

Projektowanie efektywnych algorytmów

Projekt

24/01/2025

Ivan Hancharyk 264511

Zadania 3 i 4: Tabu search oraz Algorytmy genetyczne.

Spis treści

| | | |
|------|--|----|
| 1. | Sformułowanie zadania..... | 2 |
| 2. | Metody..... | 4 |
| 2.1. | Tabu search..... | 4 |
| 2.2. | Algorytmy genetyczne..... | 5 |
| 3. | Dane testowe oraz pliki config..... | 9 |
| 4. | Procedura badawcza..... | 11 |
| 5. | Wyniki..... | 12 |
| 6. | Analiza wyników i wnioski | 31 |
| | Efektywność algorytmu Tabu Search..... | 31 |
| | Efektywność Algorytmów Genetycznych..... | 31 |
| | Porównanie Tabu Search i Algorytmów Genetycznych | 32 |
| | Podsumowanie i rekomendacje..... | 32 |

1. Sformułowanie zadania

Jak i w poprzednim projekcie zadanie polega na rozwiązaniu problemu komiwojażera (TSP), ale tym razem przy opracowaniu, implementacji i zbadaniu efektywności algorytmu **Tabu Search** (przeszukiwanie z zakazami) oraz **Algorytmu Genetycznego**.

1.1. Tabu Search

W zadaniu musimy zbadać wpływ parametrów takich jak:

- Rozmiar listy ruchów zakazanych (tabu list).
- Długość kadencji – określa jak długo dany ruch ma się znajdować na liście tabu. Z każdą iteracją algorytmu długość kadencji dla każdego ruchu na liście jest zmniejszana
- Kryterium aspiracji/aspiracji plus - pozwala ono wykonać dany ruch, nawet jeśli znajduje się na liście ruchów zakazanych.
- Sposób generowania sąsiedztwa(2-opt) i wyboru rozwiązania początkowego(losowo/NN).
- Kryteria zakończenia - brak poprawy rozwiązania przez określoną liczbę iteracji/przekroczenie limitu czasu wykonania.
- Zdarzenia krytyczne - generowanie w pewnych przypadkach nowego rozwiązania startowego.

Porównać wydajność algorytmu dla instancji symetrycznych i asymetrycznych, w tym instancji używanych w zadaniach 1 i 2 oraz danych z TSPLIB.

Zmierzyć czas obliczeń, błędy wynikowe, a także znaleźć "granice możliwości" algorytmu dla różnych rozmiarów instancji.

Hipotezy badawcze:

Efektywność parametrów: Wzrost rozmiaru listy tabu poprawi unikanie minimów lokalnych, ale będzie wydłużał czas działania algorytmu. Długość kadencji wpływa na dokładność wyników; krótsze kadencje mogą przyspieszyć algorytm kosztem większych błędów, podczas gdy dłuższe kadencje mogą zapewnić lepsze wyniki, ale zwiększą czas obliczeń. Kryterium aspiracji umożliwia wydajniejsze pokonywanie minimów lokalnych, zmniejszając błąd rozwiązania, szczególnie w instancjach o dużej złożoności.

Rodzaj instancji: Algorytm Tabu Search działa skuteczniej (mniejszy błąd i krótszy czas działania) dla instancji symetrycznych w porównaniu do asymetrycznych, ze względu na prostszą strukturę przestrzeni rozwiązań.

Granica możliwości: Tabu Search jest w stanie efektywnie rozwiązywać instancje o rozmiarze do 300 wierzchołków, przy błędzie poniżej 20%, ale czas działania gwałtownie wzrośnie dla większych instancji.

Porównanie z innymi metodami: Spodziewane jest, że możliwe będzie uzyskanie rozwiązań dla większych instancji, niż w przypadku algorytmu przeglądu zupełnego lub podziału i ograniczeń, ale będą to wyniki obciążone pewnym błędem. Dla mniejszych instancji, czyli takich, że poprzednie algorytmy rozwiązały je w czasie skończonym, spodziewane jest uzyskanie wyników w czasie kilkadziesiąt, a może nawet kilkaset razy krótszym.

1.2. Algorytmy Genetyczne

W zadaniu musimy zbadać wpływ kluczowych parametrów, takich jak:

- Rozmiar populacji – 500 (niska), 1000 (standardowa) oraz 2000 (wysoka).
- Metody selekcji (np. turniejowa, ruletkowa).
- Prawdopodobieństwo krzyżowania (***pm***): 0,1 (niska), 0,8 (standardowa) oraz 0,9 (wysoka).
- Prawdopodobieństwo mutacji (***pc***): 0,1 (niska), 0,5 (standardowa) oraz 0,85 (wysoka).
- Metoda sukcesji.

Przeanalizować zależność czasu i błędów wynikowych w zależności od rozmiaru instancji, porównując efektywność dla instancji symetrycznych i asymetrycznych z użyciem danych z TSPLIB oraz instancji z zadań 1 i 2.

Hipotezy badawcze:

Wpływ parametrów algorytmu: Zwiększenie rozmiaru populacji poprawi jakość rozwiązania (mniejszy błąd), ale znacząco wydłuży czas działania algorytmu.

Prawdopodobieństwo krzyżowania w zakresie 90% pozwala na uzyskanie najlepszych wyników pod względem błędu i czasu działania. Niskie prawdopodobieństwo mutacji (np. 5%) zapewnia stabilność wyników dla małych i średnich instancji, ale wyższe wartości mogą być korzystne w przypadku dużych instancji, zwiększając różnorodność populacji.

Metody selekcji i krzyżowania: Selekcja turniejowa zapewnia lepsze wyniki jakościowe w porównaniu do selekcji ruletkowej, kosztem dłuższego czasu działania. Metoda krzyżowania PMX działa lepiej w przypadku problemu TSP niż metoda OX, szczególnie dla większych instancji, ze względu na zachowanie struktur permutacyjnych.

Ograniczenia błędu:

- $n < 25$: błąd $\leq 0\%$,
- $24 < n < 74$: błąd $\leq 50\%$,
- $75 < n < 449$: błąd $\leq 100\%$,
- $450 < n < 2500$: błąd $\leq 150\%$.

Algorytm powinien mieścić się w tych wartościach przy rozsądnym czasie działania.

2. Metody

2.1. Tabu search

Metoda poszukiwania z zakazami (ang. Tabu Search) jest metaheurystyką do rozwiązywania problemów optymalizacyjnych, opartą na iteracyjnym przeszukiwaniu przestrzeni rozwiązań, wykorzystującą (zmienne) sąsiedztwo pewnych rozwiązań, zapamiętującą niektóre ruchy i częstość ich występowania, w celu unikania minimów lokalnych i poszukiwania rozwiązań globalnie optymalnych w rozsądnym czasie.

Opis działania algorytmu:

- **Inicjalizacja:**

- Wygenerowanie początkowego rozwiązania losowo lub za pomocą heurystyki (NN).
- Inicjalizacja listy tabu, która przechowuje zakazane ruchy, aby zapobiec powrotowi do wcześniej odwiedzonych rozwiązań.
- Rozmiar listy tabu i kadencji są obliczane w kodzie (współczynniki są podowane w pliku config.txt):
 - Rozmiar listy tabu = `tabu_list_multiplier` * liczba wierzchołków.
 - Kadencja = `tabu_tenure_multiplier` * liczba wierzchołków.

- **Główna pętla:**

- **Generowanie sąsiedztwa:** Utworzenie zbioru sąsiadów bieżącego rozwiązania poprzez wykonanie **2-zamiany** (swap) albo **2-opt** (zamiana dwóch krawędzi między sobą).
- **Ocena sąsiedztwa:** Obliczenie wartości funkcji celu dla każdego sąsiada (koszt cyklu Hamiltona).
- **Wybór najlepszego sąsiada:** Wybieranie rozwiązania spośród sąsiadów z uwzględnieniem listy tabu i kryterium aspiracji.
- **Kryterium aspiracji plus:** Kryterium aspiracji plus pozwala na rozszerzenie przeglądu sąsiedztwa, jeśli znaleziony sąsiad spełnia warunek, że jego wartość jest większa lub równa określonej progowi aspiracji ($MVal \geq Aspiration$). W takim przypadku algorytm przegląda dodatkowych **Plus** sąsiadów, ale nie więcej niż do maksymalnej liczby **maxM**. Spośród wszystkich przeglądanych sąsiadów wybierane jest najlepsze rozwiązanie. Mechanizm ten zwiększa szanse na znalezienie lepszych rozwiązań w obiecującym obszarze przestrzeni poszukiwań.
- **Aktualizacja listy tabu:** Dodanie nowego ruchu do listy i usunięcie najstarszych elementów, jeśli lista przekracza maksymalny rozmiar.
- **Aktualizacja najlepszego rozwiązania:** Jeśli nowe rozwiązanie jest lepsze niż dotychczasowe najlepsze, zostaje ono zapamiętane.
- **Obsługa zdarzeń krytycznych:** W przypadku, gdy algorytm nie znajduje żadnego sąsiada spełniającego kryteria ruchu, uruchamiany jest mechanizm zdarzenia krytycznego. Algorytm generuje nowe losowe rozwiązanie startowe. Jeśli koszt tego rozwiązania jest lepszy od dotychczas najlepszego, aktualizowane jest globalne rozwiązanie, a licznik stagnacji zostaje zresetowany. Jeśli obsługa zdarzeń krytycznych jest wyłączona, algorytm kończy działanie.

2.2. Algorytmy genetyczne

Algorytmy genetyczne oparte są na biologicznym procesie ewolucji. Założeniem jest, że gorzej przystosowane osobniki z populacji zostaną wyeliminowane z populacji, a lepiej przystosowane osobniki po wykonaniu operacji genetycznych tworzących nowe osobniki stworzą subpopulację, która będzie lepiej przystosowana niż oryginalna populacja. Powtórzenie tego procesu wielokrotnie, za każdym razem eliminując nieprzystosowane osobniki spowoduje otrzymanie efektywnego rozwiązania instancji problemu zbyt dużej, aby przeszukać całą przestrzeń rozwiązań.

Na początku inicjowana jest populacja początkowa w sposób losowy a następnie obliczany jest koszt każdej ze ścieżek wchodzących w skład populacji początkowej. Kolejnym krokiem jest wybranie za pomocą selekcji osobników, które będą wchodziły w skład populacji macierzystej (rodzicielskiej). Dla populacji rodzicielskiej wykonywane są operacje genetyczne z pewnym prawdopodobieństwem, operacja krzyżowania i mutacji. Z tak powstałej subpopulacji wybierany jest za pomocą sukcesji nowa populacja w której skład będzie wchodził pewien fragment osobników ze starej populacji i osobniki z nowopowstałej subpopulacji i cały schemat jest powtarzany tak długo aż osiągnięty zostanie czas wykonywania lub nie zmieni się nic w populacji przez kilka iteracji lub po wykonaniu określonej ilości iteracji. Na poszczególne chromosomy genu składają się ścieżki na które składają się geny czyli konkretne miasta, a z kolei na geny składają się allele które są wartościami jakie mogą przyjmować geny czyli w tym wypadku numery miast.

W implementacji tego algorytmu opisywanej w tym sprawozdaniu zostały zaimplementowane dwa sposoby na selekcje oraz dwa sposoby krzyżowania. Sposoby selekcji są następujące:

1) Selekcja poprzez metodę turniejową:

Jest to sposób selekcji oparty na procesie rywalizacji dwóch osobników, z których wygrywa ten lepiej przystosowany.

Z populacji losuje się x osobników. Spośród tych osobników wybierany jest zwycięzca, czyli ten, który ma najlepsze przystosowanie (*najniższą wartość funkcji kosztu w TSP*). Następnie dodajemy go do puli macierzystej. Należy powtórzyć podany schemat następnie $N-1$ razy (N – rozmiar populacji), aby wybrać całą pulę rodziców.

2) Selekcja poprzez metodę ruletki:

Metoda ta ma symulować losowanie ruletki. Cechą tej metody jest to, że do populacji rodzicielskiej mogą dostać się osobniki gorzej przystosowane w stosunku do pozostałych. Metoda ta odbywa się poprzez wyliczenie prawdopodobieństwa wyboru osobników z populacji na podstawie względnego przystosowania:

$$p_s(x_i) = \frac{f(x_i)}{\sum_{i=1}^N f(x_i)}, i = 1, 2, \dots, N$$

W przypadku tego wzoru $f(x_i)$ jest to funkcja przystosowania czyli w przypadku tego algorytmu koszt ścieżki, a N to ilość elementów populacji.

Następnie należy obliczyć dystrybuantę dla sekwencji osobników

$$q(x_i) = \sum_{j=1}^i p_s(x_j)$$

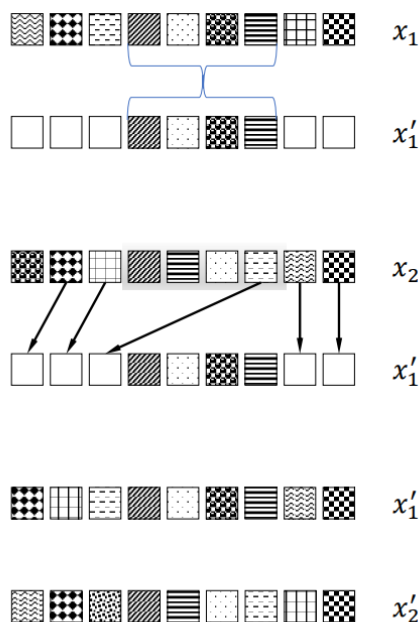
A następnie należy wylosować wartość $r \in [0,1]$ i ustalić pierwszego osobnika w populacji dla którego będzie spełniony warunek $r \leq q(x_i)$ i powtórzyć N krotnie proces losowania r oraz znajdowania osobników w populacji.

Teraz zostaną omówione metody krzyżowania. Zostały zaimplementowane metody OX oraz PMX:

1) Metoda OX (order crossover)

Metoda polega na względnym zachowaniu kolejności miast w cyklu

- W pierwszej kolejności należy wylosować w osobnikach sekcje, która zostanie przeniesiona do potomka. Do potomka pierwszego przeniesiona zostanie sekcja z pierwszego rodzica a do drugiego analogicznie z drugiego.
- Następnie w przypadku potomka z sekcją z pierwszego rodzica kopiowane są pozostałe miasta zgodnie z kolejnością z drugiego rodzica i analogicznie dla drugiego potomka.
- Ostatecznie zwracani są obaj potomkowie



Rys 1: Zobrazowanie metody OX – slajd z wykładu „Algorytmy Genetyczne w7”

2) Metoda PMX (partially matched)

Metoda polegająca na krzyżowaniu z częściowym odwzorowaniem. Przebieg tej metody wygląda następująco:

- Na początku należy wyznaczyć (np. wylosować) sekcje dopasowania, która nie może być mniejsza lub równa 1 oraz nie może być większa lub równa od długości osobnika
- Następnie należy skopiować te sekcje analogicznie do potomka pierwszego i drugiego. - Następnie trzeba skopiować sekcje które powtarzają się w jednym i drugim osobniku populacji rodzicielskiej.
- Wyznaczyć pary (i,j) w sekcjach dopasowania gdzie $i,j = 1,2,3, \dots, n; n < l - p + 1$ to są kolejne indeksy w osobnikach wybranych z puli rodzicielskiej
- Następnie należy wstawić allele, które nie występują w sekcjach dopasowania z pierwszego rodzica do drugiego potomka i analogicznie z drugiego rodzica do pierwszego potomka
- Pozostałe allele należy wstawić na miejsca w potomkach zgodnie z przyporządkowaniami w parach (i,j) .

