

Zadanie projektowe 3

Sprawozdanie

Kurs: Projektowanie efektywnych algorytmów

Prowadzący: mgr inż. Antoni Sterna

Grupa: E01-60h (środa 13:15-15:00)

Autor: Mirosław Kuźniar (nr indeksu: 248870)

Spis treści

| | | |
|-------|--|----|
| 1 | Wstęp teoretyczny | 3 |
| 1.1 | Problem optymalizacyjny..... | 3 |
| 1.2 | Metody ewolucyjne | 4 |
| 1.3 | Algorytm genetyczny | 4 |
| 1.3.1 | Operatory genetyczne | 5 |
| 2 | Najważniejsze aspekty implementacji..... | 6 |
| 2.1 | Generowanie początkowej populacji..... | 6 |
| 2.2 | Metoda sukcesji | 6 |
| 2.3 | Lokalna optymalizacja | 6 |
| 3 | Implementacja algorytmów | 6 |
| 4 | Plan eksperymentu | 6 |
| 4.1 | Rozmiar problemu | 6 |
| 4.2 | Dane testowe | 6 |
| 5 | Wyniki pomiarów..... | 7 |
| 5.1 | Wpływ wielkości populacji oraz obu metod krzyżowania | 7 |
| 5.2 | Wpływ współczynnika krzyżowania..... | 8 |
| 5.3 | Porównanie Tabu Search i Algorytmu Genetycznego | 9 |
| 6 | Omówienie wyników | 10 |
| 7 | Wnioski..... | 10 |

1 Wstęp teoretyczny

1.1 Problem optymalizacyjny

Rozważanym problemem optymalizacyjnym jest problem komiwojażera (ang. Travelling Salesman Problem). Zgodnie z definicją polega on na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. W praktyce często opisuje się go jako problem komiwojażera/wędrownego sprzedawcy który wyruszając z miasta początkowego musi odwiedzić każde z pozostałych miast tylko raz i wrócić do miasta początkowego. Ponadto trasa, którą przebędzie musi być minimalna pod względem przyjętego kosztu (np. odległości, czasu, kosztów ekonomicznych itp.). Warto także wspomnieć, że zgodnie z definicją grafu pełnego musi istnieć połączenie z danego miasta do wszystkich pozostałych.

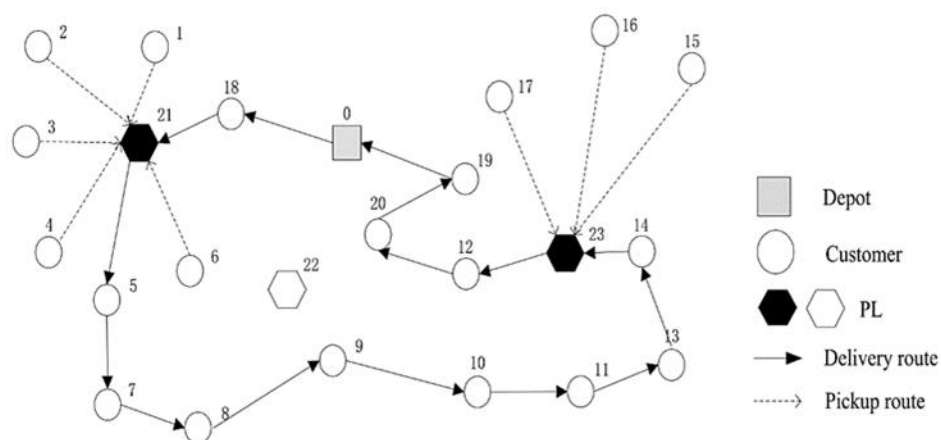
Problem komiwojażera należy do klasy problemów NP – trudnych. Problemy z tej klasy opisywane są jako takie dla których (najprawdopodobniej) nie można skonstruować algorytmów wielomianowych, czyli algorytmów efektywnych obliczeniowo.

Wyróżniamy dwa rodzaje problemu komiwojażera:

- Symetryczny problem komiwojażera (STSP), w którym waga krawędzi przyjmuje jedną wartość bez względu na kierunek poruszania się, tzn. odległość z miasta A do miasta B jest taka sama jak z miasta B do miasta A,
- Asymetryczny problem komiwojażera (ATSP), w którym waga krawędzi może przyjmować różne wartości w zależności od kierunku przemieszczania się, tzn. odległość z miasta A do miasta B może być różna niż z miasta B do miasta A.

Rozważanym dalej rodzajem będzie asymetryczny problem komiwojażera (ATSP).

