	19,43	11,85	2953	19,80
23,92	2937	19,15	15,34	2941	19,31	11,89	2950	19,68
24,04	2936	19,11	15,51	2938	19,19	12,02	2949	19,63
25,77	2933	18,99	15,68	2933	18,99	12,10	2948	19,59
26,10	2932	18,95	16,06	2930	18,86	12,14	2947	19,55
26,27	2931	18,90	16,31	2927	18,74	12,19	2943	19,39
26,61	2930	18,86	16,52	2924	18,62	12,32	2942	19,35
26,73	2929	18,82	16,61	2921	18,50	12,40	2937	19,15
26,82	2928	18,78	17,03	2918	18,38	12,48	2936	19,11
26,86	2921	18,50	17,16	2913	18,17	12,61	2935	19,07
27,74	2913	18,17	17,70	2912	18,13	12,66	2934	19,03
27,95	2911	18,09	17,79	2907	17,93	12,66	2933	18,99
28,25	2910	18,05	18,68	2902	17,73	12,78	2931	18,90
28,71	2908	17,97	18,85	2900	17,65	13,00	2930	18,86
29,13	2907	17,93	18,89	2899	17,61	13,00	2929	18,82
29,34	2906	17,89	18,97	2897	17,53	13,04	2927	18,74
29,51	2905	17,85	19,02	2895	17,44	13,08	2926	18,70
29,51	2903	17,77	19,10	2894	17,40	13,13	2925	18,66
29,59	2902	17,73	19,14	2893	17,36	13,17	2924	18,62
29,60	2898	17,57	19,18	2892	17,32	13,25	2923	18,58
30,10	2896	17,48	19,23	2890	17,24	13,29	2921	18,50
30,19	2894	17,40	19,35	2888	17,16	13,38	2920	18,46
30,95	2893	17,36	19,39	2885	17,04	13,38	2919	18,42
31,12	2892	17,32	19,56	2883	16,96	13,42	2918	18,38
31,75	2885	17,04	19,91	2882	16,92	13,46	2916	18,30
34,29	2884	17,00	19,95	2881	16,88	13,50	2915	18,26
35,05	2882	16,92	20,16	2877	16,71	13,50	2914	18,22
35,30	2873	16,55	20,84	2875	16,63	13,55	2913	18,17
36,73	2871	16,47	20,92	2874	16,59	13,59	2910	18,05
36,98	2870	16,43	21,01	2873	16,55	13,59	2909	18,01
37,15	2867	16,31	21,13	2872	16,51	13,59	2906	17,89
37,23	2866	16,27	21,39	2868	16,35	13,72	2905	17,85
38,46	2863	16,15	21,47	2862	16,11	13,72	2901	17,69
39,55	2861	16,06	21,77	2860	16,02	13,89	2899	17,61
39,97	2860	16,02	21,81	2859	15,98	13,93	2898	17,57
40,18	2858	15,94	22,28	2855	15,82	13,93	2897	17,53
40,52	2857	15,90	22,57	2854	15,78	14,02	2896	17,48