1 2 3 4 5 6 7 8 9 P_1

5 4 6 9 2 1 7 8 3 P_2

- - 3 4 5 6 - - - N_2

- - 6 9 2 1 - - - N_1

3 ↔ 6 6 ↔ 1 5 ↔ 2 9 ↔ 4

- - 3 4 5 6 7 8 - N_2

- - 6 9 2 1 7 8 - N_1

1 2 - - - - - 9 P_1

5 4 - - - - - 3 P_2

9 z P_1 na miejsce 4 w P_2 do N_2 ; 1 z P_1 na miejsce 6 w P_2 – ponieważ 6 jest w sekcji dopasowania, zatem 1 na miejsce 3 w P_2 do N_2 ; itd.

2 9 3 4 5 6 7 8 1 N_2

3 5 6 9 2 1 7 8 4 N_1

Rys2 : zobrazowanie działania metody PMX slajd z wykłady „Algorytmy Genetyczne w7”

Metoda mutacji:

Jako metodę służącą do mutacji zaimplementowana została metoda wzajemnej wymiany (ang. Reciprocal exchange) polegająca na wybraniu pary genów w osobniku a następnie zamienieniu miejscami.

Metoda sukcesji:

Metoda `partialReplacement` implementuje sukcesję "elitizm z uzupełnieniem" (**elitism with replacement**) w algorytmie genetycznym, w której nowa populacja jest budowana w dwóch etapach. Najpierw $x\%$ (x – wybierany w config) najlepszych osobników (elita) z obecnej populacji jest bezpośrednio przenoszonych do nowej populacji, co pozwala zachować najlepsze rozwiązania. Następnie pozostałe miejsca w populacji są wypełniane nowymi osobnikami generowanymi przez krzyżowanie i mutację wybranych rodziców (wybór za pomocą ruletki lub turniejowy). Pętla uzupełnia populację do pełnej liczebności, zapewniając, że w każdej generacji populacja ma stały rozmiar. Na końcu nowa populacja zastępuje starą, przygotowując algorytm do kolejnej iteracji.

3. Dane testowe oraz pliki config

Do wykonania badań wybrano następujący zestaw instancji (z TSPLIB i ze strony Dr. Jarosława Mierzwę):

Instancje symetryczne:

tsp10_as.txt 142 [0 7 4 2 9 6 8 3 5 1 0]

tsp12_as.txt 203 [0 2 11 7 10 4 8 1 5 6 3 9 0]

ftv33.txt 1286

ftv38.txt 1530

ftv64.txt 1839

ftv70.txt 1950

rbg323.txt 1326

kroB150.txt 26130

Instancje asymetryczne:

tsp10_as.txt 142 [0 7 4 2 9 6 8 3 5 1 0]

tsp12_as.txt 203 [0 2 11 7 10 4 8 1 5 6 3 9 0]

ftv33.txt 1286

ftv38.txt 1530

ftv64.txt 1839

ftv70.txt 1950

rbg323.txt 1326

Badania zostały przeprowadzone na laptopie HP Pavilion 15:

- Procesor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 2419 Mhz, 4 Core(s), 8 Logical Processor(s)
- Pamięć RAM DDR4 16.0 GB
- System Windows 10 Pro

Jest tym samym urządzeniem, na którym były prezentowane programy. Laptop pracował przy podpiętym zasilaniu, a podczas robienia pomiarów wszystkie programy poza środowiskiem pracowniczym były zamknięte.

Opis pliku konfiguracji:

Tabu search:

- **data_file:** Ścieżka do pliku wejściowego z danymi TSP.
- **output_file:** Ścieżka do pliku wynikowego, gdzie zapisywane są rezultaty algorytmu.
- **repeats:** Liczba powtórzeń algorytmu na tej samej instancji.
- **show_progress:** Określa, czy postęp algorytmu ma być wyświetlany w konsoli.
- **initial_solution:** Sposób generowania początkowego rozwiązania (np. 2 dla metody Nearest Neighbor).
- **swap_way:** Wybrana metoda generowania sąsiedztwa (np. 0 dla 2-opt).
- **aspiration_criterion:** Minimalna wartość, aby ruch tabu mógł zostać zaakceptowany.
- **aspiration_plus:** Aktywacja kryterium aspiracji plus.
- **minM:** Minimalna liczba przeglądanych sąsiadów.
- **maxM:** Maksymalna liczba przeglądanych sąsiadów.
- **plus:** Liczba dodatkowych sąsiadów przeglądanych w przypadku aspiracji plus.
- **tabu_list_multiplier:** Współczynnik określający rozmiar listy tabu.
- **tabu_tenure_multiplier:** Współczynnik określający kadencję tabu (czas zakazu ruchu).
- **enable_critical_event_handling:** Aktywacja obsługi zdarzeń krytycznych.
- **maxIterationsWithoutImprovement:** Maksymalna liczba iteracji bez poprawy, po której algorytm się zatrzymuje.
- **termination_time:** Maksymalny czas działania algorytmu w milisekundach.

Algorytm genetyczny:

- **data_file:** Ścieżka do pliku wejściowego zawierającego dane TSP (macierz odległości).
- **output_file:** Ścieżka do pliku wynikowego, gdzie zapisywane są wyniki działania algorytmu(.xls).
- **repeats:** Liczba powtórzeń algorytmu na tej samej instancji.
- **show_progress:** Określa, czy postęp algorytmu ma być wyświetlany w konsoli.
- **population_size:** Rozmiar populacji w algorytmie genetycznym.
- **max_generations:** Maksymalna liczba generacji, po których algorytm kończy działanie.
- **mutation_probability:** Prawdopodobieństwo wykonania mutacji na osobniku.
- **elitePercentage:** procent najlepszych osobników (elity), które są przenoszone do kolejnej generacji bez zmian.
- **selectionMethod:** Wybór metody selekcji osobników rodzicielskich (np. "roulette" dla ruletki).
- **crossoverMethod:** Wybór metody krzyżowania (np. "ox" dla Order Crossover).
- **crossover_probability:** Prawdopodobieństwo wykonania krzyżowania.
- **timeout:** Maksymalny czas działania algorytmu w sekundach.

4.Procedura badawcza

Procedura miała na celu potwierdzenie lub odrzucenie postawionych hipotez.

Tabu Search:

Parametry domyślne algorytmu:

- **Rozmiar listy tabu:** $3 \times n$, gdzie n to liczba miast w instancji.
 - **Kadencja tabu:** $25\% \times n$.
 - **Rozwiązanie początkowe:** NN.
 - **Kryterium aspiracji:** Akceptacja rozwiązań na liście tabu, jeśli są lepsze o 20.
 - **Aspiracja plus:** włączona plus=5, minM=3, maxM=20.
 - **Metoda wyboru rozwiązania w sąsiedztwie:** 2-opt (swap_way = 0).
 - **Kryterium zakończenia:** Maksymalna czas wykonania 15min lub brak poprawy przez 400 iteracji.
 - **Obsługa zdarzeń krytycznych:** Włączona (restart z losowym rozwiązaniem przy stagnacji).
1. Uruchomiono algorytm dla instancji o różnych rozmiarach, od 10 do 323 miast, zapisując wyniki w formacie .xls.
 2. Każda instancja była rozwiązywana 10-krotnie dla rozmiarów $n \leq 100$, a 5-krotnie dla $n > 100$, w celu analizy stabilności wyników.

Algorytm genetyczny:

Parametry domyślne algorytmu:

- **Wielkość populacji:** 500.
- **Metoda selekcji:** Turniejowa.
- **Prawdopodobieństwo krzyżowania:** 0.7.
- **Prawdopodobieństwo mutacji:** 0.3.
- **Metoda krzyżowania:** Order Crossover (OX).
- **Elita:** 0.3.
- **Kryterium zatrzymania:**
 - Brak zmian w populacji przez 300 iteracji.
 - Limit czasu: 15 minut.
- Testowano algorytm na instancjach o rozmiarze od 10 do 323 miast, zapisując wyniki w plikach .xls.
- Każda instancja była rozwiązywana 10-krotnie, aby zbadać stabilność i średnią jakość wyników.

Wyniki opracowano za pomocą programu MS Excel. Wartość błędu liczone według wzoru:

$$\delta = \frac{|x - x_0|}{x} \cdot 100\%$$

Gdzie x to wartość optymalna, a x_0 to uzyskany koszt.

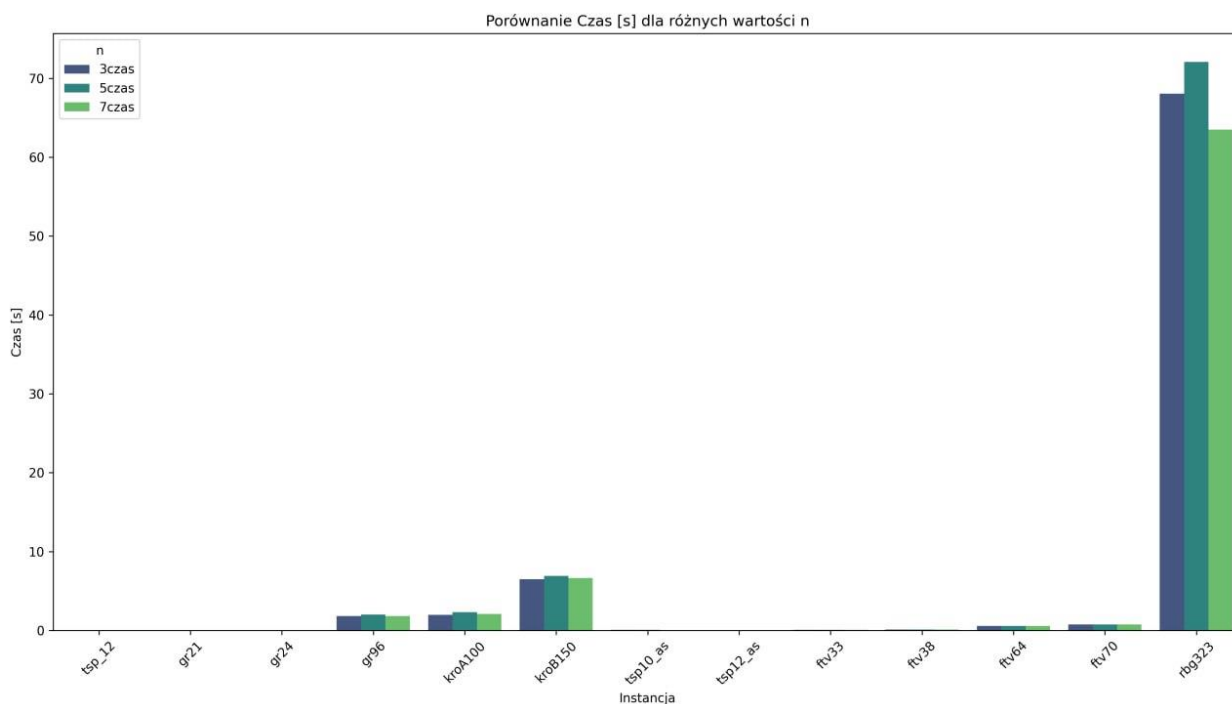
5. Wyniki

Błąd jest w %, czas dla Tabu search w sekundach, natomiast dla GA jest w milisekundach.

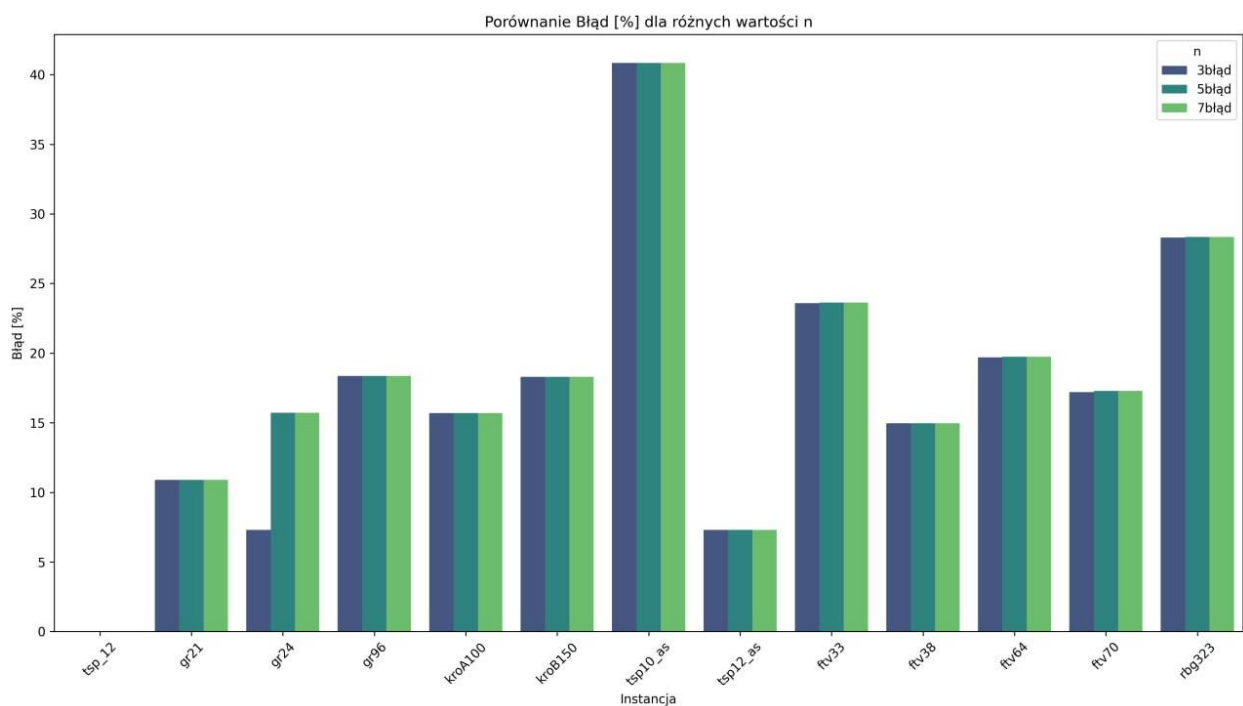
Tabu search:

| instancja | koszt optymalny | 3 x n | | | 5 x n | | | 7 x n | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 213 | 0,4 | 0,013 | 213 | 0,47 | 0,013 | 213 | 0,47 | 0,013 |
| tsp_12 | 264 | 264 | 0 | 0,013 | 264 | 0 | 0,01 | 264 | 0 | 0,014 |
| gr21 | 2707 | 3002 | 10,89 | 0,03 | 3002 | 10,89 | 0,05 | 3002 | 10,89 | 0,032 |
| gr24 | 1272 | 1334 | 7,30 | 0,044 | 1472 | 15,72 | 0,044 | 1472 | 15,72 | 0,043 |
| gr96 | 55209 | 65342 | 18,35 | 1,83 | 65342 | 18,35 | 2,01 | 65342 | 18,35 | 1,818 |
| kroA100 | 21282 | 24698 | 15,70 | 1,99 | 24698 | 15,7 | 2,32 | 24698 | 15,7 | 2,09 |
| kroB150 | 26130 | 31611 | 18,30 | 6,5 | 31611 | 18,3 | 6,9 | 31611 | 18,3 | 6,67 |
| tsp10_as | 142 | 200 | 40,84 | 0,1 | 200 | 40,84 | 0,1 | 200 | 40,84 | 0,01 |
| tsp12_as | 203 | 218 | 7,30 | 0,012 | 218 | 7,3 | 0,012 | 218 | 7,3 | 0,013 |
| ftv33 | 1286 | 1590 | 23,60 | 0,09 | 1590 | 23,63 | 0,09 | 1590 | 23,63 | 0,1 |
| ftv38 | 1530 | 1759 | 14,96 | 0,14 | 1759 | 14,96 | 0,14 | 1759 | 14,96 | 0,14 |
| ftv64 | 1839 | 2202 | 19,70 | 0,58 | 2202 | 19,73 | 0,58 | 2202 | 19,73 | 0,568 |
| ftv70 | 1950 | 2287 | 17,20 | 0,75 | 2287 | 17,28 | 0,75 | 2287 | 17,28 | 0,768 |
| rbg323 | 1326 | 1702 | 28,30 | 68,06 | 1702 | 28,35 | 72,06 | 1702 | 28,35 | 63,488 |

