**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 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 (kompilator 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.

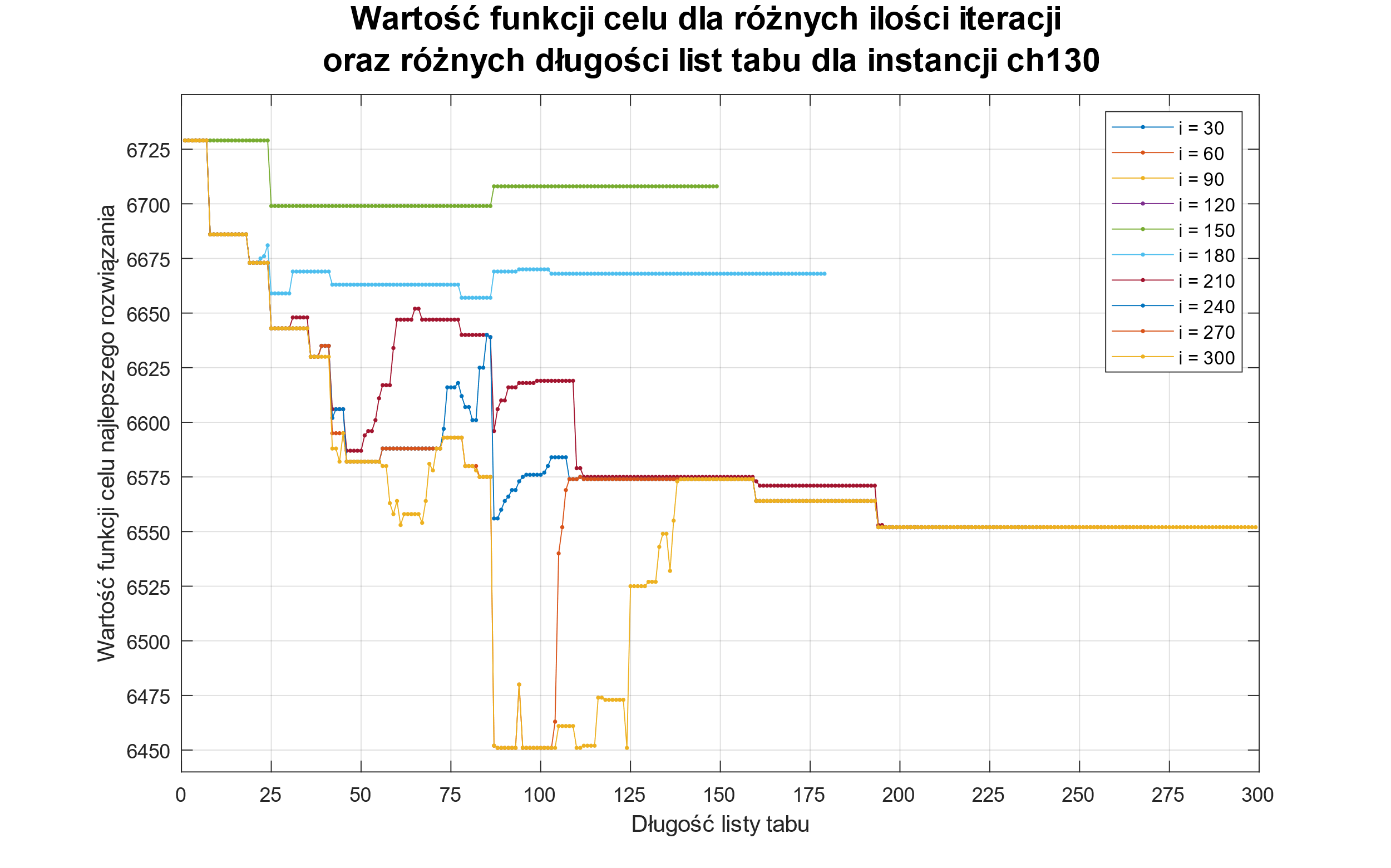
1. Instancje dostępne na stronie TSPLIB

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 |
| Ft53 | FULL\_MATRIX, ASYMMETRIC |
| Ch130 | EUC\_2D |
| Ftv47 | FULL MATRIX, ASYMMETRIC |