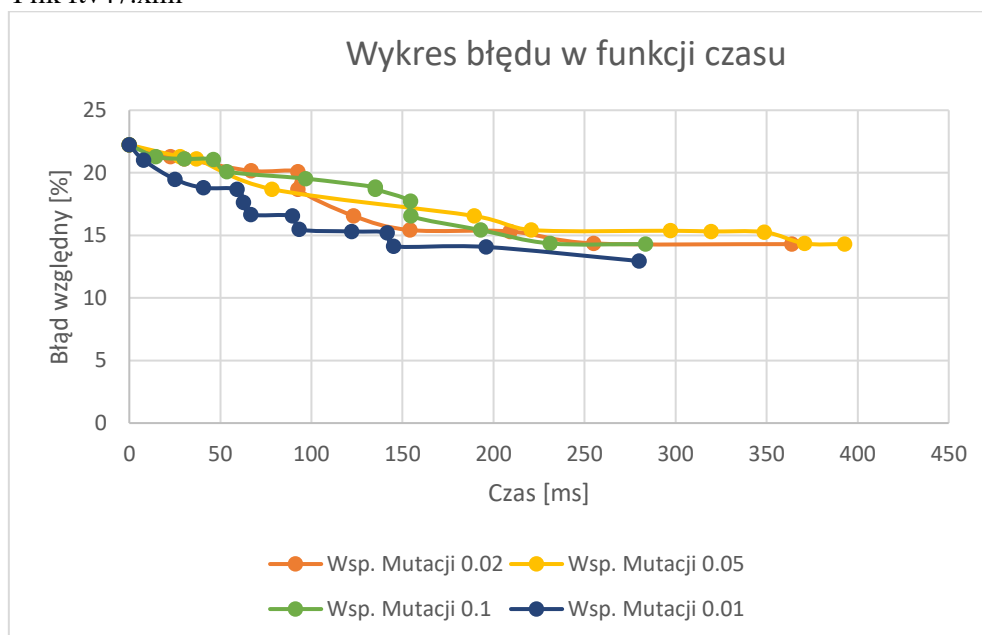
40,56	2856	15,86	22,74	2853	15,74	14,02	2894	17,40
40,85	2855	15,82	22,74	2850	15,62	14,06	2892	17,32
41,15	2852	15,70	22,92	2849	15,58	14,10	2890	17,24
41,24	2851	15,66	23,09	2845	15,42	14,10	2889	17,20
41,24	2849	15,58	23,39	2835	15,01	14,23	2888	17,16
41,40	2842	15,29	24,86	2830	14,81	14,23	2886	17,08
41,66	2841	15,25	25,32	2829	14,77	14,23	2885	17,04
41,70	2839	15,17	25,53	2825	14,60	14,28	2884	17,00
41,70	2836	15,05	25,70	2824	14,56	14,40	2881	16,88
41,79	2835	15,01	25,91	2823	14,52	14,49	2879	16,80
41,83	2832	14,89	26,00	2822	14,48	14,53	2878	16,75
41,95	2831	14,85	26,08	2818	14,32	14,70	2876	16,67
42,17	2828	14,73	26,20	2817	14,28	14,87	2874	16,59
43,81	2819	14,36	26,37	2816	14,24	14,95	2872	16,51
44,36	2818	14,32	26,37	2815	14,20	14,95	2869	16,39
45,36	2817	14,28	26,50	2812	14,08	15,17	2866	16,27
45,58	2816	14,24	26,62	2810	14,00	15,30	2865	16,23
45,79	2810	14,00	26,75	2809	13,96	15,47	2863	16,15
45,83	2808	13,91	26,84	2808	13,91	15,51	2857	15,90
45,92	2807	13,87	26,92	2806	13,83	15,68	2856	15,86
46,34	2801	13,63	27,34	2804	13,75	15,76	2855	15,82
46,59	2800	13,59	27,59	2798	13,51	15,76	2854	15,78
46,76	2799	13,55	28,02	2796	13,43	15,93	2853	15,74
46,76	2794	13,35	28,27	2794	13,35	15,97	2847	15,50
47,01	2792	13,27	28,44	2792	13,27	16,02	2845	15,42
47,18	2791	13,23	28,56	2790	13,18	16,06	2844	15,38
47,43	2789	13,14	29,87	2782	12,86	16,10	2842	15,29
47,56	2788	13,10	30,25	2781	12,82	16,27	2841	15,25
48,52	2787	13,06	30,42	2778	12,70	16,40	2834	14,97
49,99	2785	12,98	30,63	2777	12,66	16,44	2833	14,93
50,29	2784	12,94	30,80	2776	12,62	16,69	2832	14,89
50,88	2783	12,90	30,84	2775	12,58	16,70	2831	14,85
51,01	2778	12,70	30,97	2774	12,54	16,74	2829	14,77
54,97	2777	12,66	31,05	2769	12,33	16,82	2827	14,69
55,26	2774	12,54	31,18	2768	12,29	16,86	2825	14,60
57,00	2773	12,49	31,22	2766	12,21	16,91	2824	14,56
57,92	2772	12,45	31,23	2765	12,17	17,04	2823	14,52
59,68	2769	12,33	31,35	2763	12,09	17,04	2822	14,48
60,18	2765	12,17	31,39	2762	12,05	17,12	2821	14,44
63,25	2761	12,01	31,43	2761	12,01	17,17	2819	14,36
63,67	2759	11,93	31,47	2758	11,89	17,25	2813	14,12
65,10	2750	11,56	31,52	2757	11,85	17,29	2811	14,04
68,74	2745	11,36	31,56	2754	11,72	17,33	2810	14,00
72,16	2743	11,28	31,56	2751	11,60	17,50	2809	13,96
72,83	2742	11,24	31,60	2750	11,56	17,50	2808	13,91
73,34	2735	10,95	31,91	2749	11,52	17,59	2807	13,87

