



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ZARZĄDZANIA

Sprawozdanie z projektu I – Porównanie efektywności wybranych algorytmów w rozwiązaniu problemu komiwojażera.

Grupa projektowa: *Małgorzata Balwierz, Izabela Guła, Laura Cygan, Magdalena Chmaj, Wiktoria Bąk*

Kierunek: *Informatyka i ekonometria (stacjonarne)*

Przedmiot: *Inteligencja Obliczeniowa*

Kraków, 2025

Spis treści

Rozwiązania Solver.....	4
Najlepsze trasy:.....	4
NN – Algorytm Najbliższego Sąsiada	5
Wprowadzenie	5
Działanie algorytmu	5
Działanie kodu	5
Macierz 48	5
Macierz 76	5
Macierz 127	6
GA – Algorytmy Genetyczne.....	6
Wprowadzenie	6
Działanie programu	6
Macierz 48	8
Macierz 76	10
Macierz 127	12
Podsumowanie GA	14
TS – Przeszukiwanie Tabu	14
Wprowadzenie	14
Parametry i ich wartości	15
Macierz 48	15
Macierz 76	17
Macierz 127	19
Podsumowanie TS	20
Algorytm wspinaczki z multistartem	21
Wprowadzenie	21
Parametry i ich wartości	21
Macierz 48	23
Macierz 76	25
Macierz 127	28
Podsumowanie.....	30
SA – symulowane wyżarzanie	31
Wprowadzenie	31

Parametry i ich wartości	31
Działanie programu	32
Macierz 48	32
Macierz 76	32
Macierz 127	33
Podsumowanie SA.....	33
Algorytm mrówkowy	39
Wprowadzenie	39
Działanie algorytmu	40
Działanie kodu	40
Macierz 48	41
Macierz 76	43
Macierz 127	46
Bibliografia.....	50

Rozwiązania Solver

W celu stworzenia pewnego rodzaju „benchmarku” dla zastosowanych algorytmów, problem optymalizacji trasy, znany jako problem komiwojażera, został rozwiązany w programie Excel przy wykorzystaniu dodatku Solver.

Dla każdej macierzy proces wykonano 10 razy, a poniższa tabela przedstawia uzyskane wyniki uszeregowane rosnąco:

Macierz 48	Macierz 76	Macierz 127
10690	112791,7	137406,2
10730	115279,4	138048,2
10829	115279,4	138686,8
10878	117953,9	141410,2
10913	119613,5	145256,2
11012	120306,6	147512
11211	120510,8	147697,1
11262	123174,8	147868,6
11302	124149,7	148478
11335	125498,6	149127,9

Najlepsze trasy:

Macierz 48:

1, 8, 38, 31, 44, 18, 7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 46, 33, 20, 47, 21, 13, 14, 25, 39, 32, 24, 10, 45, 35, 4, 26, 42, 2, 29, 5, 48, 34, 41, 16, 22, 3, 23, 11, 12, 15, 40, 9

Macierz 76:

66, 65, 51, 52, 55, 56, 58, 57, 63, 64, 71, 72, 73, 62, 61, 59, 60, 41, 40, 39, 38, 34, 54, 53, 42, 43, 27, 26, 19, 31, 30, 29, 28, 33, 32, 35, 36, 37, 18, 17, 16, 15, 74, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 20, 4, 3, 2, 75, 76, 1, 23, 22, 21, 25, 24, 46, 45, 44, 48, 47, 69, 68, 70, 67, 49, 50

Macierz 127:

91, 64, 58, 3, 10, 120, 7, 13, 50, 100, 113, 65, 99, 92, 89, 125, 104, 71, 110, 85, 86, 87, 88, 109, 96, 119, 63, 84, 81, 126, 82, 83, 102, 101, 98, 127, 97, 28, 122, 32, 29, 33, 25, 26, 27, 31, 79, 80, 78, 117, 76, 75, 69, 70, 68, 19, 22, 23, 4, 24, 108, 6, 106, 15, 105, 114, 11, 9, 8, 67, 73, 74, 72, 18, 77, 21, 17, 20, 12, 14, 41, 36, 37, 35, 16, 1, 2, 51, 57, 54, 45, 103, 44, 40, 43, 30, 34, 39, 38, 42, 123, 95, 93, 107, 111, 112, 94, 46, 118, 48, 53, 49, 47, 55, 66, 124, 52, 56, 121, 5, 115, 90, 116, 60, 59, 62, 61

NN – Algorytm Najbliższego Sąsiada

Wprowadzenie

Algorytm najbliższego sąsiada jest jedną z podstawowych heurystyk często stosowaną w problemie komiwojażera. Jego działanie opiera się na strategii zachłannej, w której na każdym etapie wybierane jest najlepsze dostępne rozwiązanie lokalne – w tym przypadku, najbliższe nieodwiedzone miasto. Założeniem algorytmu jest to, że podejmowanie optymalnych decyzji na poziomie lokalnym może prowadzić do dobrego rozwiązania całego problemu.

Działanie algorytmu

Algorytm rozpoczyna się od losowego wyboru punktu początkowego, czyli miasta z którego wyruszamy. Następnie, z bieżącego punktu, wybierany jest najbliższy nieodwiedzony wierzchołek, czyli taki, który ma najmniejszą odległość od aktualnej lokalizacji. Algorytm przechodzi do tego punktu i oznacza go jako odwiedzony. Proces ten powtarza się, aż wszystkie wierzchołki zostaną odwiedzone. Na końcu, aby zamknąć trasę, algorytm wraca do punktu początkowego. Jego głównymi zaletami są prostota i szybkość działania.

Działanie kodu

Kod pobiera dane z Excela, po czym sprawdza i wypisuje długość trasy dla każdego parametru „miasto startowe”, oraz najlepszą opcję. Pętla iteruje więc tyle razy ile jest miast startowych (48, 72 lub 127). Miasto startowe jest jedynym parametrem w tym algorytmie

Macierz 48

Najlepsza wyznaczona trasa zaczynała się od miasta numer 10.

Jej koszt wynosi 12012.

Średnia trasa wszystkich znalezionych tras to: 13018,04167

Czas wykonywania algorytmu to: 00:00:01.4596889

Optymalna trasa przedstawia się następująco:

10 -> 24 -> 42 -> 5 -> 48 -> 39 -> 32 -> 21 -> 47 -> 11 -> 23 -> 14 -> 25 -> 13 -> 12 -> 15 -> 33 ->
46 -> 44 -> 18 -> 7 -> 28 -> 36 -> 30 -> 6 -> 37 -> 19 -> 27 -> 43 -> 17 -> 20 -> 40 ->
9 -> 1 -> 8 -> 38 -> 31 -> 22 -> 16 -> 3 -> 34 -> 41 -> 29 -> 2 -> 26 -> 4 -> 35 -> 45

Macierz 76

Najlepsza wyznaczona trasa zaczynała się od miasta numer 16.

Jej koszt wynosi 130921,0046

Średnia trasa wszystkich znalezionych tras to: 147180,072

Czas wykonywania algorytmu to: 00:00:01.5927736

Optymalna trasa przedstawia się następująco:

Jest dokonywana w tej części projektu na podstawie dwóch metod: selekcji turniejowej oraz selekcji ruletkowej. Selekcja ruletkowa jest zaimplementowana w metodzie `RouletteSelection`, natomiast selekcja turniejowa w metodzie `TournamentSelection`.

W selekcji turniejowej losowo wybierana jest grupa kilku osobników z populacji, niezależnie od ich jakości. Z tej grupy wybierany jest ten, który ma najlepszy wynik, czyli najkrótszą trasę. Dzięki temu silniejsze osobniki mają większe szanse, ale słabsze też mogą czasem zostać wybrane, co zwiększa różnorodność.

W selekcji ruletkowej każdy osobnik ma przypisane szanse na wybór, które są tym większe, im lepszy (krótszy) jest jego wynik. Losowana jest wartość, a algorytm przegląda populację, dodając szanse osobników, aż trafi na tego, który odpowiada wylosowanej wartości. Dzięki temu lepsze rozwiązania są wybierane częściej, ale słabsze również mają niewielką szansę na udział.

Krzyżowanie

Krzyżowanie odbywa się metodami PMX (Partially Mapped Crossover) i OX (Order Crossover), zaimplementowanymi w metodach PMX i OX. W PMX najpierw kopiowana jest część trasy od jednego rodzica, a brakujące miasta są uzupełniane według specjalnych reguł dopasowania. W OX część trasy też jest kopiowana, ale reszta jest wypełniana w kolejności, w jakiej występują u drugiego rodzica.

Mutacje

Mutacje są realizowane trzema metodami: **swap**, **reverse**, i **insert**, zaimplementowanymi w metodzie Mutate. W swap dwa losowe geny zamieniają się miejscami, w reverse fragment trasy jest odwracany, a w insert losowy gen jest przenoszony na inną pozycję w trasie.

Parametry

W kodzie zostały wyodrębnione następujące parametry:

- **PopulationSize (rozmiar populacji):** To liczba tras, które algorytm rozważa jednocześnie. Każda trasa to jeden kandydat na optymalne rozwiązanie. Więcej tras w populacji oznacza większe szanse na znalezienie krótszej drogi, ale wymaga więcej obliczeń.
- **MaxIterations (maksymalna liczba iteracji):** Określa, ile razy algorytm będzie próbował ulepszać rozwiązania. W każdej iteracji tworzy nowe, lepsze trasy na podstawie aktualnych. Im więcej pokoleń, tym bardziej algorytm może zbliżyć się do najkrótszej trasy.
- **ParentSelection (metoda selekcji rodziców):** Decyduje, jak wybierane są trasy do krzyżowania (turniejowa lub ruletkowa). Różnicowanie pozwala testować, która metoda lepiej balansuje między eksploracją nowych rozwiązań (ruletka) a eksploatacją istniejących dobrych tras (turniej).
- **MutationType (typ mutacji):** Określa sposób zmiany trasy, aby wprowadzić różnorodność (swap, reverse, insert). Mutacje pomagają uniknąć utknięcia w lokalnych minimach.
- **CrossoverType (typ krzyżowania):** Decyduje, w jaki sposób geny rodziców są łączone, aby stworzyć nowe trasy (PMX lub OX). Wpływa na jakość i różnorodność nowych rozwiązań.

Użyłam poniższych kombinacji parametrów:

```
int[] populationSizes = { 50, 100, 200, 500 };
```

```
int[] maxIterationsArray = { 500, 1000, 2000, 5000 };
```

```
string[] parentSelectionMethods = { "tournament", "roulette" };  
string[] mutationTypes = { "swap", "reverse", "insert" };  
string[] crossoverTypes = { "PMX", "OX" };
```

Finalnie parametry zostają powtórzone dla każdej kombinacji mutacji (3 metody), selekcji (2 metody), krzyżowania (2 metody), rozmiaru populacji (4 rozmiary) oraz maksymalnej wartości iteracji (4 wartości), co daje nam 192 różne kombinacje tras.

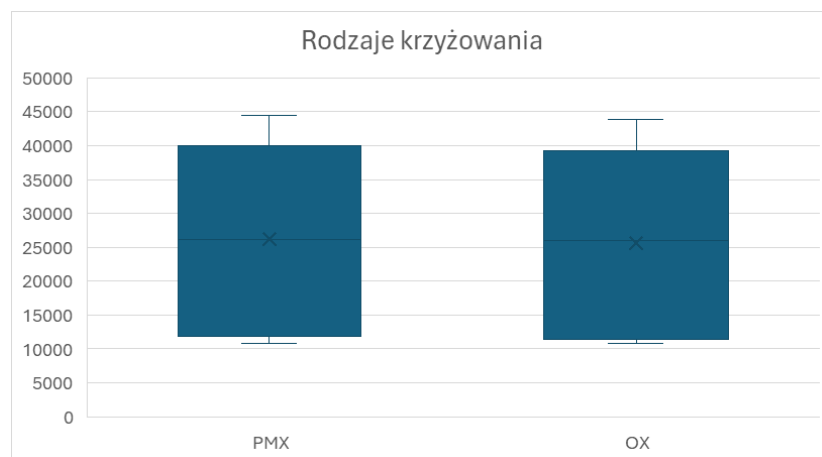
Macierz 48

Najkrótsza trasa przy tej macierzy wynosi 10767, została wskazana dla kombinacji parametrów:

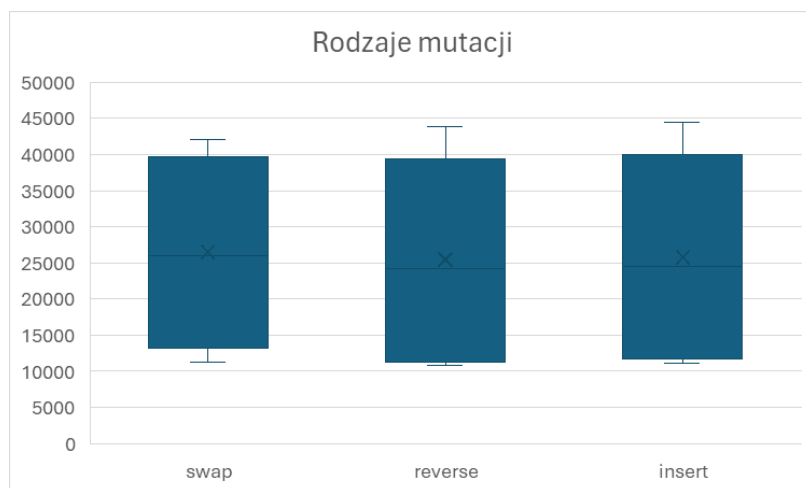
- **PopulationSize:** 100
- **MaxIterations:** 500
- **ParentSelection:** tournament
- **MutationType:** reverse
- **CrossoverType:** OX

Średnia długość trasy ogółem dla tej macierzy wynosi: 25913,13.

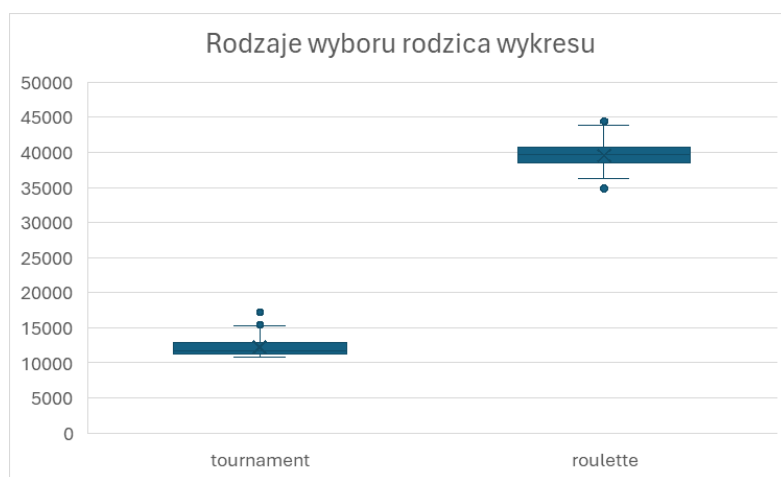
Wizualizacja wyników dla konkretnych parametrów wygląda następująco:



W przypadku rodzajów krzyżowania widzimy, że nie występuje między nimi istotna różnica - wartości są podobne. Możemy więc uznać, że ich działanie przynosi praktycznie identyczne skutki.

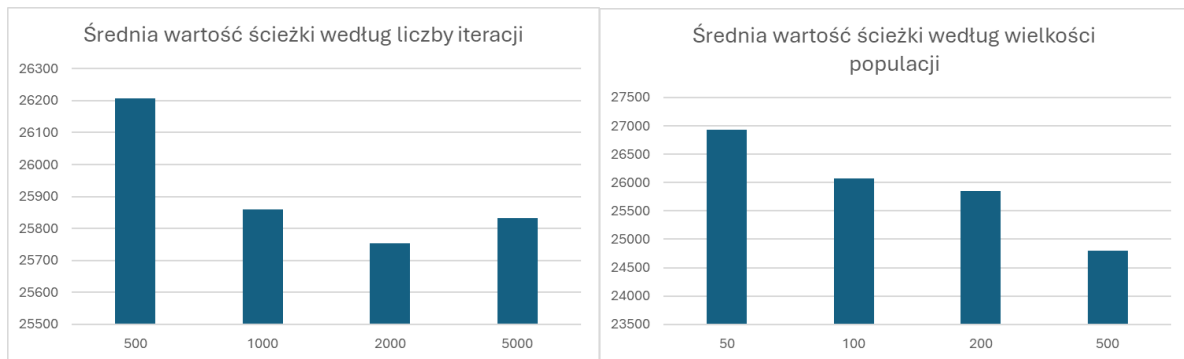


W przypadku rodzajów mutacji również widzimy, że nie występuje między nimi istotne różnice, jedynie rodzaj “swap” różni się w jakiś sposób. Najniższe wartości tego rodzaju mutacji są wyższe niż w przypadku pozostałych metod. Możemy uznać, że ich działanie tych metod jest podobne, reverse i insert działają niemalże identycznie.



W przypadku rodzajów selekcji rodziców widzimy, że występują między metodami spore różnice wartości. Przy tournament rozrzut danych jest mniejszy, wartości są o wiele niższe niż przy metodzie roulette, co jest dla nas korzystniejsze, ponieważ szukamy najniższej trasy. Przy roulette rozrzut danych jest większy, co może sugerować, że metoda jest mniej stabilna, wartości są o wiele wyższe niż w metodzie tournament, co jest dla nas niekorzystne, ponieważ szukamy najniższej trasy.

Metoda turniejowa (tournament) jest bardziej efektywna w tym przypadku, prowadząc do lepszych wyników (krótszych tras) i większej przewidywalności wyników w porównaniu do metody ruletkowej (roulette).



Na podstawie wykresów średnich wartości ścieżek według liczby iteracji i wielkości populacji możemy wywnioskować, że im większa wielkość populacji tym lepszy (mniejszy) wynik możemy uzyskać, natomiast im większa liczba iteracji tym również lepszy (mniejszy) wynik uzyskujemy, jednak dla tej macierzy 5000 jest już zbyt wygórowaną liczbą iteracji, dlatego przy 2000 uzyskujemy najmniejszą wartość.

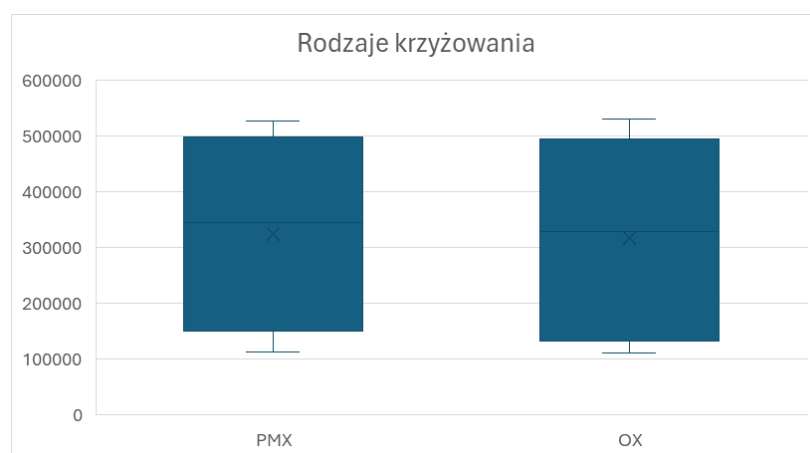
Macierz 76

Najkrótsza trasa przy tej macierzy wynosi 109879, została wskazana dla kombinacji parametrów:

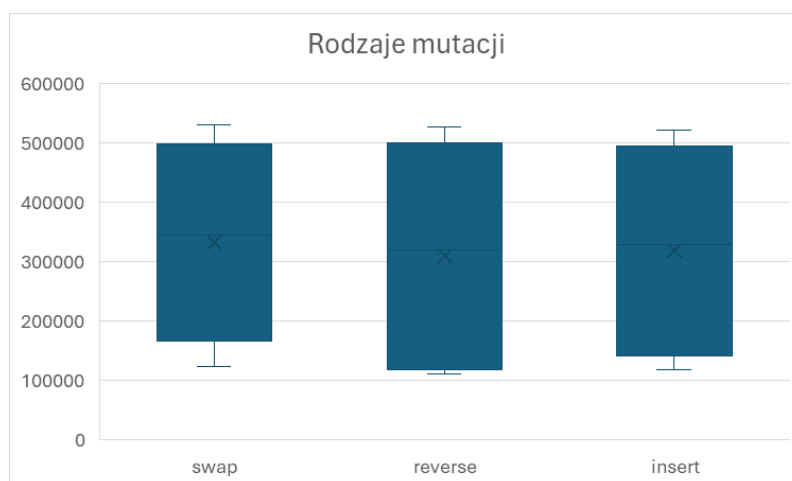
- **PopulationSize:** 100
- **MaxIterations:** 5000
- **ParentSelection:** tournament
- **MutationType:** reverse
- **CrossoverType:** OX

Średnia długość trasy ogółem dla tej macierzy wynosi: 319641,8

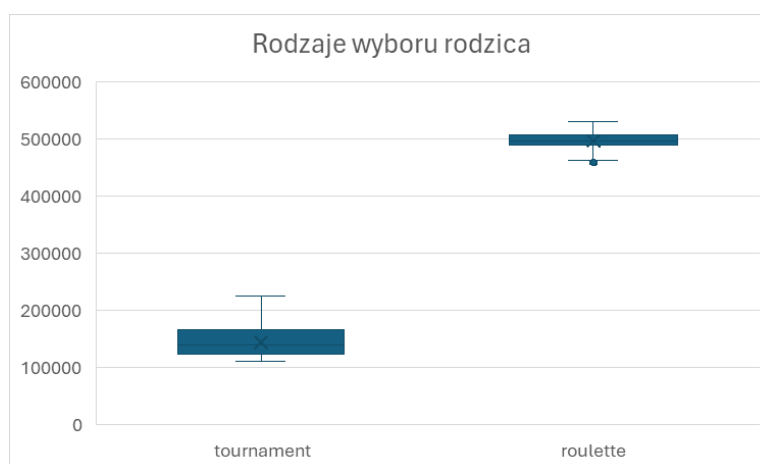
Wizualizacja wyników dla konkretnych parametrów wygląda następująco:



W przypadku rodzajów krzyżowania widzimy, że nie występuje między nimi istotna różnica - wartości są podobne. Możemy więc uznać, że ich działanie przynosi praktycznie identyczne skutki.