Pomimo swojej złożoności problem komiwojażera nie jest tylko tematem dysput akademickich ale także znajduje zastosowanie w praktycznych aplikacjach. Dla przykładu, wydajne rozwiązania TSP są często wykorzystywane w tzw. dostawach ostatniej mili (ang. last mile delivery), gdzie towar przemieszczany jest z węzła transportowego, takiego jak skład lub magazyn do klienta. Dostawy te stanowią główne źródło różnic kosztów w całym łańcuchu dostaw. Dlatego wiele firm z sektora logistyki dąży do zminimalizowania kosztów dostawy ostatniej mili wykorzystując wydajne algorytmy bazujące na problemie komiwojażera.



1.2 Metody ewolucyjne

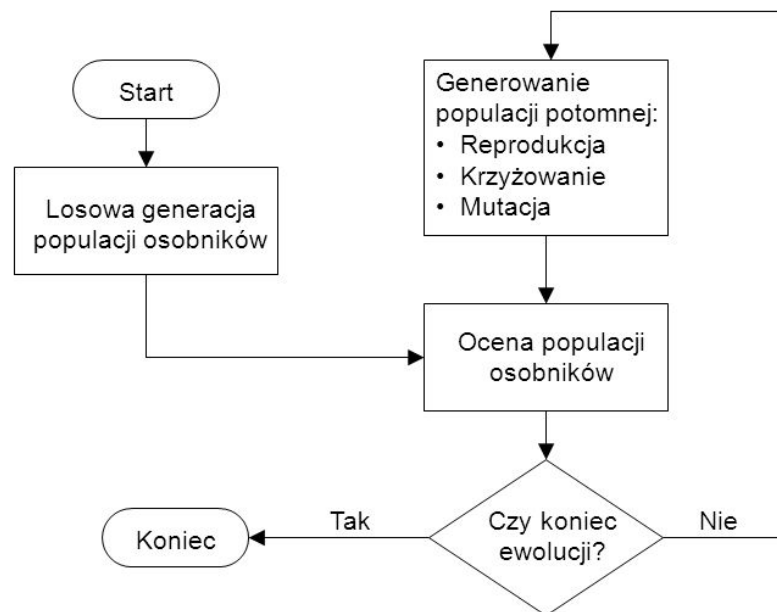
Stanowią rodzaj heurystycznego przeszukiwania przestrzeni rozwiązań naśladujący naturalne procesy ewolucji. Poszukiwanie polega, na tworzeniu rozwiązania opartego na współpracy osobników wykorzystujących wiedzę o przestrzeni rozwiązań oraz na przekazywaniu tej wiedzy kolejnym pokoleniom z wykorzystaniem mechanizmów ewolucyjnych. Metody ewolucyjne umożliwiają:

- szybsze przeszukiwanie przestrzeni rozwiązań
- przeszukiwanie większej przestrzeni rozwiązań
- unikanie pułapki ekstremum lokalnego

1.3 Algorytm genetyczny

Algorytm genetyczny opiera się na ewolucji darwinowskiej. Środowisko ma ograniczone zasoby. Przeżywają i rozmnażają się osobniki najlepiej przystosowane. W procesie rozmnażania powstają potomkowie na ogół lepiej przystosowani od rodziców. Czasami u potomstwa mogą występować losowe zmiany zmieniające ich przystosowanie. Wynikiem tego jest coraz lepiej przystosowana populacja do warunków środowiska.

Ogólny schemat algorytmu genetycznego prezentuje się następująco:



wybór populacji początkowej chromosomów (losowy)