77,17	2731	10,79	32,12	2747	11,44	17,59	2806	13,83
77,30	2729	10,71	32,67	2746	11,40	17,76	2804	13,75
77,73	2722	10,43	32,83	2745	11,36	17,84	2801	13,63
78,40	2717	10,22	32,88	2740	11,16	18,05	2800	13,59
78,40	2715	10,14	33,04	2736	10,99	18,10	2799	13,55
79,16	2710	9,94	33,38	2735	10,95	18,14	2797	13,47
80,13	2709	9,90	34,64	2726	10,59	18,18	2795	13,39
80,38	2708	9,86	35,66	2725	10,55	18,35	2793	13,31
80,80	2707	9,82	35,66	2716	10,18	18,48	2792	13,27
81,43	2703	9,66	37,02	2714	10,10	18,56	2790	13,18
81,56	2699	9,49	37,56	2713	10,06	18,73	2788	13,10
81,98	2698	9,45	39,76	2712	10,02	18,77	2785	12,98
82,53	2695	9,33	40,66	2711	9,98	18,94	2784	12,94
			41,37	2710	9,94	18,94	2783	12,90
			41,46	2708	9,86	19,07	2782	12,86
			41,75	2707	9,82	19,20	2781	12,82
			42,18	2703	9,66	19,46	2774	12,54
			43,66	2702	9,61	20,17	2773	12,49
			43,75	2699	9,49	20,57	2772	12,45
			44,25	2693	9,25	20,66	2771	12,41
			44,42	2692	9,21	20,82	2768	12,29
			44,67	2689	9,09	20,87	2763	12,09
			44,76	2688	9,05	20,99	2762	12,05
			45,52	2679	8,68	21,08	2761	12,01
			45,95	2678	8,64	21,12	2757	11,85
			45,99	2669	8,28	21,29	2756	11,81
			46,66	2668	8,24	21,50	2753	11,68
			47,42	2665	8,11	21,50	2752	11,64
			47,72	2664	8,07	21,67	2747	11,44
			47,89	2663	8,03	22,06	2746	11,40
			48,15	2654	7,67	22,14	2744	11,32
			48,36	2652	7,59	22,18	2739	11,12
			49,63	2647	7,38	22,56	2738	11,08
			50,01	2645	7,30	22,95	2735	10,95
			50,43	2644	7,26	23,12	2734	10,91
			52,08	2643	7,22	23,20	2733	10,87
			52,25	2641	7,14	23,37	2732	10,83
			53,65	2638	7,02	23,50	2731	10,79
			55,17	2636	6,94	24,31	2724	10,51
			55,51	2635	6,90	24,69	2723	10,47
			56,60	2633	6,82	24,86	2720	10,34
			59,35	2632	6,77	24,94	2718	10,26
			59,52	2631	6,73	25,15	2717	10,22
			59,73	2629	6,65	25,41	2715	10,14
			60,02	2627	6,57	25,45	2714	10,10
			60,16	2625	6,49	25,71	2711	9,98