Tabela 1: wyniki wpływu rozmiaru listy tabu



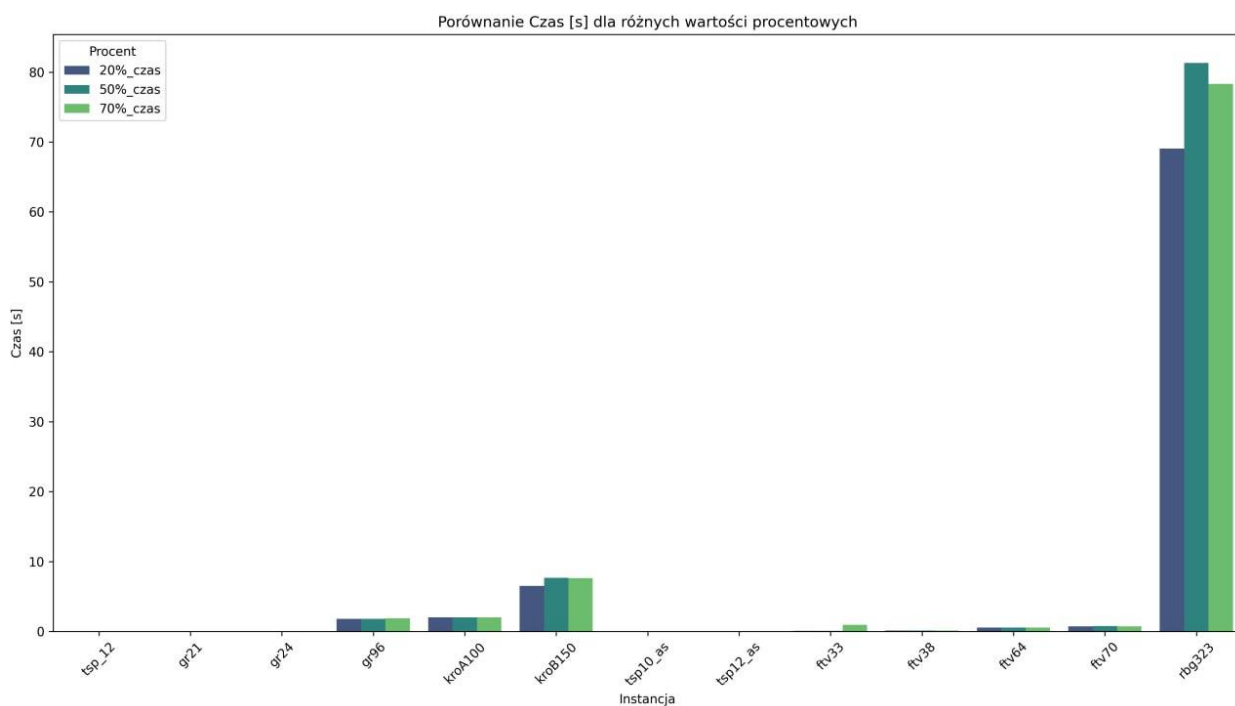
Rysunek 3: wykres zależności czasu od rozmiaru instancji dla różnych rozmiarów listy tabu



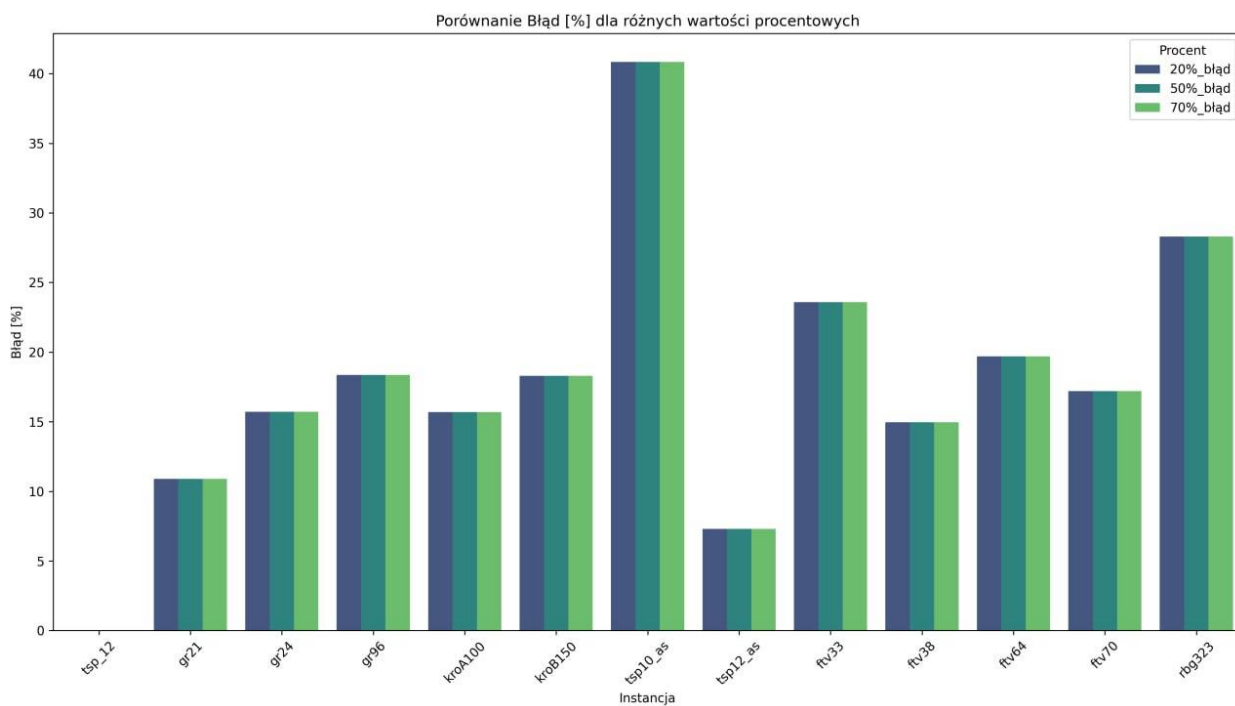
Rysunek 4: wykres zależności błędu od rozmiaru instancji dla różnych rozmiarów listy tabu

| instancja | koszt optymalny | 20% x n | | | 50% x n | | | 70% x n | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 213 | 0,4 | 0,012 | 213 | 0,4 | 0,012 | 213 | 0,4 | 0,014 |
| tsp_12 | 264 | 264 | 0 | 0,014 | 264 | 0 | 0,0131 | 264 | 0 | 0,013 |
| gr21 | 2707 | 3002 | 10,89 | 0,032 | 3002 | 10,89 | 0,033 | 3002 | 10,89 | 0,032 |
| gr24 | 1272 | 1472 | 15,72 | 0,043 | 1472 | 15,72 | 0,049 | 1472 | 15,72 | 0,045 |
| gr96 | 55209 | 65342 | 18,35 | 1,81 | 65342 | 18,35 | 1,804 | 65342 | 18,35 | 1,9 |
| kroA100 | 21282 | 24698 | 15,70 | 2,007 | 24698 | 15,70 | 2,03 | 24698 | 15,70 | 2,01 |
| kroB150 | 26130 | 31611 | 18,30 | 6,54 | 31611 | 18,30 | 7,66 | 31611 | 18,30 | 7,65 |
| tsp10_as | 142 | 200 | 40,84 | 0,012 | 200 | 40,84 | 0,01 | 200 | 40,84 | 0,014 |
| tsp12_as | 203 | 218 | 7,30 | 0,013 | 218 | 7,30 | 0,01 | 218 | 7,30 | 0,013 |
| ftv33 | 1286 | 1590 | 23,60 | 0,098 | 1590 | 23,60 | 0,1 | 1590 | 23,60 | 0,96 |
| ftv38 | 1530 | 1759 | 14,96 | 0,143 | 1759 | 14,96 | 0,14 | 1759 | 14,96 | 0,14 |
| ftv64 | 1839 | 2202 | 19,70 | 0,572 | 2202 | 19,70 | 0,58 | 2202 | 19,70 | 0,57 |
| ftv70 | 1950 | 2287 | 17,20 | 0,748 | 2287 | 17,20 | 0,77 | 2287 | 17,20 | 0,74 |
| rbg323 | 1326 | 1702 | 28,30 | 69,07 | 1702 | 28,30 | 81,29 | 1702 | 28,30 | 78,3 |

Tabela 2: wyniki wpływu długości kadencji



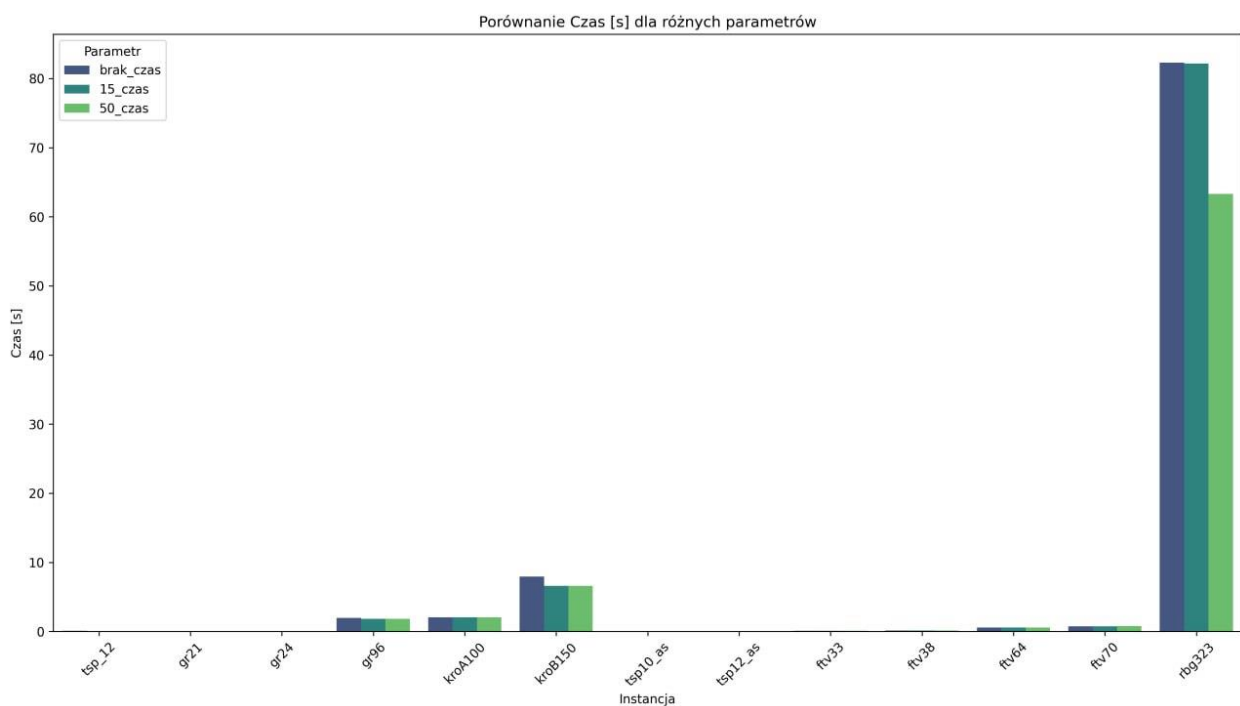
Rysunek 5: wykres zależności czasu od rozmiaru instancji dla różnych długości kadencji



