**ALGORYTMY METAHEURYSTYCZNE – SPRAWOZDANIE**

**ETAP 2**

**MIKOŁAJ JANUSZ**

**ŁUKASZ GOLICZEWSKI**

1. **ZŁOŻONOŚCI ALGORYTMÓW**

|  |  |  |
| --- | --- | --- |
|  | TS | TS dla ruchu typu invert dla niesymetrycznych instancji |
| czasowa |  |  |
| pamięciowa |  |  |

Tabela 1. Złożoności obliczeniowe zaimplementowanych algorytmów.

Oznaczenia:

1. – maksymalna liczba iteracji algorytmu Tabu Search
2. – długość listy tabu

Pozostałe parametry TS nie wpływają na złożoności algorytmów.

Uwagi:

1. Złożoność czasowa uwzględnia parametry oraz , ponieważ podstawową iterację algorytmu wykonujemy razy, a w każdej z iteracji po wejściu do podwójnej pętli for musimy przejść po całej liście tabu. W przypadku niesymetrycznej instancji dla ruchu typu invert musimy dodatkowo przejść po wszystkich wierzchołkach przy liczeniu funkcji celu, dlatego złożoności czasowe się różnią.
2. Złożoność pamięciowa uwzględnia wszystkie struktury danych, które zostały użyte do rozwiązania problemu poprzez zadany algorytm. Poza złożonością należy uwzględnić listę tabu, która może rozmiarem przekraczać pamięć zarezerwowaną dla wszystkich pozostałych struktur danych, jednak złożoność pamięciowa całego algorytmu będzie .
3. **SZCZEGÓŁY SPRZĘTOWE I IMPLEMENTACYJNE**
4. JĘZYK PROGRAMOWANIA : Java (cały projekt realizowany w Mavenie)
5. KOMPILATOR : javac (kompliator domyślny w Intelij Idea Ultimate, z którego korzystaliśmy podczas realizacji zadań)
6. METODA KOMPILACJI: domyślna kompilacja i budowanie projektu korzystając z funkcji wbudowanych w IDE
7. SYSTEM OPERACYJNY: Windows 10
8. PROCESOR: AMD Ryzen 5 4500U with Radeon Graphics (6 rdzeni fizycznych, 6 rdzeni logicznych
9. TAKTOWANIE PROCESORA: 2,3 GH
10. RAM – 7,4 GB dostępne (domyślnie 8 GB)
11. **OPIS INSTANCJI**

Podczas badań korzystaliśmy z kilku rodzajów instancji.

1. Instancje losowe

Instancje losowe generowane są poprzez losowy wybór punktów z płaszczyzny 2D wraz z ograniczeniami górnymi dla obu współrzędnych. Obie współrzędne są wybierane zgodnie z rozkładem jednostajnym.