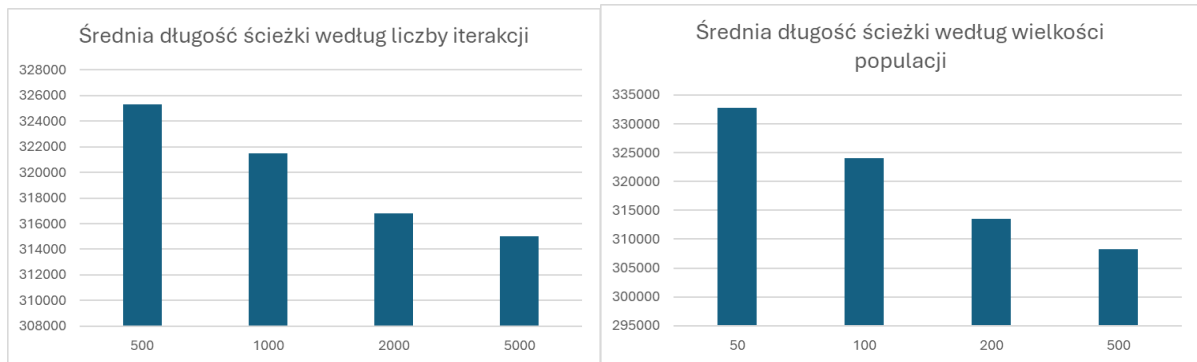


W przypadku rodzajów mutacji również widzimy, że są podobne. Jednak występują między nimi niewielkie różnice. Najniższe wartości mutacji swap są wyższe niż w przypadku pozostałych metod. Najlepsze wyniki - ogólnie najmniejsze uzyskujemy dla metody reverse.



W przypadku rodzajów selekcji rodziców widzimy, że występują między metodami spore różnice wartości. Przy tournament rozrzut danych jest większy, wartości są o wiele niższe niż przy metodzie roulette, co jest dla nas korzystniejsze, ponieważ szukamy najniższej trasy. Przy roulette rozrzut danych jest mniejszy, co może sugerować, że metoda jest stabilniejsza, jednak wartości są o wiele wyższe niż w metodzie tournament, co jest dla nas niekorzystne, ponieważ szukamy najniższej trasy.

Metoda turniejowa (tournament) jest bardziej efektywna w tym przypadku, prowadząc do lepszych wyników (krótszych tras) w porównaniu do metody ruletkowej (roulette).



Na podstawie wykresów średnich wartości ścieżek według liczby iteracji i wielkości populacji możemy wywnioskować, że im większa wielkość populacji tym lepszy (mniejszy) wynik możemy uzyskać, również im większa liczba iteracji tym również lepszy (mniejszy) wynik uzyskujemy.

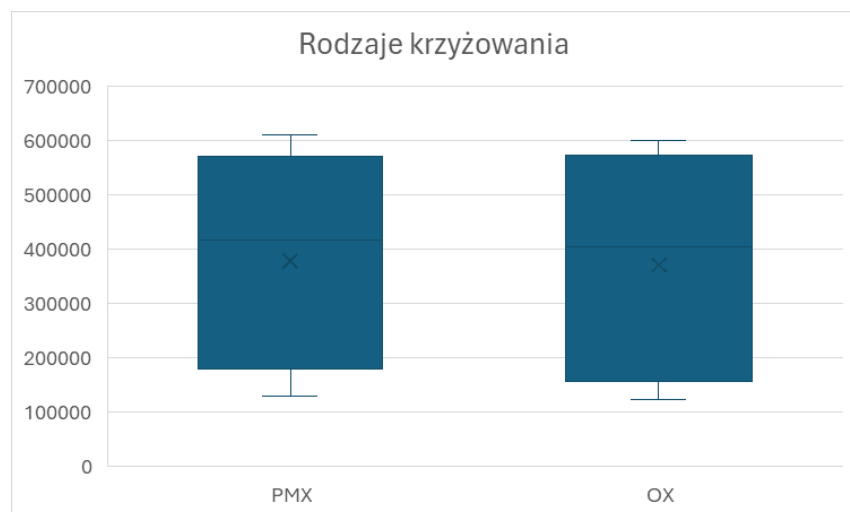
Macierz 127

Najkrótsza trasa przy tej macierzy wynosi 121798, została wskazana dla kombinacji parametrów:

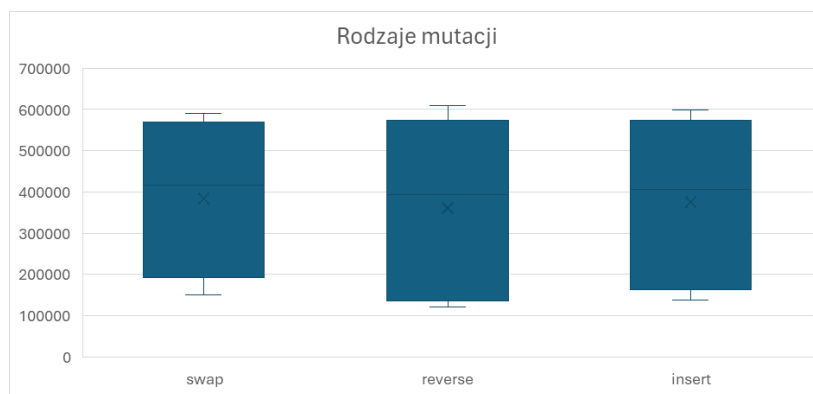
- **PopulationSize:** 500
- **MaxIterations:** 5000
- **ParentSelection:** tournament
- **MutationType:** reverse
- **CrossoverType:** OX

Średnia długość trasy ogółem dla tej macierzy wynosi: 373761,2.

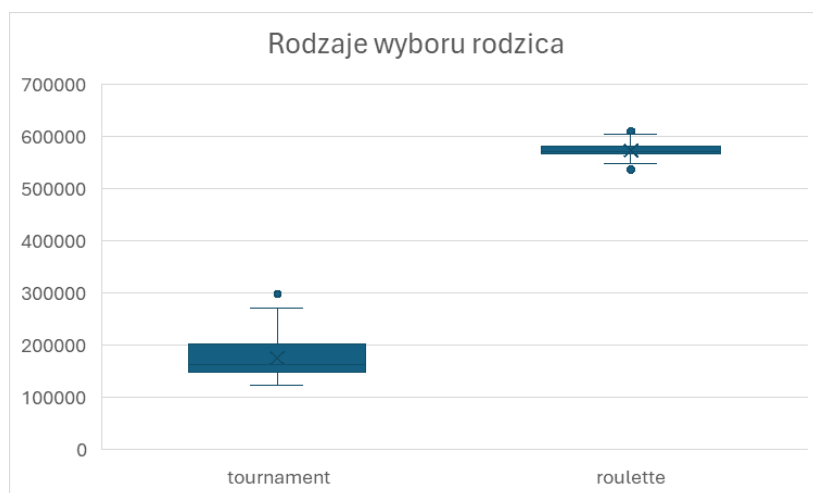
Wizualizacja wyników dla konkretnych parametrów wygląda następująco:



W przypadku rodzajów krzyżowania widzimy, że nie występuje między nimi istotna różnica - wartości są podobne. Możemy więc uznać, że ich działanie przynosi praktycznie identyczne skutki.

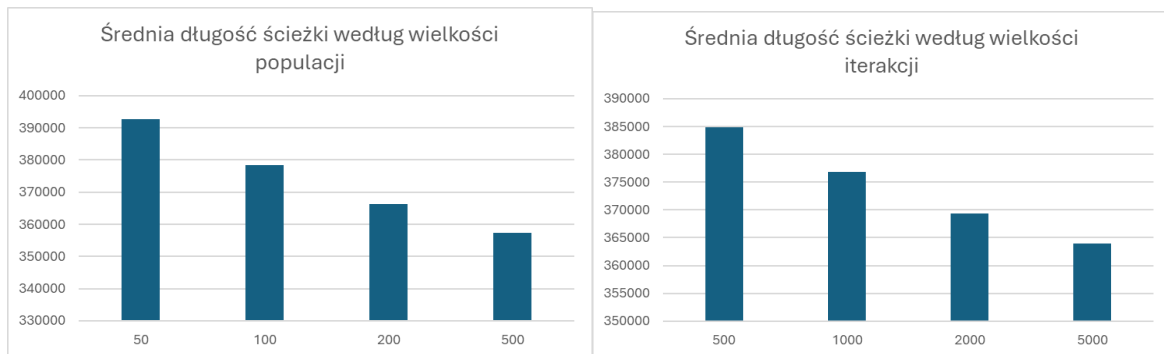


W przypadku rodzajów mutacji również widzimy, że są podobne. Jednak występują między nimi niewielkie różnice. Najniższe wartości mutacji swap są wyższe niż w przypadku pozostałych metod. Najlepsze wyniki - ogólnie najmniejsze uzyskujemy dla metody reverse.



W przypadku rodzajów selekcji rodziców widzimy, że występują między metodami spore różnice wartości. Przy tournament rozrzut danych jest większy, wartości są o wiele niższe niż przy metodzie roulette, co jest dla nas korzystniejsze, ponieważ szukamy najniższej trasy. Przy roulette rozrzut danych jest mniejszy, co może sugerować, że metoda jest stabilniejsza, jednak wartości są o wiele wyższe niż w metodzie tournament, co jest dla nas niekorzystne, ponieważ szukamy najniższej trasy.

Metoda turniejowa (tournament) jest bardziej efektywna w tym przypadku, prowadząc do lepszych wyników (krótszych tras) w porównaniu do metody ruletkowej (roulette).



Na podstawie wykresów średnich wartości ścieżek według liczby iteracji i wielkości populacji możemy wywnioskować, że im większa wielkość populacji tym lepszy (mniejszy) wynik możemy uzyskać, również im większa liczba iteracji tym również lepszy (mniejszy) wynik uzyskujemy.

Podsumowanie GA

Na podstawie analizy można wskazać kilka istotnych wniosków.

Program optymalizuje trasy w problemie komiwojażera za pomocą algorytmu genetycznego. Większa liczba populacji i iteracji daje lepsze wyniki, ale wydłuża czas działania. Najlepsze efekty uzyskano zawsze przy kombinacji:

- ParentSelection: tournament,
- MutationType: reverse,
- CrossoverType: OX,

co pokazuje, że te ustawienia są najbardziej skuteczne. Oznacza to, że ta kombinacja metod daje nam najlepszy możliwy wynik.

TS – Przeszukiwanie Tabu

Wprowadzenie

Algorytm przeszukiwania tabu (Tabu Search) to metaheurystyka, która iteracyjnie poszukuje najlepszego rozwiązania w przestrzeni rozwiązań poprzez eksplorację sąsiedztwa aktualnego rozwiązania. W każdej iteracji wybierane jest najlepsze dostępne rozwiązanie z sąsiedztwa, nawet jeśli pogarsza ono wynik, co pozwala na wyjście z lokalnych minimów. Aby uniknąć powrotu do wcześniej odwiedzonych rozwiązań, wprowadzana jest lista tabu, która tymczasowo zabrania wykonywania określonych ruchów lub wyboru rozwiązań. Cały proces pozwala na efektywne przeszukiwanie przestrzeni rozwiązań i znajduje zastosowanie w problemach, gdzie optymalizacja klasycznymi metodami jest trudna lub czasochłonna.

Parametry i ich wartości

W implementacji algorytmu zastosowane zostały cztery parametry:

- **Liczba iteracji** - określa maksymalną liczbę iteracji algorytmu, czyli ile razy zostanie przeszukane sąsiedztwo w celu znalezienia lepszego rozwiązania.
- **Liczba iteracji bez poprawy** - definiuje limit kolejnych iteracji bez poprawy rozwiązania, po którym algorytm przerywa działanie.
- **Długość listy tabu** - ustala rozmiar listy tabu, czyli liczbę ruchów, które będą blokowane, aby zapobiec powrotowi do wcześniej odwiedzonych rozwiązań.
- **Rodzaj sąsiedztwa** - określa typ generowanego sąsiedztwa, gdzie "swap", "reverse" i "insert" to różne sposoby modyfikacji rozwiązania.

Parametr	Wartość			
	100	250	500	2000
Liczba iteracji	100	250	500	2000
Liczba iteracji bez poprawy	25	50	100	200
Długość listy tabu	25	50	100	200
Rodzaj sąsiedztwa	swap	reverse	insert	-

Taka ilość parametrów i wartości pozwala na stworzenie 192 kombinacji. Każda z nich zostanie kilkakrotnie policzona w celu sprawdzenia, czy któreś kombinacje pozwalają na osiągnięcie optymalnej drogi.

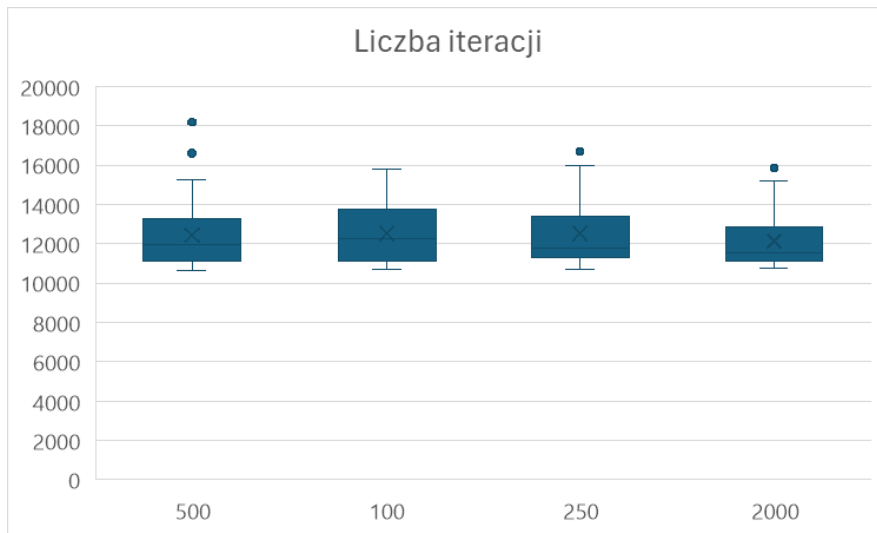
Macierz 48

Wszystkie kombinacje zostały policzone w celu znalezienia optymalnej drogi i wysnucia wniosków. W każdym przypadku można wyciągnąć wnioski mówiące o tym jak dane wartości parametrów wpływają na uzyskane wartości.

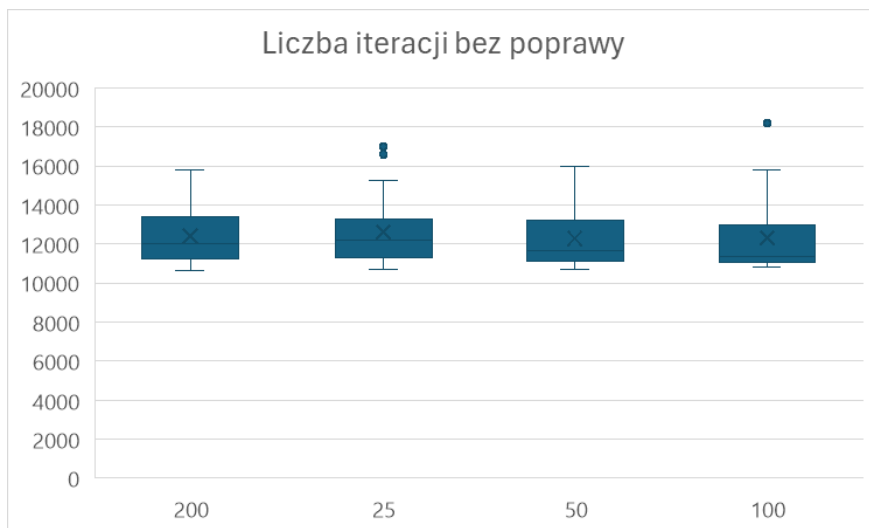
Średni czas generowania 192 kombinacji: 167,23 sekund.

Średnia długość trasy: 12261,82

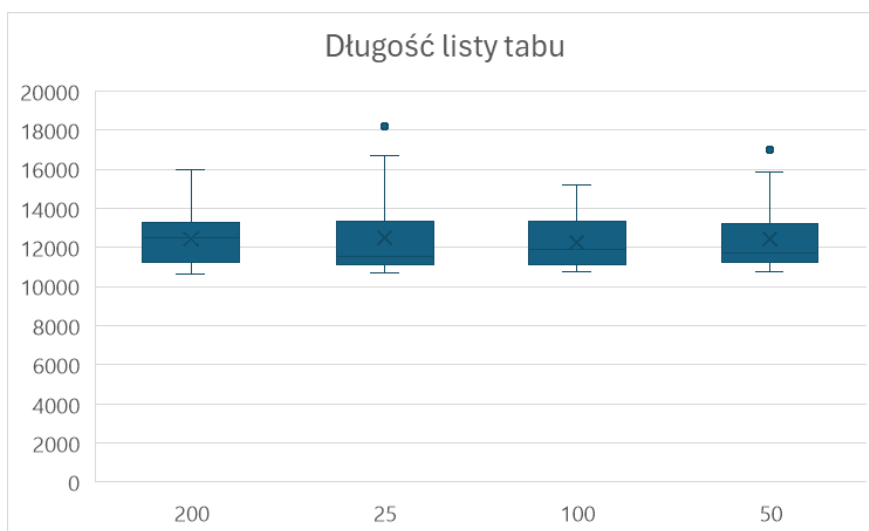
- Liczba iteracji – parametr ten wskazuje, że nawet najmniejsza liczba iteracji pozwala na otrzymanie zadowalających wyników. Jednak w przypadku największej liczby iteracji zauważalne jest spłaszczenie pudełka, sugerujące większą stabilność wyników.



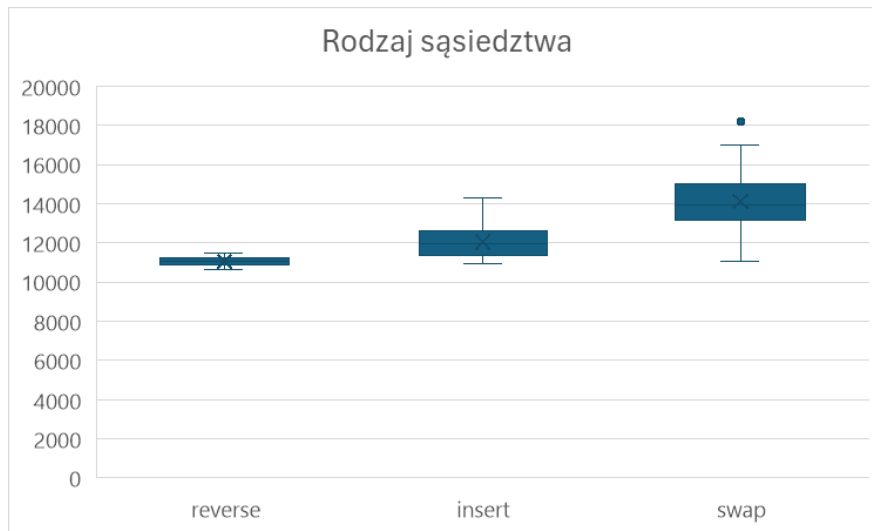
- Liczba iteracji bez poprawy – podobnie jak w poprzednim przypadku, przy każdej wartości parametru można odnotować zadowalające wyniki



- Długość listy tabu - różne wartości tego parametru, również pozwalają osiągnąć dobre wyniki.