ocena przystosowania chromosomów

sprawdzanie warunku zatrzymania

selekcja chromosomów - wybór populacji macierzystej

krzyżowanie chromosomów z populacji rodzicielskiej

mutacja - może być również wykonana przed krzyżowaniem

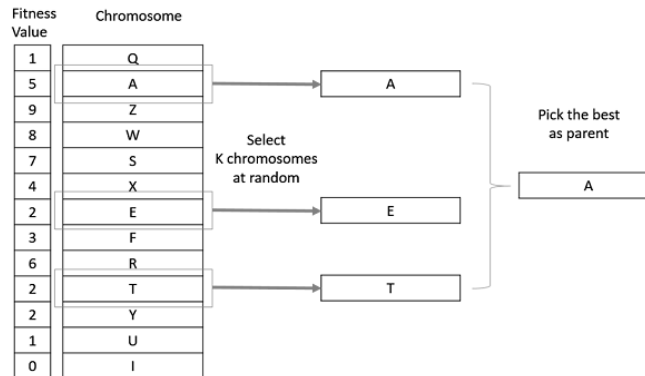
ocena przystosowania chromosomów

utworzenie nowej populacji wg. modelu sukcesji

wyprowadzenie najlepszego rozwiązania

1.3.1 Operatory genetyczne

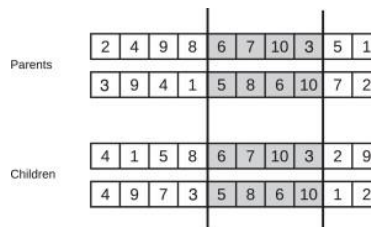
- Selekcja - polega na wyborze z bieżącej populacji osobników, których materiał genetyczny zostanie poddany operacji krzyżowania oraz mutacji i przekazany osobnikom potomnym. Metody selekcji:
 - Ruletkowa
 - Rankingowa
 - **Turniejowa** – populację dzieli się na szereg dowolnie licznych grup. Następnie z każdej z nich wybiera się osobnika o najlepszym przystosowaniu



- Krzyżowanie - polega na wymianie materiału genetycznego pomiędzy losowo wybranymi parami osobników. W wyniku krzyżowania powstają nowe osobniki, które mogą wejść w skład nowej populacji. Krzyżowanie zachodzi z prawdopodobieństwem $p_c \in (0.6, 1.0)$. Metody krzyżowania:

- **OX - Ordered Crossover**

W krzyżowaniu OX potomków tworzy się na podstawie pod tras pobranych z rodziców (sekcji dopasowania) oraz ich uzupełnienia tak żeby nie powstał konflikt



- PMX - Partially Mapped Crossover
- EX - Edge Crossover
- SXX - Subtour Exchanged Crossover
- PX - Partition Crossover
- Mutacja - polega na zamianie wartości losowo wybranego genu. Celem użycia operatora mutacji jest zapewnienie zmienności chromosomów. Zachodzi z prawdopodobieństwem $p_m \in (0.01, 0.1)$. Metody mutacji:

- **Insertion** - przestawia losowo wybrane miasto na inną pozycję

$$\pi = \langle \pi(1), \dots, \pi(i-1), \pi(i), \pi(i+1), \dots, \pi(j-1), \pi(j), \pi(j+1), \dots, \pi(n) \rangle$$

$$\pi' = \langle \pi(1), \dots, \pi(i-1), \pi(j), \pi(i), \pi(i+1), \dots, \pi(j-1), \pi(j+1), \dots, \pi(n) \rangle$$

- Inversion - wybiera losowo podciągi miast i zamienia ich kolejność

- **Transposition** - zamienia dwa losowo wybrane miasta

$$\pi = \langle \pi(1), \dots, \pi(i-1), \pi(i), \pi(i+1), \dots, \pi(j-1), \pi(j), \pi(j+1), \dots, \pi(n) \rangle$$

$$\pi = \langle \pi(1), \dots, \pi(i-1), \pi(j), \pi(i+1), \dots, \pi(j-1), \pi(i), \pi(j+1), \dots, \pi(n) \rangle$$

- **Displacement** - zamienia w wybranym losowo podciągu miast pierwsze z ostatnim

2 Najważniejsze aspekty implementacji

2.1 Generowanie początkowej populacji

Populacja początkowa została utworzona w sposób losowy. Dla wszystkich osobników populacji stworzono permutację od 1...n-1 a następnie przetasowano z użyciem funkcji `std::random_shuffle`.

2.2 Metoda sukcesji

Wytworzenie populacji nowego pokolenia polegało na przeniesieniu ustalonego procentu najlepiej przystosowanych osobników z poprzedniego pokolenia do nowego. Zauważono, że w przypadku gdy nowa populacja nie zawiera najlepszych osobników z poprzedniego pokolenia błąd względny uzyskiwanego rozwiązania jest znacznie większy.

Aby skutecznie wyodrębnić najlepiej przystosowanych osobników posortowano populację względem funkcji dopasowania za pomocą funkcji `std::sort`. Następnie wybrano odpowiednią liczbę osobników o najmniejszych indeksach

2.3 Lokalna optymalizacja

Aby dodatkowo polepszyć wyniki uzyskiwane za pomocą algorytmu genetycznego zaimplementowano mechanizm lokalnej optymalizacji. Dla każdego nowopowstałego osobnika (po operacji krzyżowania i mutacji) zostaje znaleziony najlepszy sąsiad i to on jest wpisywany do nowej populacji.