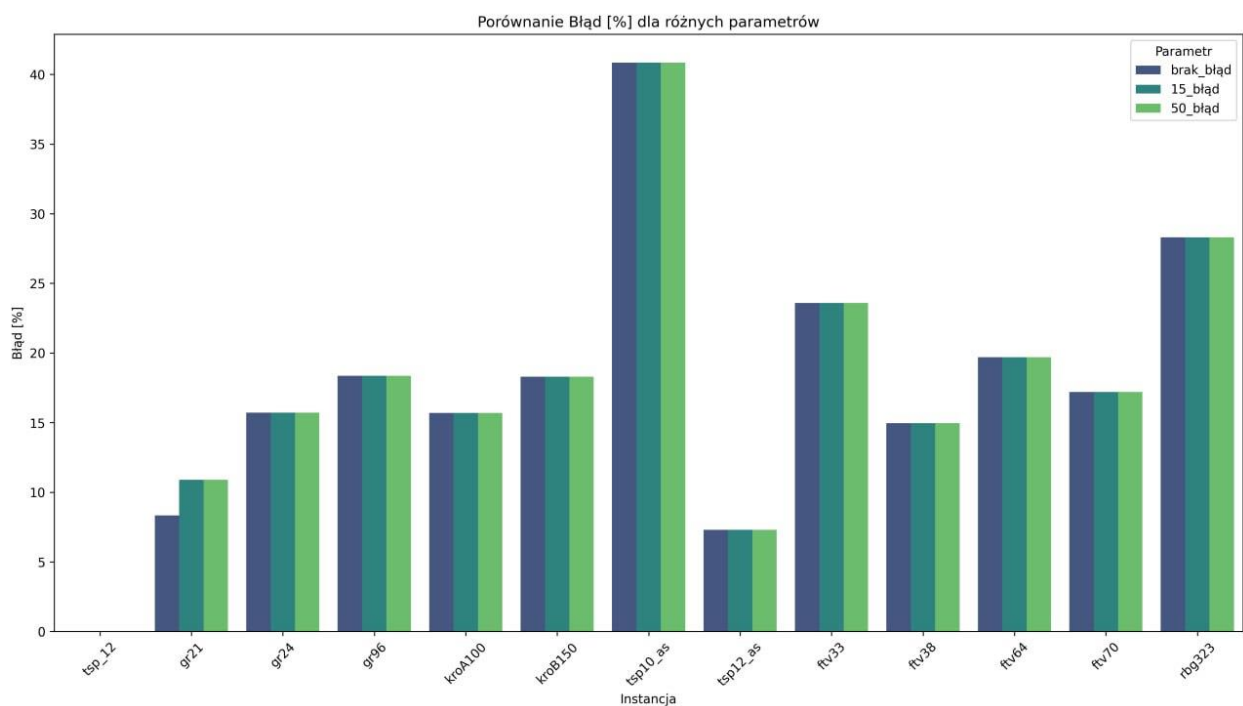
Rysunek 6: wykres zależności błędu od rozmiaru instancji dla różnych długości kadencji

| instancja | koszt optymalny | brak | | | 15 | | | 50 | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 213 | 0,4 | 0,014 | 213 | 0,4 | 0,013 | 213 | 0,4 | 0,0134 |
| tsp_12 | 264 | 264 | 0 | 0,105 | 264 | 0 | 0,014 | 264 | 0 | 0,0133 |
| gr21 | 2707 | 2933 | 8,34 | 0,035 | 3002 | 10,89 | 0,034 | 3002 | 10,89 | 0,034 |
| gr24 | 1272 | 1472 | 15,72 | 0,055 | 1472 | 15,72 | 0,051 | 1472 | 15,72 | 0,045 |
| gr96 | 55209 | 65342 | 18,35 | 1,98 | 65342 | 18,35 | 1,832 | 65342 | 18,35 | 1,82 |
| kroA100 | 21282 | 24698 | 15,70 | 2,05 | 24698 | 15,70 | 2,03 | 24698 | 15,70 | 2,03 |
| kroB150 | 26130 | 31611 | 18,30 | 7,93 | 31611 | 18,30 | 6,62 | 31611 | 18,30 | 6,61 |
| tsp10_as | 142 | 200 | 40,84 | 0,01 | 200 | 40,84 | 0,01 | 200 | 40,84 | 0,01 |
| tsp12_as | 203 | 218 | 7,30 | 0,017 | 218 | 7,30 | 0,014 | 218 | 7,30 | 0,013 |
| ftv33 | 1286 | 1590 | 23,60 | 0,105 | 1590 | 23,60 | 0,097 | 1590 | 23,60 | 0,101 |
| ftv38 | 1530 | 1759 | 14,96 | 0,146 | 1759 | 14,96 | 0,14 | 1759 | 14,96 | 0,141 |
| ftv64 | 1839 | 2202 | 19,70 | 0,58 | 2202 | 19,70 | 0,583 | 2202 | 19,70 | 0,58 |
| ftv70 | 1950 | 2287 | 17,20 | 0,74 | 2287 | 17,20 | 0,756 | 2287 | 17,20 | 0,78 |
| rbg323 | 1326 | 1702 | 28,30 | 82,28 | 1702 | 28,30 | 82,18 | 1702 | 28,30 | 63,3 |

Tabela 3: wyniki wpływu kryterium aspiracji



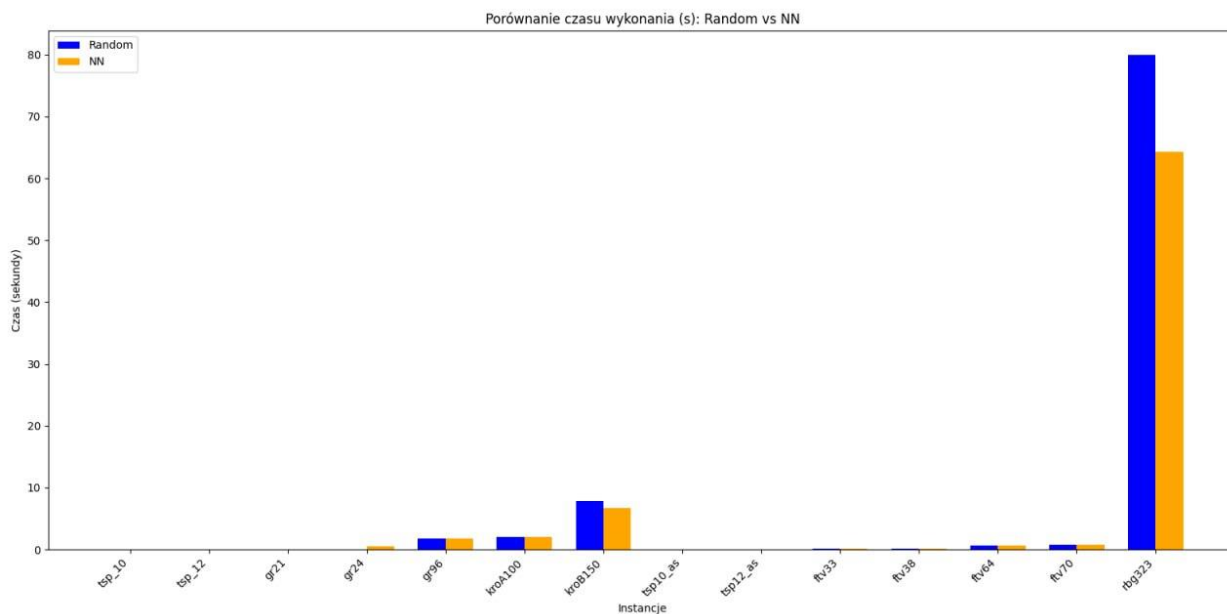
Rysunek 7: wykres zależności czasu od rozmiaru instancji dla różnych wartości kryterium aspiracji



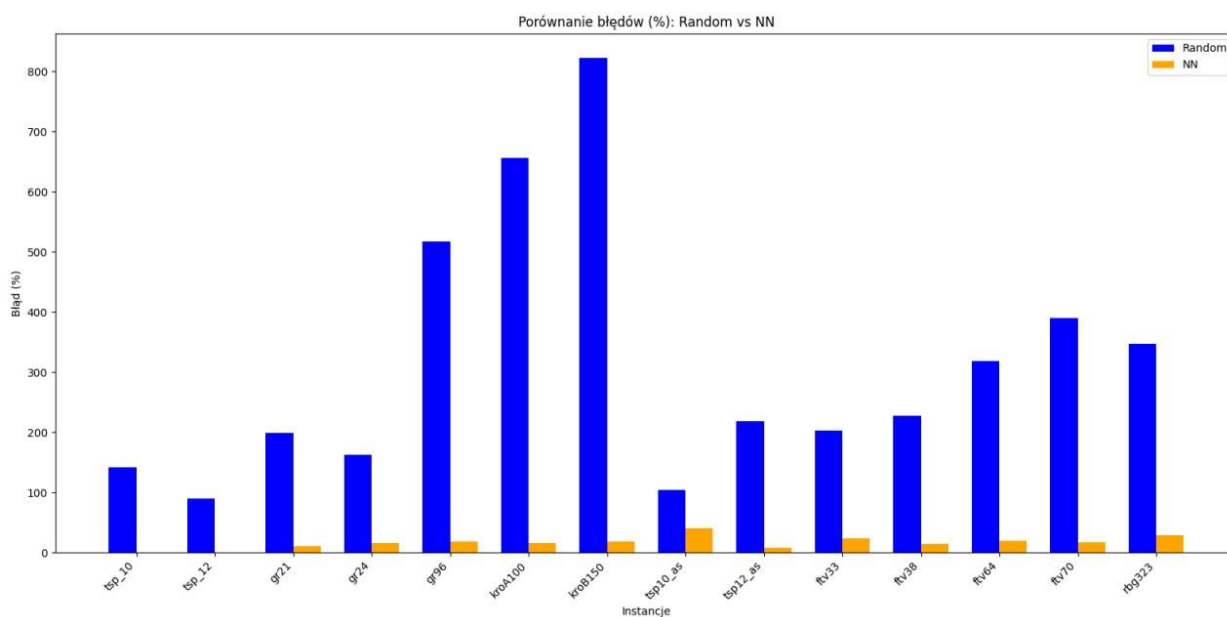
Rysunek 8: wykres zależności błędu od rozmiaru instancji dla różnych wartości kryterium aspiracji

| instancja | koszt optymalny | random | | | NN | | |
|-----------|-----------------|--------------|---------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 511 | 141,03 | 0,014 | 213 | 0,4 | 0,016 |
| tsp_12 | 264 | 501 | 89,77 | 0,015 | 264 | 0 | 0,014 |
| gr21 | 2707 | 8082 | 198,55 | 0,032 | 3002 | 10,89 | 0,03 |
| gr24 | 1272 | 3341 | 162,65 | 0,045 | 1472 | 15,72 | 0,46 |
| gr96 | 55209 | 340912 | 517,49 | 1,83 | 65342 | 18,35 | 1,82 |
| kroA100 | 21282 | 160816 | 655,64 | 1,99 | 24698 | 15,70 | 2,05 |
| kroB150 | 26130 | 240812 | 821,592 | 7,82 | 31611 | 18,30 | 6,67 |
| tsp10_as | 142 | 290 | 104,22 | 0,012 | 200 | 40,84 | 0,01 |
| tsp12_as | 203 | 646 | 218,22 | 0,015 | 218 | 7,30 | 0,013 |
| ftv33 | 1286 | 3888 | 202,33 | 0,102 | 1590 | 23,60 | 0,098 |
| ftv38 | 1530 | 5005 | 227,12 | 0,14 | 1759 | 14,96 | 0,147 |
| ftv64 | 1839 | 7697 | 318,54 | 0,62 | 2202 | 19,70 | 0,59 |
| ftv70 | 1950 | 9539 | 389,17 | 0,79 | 2287 | 17,20 | 0,74 |
| rbg323 | 1326 | 5929 | 347,13 | 79,91 | 1702 | 28,30 | 64,35 |

Tabela 4: wyniki wpływu metody generowania rozwiązania początkowego



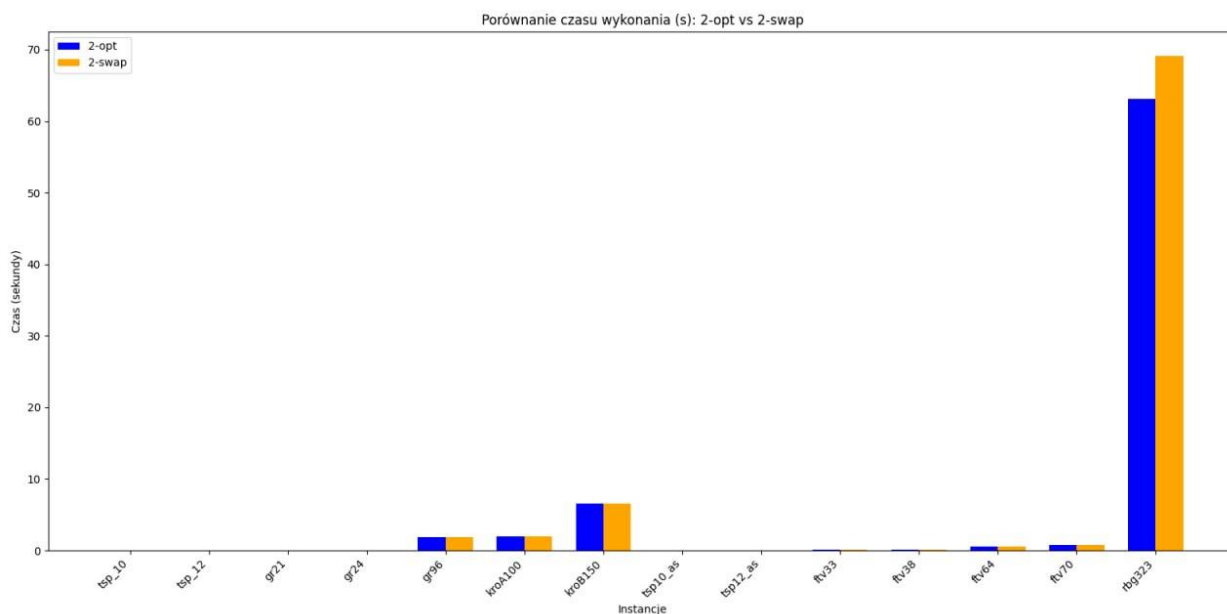
Rysunek 9: wykres zależności czasu od rozmiaru instancji dla różnych metod tworzenia rozwiązania pierwotnego



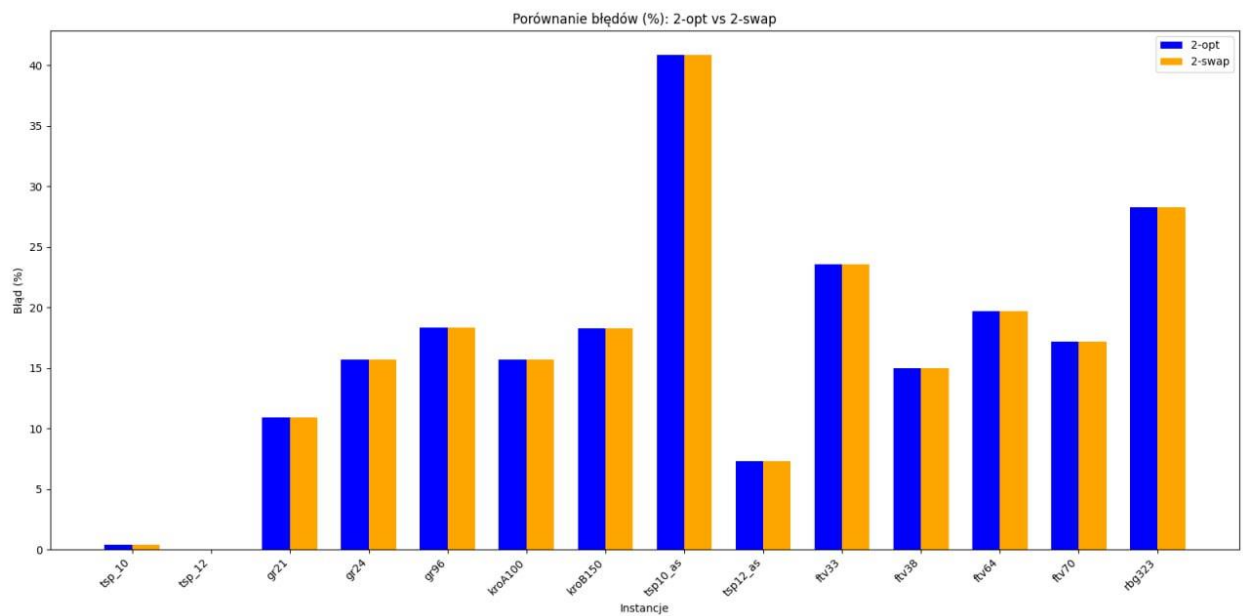
Rysunek 10: wykres zależności błędu od rozmiaru instancji dla różnych metod tworzenia rozwiązania pierwotnego

| instancja | koszt optymalny | 2-opt | | | 2-swap | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 213 | 0,4 | 0,013 | 213 | 0,4 | 0,012 |
| tsp_12 | 264 | 264 | 0 | 0,014 | 264 | 0 | 0,014 |
| gr21 | 2707 | 3002 | 10,89 | 0,033 | 3002 | 10,89 | 0,032 |
| gr24 | 1272 | 1472 | 15,72 | 0,046 | 1472 | 15,72 | 0,043 |
| gr96 | 55209 | 65342 | 18,35 | 1,82 | 65342 | 18,35 | 1,81 |
| kroA100 | 21282 | 24698 | 15,70 | 2 | 24698 | 15,70 | 2,007 |
| kroB150 | 26130 | 31611 | 18,30 | 6,58 | 31611 | 18,30 | 6,54 |
| tsp10_as | 142 | 200 | 40,84 | 0,012 | 200 | 40,84 | 0,012 |
| tsp12_as | 203 | 218 | 7,30 | 0,013 | 218 | 7,30 | 0,013 |
| ftv33 | 1286 | 1590 | 23,60 | 0,1 | 1590 | 23,60 | 0,098 |
| ftv38 | 1530 | 1759 | 14,96 | 0,14 | 1759 | 14,96 | 0,143 |
| ftv64 | 1839 | 2202 | 19,70 | 0,59 | 2202 | 19,70 | 0,572 |
| ftv70 | 1950 | 2287 | 17,20 | 0,75 | 2287 | 17,20 | 0,748 |
| rbg323 | 1326 | 1702 | 28,30 | 63,14 | 1702 | 28,30 | 69,07 |

