



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ZARZĄDZANIA

**Porównanie efektywności wybranych algorytmów w rozwiązaniu
problemu komiwojażera.**

*Comparison of the efficiency of selected algorithms in solving the
traveling salesman problem.*

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	11
Macierz 127.....	15
Podsumowanie GA	19
TS – Przeszukiwanie Tabu	19
Wprowadzenie	19
Parametry i ich wartości	20
Macierz 48	20
Macierz 76	24
Macierz 127.....	28
Podsumowanie TS	32
Algorytm wspinaczki z multistartem	32
Wprowadzenie	32
Parametry i ich wartości	33
Macierz 48	34
Macierz 76	38
Macierz 127.....	43
Podsumowanie	47

SA – symulowane wyżarzanie.....	47
Wprowadzenie	47
Parametry i ich wartości	47
Działanie programu	48
Macierz 48	48
Macierz 76	49
Macierz 127.....	50
Podsumowanie SA.....	51
Algorytm mrówkowy.....	59
Wprowadzenie	59
Działanie algorytmu	59
Działanie kodu.....	60
Macierz 48	61
Macierz 76	63
Macierz 127.....	66
PODSUMOWANIE	69
Macierz 48 miast	69
Macierz 76 miast	69
Macierz 127 miast.....	70
Podsumowanie algorytmów	70
Bibliografia	72

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

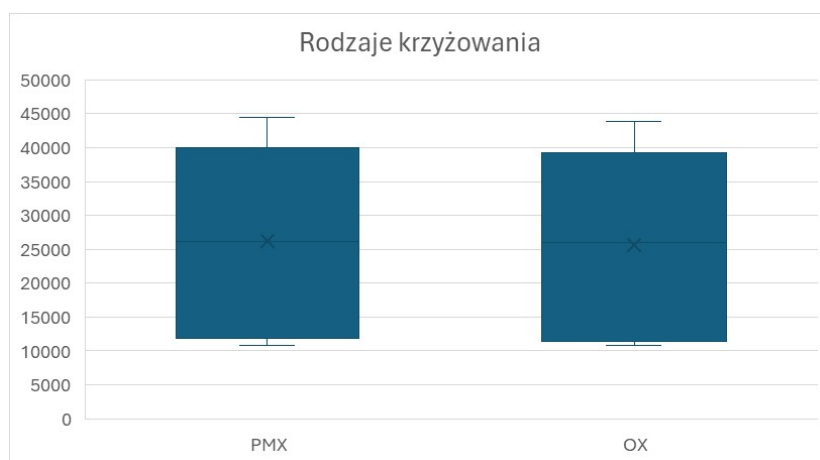
Najkrótsza trasa przy tej macierzy wynosi 10767, została wskazana dla kombinacji parametrów:

- **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:

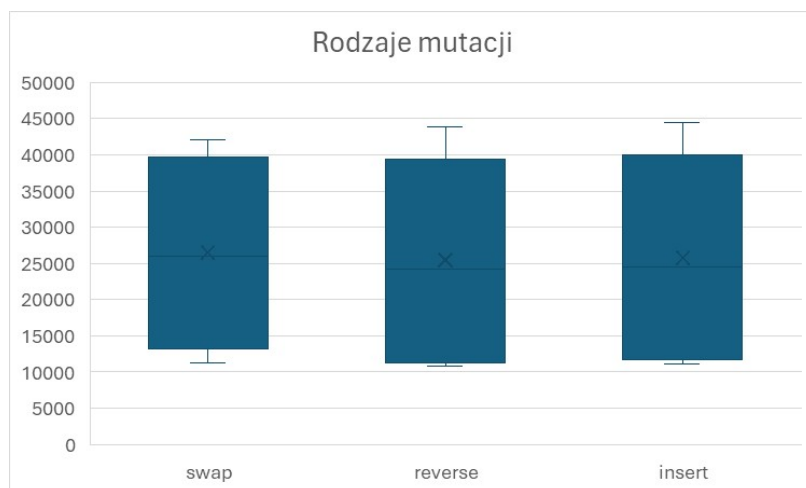
Krzyżowanie



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.

Rodzaj krzyżowania	Średnia trasa	Min. trasa	Max trasa
PMX	26209,44	10815	44513
OX	25616,81	10767	43909

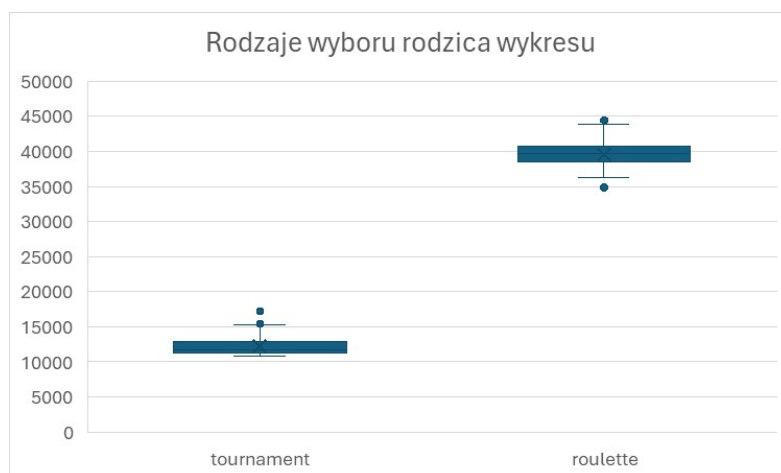
Mutacja



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.

Rodzaj mutacji	Średnia trasa	Min. trasa	Max trasa
swap	26536,14	11265	42131
reverse	25456,61	10767	43919
insert	25646,63	11075	44513

Wybór rodzica

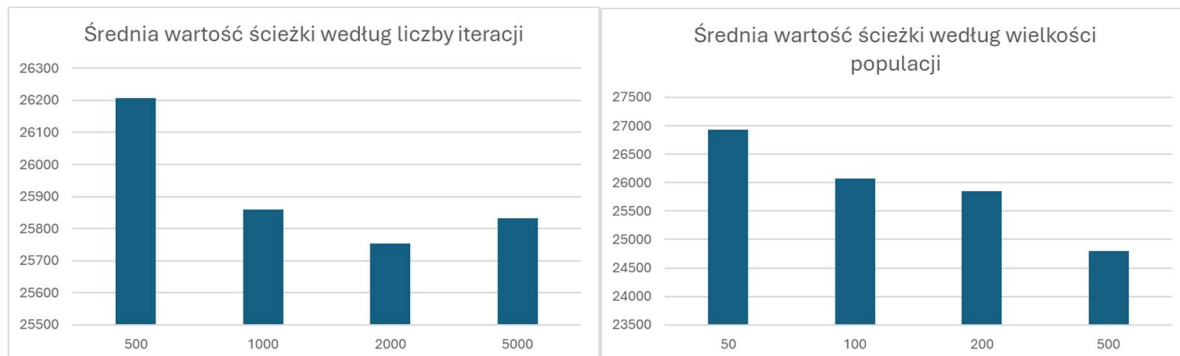


W przypadku rodzajów selekcji rodziców widzimy, że występują między metodami spore różnice wartości. Przy tournament rozrzut danych jest mniejszy, wartości są o wiele niższe niż przy metodzie roulette, co jest dla nas korzystniejsze, ponieważ szukamy najniższej trasy. Przy roulette rozrzut danych jest 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).

Rodzaj wyboru rodzica	Średnia trasa	Min. trasa	Max trasa
tournament	12255,49	10767	17203
roulette	39570,76	34857	44513

Iteracje i wielkość populacji



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.

Rozmiar populacji	Średnia trasa	Min. trasa	Max trasa
50	26926,65	10946	44513
100	26065,65	10767	43056
200	25855,94	10966	41177
500	24804,27	10815	40451

Liczba iteracji	Średnia trasa	Min. trasa	Max trasa
500	26207,46	10767	43919
1000	25860,44	10843	44513
2000	25752,58	10889	42965
5000	25832,02	10946	42757

10 najlepszych wyników:

Rozmiar populacji	Liczba iteracji	Metody wyboru rodziców	Metody mutacji	Metody krzyżowania	Dopasowanie
100	500	tournament	reverse	OX	10767

500	500	tournament	reverse	PMX	10815
100	1000	tournament	reverse	PMX	10843
500	1000	tournament	reverse	OX	10857
500	2000	tournament	reverse	OX	10889
50	5000	tournament	reverse	OX	10946
500	500	tournament	reverse	OX	10962
200	5000	tournament	reverse	OX	10966
200	500	tournament	reverse	OX	11002
50	2000	tournament	reverse	OX	11021

Na zestawieniu 10 najlepszych wyników możemy zaobserwować, że metoda mutacji oraz metoda selekcji rodziców nie zmieniają się - najlepsze wyniki osiągamy przy reverse oraz tuornament. W metodach krzyżowania przeważa też metoda OX, PMX mimo podobieństwa pojawia się najczęściej. Te 3 parametry powtarzają się prawie w każdym przypadku, dlatego możemy uznać, że mają największy wpływ na wynik dopasowania.

10 najgorszych wyników:

Rozmiar populacji	Liczba iteracji	Metody wyboru rodziców	Metody mutacji	Metody krzyżowania	Dopasowanie
100	500	roulette	insert	OX	41750
100	5000	roulette	swap	PMX	41807
50	500	roulette	swap	OX	42131
50	1000	roulette	reverse	OX	42681
50	5000	roulette	reverse	PMX	42757
50	2000	roulette	reverse	PMX	42965
100	500	roulette	insert	PMX	43056
50	500	roulette	reverse	OX	43909
50	500	roulette	reverse	PMX	43919
50	1000	roulette	insert	PMX	44513

Na zestawieniu 10 najgorszych wyników możemy zaobserwować, że metoda selekcji rodziców nie zmienia się - najgorsze wyniki osiągamy przy roulette. W metodach mutacji reverse pojawia się najczęściej. Jeśli chodzi o rozmiar populacji – mniejsze wartości pojawiają się w tym zestawieniu. W metodach krzyżowania przeważa też metoda PMX, jednak metody działają praktycznie identycznie, a w tabeli występują na zmianę. Wyszczególnione parametry powtarzają się w wielu z najgorszych przypadków, dlatego możemy uznać, że mają najgorszy wpływ na wynik dopasowania.

Macierz 76

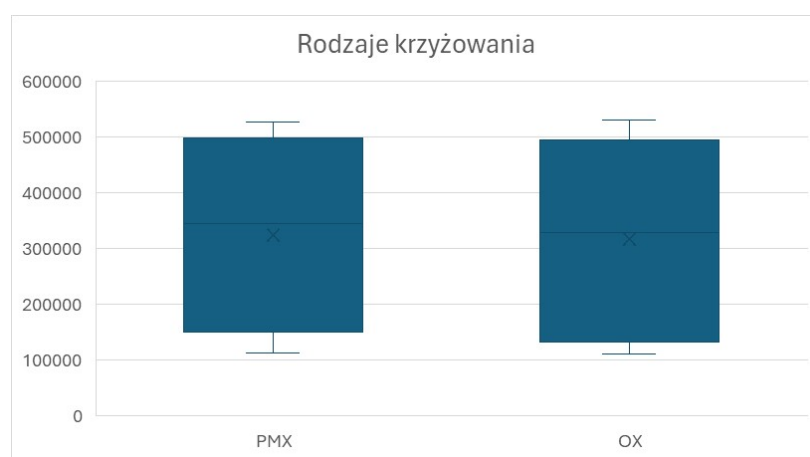
Najkrótsza trasa przy tej macierzy wynosi 109879, została wskazana dla kombinacji parametrów:

- **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:

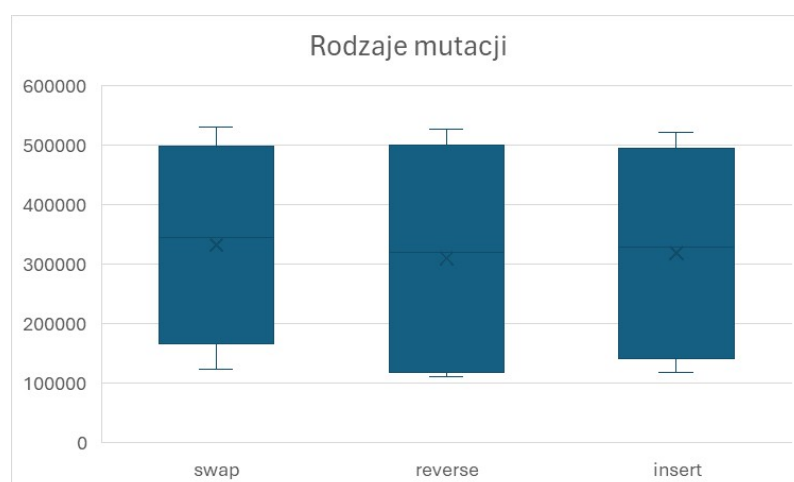
Krzyżowanie



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.

Rodzaj krzyżowania	Średnia trasa	Min. trasa	Max trasa
OX	316028,33	109879	530138
PMX	323255,32	111210	527034

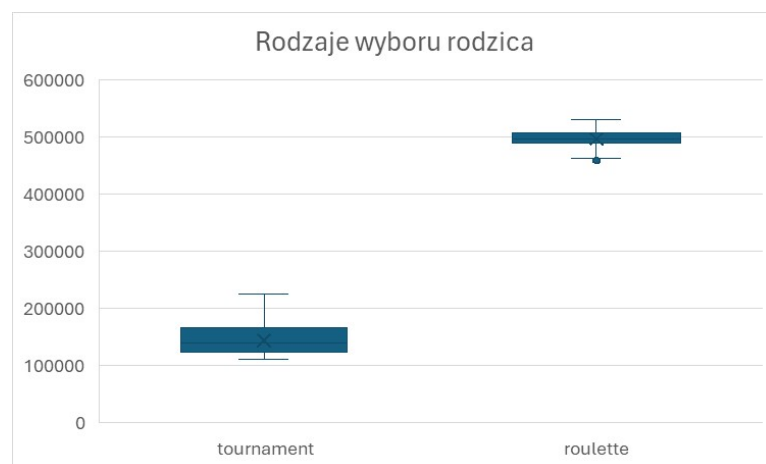
Mutacja



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.

Rodzaj mutacji	Średnia trasa	Min. trasa	Max trasa
insert	317886,97	116525	520840
reverse	309539,00	109879	527034
swap	331499,52	123133	530138

Selekcja rodzica

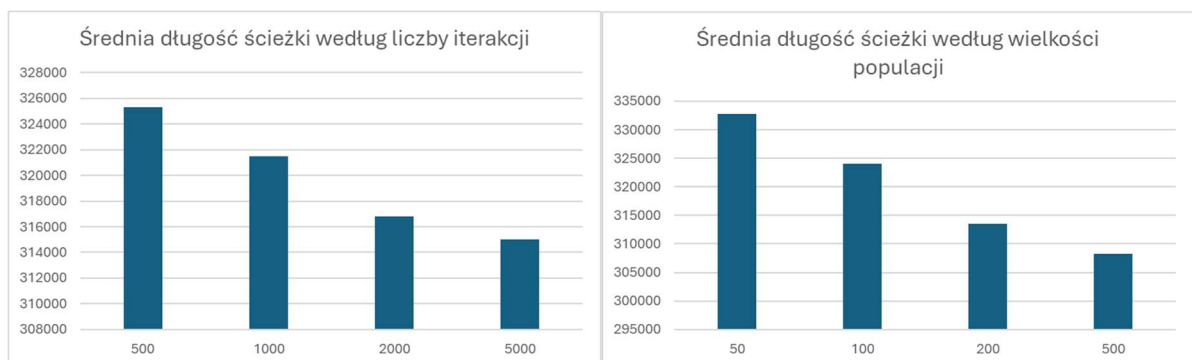


W przypadku rodzajów selekcji rodziców widzimy, że występują między metodami spore różnice wartości. Przy tournament rozrzut danych jest większy, wartości są o wiele niższe niż przy metodzie roulette, co jest dla nas korzystniejsze, ponieważ szukamy najniższej trasy. Przy roulette rozrzut danych jest 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).

Rodzaj wyboru rodzica	Średnia trasa	Min. trasa	Max trasa
tournament	496128,26	457898	530138
roulette	143155,40	109879	224740

Liczba iteracji i wielkość populacji



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.

Rozmiar populacji	Średnia trasa	Min. trasa	Max trasa
50	332737,90	115456	530138
100	324078,40	109879	520840
200	313530,42	112530	509449
500	308220,60	111210	508599

Liczba iteracji	Średnia trasa	Min. trasa	Max trasa
500	325305,31	117201	520840
1000	321465,67	112808	530138
2000	316798,06	111210	527034
5000	314998,27	109879	519081

10 najlepszych wyników:

Rozmiar populacji	Liczba iteracji	Metody wyboru rodziców	Metody mutacji	Metody krzyżowania	Dopasowanie
100	5000	tournament	reverse	OX	109879
500	2000	tournament	reverse	PMX	111210
500	2000	tournament	reverse	OX	111674
200	2000	tournament	reverse	PMX	112530
200	1000	tournament	reverse	OX	112808
500	5000	tournament	reverse	OX	112818
500	1000	tournament	reverse	PMX	112873
200	1000	tournament	reverse	PMX	113433
200	2000	tournament	reverse	OX	113780
500	1000	tournament	reverse	OX	113895

Na zestawieniu 10 najlepszych wyników możemy zaobserwować, że metoda mutacji oraz metoda selekcji rodziców nie zmieniają się - najlepsze wyniki osiągamy przy reverse oraz tuornament. W metodach krzyżowania przeważa też metoda OX, jednak metody działają praktycznie identycznie, a w tabeli występują na zmianę. W przypadku liczby iteracji widzimy, że przeważa wartość 2000. Te parametry powtarzają się prawie w każdym z najlepszych przypadków, dlatego możemy uznać, że mają największy wpływ na wynik dopasowania.