- Rodzaj sąsiedztwa - ten parametr pozwala wysnuć najciekawsze wnioski. Przy wykorzystaniu sąsiedztwa „reverse”, każda kombinacja osiąga najlepsze wyniki. Dla sąsiedztwa „insert” również są one zadowalające, jednak nie tak podobne do siebie. Najgorsze wyniki powstają przy wykorzystaniu sąsiedztwa „swap”.



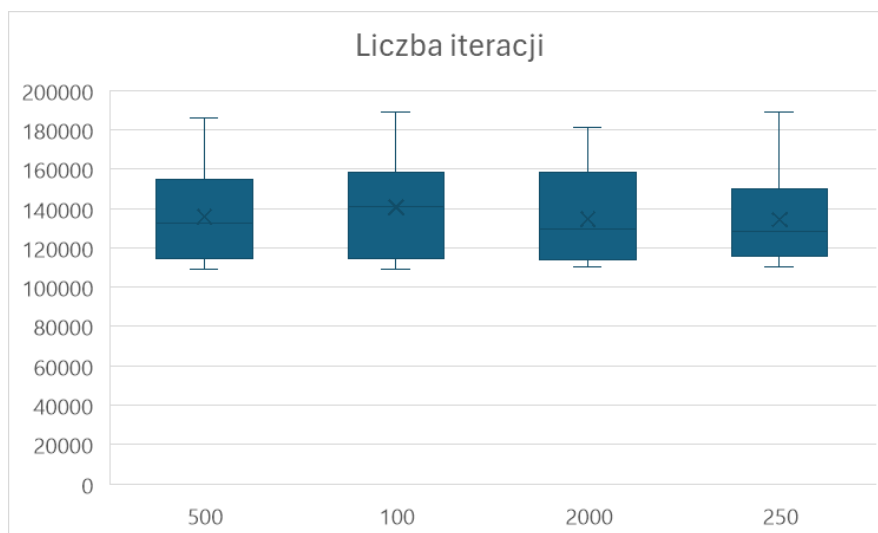
Pięć najlepszych wyników:

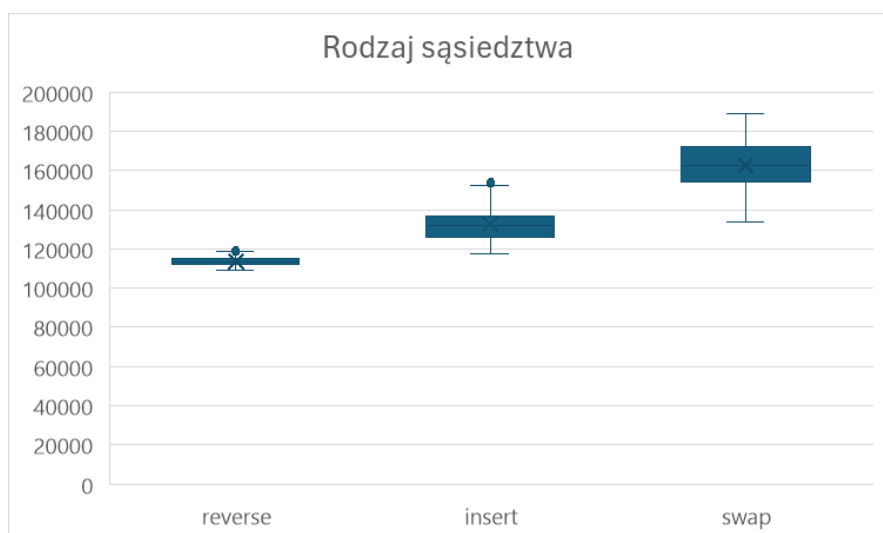
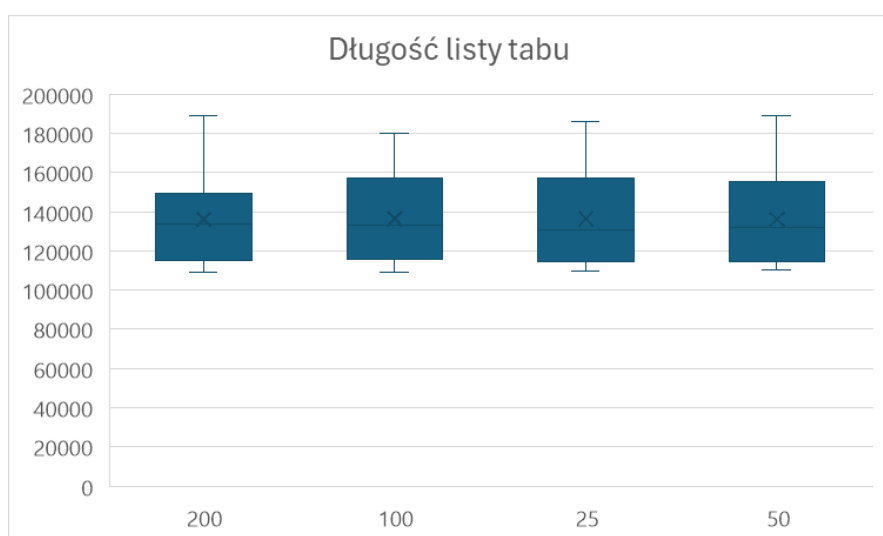
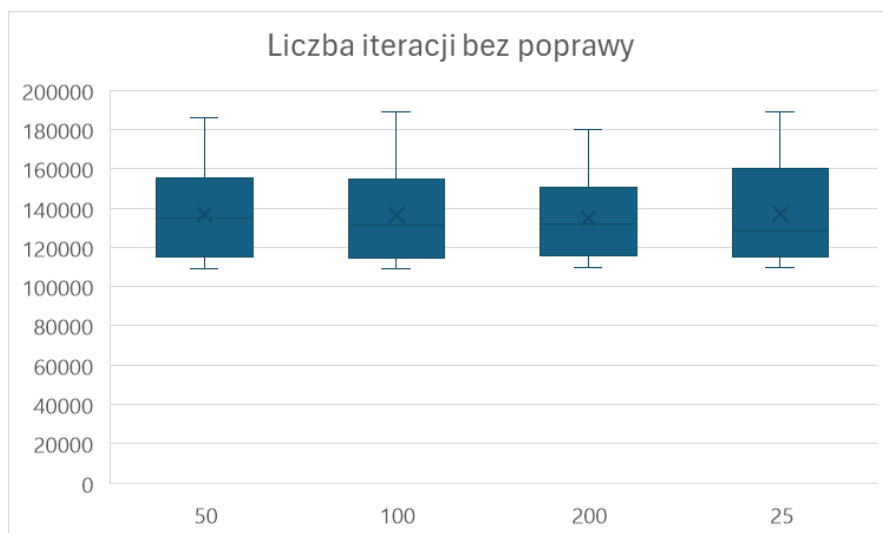
Iterations	No Improvement	Tabu Length	Neighborhood	Best Path Length	Elapsed Time (s)
100	25	25	reverse	10628	0,2235058
2000	25	25	reverse	10730	0,1836302
2000	200	200	reverse	10812	2,2491803
250	25	200	reverse	10831	0,2810408
500	100	100	reverse	10840	0,8661831

Macierz 76

Średni czas generowania 192 kombinacji: 571,58 sekund

Średnia długość trasy: 136362,08





Na podstawie wykresów pudełkowych dla wartości każdego z parametrów dla macierzy 76 można wysnuć bardzo podobne wnioski jak dla macierzy 48. W przypadku większości parametrów nie widać istotnego wpływu żadnej z ich wartości. Natomiast w parametrze określającym rodzaj sąsiedztwa

widoczne są duże różnice. Algorytm wyszukuje najkrótszą drogę i najbardziej zbliżone do siebie wyniki dla sąsiedztwa „reverse”.

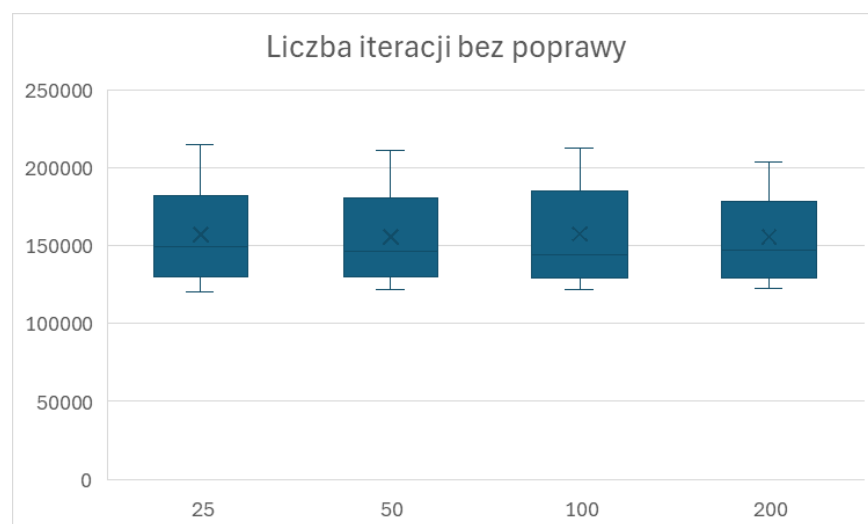
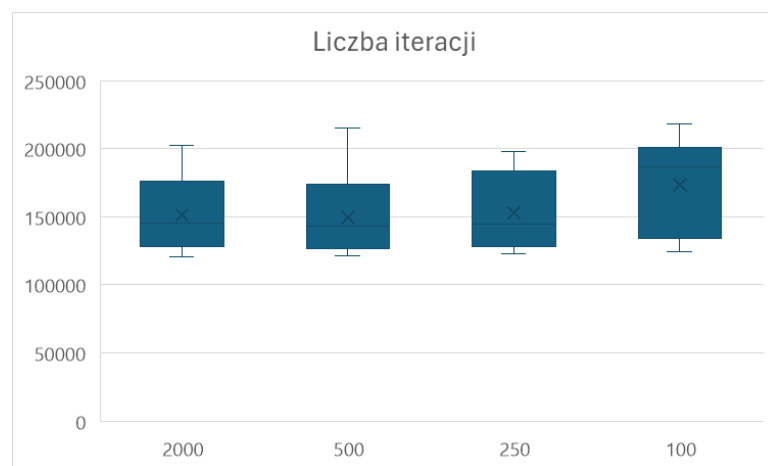
Pięć najlepszych wyników:

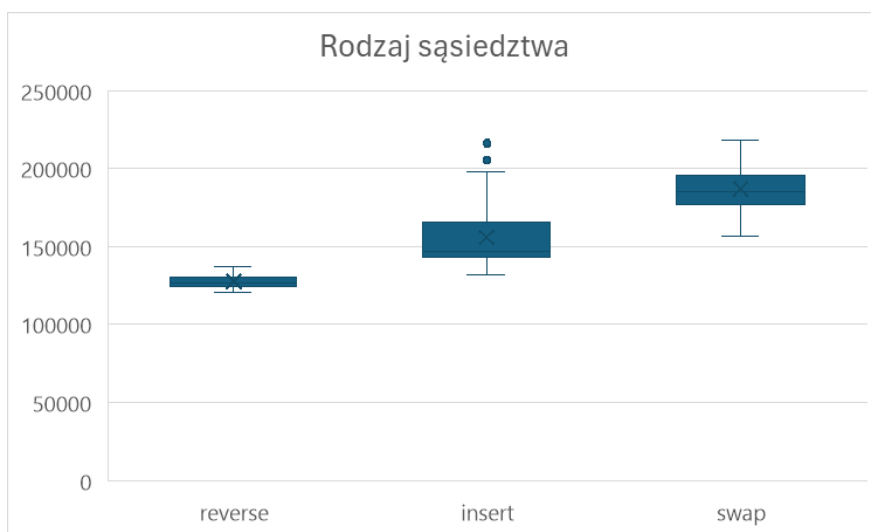
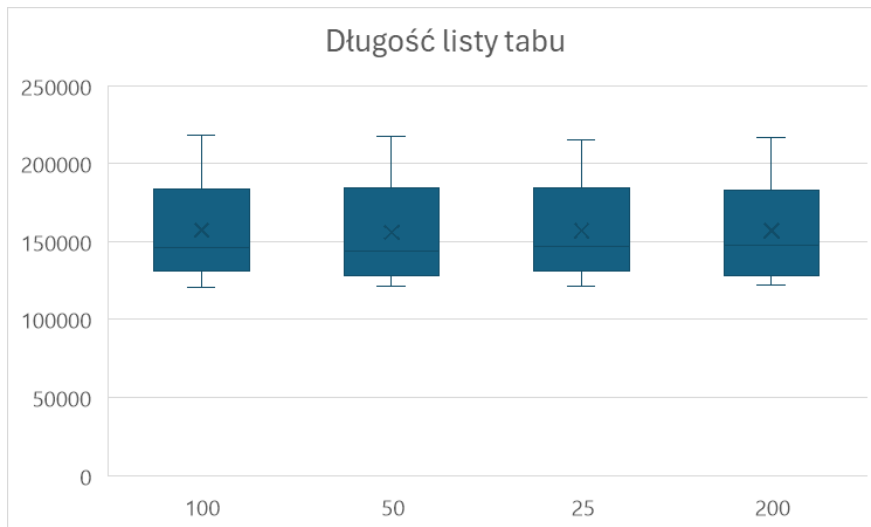
Iterations	No Improvement	Tabu Length	Neighborhood	Best Path Length	Elapsed Time (s)
250	200	50	reverse	108396,1049	1,6441177
500	200	100	reverse	109341,8458	3,6821825
100	25	5	reverse	109712,2494	0,6253996
100	200	100	reverse	109922,1569	0,6354367
2000	50	20	reverse	110041,0146	0,5004213

Macierz 127

Średni czas generowania 192 kombinacji: 2258,12 sekund.

Średnia długość trasy: 156872,11





Największa macierz również potwierdza wcześniejsze wnioski. Największy wpływ na długość drogi w problemie Komiwojażera przy użyciu algorytmu poszukiwania tabu ma parametr dotyczący rodzaju sąsiedztwa. Sąsiedztwo „reverse” pozwala osiągnąć najlepsze wyniki.

Pięć najlepszych wyników:

Iterations	No Improvement	Tabu Length	Neighborhood	Best Path Length	Elapsed Time (s)
500	100	5	reverse	119382,0785	3,4550098
500	200	50	reverse	120678,687	9,6587551
2000	100	100	reverse	121587,6714	7,4636091
250	25	50	reverse	121656,0172	4,6844476
250	100	50	reverse	122649.4952	5.6582553

Podsumowanie TS

W przeprowadzonych testach na różnych macierzach widać, że parametr dotyczący rodzaju sąsiedztwa ma kluczowy wpływ na jakość wyników. Najlepsze rozwiązania osiągane są przy wykorzystaniu sąsiedztwa „reverse”, podczas gdy „swap” daje najgorsze rezultaty. Zwiększanie liczby iteracji i długości listy tabu nie prowadziło do znaczącej poprawy wyników, a efekty były zbliżone, niezależnie

od wybranych wartości tych parametrów. Czas obliczeń rośnie wraz ze wzrostem rozmiaru macierzy, ale algorytm nadal skutecznie generuje rozwiązania bliskie optymalnym, nawet dla dużych macierzy.

Algorytm wspinaczki z multistartem

Wprowadzenie

Algorytm wspinaczki (hill climbing) to metoda lokalnego przeszukiwania wykorzystywana do rozwiązywania problemów optymalizacyjnych, zwłaszcza takich, które wymagają znalezienia najlepszego rozwiązania w dużej przestrzeni stanów. W wariantie stochastycznym algorytm działa w oparciu o iteracyjne ulepszanie bieżącego rozwiązania, losowo wybierając jednego sąsiada obecnego rozwiązania i sprawdzając, czy poprawia on wartość funkcji celu. Jeśli tak, wybrany sąsiad staje się aktualnym rozwiązaniem, a proces jest kontynuowany. Proces przypomina wspinaczkę po krajobrazie, gdzie wysokość odpowiada wartości funkcji oceny dla danego rozwiązania. Algorytm cechuje się prostotą i niewielkimi wymaganiami pamięciowymi, ponieważ operuje wyłącznie na bieżącym stanie i jego losowo wybranym sąsiedzie, zamiast utrzymywać pełne drzewo poszukiwań. Niestety, jednym z jego kluczowych ograniczeń jest skłonność do zatrzymywania się w lokalnych maksimach, gdzie nie ma lepszych sąsiadów, mimo że globalne maksimum znajduje się dalej w przestrzeni.

Aby zwiększyć skuteczność i szanse na znalezienie rozwiązania globalnego, stosuje się wariant algorytmu zwany wspinaczką z multistartami (multi-start hill climbing). Polega on na wielokrotnym uruchamianiu podstawowego algorytmu wspinaczki z losowo wybranych punktów startowych. Za każdym razem proces jest powtarzany do momentu osiągnięcia lokalnego maksimum lub wyczerpania dozwolonej liczby iteracji, a następnie wybierane jest najlepsze z uzyskanych rozwiązań. Dzięki takiemu podejściu możliwe jest zminimalizowanie wpływu lokalnych maksimów i zwiększenie prawdopodobieństwa znalezienia globalnego optimum, pod warunkiem odpowiedniej liczby restartów.

Parametry i ich wartości

W implementacji algorytmu zastosowano cztery kluczowe parametry:

- **Liczba iteracji** – określa maksymalną liczbę iteracji w ramach jednej wspinaczki, czyli ile razy algorytm przeszuka sąsiedztwo w poszukiwaniu lepszego rozwiązania.
- **Liczba iteracji bez poprawy** – definiuje maksymalną liczbę iteracji, podczas których nie nastąpi poprawa rozwiązania. Po przekroczeniu tego limitu algorytm kończy bieżącą wspinaczkę.
- **Liczba restartów** – ustala, ile razy algorytm zostanie uruchomiony od nowego, losowego punktu początkowego, co pozwala na zwiększenie szans na znalezienie globalnego optimum.
- **Rodzaj sąsiedztwa** – wskazuje sposób generowania sąsiednich rozwiązań („swap”, „reverse” lub „insert”), co wpływa na sposób modyfikacji bieżącego rozwiązania.

Początkowo założone wartości parametrów prezentują się następująco:

Parametr	Wartość			
Liczba iteracji	100	500	1000	5000
Liczba iteracji bez poprawy	50	250	500	1000
Liczba restartów	10	25	50	100

Rodzaj sąsiedztwa	swap	reverse	insert	-
-------------------	------	---------	--------	---

Taka liczba parametrów i ich wartości umożliwia utworzenie 192 różnych kombinacji. Najlepsza otrzymana w ten sposób długość trasy dla każdej macierzy przedstawiona jest poniżej.