Tabela 5: wyniki wpływu metody rozwiązania w sąsiedztwie



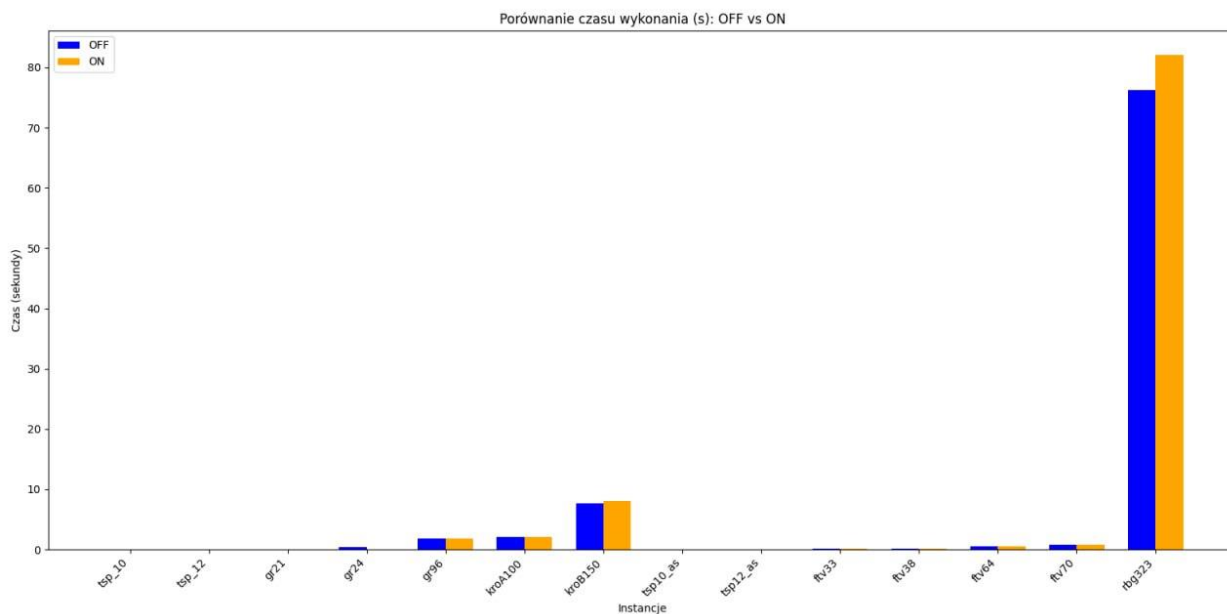
Rysunek 11: wykres zależności czasu od rozmiaru instancji dla różnych metod przeglądania sąsiedztwa



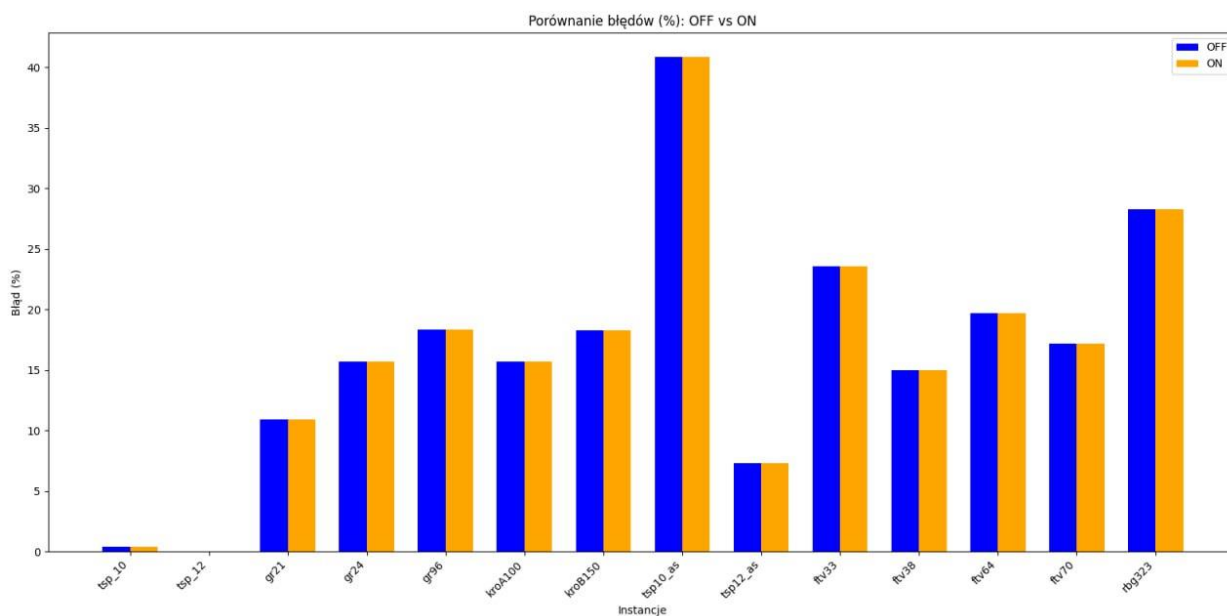
Rysunek 12: wykres zależności błędu od rozmiaru instancji dla różnych metod przeglądów sąsiedztwa

| instancja | koszt optymalny | zdarzenia krytyczne - off | | | zdarzenia krytyczne - on | | |
|-----------|-----------------|---------------------------|-------|-------------|--------------------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 213 | 0,4 | 0,013 | 213 | 0,4 | 0,012 |
| tsp_12 | 264 | 264 | 0 | 0,014 | 264 | 0 | 0,013 |
| gr21 | 2707 | 3002 | 10,89 | 0,032 | 3002 | 10,89 | 0,340,034 |
| gr24 | 1272 | 1472 | 15,72 | 0,43 | 1472 | 15,72 | 0,045 |
| gr96 | 55209 | 65342 | 18,35 | 1,83 | 65342 | 18,35 | 1,78 |
| kroA100 | 21282 | 24698 | 15,70 | 2,03 | 24698 | 15,70 | 2,02 |
| kroB150 | 26130 | 31611 | 18,30 | 7,71 | 31611 | 18,30 | 8,01 |
| tsp10_as | 142 | 200 | 40,84 | 0,013 | 200 | 40,84 | 0,015 |
| tsp12_as | 203 | 218 | 7,30 | 0,015 | 218 | 7,30 | 0,013 |
| ftv33 | 1286 | 1590 | 23,60 | 0,098 | 1590 | 23,60 | 0,1 |
| ftv38 | 1530 | 1759 | 14,96 | 0,147 | 1759 | 14,96 | 0,14 |
| ftv64 | 1839 | 2202 | 19,70 | 0,581 | 2202 | 19,70 | 0,58 |
| ftv70 | 1950 | 2287 | 17,20 | 0,746 | 2287 | 17,20 | 0,75 |
| rbg323 | 1326 | 1702 | 28,30 | 76,29 | 1702 | 28,30 | 82,01 |

Tabela 6: wyniki wpływu obsługi zdarzeń krytycznych



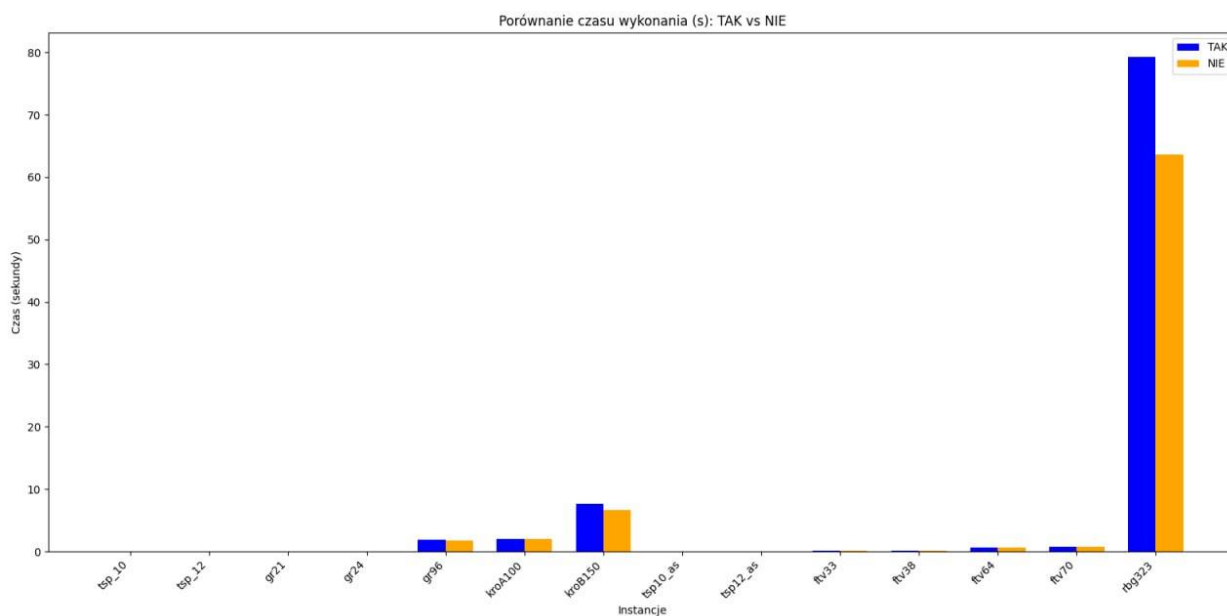
Rysunek 13: wykres zależności czasu od rozmiaru instancji przy włączonej/wyłączonej obsłudze zdarzeń krytycznych



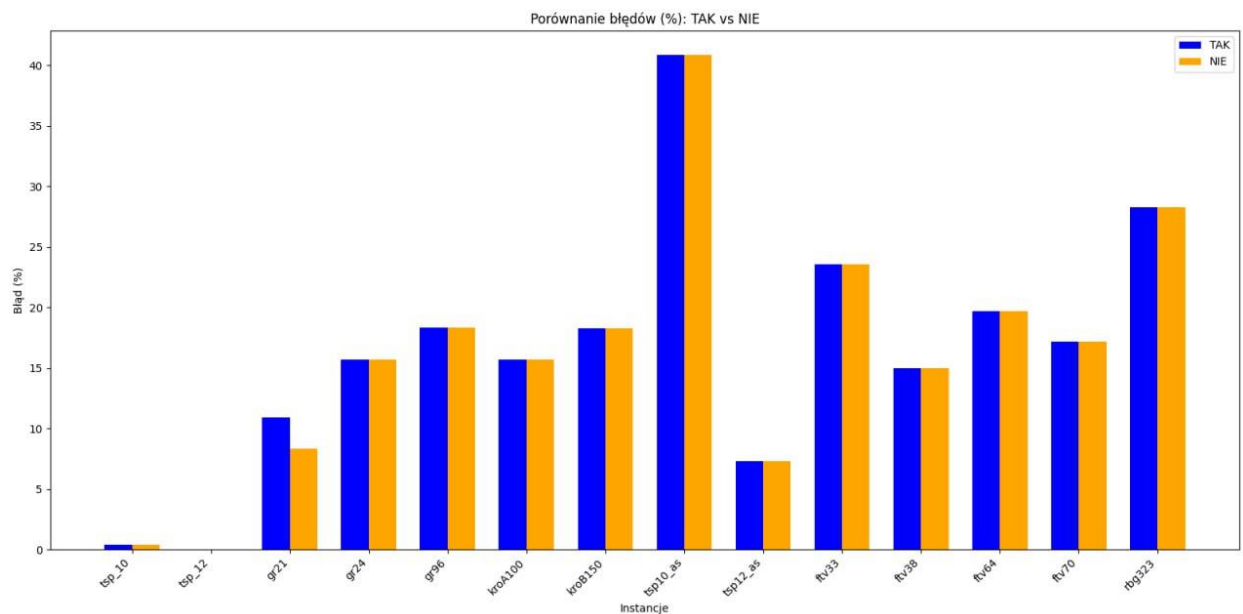
Rysunek 14: wykres zależności błędów od rozmiaru instancji przy włączonej/wyłączonej obsłudze zdarzeń krytycznych

| instancja | pszt optymaln | TAK | | | NIE | | |
|-----------|---------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 213 | 0,4 | 0,013 | 213 | 0,4 | 0,014 |
| tsp_12 | 264 | 264 | 0 | 0,013 | 264 | 0 | 0,017 |
| gr21 | 2707 | 3002 | 10,89 | 0,031 | 2933 | 8,34 | 0,035 |
| gr24 | 1272 | 1472 | 15,72 | 0,044 | 1472 | 15,72 | 0,048 |
| gr96 | 55209 | 65342 | 18,35 | 1,83 | 65342 | 18,35 | 1,82 |
| kroA100 | 21282 | 24698 | 15,70 | 2,04 | 24698 | 15,70 | 2,04 |
| kroB150 | 26130 | 31611 | 18,30 | 7,62 | 31611 | 18,30 | 6,59 |
| tsp10_as | 142 | 200 | 40,84 | 0,013 | 200 | 40,84 | 0,013 |
| tsp12_as | 203 | 218 | 7,30 | 0,014 | 218 | 7,30 | 0,015 |
| ftv33 | 1286 | 1590 | 23,60 | 0,097 | 1590 | 23,60 | 0,103 |
| ftv38 | 1530 | 1759 | 14,96 | 0,146 | 1759 | 14,96 | 0,14 |
| ftv64 | 1839 | 2202 | 19,70 | 0,59 | 2202 | 19,70 | 0,586 |
| ftv70 | 1950 | 2287 | 17,20 | 0,74 | 2287 | 17,20 | 0,762 |
| rbg323 | 1326 | 1702 | 28,30 | 79,21 | 1702 | 28,30 | 63,642 |