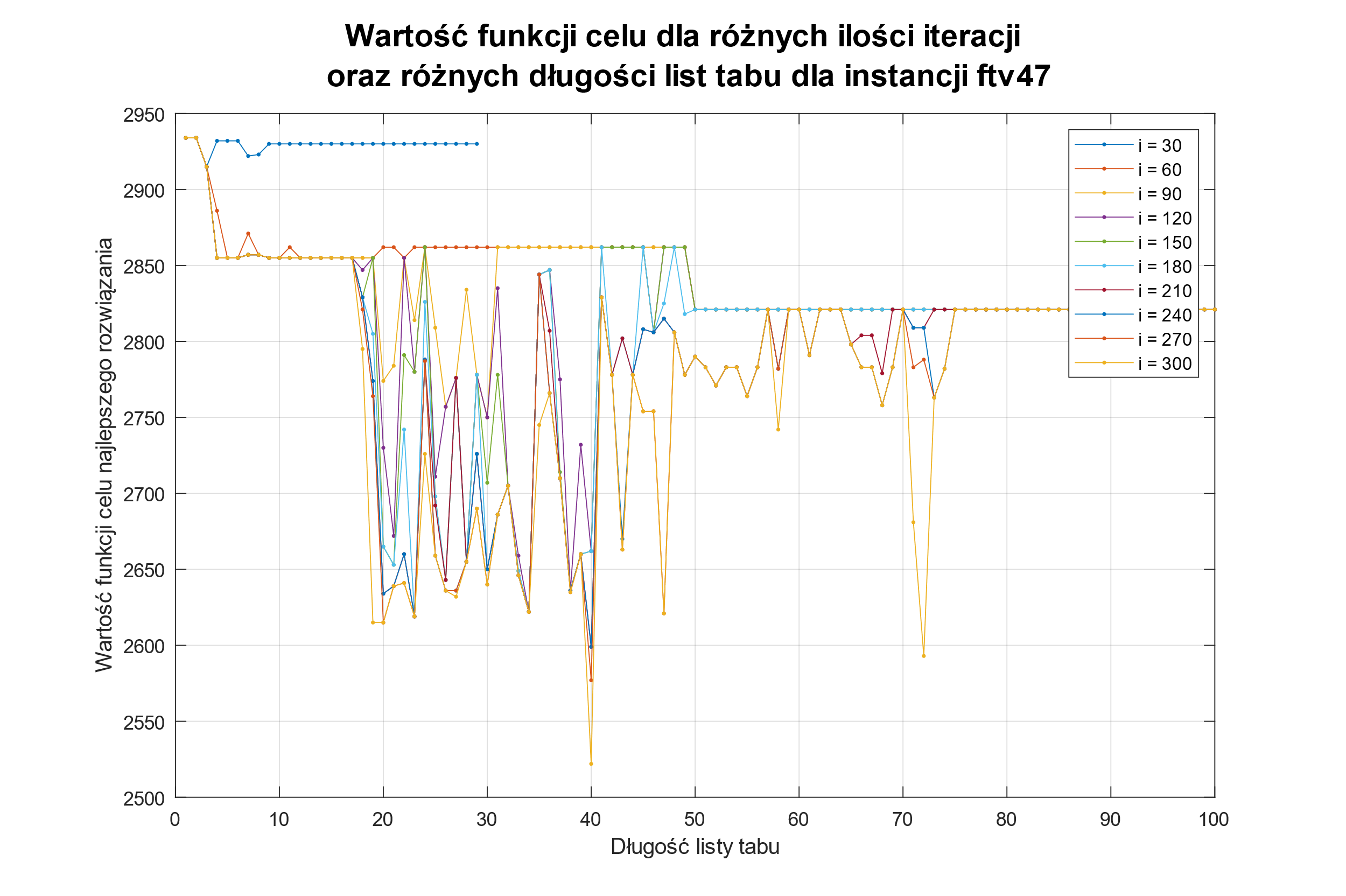
Tabela 2. Wykaz instancji z TSPLIB użytych w eksperymentach.