3 Implementacja algorytmów

Wybrany językiem programowania jest C++. Algorytmy operują na zmiennych dziesiętnych `int` lub zmiennoprzecinkowych `float`. Do przechowywania danych wykorzystano strukturę danych z biblioteki STL - `vector`.

4 Plan eksperymentu

4.1 Rozmiar problemu

W przypadku algorytmu genetycznego testy efektywności przeprowadzono dla trzech przykładów wyraźnie różniących się rozmiarem. Wykorzystano instancje o rozmiarze 17, 43 i 403 miast.

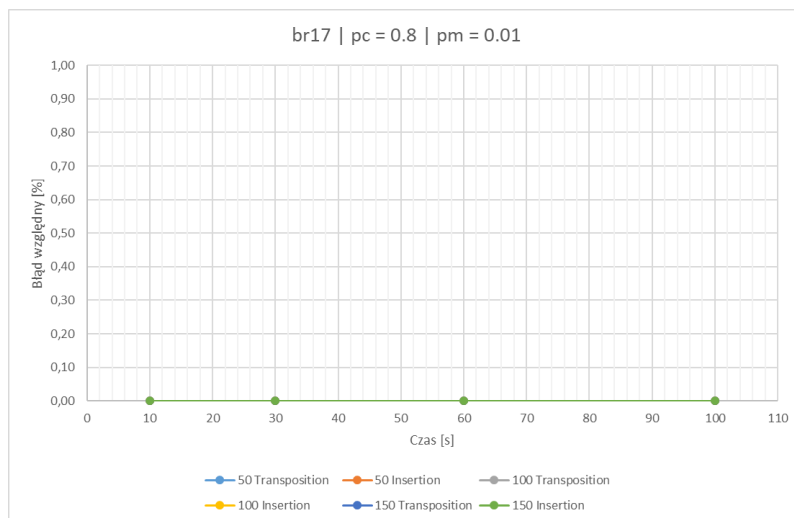
4.2 Dane testowe

Przykłady testowe zaczerpnięto ze strony http://staff.iiar.pwr.wroc.pl/antoni.sterna/pea/PEA_testy.htm. Wykorzystano pliki br17 o rozwiązaniu optymalnym 39, p43 o rozwiązaniu optymalnym 5620 oraz rbg403 o rozwiązaniu optymalnym 2465.