Tabela 7: wyniki wpływu kryterium aspiracji plus



Rysunek 15: wykres zależności czasu od rozmiaru instancji przy włączonej/wyłączonej obsłudze kryterium aspiracji plus

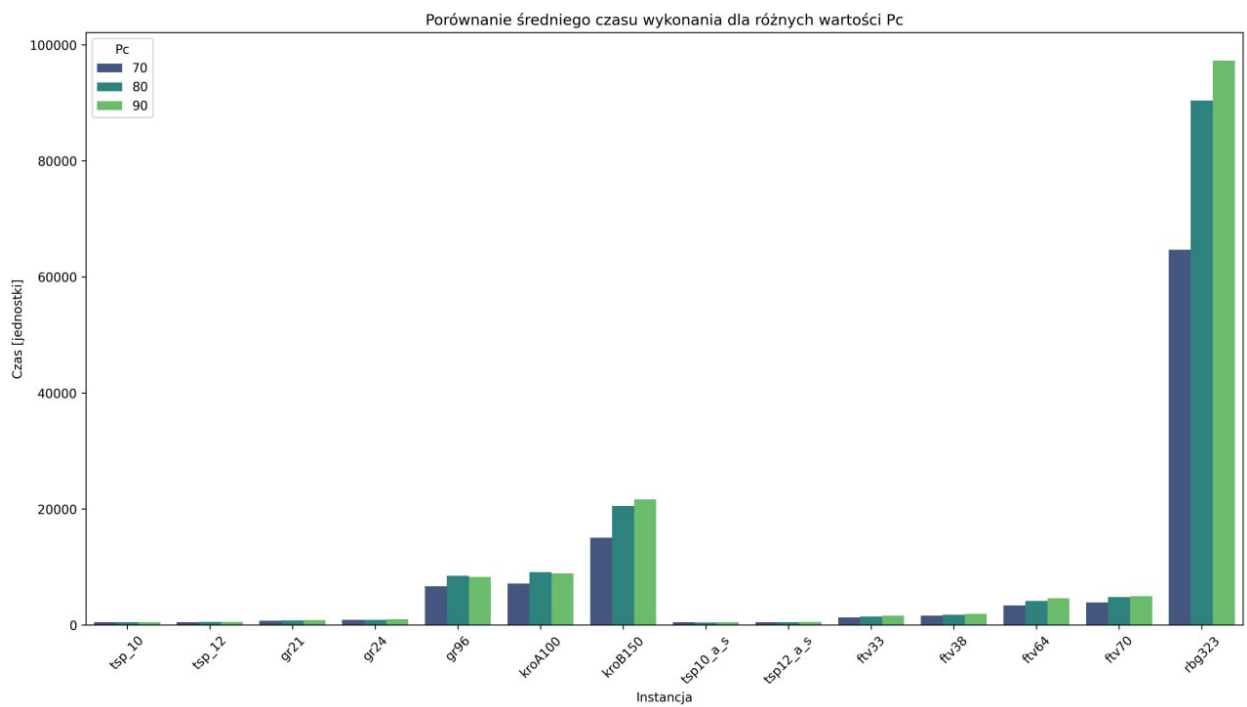


Rysunek 16: wykres zależności błędu od rozmiaru instancji przy włączonej/wyłączonej obsłudze kryterium aspiracji plus

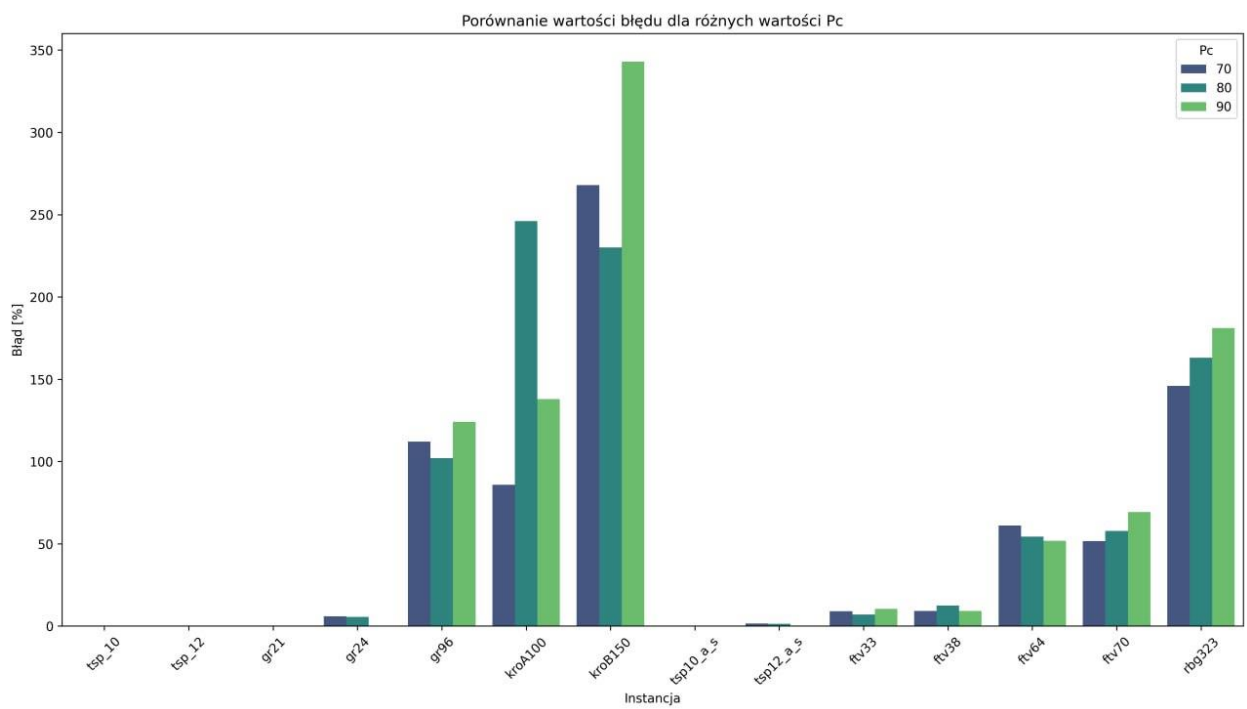
Algorytmy genetyczne:

| instancja | koszt optymalny | Pc=70% | | | Pc=80% | | | Pc=90% | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 212 | 0 | 475 | 212 | 0 | 463 | 212 | 0 | 502 |
| tsp_12 | 264 | 264 | 0 | 502 | 264 | 0 | 538 | 264 | 0 | 553 |
| gr21 | 2707 | 2707 | 0 | 764 | 2707 | 0 | 813 | 2707 | 0 | 857 |
| gr24 | 1272 | 1348 | 5,97 | 885 | 1342 | 5,5 | 915 | 1272 | 0 | 983 |
| gr96 | 55209 | 117078 | 112 | 6673 | 111716 | 102 | 8490 | 123875 | 124 | 8262 |
| kroA100 | 21282 | 39548 | 85,82 | 7152 | 73734 | 246 | 9109 | 50653 | 138 | 8925 |
| kroB150 | 26130 | 96413 | 268 | 15075 | 86313 | 230 | 20549 | 115847 | 343 | 21670 |
| tsp10_as | 142 | 142 | 0 | 471 | 142 | 0 | 457 | 142 | 0 | 488 |
| tsp12_as | 203 | 206 | 1,47 | 488 | 206 | 1,4 | 511 | 203 | 0 | 547 |
| ftv33 | 1286 | 1403 | 9,09 | 1307 | 1377 | 7,07 | 1477 | 1420 | 10,41 | 1603 |
| ftv38 | 1530 | 1671 | 9,21 | 1599 | 1721 | 12,48 | 1771 | 1672 | 9,2 | 1942 |
| ftv64 | 1839 | 2972 | 61 | 3381 | 2841 | 54,4 | 4146 | 953 | 51,8 | 4644 |
| ftv70 | 1950 | 2957 | 51,64 | 3918 | 3076 | 57,74 | 4830 | 3301 | 69,28 | 4993 |
| rbg323 | 1326 | 3274 | 146 | 64702 | 3494 | 163 | 90371 | 3728 | 181 | 97274 |

Tabela 8: wyniki wpływu prawdopodobieństwa krzyżowania



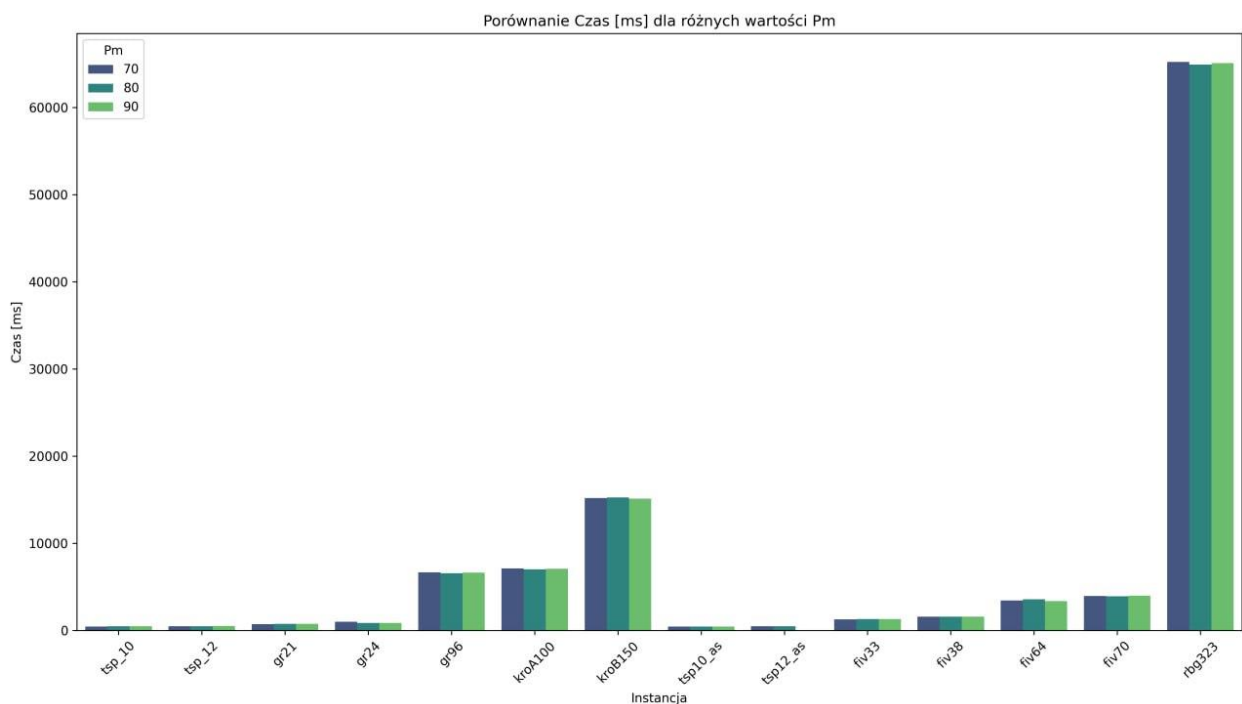
Rysunek 17: wykres zależności czasu od rozmiaru instancji przy różnych wartościach prawdopodobieństwa krzyżownia



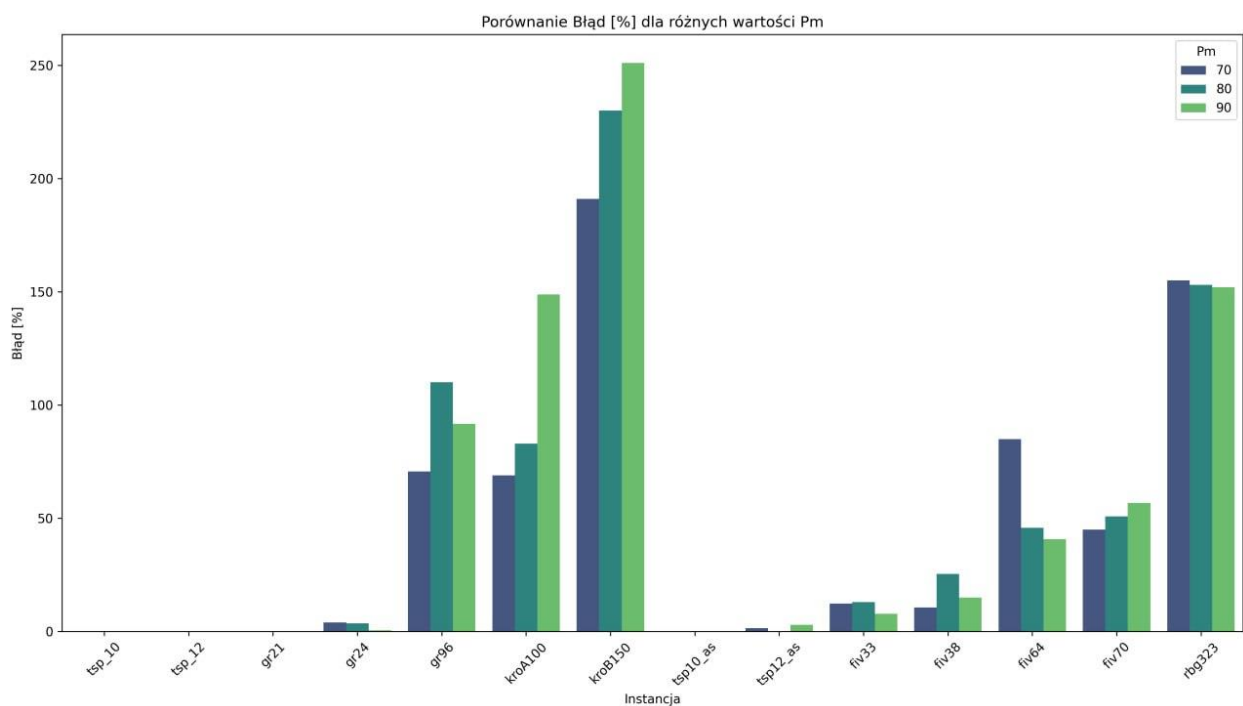
Rysunek 18: wykres zależności błędów od rozmiaru instancji przy różnych wartościach prawdopodobieństwa krzyżownia

| instancja | koszt optymalny | Pm=10% | | | Pm=30% | | | Pm=50% | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|--------------|--------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 212 | 0 | 470 | 212 | 0 | 477 | 212 | 0 | 472 |
| tsp_12 | 264 | 264 | 0 | 486 | 264 | 0 | 495 | 264 | 0 | 521 |
| gr21 | 2707 | 2707 | 0 | 741 | 2707 | 0 | 762 | 2707 | 0 | 758 |
| gr24 | 1272 | 1323 | 4 | 988 | 1318 | 3,61 | 851 | 1279 | 0,5 | 857 |
| gr96 | 55209 | 94212 | 70,64 | 6666,72 | 115965 | 110 | 6573 | 105852 | 91,72 | 6631 |
| kroA100 | 21282 | 35963 | 68,98 | 7135 | 39123 | 83 | 7015 | 52967 | 148,88 | 7081 |
| kroB150 | 26130 | 76285 | 191 | 15200 | 86346 | 230 | 15268 | 91732 | 251 | 15124 |
| tsp10_as | 142 | 142 | 0 | 441 | 142 | 0 | 457 | 142 | 0 | 445 |
| tsp12_as | 203 | 206 | 1,47 | 477 | 203 | 0 | 486 | 206 | 3 | 1,47 |
| ftv33 | 1286 | 1444 | 12,28 | 1282 | 1453 | 12,98 | 1326 | 1386 | 7,77 | 1308 |
| ftv38 | 1530 | 1693 | 10,65 | 1575 | 1919 | 25,42 | 1576 | 1759 | 14,96 | 1596 |
| ftv64 | 1839 | 3403 | 85 | 3452 | 2682 | 45,84 | 3572 | 2591 | 40,8 | 3363 |
| ftv70 | 1950 | 2835 | 45 | 3975 | 2941 | 50,82 | 3926 | 3057 | 56,76 | 3982 |
| rbg323 | 1326 | 3386 | 155 | 65200 | 3359 | 153 | 64916 | 3353 | 152 | 65086 |