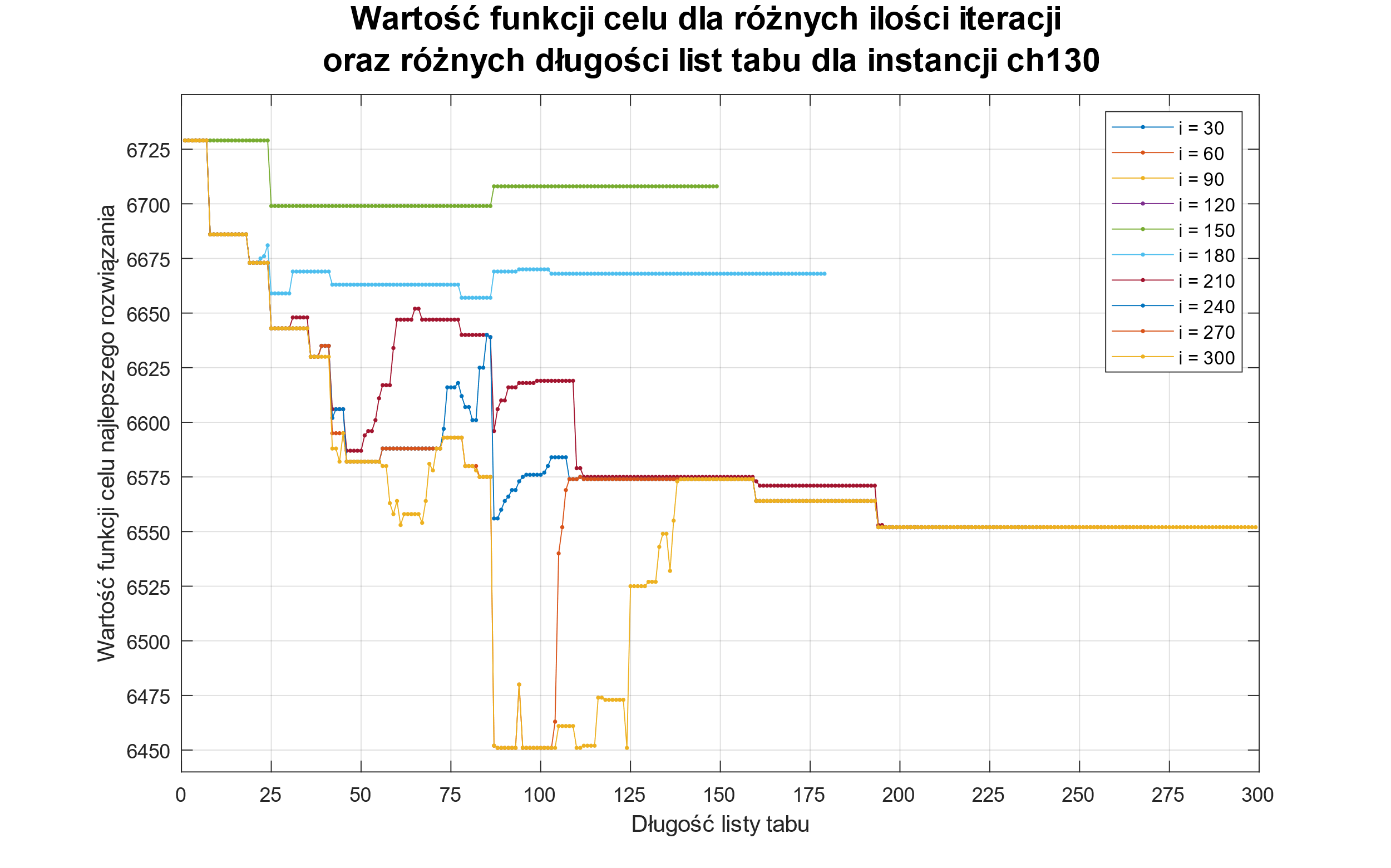
1. Instancje dostępne na stronie TSPLIB

TSPLIB posiada szereg instancji, dla których znaleziono rozwiązania optymalne (albo bardzo bliskie rozwiązaniu optymalnemu). Niektóre z instancji są dobrze znanymi problemami, a niektóre są przykładami bazującymi na rzeczywistych odległościach i współrzędnych punktów geograficznych. Podczas badań wykorzystaliśmy następujące instancje:

|  |  |
| --- | --- |
| Nazwa | Typ |
| eil76 | EUC\_2D, |
| kroA100 | EUC\_2D |
| Eil101 | EUC\_2D |
| Lin105 | EUC\_2D |
| Pr136 | EUC\_2D |
| Pr144 | EUC\_2D |
| Pr152 | EUC\_2D |
| Rat195 | EUC\_2D |
| Tsp225 | EUC\_2D |
| Pr226 | EUC\_2D |
| Rbg323 | FULL\_MATRIX, ASYMMETRIC |
| Bays29 | GEO |
| Ry48p | FULL MATRIX, ASYMMETRIC |
| St70 | EUC\_2D |
| A280 | EUC\_2D |
| Berlin52 | EUC\_2D |
| Bier127 | EUC\_2D |
| Br17 | FULL MATRIX |
| Brd14051 | EUC\_2D |
| Ch130 | EUC\_2D |
| Ch150 | EUC\_2D |
| D198 | EUC\_2D |
| D493 | EUC\_2D |
| D657 | EUC\_2D |
| Eil51 | EUC\_2D |
| Eil76 | EUC\_2D |
| Ftv70 | FULL MATRIX, ASYMMETRIC |
| Kro124p | FULL MATRIX, ASYMMETRIC |
| P43 | FULL MATRIX, ASYMMETRIC |
| kroA200 | EUC\_2D |
| Dantzig42 | EUC\_2D |
| Ftv44 | FULL MATRIX |
| Lin105 | EUC\_2D |

Tabela 2. Wykaz instancji z TSPLIB użytych w eksperymentach.

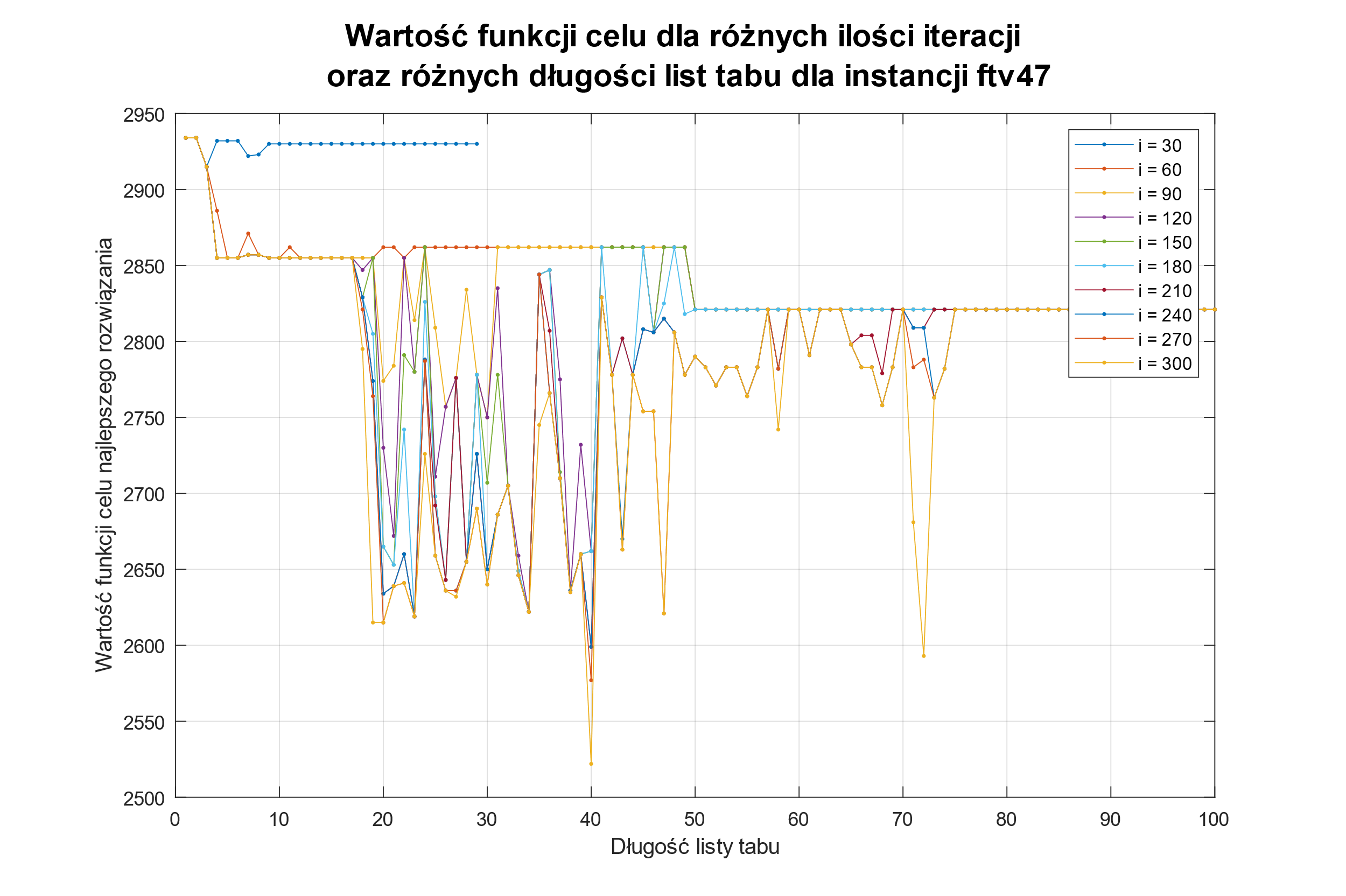
1. **EKSPERYMENTY**
2. **Typ ruchu – invert/insert/swap**
3. **Rozwiązanie startowe**
4. **Kick**
5. Badanie wpływu długości listy tabu oraz maksymalnej liczby iteracji na jakość rozwiązania
   1. CEL I METODOLOGIA: Dla liczby iteracji badamy wpływ długości listy tabu na jakość rozwiązania wyznaczając rozwiązanie dla długości listy .
   2. OPIS REZULTATÓW:



Wykres 1. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
w zależności od długości listy tabu dla różnych liczb iteracji algorytmu – instancja ch130.tsp

Na wykresie przedstawiono fragment danych, których całość dostępna jest w osobnym pliku dołączonym do sprawozdania. Prawie we wszystkich przypadkach (dla sensownej liczby iteracji oraz długości listy w zależności od wielkości instancji) zauważalne są poprawy jakości najlepszego rozwiązania w pewnym przedziale. Przedział wyraźnych popraw jakości najlepszego rozwiązania jest różny w zależności od przyjmowanych parametrów, jednak nie znajduje się on nigdy w pobliżu największych ani najmniejszych rozważanych długości list tabu. Warty odnotowania jest także wspólny skok wykonywany przez algorytmy puszczane dla liczby iteracji oraz (okolice długości listy tabu 85 – 105), który umożliwia znalezienie w obu przypadkach tego samego najlepszego rozwiązania. Możliwą przyczyną takiej sytuacji może być znalezienie dobrego rozwiązania już dla 270 iteracji i brak poprawy

sytuacji dla kolejnych 30 iteracji. Brak zmian wartości funkcji celu dla długości list tabu zbliżonych do liczby wykonywanych iteracji uzasadnić może fakt, iż na liście tabu znajduje się zbyt wiele elementów, aby znaleźć nowe rozwiązania, które poprawiają wartość funkcji celu.



Wykres 2. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
w zależności od długości listy tabu dla różnych liczb iteracji algorytmu – instancja ftv47.atsp

Na wykresie przedstawiono fragment danych, których całość dostępna jest w osobnym pliku dołączonym do sprawozdania. Tak jak na poprzednim wykresie, najlepsze wartości funkcji celu udaje się znaleźć w pewnym przedziale (w tym przypadku jest to przedział ). Interesujący jest pik w długości listy tabu , gdzie wiele z testów wyznacza najlepsze rozwiązanie dla zadanej liczby iteracji.

Elementem wspólnym obu wykresów jest stopniowa poprawa jakości rozwiązań dla kolejnych długości list tabu wraz ze wzrostem liczby iteracji. Wydaje się jednak, że od pewnej liczby iteracji wartość najlepszego rozwiązania nie będzie ulegać gwałtownej poprawie.