10 najgorszych wyników:

Rozmiar populacji	Liczba iteracji	Metody wyboru rodziców	Metody mutacji	Metody krzyżowania	Dopasowanie
100	2000	roulette	insert	PMX	516451
100	500	roulette	insert	OX	518435
50	2000	roulette	swap	PMX	518720
50	5000	roulette	insert	OX	518889
100	5000	roulette	reverse	OX	519081
50	2000	roulette	insert	OX	519733
100	1000	roulette	reverse	PMX	520578
100	500	roulette	insert	PMX	520840
50	2000	roulette	reverse	PMX	527034
50	1000	roulette	swap	OX	530138

Na zestawieniu 10 najgorszych wyników możemy zaobserwować, że metoda selekcji rodziców nie zmienia się - najgorsze wyniki osiągamy przy roulette. W metodach mutacji insert pojawia się najczęściej. Jeśli chodzi o rozmiar populacji – mniejsze wartości pojawiają się w tym zestawieniu. W metodach krzyżowania przeważa też metoda PMX, jednak metody działają praktycznie identycznie, a w tabeli występują na zmianę. Wyszczególnione parametry powtarzają się w wielu z najgorszych przypadków, dlatego możemy uznać, że mają najgorszy wpływ na wynik dopasowania.

Macierz 127

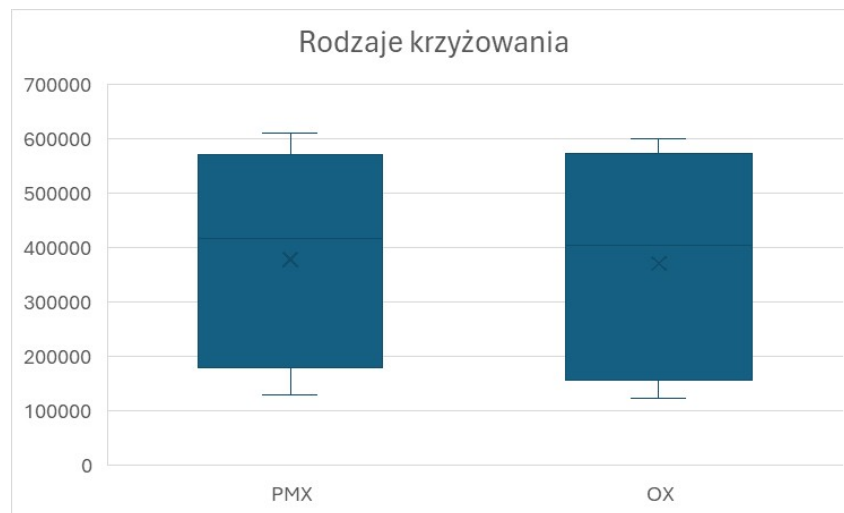
Najkrótsza trasa przy tej macierzy wynosi 121798, została wskazana dla kombinacji parametrów:

- **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:

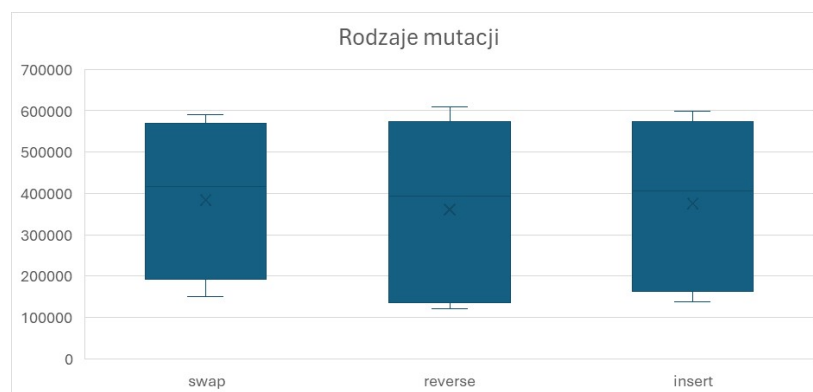
Krzyżowanie



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.

Rodzaj krzyżowania	Średnia trasa		Min. trasa	Max trasa
PMX	370059,54	121798	598519	
OX	377462,82	129038	609809	

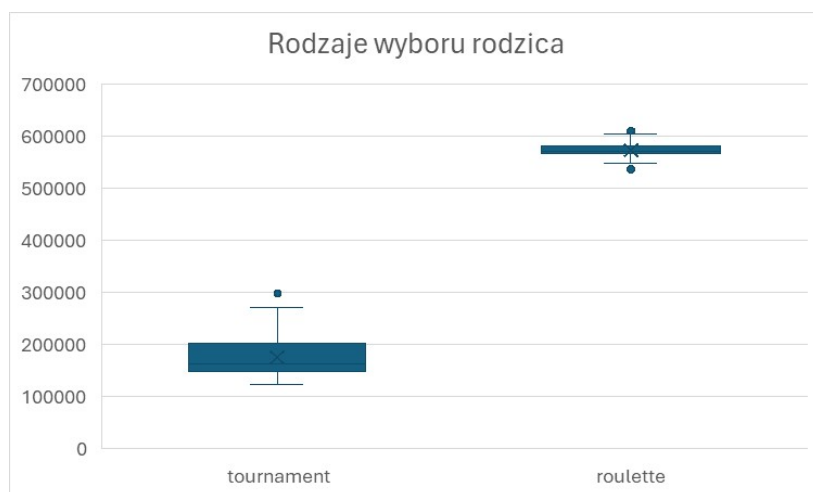
Mutacja



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.

Rodzaj mutacji	Średnia trasa		Min. trasa	Max trasa
insert	375928,83	139018	598028	
reverse	361429,89	121798	609809	
swap	383924,83	150791	590445	

Selekcja rodzica

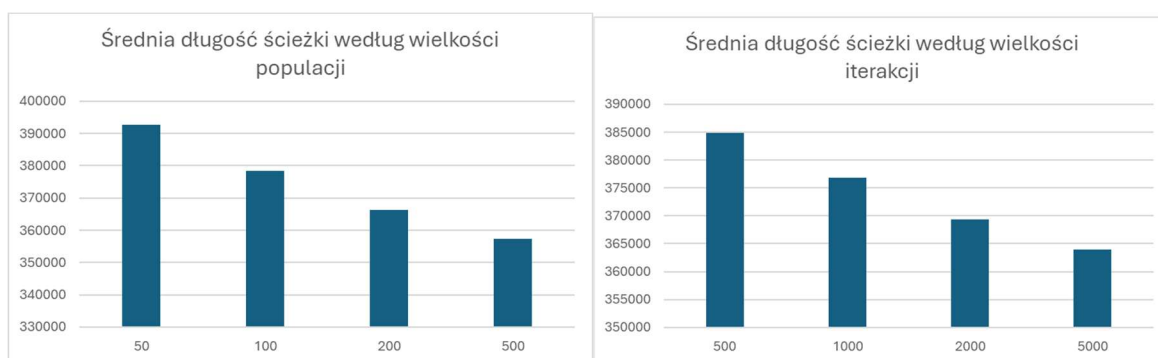


W przypadku rodzajów selekcji rodziców widzimy, że występują między metodami spore różnice wartości. Przy tournament rozrzut danych jest większy, wartości są o wiele niższe niż przy metodzie roulette, co jest dla nas korzystniejsze, ponieważ szukamy najniższej trasy. Przy roulette rozrzut danych jest 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).

Rodzaj wyboru rodzica	Średnia trasa		Min. trasa	Max trasa
roulette	572849,19	535671	609809	
tournament	174673,18	121798	297820	

Wielkość populacji i liczba iteracji



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.

Rozmiar populacji	Średnia trasa		Min. trasa	Max trasa
50	392707,73	131095	609809	
100	378530,25	131311	598519	
200	366403,35	127950	578796	
500	357403,40	121798	572303	

Liczba iteracji	Średnia trasa		Min. trasa	Max trasa
500	384895,23	129426	609809	
1000	376850,19	131817	596858	
2000	369299,42	129018	598028	
5000	363999,90	121798	602631	

10 najlepszych wyników:

Rozmiar populacji	Liczba iteracji	Metody wyboru rodziców	Metody mutacji	Metody krzyżowania	Dopasowanie
500	5000	tournament	reverse	OX	109879
200	5000	tournament	reverse	OX	111210
200	2000	tournament	reverse	OX	111674
200	5000	tournament	reverse	PMX	112530
500	500	tournament	reverse	PMX	112808
50	5000	tournament	reverse	PMX	112818
100	5000	tournament	reverse	PMX	112873
500	1000	tournament	reverse	PMX	113433
500	2000	tournament	reverse	OX	113780
200	2000	tournament	reverse	PMX	113895

Na zestawieniu 10 najlepszych wyników możemy zaobserwować, że metoda mutacji oraz metoda selekcji rodziców nie zmieniają się - najlepsze wyniki osiągamy przy reverse oraz tuornament. W metodach krzyżowania przeważa też metoda OX, jednak metody działają praktycznie identycznie, a w tabeli występują na zmianę. W przypadku liczby iteracji widzimy, że przeważa wartość 5000. Natomiast w wielkości próby najwięcej przewija się 200 i 500. Wyszczególnione parametry powtarzają się w wielu z najlepszych przypadków, dlatego możemy uznać, że mają największy wpływ na wynik dopasowania.

10 najgorszych wyników:

Rozmiar populacji	Liczba iteracji	Metody wyboru rodziców	Metody mutacji	Metody krzyżowania	Dopasowanie
-------------------	-----------------	------------------------	----------------	--------------------	-------------

100	5000	roulette	insert	OX	587382
50	500	roulette	reverse	OX	588382
100	1000	roulette	swap	PMX	588613
50	5000	roulette	swap	PMX	590445
50	5000	roulette	insert	PMX	593494
50	1000	roulette	insert	OX	596858
50	2000	roulette	insert	PMX	598028
100	500	roulette	reverse	OX	598519
50	5000	roulette	reverse	PMX	602631
50	500	roulette	reverse	PMX	609809

Na zestawieniu 10 najgorszych wyników możemy zaobserwować, że metoda selekcji rodziców nie zmienia się - najgorsze wyniki osiągamy przy roulette. W metodach mutacji insert i reverse pojawiają się praktycznie naprzemiennie. Jeśli chodzi o rozmiar populacji – mniejsze wartości pojawiają się w tym zestawieniu. W metodach krzyżowania przeważa też metoda PMX, jednak metody działają praktycznie identycznie, a w tabeli występują na zmianę. Wyszczególnione parametry powtarzają się w wielu z najgorszych przypadków, dlatego możemy uznać, że mają najgorszy wpływ na wynik dopasowania.

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, jednak jest to zauważalne coraz bardziej przy coraz większych macierzach. 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			
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

10 najlepszych wyników:

Liczba iteracji	Liczba iteracji bez poprawy	Długość listy tabu	Rodzaj sąsiedztwa	Dystans	Czas (sekundy)
100	25	25	reverse	10628	0,22
2000	25	25	reverse	10730	0,18
2000	200	200	reverse	10812	2,25
250	25	200	reverse	10831	0,28

500	100	100	reverse	10840	0,87
250	200	25	reverse	10841	0,81
500	200	25	reverse	10849	0,74
2000	25	200	reverse	10855	0,30
100	25	50	reverse	10859	0,31
250	25	100	reverse	10872	0,26

Trasa najlepszego rozwiązania:

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 30

Przedstawiona tabela zawiera zestawienie dziesięciu najlepszych wyników uzyskanych za pomocą algorytmu Tabu Search, każdy wiersz reprezentuje inną konfigurację parametrów. Choć w całej analizie brano pod uwagę różne rodzaje sąsiedztwa, wszystkie najlepsze rozwiązania uzyskano dla sąsiedztwa typu "reverse". Wskazuje to, że ten rodzaj sąsiedztwa jest szczególnie skuteczny dla badanego problemu. Najlepszy wynik osiągnięto przy stosunkowo niewielkiej liczbie iteracji (100), krótkiej liście tabu oraz ograniczeniu liczby iteracji bez poprawy do 25. Co ciekawe, inne ustawienia z większą liczbą iteracji, np. 2000 iteracji przy takich samych wartościach listy tabu i iteracji bez poprawy, uzyskały nieco gorszy dystans.

10 najgorszych wyników:

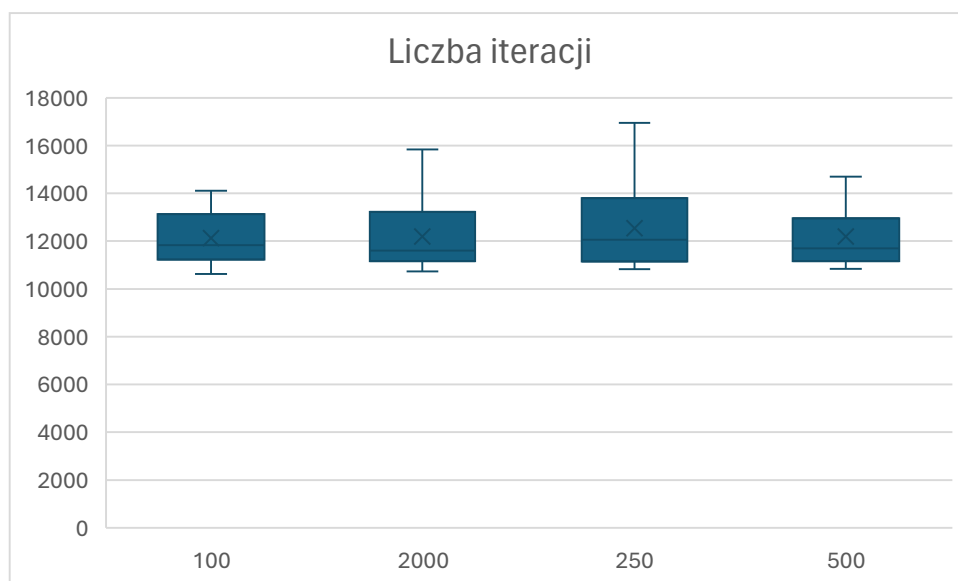
Liczba iteracji	Liczba iteracji bez poprawy	Długość listy tabu	Rodzaj sąsiedztwa	Dystans	Czas (sekundy)
250	25	50	swap	16951	0,45
250	25	200	Swap	16410	0,50
250	100	25	Swap	16259	0,73
2000	100	50	swap	15835	0,72
2000	100	100	Swap	15529	1,03
2000	25	25	Swap	15340	0,25
250	200	200	swap	15192	2,73
250	200	25	Swap	14924	0,81
250	50	100	Swap	14828	0,44
500	100	50	swap	14698	0,94

Trasa najgorszego rozwiązania:

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

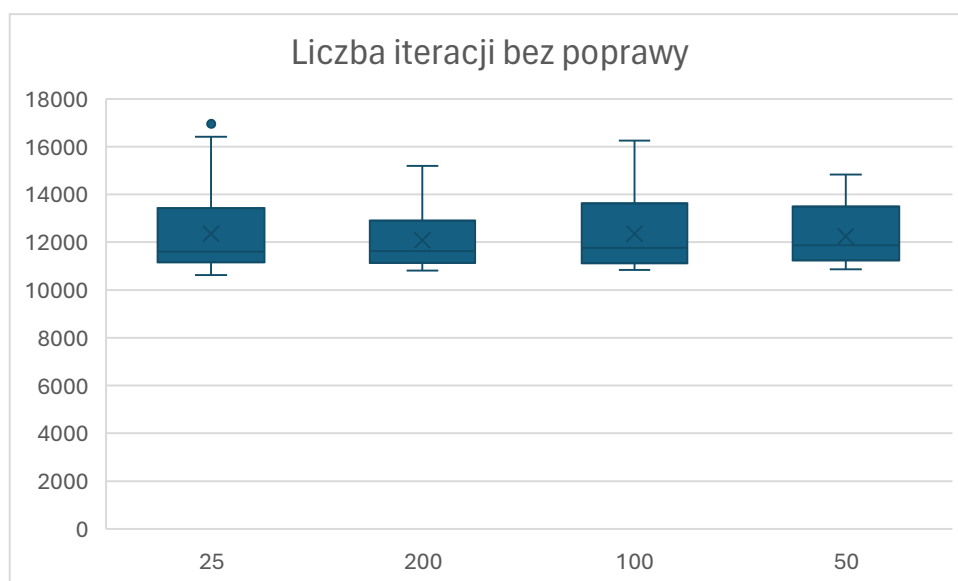
Analiza dziesięciu najgorszych wyników algorytmu pokazuje, że wszystkie zostały uzyskane dla sąsiedztwa typu "swap", które wyraźnie nie sprawdza się w kontekście badanego problemu. Najgorszy dystans osiągnięto przy niewielkiej liczbie iteracji, krótkiej liście tabu oraz niskim limicie iteracji bez poprawy. Z kolei lepsze wyniki w tej grupie, choć nadal słabe, można zaobserwować dla większych długości list tabu i wyższych limitów iteracji bez poprawy, ale kosztem znacznie dłuższego czasu obliczeń. Warto zauważyć, że nawet przy wysokich wartościach iteracji, dystanse pozostawały znacznie gorsze niż dla konfiguracji z "reverse", co podkreśla nieskuteczność "swap" w porównaniu z innymi rodzajami sąsiedztwa.