Tabela 9: wyniki wpływu prawdopodobieństwa mutacji



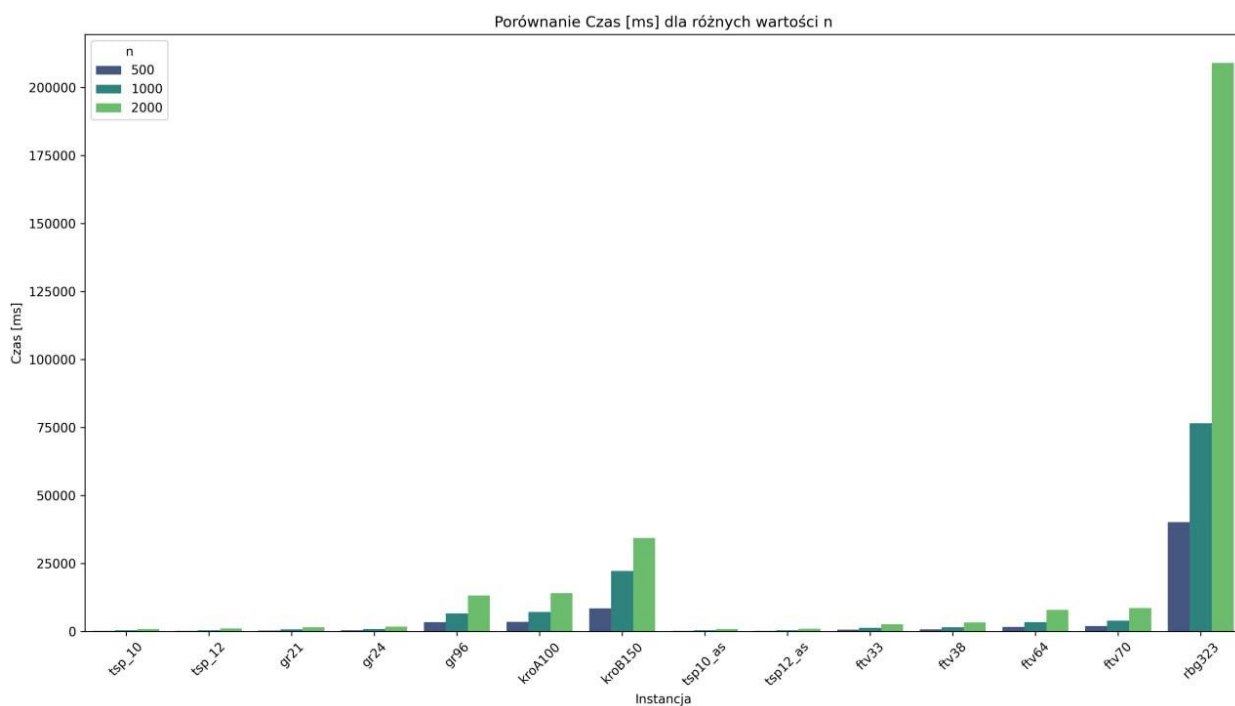
Rysunek 19: wykres zależności czasu od rozmiaru instancji przy różnych wartościach prawdopodobieństwa mutacji



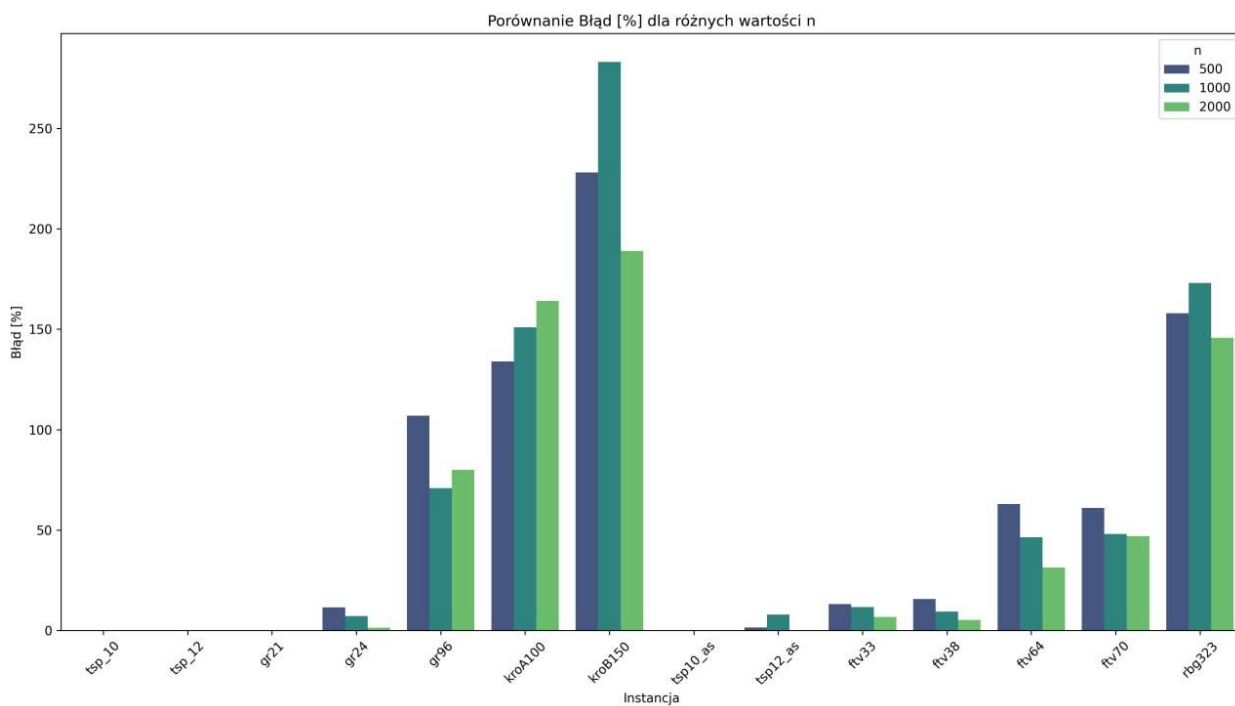
Rysunek 20: wykres zależności błędu od rozmiaru instancji przy różnych wartościach prawdopodobieństwa mutacji

| instancja | koszt optymalny | n=500 | | | n=1000 | | | n=2000 | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|--------------|--------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 212 | 0 | 233 | 212 | 0 | 462 | 212 | 0 | 941 |
| tsp_12 | 264 | 264 | 0 | 259 | 264 | 0 | 514 | 264 | 0 | 1121 |
| gr21 | 2707 | 2707 | 0 | 369 | 2707 | 0 | 765 | 2707 | 0 | 1542 |
| gr24 | 1272 | 1418 | 11,47 | 429 | 1364 | 7,2 | 854 | 1289 | 1,33 | 1739 |
| gr96 | 55209 | 114551 | 107 | 3403 | 94334 | 70,86 | 6607 | 99339 | 79,93 | 13252 |
| kroA100 | 21282 | 49843 | 134 | 3598 | 53539 | 151 | 7145 | 56213 | 164 | 14135 |
| kroB150 | 26130 | 85858 | 228 | 8496 | 100191 | 283 | 22230 | 75728 | 189 | 34375 |
| tsp10_as | 142 | 142 | 0 | 218 | 142 | 0 | 448 | 142 | 0 | 893 |
| tsp12_as | 203 | 206 | 1,47 | 244 | 211 | 8 | 479 | 203 | 0 | 979 |
| ftv33 | 1286 | 1456 | 13,2 | 639 | 1436 | 11,6 | 1299 | 1373 | 6,76 | 2624 |
| ftv38 | 1530 | 1771 | 15,75 | 786 | 1675 | 9,47 | 1568 | 1610 | 5,22 | 3338 |
| ftv64 | 1839 | 2998 | 63 | 1681 | 2692 | 46,38 | 3401 | 2416 | 31,37 | 7977 |
| ftv70 | 1950 | 3140 | 61 | 2027 | 2895 | 48 | 3934 | 2867 | 47,02 | 8598 |
| rbg323 | 1326 | 3423 | 158 | 40183 | 3632 | 173 | 76584 | 3259 | 145,77 | 208973 |

Tabela 10: wyniki wpływu wielkości populacji



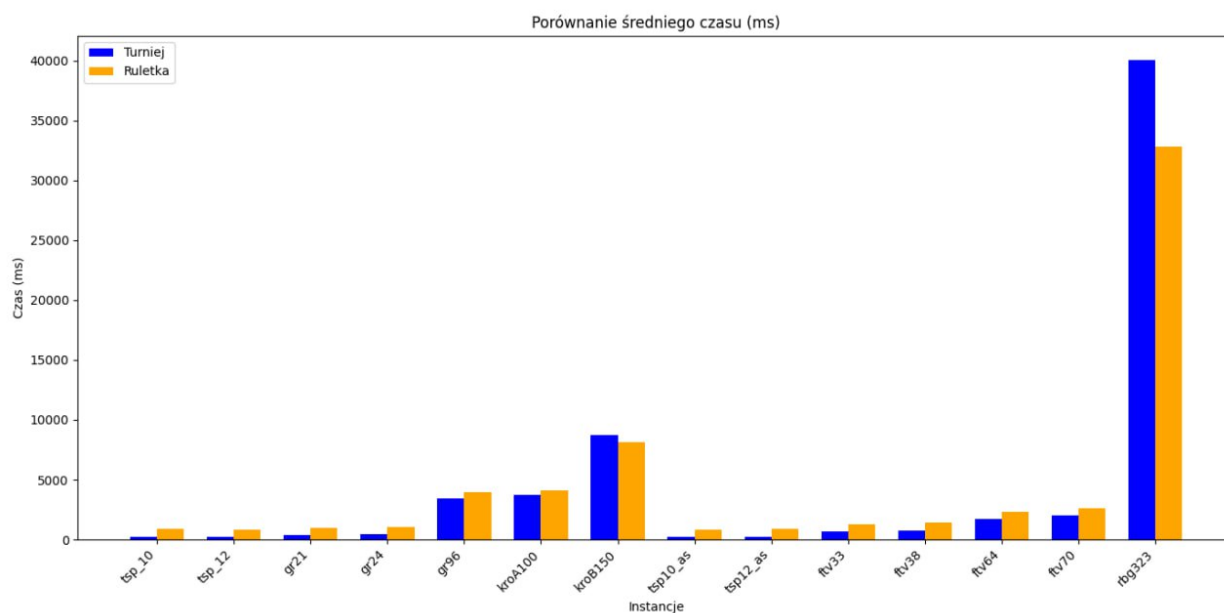
Rysunek 21: wykres zależności czasu od rozmiaru instancji przy różnych wartościach wielkości populacji



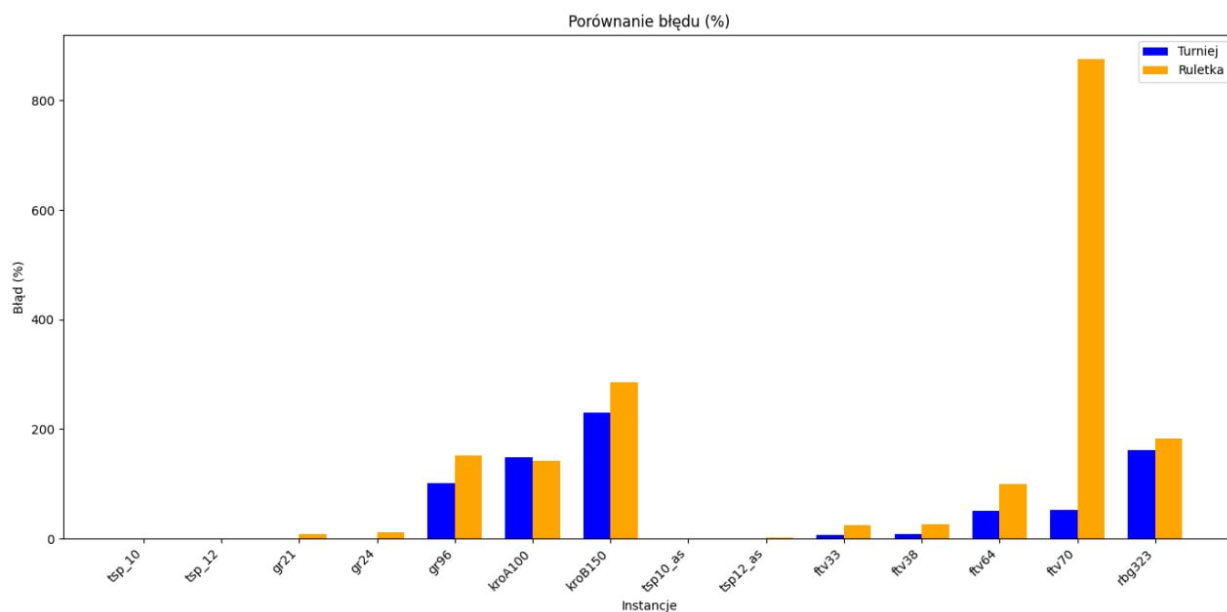
Rysunek 22: wykres zależności błędów od rozmiaru instancji przy różnych wartościach wielkości populacji

| | | Turniej | | | Ruletka | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|
| instancja | koszt optymalny | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 212 | 0 | 233 | 212 | 0 | 879 |
| tsp_12 | 264 | 264 | 0 | 255 | 264 | 0 | 846 |
| gr21 | 2707 | 2707 | 0 | 376 | 2915 | 7,68 | 972 |
| gr24 | 1272 | 1278 | 0,47 | 430 | 1415 | 11,24 | 1032 |
| gr96 | 55209 | 111667 | 102 | 3454 | 139433 | 152 | 3922 |
| kroA100 | 21282 | 52923 | 148 | 3732 | 51591 | 142 | 4142 |
| kroB150 | 26130 | 86440 | 230 | 8736 | 100865 | 286 | 8107 |
| tsp10_as | 142 | 142 | 0 | 215 | 142 | 0 | 844 |
| tsp12_as | 203 | 203 | 0 | 233 | 206 | 1,47 | 869 |
| ftv33 | 1286 | 1379 | 7,23 | 662 | 1593 | 23,8 | 1284 |
| ftv38 | 1530 | 1668 | 9,01 | 780 | 1930 | 26,14 | 1395 |
| ftv64 | 1839 | 2775 | 50,89 | 1703 | 3663 | 99,18 | 2298 |
| ftv70 | 1950 | 2961 | 51,84 | 2005 | 3641 | 876 | 2588 |
| rbg323 | 1326 | 3467 | 161 | 40065 | 3750 | 182 | 32824 |