1. **EKSPERYMENTY**
2. 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: Poniżej przedstawiono wykres prezentujący zależność między długością listy tabu a wartościami funkcji celu zwracanymi przez algorytm TS dla zróżnicowanej liczby iteracji algorytmu.



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

Na wykresie 1. 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 listy 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 uzasadniać 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   
od długości listy tabu dla różnych liczb iteracji algorytmu – instancja ftv47.atsp

Na wykresie 2. przedstawiono fragment danych, których całość dostępna jest w osobnym pliku dołączonym do sprawozdania. Tak jak na wykresie 1., 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.

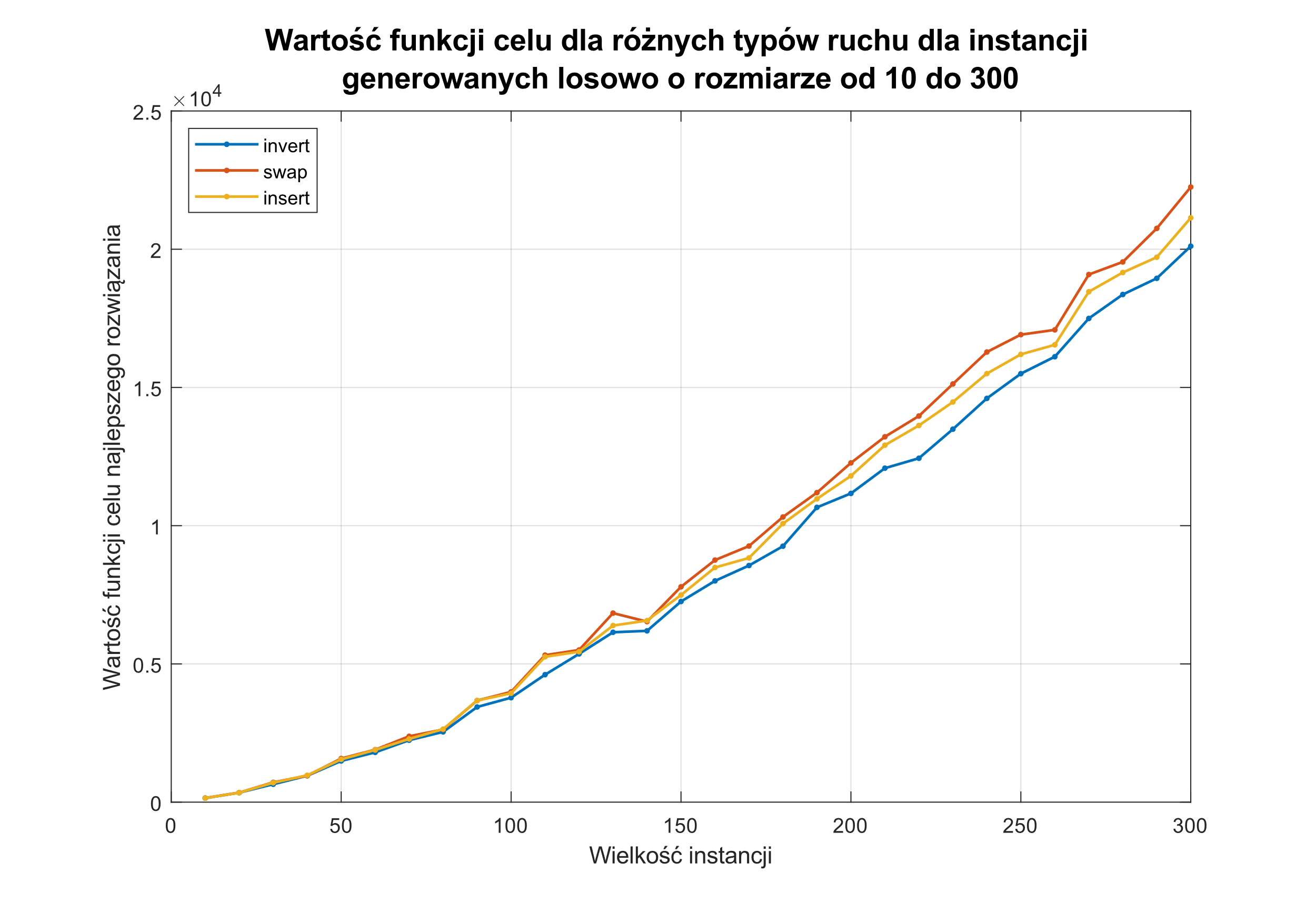
* 1. WNIOSKI:

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 oraz 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 przedstawiono tabelę oraz wykres prezentujące zależność między rozmiarem instancji a wartościami funkcji celu zwracanymi przez algorytm TS dla różnych typów ruchu.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Wartości funkcji celu | | |
| 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   
od wielkości instancji dla różnych typów ruchu – instancje losowe.



Wykres 3. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
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.

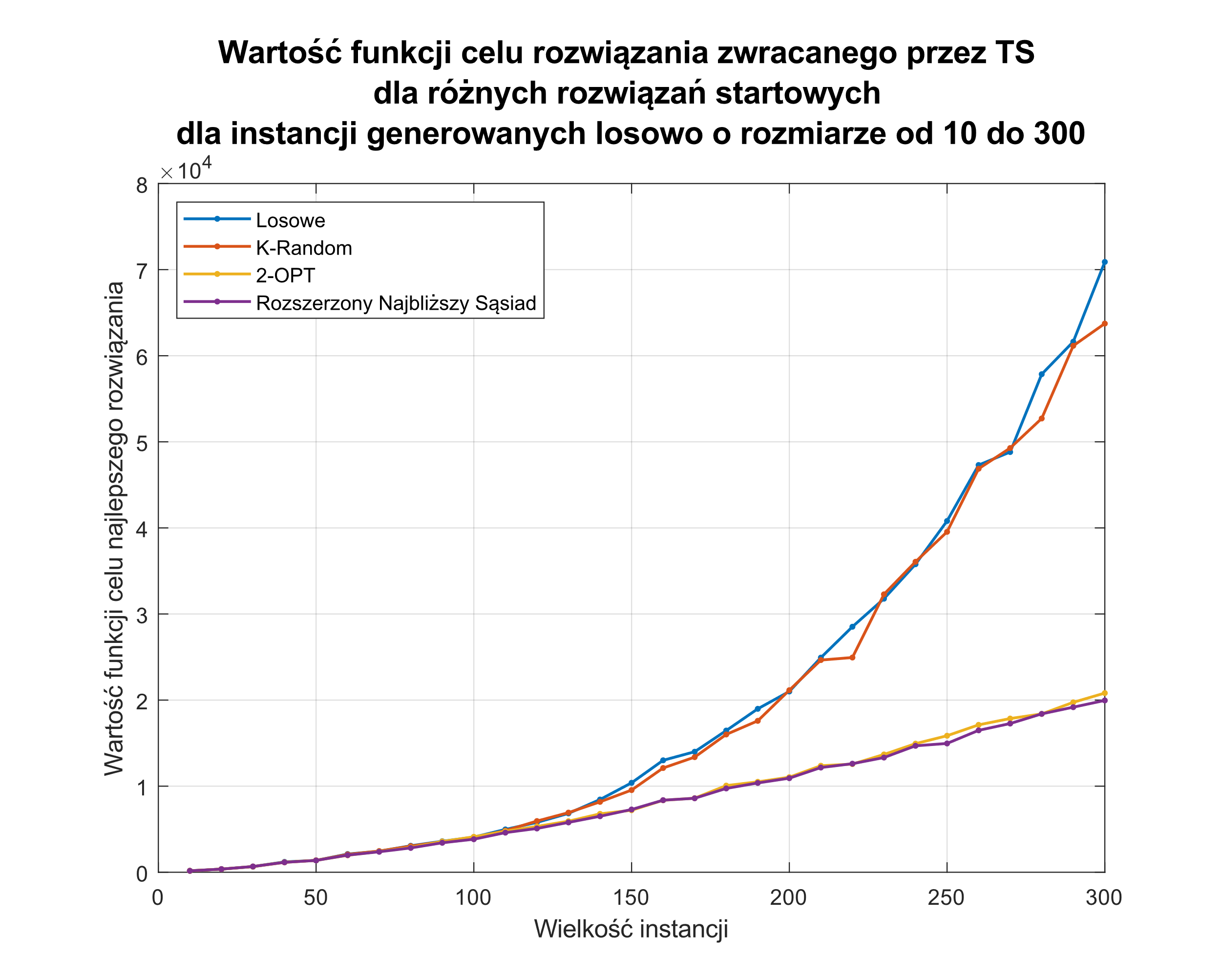
* 1. WNIOSKI:

Na podstawie testów Wilcoxona wykonanych dla wyników przeprowadzonych eksperymentów warianty algorytmu Tabu Search istotnie różnią się od siebie. Łącząc to z danymi w tabeli 3. oraz wykresem 3. 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 jakość rozwiązania zwracanego przez algorytm Tabu Search.
   1. CEL I METODOLOGIA: Celem eksperymentu jest rozstrzygnięcie, czy wybór rozwiązania początkowego ma wpływ na jakość rozwiązania zwracanego przez algorytm Tabu Search. Porównujemy następujące rozwiązania startowe: losowe, zwracane przez k-Random (dla k = 100), zwracane przez algorytm 2-OPT i zwracane przez rozszerzony algorytm najbliższego sąsiada. 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.
   2. OPIS REZULTATÓW: Poniżej przedstawiono tabelę oraz wykres prezentujące zależność między rozwiązaniem startowym a wartością funkcji celu rozwiązania zwracanego przez algorytm TS.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Wartości funkcji celu | | | |
| rozmiar instancji | losowe | k-Random | 2-OPT | ext. NN |
| 10 | **167** | **167** | **167** | **167** |
| 20 | **364** | **364** | **364** | **364** |
| 30 | **659** | **659** | 663 | 667 |
| 40 | 1205 | **1149** | 1177 | 1158 |
| 50 | 1375 | 1381 | 1381 | **1372** |
| 60 | 2115 | 2076 | **1975** | 1988 |
| 70 | 2445 | 2465 | 2411 | **2377** |
| 80 | 3073 | 3035 | 2885 | **2825** |
| 90 | 3600 | 3555 | 3576 | **3422** |
| 100 | 4079 | 4050 | 4117 | **3835** |
| 110 | 4974 | 4836 | 4758 | **4602** |
| 120 | 5788 | 5945 | 5318 | **5084** |
| 130 | 6835 | 6925 | 5931 | **5788** |
| 140 | 8447 | 8177 | 6785 | **6503** |
| 150 | 10388 | 9544 | **7202** | 7290 |
| 160 | 13003 | 12108 | **8360** | 8372 |
| 170 | 13999 | 13372 | 8604 | **8592** |
| 180 | 16458 | 16006 | 10063 | **9736** |
| 190 | 18984 | 17588 | 10504 | **10372** |
| 200 | 20999 | 21149 | 11042 | **10925** |
| 210 | 24929 | 24648 | 12372 | **12162** |
| 220 | 28518 | 24943 | **12564** | 12608 |
| 230 | 31777 | 32286 | 13681 | **13323** |
| 240 | 35779 | 36067 | 14938 | **14687** |
| 250 | 40802 | 39544 | 15858 | **14961** |
| 260 | 47302 | 46872 | 17121 | **16484** |
| 270 | 48814 | 49276 | 17854 | **17274** |
| 280 | 57855 | 52712 | **18391** | 18396 |
| 290 | 61626 | 61155 | 19738 | **19175** |
| 300 | 70895 | 63735 | 20813 | **19958** |

Tabela 5. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
od wielkości instancji dla różnych rozwiązań startowych – instancje losowe.