Analiza wpływu wartości parametrów przy użyciu wykresów pudełkowych:



Liczba iteracji	Średnia trasa	Minimum	Maksimum
100	12126,4	10628	14110
250	12546,7	10831	16951
500	12186,2	10840	14698
2000	12187,9	10730	15835

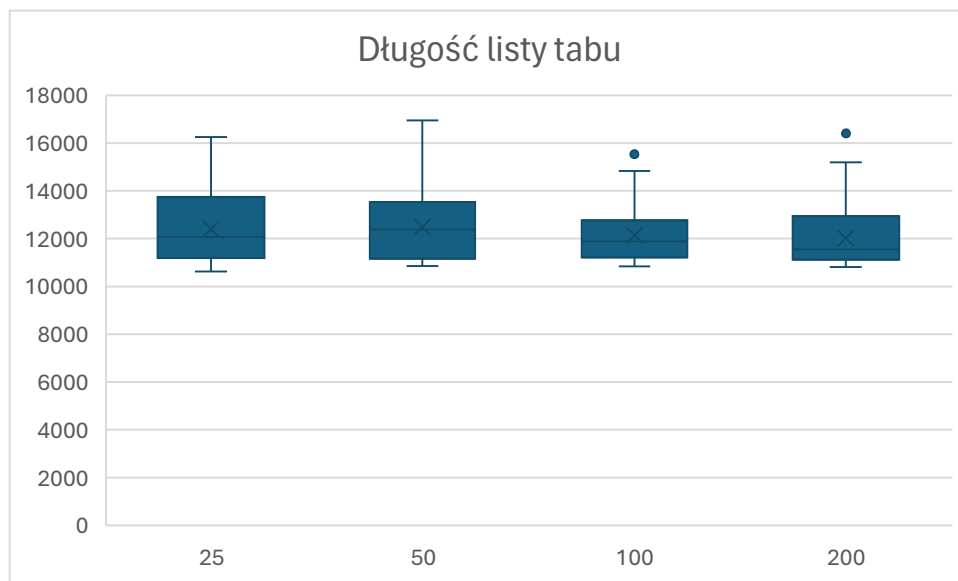
Analiza wyników pokazuje, że najlepsze jakościowo rozwiązania uzyskano przy 100 iteracjach, gdzie średnia trasa była najniższa, a minimalny dystans wyniósł 10628. Wzrost liczby iteracji do 250 zwiększa zarówno średnią trasę, jak i maksymalny dystans, co wskazuje na większy rozrzut wyników i mniej stabilne działanie algorytmu. Dla 500 i 2000 iteracji średnia trasa pozostaje na podobnym poziomie, co sugeruje, że zwiększanie liczby iteracji powyżej pewnego progu nie wpływa znacząco na jakość, a może prowadzić do wydłużenia obliczeń. Optimalizacja działa najlepiej przy mniejszej liczbie iteracji, zapewniając jednocześnie stabilne wyniki.



Liczba iteracji bez poprawy	Średnia trasa	Minimum	Maksimum
25	12351,5	10628	16951

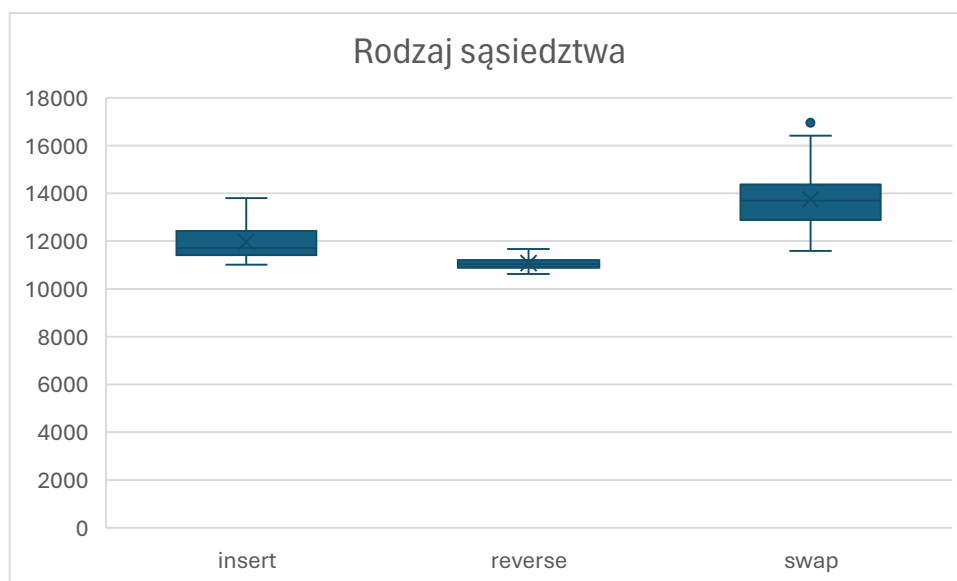
50	12251,4	10873	14828
100	12354,2	10840	16259
200	12090,3	10812	15192

Analiza wyników w zależności od liczby iteracji bez poprawy wskazuje, że najniższa średnia trasa została osiągnięta przy wartości 200, gdzie również odnotowano stosunkowo niskie maksimum. Najgorsze wyniki pojawiły się przy najmniejszej liczbie iteracji bez poprawy, co wskazuje na większy rozrzut wyników i niestabilność. Przy wartościach pośrednich, takich jak 50 i 100, średnia trasa była wyższa, a rozrzut wyników mniejszy niż dla 25, ale wyższy niż dla 200. Można wnioskować, że wyższe limity iteracji bez poprawy sprzyjają uzyskaniu lepszej jakości i większej stabilności wyników.



Długość listy tabu	Średnia trasa	Minimum	Maksimum
25	12412,2	10628	16259
50	12489,1	10859	16951
100	12132,7	10840	15529
200	12013,44	10812	16410

Analiza wyników w zależności od liczby iteracji bez poprawy wskazuje, że najniższa średnia trasa została osiągnięta przy wartości 200, gdzie również odnotowano stosunkowo niskie maksimum. Najgorsze wyniki pojawiły się przy najmniejszej liczbie iteracji bez poprawy, co wskazuje na większy rozrzut wyników i niestabilność. Przy wartościach pośrednich, takich jak 50 i 100, średnia trasa była wyższa, a rozrzut wyników mniejszy niż dla 25, ale wyższy niż dla 200. Można wnioskować, że wyższe limity iteracji bez poprawy sprzyjają uzyskaniu lepszej jakości i większej stabilności wyników.



Rodzaj sąsiedztwa	Średnia trasa	Minimum	Maksimum
swap	13743,9	11587	16951
reverse	11078,2	10628	11673
insert	11963,38	11009	13804

Analiza wpływu rodzaju sąsiedztwa wskazuje, że najlepsze wyniki osiągnięto dla "reverse", które uzyskało najniższą średnią trasę oraz najniższe minimum, a także charakteryzowało się bardzo małym rozrzutem wyników. Sąsiedztwo "insert" osiągnęło gorszą średnią trasę i wyższy rozrzut wyników, co wskazuje na jego mniejszą efektywność. Najsłabsze wyniki uzyskano dla "swap", gdzie średnia trasa była najwyższa, a maksimum wyniosło aż 16951, co świadczy o jego niskiej skuteczności i dużej niestabilności.

Macierz 76

Średni czas generowania 192 kombinacji: 571,58 sekund

Średnia długość trasy: 136362,08

10 najlepszych wyników:

Liczba iteracji	Liczba iteracji bez poprawy	Długość listy tabu	Rodzaj sąsiedztwa	Dystans	Czas (sekundy)
250	200	100	reverse	108396,1	1,64
500	200	200	reverse	109341,8	3,68
100	25	25	reverse	109712,2	0,63
100	200	200	reverse	109922,2	0,64
2000	50	50	reverse	110041,0	0,50
2000	100	50	reverse	110115,4	0,81
2000	50	200	reverse	110147,1	1,01
250	25	25	reverse	110380,0	0,30
2000	50	25	reverse	110456,7	0,43
500	25	25	reverse	110608,1	0,32

Trasa najlepszego rozwiązania:

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

Analiza dziesięciu najlepszych wyników wskazuje, że wszystkie uzyskano dla sąsiedztwa typu "reverse", co potwierdza jego skuteczność w optymalizacji badanego problemu. Najlepszy dystans osiągnięto przy 250 iteracjach, 200 iteracjach bez poprawy i długości listy tabu równej 50, co wskazuje na korzyści wynikające z dłuższych list tabu w stabilizowaniu wyników. Krótsze listy tabu również generowały dobre wyniki, ale często kosztem nieco wyższych dystansów, mimo krótszego czasu obliczeń. Widać także, że zwiększanie liczby iteracji do 2000 nie zawsze przekładało się na lepszy dystans, co sugeruje, że w tym problemie większa liczba iteracji nie zawsze jest kluczowa.

10 najgorszych wyników:

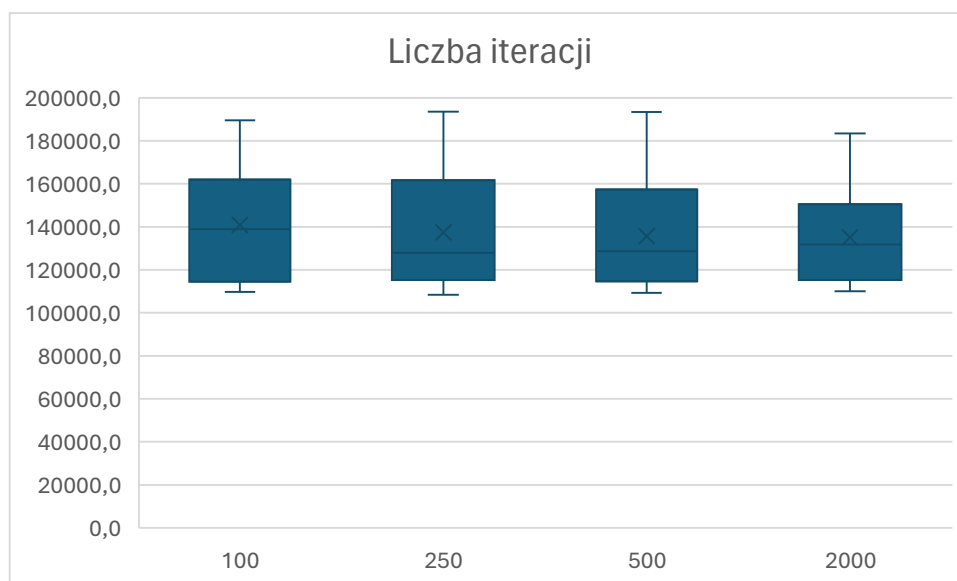
Liczba iteracji	Liczba iteracji bez poprawy	Długość listy tabu	Rodzaj sąsiedztwa	Dystans	Czas (sekundy)
250	200	25	swap	193514,0	0,74
500	25	25	swap	193426,2	0,27
100	50	100	swap	189462,7	0,54
250	25	50	swap	189351,0	0,77
100	100	25	swap	183475,8	0,30
2000	100	25	swap	183355,7	0,52
250	100	50	swap	182497,9	0,96
250	200	100	swap	179699,2	2,16
100	25	100	swap	179046,1	0,57
100	50	50	swap	178896,2	0,46

Trasa najgorszego rozwiązania:

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

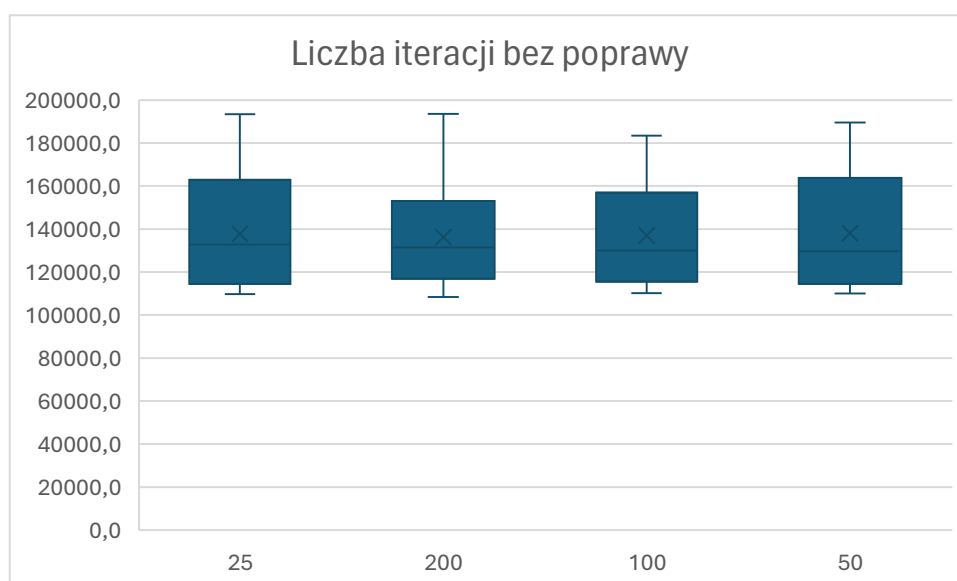
Wyniki dla dziesięciu najgorszych konfiguracji wskazują, że najgorsze wyniki uzyskano przy sąsiedztwie typu "swap", z najwyższymi dystansami i znacznym rozrzutem wyników. Niezależnie od liczby iteracji, zwiększanie długości listy tabu nie wpłynęło znacząco na poprawę wyników. Sąsiedztwo "swap" wykazało się wyraźnie gorszą efektywnością niż "reverse", generując znacznie wyższe dystanse i mniej stabilne rozwiązania.

Analiza wpływu wartości parametrów przy użyciu wykresów pudełkowych:



Liczba iteracji	Średnia trasa	Minimum	Maksimum
100	140797,0	109712,2	183475,8
250	137362,5	108396,1	193514,0
500	135661,3	109341,8	193426,2
2000	125044,3	11041,0	183355,7

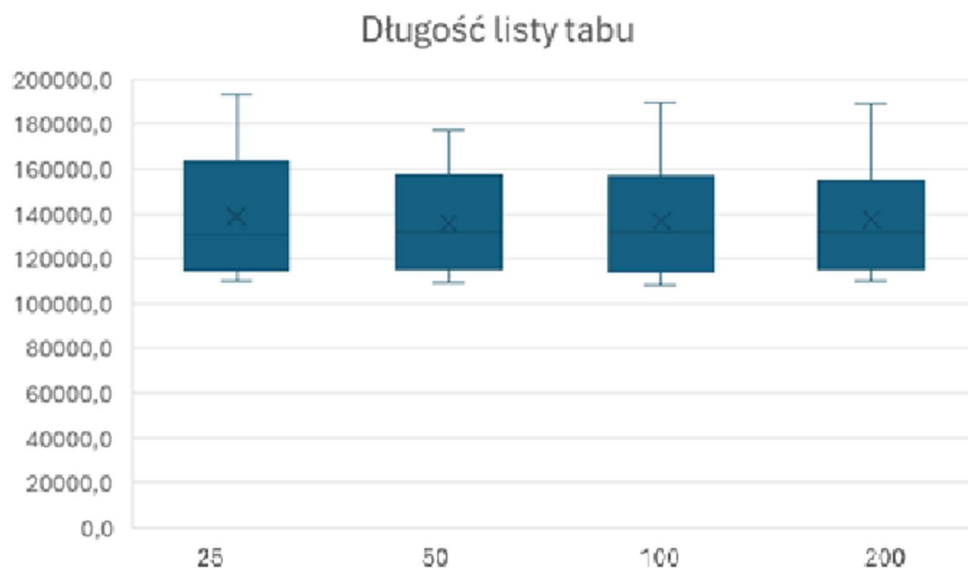
Wyniki wskazują, że największa średnia trasa występuje przy 100 iteracjach, a jej jakość poprawia się przy zwiększeniu liczby iteracji, osiągając najniższe średnie dystanse przy 2000 iteracjach. Zwiększanie liczby iteracji prowadzi do mniejszego rozrzutu wyników, co sugeruje, że większa liczba iteracji może stabilizować rozwiązania, ale niekoniecznie poprawia jakość średniej trasy. Najniższe minimum pojawiło się przy 2000 iteracjach, co może wskazywać na lepszą konwergencję algorytmu w długim okresie.