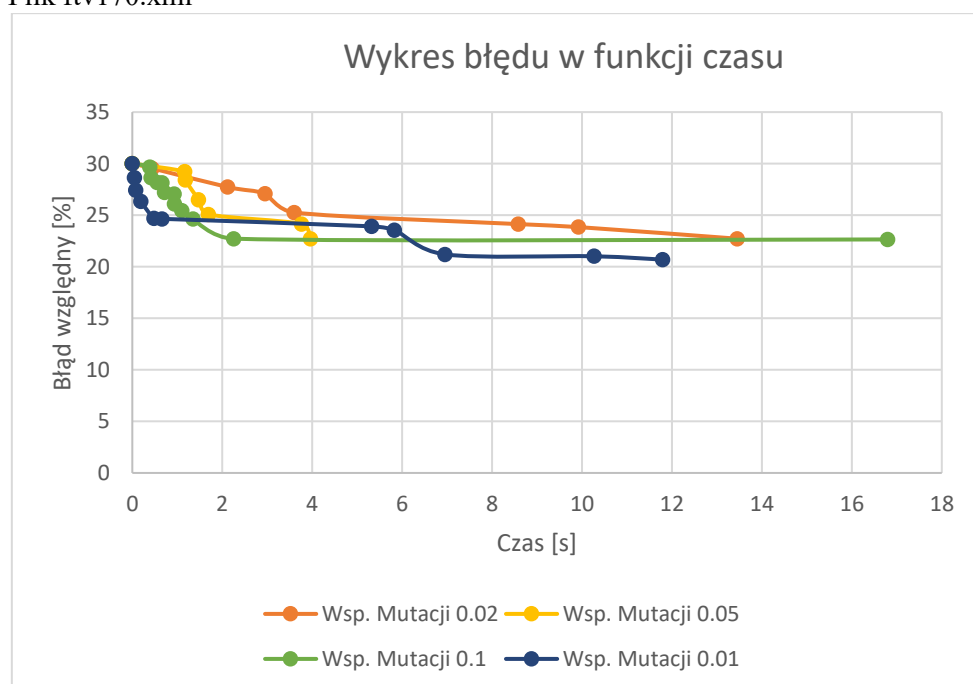
			60,75	2621	6,33	25,75	2707	9,82
			68,35	2620	6,29	26,43	2705	9,74
			70,63	2616	6,13	26,86	2703	9,66
			71,94	2612	5,96	26,86	2696	9,37
			73,37	2606	5,72	28,90	2695	9,33
			73,83	2603	5,60	28,98	2693	9,25
			77,45	2600	5,48	29,28	2692	9,21
			78,12	2598	5,40	29,62	2690	9,13
			79,76	2596	5,31	29,87	2683	8,84
			80,69	2595	5,27	32,90	2682	8,80
			81,87	2594	5,23	33,02	2680	8,72
			84,83	2593	5,19	33,11	2674	8,48
						33,32	2672	8,40
						33,62	2662	7,99
						33,66	2660	7,91
						34,34	2652	7,59
						34,81	2648	7,42
						35,06	2645	7,30
						36,20	2642	7,18
						36,62	2640	7,10
						37,09	2638	7,02
						38,78	2637	6,98
						38,99	2631	6,73
						40,68	2629	6,65
						40,90	2628	6,61
						41,15	2626	6,53
						41,70	2625	6,49
						41,96	2624	6,45
						42,35	2623	6,41
						45,18	2621	6,33
						48,88	2618	6,21
						50,49	2616	6,13
						52,01	2610	5,88
						56,80	2604	5,64
						57,27	2602	5,56
						59,86	2594	5,23
						60,66	2592	5,15
						61,43	2590	5,07
						61,56	2587	4,95
						62,03	2585	4,87
						63,81	2580	4,67
						66,33	2578	4,58
						68,81	2569	4,22