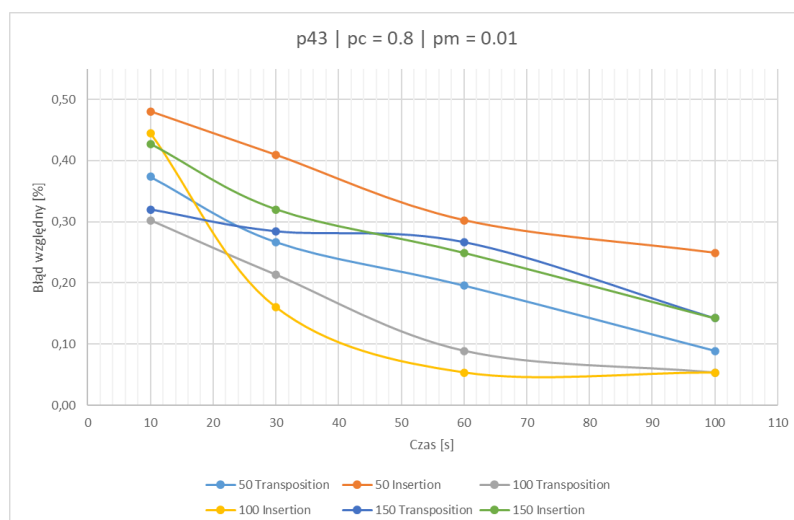
5 Wyniki pomiarów

5.1 Wpływ wielkości populacji oraz obu metod krzyżowania

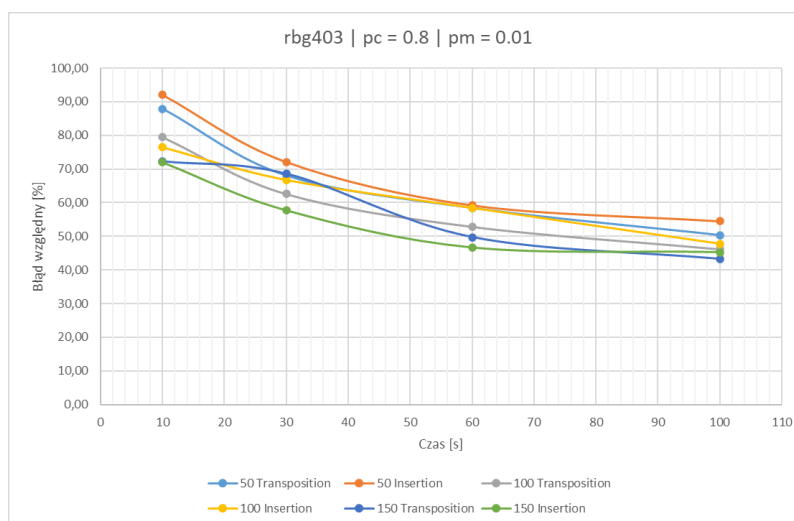
| Instancja: | | | | | |
|-------------------|----------------|----------|----------|----------|----------|
| br17 | | Czas [s] | | | |
| | | 10 | 30 | 60 | 100 |
| Rozmiar populacji | Metoda mutacji | Błąd [%] | Błąd [%] | Błąd [%] | Błąd [%] |
| 50 | Transposition | 0,00 | 0,00 | 0,00 | 0,00 |
| 50 | Insertion | 0,00 | 0,00 | 0,00 | 0,00 |
| 100 | Transposition | 0,00 | 0,00 | 0,00 | 0,00 |
| 100 | Insertion | 0,00 | 0,00 | 0,00 | 0,00 |
| 150 | Transposition | 0,00 | 0,00 | 0,00 | 0,00 |
| 150 | Insertion | 0,00 | 0,00 | 0,00 | 0,00 |



| Instancja: | | | | | |
|-------------------|----------------|----------|----------|----------|----------|
| p43 | | Czas [s] | | | |
| | | 10 | 30 | 60 | 100 |
| Rozmiar populacji | Metoda mutacji | Błąd [%] | Błąd [%] | Błąd [%] | Błąd [%] |
| 50 | Transposition | 0,37 | 0,27 | 0,20 | 0,09 |
| 50 | Insertion | 0,48 | 0,41 | 0,30 | 0,25 |
| 100 | Transposition | 0,30 | 0,21 | 0,09 | 0,05 |
| 100 | Insertion | 0,44 | 0,16 | 0,05 | 0,05 |
| 150 | Transposition | 0,32 | 0,28 | 0,27 | 0,14 |
| 150 | Insertion | 0,43 | 0,32 | 0,25 | 0,14 |



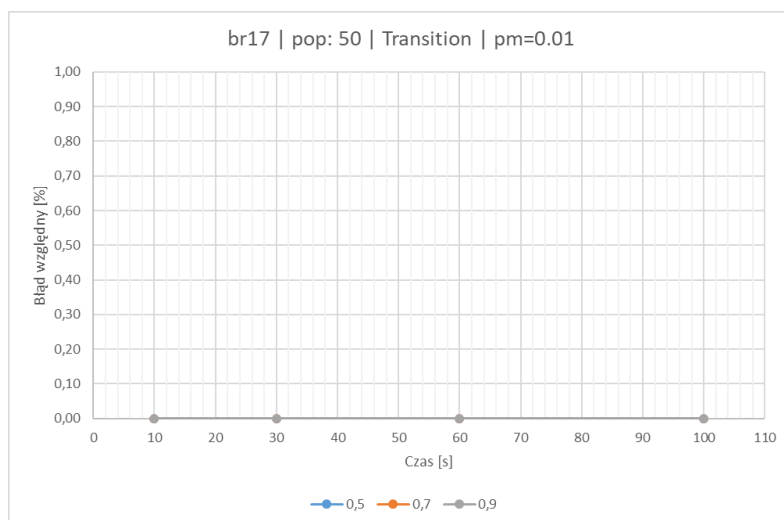
| Instancja: | | | | | |
|-------------------|----------------|----------|----------|----------|----------|
| rbg403 | | Czas [s] | | | |
| | | 10 | 30 | 60 | 100 |
| Rozmiar populacji | Metoda mutacji | Błąd [%] | Błąd [%] | Błąd [%] | Błąd [%] |
| 50 | Transposition | 87,87 | 67,99 | 58,38 | 50,30 |
| 50 | Insertion | 92,05 | 72,09 | 59,23 | 54,48 |
| 100 | Transposition | 79,51 | 62,60 | 52,90 | 46,21 |
| 100 | Insertion | 76,43 | 66,73 | 58,46 | 47,79 |
| 150 | Transposition | 72,29 | 68,60 | 49,74 | 43,25 |
| 150 | Insertion | 71,97 | 57,69 | 46,61 | 45,23 |