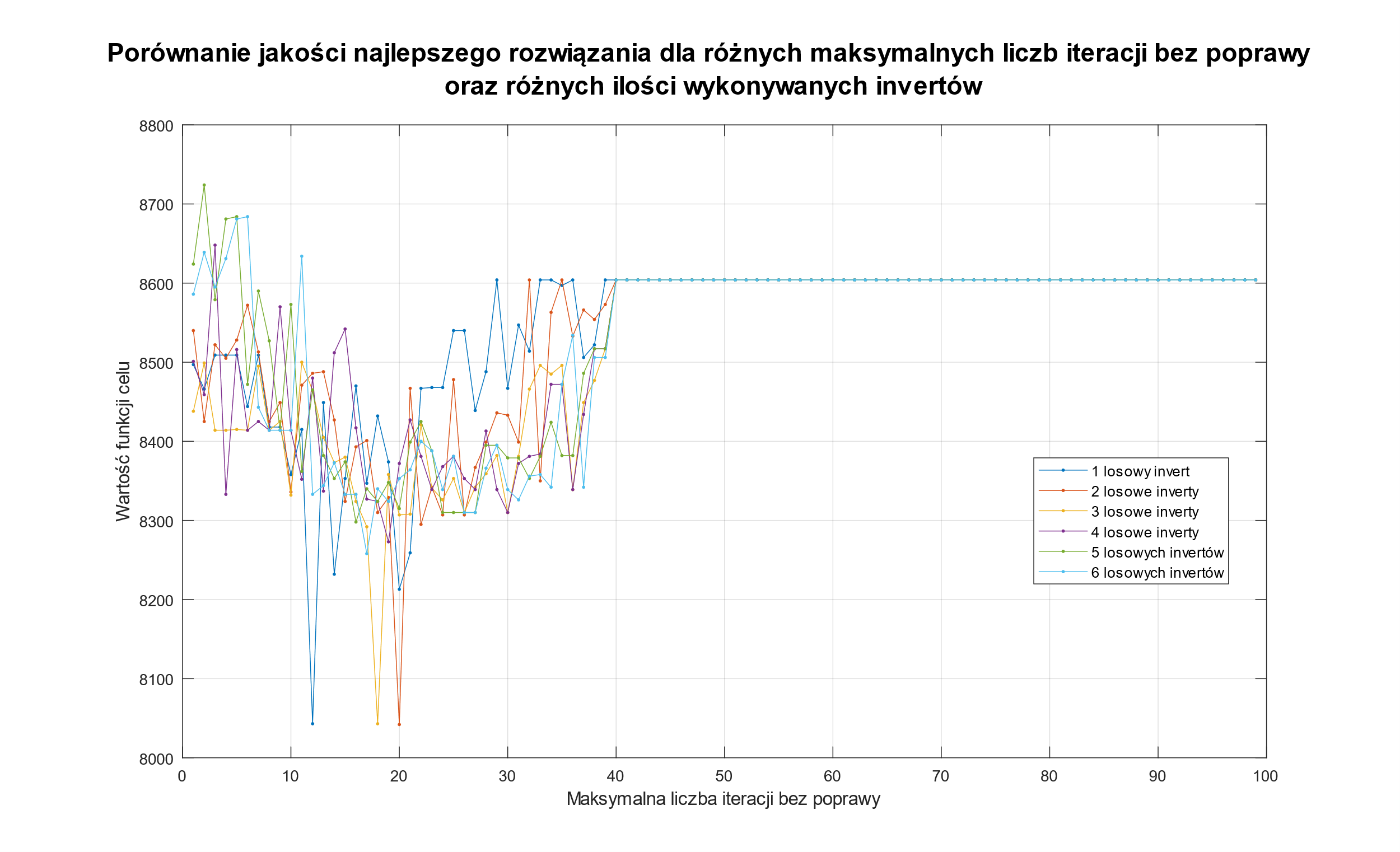
Wykres 4. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
od wielkości instancji dla różnych rozwiązań startowych – instancje losowe.