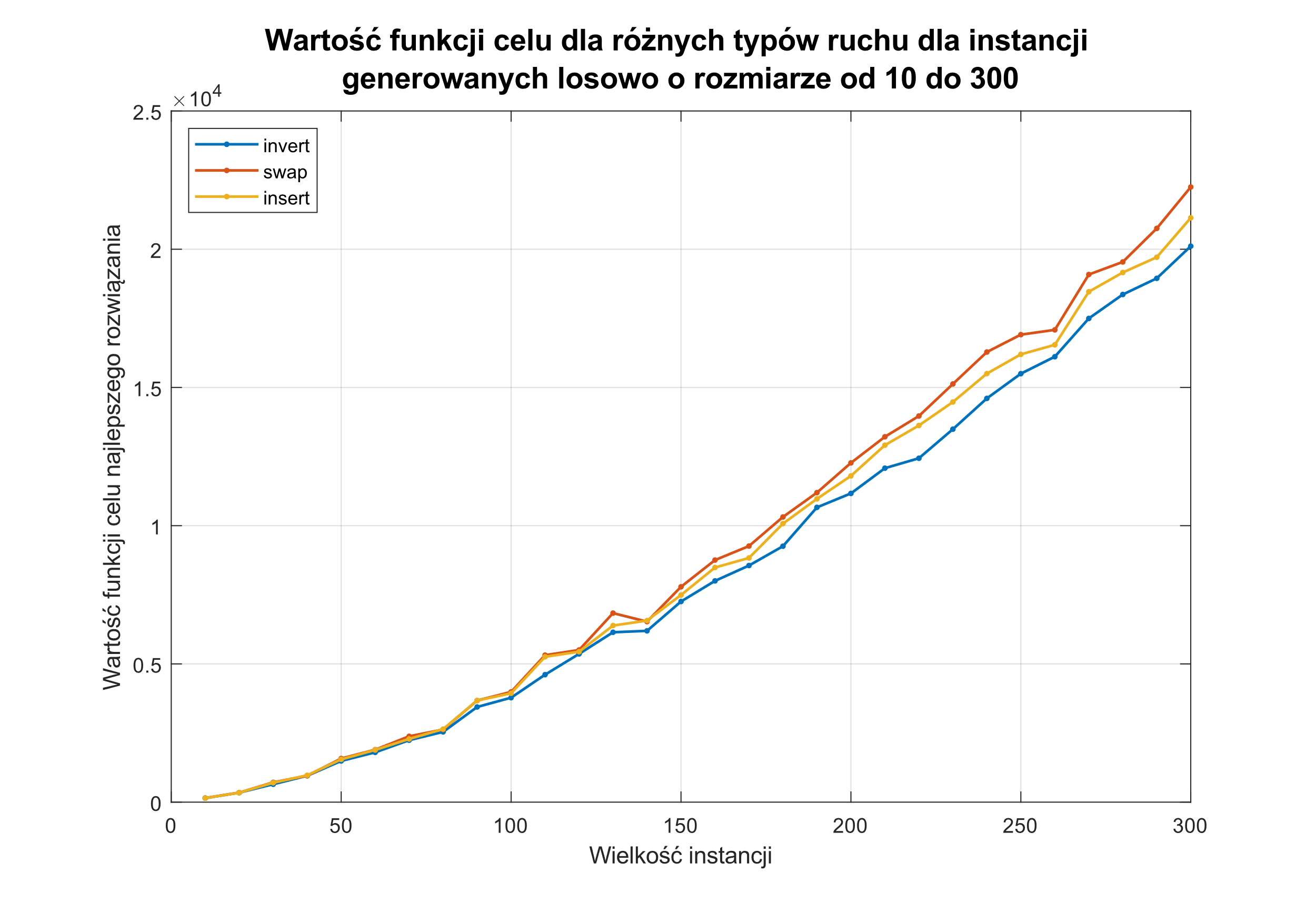
* 1. WNIOSKI:

Na podstawie przeprowadzonych eksperymentów nie da się wyznaczyć dokładnych wartości parametrów oraz , dla których wartość funkcji celu jest najmniejsza, jednak bez wątpienia oba parametry mają na to wpływ. Można stwierdzić, że rozważanie algorytmu Tabu Search dla TSP jest bezcelowe dla krótkich list tabu i małych liczb iteracji algorytmu.