5.2 Wpływ współczynnika krzyżowania

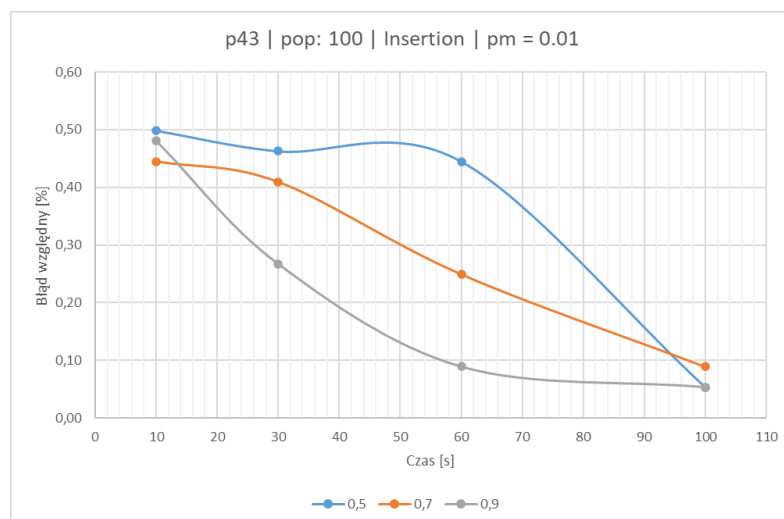
| Instancja: | Najlepsza kombinacja: | Wsółczynnik mutacji: |
|------------|-----------------------|----------------------|
| br17 | 50 Transposition | 0,01 |

| Współczynnik krzyżowania | Czas [s] | | | |
|--------------------------|----------|------|------|------|
| | 10 | 30 | 60 | 100 |
| 0,5 | 0,00 | 0,00 | 0,00 | 0,00 |
| 0,7 | 0,00 | 0,00 | 0,00 | 0,00 |
| 0,9 | 0,00 | 0,00 | 0,00 | 0,00 |



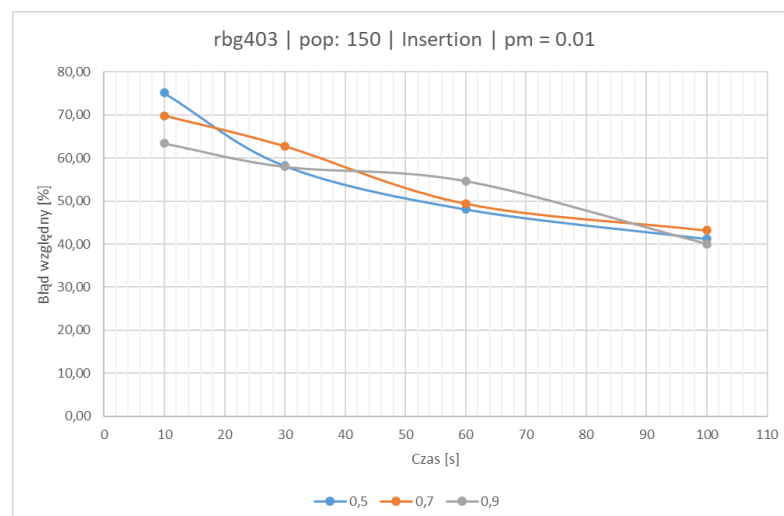
| Instancja: | Najlepsza kombinacja: | Wsółczynnik mutacji: |
|------------|-----------------------|----------------------|
| p43 | 100 Insertion | 0,01 |

| Współczynnik krzyżowania | Czas [s] | | | |
|--------------------------|----------|------|------|------|
| | 10 | 30 | 60 | 100 |
| 0,5 | 0,50 | 0,46 | 0,44 | 0,05 |
| 0,7 | 0,44 | 0,41 | 0,25 | 0,09 |
| 0,9 | 0,48 | 0,27 | 0,09 | 0,05 |



| Instancja: | Najlepsza kombinacja: | Wsółczynnik mutacji: |
|------------|-----------------------|----------------------|
| rbg403 | 150 Insertion | 0,01 |