Liczba iteracji bez poprawy	Średnia trasa	Minimum	Maksimum
-----------------------------	---------------	---------	----------

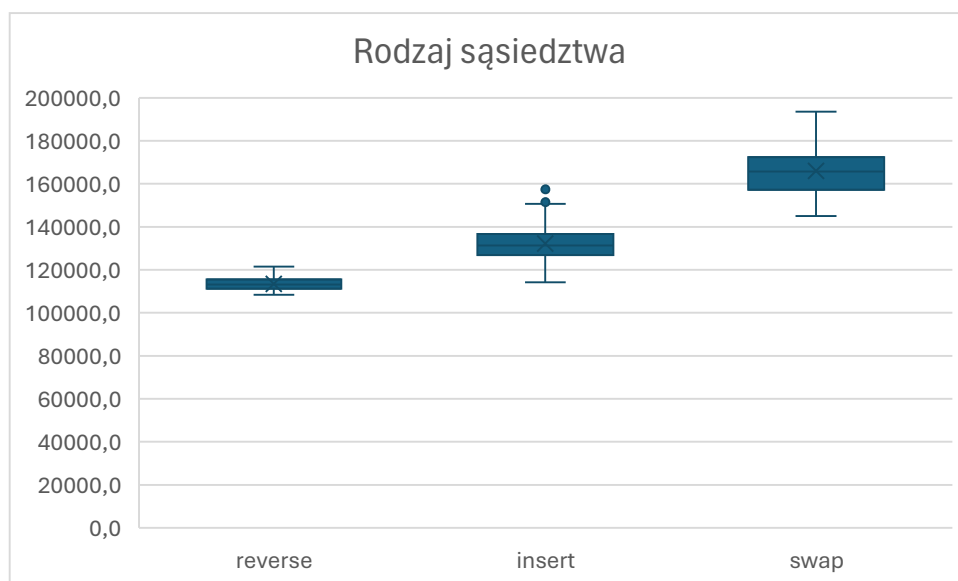
25	137684,0	109712,2	193426,2
50	138036,3	110041,0	189462,7
100	136978,4	110115,4	183475,8
200	136166,3	109922,2	193514,0

Wyniki pokazują, że liczba iteracji bez poprawy ma niewielki wpływ na średnią trasę, której wartość utrzymuje się na podobnym poziomie w zakresie od 136166,3 do 138036,3. Zwiększanie liczby iteracji bez poprawy prowadzi do minimalnego wzrostu średniego dystansu, ale nie przynosi znaczącej poprawy jakości wyników. Warto również zauważyć, że maksymalne wartości dystansów są stosunkowo wysokie, szczególnie przy 25 iteracjach bez poprawy, co może sugerować większą niestabilność przy mniejszej liczbie iteracji. Wskazuje to na możliwą korzyść z dłuższego czasu poszukiwania, ale przy minimalnej liczbie iteracji bez poprawy.



Długość listy tabu	Średnia trasa	Minimum	Maksimum
25	138805,9	109712,2	193514,0
50	137618,2	110041,0	189351,0
100	136614,8	108396,1	189462,7
200	135826,1	109341,8	177112,3

Wyniki pokazują, że wydłużenie długości listy tabu prowadzi do stopniowej poprawy średniej trasy, z najlepszym wynikiem osiągniętym przy długości listy tabu wynoszącej 200. Krótsze listy tabu generują wyższe średnie dystanse oraz większy rozrzut wyników, co wskazuje na mniejszą stabilność. Zwiększenie długości listy tabu pomaga uzyskać bardziej stabilne i jakościowo lepsze rozwiązania.



Rodzaj sąsiedztwa	Średnia trasa	Minimum	Maksimum
swap	165951,2998	144954,9	193514,0
reverse	113474,9709	108396,1	121475,9
insert	132222,5116	114251,7	151525,9

Analiza wyników wskazuje, że najlepsze wyniki uzyskano przy sąsiedztwie "reverse", które charakteryzuje się najniższą średnią trasą oraz najmniejszym rozrzutem wyników. Sąsiedztwo "insert" osiągnęło gorsze wyniki, z wyższą średnią trasą i większym zakresem wartości. Najgorsze wyniki uzyskano dla sąsiedztwa "swap", gdzie średnia trasa wyniosła aż 165951,3 a maksymalne wartości dystansów sięgały aż 193514,0.

Macierz 127

Średni czas generowania 192 kombinacji: 2258,12 sekund.

Średnia długość trasy: 156872,11

10 najlepszych wyników:

Liczba iteracji	Liczba iteracji bez poprawy	Długość listy tabu	Rodzaj sąsiedztwa	Dystans	Czas (sekundy)
500	100	25	reverse	119382,08	3,46
500	200	100	reverse	120678,69	9,66
2000	100	200	reverse	121587,67	7,46
250	25	100	reverse	121656,02	4,68
250	100	100	reverse	122649,50	5,66
500	25	25	reverse	122649,50	2,31
2000	100	50	reverse	122784,42	5,01
250	100	50	reverse	122797,38	4,18
250	200	25	reverse	122838,19	3,87
250	25	25	reverse	122845,51	2,42

Trasa najlepszego rozwiązania:

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

Wyniki dla dziesięciu najlepszych konfiguracji wskazują, że najniższy dystans osiągnięto przy 500 iteracjach, 100 iteracjach bez poprawy oraz długości listy tabu 25, gdzie dystans wyniósł 119382,08. Dłuższe listy tabu oraz większa liczba iteracji bez poprawy nie poprawiły znacząco jakości rozwiązań, ale prowadziły do wydłużenia czasu obliczeń. Najlepsze rozwiązania są uzyskiwane przy średnich długościach list tabu oraz przy 250-500 iteracjach, a czas obliczeń jest dość zróżnicowany, z najkrótszymi czasami przy mniejszych wartościach liczby iteracji. Sąsiedztwo "reverse" znów okazało się zdecydowanie najlepszym wyborem dla tego problemu, niezależnie od parametrów.

10 najgorszych wyników:

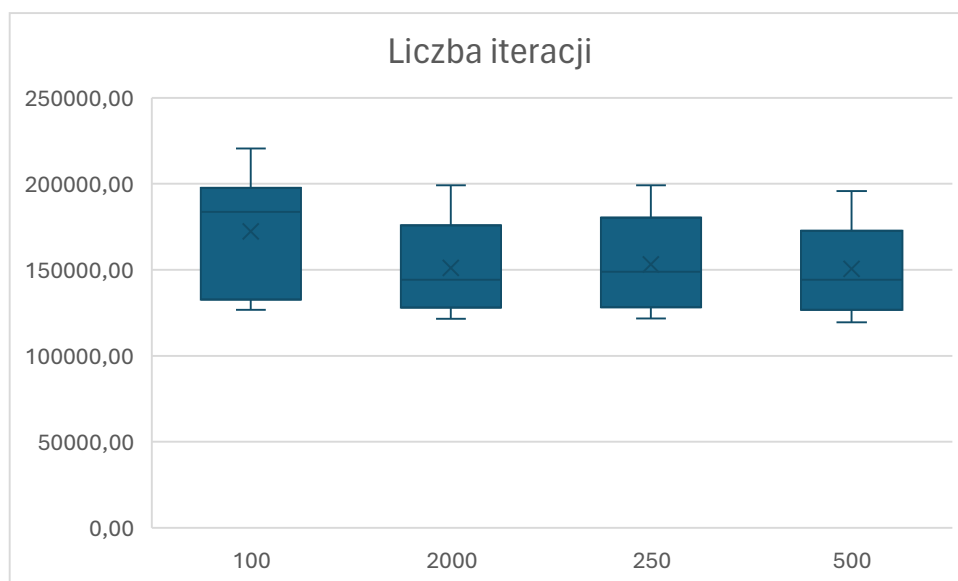
Liczba iteracji	Liczba iteracji bez poprawy	Długość listy tabu	Rodzaj sąsiedztwa	Dystans	Czas (sekundy)
100	50	100	swap	220576,13	2,21
100	25	100	swap	214839,95	2,13
100	25	200	swap	213938,93	2,32
100	200	50	swap	209099,99	1,80
100	100	100	swap	208904,12	2,25
100	50	200	swap	205141,79	2,40
100	50	25	swap	205111,40	1,50
100	25	100	insert	202920,97	2,38
100	100	100	insert	202813,98	2,25
100	200	525	swap	202660,38	1,49

Trasa najgorszego rozwiązania:

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

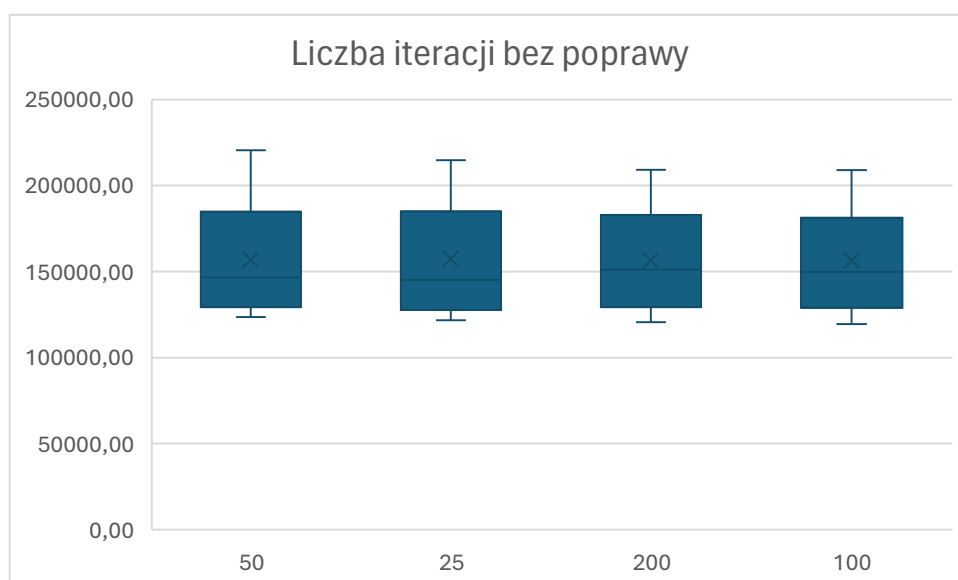
Wyniki dla dziesięciu najgorszych konfiguracji wskazują, że sąsiedztwo "swap" generuje zdecydowanie wyższe dystanse w porównaniu do innych metod, z wartościami powyżej 200000, co wskazuje na jego mniejszą efektywność. Długość listy tabu oraz liczba iteracji bez poprawy nie mają dużego wpływu na poprawę jakości rozwiązania, a czas obliczeń jest stosunkowo krótki, z najniższymi wartościami przy mniejszych długościach list tabu. Dla sąsiedztwa "insert" również uzyskano wysokie dystanse, choć średnia trasa była nieco lepsza niż w przypadku "swap".

Analiza wpływu wartości parametrów przy użyciu wykresów pudełkowych:



Liczba iteracji	Średnia trasa	Minimum	Maksimum
100	172325,3	126820,2	220576,1
250	153192,7	121656,0	199196,6
500	150511,0	119382,1	195707,8
2000	151083,7	121587,7	199208,9

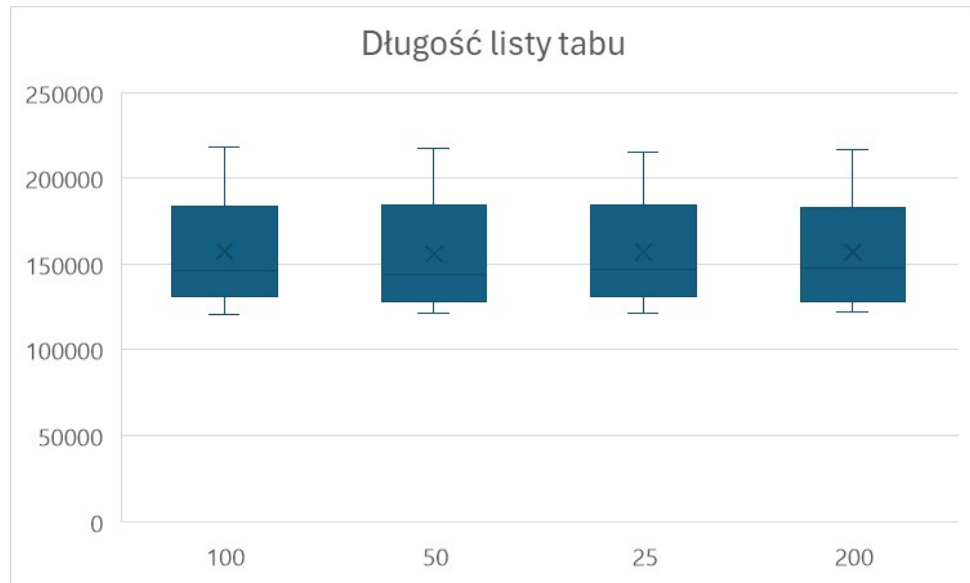
Wyniki pokazują, że średnia trasa jest najwyższa przy 100 iteracjach, a przy zwiększeniu liczby iteracji wartość średnia spada, osiągając najniższą średnią przy 500 iteracjach. Mimo tego, rozrzut wyników pozostaje stosunkowo wysoki, zwłaszcza przy mniejszych liczbach iteracji, co sugeruje, że większa liczba iteracji pomaga w stabilizacji, ale niekoniecznie poprawia jakość rozwiązania.



Liczba iteracji bez poprawy	Średnia trasa	Minimum	Maksimum
25	157406,6	121656,0	214839,9
50	156908,1	123546,9	220576,1
100	156348,6	119382,1	208904,1

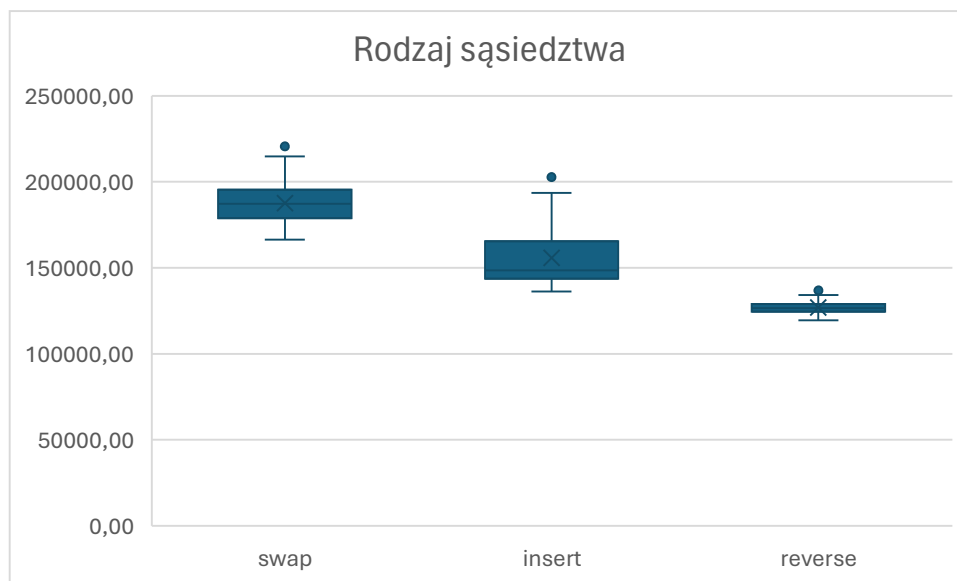
200	156449,40	120678,7	209099,9
-----	-----------	----------	----------

Wyniki pokazują, że liczba iteracji bez poprawy ma minimalny wpływ na średnią trasę, która utrzymuje się na poziomie około 156000, niezależnie od jej wartości. Zwiększanie liczby iteracji bez poprawy nie prowadzi do znaczącej poprawy jakości rozwiązania, a rozrzut wyników pozostaje stosunkowo duży, szczególnie w przypadku mniejszych wartości iteracji bez poprawy, gdzie maksymalne dystanse osiągają wyższe wartości.



Długość listy tabu	Średnia trasa	Minimum	Maksimum
25	157218,2	119382,1	205111,4
50	156515,4	122784,4	209099,9
100	156984,5	120678,7	220576,1
200	156394,6	121587,7	213938,93

Wyniki wskazują, że długość listy tabu ma minimalny wpływ na średnią trasę, której wartość utrzymuje się na poziomie około 156000. Najniższe minimum występuje przy długości listy tabu 25, a maksimum waha się w dość szerokim zakresie, osiągając najwyższą wartość przy długości 100. Zwiększenie długości listy tabu do 200 prowadzi do nieznacznej poprawy stabilności, ale nie wpływa istotnie na jakość rozwiązania.



Rodzaj sąsiedztwa	Średnia trasa	Minimum	Maksimum
swap	187571,2	166288,8	220576,1
reverse	127011,1	119382,1	137341,1
insert	155755,3	136204,4	202920,9

Wyniki pokazują, że najlepsze wyniki uzyskano przy sąsiedztwie "reverse", które charakteryzuje się najniższą średnią trasą oraz najmniejszym rozrzutem wyników. Sąsiedztwo "insert" osiągnęło gorsze wyniki, a jego rozrzut był większy, co wskazuje na mniej stabilne rozwiązania. Najgorsze wyniki uzyskano przy sąsiedztwie "swap", gdzie średnia trasa wyniosła aż 187571,2, a maksimum sięgało 220576,1, co świadczy o jego niższej efektywności w tym przypadku.

Podsumowanie TS

W przeprowadzonych testach na różnych macierzach widać, że parametr dotyczący rodzaju sąsiedztwa ma kluczowy wpływ na jakość wyników. Najlepsze rozwiązania osiągnane 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

Można zauważyć, że wszystkie dziesięć najlepszych wyników uzyskanych zostało za pomocą metody reverse. Liczba restartów jest bardziej zróżnicowana, choć w tabeli znajdują się raczej wyższe wartości. Maksymalna liczba iteracji bez poprawy zdaje się nie mieć wpływu na uzyskiwanie najlepszych wyników, natomiast maksymalna liczba iteracji zazwyczaj wynosiła 20000 lub 10000, czyli wartości wysokie.