1. Badanie wpływu typu ruchu wykorzystywanego w algorytmie TS na jakość rozwiązania.
   1. CEL I METODOLOGIA: Celem eksperymentu jest rozstrzygnięcie, czy wybór typu ruchu spośród następujących: invert, swap, insert ma wpływ na jakość rozwiązania zwracanego przez algorytm Tabu Search. Testy przeprowadzono dla losowo generowanych instancji o wielkościach ze zbioru {10, 20, …, 300}. Długość listy tabu została ustawiona na wartość 50. Eksperymenty zostały wykonane dla 100 iteracji algorytmu. Dla każdego z ruchów w celu zminimalizowania czasu działania zaimplementowano akcelerację przy wyliczaniu wartości funkcji celu.
   2. OPIS REZULTATÓW: Poniżej przedstawiony tabelę oraz wykres prezentujące zależność między typem ruchu a wartościami funkcji celu zwracanymi przez algorytm TS.

|  |  |  |  |
| --- | --- | --- | --- |
| rozmiar instancji | Tabu Search - invert | Tabu Search - swap | Tabu Search - insert |
| 10 | **146** | 148 | **146** |
| 20 | **344** | **344** | **344** |
| 30 | **647** | 717 | 701 |
| 40 | **955** | 955 | 972 |
| 50 | **1490** | 1578 | 1547 |
| 60 | **1801** | 1897 | 1887 |
| 70 | **2241** | 2382 | 2286 |
| 80 | **2541** | 2627 | 2643 |
| 90 | **3442** | 3675 | 3674 |
| 100 | **3778** | 3987 | 3937 |
| 110 | **4611** | 5317 | 5257 |
| 120 | **5362** | 5503 | 5442 |
| 130 | **6145** | 6837 | 6388 |
| 140 | **6196** | 6530 | 6567 |
| 150 | **7257** | 7789 | 7492 |
| 160 | **8003** | 8756 | 8487 |
| 170 | **8558** | 9263 | 8833 |
| 180 | **9258** | 10315 | 10075 |
| 190 | **10662** | 11196 | 10970 |
| 200 | **11167** | 12271 | 11801 |
| 210 | **12080** | 13216 | 12912 |
| 220 | **12439** | 13966 | 13624 |
| 230 | **13490** | 15125 | 14473 |
| 240 | **14604** | 16281 | 15501 |
| 250 | **15498** | 16910 | 16197 |
| 260 | **16112** | 17083 | 16542 |
| 270 | **17496** | 19085 | 18462 |
| 280 | **18362** | 19540 | 19158 |
| 290 | **18947** | 20753 | 19710 |
| 300 | **20110** | 22249 | 21135 |

Tabela 3. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
w zależności od wielkości instancji dla różnych typów ruchu – instancje losowe



Wykres 3. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
w zależności od wielkości instancji dla różnych typów ruchu – instancje losowe

Spośród ruchów: invert, swap oraz insert najniższe wartości funkcji celu zwracał algorytm Tabu Search z ruchem invert. Nieco większe wartości zwracał Tabu Search z ruchem insert, a największe ten sam algorytm z ruchem swap. Dla małych instancji różnice te są bardzo nieznaczne lub nawet równe 0. Wraz z wzrostem wielkości instancji powoli te różnice rosną.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Porównywane warianty algorytmu TS  z danym typem ruchu | Suma rang dodatnich | Suma rang ujemnych | W-value | p-value |
| Invert - Swap | 0 | 406 | 0 | < 0.00001 |
| Invert - Insert | 0 | 406 | 0 | < 0.00001 |
| Swap - Insert | 416.5 | 18.5 | 18.5 | < 0.00001 |

Tabela 4. Wyniki testu Wilcoxona dla 30 instancji losowych.