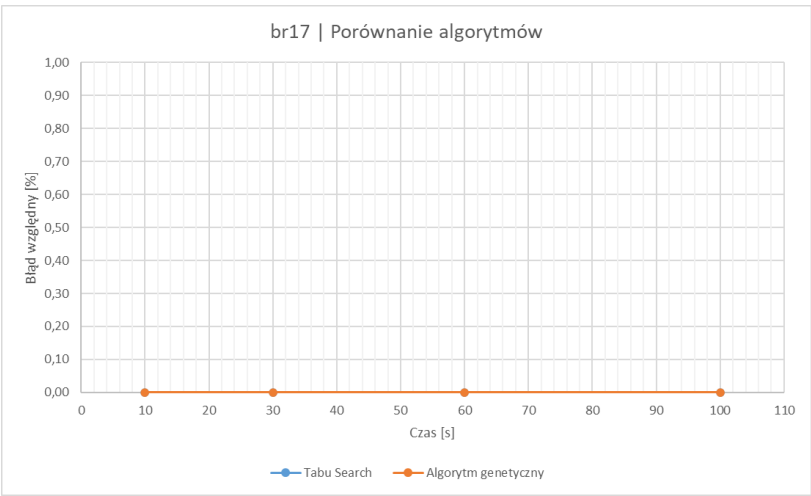
| Współczynnik krzyżowania | Czas [s] | | | |
|--------------------------|----------|-------|-------|-------|
| | 10 | 30 | 60 | 100 |
| 0,5 | 75,17 | 58,22 | 48,11 | 41,22 |
| 0,7 | 69,82 | 62,72 | 49,41 | 43,20 |
| 0,9 | 63,45 | 58,01 | 54,69 | 40,08 |



5.3 Porównanie Tabu Search i Algorytmu Genetycznego

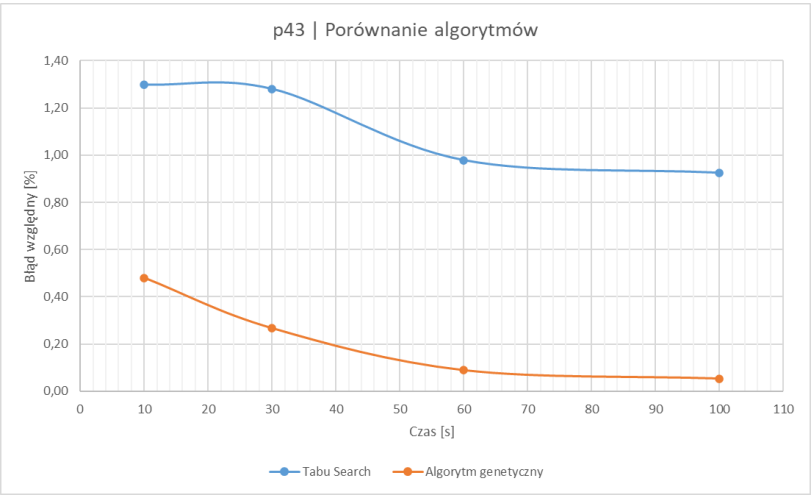
| | | | |
|------------|-----------------------|----------------------|--------------------------|
| Instancja: | Najlepsza kombinacja: | Wsółczynnik mutacji: | Współczynnik krzyżowania |
| br17 | 50 Transposition | 0,01 | 0,7 |

| Algorytm | Czas [s] | | | |
|---------------------|----------|----------|----------|----------|
| | 10 | 30 | 60 | 100 |
| Algorytm | Błąd [%] | Błąd [%] | Błąd [%] | Błąd [%] |
| Tabu Search | 0,00 | 0,00 | 0,00 | 0,00 |
| Algorytm genetyczny | 0,00 | 0,00 | 0,00 | 0,00 |



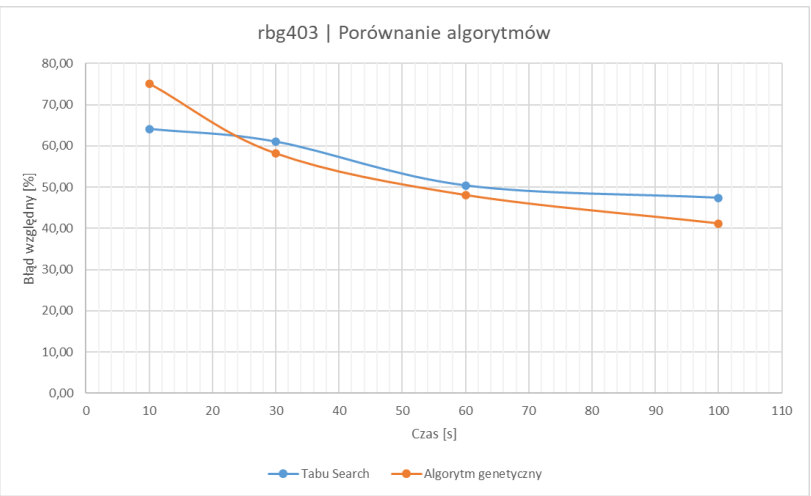
| | | | |
|------------|-----------------------|----------------------|--------------------------|
| Instancja: | Najlepsza kombinacja: | Wsółczynnik mutacji: | Współczynnik krzyżowania |
| p43 | 100 Insertion | 0,01 | 0,9 |