Dziesięć najgorszych wyników:

Najgorszy Koszt	Rodzaj Sąsiedztwa	Liczba Restartów	Maks. Iteracje Bez Poprawy	Maksymalna liczba Iteracji
17273	swap	20	500	1000
17128	swap	50	5000	1000
16986	swap	20	2500	1000
16923	swap	20	5000	1000
16460	swap	20	10000	1000
16455	swap	50	10000	1000
16384	swap	100	2500	1000
16137	swap	100	5000	1000
16071	swap	500	5000	1000
15920	swap	100	500	1000

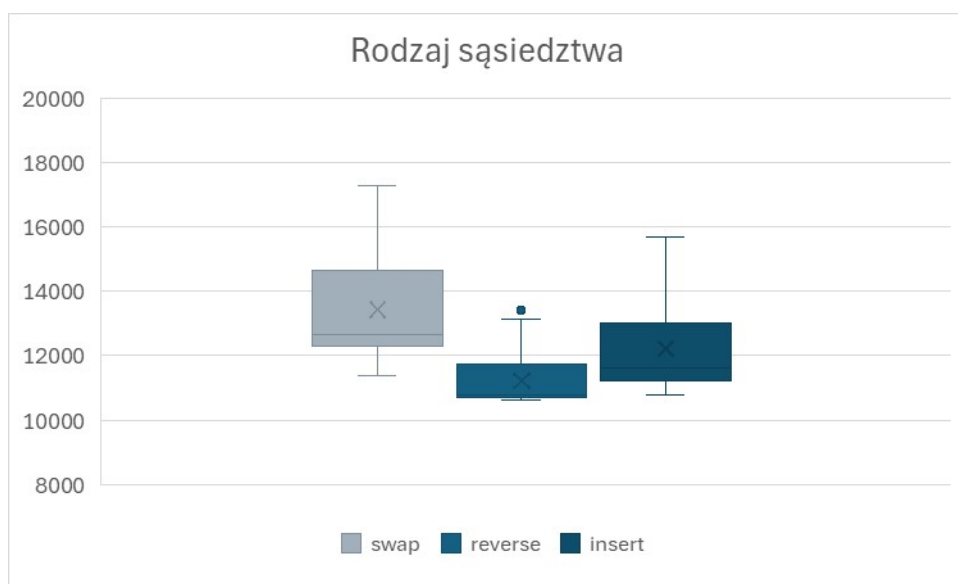
Najgorsza trasa:

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

Czas wykonania dla najgorszej trasy: 0,028 sekundy

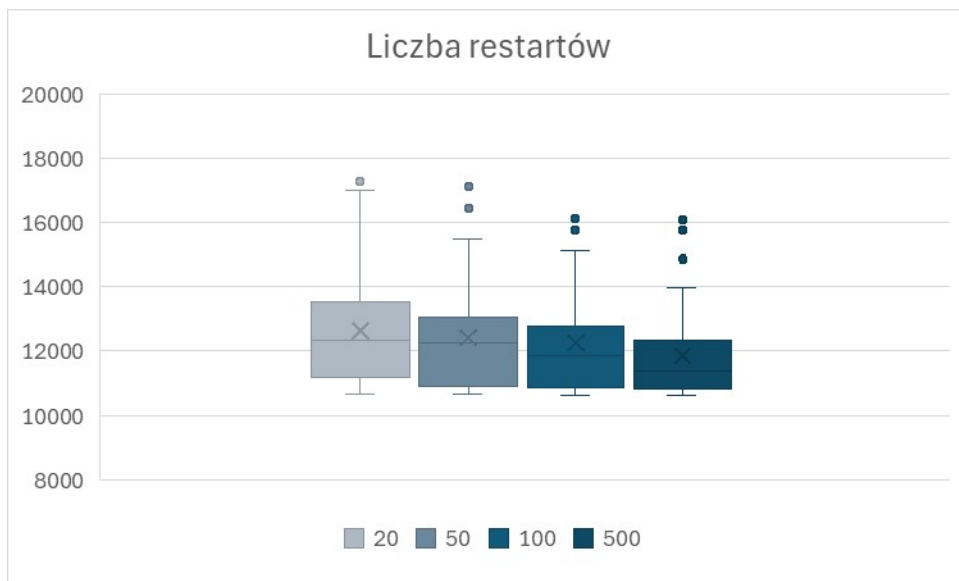
Wszystkie dziesięć najgorszych wyników uzyskanych zostało dla rodzaju sąsiedztwa “swap”, co sugeruje jego najgorsze radzenie sobie w zadanym algorytmie. Liczba restartów jest raczej niska, choć w tabeli znajdują się także wyższe wartości. Maksymalna liczba iteracji dla każdej z przedstawionych kombinacji wynosiła 1000, natomiast maksymalna liczba iteracji bez poprawy ponownie nie zdaje się wykazywać jakiegokolwiek zależności. Wartym zauważenia jest, że najgorsza kombinacja wykonywała się znacznie krócej niż najlepsza uzyskana kombinacja, co wynika przede wszystkim z dużych różnic w liczbie restartów i liczbie maksymalnych iteracji.

Wpływ parametrów na wyniki:



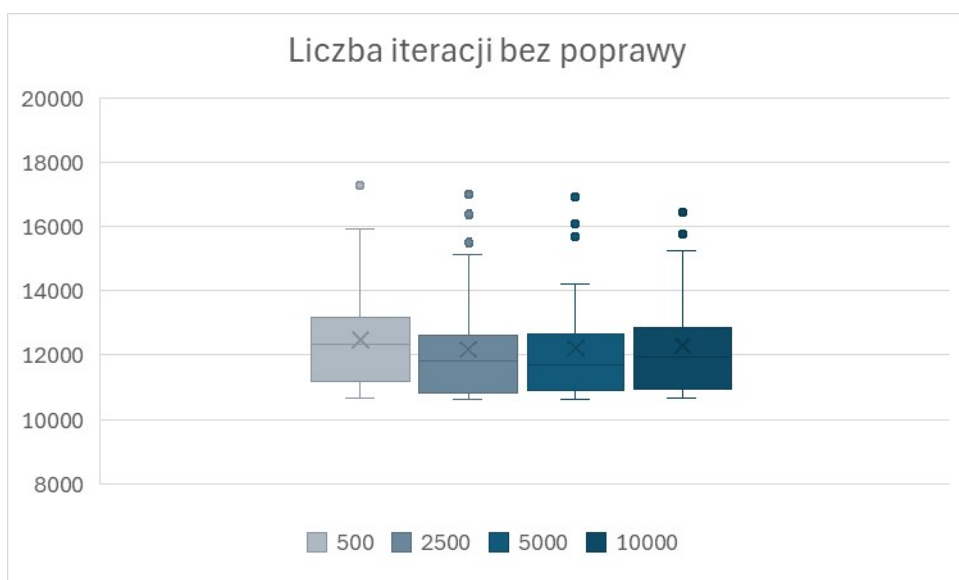
Rodzaj sąsiedztwa	Średnia trasa	Minimum	Maksimum
swap	13433	11388	17273
reverse	11225,13	10628	13400
insert	12212,27	10784	15681

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. Najślabsze wyniki osiąga “swap”, posiadając najwyższą średnią, medianę oraz największy zakres wyników.



Liczba restartów	Średnia trasa	Minimum	Maksimum
20	12625,13	10671	17273
50	12413,40	10663	17128
100	12259,35	10628	16384
500	11862,65	10628	16071

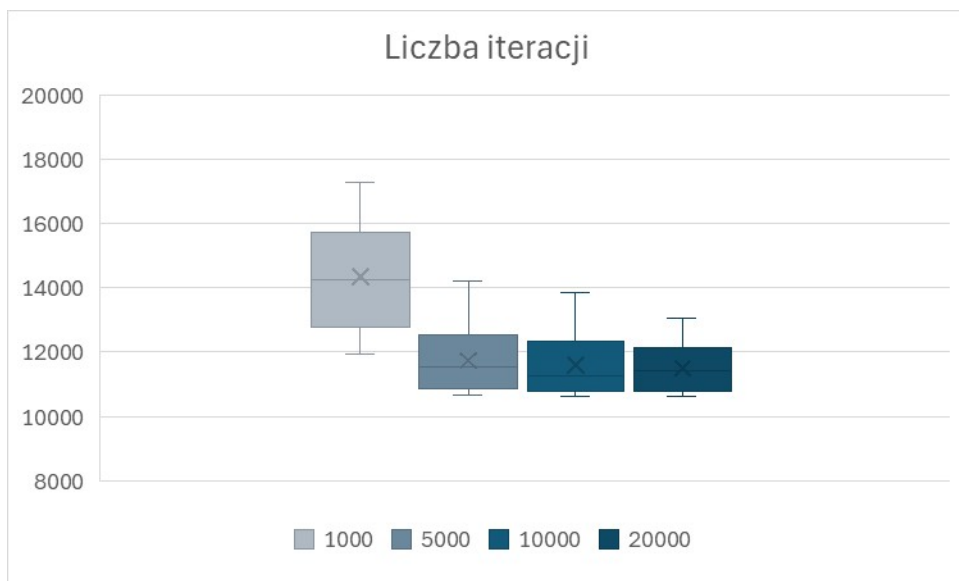
Widocznym jest, że wraz ze wzrostem liczby restartów średnie i mediany wyników maleją. Najlepsze (najniższe) wartości osiągnęte są dla 500 restartów, a także dla tej wartości najmniejszy jest ich rozrzut, czyli algorytm staje się bardziej stabilny. Im większa jest liczba restartów, tym większa jest szansa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



Liczba iteracji bez poprawy	Średnia trasa	Minimum	Maksimum
500	12483,19	10663	17273
2500	12174,38	10628	16986

5000	12214,48	10628	17128
10000	12288,48	10653	16460

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. Najlepsze wyniki (najniższa średnia i minimum) uzyskane zostały dla 2500 iteracji, czyli wcale nie wartości największej. Zwiększanie tego parametru w teorii pozwala algorytmowi eksplorować przestrzeń rozwiązań dłużej, ale różnice w wynikach są niewielkie.



Maksymalna liczba iteracji	Średnia trasa	Minimum	Maksimum
1000	14339,42	11933	17273
5000	11736,67	10663	14225
10000	11595,46	10638	13830
20000	11488,98	10628	13067

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. Pomiedzy 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

Wszystkie wyniki z zestawienia uzyskane zostały przy użyciu ruchu reverse, co potwierdza jego największą skuteczność. Liczba restartów jest bardzo zróżnicowana, tak samo jak maksymalna liczby iteracji bez poprawy, jednak wyraźnie możemy zaobserwować, że w prawie wszystkich kombinacjach maksymalna liczba iteracji wynosiła 20000, czyli najlepsze wyniki uzyskane zostały dla największej liczby iteracji.

Dziesięć najgorszych wyników:

Najgorszy Koszt	Rodzaj Sąsiedztwa	Liczba Restartów	Maks. Iteracje Bez Poprawy	Maksymalna liczba Iteracji
246845,5	swap	20	2500	1000
246355,9	swap	50	10000	1000
244602,3	swap	20	5000	1000
239302,8	swap	20	500	1000
233467,3	swap	100	500	1000
232765,8	swap	50	500	1000
231948,7	swap	100	10000	1000
231713,4	swap	500	10000	1000
231404,6	swap	50	5000	1000
230827,5	swap	500	500	1000

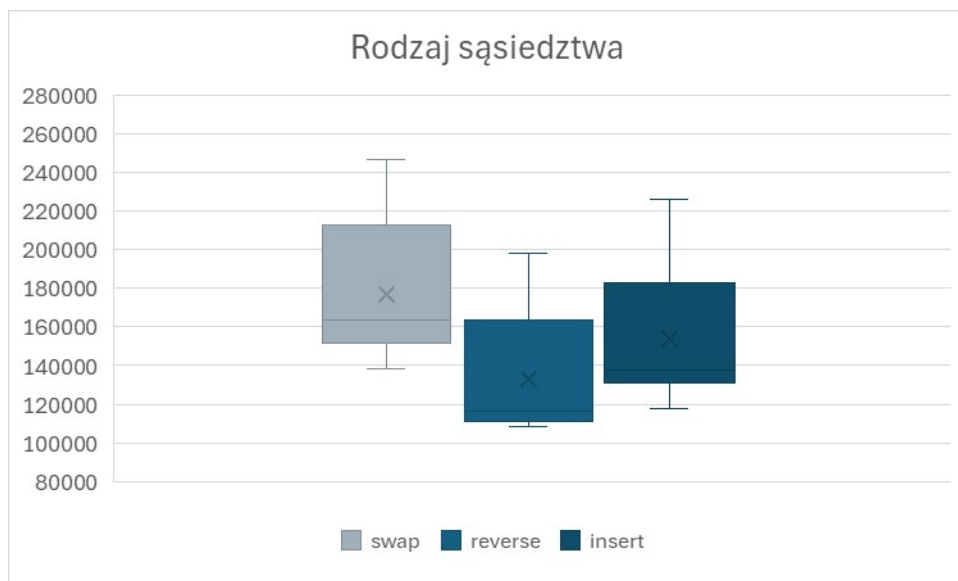
Najgorsza trasa:

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

Czas wykonania dla najgorszej trasy: 0,013 sekundy

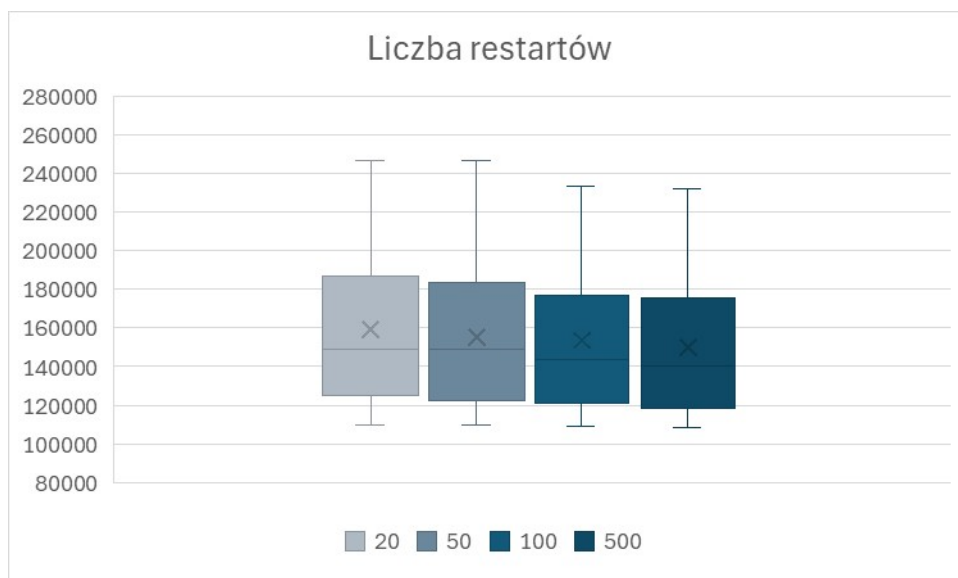
Wszystkie wyniki w zestawieniu najgorszych tras uzyskane zostały za pomocą ruchu swap i maksymalnej liczby iteracji ustawionej na 1000, co pokrywa się z wynikami dla Macierzy 48. Liczba restartów i maksymalnej liczby iteracji bez poprawy jest bardziej zróżnicowana i trudno jest zauważyć jakąkolwiek zależność. Wartym podkreślenia jest znacząca różnica w czasach wykonywania najgorszej i najlepszej kombinacji (0,013 s i 7,77 s).

Wpływ parametrów na wyniki:



Rodzaj sąsiedztwa	Średnia trasa	Minimum	Maksimum
swap	176866,76	138120,45	246845,53
reverse	132820,97	108644,97	198097,35
insert	154047,23	117530,83	225859,91

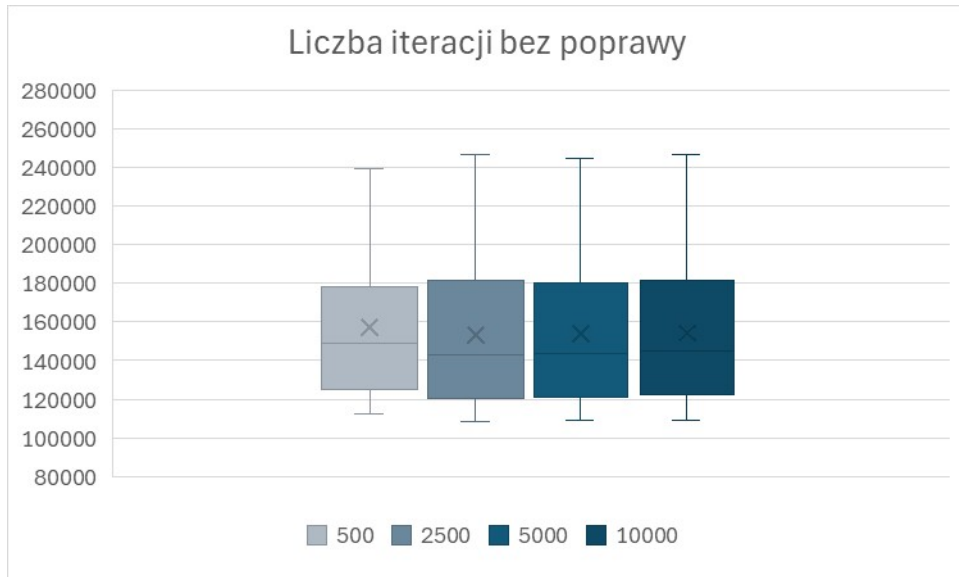
Ponownie zauważyć można najniższe wyniki dla ruchu “reverse”, pośrednie dla ruchu “insert” i najgorsze dla ruchu “swap”.