Macierz 48:

Najlepszy **wynik:** 10688

Trasa:

14 → 34 → 41 → 16 → 22 → 3 → 23 → 11 → 12 → 15 → 40 → 9 → 1 → 8 → 38 → 31 → 44 →
18 → 7 → 28 → 6 → 37 → 19 → 27 → 17 → 43 → 30 → 36 → 46 → 33 → 20 → 47 → 13 → 21 →
39 → 32 → 24 → 10 → 45 → 35 → 4 → 26 → 42 → 2 → 29 → 5 → 48 → 25

Macierz 76:

Najlepszy **wynik:** 117992

Trasa:

1 → 23 → 22 → 24 → 46 → 45 → 44 → 48 → 47 → 69 → 68 → 70 → 67 → 50 → 49 → 52 → 33
→ 32 → 31 → 19 → 20 → 4 → 3 → 7 → 8 → 6 → 5 → 10 → 11 → 9 → 12 → 13 → 14 → 74 → 15
→ 16 → 17 → 18 → 37 → 36 → 39 → 38 → 35 → 34 → 40 → 41 → 60 → 61 → 59 → 58 → 57 →
62 → 63 → 64 → 73 → 72 → 71 → 65 → 66 → 51 → 56 → 55 → 53 → 54 → 42 → 43 → 28 → 29
→ 30 → 26 → 27 → 25 → 21 → 2 → 75 → 76

Macierz 127:

Najlepszy wynik: 165525,31

Trasa:

111 → 112 → 56 → 5 → 124 → 65 → 113 → 64 → 91 → 62 → 61 → 59 → 73 → 75 → 71 → 74
→ 68 → 67 → 19 → 72 → 18 → 22 → 4 → 23 → 8 → 17 → 21 → 20 → 27 → 12 → 14 → 1 → 3
→ 116 → 60 → 90 → 10 → 13 → 120 → 100 → 58 → 9 → 11 → 106 → 114 → 7 → 51 → 37 →
36 → 35 → 40 → 103 → 44 → 45 → 54 → 57 → 121 → 52 → 115 → 50 → 2 → 16 → 30 → 34 →
39 → 42 → 43 → 41 → 15 → 24 → 6 → 105 → 108 → 77 → 81 → 78 → 80 → 79 → 31 → 38 →
26 → 25 → 33 → 32 → 29 → 28 → 122 → 95 → 93 → 127 → 123 → 97 → 98 → 101 → 102 → 83
→ 84 → 76 → 117 → 82 → 126 → 63 → 119 → 69 → 70 → 96 → 109 → 88 → 87 → 86 → 85 →
110 → 104 → 99 → 92 → 89 → 125 → 66 → 55 → 47 → 49 → 53 → 46 → 118 → 48 → 94 → 107

Na podstawie wyników uzyskanych za pomocą Solvera, jak i innych opisywanych algorytmów można jednak wywnioskować, że nie są to trasy zadawalające, szczególnie dla Macierzy 127. Z tego powodu zwiększono wartości parametrów i wykonano algorytm ponownie.

Parametr	Wartość			
Liczba iteracji	1000	5000	10000	20000
Liczba iteracji bez poprawy	500	2500	5000	10000
Liczba restartów	20	50	100	500
Rodzaj sąsiedztwa	swap	reverse	insert	-

Macierz 48

Średni wynik: 12290,13

Czas wykonania wszystkich kombinacji: 161,983 sekundy

Dziesięć najlepszych wyników:

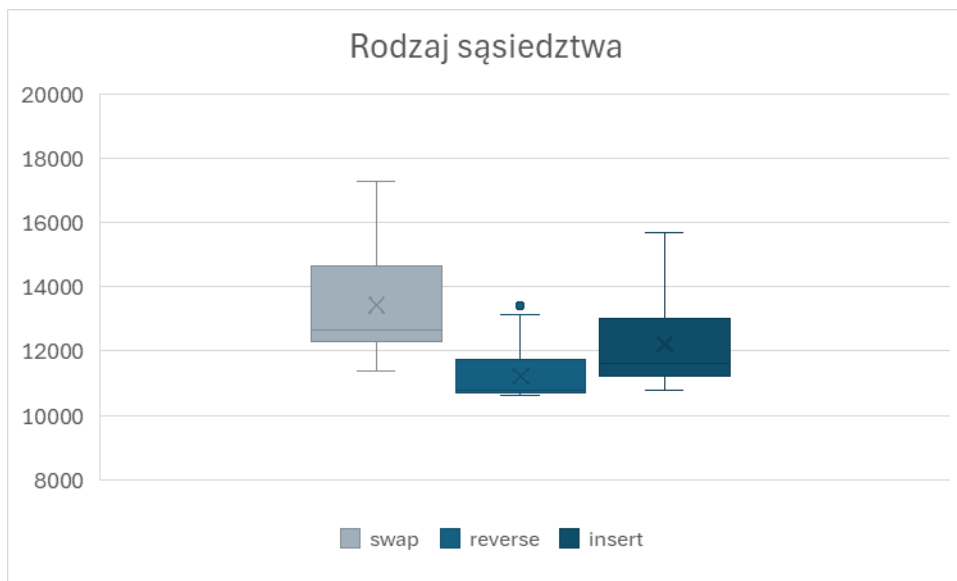
Najlepszy Koszt	Rodzaj Sąsiedztwa	Liczba Restartów	Maks. Iteracje Bez Poprawy	Maksymalna liczba Iteracji
10628	reverse	100	2500	20000
10628	reverse	500	5000	20000
10638	reverse	500	5000	10000
10653	reverse	500	10000	10000
10653	reverse	100	5000	20000
10653	reverse	500	10000	20000
10663	reverse	500	5000	5000
10663	reverse	100	2500	10000
10663	reverse	50	500	20000
10663	reverse	100	10000	20000

Najlepsza trasa:

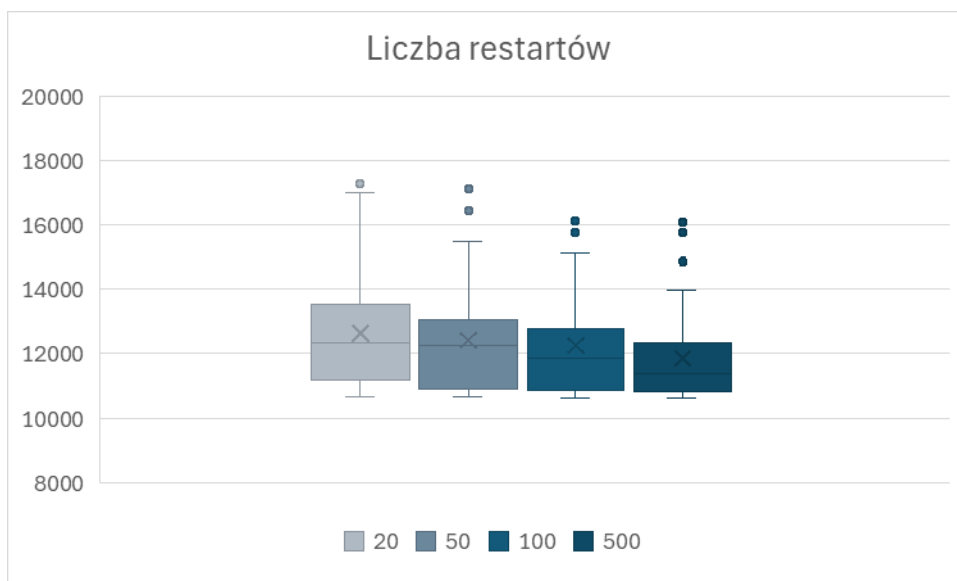
30 → 43 → 17 → 27 → 19 → 37 → 6 → 28 → 7 → 18 → 44 → 31 → 38 → 8 → 1 → 9 → 40 → 15 → 12 → 11 → 13 → 25 → 14 → 23 → 3 → 22 → 16 → 41 → 34 → 29 → 2 → 26 → 4 → 35 → 45 → 10 → 24 → 42 → 5 → 48 → 39 → 32 → 21 → 47 → 20 → 33 → 46 → 36

Czas wykonania dla najlepszej trasy: 0,887 sekundy

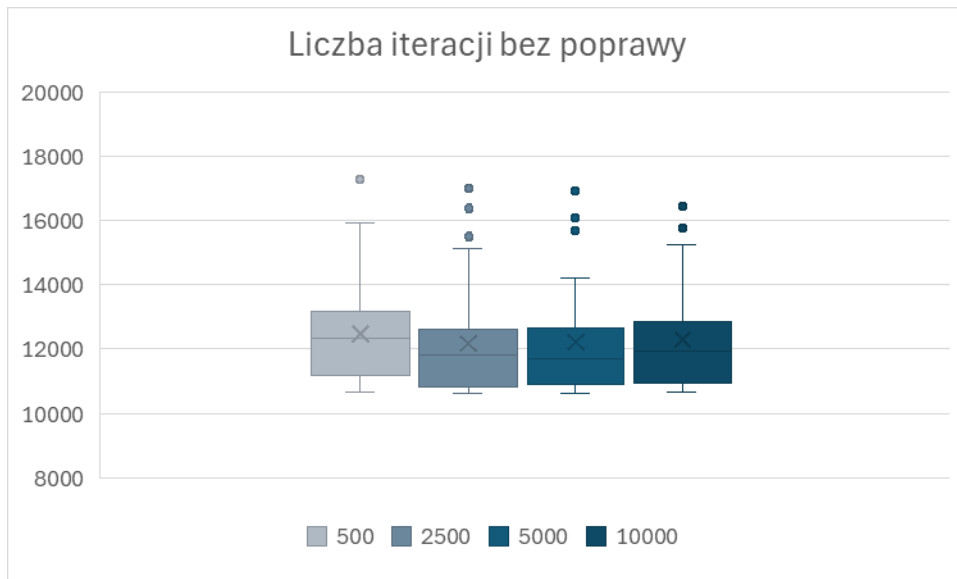
Wpływ parametrów na wyniki:



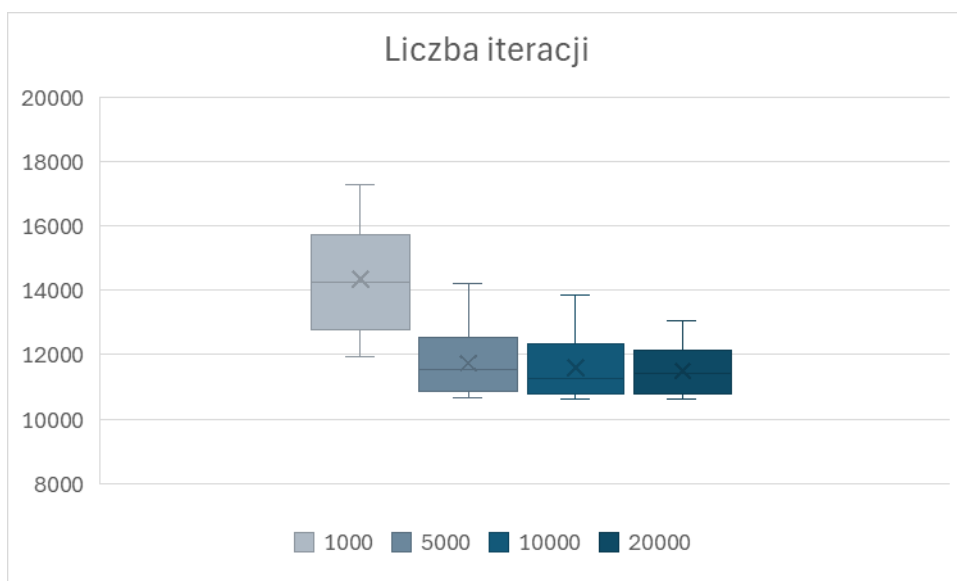
Już po tabeli z najlepszymi wynikami można było zauważyć, że wszystkie wartości tam zapisane uzyskane zostały przy użyciu ruchu “reverse”. Wykazuje on lepsze wyniki (niższe wartości funkcji celu) w porównaniu do pozostałych, a także wyniki te charakteryzują się najmniejszym rozrzutem, co sugeruje większą stabilność w uzyskiwaniu dobrych rozwiązań. Najsłabsze wyniki osiąga “swap”, posiadając najwyższą średnią, medianę oraz największy zakres wyników.



Widocznym jest, że wraz ze wzrostem liczby restartów średnie i mediany wyników maleją. Najlepsze (najniższe) wartości osiągnęte są dla 500 restartów, a także dla tej wartości najmniejszy jest ich rozrzut, czyli algorytm staje się bardziej stabilny. Im większa jest liczba restartów, tym większa jest szansa na uniknięcie utknięcia w lokalnym maksimum i znalezienie lepszego, globalnego rozwiązania.



Dla maksymalnej liczby iteracji bez poprawy trudno jest zauważyć jakąkolwiek zależność. Dla 500 iteracji bez poprawy wyniki są najgorsze, jednak może to wynikać z pewnej dozy losowości. Zwiększanie tego parametru pozwala algorytmowi eksplorować przestrzeń rozwiązań dłużej, ale różnice w wynikach są niewielkie.



Im większa jest maksymalna liczba iteracji, tym większa szansa, że algorytm faktycznie wykona więcej ruchów. W efekcie można zauważyć odwrotną zależność między maksymalną liczbą iteracji a długością wyznaczonej trasy – wraz ze wzrostem liczby iteracji długość trasy ulega skróceniu, co świadczy o poprawie jakości rozwiązania. Pomiędzy wartościami 10000 i 20000 jest już jednak niewielka różnica, co wskazuje na efekt saturacji – dalsze zwiększanie parametru prawdopodobnie nie poprawiłoby znacząco wyniku. Również rozrzut wyników zmniejsza się wraz z maksymalną liczbą iteracji, co wynika z redukcji losowości w znajdowanym najlepszym wyniku.

Macierz 76

Średni wynik: 154578,31

Czas wykonania wszystkich kombinacji: 266,953 sekundy

Dziesięć najlepszych wyników:

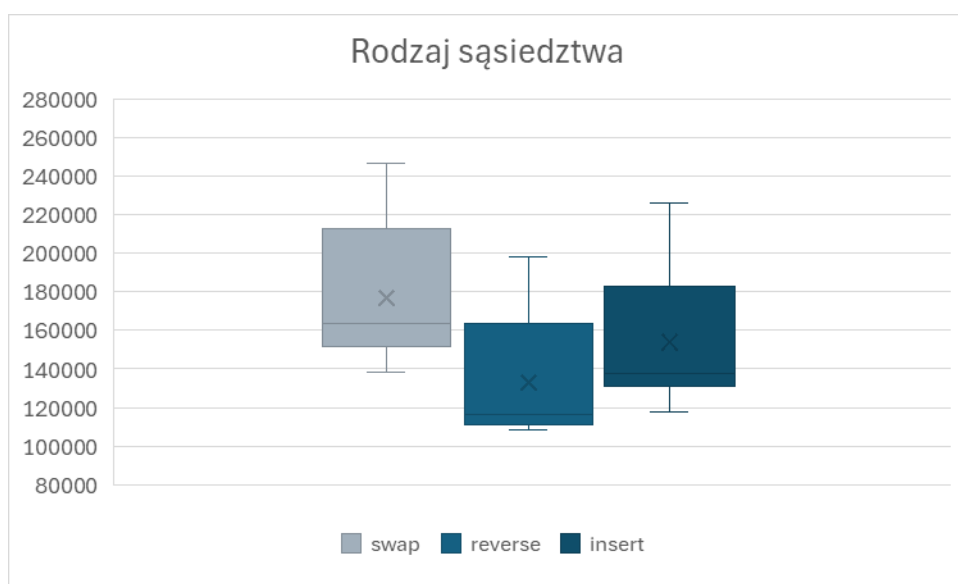
Najlepszy Koszt	Rodzaj Sąsiedztwa	Liczba Restartów	Maks. Iteracje Bez Poprawy	Maksymalna liczba Iteracji
108644,97	reverse	500	2500	20000
109278,46	reverse	500	5000	20000
109332,96	reverse	100	5000	20000
109335,25	reverse	500	10000	20000
109540,74	reverse	20	5000	20000
109792,04	reverse	100	10000	20000
109873,27	reverse	50	10000	20000
110013,16	reverse	100	2500	20000
110435,38	reverse	500	5000	10000
110715,36	reverse	50	2500	20000

Najlepsza trasa:

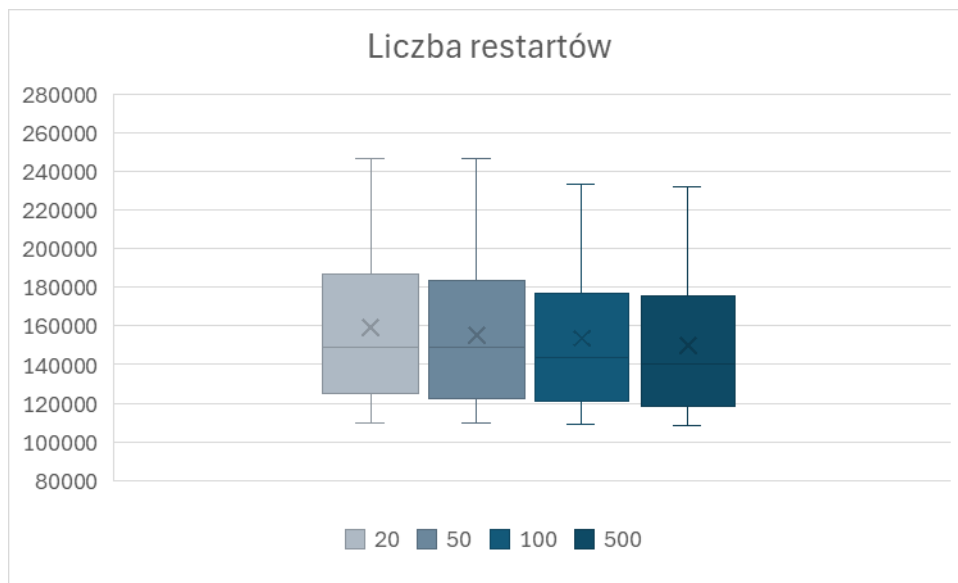
51 → 66 → 65 → 71 → 72 → 73 → 64 → 63 → 62 → 61 → 41 → 60 → 59 → 58 → 57 → 56 → 55 → 52 → 53 → 54 → 42 → 43 → 27 → 26 → 30 → 19 → 31 → 29 → 28 → 33 → 32 → 35 → 34 → 40 → 39 → 38 → 36 → 37 → 18 → 17 → 16 → 15 → 74 → 14 → 13 → 12 → 11 → 10 → 9 → 8 → 7 → 6 → 5 → 20 → 4 → 3 → 2 → 75 → 76 → 1 → 23 → 22 → 21 → 25 → 24 → 46 → 45 → 44 → 48 → 47 → 69 → 68 → 70 → 67 → 50 → 49

Czas wykonania dla najlepszej trasy: 7,77 sekundy

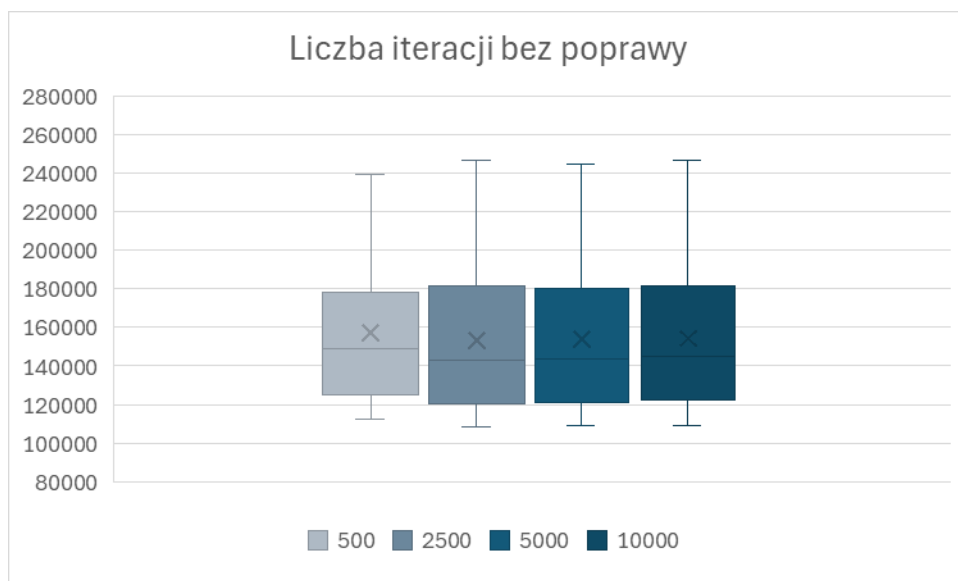
Wpływ parametrów na wyniki:



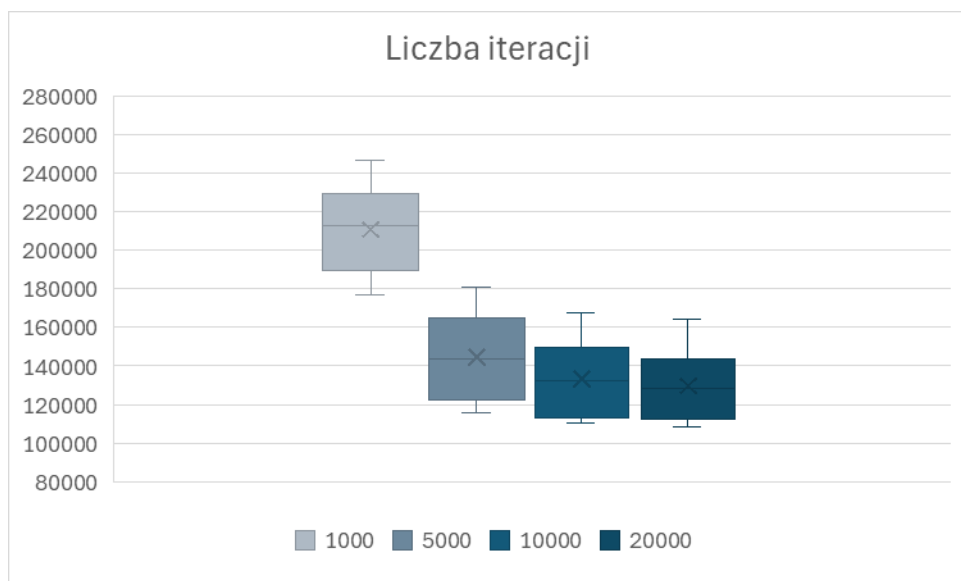
Ponownie zauważyć można najniższe wyniki dla ruchu “reverse”, pośrednie dla ruchu “insert” i najgorsze dla ruchu “swap”.