Dane w tabeli 5. oraz na wykresie 4. wskazują na to, że najniższe wartości funkcji celu zwrócił algorytm Tabu Search startujący z rozwiązania uzyskanego przez rozszerzony algorytm najbliższego sąsiada oraz przez algorytm 2-OPT. Różnice między wynikami zwracanymi przez te dwa warianty są niewielkie. Znacznie wyższe wartości funkcji celu zwracał algorytm Tabu Search rozpoczynający działanie z rozwiązania uzyskanego przez algorytm k-Random, a jeszcze nieco wyższe, gdy rozpoczynał z rozwiązania losowego. Jednak dla małych instancji (o rozmiarze do ok. 100) różnice między wszystkimi czterema wariantami nie są duże.

* 1. WNIOSKI:

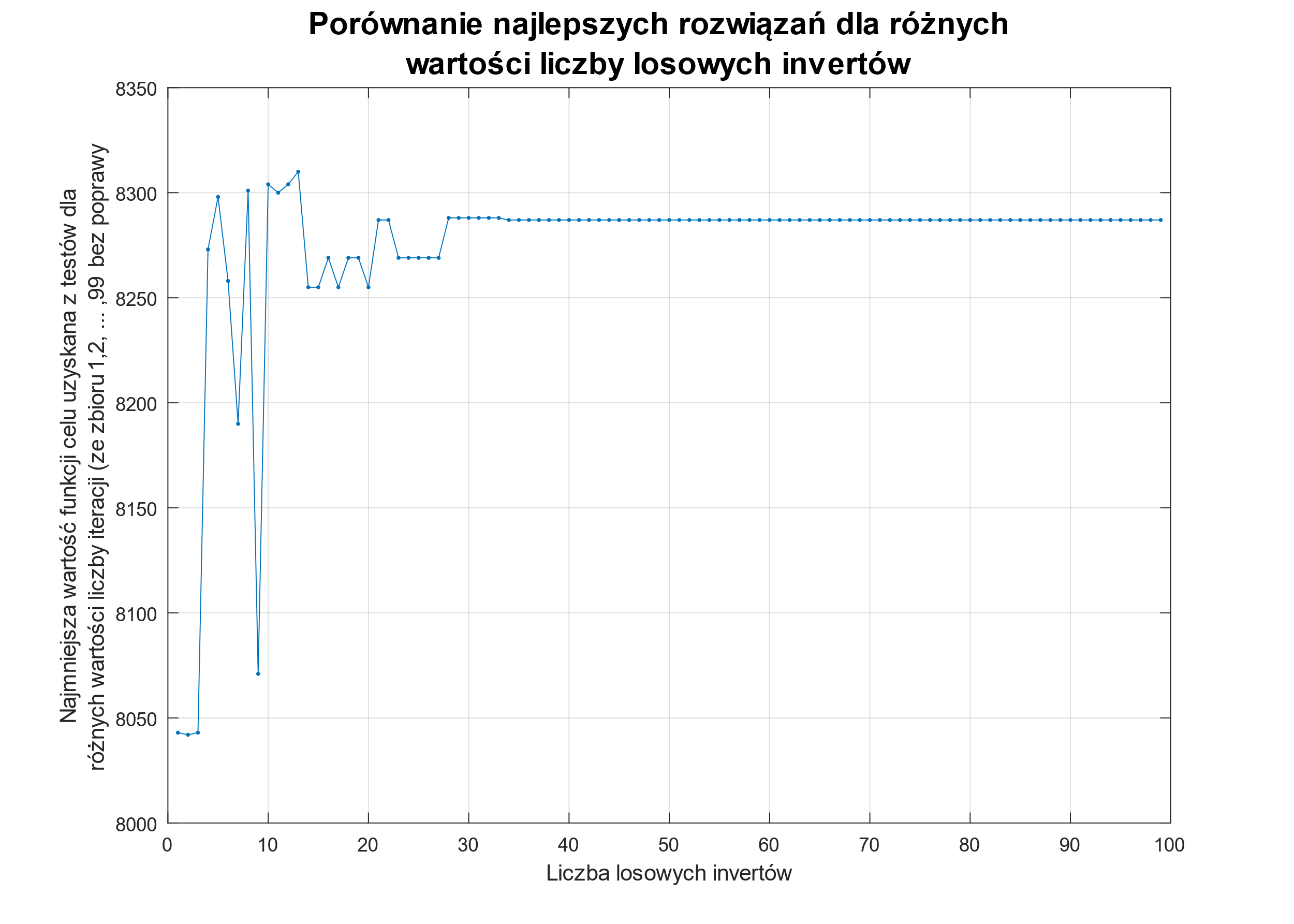
Z danych w tabeli 5. oraz na wykresie 4. wynika, że najefektywniej działa algorytm Tabu Search startujący z rozwiązania zwracanego przez algorytm 2-OPT lub rozszerzony algorytm najbliższego sąsiada. Znacznie mniej efektywne są warianty rozpoczynające od rozwiązania losowego lub rozwiązania zwracanego przez algorytm k-Random. Różnice między tymi dwiema grupami zwiększają się wraz ze wzrostem wielkości instancji.