Liczba restartów	Średnia trasa	Minimum	Maksimum
20	159304,97	109540,74	246845,53
50	155368,98	109873,27	246355,85
100	153584,34	109332,96	233467,33

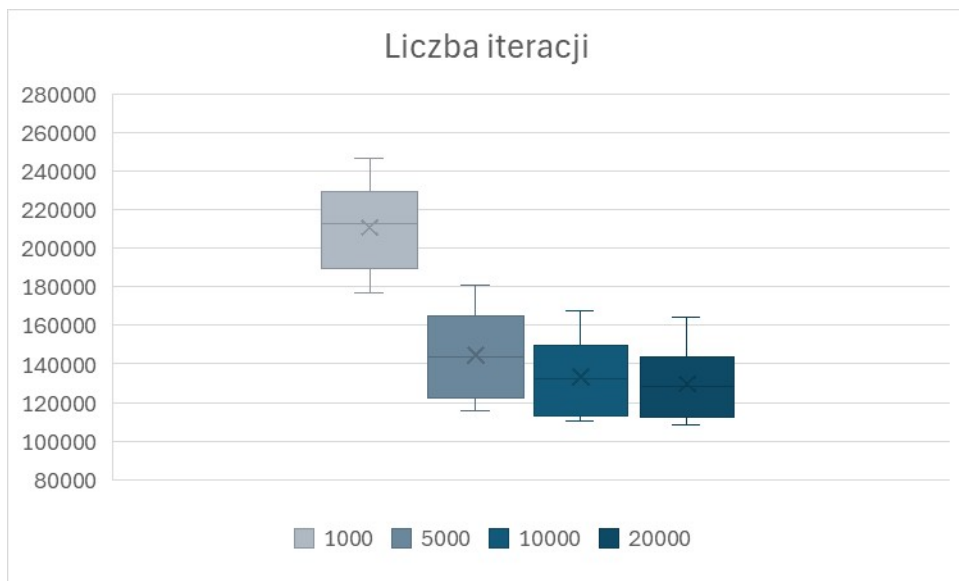
500	150054,98	108644,97	231713,38
-----	-----------	-----------	-----------

Ponownie wraz ze wzrostem liczby restartów wyniki polepszają się nieznacznie, co może sugerować słaby, acz istniejący wpływ tego parametru na wyniki.



Liczba iteracji bez poprawy	Średnia trasa	Minimum	Maksimum
500	157066,25	112476,44	239302,85
2500	153248,83	108644,97	246845,53
5000	153766,30	109278,46	244602,26
10000	154231,89	109335,25	246355,85

Tutaj też 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. Najlepsze wyniki (najniższa średnia i minimum) ponownie zostały uzyskane dla maksymalnej liczby iteracji bez poprawy wynoszącej 2500.



Maksymalna liczba iteracji	Średnia trasa	Minimum	Maksimum
1000	210812,47	177084,88	246845,53
5000	144704,09	115596,45	181049,81
10000	133303,33	110435,38	167495,90
20000	129493,38	108644,97	164186,23

Zdecydowanie najgorsze wyniki osiągnane są dla najmniejszej liczby maksymalnych iteracji (1000). Wraz ze wzrostem liczby iteracji średnia długość trasy zmniejsza się, choć różnicę między wartością 10000 i 20000 jest już ciężiej zauważyć na wykresie.

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

Zestawienie najlepszych wyników dla macierzy 127 charakteryzuje się tymi samymi zależnościami co przy mniejszych macierzach. Wszystkie najkrótsze trasy wyznaczone zostały przy użyciu ruchu reverse i maksymalnej liczbie iteracji wynoszącej 20000. Wartości maksymalnej liczby iteracji bez poprawy są zróżnicowane, acz należą do tych wyższych, natomiast przy liczbie restartów nie widać żadnej zależności.

Dziesięć najgorszych wyników:

Najgorszy Koszt	Rodzaj Sąsiedztwa	Liczba Restartów	Maks. Iteracje Bez Poprawy	Maksymalna liczba Iteracji
330779,7	swap	20	2500	1000
330467,1	swap	20	5000	1000
327468,8	swap	20	10000	1000
323396,7	swap	50	500	1000
322348,7	swap	100	500	1000
320948,8	swap	50	5000	1000
319238,6	insert	20	10000	1000
319234,1	insert	50	500	1000
318000,5	swap	100	10000	1000
317639,7	swap	20	500	1000

Najgorsza trasa:

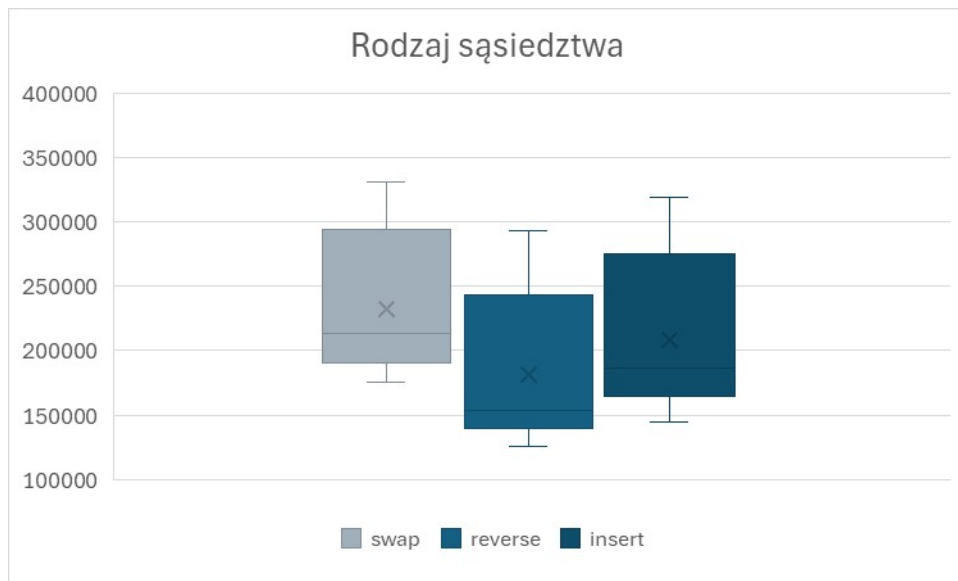
92 → 59 → 24 → 23 → 20 → 81 → 84 → 6 → 50 → 48 → 112 → 111 → 53 → 124 → 125 → 113 → 99 → 12 → 115 → 105 → 4 → 14 → 28 → 97 → 93 → 31 → 15 → 18 → 79 → 19 → 8 → 87 → 109 → 75 → 17 → 22 → 80 → 26 → 78 → 76 → 102 → 101 → 98 → 32 → 29 → 42 → 10 → 52 → 64 → 100 → 72 → 108 → 9 → 67 → 74 → 73 → 110 → 88 → 63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

Czas wykonania dla najgorszej trasy: 0,027 sekundy

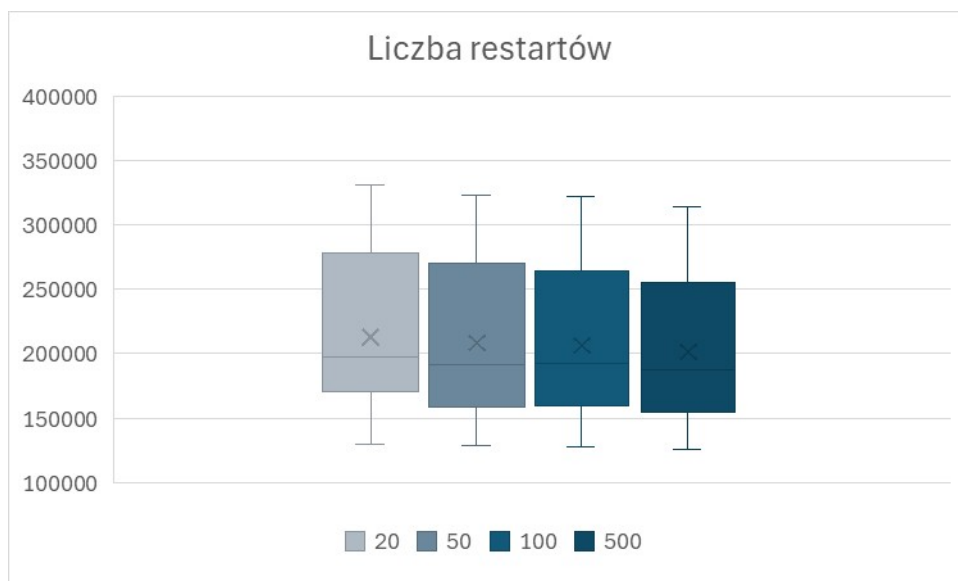
Dla macierzy 127 w zestawieniu najgorszych wyników pojawiły się pojedyncze kombinacje parametrów z ruchem insert, jednak wciąż jest ono zdominowane przez ruch swap, co potwierdza jego najniższą skuteczność. Maksymalna liczba iteracji zawsze wynosi 1000, a w maksymalnej liczbie iteracji bez poprawy i liczbie restartów trudno zauważyć jest znaczącą zależność, co pokrywa się z wynikami uzyskanymi wcześniej.

Wpływ parametrów na wyniki:

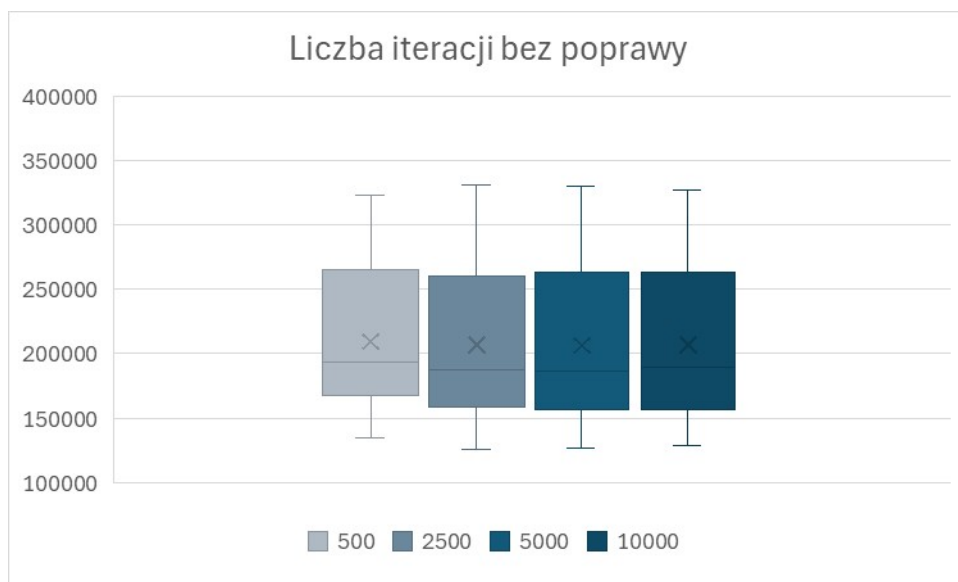
Wnioski dla poniższych wykresów pokrywają się z tymi wyciągniętymi na podstawie wyników dla Macierzy 48 i 76, nie będą więc ponownie powtarzane.



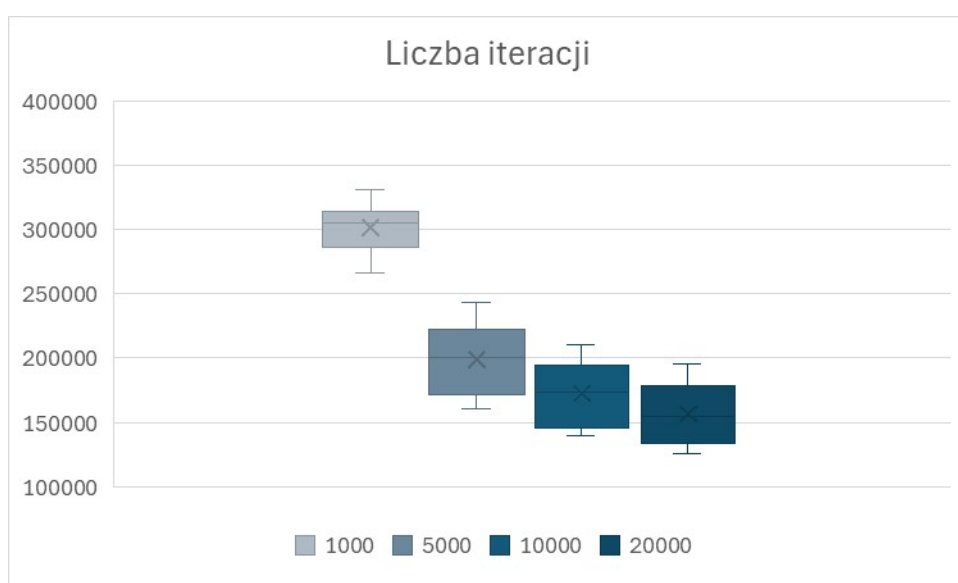
Rodzaj sąsiedztwa	Średnia trasa	Minimum	Maksimum
swap	232484,6	175782,7	330779,7
reverse	181283,5	125552,3	292978,8
insert	208342,7	144948,9	319238,6



Liczba restartów	Średnia trasa	Minimum	Maksimum
20	212970,92	129411,57	330779,69
50	208548,24	129004,68	323396,72
100	206549,30	127629,66	322348,65
500	201412,61	125552,30	314320,12



Liczba iteracji bez poprawy	Średnia trasa	Minimum	Maksimum
500	209590,72	134693,12	323396,72
2500	206769,71	125552,30	330779,69
5000	206420,55	126458,11	330467,13
10000	206700,11	129004,68	327468,75



Maksymalna liczba iteracji	Średnia trasa	Minimum	Maksimum
----------------------------	---------------	---------	----------

1000	301697,13	265771,30	330779,69
5000	198952,66	160020,33	243594,94
10000	172351,42	139323,28	210364,55
20000	156479,87	125552,30	195633,05

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.

Po początkowej zmianie parametrów wyniki uzyskane za pomocą algorytmu wspinaczki z multistartem okazały się dużo lepsze niż wyniki ustalone za pomocą dodatku Solver czy algorytmu najbliższego sąsiada.

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 utknięcia 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].

10 najgorszych wyników:

Temperatura Początkowa	Redukcja Temperatury	Iteracje na Temperaturze	Maksymalna Liczba Iteracji bez Poprawy	Rodzaj Sąsiedztwa	Długość Najlepszej Trasy	Średnia Długość Trasy
500	0,95	150	20	swap	31062	37800,98
1500	0,9	100	20	swap	30591	37881,31
1000	0,85	100	20	swap	30348	36581,79
1500	0,9	50	20	insert	30322	39508,77
1500	0,85	150	20	swap	30268	37977,84
1000	0,85	150	20	swap	30203	35776,03
1000	0,95	50	20	insert	30194	37570,97
1500	0,9	50	20	swap	30052	37146,69
1000	0,9	50	20	swap	29794	36150,76
2000	0,85	100	20	swap	29705	37435,69

Długość najgorszej trasy wynosi 31062.

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

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].

10 najgorszych wyników:

Temperatura Początkowa	Redukcja Temperatury	Iteracje na Temperaturze	Maksymalna Liczba Iteracji bez Poprawy	Rodzaj Sąsiedztwa	Długość Najlepszej Trasy	Średnia Długość Trasy
500	0,95	100	20	insert	492098	514597,3
500	0,95	150	20	swap	480663	511475,3
1000	0,85	200	20	swap	470113,1	481339,3
500	0,9	200	20	swap	467367	499517,4
1000	0,98	100	20	insert	456881,6	481671,4
1500	0,95	50	20	insert	454014	504031,8
2000	0,98	50	20	swap	449776,4	484405,3
500	0,98	100	20	insert	447321,9	489395
1500	0,98	100	20	swap	447271,6	473223,6
500	0,98	50	20	insert	446427,9	506994,4

Długość najgorszej trasy wynosi 492098.

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

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].

10 najgorszych wyników:

Temperatura Początkowa	Redukcja Temperatury	Iteracje na Temperaturze	Maksymalna Liczba Iteracji bez Poprawy	Rodzaj Sąsiedztwa	Długość Najlepszej Trasy	Średnia Długość Trasy
1500	0,85	50	20	swap	515976,6	561206,7
1000	0,85	200	20	insert	510944,6	551727,3
500	0,85	100	20	swap	508228,4	559935,4
500	0,95	200	20	insert	508189	555468,9
500	0,98	150	20	insert	501658,6	560100
1500	0,9	150	20	swap	498308,4	550407,9
1500	0,9	50	20	swap	498171,5	551463,2
1000	0,85	150	20	reverse	497174,6	540725,1
1500	0,85	50	20	insert	492958,6	538493,3
2000	0,85	200	20	insert	491525	536754,7

Długość najgorszej trasy wynosi 515976,6.