Mniej zauważalna jest tu odwrotna zależność między liczbą restartów a długością trasy, którą można było dostrzec przy Macierzy 48. Wyniki wraz ze wzrostem liczby restartów polepszają się nieznacznie, co może sugerować słaby, acz istniejący wpływ tego parametru na wyniki.



Ponownie maksymalna liczba iteracji bez poprawy zdaje się nie mieć wpływu na wyniki. Dla najmniejszej liczby iteracji bez poprawy (500) średnia długość trasy jest największa, jednak nie jest to znacząca zależność.



Zdecydowanie najgorsze wyniki osiągnęte są dla najmniejszej liczby maksymalnych iteracji (1000). Wraz ze wzrostem liczby iteracji średnia długość trasy zmniejsza się, choć różnica między wartością 10000 i 20000 nie jest już tak znacząca.

Macierz 127

Średni wynik: 207370,27

Czas wykonania wszystkich kombinacji: 389,668 sekundy

Dziesięć najlepszych wyników:

Najlepszy Koszt	Rodzaj Sąsiedztwa	Liczba Restartów	Maks. Iteracje Bez Poprawy	Maksymalna liczba Iteracji
125552,3	reverse	500	2500	20000
126458,11	reverse	500	5000	20000
127629,66	reverse	100	5000	20000
128601,57	reverse	100	2500	20000
129004,69	reverse	50	10000	20000
129411,57	reverse	20	10000	20000
129428,67	reverse	500	10000	20000
129456,98	reverse	100	10000	20000
130516,16	reverse	50	5000	20000
130615,27	reverse	50	2500	20000

Najlepsza trasa:

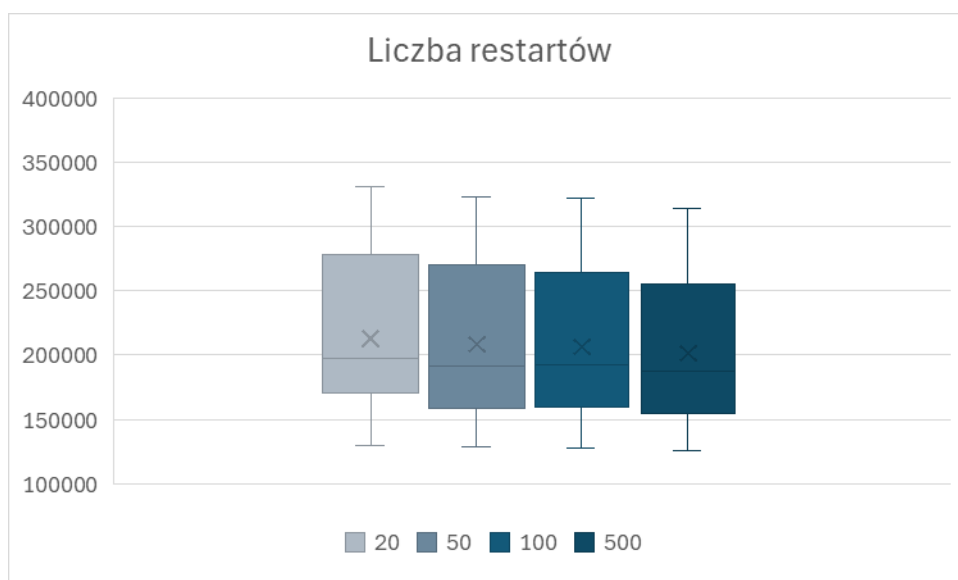
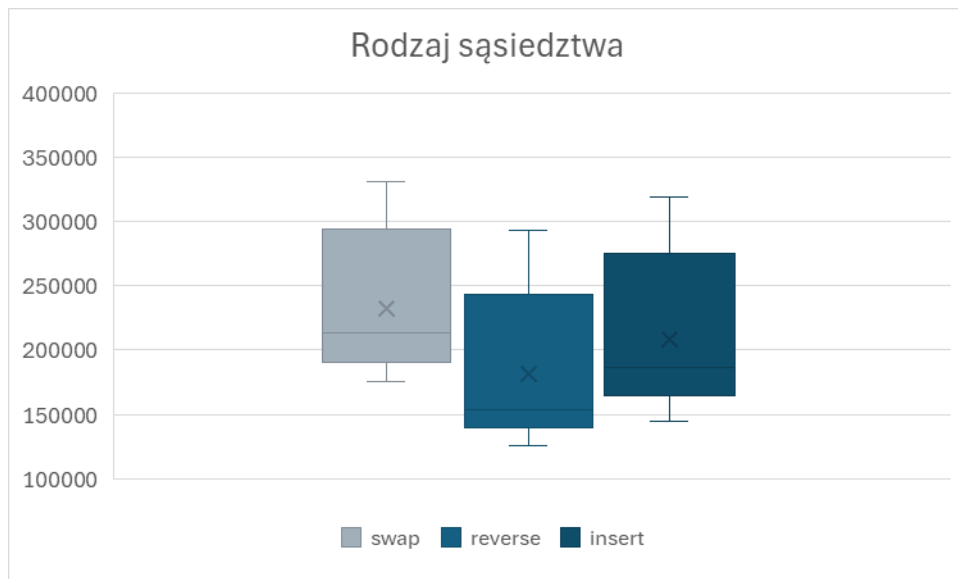
39 → 34 → 43 → 30 → 41 → 37 → 36 → 35 → 40 → 95 → 123 → 97 → 98 → 101 → 102 → 83 → 82 → 126 → 81 → 84 → 117 → 78 → 76 → 74 → 18 → 77 → 79 → 80 → 20 → 108 → 15 → 106 → 6 → 23 → 19 → 17 → 21 → 22 → 4 → 24 → 9 → 8 → 72 → 67 → 73 → 68 → 71 → 70 → 69 → 75 → 63 → 119 → 96 → 109 → 88 → 87 → 86 → 85 → 110 → 104 → 125 → 89 → 92 → 99 → 65 → 113 → 66 → 55 → 47 → 49 → 53 → 48 → 118 → 46 → 94 → 112 → 111 → 107 → 127 →

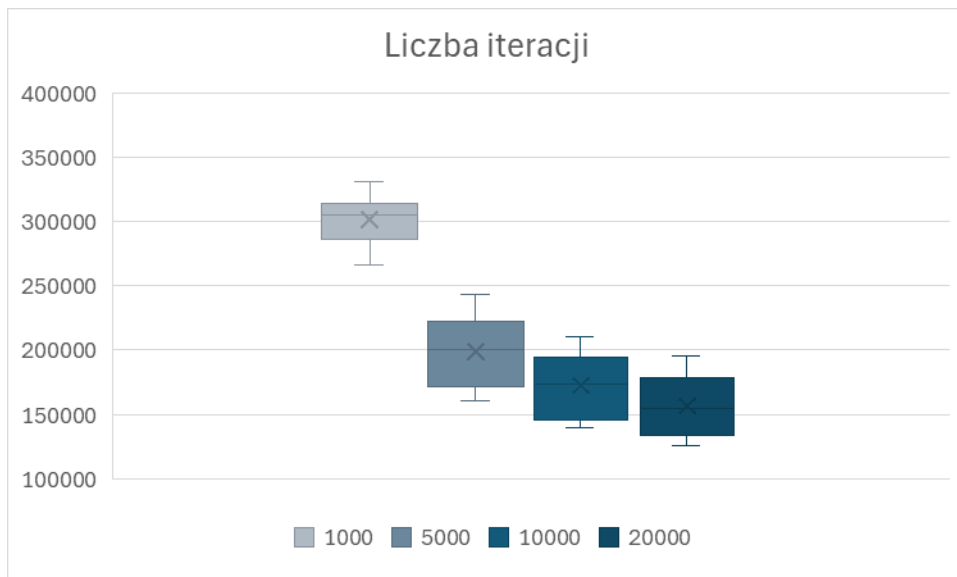
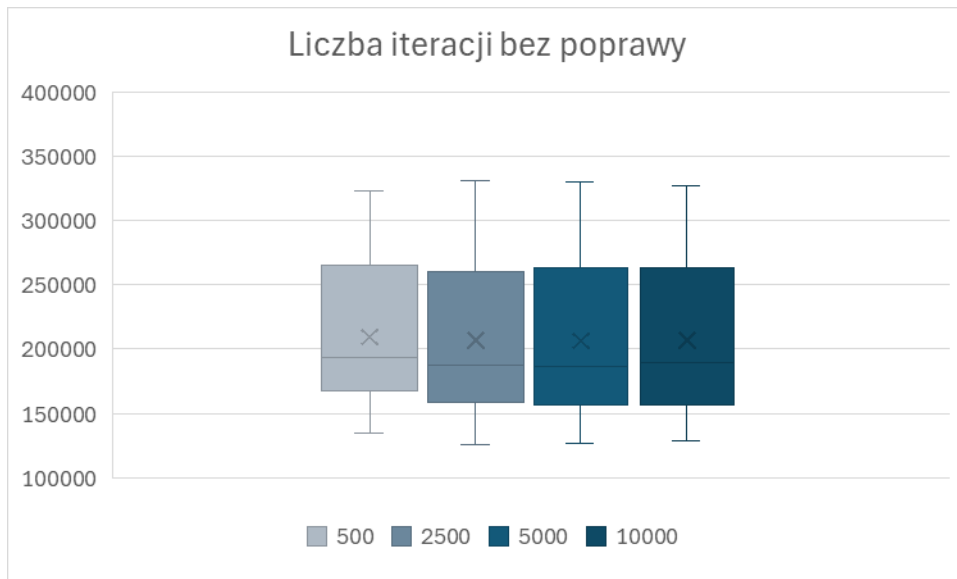
93 → 103 → 44 → 45 → 54 → 57 → 121 → 56 → 124 → 52 → 5 → 51 → 2 → 50 → 115 → 13 → 7 → 120 → 10 → 100 → 64 → 58 → 91 → 60 → 59 → 62 → 61 → 90 → 116 → 3 → 11 → 114 → 105 → 1 → 16 → 14 → 12 → 31 → 27 → 26 → 38 → 25 → 33 → 29 → 32 → 28 → 122 → 42

Czas wykonania dla najlepszej trasy: 13,752 sekundy

Wpływ parametrów na wyniki:

Wnioski dla poniższych wykresów pokrywają się z tymi wyciągniętymi na podstawie wyników dla Macierzy 48 i 76, nie będą więc ponownie powtarzane.





Podsumowanie

Analiza wyników algorytmu dla trzech macierzy o różnych rozmiarach (48, 76, 127) pozwala stwierdzić, że rodzaj sąsiedztwa "reverse" jest zdecydowanie najefektywniejszym ruchem. W każdym przypadku osiągał on najniższe koszty tras, charakteryzując się również największą stabilnością wyników. Pozostałe rodzaje ruchów, takie jak "insert" i "swap", wypadały gorzej, szczególnie ten ostatni, który generował najwyższe wartości funkcji celu oraz największy rozrzut wyników.

Wzrost liczby restartów wyraźnie poprawia jakość rozwiązań i stabilność algorytmu, co szczególnie widoczne jest dla najmniejszej macierzy. Większa liczba restartów pozwala algorytmowi unikać utknięcia w lokalnych minimach i zwiększa szanse na znalezienie globalnie najlepszego rozwiązania. W przypadku większych macierzy wpływ liczby restartów na wynik staje się mniej wyraźny, ale nadal jest obecny.

Z kolei maksymalna liczba iteracji bez poprawy okazuje się parametrem o ograniczonym wpływie na jakość wyników. Chociaż niższe wartości tego parametru (np. 500 iteracji bez poprawy) prowadzą do nieco gorszych rozwiązań, różnice są niewielkie. Ważniejszym czynnikiem jest maksymalna liczba iteracji – większe wartości tego parametru prowadzą do istotnego skrócenia tras, choć dla 20000 iteracji poprawa jest niewielka, co sugeruje osiągnięcie efektu saturacji.

SA – symulowane wyżarzanie

Wprowadzenie

Symulowane wyżarzanie (ang. Simulated Annealing, SA) to metaheurystyczna metoda optymalizacji inspirowana fizycznym procesem wyżarzania materiałów. Cechą charakterystyczną algorytmu jest wykorzystywanie temperatury jako parametru sterującego. SA jest metodą iteracyjną – rozwiązanie wylosowane na początku jest modyfikowane w kolejnych krokach, w przypadku poprawy wyniku wybieramy zawsze lepsze rozwiązanie, ale z pewnym prawdopodobieństwem może być również zaakceptowane rozwiązanie gorsze. W przypadku problemu komiwojażera (TSP), SA stara się znaleźć optymalną kolejność odwiedzania miast przy minimalnym koszcie (odległości) podróży.

Parametry i ich wartości

Zaimplementowany algorytm SA wykorzystuje pięć parametrów:

- Początkowa temperatura (*initial_temperatures*) – określa początkową wartość temperatury. Wyższa temperatura oznacza zazwyczaj bardziej chaotyczne wartości rozwiązań, co zmniejsza ryzyko utkwienia w lokalnym minimum na początkowym etapie.
- Sposób obniżania temperatury – w zaimplementowanym algorytmie aktualna wartość temperatury jest mnożona przez współczynnik redukcji temperatury (*alphas*), który określa tempo obniżania temperatury w kolejnych iteracjach. Powolne chłodzenie (niskie wartości) pozwala na lepszą eksplorację, ale zwiększa czas działania algorytmu.
- Liczba iteracji wykonywanych dla jednej temperatury (*n_iter_per_temps*) – parametr kontrolujący ile prób algorytm wykona zanim przejdzie do obniżenia temperatury.
- Maksymalna liczba iteracji bez poprawy (*max_iter_without_improvements*) - limit iteracji, po którym algorytm zakończy działanie, jeśli nie znajdzie lepszego rozwiązania.
- Rodzaj sąsiedztwa (*neighbourhood_types*) – parametr określający sposób generowania nowych rozwiązań na podstawie aktualnego rozwiązania.

Parametr	Wartość			
Początkowa temperatura	500	1000	1500	2000
Sposób obniżania temperatury	0.85	0.90	0.95	0.98
Liczba iteracji wykonywanych dla jednej temperatury	50	100	150	200
Maksymalna liczba iteracji bez poprawy	20	50	100	150
Rodzaj sąsiedztwa	swap	insert	reverse	

Parametry te dają możliwość utworzenia 768 kombinacji.

Działanie programu

Program wczytuje macierz odległości z pliku Excel, gdzie wiersze i kolumny reprezentują miasta, a wartości - odległości między nimi. Funkcja *route_length* oblicza całkowitą długość trasy, biorąc pod uwagę odległości między kolejnymi miastami oraz powrót do miasta początkowego. Funkcja *generate_neighbor* tworzy nową trasę na podstawie obecnej za pomocą przeszukiwania trzech rodzajów ruchów: swap (zamiana miast), reverse (odwrócenie fragmentu trasy) lub insert (przeniesienie miasta w inne miejsce). Algorytm iteracyjnie optymalizuje trasę stopniowo obniżając temperaturę i akceptując rozwiązania lepsze lub spełniające warunek akceptacji (zgodnie z regułą Boltzmanna). Program testuje różne kombinacje parametrów wielokrotnie (10 razy) i dla każdej z nich zapisuje do pliku excel wyniki takie jak: minimalna i średnia odległość, średni czas wykonywania algorytmu oraz trasę, która dała najlepszy wynik.

Macierz 48

10 najlepszych wyników:

Temperatura Początkowa	Redukcja Temperatury	Iteracje na Temperaturze	Maksymalna Liczba Iteracji bez Poprawy	Rodzaj Sąsiedztwa	Długość Najlepszej Trasy	Średnia Długość Trasy
2000	0,98	100	150	reverse	10933	23970,71
1000	0,98	100	150	reverse	10955	20340,66
500	0,98	150	150	reverse	10982	18180,32
1000	0,98	200	150	insert	11003	21423,1
1500	0,95	50	150	reverse	11030	19696,56
2000	0,85	200	150	reverse	11042	23996,31
1000	0,85	150	150	reverse	11047	19496,19
2000	0,9	150	150	reverse	11053	22599,06
1000	0,98	150	150	reverse	11067	20938,28
2000	0,9	150	100	reverse	11076	23014,53

Długość najkrótszej trasy wynosi: 10933.

Najkrótsza trasa: [2, 29, 41, 16, 22, 3, 34, 14, 25, 13, 23, 11, 12, 15, 40, 1, 8, 9, 38, 31, 44, 18, 7, 36, 43, 17, 27, 19, 37, 6, 30, 28, 46, 33, 20, 47, 21, 32, 39, 48, 5, 42, 10, 24, 45, 35, 26, 4].

Czas wykonywania algorytmu dla macierzy z 48 miastami wynosi około 620,24 sekund.

Macierz 76

10 najlepszych wyników:

Temperatura Początkowa	Redukcja Temperatury	Iteracje na Temperaturze	Maksymalna Liczba Iteracji bez Poprawy	Rodzaj Sąsiedztwa	Długość Najlepszej Trasy	Średnia Długość Trasy
1500	0,98	100	150	reverse	121687,9	185469,9
2000	0,85	200	150	reverse	123707,3	195776,7
1000	0,95	150	150	reverse	124159,6	175366,7
1500	0,98	100	150	reverse	124553,6	180482,4
1000	0,98	150	150	reverse	125094,9	181366,9

1000	0,98	150	150	reverse	125246,3	173598,7
1000	0,85	150	150	reverse	125416,9	187375,8
1500	0,95	200	150	reverse	125727	185038,6
1500	0,98	200	150	reverse	126184,1	184715,7
1000	0,95	200	150	reverse	126669,6	186427,7

Długość najkrótszej trasy wynosi: 121687,9.

Najkrótsza trasa: [37, 33, 32, 29, 28, 27, 26, 30, 31, 19, 20, 10, 9, 6, 8, 7, 5, 4, 3, 75, 76, 2, 1, 23, 22, 21, 25, 24, 46, 45, 44, 48, 47, 68, 70, 69, 67, 50, 49, 43, 42, 54, 53, 52, 51, 65, 66, 55, 56, 71, 72, 73, 64, 63, 62, 61, 57, 58, 59, 41, 60, 40, 39, 38, 34, 35, 36, 16, 15, 74, 13, 14, 12, 11, 17, 18].

Czas wykonywania algorytmu dla macierzy z 76 miastami wynosi około 398,72 sekund.

Macierz 127

10 najlepszych wyników:

Temperatura Początkowa	Redukcja Temperatury	Iteracje na Temperaturze	Maksymalna Liczba Iteracji bez Poprawy	Rodzaj Sąsiedztwa	Długość Najlepszej Trasy	Średnia Długość Trasy
1500	0,98	200	150	reverse	145070,2	227571
1500	0,98	200	150	reverse	147457,3	241112,5
2000	0,98	200	150	reverse	148808,8	241490,6
1500	0,98	200	150	reverse	148815,2	230127,2
1000	0,98	150	150	reverse	150079,4	229343,1
2000	0,98	200	150	reverse	150272,2	252996,2
1500	0,98	100	150	reverse	151250,8	240525,4
1000	0,98	150	150	reverse	152737,2	217711
2000	0,98	100	150	reverse	153693,5	250262,4
2000	0,95	200	150	reverse	153923,3	249825,6

Długość najkrótszej trasy wynosi: 145070,2.