1. Badanie właściwości mechanizmu kick i wpływu dodatkowych parametrów na jakość rozwiązania zwracanego przez algorytm Tabu Search.
   1. CEL I METODOLOGIA: Celem eksperymentu jest zbadanie właściwości mechanizmu kick, tzn. liczby losowych invertów, które wykonywane są, gdy algorytm przez iteracji nie polepszy aktualnego najlepszego rozwiązania. W tym celu dla instancji ft53 badamy wpływ obu powyższych parametrów na jakość najlepszego rozwiązania dla ustalonych wartości pozostałych parametrów TS - .
   2. OPIS REZULTATÓW:



Wykres 5. Zależność wartości funkcji celu zwracanej przez algorytm Tabu Search   
z mechanizmem kick od parametru – maksymalnej liczby iteracji bez poprawy –   
dla różnych wartości parametru – liczby wykonywanych losowych invertów.

Na wykresie 5. przedstawiono wartości funkcji celu dla kolejnych wartości parametrów } oraz Najmniejszą wartość funkcji celu algorytm uzyskuje dla i . Od pewnej wartości wartość funkcji celu dla wszystkich badanych liczb invertów pozostaje stała i jest równa wartości zwracanej przez nieulepszony algorytm TS (8604). Tłumaczyć to może fakt, iż algorytm nie znajduje od tak długich cykli bez poprawy, przez co żaden kick nie jest wykonywany.



Wykres 6. Porównanie wartości funkcji celu najlepszych rozwiązań zwracanych w eksperymentach opisanych wykresem 5. dla parametru – liczby losowych invertów – rozszerzonego do zbioru {1, 2, …, 100}

Na wykresie 6. przedstawiono najmniejszą wartość funkcji celu uzyskiwaną spośród wszystkich wywołań algorytmu TS dla kolejnych liczb invertów, czyli minima z ciągów danych przedstawionych na wykresie 5.(jednak tutaj dane poszerzone są do 100 losowych invertów). Na jego podstawie jesteśmy w stanie wywnioskować, jaka liczba losowych invertów jest najlepsza. Najlepsze wyniki uzyskujemy dla stosunkowo małych liczb losowych invertów, a od pewnego miejsca wartości nie ulegają zmianom. Poza tym nie da się jednak zaobserwować żadnego trendu.

* 1. WNIOSKI:

Z przeprowadzonych badań wynika, że wzbogacenie algorytmu TS o kicka może skutkować poprawą jakości rozwiązania, jednak tylko dla specyficznych parametrów i zależnych od instancji. Ze względu na małą dostępną moc obliczeniową, przeprowadzenie badań dla większej ilości instancji było niemożliwe, a więc nie jesteśmy w stanie uogólnić tego wniosku.