

# ***Zastosowanie algorytmów do optymalizacji problemu komiwojażera***

## Spis treści

1.	Opis projektu .....	2
2.	Wyniki dla Solvera .....	2
3.	Algorytm najbliższego sąsiada .....	3
4.	Algorytm symulowanego wyżarzania .....	5
5.	Algorytm wspinaczki z multistartem .....	26
6.	Algorytm genetyczny .....	39
7.	Algorytm Tabu Search .....	54
8.	Algorytm kolonii pszczoł .....	66
9.	Podsumowanie .....	80
10.	Bibliografia .....	82

## 1. Opis projektu

Celem projektu jest rozwiązanie problemu komiwojażera (TSP) za pomocą różnych algorytmów heurystycznych, w tym Algorytmu Najbliższego Sąsiada (NN), Wspinaczki z Multistartem (IHC), Symulowanego Wyżarzania (SA), Przeszukiwania Tabu (TS), Algorytmu Genetycznego (GA) oraz Algorytmu Kolonii Pszczół. Projekt zakłada zaimplementowanie tych metod, przeprowadzenie serii eksperymentów z różnymi wartościami parametrów (takimi jak liczba iteracji, temperatura początkowa, metoda krzyżowania – w zależności od algorytmu) oraz analizę uzyskanych wyników, w tym porównanie ich z rozwiązaniem uzyskanym za pomocą narzędzia Solver w Excelu.

## 2. Wyniki dla Solvera

W Excelu została wykonana optymalizacja trasy dla problemu komiwojażera za pomocą narzędzia Solver. Dla każdego zestawu danych proces powtórzono 10 razy. Poniżej zaprezentowane są wyniki w postaci długości tras uzyskanych w kolejnych próbach oraz trasy dla najlepszego wyniku.

### Liczba miast 48

1	2	3	4	5	6	7	8	9	10
10973	10792	10902	10917	10648	11032	11011	10981	10831	10688

#### Trasa:

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

**Średni wynik:** 10 877.5

### Liczba miast 76

1	2	3	4	5	6	7	8	9	10
116044.6	111876.6	111228	113119.9	112800.8	116737.5	114112.7	111239.6	111870.1	117085.6

#### Trasa:

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

**Średni wynik:** 113 611.5334

### Liczba miast 127

1	2	3	4	5	6	7	8	9	10
136012.15	137281.04	134084.27	128797.07	131433.84	131661.18	132018.89	130203.79	125806.44	133745.63

#### Trasa:

96, 109, 88, 87, 86, 85, 110, 70, 71, 68, 74, 73, 67, 104, 125, 89, 92, 99, 65, 66, 55, 47, 53, 49,

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

**Średni wynik:** 132 104.4289

### 3. Algorytm najbliższego sąsiada

Algorytm najbliższego sąsiada (Nearest Neighbor Algorithm, NN) to jedna z najprostszych heurystyk stosowanych do rozwiązywania problemu komiwojażera, w którym celem jest znalezienie najkrótszej możliwej trasy, odwiedzającej każde z zadanych miast dokładnie raz, a następnie wracającej do punktu początkowego. Algorytm najbliższego sąsiada polega na wybieraniu w każdym kroku miasta, które znajduje się najbliżej aktualnej lokalizacji. Proces ten powtarza się, aż wszystkie miasta zostaną odwiedzone. Choć jest to metoda szybka i prosta w implementacji, nie gwarantuje ona uzyskania optymalnego rozwiązania. Wynika to z faktu, że algorytm koncentruje się na lokalnej optymalności, co może prowadzić do wyników znacząco odbiegających od najbardziej efektywnej trasy.

W poniższych tabelach przedstawiono wyniki dla każdego zestawu miast, uwzględniając długości tras rozpoczynających się od różnych miast startowych.