Każdy z testów Wilcoxona pozwala odrzucić hipotezę H0 zakładającą brak istotnej różnicy w jakości porównywanych wariantów. Wobec tego warianty te w znaczący sposób różnią się od siebie.

* 1. WNIOSKI:

Na podstawie testów Wilcoxona wykonanych dla wyników przeprowadzonych eksperymentów warianty algorytmu Tabu Search istotnie różnią się od siebie. Łącząc to z danymi w tabeli oraz wykresem można stwierdzić, że najlepszym okazał się Tabu Search z ruchem invert. Drugi pod względem jakości wyników jest Tabu Search używający inserta, a najgorzej działał ten bazujący na ruchu swap.

1. Badanie wpływu wyboru rozwiązania początkowego na czas potrzebny do zakończenia działania algorytmu 2-OPT.
   1. CEL I METODOLOGIA: Celem eksperymentu jest rozstrzygnięcie, czy wybór rozwiązania początkowego ma wpływ na czas potrzebny do zakończenia działania algorytmu 2-OPT. Dla różnych instancji z TSPLIB o różnych wielkościach i typach mierzymy czasy potrzebne do zakończenia działania algorytmu 2-OPT, rozpoczynając poszukiwanie rozwiązania z rozwiązania losowego oraz z wyniku działania rozszerzonego algorytmu najbliższego sąsiada. To samo wykonujemy dla losowych instancji. Dla każdej z instancji porównujemy otrzymane wyniki i sprawdzamy na ile istotne są różnice między nimi.
   2. OPIS REZULTATÓW:

Poniższy wykres przedstawia zależność między wielkością instancji a czasami potrzebnymi do zakończenia działania algorytmu 2opt startującego z losowego rozwiązania oraz rozwiązania wyznaczonego przez algorytm ext. NN dla instancji TSPLIB.

Dla każdej z instancji końcowa funkcja celu dla rozwiązania początkowego zadanego ext. NN jest wyraźnie mniejsza niż funkcja celu dla losowego rozwiązania początowego.

Wraz ze wzrostem liczby miast uwidaczniają się tendencje wzrostu wartości funkcji celu dla obu typów rozwiązania początkowego.

Poniższy wykres przedstawia zależność między wielkością instancji a czasami potrzebnymi do zakończenia działania algorytmu 2opt startującego z losowego rozwiązania oraz rozwiązania wyznaczonego przez algorytm ext. NN dla instancji losowych.

**ZBADAJ TO**

**WSTAW WYKRES**

**OPISZ WYKRES**

* 1. WNIOSKI:

Z obu wykresów jasno wynika, że czas potrzebny do zakończenia działania algorytmu 2-opt jest dużo mniejszy, gdy rozpoczynamy z rozwiązania wyznaczonego wcześniej przez ext. NN. Wynikać to może z tego, że 2opt dla lepszego rozwiązania startowego potrzebuje mniej iteracji, aby nie móc już dalej znaleźć żadnej poprawy.

1. Badanie jakości najlepszego rozwiązania w zależności od czasu działania algorytmów 2-OPT, ext. NN oraz k-Random.
   1. CEL I METODOLOGIA: Celem eksperymentu jest zbadanie, czy wyżej wymienione algorytmy da się sensownie porównać pod względem jakości rozwiązania dla określonych czasów działania. Badamy jakość rozwiązania w ściśle określonych czasach działania każdego z algorytmów dla wybranej instancji. Porównujemy ze sobą rozwiązania otrzymane dla różnych algorytmów.
   2. OPIS REZULTATÓW:

Poniższy wykres przedstawia wartości funkcji celu najlepszego rozwiązania w zależności od czasów zadanych algorytmom:

**OPISZ WYKRES**

**WYCIĄGNIJ WNIOSKI**

**KONIEC**