Tabela 11: wyniki wpływu metody selekcji



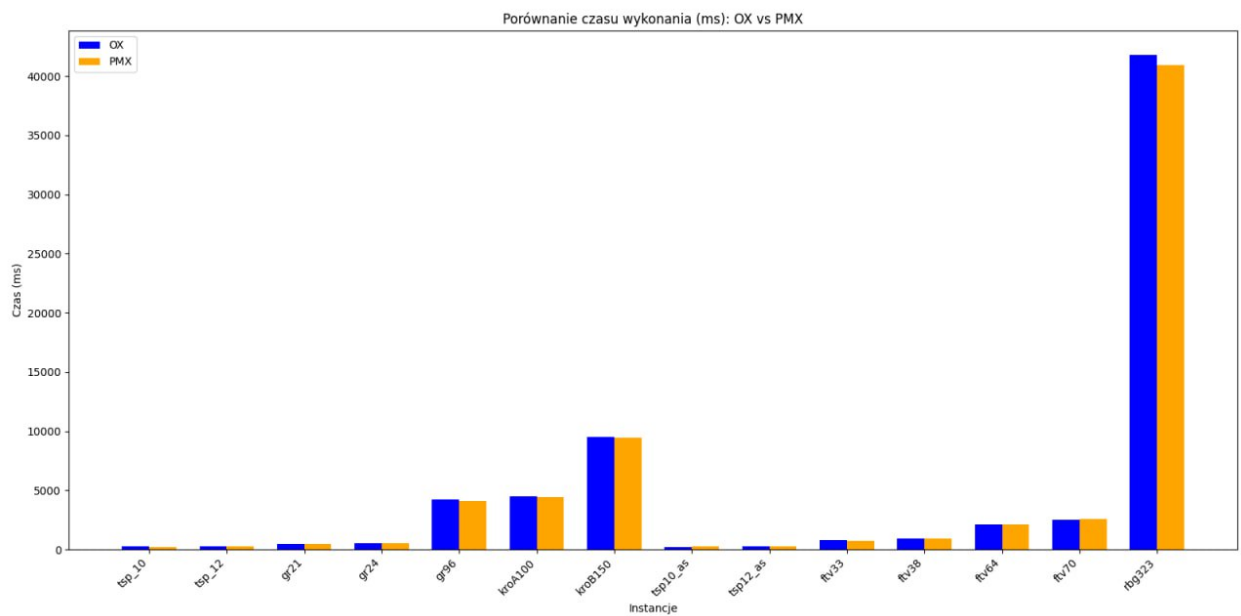
Rysunek 23: wykres zależności czasu od rozmiaru instancji przy różnych metodach selekcji



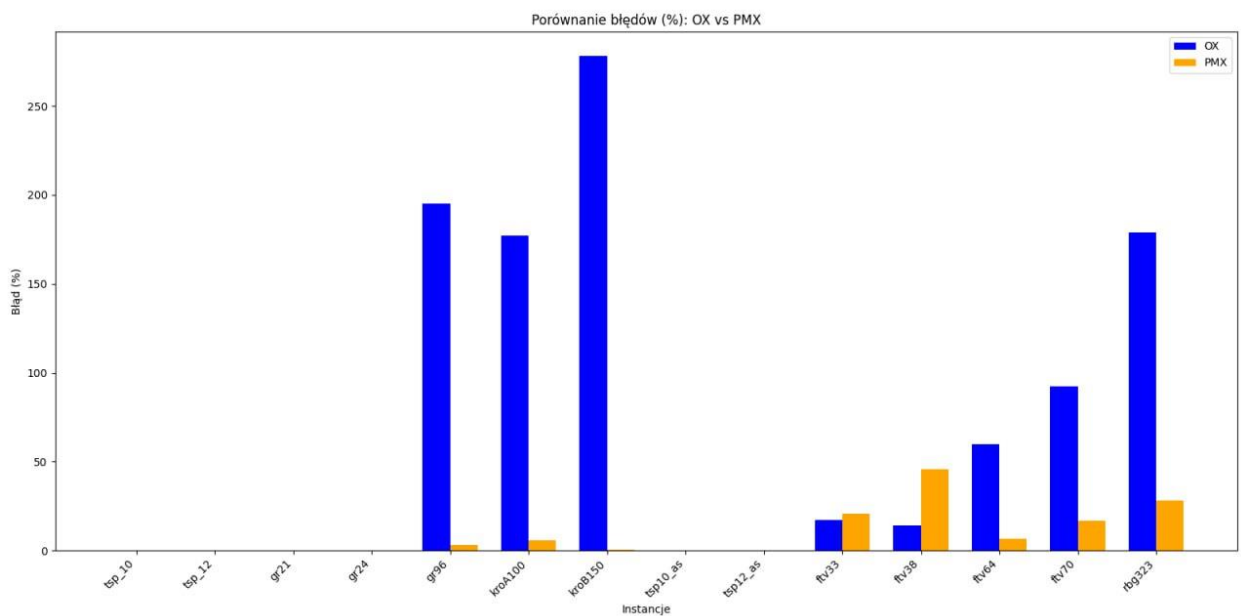
Rysunek 24: wykres zależności błędu od rozmiaru instancji przy różnych metodach selekcji

| instancja | koszt optymalny | OX | | | PMX | | |
|-----------|-----------------|--------------|-------|-------------|--------------|-------|-------------|
| | | średni koszt | błąd | czas średni | średni koszt | błąd | czas średni |
| tsp_10 | 212 | 212 | 0 | 247 | 212 | 0 | 235 |
| tsp_12 | 264 | 264 | 0 | 268 | 264 | 0 | 257 |
| gr21 | 2707 | 2707 | 0 | 457 | 2707 | 0 | 448 |
| gr24 | 1272 | 1272 | 0 | 516 | 1272 | 0 | 505 |
| gr96 | 55209 | 163335 | 195 | 4206 | 56847 | 2,96 | 4108 |
| kroA100 | 21282 | 59085 | 177 | 4521 | 22520 | 5,81 | 4431 |
| kroB150 | 26130 | 98939 | 278 | 9518 | 26217 | 0,33 | 9421 |
| tsp10_as | 142 | 142 | 0 | 233 | 142 | 0 | 240 |
| tsp12_as | 203 | 203 | 0 | 258 | 203 | 0 | 247 |
| ftv33 | 1286 | 1508 | 17,26 | 774 | 1556 | 20,9 | 750 |
| ftv38 | 1530 | 1744 | 13,98 | 953 | 2230 | 45,7 | 930 |
| ftv64 | 1839 | 2937 | 59,7 | 2116 | 1960 | 6,57 | 2100 |
| ftv70 | 1950 | 3752 | 92,41 | 2503 | 2280 | 16,92 | 2590 |
| rbg323 | 1326 | 3712 | 179 | 41752 | 1698 | 28,05 | 40932 |

Tabela 12: wyniki wpływu metody krzyżowania



Rysunek 25: wykres zależności czasu od rozmiaru instancji przy różnych metodach krzyżowania



Rysunek 26: wykres zależności błędu od rozmiaru instancji przy przy różnych metodach krzyżowania

| Nazwa pliku | BF | NN | Random | BFS | DFS | LowestCost |
|-------------|---------|---------|---------|---------|--------|------------|
| tsp_4_1 | 0 | 0,00035 | 0 | 0 | 0 | 0 |
| tsp_6_1 | 1 | 0,00197 | 0,002 | 0,005 | 0,003 | 0,008 |
| tsp_10 | 2,932 | 0,045 | 1,761 | 15,683 | 0,345 | 0,6 |
| tsp_12 | 125,215 | 0,016 | 247,457 | 161,732 | 22,695 | 50,48 |
| tsp_13 | 1800 | 0,02 | 1800 | 1800 | 73,057 | 165 |
| tsp_14 | 1800 | 0,022 | 1800 | 1800 | 269,23 | 330,43 |
| tsp_101 | 1800 | 969,906 | 1800 | 1800 | 1800 | 1800 |
| tsp6_as | 0,001 | 0,007 | 0,004 | 0,004 | 0,001 | 0,009 |
| tsp8_as | 0,048 | 0,005 | 0,068 | 0,187 | 0,019 | 0,427 |
| tsp10_as | 3,105 | 0,007 | 3,89 | 13,837 | 0,149 | 0,8 |
| tsp12_as | 122,87 | 0,011 | 255,63 | 244,486 | 1,83 | 5,2 |
| tsp13_as | 1800 | 0,012 | 1800 | 1800 | 2,142 | 4,06 |
| tsp14_as | 1800 | 0,012 | 1800 | 1800 | 9,87 | 10,90 |

Tabela 9: wyniki czasy w sekundach dla algorytmów z projektów 1 i 2

6. Analiza wyników i wnioski

Na podstawie analizy wyników eksperymentów z algorytmami Tabu Search i Algorytmami Genetycznymi w odniesieniu do problemu komiwojażera (TSP), można sformułować następujące wnioski:

Efektywność algorytmu Tabu Search

1. **Skuteczność parametrów:**
 - **Rozmiar listy tabu:** Zwiększenie listy tabu poprawiło zdolność algorytmu do unikania minimów lokalnych, co szczególnie uwidoczniło się w instancjach średniej wielkości (np. 150 miast), jednak prowadziło do wydłużenia czasu działania w większych problemach.
 - **Kadencja tabu:** Optymalną wartością okazała się kadencja $20\% \times n$, co pozwalało na utrzymanie równowagi między dokładnością a czasem obliczeń.
 - **Kryterium aspiracji i aspiracja plus:** Kryterium aspiracji znacząco zmniejszało błędy wynikowe, umożliwiając akceptowanie rozwiązań znajdujących się na liście tabu, jeśli były wystarczająco dobre. Dodatkowe rozszerzenie przez aspirację plus było szczególnie efektywne dla instancji asymetrycznych i większych problemów.
2. **Porównanie z wynikami algorytmów zadań 1 i 2 (Brute Force i Branch and Bound):**
 - W porównaniu z algorytmem brute force, Tabu Search był zdecydowanie szybszy w instancjach większych niż 50 miast, jednak kosztem wyższego błędu wynikowego.
 - Dla małych instancji ($n < 25$) algorytmy brute force oraz podział i ograniczenia były dokładniejsze (0% błędu), natomiast Tabu Search osiągał błędy na poziomie kilku procent.
3. **Rodzaj instancji:**
 - Algorytm lepiej radził sobie z instancjami symetrycznymi, w których błędy były niższe i czas działania krótszy w porównaniu do asymetrycznych.
4. **Granice możliwości:**
 - Tabu Search działał efektywnie w instancjach do około 300 wierzchołków, z błędem na poziomie 20%. Dla większych problemów czas działania stawał się nieproporcjonalnie długi.

Efektywność Algorytmów Genetycznych

1. **Wpływ parametrów:**
 - **Rozmiar populacji:** Zwiększenie populacji do 1000 osobników poprawiło wyniki dla instancji większych niż 100 miast, ale kosztem wydłużenia czasu działania. Dla mniejszych instancji populacja 500 była wystarczająca.
 - **Krzyżowanie i mutacja:** Prawdopodobieństwo krzyżowania na poziomie 90% oraz mutacji na poziomie 10% zapewniło najlepsze wyniki pod względem jakości i różnorodności rozwiązań.
 - **Metoda selekcji:** Selekcja turniejowa dawała lepsze wyniki jakościowe, szczególnie w dużych instancjach, choć była bardziej czasochłonna niż selekcja ruletkowa.

- **Krzyżowanie PMX:** Było bardziej efektywne w zachowaniu strukturalnych zależności rozwiązań w dużych problemach ($n > 150$), szczególnie w asymetrycznych instancjach.
- 2. **Porównanie z algorytmami zadań 1 i 2:**
 - Algorytmy genetyczne były znacznie szybsze niż brute force i Branch and Bound w dużych instancjach ($n > 50$), gdzie algorytmy przeglądu zupełnego stają się niepraktyczne.
 - Dla małych instancji ($n < 25$) algorytmy genetyczne były mniej efektywne czasowo i osiągały podobne wyniki do brute force.
- 3. **Rodzaj instancji:**
 - Algorytmy genetyczne wykazywały większą elastyczność niż Tabu Search w instancjach asymetrycznych, gdzie lepiej radziły sobie z eksploracją bardziej złożonej przestrzeni rozwiązań.
- 4. **Granice możliwości:**
 - Algorytmy genetyczne skutecznie rozwiązywały instancje do 100 miast, z błędami mieszczącymi się w zadanych granicach.

Porównanie Tabu Search i Algorytmów Genetycznych

1. **Czas działania:**
 - Tabu Search był szybszy w mniejszych instancjach ($n \leq 100$), co czyniło go bardziej odpowiednim do szybkiego rozwiązywania problemów średniej wielkości.
 - Algorytmy genetyczne lepiej radziły sobie z bardzo dużymi instancjami ($n > 300$), gdzie czas działania Tabu Search gwałtownie wzrastał.
2. **Jakość wyników:**
 - Tabu Search osiągał mniejsze błędy dla symetrycznych i mniejszych instancji, ale algorytmy genetyczne były bardziej skuteczne w asymetrycznych i dużych problemach, szczególnie przy optymalnych ustawieniach populacji i mutacji.
3. **Zastosowanie w praktyce:**
 - Tabu Search jest idealny do problemów, w których kluczowa jest szybkość działania przy akceptowalnym poziomie błędu.
 - Algorytmy genetyczne sprawdzają się lepiej w przypadkach, gdzie priorytetem jest dokładność, a czas działania jest mniej istotny.

Podsumowanie i rekomendacje

1. **Potwierdzenie hipotez:** Badania potwierdziły, że oba algorytmy są efektywne w rozwiązywaniu problemu TSP, ale ich skuteczność różni się w zależności od wielkości i rodzaju instancji.
2. **Optymalizacja:** Zarówno Tabu Search, jak i algorytmy genetyczne mogą być jeszcze bardziej zoptymalizowane przez dostosowanie parametrów do specyfiki instancji. W szczególności Algorytmy genetyczny, które wykazali bardzo mocną czujność na rozmiar instancji. Badania pokazały, że jeden zestaw parametrów już może nie być korzystny do innego większego bądź mniejszego. I zależy tu może może więcej niż na jednej wartości i jednej nie optymalnej wartości, któregoś etapów postępu algorytmu.

Wyniki badawcze wskazują, że oba algorytmy mogą być efektywnie stosowane w różnych scenariuszach problemu komiwojagera, co czyni je wartościowym narzędziem w rozwiązywaniu złożonych problemów optymalizacyjnych.