3.5.3 Wykresy porównawcze dla testowanych współczynników mutacji

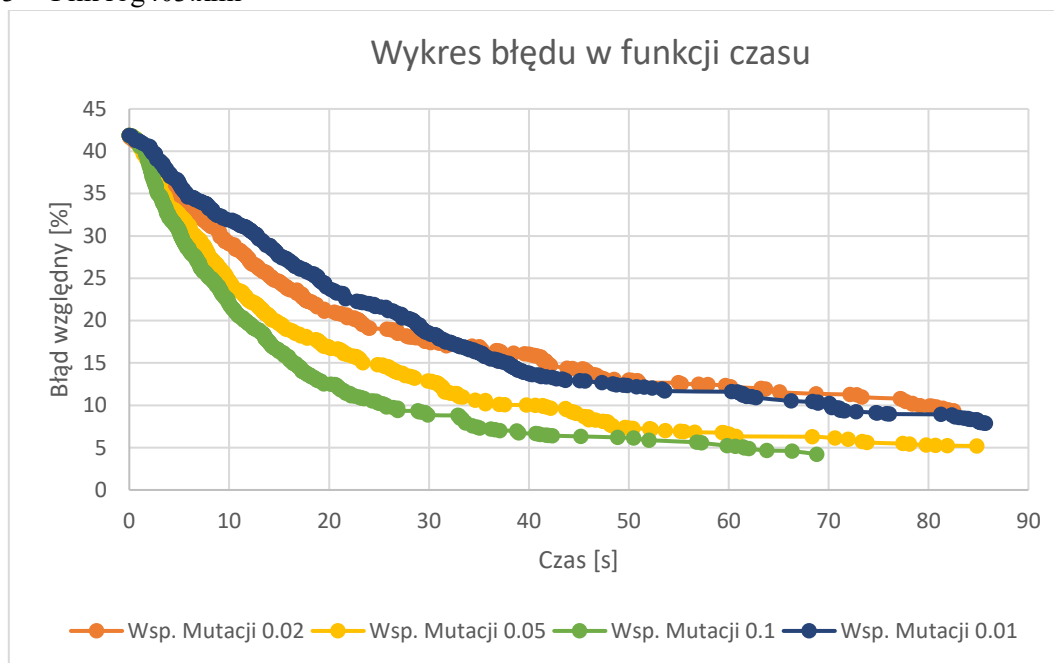
3.5.3.1 Plik ftv47.xml



3.5.3.2 Plik ftv170.xml



3.5.3.3 Plik rbg403.xml



3.5.4 Omówienie wyników

Nie porównywałem ponownie wyników z algorytmem Tabu, ponieważ najlepszy wynik dla pliku ftv170.xml został taki sam jaki już był porównywany. Tzn. zmiana współczynnika mutacji nie wpłynęła pozytywnie na wyniki dla tego pliku. Tak samo zresztą było dla pliku ftv47.xml, jedyne zmiany odnotowałem dla pliku rbg403.xml. Zmiany te były znaczące, rzędu 3,5%, dla większych współczynników mutacji. Może to być związane z tym jak algorytm genetyczny działa z poszczególnymi plikami. Jak można było zauważyć w omówionych testach właśnie dla pliku rbg403.xml odbywa się najwięcej zamian najlepszego wyniku.

4 Podsumowanie projektu

Podczas projektu został zbadany algorytm genetyczny dla problemu komiwojażera. Algorytm ten będąc algorytmem metaheurystycznym, nie generuje dokładnych rozwiązań, jednakże dobrze radzi sobie z dużymi problemami. Jak już wcześniej pisałem, powinien on prawdopodobnie generować delikatnie gorsze wyniki niż Tabu Search, ale z racji tego że moje algorytmy z poprzedniego projektu nie były wybitne, okazał się on znacząco lepszy. Badania przeprowadzone podczas tego, ale również poprzednich projektów, pokazały jak bardzo ważnymi algorytmami są algorytmy optymalizacyjne. Gdyby nie one nie byłibyśmy w stanie nawet oszacować zbliżonych wyników dla tak trudnych problemów. Dobrym argumentem jest to, iż algorytmy dokładne takie jak dynamiczne programowanie, Brute Force, czy Branch & Bound działały w akceptowalnym czasie jedynie do problemów rzędu do 30 wierzchołków, niektóre nawet mniej. Zatem czym w obliczu 403 wierzchołków jest błąd rzędu 4,2%, to jest znikomy błąd, a jednak problem był ogromny.

Podczas implementowania algorytmu, myślałem o metodzie selekcji typu ruletka oraz losowym generowaniu populacji. Jednak po zaimplementowaniu tych metod błąd względny wynosił około ponad 700%, po zmianie ruletki na selekcję turniejową błąd ten zredukował się do około 400%. Jednak to również był ogromny błąd, dlatego zmieniłem jeszcze generowanie populacji na zachłanne, dzięki czemu osiągnąłem wyniki prezentowane w powyższych badaniach.

5 Bibliografia

- <https://www.aragorn.wi.pb.edu.pl/~wkwedlo/EA5.pdf>
- [http://www.imio.polsl.pl/Dopobrania/Cw%20MH%2007%20\(TSP\).pdf](http://www.imio.polsl.pl/Dopobrania/Cw%20MH%2007%20(TSP).pdf)
- <https://www.geeksforgeeks.org/genetic-algorithms/>
- https://en.wikipedia.org/wiki/Genetic_algorithm
- <https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>
- D.A. Goldberg „Algorytmy genetyczne i ich zastosowania”
- Z. Michalewicz „Algorytmy genetyczne + struktury danych = programy ewolucyjne”
- Wykłady – dr inż. Tomasz Kapłon