| Algorytm | Czas [s] | | | |
|---------------------|----------|----------|----------|----------|
| | 10 | 30 | 60 | 100 |
| Algorytm | Błąd [%] | Błąd [%] | Błąd [%] | Błąd [%] |
| Tabu Search | 1,30 | 1,28 | 0,98 | 0,93 |
| Algorytm genetyczny | 0,48 | 0,27 | 0,09 | 0,05 |



| | | | |
|------------|-----------------------|----------------------|--------------------------|
| Instancja: | Najlepsza kombinacja: | Wsółczynnik mutacji: | Współczynnik krzyżowania |
| rbg403 | 150 Insertion | 0,01 | 0,5 |

| Algorytm | Czas [s] | | | |
|---------------------|----------|----------|----------|----------|
| | 10 | 30 | 60 | 100 |
| Algorytm | Błąd [%] | Błąd [%] | Błąd [%] | Błąd [%] |
| Tabu Search | 64,14 | 61,05 | 50,43 | 47,42 |
| Algorytm genetyczny | 75,17 | 58,22 | 48,11 | 41,22 |



6 Omówienie wyników

W pierwszej części pomiarów przedstawione zostały uzyskiwane wyniki dla różnych rozmiarów populacji oraz zaimplementowanych metod mutacji. Zauważyć można, że w przypadku instancji br17 niezależnie od czasu, metody mutacji czy wielkości populacji uzyskiwano wynik optymalny. W przypadku instancji p43 wyróżniającą się kombinacją jest wielkość populacji równa 100 i Insertion jako metoda mutacji. Pozwala osiągnąć błąd rzędu setnych procenta przy dostatecznie długim czasie działania algorytmu. W przypadku instancji rbg403 wyróżniającą się kombinacją jest wielkość populacji równa 150 i Insertion jako metoda mutacji pozwalająca osiągnąć wynik rzędu 40 – 50% przy odpowiednim czasie działania.

Kolejno omówiony zostanie wpływ zmiennego współczynnika krzyżowania na uzyskane wyniki. Jako punkt odniesienia przyjęto najlepsze kombinacje wielkość populacji i metody mutacji z poprzedniego zadania. W przypadku instancji br17 ponownie zmiana współczynnika nie ma wpływu na wyniki, gdyż zawsze zostaje osiągnięte rozwiązanie optymalne. W przypadku instancji p43 najlepsze wyniki osiągnięto dla współczynnika krzyżowania 0,9. A w przypadku instancji rbg403 dla 0,5.

Ostatnią częścią będzie porównanie działania Algorytmu Genetycznego z Tabu Search zaimplementowanym w poprzedni etapie. Dla instancji br17 oba algorytmy osiągają rozwiązanie optymalne. W przypadku instancji p43 można wyraźnie zauważyć, że lepsze wyniki są osiągane przez Algorytm Genetyczny. W przypadku instancji rbg403 podobnie, z tą różnicą, że przewaga GA zaczyna być zauważalna przy dłuższych czasach wykonania algorytmów.

7 Wnioski

Podczas formułowania wniosków należy pamiętać, że testowaniu podlegał algorytm oparty na metodzie ewolucyjnej. Jedną z jego cech jest to, że nie daje gwarancji uzyskania rozwiązania optymalnego. W ogólności jednak jest w stanie uzyskać rozwiązanie zbliżone do optymalnego w akceptowalnym czasie. Wyniki badania algorytmu genetycznego wpisują się w tę cechę. Przy ograniczonym czasie jest w stanie uzyskać dobre wyniki. Warto także zauważyć, że zwiększenie czasu pracy algorytmu wpływa pozytywnie na jakość zwracanych przez niego rozwiązań.

Jest to nawet bardziej zauważalne niż w przypadku poprzednio zaimplementowanych algorytmów symulowanego wyżarzania i przeszukania z zakazami. W algorytmie genetycznym widać znaczą poprawę w przypadku działania przez dłuższy czas. Szczególnie dotyczy się to większych instancji gdzie wydłużenie czasu pracy może powodować nawet dwukrotne polepszenie wyników.