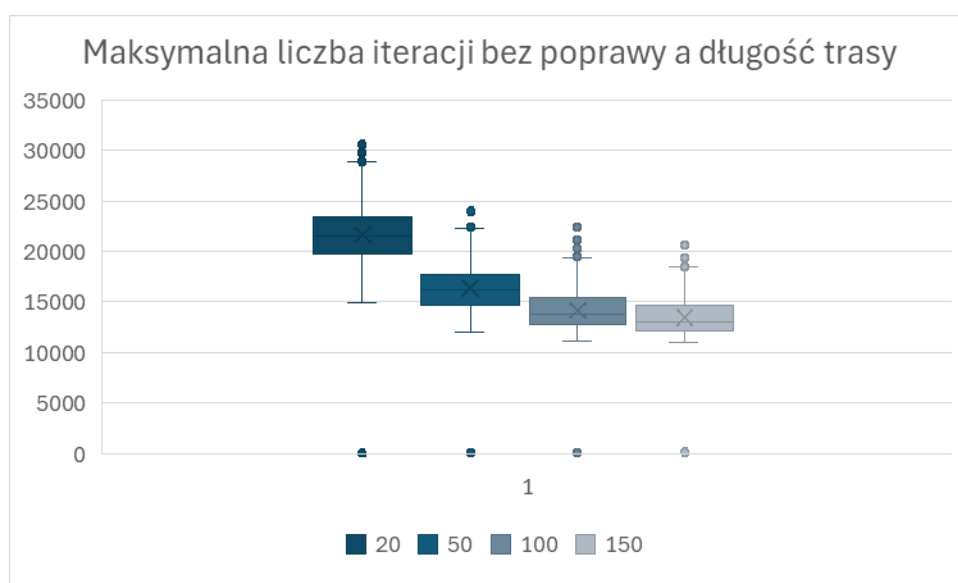
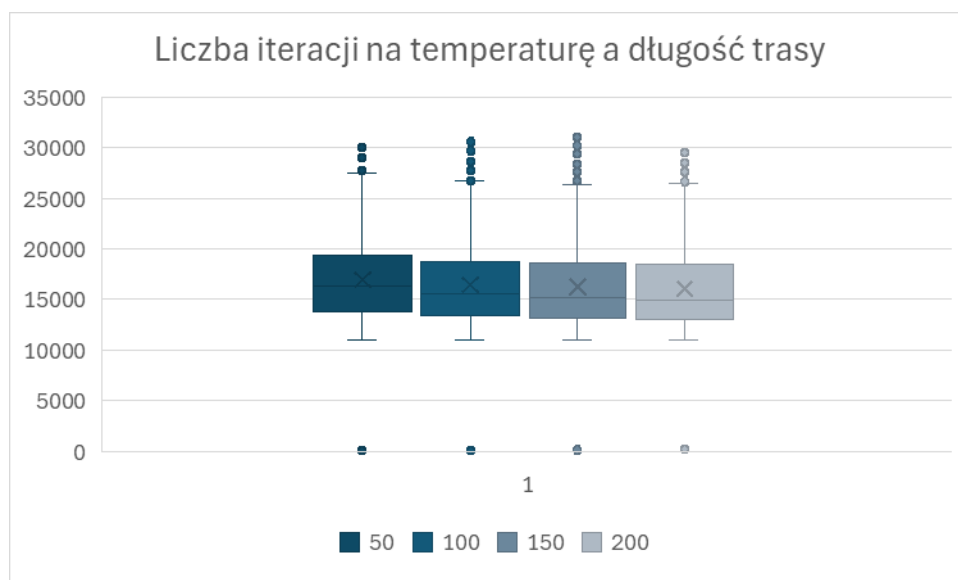
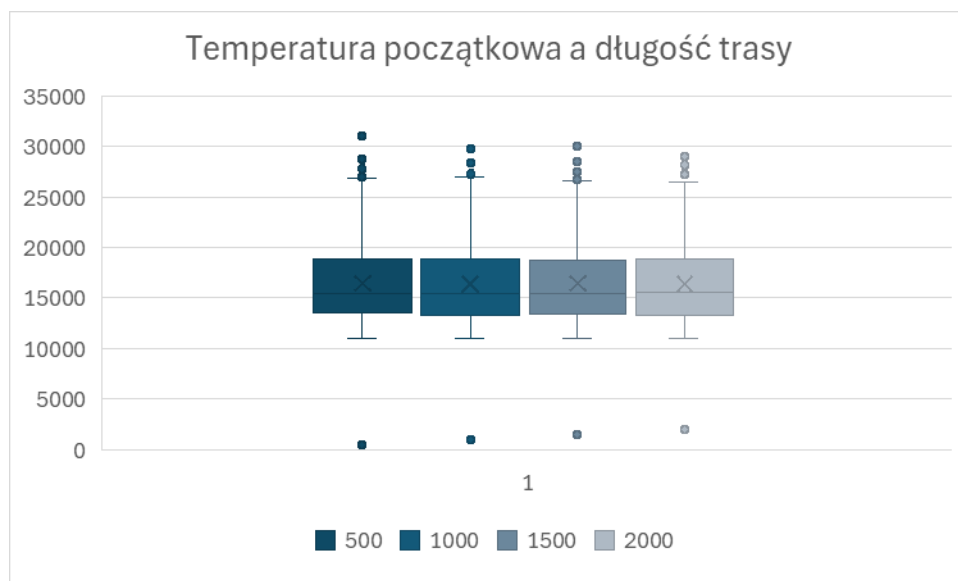
Najkrótsza trasa: [116, 3, 8, 72, 23, 22, 19, 108, 106, 105, 114, 10, 120, 7, 11, 9, 21, 17, 4, 24, 15, 20, 6, 14, 30, 43, 40, 42, 39, 34, 123, 98, 97, 28, 32, 122, 29, 33, 25, 38, 26, 27, 31, 12, 41, 36, 35, 37, 1, 16, 51, 57, 54, 44, 45, 103, 47, 53, 49, 118, 48, 94, 95, 127, 93, 107, 111, 112, 46, 124, 55, 66, 52, 56, 121, 2, 5, 50, 13, 115, 100, 58, 64, 113, 65, 89, 99, 92, 125, 104, 110, 71, 68, 70, 69, 75, 109, 85, 86, 87, 88, 96, 119, 63, 102, 101, 83, 82, 126, 84, 81, 117, 78, 80, 79, 76, 77, 18, 74, 73, 67, 90, 61, 91, 62, 59, 60].

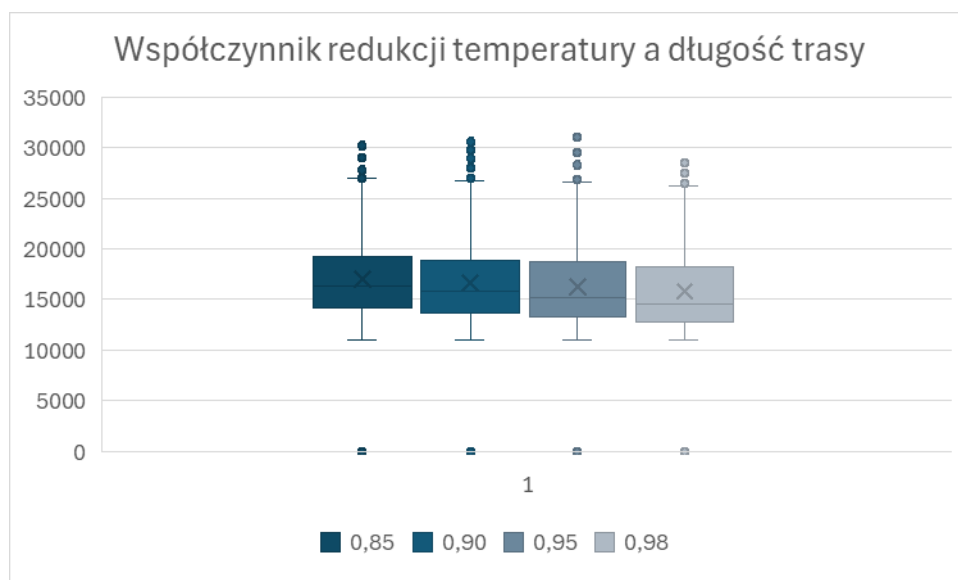
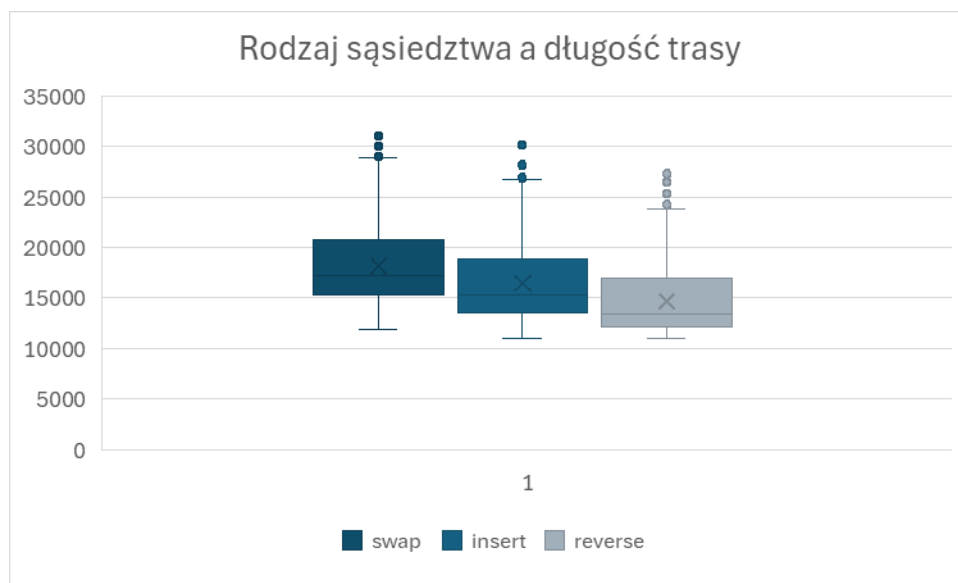
Czas wykonywania algorytmu dla macierzy z 127 miastami wynosi około 1024,73 sekund.

Podsumowanie SA

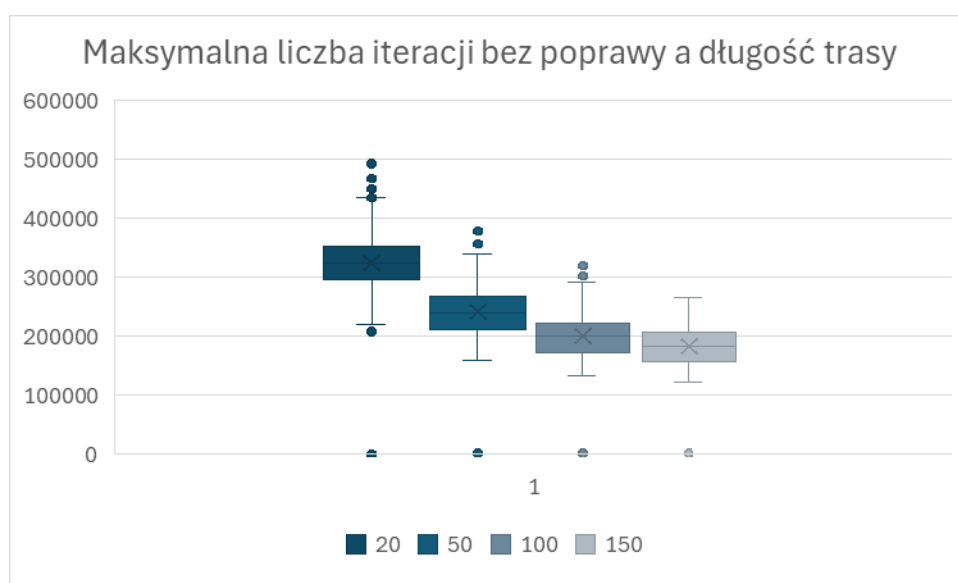
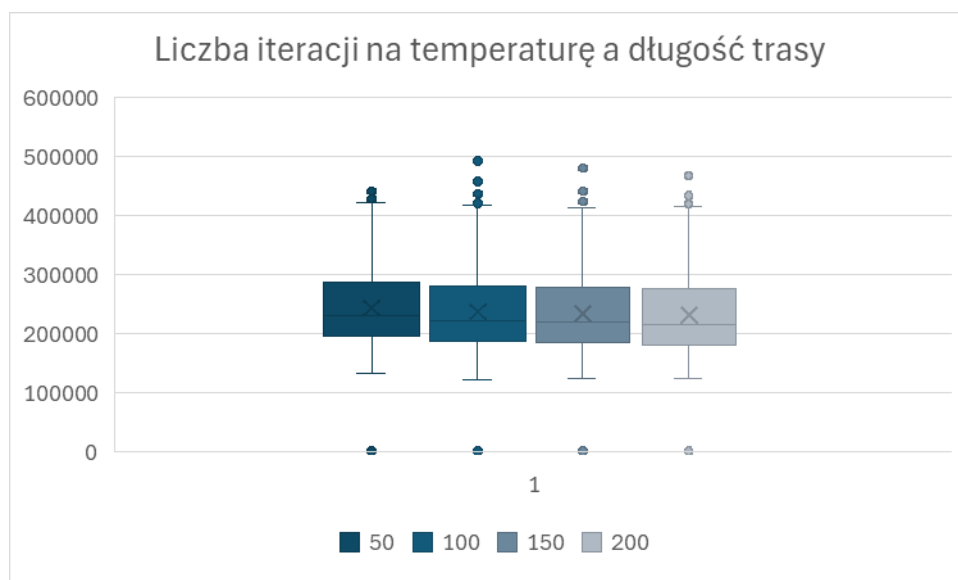
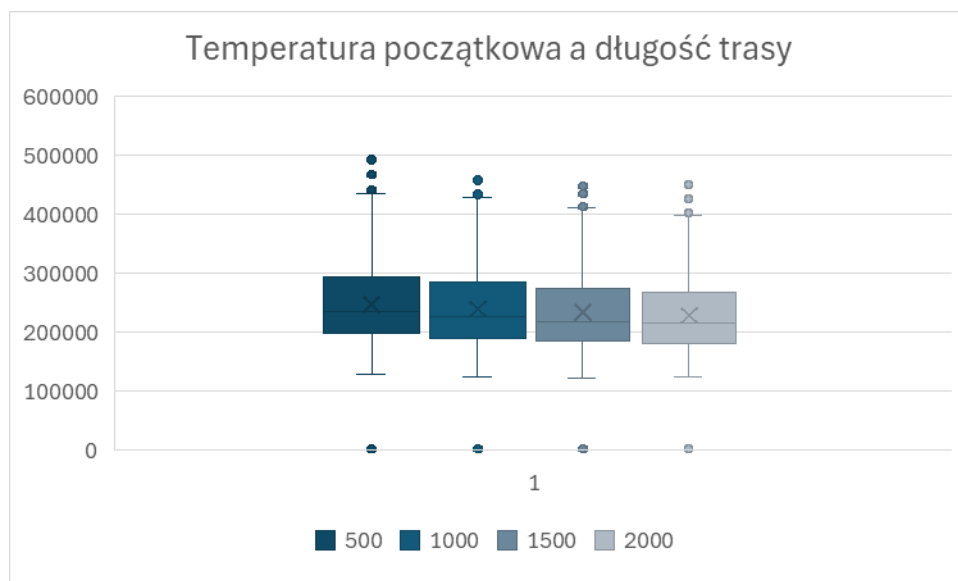
Wpływ poszczególnych parametrów na długość najlepszej trasy:

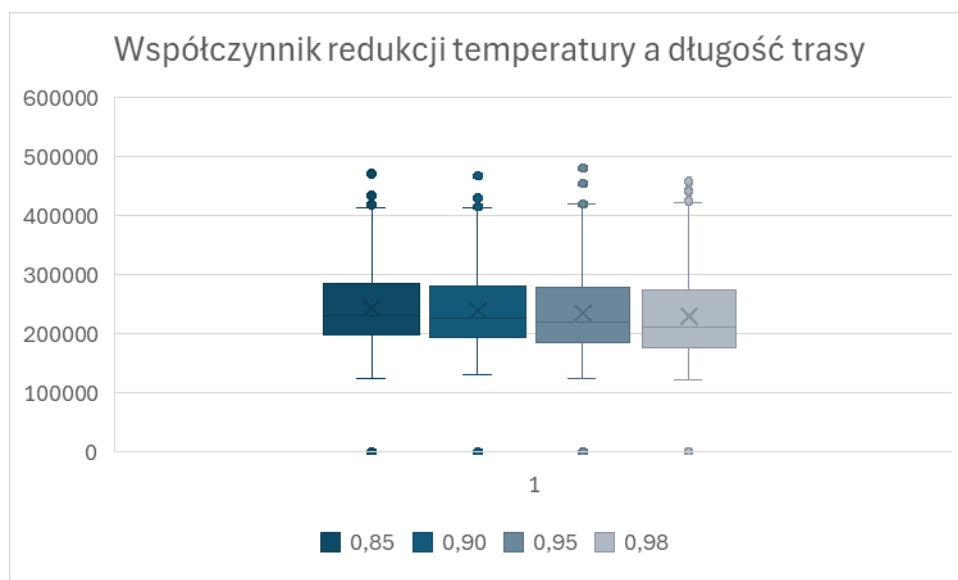
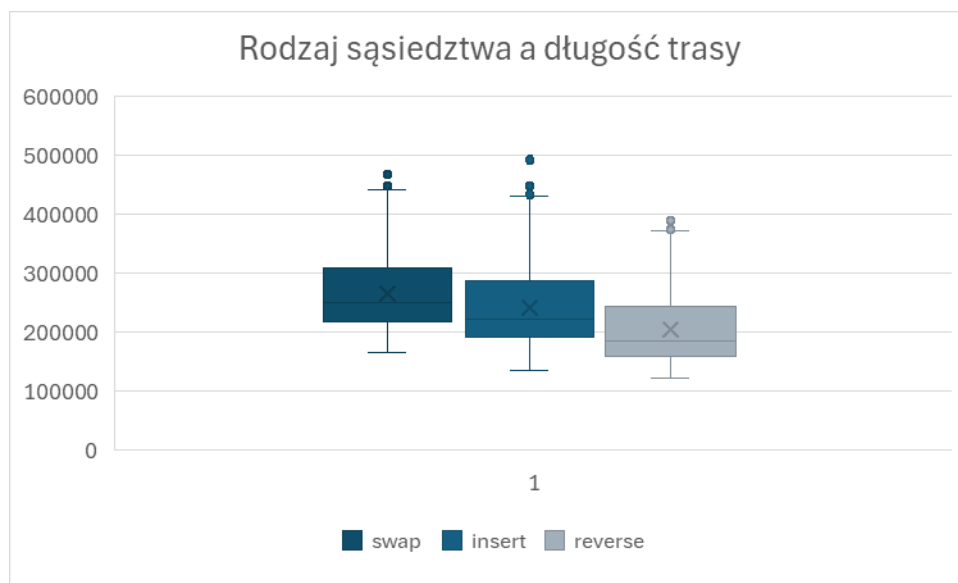
- dla 48 miast:



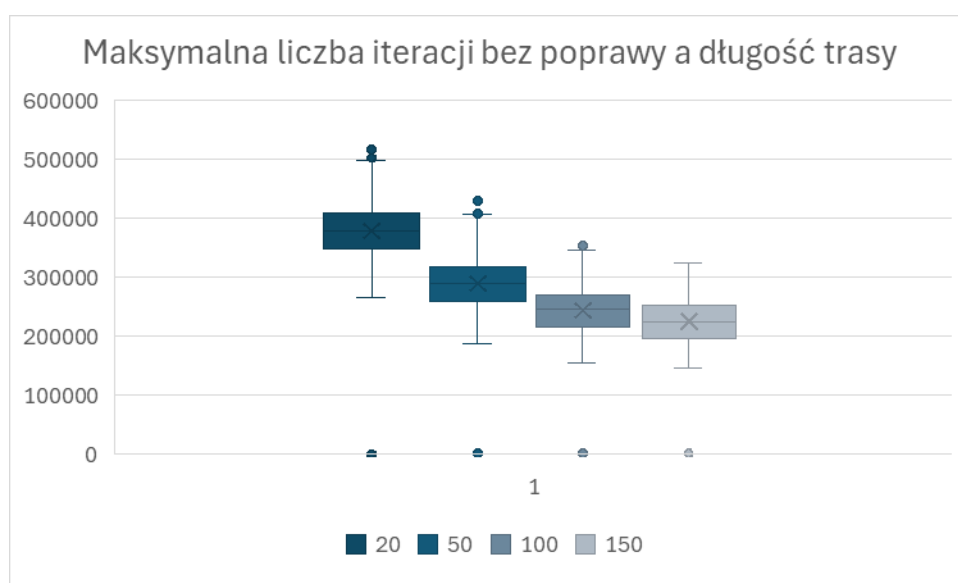
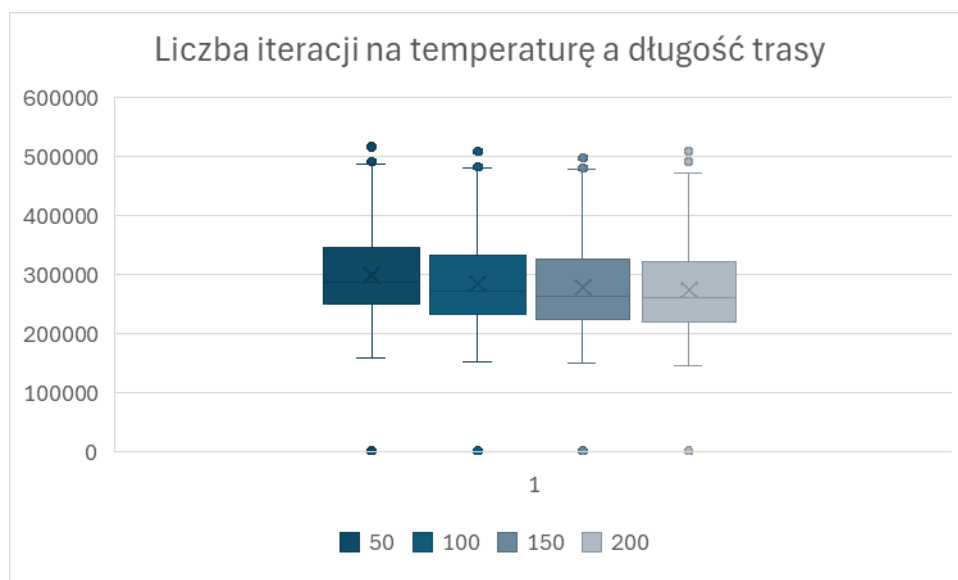
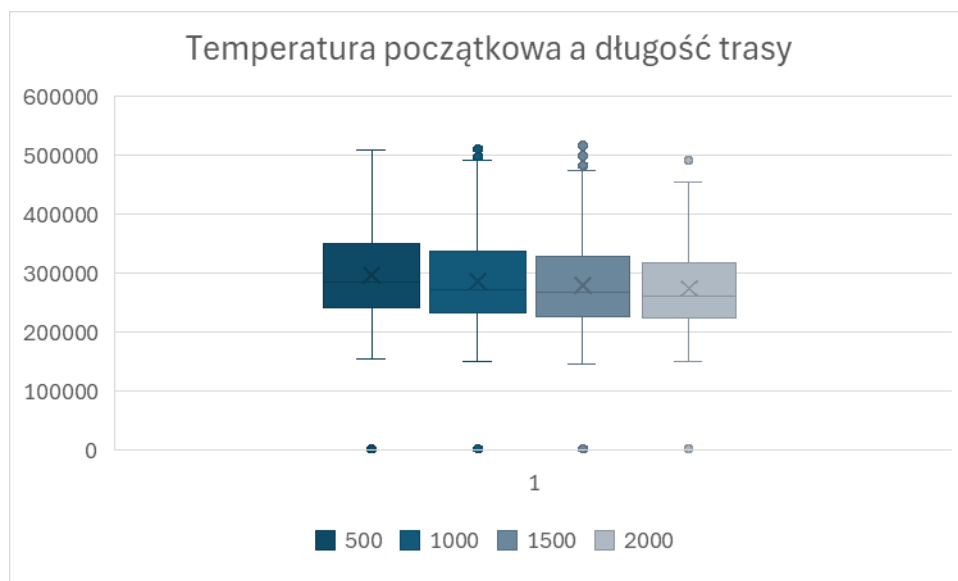


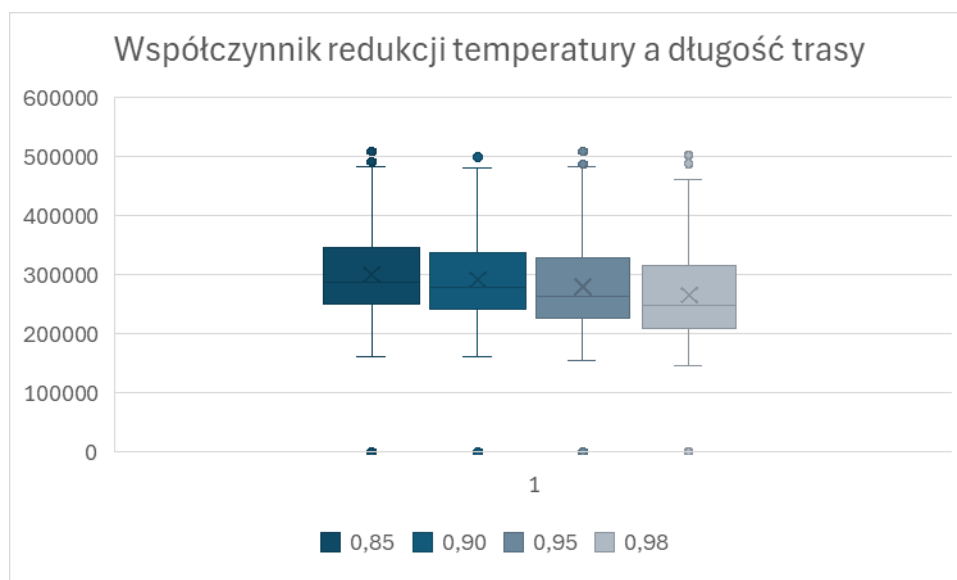
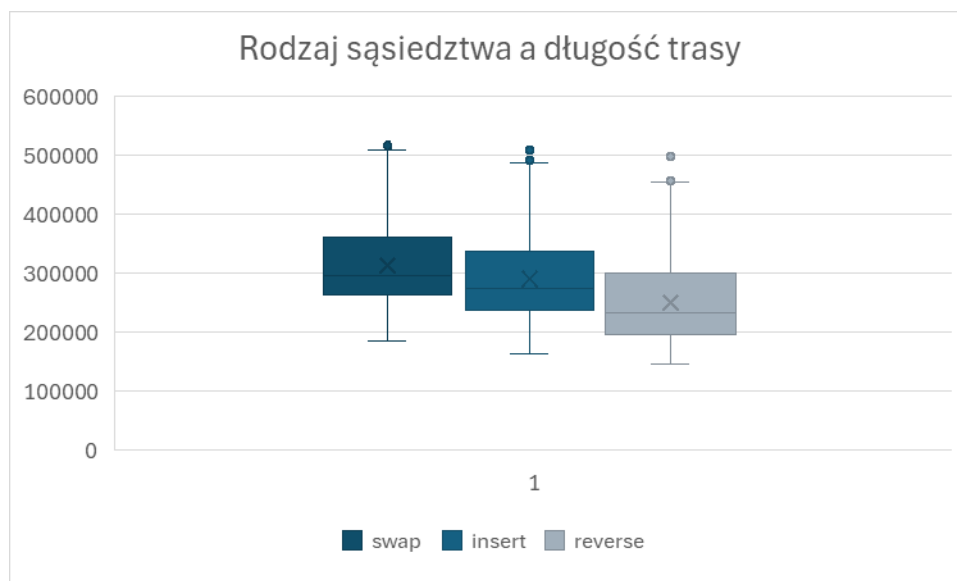
- dla 76 miast:





- dla 127 miast:





Zwiększając temperaturę początkową otrzymujemy lepsze wyniki dla macierzy 76 i 127, dla macierzy 48 wyniki są zbliżone bez względu na wielkość tego parametru. Wzrost liczby iteracji wykonywanych na danej temperaturze oraz wzrost maksymalnej liczby iteracji bez poprawy prowadzi do uzyskania lepszych wyników dla każdej macierzy odległości. Spośród wyboru rodzajów sąsiedztwa najlepiej sprawdza się metoda reverse dla każdego zestawu danych. Z kolei badając wpływ redukcji temperatury można zauważyć, że im wyższy współczynnik α , tym lepsze wyniki otrzymujemy.

Algorytm mrówkowy

Wprowadzenie

Jak sama nazwa wskazuje algorytm jest inspirowany sposobem, w jaki mrówki w naturze znajdują najkrótsze ścieżki do pożywienia. W tym procesie mrówki poruszają się, pozostawiając na trasie ślady feromonowe. Im częściej dana ścieżka jest używana przez kolejne mrówki, tym więcej feromonów się na niej gromadzi, co zwiększa prawdopodobieństwo, że następne mrówki również ją wybiorą. W miarę

upływu czasu ścieżki, rzadziej uczęszczane, tracą feromony wskutek ich naturalnego odparowywania. Mechanizm ten prowadzi do wzmocnienia najkrótszych tras, które stają się bardziej atrakcyjne

Działanie algorytmu

W algorytmie komputerowym ten proces jest odwzorowany w następujący sposób. Sztuczne mrówki eksplorują graf reprezentujący przestrzeń poszukiwań. Każda mrówka buduje rozwiązanie, przemieszczając się po grafie na podstawie prawdopodobieństwa zależnego od dwóch czynników: intensywności feromonów na krawędziach oraz lokalnej heurystyki, która ocenia potencjalną jakość wyboru (długość krawędzi w grafie).

Po zakończeniu trasy przez wszystkie mrówki następuje aktualizacja feromonów. Trasy, które przyniosły lepsze rozwiązania (krótsze ścieżki), otrzymują większe wzmocnienie feromonowe, proporcjonalne do jakości rozwiązania. Jednocześnie feromony na wszystkich krawędziach ulegają częściowemu odparowaniu, co zapobiega zbyt wczesnemu skupieniu się algorytmu na jednym rozwiązaniu i pozwala na eksplorację nowych możliwości. Dzięki mechanizmowi wzmacniania najlepszych rozwiązań oraz odparowywania feromonów algorytm mrówkowy jest w stanie unikać pułapek lokalnych minimów i skutecznie poszukiwać optymalnych rozwiązań w złożonych problemach optymalizacyjnych.

Działanie kodu

Algorytm testuje różne kombinacje zmiennych parametrów, aby znaleźć optymalne ustawienia. Wyniki działania są zapisywane do pliku Excel, a na koniec wyświetlana jest najlepsza znaleziona trasa i jej długość.

W kodzie testowane są różne kombinacje parametrów, aby znaleźć optymalne ustawienia:

- Liczba mrówek (numAnts) – określa liczbę mrówek uczestniczących w budowie tras w każdej iteracji algorytmu. Każda mrówka generuje jedną trasę na podstawie poziomów feromonów i heurystyki.
- Liczba iteracji (numIterations) - Oznacza liczbę iteracji wewnętrznych algorytmu dla danej konfiguracji parametrów. W każdej iteracji mrówki budują trasy, a poziomy feromonów są aktualizowane.
- Początkowy poziom feromonów (initialPheromone) - Początkowa ilość feromonów na każdej krawędzi grafu. Jest to wartość, od której algorytm rozpoczyna działanie.
- Wzmocnienia feromonów (Q) - Wartość, która określa ilość feromonów, jakie są dodawane do krawędzi na podstawie jakości rozwiązania. Feromony są proporcjonalne do odwrotności długości trasy znalezionej przez mrówkę.
- Typ sąsiedztwa (neighborhood) - typ operacji modyfikującej ścieżkę w celu lokalnej optymalizacji

Zdecydowano się na zastosowanie następujących wartości:

- List<int> numAntsList = { 100, 200, 500, 700 };
- List<int> numIterationsList = { 50, 100, 200, 500 };
- List<double> initialPheromoneList = { 1.0, 2.0, 3.0, 4.0 };
- List<double> QList = { 50.0, 100.0, 150.0, 200.0 };
- List<string> neighborhoodTypes = { "reverse", "swap", "insert" };

Aby przedstawić wyniki w przejrzysty sposób, zdecydowano się na analizę wszystkich możliwych wartości jednego parametru, przy jednoczesnym ustaleniu stałych wartości dla pozostałych. Stałe wartości dla parametrów, które nie są w danym momencie poddane analizie to:

- Liczba mrówek = 500;
- Liczba iteracji = 100;
- Początkowy poziom feromonów = 1;
- Wzmocnienia feromonów = 100;
- Typ sąsiedztwa = reverse;

Pozostałe parametry algorytmu zostały ustawione jako stałe. Są to:

- Alpha ($\alpha=1.0$): Współczynnik wpływu feromonów – określa, jak dużą rolę odgrywa ślad feromonowy pozostawiony przez inne mrówki w procesie wyboru ścieżki. Wyższa wartość α zwiększa znaczenie feromonów w decyzjach.
- Beta ($\beta=2.0$): Współczynnik wpływu heurystyki – definiuje, jak bardzo heurystyka (np. odległość między wierzchołkami) wpływa na wybór ścieżki. Większa wartość β oznacza większe znaczenie czynnika heurystycznego.
- EvaporationRate ($r=0.5$): Współczynnik parowania feromonów – kontroluje szybkość zaniku feromonów na ścieżkach. Niższe wartości sprzyjają dłuższemu utrzymywaniu śladów, natomiast wyższe powodują szybsze ich zanikanie, co zapobiega nadmiernej eksploatacji tych samych tras.

Dla uniknięcia losowości każda kombinacja parametrów została wykonana 20 razy. W tabelach z prezentowanymi wynikami średnia obliczana jest dla tych prób, a najlepszy koszt to najkrótsza trasa z tych dwudziestu powtórzeń

Macierz 48

Liczba mrówek

Liczba mrówek	Najlepszy koszt	Średnia odległość
100	11149	11448,25
200	11204	11403,25
500	10886	11222,75
700	10953	11205,3

Najniższy koszt wystąpił dla 500 mrówek i jest to 10886. Podczas analizy średnich odległości można zauważyć, że wraz ze wzrostem liczby mrówek następuje delikatne zmniejszenie się kosztu jednak trend ten jest dość nieregularny i tym przypadku najkrótsza trasa występuje dla 500, a nie 700 mrówek.

Trasa najlepszej ścieżki:

9 -> 23 -> 41 -> 4 -> 47 -> 38 -> 31 -> 20 -> 46 -> 19 -> 32 -> 14 -> 39 -> 8 -> 0 -> 7 -> 37 -> 30 -> 43 -> 17 -> 6 -> 27 -> 5 -> 36 -> 18 -> 26 -> 42 -> 16 -> 29 -> 35 -> 45 -> 11 -> 10 -> 22 -> 12 -> 24 -> 13 -> 33 -> 2 -> 21 -> 15 -> 40 -> 28 -> 1 -> 25 -> 3 -> 34 -> 44

Czas wykonania: 00:04:26.3201434

Liczba iteracji

Liczba iteracji	Najlepsza koszt	Średnia odległość
-----------------	-----------------	-------------------

50	10972	11308,9
100	11027	11296,7
200	10896	11139,55
500	10939	11054,35

W przypadku zmiany liczby iteracji wewnętrznych w algorytmie mrówkowym można zauważyć tą samą zależność co przy zmienianiu liczby mrówek. Koszt zmniejsza się wraz ze zwiększaniem się liczby iteracji. Najlepszy wynik wyszedł dla 200 iteracji jednak najniższa średnia występuje dla 500 iteracji.

Najlepsza ścieżka dla wartości:	Najlepsza odległość
200	10896

Trasa najlepszej ścieżki:

38 -> 31 -> 20 -> 12 -> 24 -> 13 -> 22 -> 10 -> 46 -> 19 -> 32 -> 45 -> 35 -> 29 -> 42 -> 16 -> 26 -> 18 -> 36 -> 5 -> 27 -> 6 -> 17 -> 43 -> 30 -> 37 -> 7 -> 0 -> 8 -> 39 -> 14 -> 11 -> 2 -> 21 -> 15 -> 40 -> 33 -> 28 -> 1 -> 25 -> 3 -> 34 -> 44 -> 23 -> 9 -> 41 -> 47 -> 4

Czas wykonywania: 00:13:47.5777820

Początkowy poziom feromonów

Początkowy poziom feromonów	Najlepszy koszt	Średnia odległość
1	11047	11259,1
2	10861	11243,7
3	10986	11280,9
4	11036	11279,15

Najlepszy wynik wyszedł dla wartości 2 natomiast po analizie średnich odległości, można dojść do wniosku że liczba początkowego poziomu feromonów nie ma znaczącego wpływu na długość trasy.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
2	10861

Najlepsza trasa:

31 -> 38 -> 4 -> 47 -> 41 -> 9 -> 23 -> 44 -> 34 -> 3 -> 25 -> 1 -> 28 -> 33 -> 40 -> 15 -> 21 -> 2 -> 22 -> 10 -> 11 -> 14 -> 39 -> 8 -> 0 -> 7 -> 37 -> 30 -> 43 -> 17 -> 6 -> 27 -> 5 -> 29 -> 42 -> 16 -> 26 -> 18 -> 36 -> 35 -> 45 -> 32 -> 19 -> 46 -> 20 -> 12 -> 13 -> 24

Czas wykonania: 00:05:38.9974655

Wzmocnienie feromonów

Wzmocnienie feromonów	Najlepsza odległość	Średnia odległość
-----------------------	---------------------	-------------------

50	10960	11215,4
100	11095	11284,6
150	11041	11295,85
200	11110	11244,3

Parametr ten również nie jest kluczowy do szukania trasy z najlepszym kosztem. Najkrótsza znaleziona trasa to 10960 dla wartości parametru 50. Średnie natomiast są bardzo podobne dla wszystkich wartości parametru

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
50	10960

Najlepsza trasa

39 -> 8 -> 0 -> 7 -> 37 -> 30 -> 43 -> 17 -> 6 -> 27 -> 35 -> 29 -> 5 -> 36 -> 18 -> 26 -> 42 -> 16 -> 19 -> 32 -> 45 -> 14 -> 11 -> 10 -> 22 -> 13 -> 24 -> 12 -> 46 -> 20 -> 31 -> 38 -> 47 -> 4 -> 41 -> 9 -> 23 -> 44 -> 34 -> 3 -> 25 -> 1 -> 28 -> 33 -> 40 -> 15 -> 21 -> 2

Czas wykonania: 00:05:11.0815062

Zmienienie sąsiedztwa

Zmienianie sąsiedztwa	Najlepsza odległość	Średnia lokalna odległość
reverse	10979	11253,35
swap	11231	11441,25
insert	11156	11445,85

Dla tego parametru najlepiej wypada „reverse”. Posiada on najniższą średnią i to właśnie dla niego została znaleziona najkrótsza trasa, a mianowicie 10979

Najlepsza trasa:

40 -> 15 -> 21 -> 2 -> 22 -> 13 -> 24 -> 12 -> 10 -> 11 -> 14 -> 39 -> 8 -> 0 -> 7 -> 37 -> 30 -> 43 -> 6 -> 17 -> 35 -> 27 -> 5 -> 36 -> 18 -> 26 -> 42 -> 16 -> 29 -> 45 -> 32 -> 19 -> 46 -> 20 -> 38 -> 31 -> 23 -> 9 -> 44 -> 34 -> 3 -> 25 -> 41 -> 1 -> 28 -> 4 -> 47 -> 33

Czas wykonania: 00:04:25.6598246

Podsumowanie

W przypadku tak dobranych wartości dla parametrów algorytmu mrówkowego najlepsza znaleziona trasa

31 -> 38 -> 4 -> 47 -> 41 -> 9 -> 23 -> 44 -> 34 -> 3 -> 25 -> 1 -> 28 -> 33 -> 40 -> 15 -> 21 -> 2 -> 22 -> 10 -> 11 -> 14 -> 39 -> 8 -> 0 -> 7 -> 37 -> 30 -> 43 -> 17 -> 6 -> 27 -> 5 -> 29 -> 42 -> 16 -> 26 -> 18 -> 36 -> 35 -> 45 -> 32 -> 19 -> 46 -> 20 -> 12 -> 13 -> 24

Jej koszt wynosi 10861. Występuje ona dla następujących wartości parametrów: { 500; 100; 2; 200; „reverse” }

Macierz 76

Liczba mrówek

Wartość parametru	Najlepsza odległość	Średnia lokalna odległość
100	116842,3434	119892,6789
200	116988,3049	119328,9693
500	115254,2639	118128,7086
700	115519,8458	117908,3313

Podobnie jak w przypadku macierzy z mniejszą ilością miast, można zaobserwować wpływ liczby mrówek na zmniejszanie się długości trasy. Optymalna wartość parametru to również 500 natomiast średnie odległości ponownie wykazują spadek wraz ze zwiększaniem się wartości parametru.