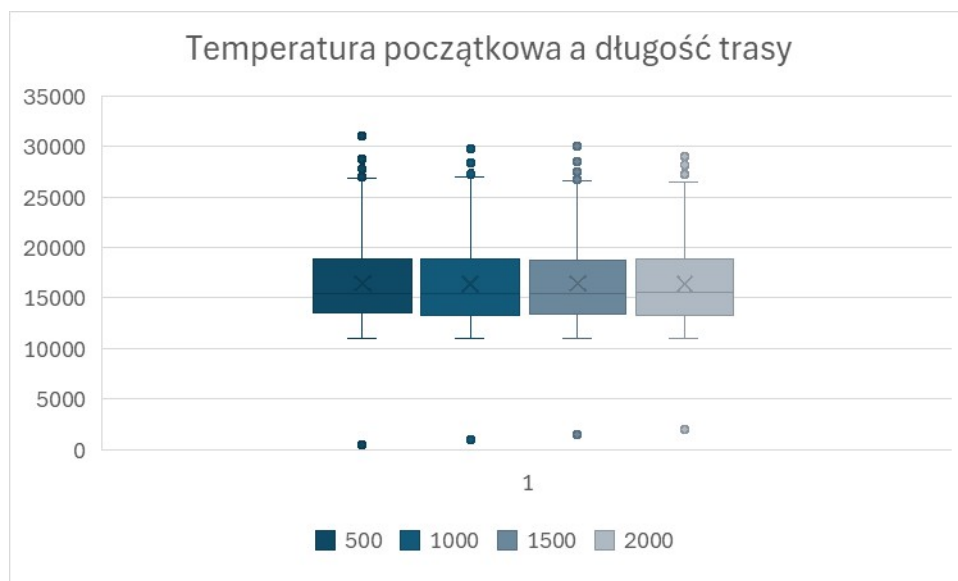
Trasa najgorszego rozwiązania: [23, 26, 22, 9, 58, 96, 83, 21, 43, 17, 108, 53, 123, 97, 30, 14, 36, 70, 116, 6, 110, 71, 119, 86, 5, 45, 121, 29, 115, 94, 80, 33, 39, 105, 1, 8, 124, 10, 42, 68, 7, 99, 11, 13, 50, 113, 88, 93, 103, 54, 98, 32, 107, 95, 114, 101, 63, 62, 67, 60, 25, 34, 2, 27, 76, 78,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

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

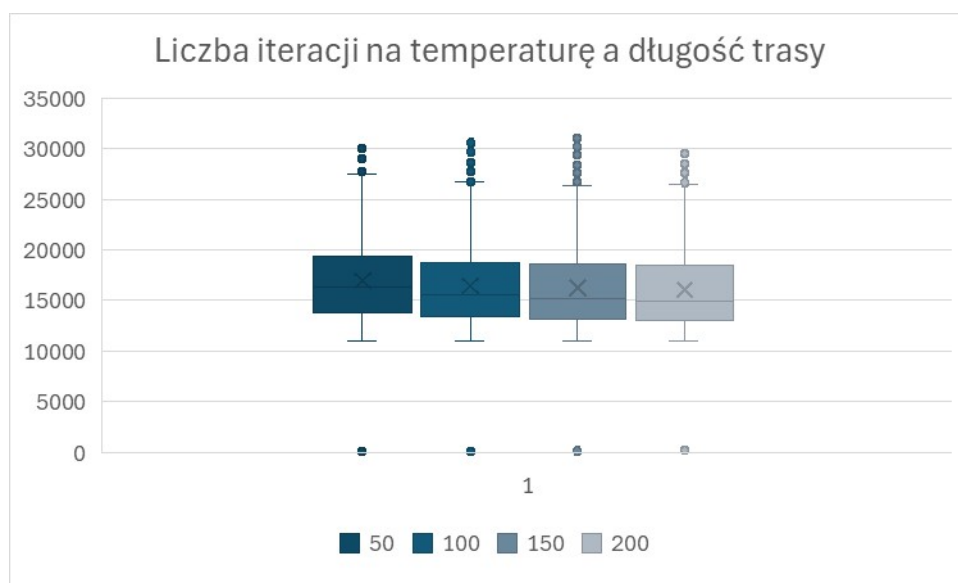
Podsumowanie SA

Wpływ poszczególnych parametrów na długość najlepszej trasy:

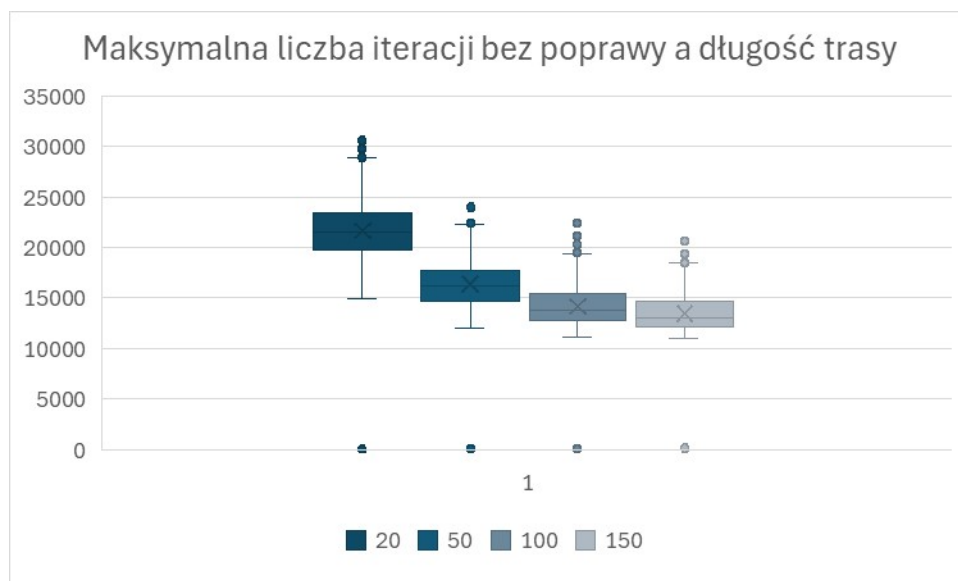
- dla 48 miast:



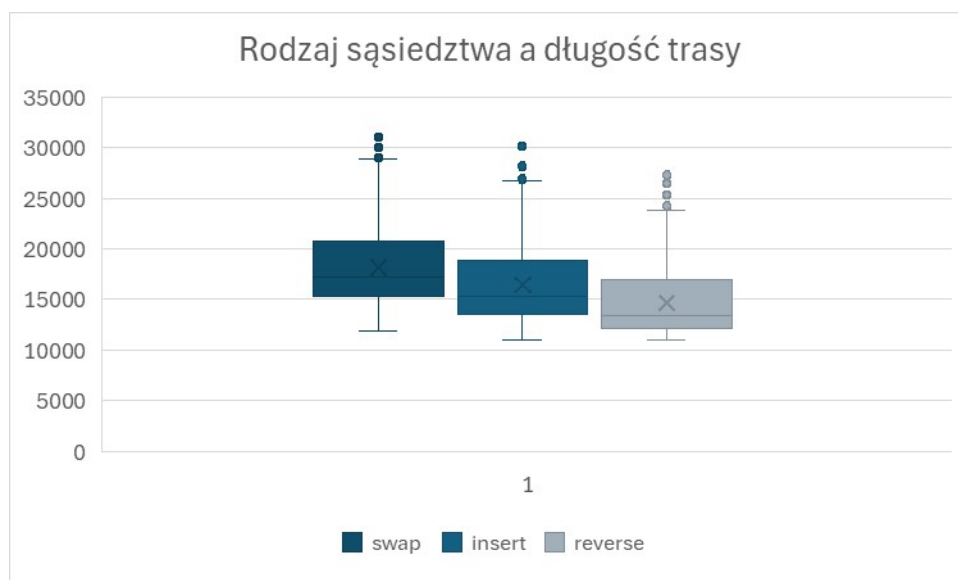
Temperatura początkowa	Średnia trasa	Minimum	Maksimum
500	16472,6	10982	31062
1000	16413,35	10955	30348
1500	16441,07	11030	30591
2000	16416,89	10933	29705



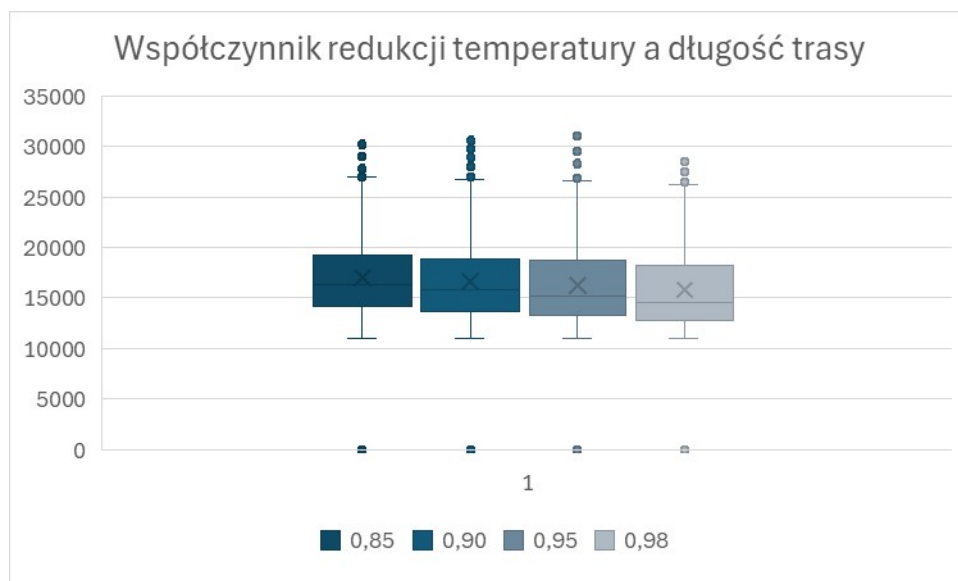
Liczba iteracji wykonywanych na temperaturze	Średnia długość trasy	Minimum	Maksimum
50	16954,8	11030	30322
100	16444,95	10933	30591
150	16268,92	10982	31062
200	16075,23	11003	29477



Maksymalna liczba iteracji bez poprawy	Średnia długość trasy	Minimum	Maksimum
20	21655,43	14971	31062
50	16389,88	12038	24661
100	14200,94	11076	22373
150	13497,65	10933	20660

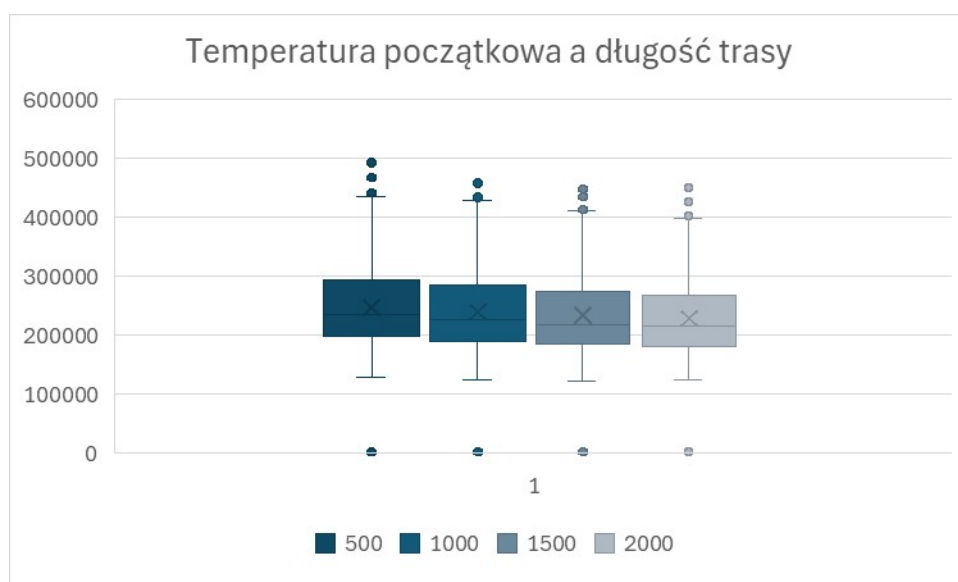


Rodzaj sąsiedztwa	Średnia długość trasy	Minimum	Maksimum
swap	11984,5	11854	12107
insert	16438,68	11003	30322
reverse	14695,05	10933	27308

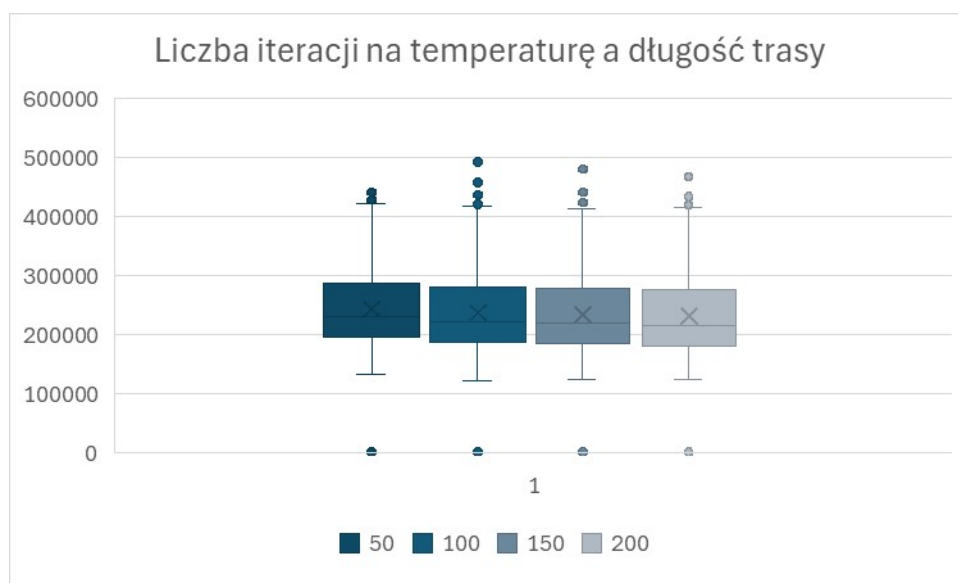


Współczynnik redukcji temperatury	Średnia długość trasy	Minimum	Maksimum
0,85	17026,8	11042	30348
0,9	16635,74	11053	30591
0,95	16242,34	11030	31062
0,98	15839,02	10933	29242

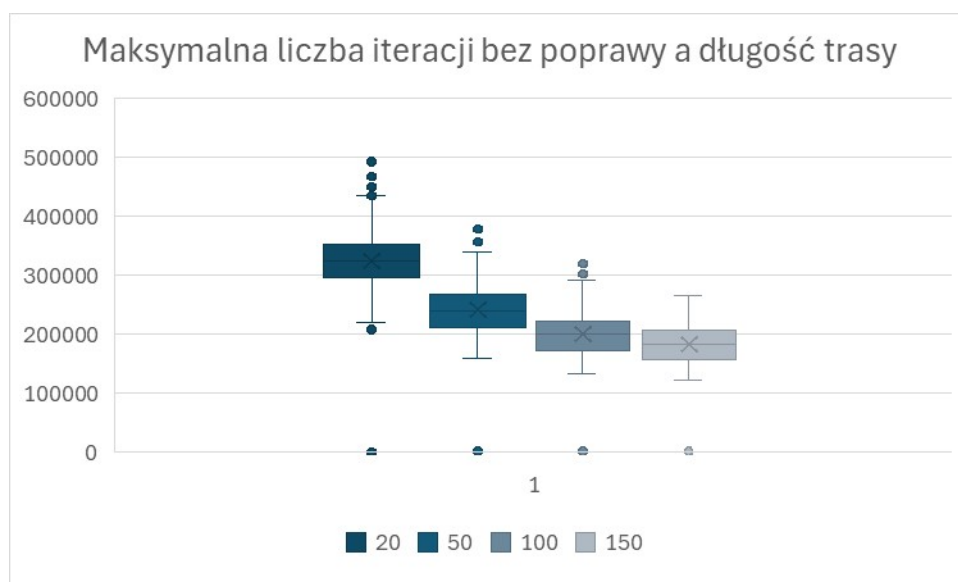
- dla 76 miast:



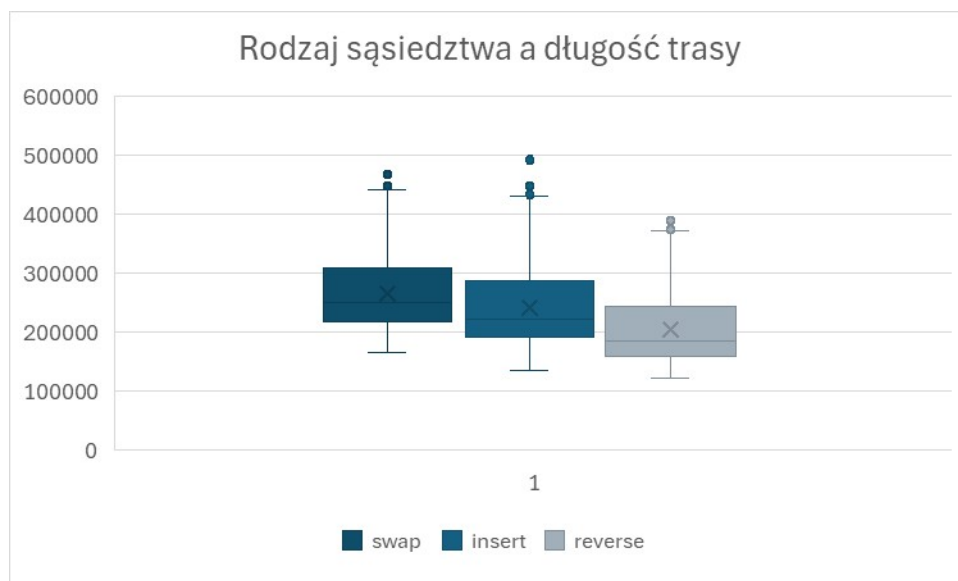
Temperatura początkowa	Średnia trasa	Minimum	Maksimum
500	246281,9	128616,1	492098
1000	238275,3	124159,6	470113,1
1500	232710,7	121687,9	454014
2000	227355,5	123707,3	449776,4



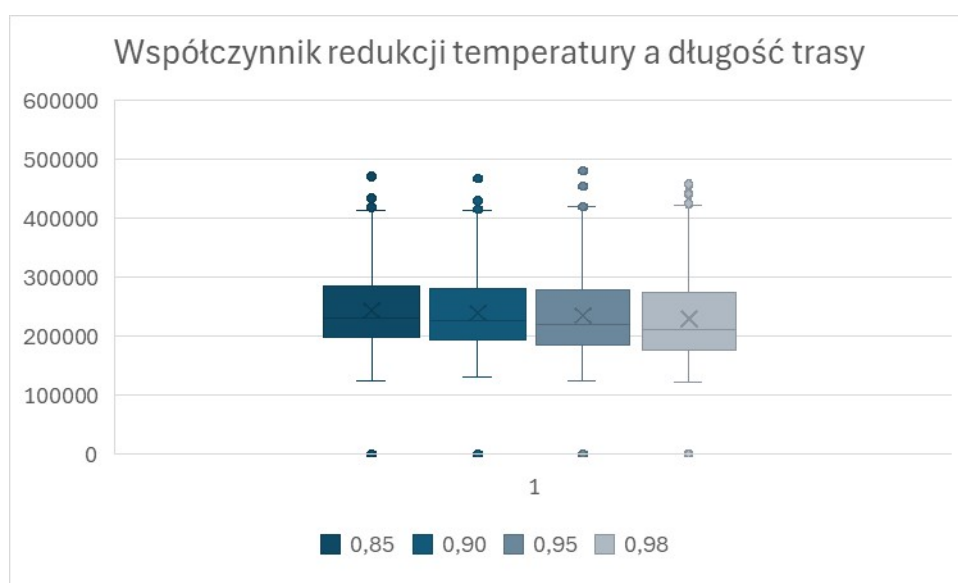
Liczba iteracji wykonywanych na temperaturze	Średnia długość trasy	Minimum	Maksimum
50	242983,2	131613,9	454014
100	236928,4	121687,9	492098
150	233434,1	124159,6	480663
200	231277,5	123707,3	470113,1



Maksymalna liczba iteracji bez poprawy	Średnia długość trasy	Minimum	Maksimum
20	323699	206936,8	492098
50	240430,1	157035,7	376914,6
100	198923,2	130809,9	318583,1
150	181671,3	121687,9	264540,2

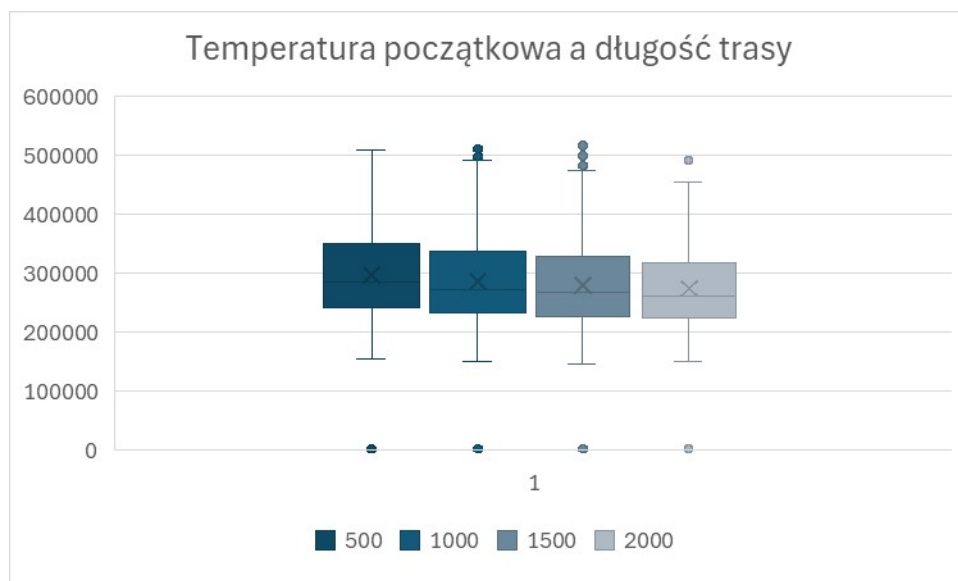


Rodzaj sąsiedztwa	Średnia długość trasy	Minimum	Maksimum
swap	180994,1	170765,9	188320,8
insert	240943,7	134995,6	492098
reverse	203338,8	121687,9	395433,2

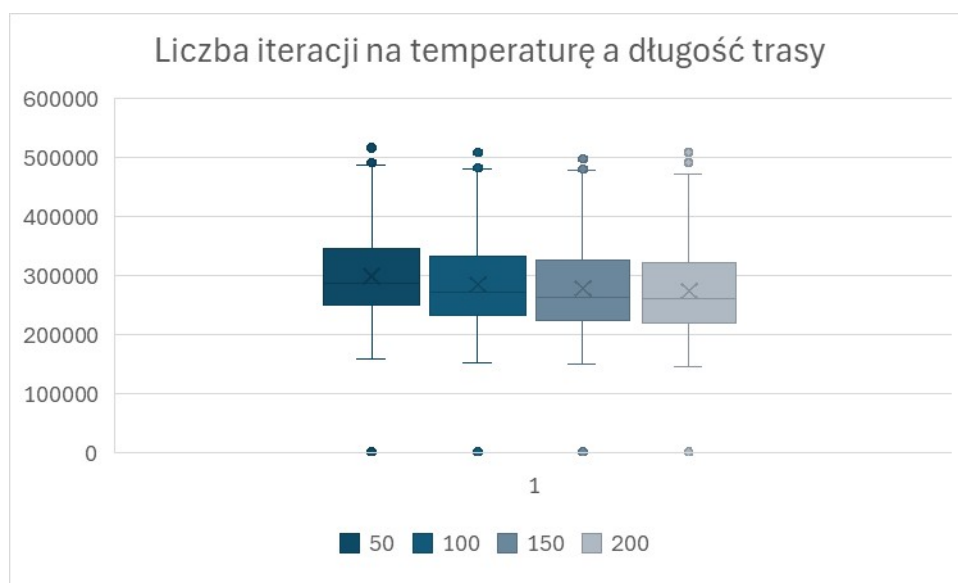


Współczynnik redukcji temperatury	Średnia długość trasy	Minimum	Maksimum
0,85	243085,1	123707,3	470113,1
0,9	238898,1	130050,6	467367
0,95	234392,8	124159,6	492098
0,98	228247,3	121687,9	456881,6

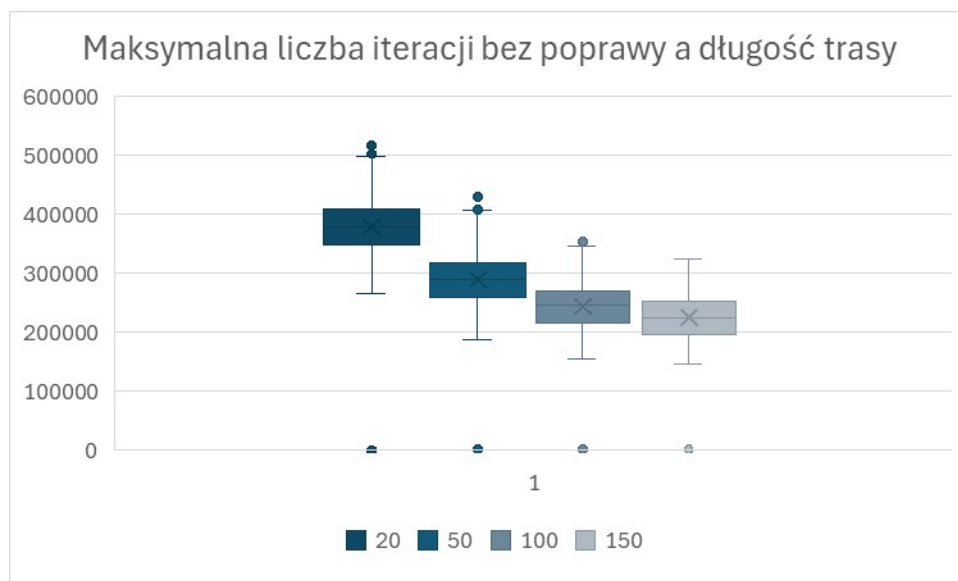
- dla 127 miast:



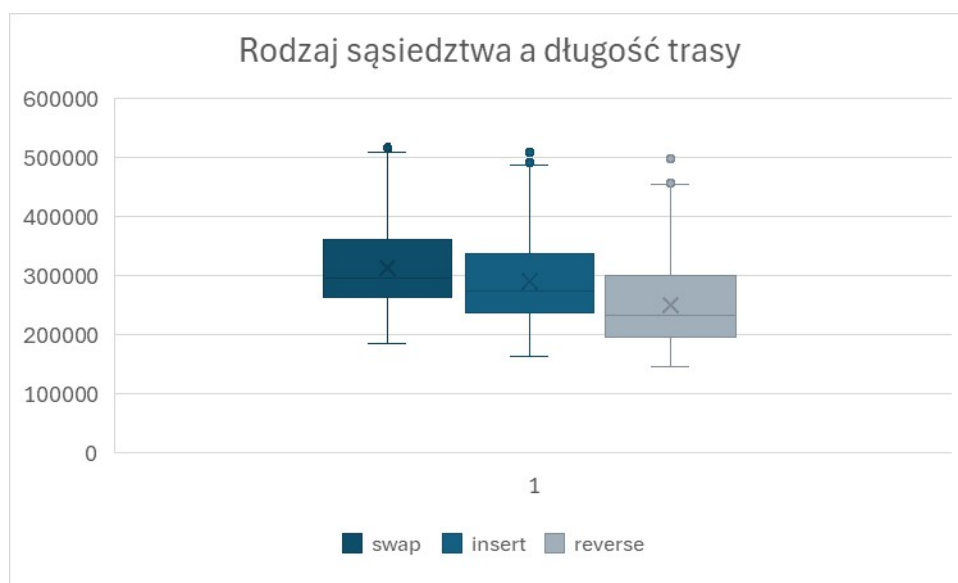
Temperatura początkowa	Średnia trasa	Minimum	Maksimum
500	296749,7	154029,5	508228,4
1000	284948,1	150079,4	510944,6
1500	278665	145070,2	515976,6
2000	273408,7	148808,8	491525



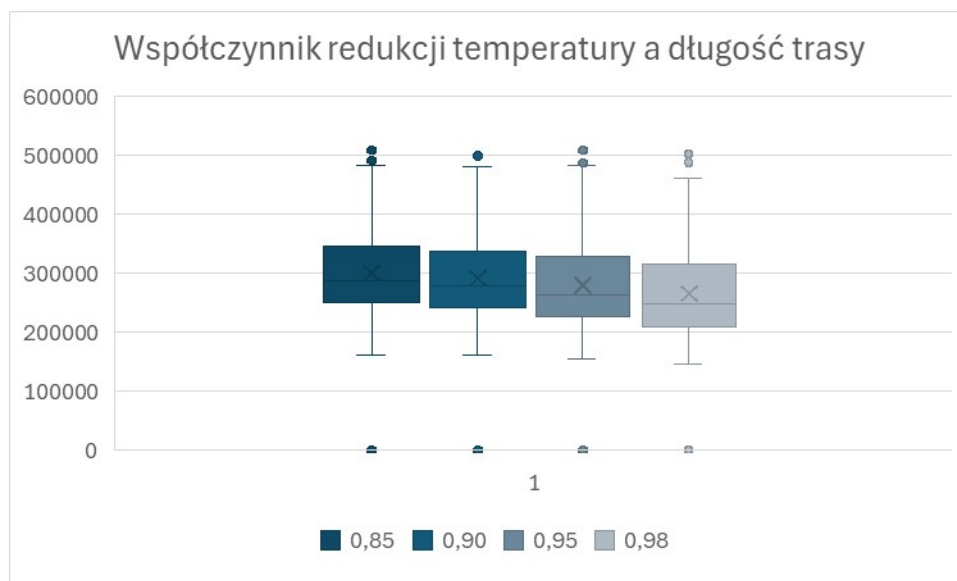
Liczba iteracji wykonywanych na temperaturze	Średnia długość trasy	Minimum	Maksimum
50	298205,6	158767,8	515976,6
100	284551,3	151250,8	508228,4
150	277799,8	150079,4	501658,6
200	273280,8	145070,2	510944,6



Maksymalna liczba iteracji bez poprawy	Średnia długość trasy	Minimum	Maksimum
20	378222,5	265191,4	515976,6
50	288889,9	186760,6	428949
100	242457	154677,3	363017,8
150	224268,1	145070,2	323231,3



Rodzaj sąsiedztwa	Średnia długość trasy	Minimum	Maksimum
swap	206105,2	184981,9	231796,2
insert	289104,1	162915,6	510944,6
reverse	249331,3	145070,2	497174,6



Współczynnik redukcji temperatury	Średnia długość trasy	Minimum	Maksimum
0,85	299517,7	161145	515976,6
0,9	290943,7	159109,1	498308,4
0,95	278764,8	153923,3	508189
0,98	264611,2	145070,2	501658,6

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

Zmianienie 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 kontekście wyznaczanie najlepszego kosztu.

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” }

PODSUMOWANIE

Macierz 48 miast

- Najlepszy wynik: Tabu Search (TS) i Wspinaczka z Multistartem (WS) – 10628
- Najgorszy wynik: Najbliższy Sąsiad (NN) – 12012

W tej macierzy Tabu Search (TS) oraz Wspinaczka z Multistartem (WS) uzyskały identyczny najlepszy wynik (10628). To sugeruje, że oba algorytmy dobrze radzą sobie z mniejszymi instancjami problemu.

Z kolei Najbliższy Sąsiad (NN) osiągnął znacznie gorszy wynik (12012), co pokazuje, że metoda ta, choć szybka, rzadko prowadzi do optymalnych tras.

Algorytm mrówkowy (AM) uzyskał 10861, co było lepsze od Symulowanego Wyżarzania (SA) (10933), ale nadal nie dorównało najlepszym metodom.

Macierz 76 miast

- Najlepszy wynik: Tabu Search (TS) – 108396
- Najgorszy wynik: Symulowane Wyżarzanie (SA) – 121687

Dla większej liczby miast Tabu Search (TS) ponownie wykazał się największą skutecznością, osiągając wynik 108396, co było najlepszym rezultatem w tej grupie.

Z kolei Symulowane Wyżarzanie (SA) wypadło najgorzej (121687)

Macierz 127 miast

- Najlepszy wynik: Tabu Search (TS) – 119382
- Najgorszy wynik: Symulowane Wyżarzanie (SA) – 145070

Dla największej macierzy Tabu Search (TS) ponownie zdeklasował inne algorytmy, uzyskując wynik 119382, co pokazuje jego wysoką skuteczność także dla bardziej złożonych problemów.

Symulowane Wyżarzanie (SA) osiągnęło najsłabszy wynik (145070), co wskazuje, że dla dużych instancji jego strategia stopniowego zmniejszania temperatury może nie być wystarczająco efektywna.

Podsumowanie algorytmów

- Tabu Search (TS) okazał się najlepszym algorytmem w każdej macierzy, co sugeruje, że dobrze radzi sobie zarówno z małymi, jak i dużymi instancjami problemu.
- Wspinaczka z Multistartem (WS) była konkurencyjna dla małej macierzy (48 miast), ale dla większych instancji nie utrzymała przewagi.
- Algorytm Genetyczny (GA) był solidnym rozwiązaniem, ale w żadnym przypadku nie osiągnął najlepszych wyników.
- Algorytm Mrówkowy (AM) wypadł przeciętnie – jego skuteczność była ograniczona przez liczbę kombinacji.
- Symulowane Wyżarzanie (SA) osiągnęło najsłabsze wyniki dla większych instancji, co sugeruje, że jego strategia wymagałaby dłuższego czasu działania, aby osiągnąć lepsze wyniki.
- Najbliższy Sąsiad (NN) zawsze wypadł najgorzej lub blisko najgorszego wyniku, co potwierdza, że jego heurystyka jest szybka, ale nieefektywna dla optymalnych tras.

Bibliografia

- https://brain.fuw.edu.pl/edu/index.php/Algorytmy_Genetyczne
- http://staff.iar.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