Najlepsza ścieżka dla wartości:	Najlepsza odległość
500	115254,2639

Najlepsza trasa:

59 -> 40 -> 39 -> 38 -> 37 -> 35 -> 36 -> 17 -> 16 -> 10 -> 11 -> 12 -> 13 -> 73 -> 15 -> 14 -> 8 -> 9 -> 4 -> 19 -> 18 -> 30 -> 29 -> 28 -> 27 -> 42 -> 41 -> 53 -> 52 -> 51 -> 50 -> 65 -> 64 -> 70 -> 71 -> 72 -> 63 -> 62 -> 61 -> 60 -> 58 -> 57 -> 56 -> 55 -> 54 -> 48 -> 49 -> 69 -> 66 -> 67 -> 68 -> 46 -> 47 -> 43 -> 44 -> 45 -> 23 -> 24 -> 20 -> 21 -> 22 -> 0 -> 1 -> 74 -> 75 -> 7 -> 6 -> 5 -> 2 -> 3 -> 25 -> 26 -> 32 -> 31 -> 34 -> 33

Czas wykonania: 00:10:48.6141321

Liczba iteracji

Wartość parametru	Najlepsza lokalna odległość	Średnia lokalna odległość
50	116854,1275	119457,3392
100	116484,3198	118268,805
200	114298,9178	116935,9264
500	114769,7298	116431,0858

W tym przypadku wnioski dotyczące zmieniania parametru liczby iteracji również są takie same jak dla macierzy 48.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
200	114298,9178

Najlepsza trasa:

13 -> 12 -> 11 -> 10 -> 16 -> 17 -> 36 -> 35 -> 37 -> 38 -> 39 -> 33 -> 34 -> 31 -> 32 -> 27 -> 28 -> 25 -> 26 -> 42 -> 41 -> 53 -> 52 -> 51 -> 48 -> 49 -> 50 -> 55 -> 54 -> 57 -> 58 -> 59 -> 40 -> 60 -> 61 -> 56 -> 62 -> 63 -> 72 -> 71 -> 70 -> 64 -> 65 -> 69 -> 66 -> 67 -> 68 -> 46 -> 47 -> 43 -> 44 -> 45 -> 23 -> 24 -> 20 -> 21 -> 22 -> 0 -> 1 -> 74 -> 75 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 18 -> 30 -> 29 -> 19 -> 9 -> 8 -> 15 -> 14 -> 73

Czas wykonywania: 00:46:52.0799582

Początkowy poziom feromonów

Wartość parametru	Najlepsza odległość	Średnia odległość
1	116258,147	118009,8473
2	116273,9781	118533,776
3	116247,3897	118372,2815
4	115535,9802	117914,0473

Nie można tutaj zaobserwować żadnego trendu natomiast zarówno średnia jak i najlepsza odległość występują dla wartości 4.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
4	115535,9802

Najlepsza trasa:

57 -> 54 -> 55 -> 50 -> 65 -> 64 -> 49 -> 48 -> 51 -> 52 -> 53 -> 41 -> 42 -> 27 -> 32 -> 31 -> 28 -> 26 -> 25 -> 29 -> 30 -> 18 -> 19 -> 4 -> 5 -> 6 -> 7 -> 2 -> 3 -> 9 -> 8 -> 11 -> 10 -> 15 -> 14 -> 12 -> 13 -> 73 -> 75 -> 74 -> 1 -> 0 -> 22 -> 21 -> 20 -> 24 -> 23 -> 45 -> 44 -> 43 -> 47 -> 46 -> 68 -> 67 -> 66 -> 69 -> 70 -> 71 -> 72 -> 63 -> 62 -> 56 -> 61 -> 60 -> 58 -> 59 -> 40 -> 39 -> 33 -> 34 -> 35 -> 36 -> 17 -> 16 -> 37 -> 38

Czas wykonania: 00:13:34.5688170

Wzmocnienie feromonów

Wartość parametru	Najlepsza lokalna odległość	Średnia lokalna odległość
50	115458,9471	118201,2706
100	114743,4883	117921,5411
150	115248,1666	118097,9914
200	116384,3316	118390,7301

Najlepsza wartość dla tego parametru to 100. Dla tej właśnie liczby średnia odległość jest najmniejsza i znaleziona została najkrótsza trasa.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
100	114743,4883

Najlepsza trasa:

10 -> 9 -> 8 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1 -> 75 -> 74 -> 0 -> 22 -> 21 -> 20 -> 24 -> 23 -> 45 -> 44 -> 43 -> 47 -> 46 -> 68 -> 67 -> 66 -> 69 -> 70 -> 71 -> 72 -> 63 -> 62 -> 56 -> 57 -> 58 -> 59 -> 40 -> 39 -> 33 -> 34 -> 31 -> 32 -> 27 -> 28 -> 29 -> 30 -> 18 -> 19 -> 25 -> 26 -> 42 -> 41 -> 53 -> 52 -> 51 -> 48 -> 49 -> 50 -> 65 -> 64 -> 55 -> 54 -> 61 -> 60 -> 37 -> 38 -> 35 -> 36 -> 17 -> 16 -> 15 -> 14 -> 13 -> 12 -> 73 -> 11

Czas wykonania: 00:12:48.9963135

Zmniejszenie sąsiedztwa

Wartość parametru	Najlepsza odległość	Średnia odległość
reverse	116044,8903	117943,4132
swap	117587,9145	120676,6937
insert	117848,4349	120194,2625

Z wszystkich rodzajów sąsiedztwa reverse prezentuje najlepsze wyniki. Posiada on wyraźnie najmniejszą średnią i to właśnie dla tej wartości parametru znaleziona została najlepsza trasa.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
reverse	116044,8903

Najlepsza trasa:

37 -> 38 -> 39 -> 40 -> 59 -> 58 -> 57 -> 54 -> 55 -> 64 -> 65 -> 50 -> 49 -> 48 -> 51 -> 52 -> 53 -> 41 -> 42 -> 27 -> 26 -> 25 -> 28 -> 29 -> 30 -> 18 -> 19 -> 4 -> 3 -> 2 -> 7 -> 6 -> 5 -> 8 -> 9 -> 10 -> 11 -> 12 -> 13 -> 14 -> 15 -> 73 -> 75 -> 74 -> 1 -> 0 -> 22 -> 21 -> 20 -> 24 -> 23 -> 45 -> 44 -> 43 -> 47 -> 46 -> 68 -> 67 -> 66 -> 69 -> 70 -> 71 -> 72 -> 63 -> 62 -> 56 -> 61 -> 60 -> 33 -> 32 -> 31 -> 34 -> 35 -> 36 -> 17 -> 16

Czas wykonania: 00:10:50.0462276

Podsumowanie

Najkrótsza wyznaczona odległość wynosiła 114298,9178. Jest to odległość dla trasy: 13 -> 12 -> 11 -> 10 -> 16 -> 17 -> 36 -> 35 -> 37 -> 38 -> 39 -> 33 -> 34 -> 31 -> 32 -> 27 -> 28 -> 25 -> 26 -> 42 -> 41 -> 53 -> 52 -> 51 -> 48 -> 49 -> 50 -> 55 -> 54 -> 57 -> 58 -> 59 -> 40 -> 60 -> 61 -> 56 -> 62 -> 63 -> 72 -> 71 -> 70 -> 64 -> 65 -> 69 -> 66 -> 67 -> 68 -> 46 -> 47 -> 43 -> 44 -> 45 -> 23 -> 24 -> 20 -> 21 -> 22 -> 0 -> 1 -> 74 -> 75 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 18 -> 30 -> 29 -> 19 -> 9 -> 8 -> 15 -> 14 -> 73

Użyte wartości parametrów to : {500, 200, 1, 100, „reverse” }

Macierz 127

Liczby mrówek

Wartość parametru	Najlepsza odległość	Średnia odległość
100	129192,2074	131821,5041
200	127712,7497	130035,6672
500	127345,6314	129278,4593
700	126235,8219	128691,6627

Wyniki dla tej macierzy potwierdzają odwrotnie proporcjonalną zależność między parametrem a odległością ścieżki. W tym przypadku najkrótsza trasa została znaleziona dla 700 mrówek.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
700	126235,8219

Najlepsza trasa:

60 -> 61 -> 59 -> 115 -> 89 -> 58 -> 90 -> 57 -> 63 -> 99 -> 2 -> 9 -> 119 -> 12 -> 114 -> 49 -> 50 -> 1 -> 34 -> 36 -> 35 -> 40 -> 13 -> 11 -> 30 -> 26 -> 29 -> 42 -> 33 -> 38 -> 41 -> 37 -> 25 -> 31 -> 28 -> 32 -> 121 -> 27 -> 24 -> 79 -> 78 -> 76 -> 17 -> 20 -> 16 -> 19 -> 107 -> 23 -> 22 -> 3 -> 21 -> 18 -> 71 -> 7 -> 66 -> 72 -> 73 -> 67 -> 70 -> 69 -> 68 -> 74 -> 75 -> 77 -> 116 -> 83 -> 80 -> 125 -> 81 -> 82 -> 101 -> 100 -> 62 -> 118 -> 95 -> 108 -> 87 -> 86 -> 85 -> 84 -> 109 -> 103 -> 124 -> 88 -> 91 -> 98 -> 64 -> 112 -> 65 -> 54 -> 46 -> 48 -> 52 -> 117 -> 47 -> 45 -> 93 -> 111 -> 110 -> 106 -> 126 -> 97 -> 96 -> 122 -> 94 -> 92 -> 39 -> 43 -> 102 -> 44 -> 53 -> 56 -> 120 -> 55 -> 123 -> 51 -> 4 -> 0 -> 15 -> 104 -> 6 -> 113 -> 10 -> 8 -> 5 -> 105 -> 14

Czas wykonania: 00:28:35.6618184

Liczba iteracji

Wartość parametru	Najlepsza odległość	Średnia odległość
50	126625,76	130770,2784
100	126239,7153	129299,4513
200	126906,8669	128304,3272
500	124264,2947	127123,0758

Wnioski dla tego parametru są takie same jak w przypadku dwóch poprzednich macierzy o mniejszej ilości miast.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
500	124264,2947

Najlepsza trasa:

11 -> 13 -> 40 -> 35 -> 36 -> 34 -> 39 -> 42 -> 33 -> 38 -> 41 -> 32 -> 121 -> 27 -> 31 -> 28 -> 24 -> 25 -> 37 -> 29 -> 26 -> 30 -> 79 -> 78 -> 76 -> 17 -> 20 -> 16 -> 19 -> 107 -> 5 -> 23 -> 22 -> 3 -> 21 -> 18 -> 71 -> 7 -> 8 -> 10 -> 113 -> 104 -> 6 -> 119 -> 9 -> 2 -> 99 -> 57 -> 63 -> 90 -> 60 -> 61 -> 59 -> 115 -> 89 -> 58 -> 66 -> 72 -> 73 -> 67 -> 70 -> 69 -> 68 -> 74 -> 77 -> 75 -> 116 -> 83 -> 80 -> 125 -> 81 -> 82 -> 101 -> 100 -> 62 -> 118 -> 95 -> 108 -> 87 -> 86 -> 85 -> 84 -> 109 -> 103 -> 124 -> 88 -> 91 -> 98 -> 64 -> 112 -> 65 -> 54 -> 46 -> 48 -> 52 -> 117 -> 47 -> 45 -> 93 -> 111 -> 110 -> 106 -> 126 -> 97 -> 96 -> 122 -> 94 -> 92 -> 44 -> 102 -> 43 -> 53 -> 56 -> 120 -> 4 -> 49 -> 12 -> 114 -> 51 -> 123 -> 55 -> 50 -> 1 -> 15 -> 0 -> 105 -> 14

Czas wykonania: 01:17:37.8492630

Początkowy poziom feromonów

Wartość parametru	Najlepsza odległość	Średnia odległość
1	127088,3251	129400,4839
2	126502,8909	129273,8406
3	125802,8599	129072,2214
4	127268,5465	129181,0018

Nie zaobserwowano żadnego wpływu tego parametru na ostateczne wyniki. Średnie jak i znalezione najlepsze odległości są do siebie bardzo zbliżone niezależnie od początkowego poziomu feromonów.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
3	125802,8599

Najlepsza trasa:

30 -> 26 -> 25 -> 37 -> 38 -> 41 -> 29 -> 42 -> 33 -> 39 -> 34 -> 36 -> 35 -> 40 -> 13 -> 15 -> 0 -> 6 -> 119 -> 9 -> 2 -> 99 -> 57 -> 63 -> 90 -> 58 -> 59 -> 61 -> 60 -> 89 -> 115 -> 10 -> 8 -> 113 -> 104 -> 14 -> 105 -> 5 -> 23 -> 22 -> 3 -> 21 -> 18 -> 71 -> 7 -> 66 -> 72 -> 73 -> 67 -> 70 -> 69 -> 68 -> 74 -> 75 -> 77 -> 116 -> 83 -> 80 -> 125 -> 81 -> 82 -> 101 -> 100 -> 62 -> 118 -> 95 -> 108 -> 109 -> 84 -> 85 -> 86 -> 87 -> 103 -> 124 -> 88 -> 91 -> 98 -> 64 -> 112 -> 65 -> 54 -> 123 -> 51 -> 4 -> 120 -> 56 -> 53 -> 44 -> 102 -> 43 -> 1 -> 50 -> 49 -> 12 -> 114 -> 55 -> 46 -> 52 -> 48 -> 117 -> 47 -> 45 -> 93 -> 111 -> 110 -> 106 -> 126 -> 92 -> 94 -> 122 -> 96 -> 97 -> 28 -> 31 -> 27 -> 121 -> 32 -> 24 -> 78 -> 79 -> 76 -> 17 -> 20 -> 16 -> 19 -> 107 -> 11

Czas wykonania: 00:33:34.6643501

Liczba wzmocnienia feromonów

Wartość parametru	Najlepsza lokalna odległość	Średnia lokalna odległość
50	126723,7324	129513,167
100	126608,608	128988,945
150	126963,2411	129732,0924
200	125336,3099	129202,3803

W tym przypadku również nie występuje żadna zależność. Wyniki są do siebie podobne, co oznacza, że liczba wzmocnienia feromonów nie ma znaczenia w wyznaczaniu najlepszej trasy.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
200	125336,3099

Najlepsza trasa:

80 -> 83 -> 116 -> 77 -> 75 -> 74 -> 68 -> 69 -> 70 -> 67 -> 73 -> 72 -> 66 -> 7 -> 71 -> 18 -> 21 -> 3 -> 22 -> 23 -> 5 -> 105 -> 14 -> 107 -> 19 -> 16 -> 20 -> 17 -> 76 -> 78 -> 79 -> 30 -> 26 -> 29 -> 11 -> 13 -> 40 -> 35 -> 36 -> 34 -> 39 -> 42 -> 33 -> 38 -> 41 -> 37 -> 25 -> 24 -> 32 -> 121 -> 27 -> 28 -> 31 -> 82 -> 81 -> 125 -> 101 -> 100 -> 97 -> 96 -> 122 -> 94 -> 92 -> 126 -> 106 -> 110 -> 111 -> 93 -> 45 -> 117 -> 52 -> 46 -> 48 -> 47 -> 44 -> 102 -> 43 -> 53 -> 56 -> 50 -> 1 -> 15 -> 0 -> 6 -> 104 -> 113 -> 10 -> 8 -> 2 -> 89 -> 115 -> 59 -> 61 -> 60 -> 90 -> 57 -> 63 -> 99 -> 9 -> 119 -> 12 -> 114 -> 49 -> 4 -> 120 -> 55 -> 123 -> 51 -> 54 -> 65 -> 112 -> 64 -> 98 -> 91 -> 88 -> 124 -> 103 -> 58 -> 109 -> 84 -> 85 -> 86 -> 87 -> 108 -> 95 -> 118 -> 62

Czas wykonania: 00:33:29.4288807

Zmianienie sąsiedztwa

Wartość parametru	Najlepsza odległość	Średnia odległość
reverse	125308,0577	129137,2232
swap	126971,3858	130591,5317

insert	127618,2538	130538,2475
--------	-------------	-------------

Dla parametru rodzaj sąsiedztwa wnioski pokrywają się z poprzednimi macierzami.

Najlepsza globalna ścieżka dla wartości:	Najlepsza globalna odległość
reverse	125308,0577

Najlepsza trasa:

11 -> 13 -> 40 -> 35 -> 36 -> 34 -> 39 -> 42 -> 33 -> 38 -> 41 -> 29 -> 26 -> 30 -> 79 -> 78 -> 76 -> 17 -> 20 -> 16 -> 19 -> 107 -> 14 -> 105 -> 5 -> 23 -> 22 -> 3 -> 21 -> 18 -> 71 -> 7 -> 66 -> 72 -> 73 -> 67 -> 70 -> 69 -> 68 -> 74 -> 75 -> 77 -> 116 -> 83 -> 80 -> 125 -> 81 -> 82 -> 101 -> 100 -> 62 -> 118 -> 95 -> 108 -> 87 -> 86 -> 85 -> 84 -> 109 -> 103 -> 124 -> 88 -> 91 -> 98 -> 64 -> 112 -> 65 -> 54 -> 46 -> 48 -> 52 -> 47 -> 117 -> 45 -> 93 -> 111 -> 110 -> 106 -> 126 -> 92 -> 94 -> 122 -> 96 -> 97 -> 31 -> 28 -> 24 -> 32 -> 121 -> 27 -> 37 -> 25 -> 0 -> 15 -> 1 -> 50 -> 49 -> 12 -> 114 -> 9 -> 119 -> 6 -> 104 -> 113 -> 10 -> 8 -> 2 -> 115 -> 89 -> 58 -> 59 -> 61 -> 60 -> 90 -> 63 -> 99 -> 57 -> 51 -> 123 -> 55 -> 4 -> 120 -> 56 -> 53 -> 102 -> 43 -> 44

Czas wykonania: 00:26:10.8165431

Podsumowanie

Najlepsza znaleziona trasa dla 127 miast to:

11 -> 13 -> 40 -> 35 -> 36 -> 34 -> 39 -> 42 -> 33 -> 38 -> 41 -> 32 -> 121 -> 27 -> 31 -> 28 -> 24 -> 25 -> 37 -> 29 -> 26 -> 30 -> 79 -> 78 -> 76 -> 17 -> 20 -> 16 -> 19 -> 107 -> 5 -> 23 -> 22 -> 3 -> 21 -> 18 -> 71 -> 7 -> 8 -> 10 -> 113 -> 104 -> 6 -> 119 -> 9 -> 2 -> 99 -> 57 -> 63 -> 90 -> 60 -> 61 -> 59 -> 115 -> 89 -> 58 -> 66 -> 72 -> 73 -> 67 -> 70 -> 69 -> 68 -> 74 -> 77 -> 75 -> 116 -> 83 -> 80 -> 125 -> 81 -> 82 -> 101 -> 100 -> 62 -> 118 -> 95 -> 108 -> 87 -> 86 -> 85 -> 84 -> 109 -> 103 -> 124 -> 88 -> 91 -> 98 -> 64 -> 112 -> 65 -> 54 -> 46 -> 48 -> 52 -> 117 -> 47 -> 45 -> 93 -> 111 -> 110 -> 106 -> 126 -> 97 -> 96 -> 122 -> 94 -> 92 -> 44 -> 102 -> 43 -> 53 -> 56 -> 120 -> 4 -> 49 -> 12 -> 114 -> 51 -> 123 -> 55 -> 50 -> 1 -> 15 -> 0 -> 105 -> 14

Jej odległość wynosi 124264,2947. Została ona wyznaczona dla następujących wartości parametrów: { 500, 500, 1, 100, „reverse” }

Bibliografia

- https://brain.fuw.edu.pl/edu/index.php/Algorytmy_Genetyczne
- http://staff.iiar.pwr.wroc.pl/dariusz.banasiak/si/SI_wyklad10.pdf
- http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w3.pdf
- https://torus.uck.pk.edu.pl/~amarsz/dydaktyka/wsi/AG_skrypt.pdf
- <https://www.geeksforgeeks.org/what-is-tabu-search/>
- https://ii.uni.wroc.pl/~prz/2011lato/ah/opracowania/szuk_tabu.opr.pdf
- https://kcir.pwr.edu.pl/~witold/aiarr/2007_projekty/vlsi/
- http://algorytmy.ency.pl/artykul/symulowane_wyzarzanie
- https://en.wikipedia.org/wiki/Hill_climbing
- https://en.wikibooks.org/wiki/Artificial_Intelligence/Search/Iterative_Improvement/Hill_Climbing
- http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w2.pdf
- <https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/#types-of-hill-climbing-in-artificial-intelligence>
- <https://www.is.umk.pl/~duch/Wyklady/AI/AI03-3.pdf>
- https://www.cs.put.poznan.pl/mrdom/teaching/laboratories/OptKomb/CI_wyklad_ewoluc_4.pdf
- https://app.assembla.com/wiki/show/easytsp/Algorytm_Mrowkowy