#### Liczba miast 48

Czas wykonywania: 76.3241ms

1	12861	13	13123	25	13131	37	13447
2	12435	14	12956	26	12576	38	13198
3	13000	15	13285	27	13506	39	12617
4	12515	16	13176	28	13356	40	12918
5	12439	17	13495	29	12601	41	12962
6	13491	18	13503	30	13356	42	12435
7	13617	19	13546	31	13619	43	13356
8	12861	20	13452	32	12837	44	13882
9	12928	21	13319	33	12939	45	12103
10	12012	22	13160	34	12731	46	12908
11	12964	23	13251	35	12012	47	13250
12	13282	24	12020	36	13517	48	12918

#### Kolejność miast dla najkrótszych tras:

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

35, 45, 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

**Średni wynik:** 13 018.04

Algorytm NN nie uzyskał wyniku niższego niż Solver (10648), co może sugerować ograniczoną skuteczność tej metody w poszukiwaniu najlepszego rozwiązania.

### Liczba miast 76

Czas wykonywania: 132.9184ms

1	153461.9	20	157076	39	143815.5	58	151370.9
2	153150	21	146571.2	40	145677.4	59	150844.4
3	151762.4	22	149745.8	41	151361.5	60	150842.2
4	143098.5	23	148815.6	42	142883.7	61	151317
5	151944.1	24	146868.7	43	134688.8	62	153524.5
6	148347.8	25	149241.5	44	147532.3	63	151109.1
7	154496.5	26	142174.4	45	154841	64	151111.9
8	154575.7	27	141588.5	46	151006.2	65	156884
9	150803.2	28	151142.1	47	147901	66	156998.7
10	142395.4	29	141100.5	48	150277.9	67	151286.8
11	131607.5	30	137132.5	49	145860.2	68	151653.5
12	133895.7	31	142610.5	50	151665.2	69	154700.1
13	135900.8	32	134397.5	51	153822.7	70	149571.6
14	135402.4	33	137933	52	154235.4	71	154936.8
15	132173.7	34	145824.3	53	155042.7	72	155012.8
16	130921	35	142460.8	54	147720.2	73	154710.5
17	147223.2	36	138647.7	55	147136.2	74	135240.3
18	146281.9	37	139638.3	56	149660	75	150504.8
19	147762.2	38	140472.9	57	149900.3	76	150391.9

### Kolejność miast dla najkrótszej trasy:

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

Średni wynik: 147 180.1

W tym przypadku Solver również znalazł lepsze istniejące rozwiązanie dla tego problemu (111 228).

### Liczba miast 127

Czas wykonywania: 157.5322ms

1	135751.8	27	143520.8	53	149122.2	79	150091.4	105	150087.8
2	136503.4	28	144102.1	54	149556.1	80	143817.6	106	145227.7
3	137585.7	29	144283.2	55	149871.2	81	151068	107	138538.6
4	148627	30	143496.7	56	150464.9	82	150226.5	108	148896.2
5	150182.6	31	149502.5	57	140948	83	150345.3	109	146067.2
6	148797.9	32	143774.1	58	148005.9	84	151060.8	110	147752
7	136444.4	33	144451.4	59	148044.3	85	156358.6	111	144399.1
8	141328.5	34	143180.9	60	149043.5	86	156516.8	112	145058.4

9	144610.9	35	145824.9	61	148916.7	87	156573.9	113	146452.8
10	137533.3	36	143747.7	62	148855.8	88	156411	114	150321.4
11	148953	37	143911.6	63	148451.7	89	152143.3	115	150203.6
12	144129.4	38	138955.3	64	148022.2	90	148187.6	116	148297.9
13	150412.5	39	142508.4	65	149538.4	91	139600.5	117	133970.6
14	144229.3	40	141493.9	66	148691	92	150928	118	145465.2
15	148515.6	41	145974.7	67	151425.6	93	149467.2	119	149964.1
16	149013.6	42	141504.4	68	149439	94	147153.3	120	137313.2
17	148271.4	43	147608.7	69	151117.3	95	147185.5	121	141757.5
18	150137.7	44	140306.2	70	143786.6	96	146361.2	122	144526.6
19	148656.6	45	140237.1	71	143891.8	97	139171.9	123	146414.9
20	141123.5	46	146775.2	72	141211.4	98	139171.9	124	141860.8
21	148394.9	47	145411.8	73	149227.1	99	152060.3	125	148492.5
22	148342.4	48	145099.8	74	144178.3	100	148231.9	126	150307.6
23	141472.4	49	149316.2	75	143899.2	101	148247.9	127	144222.2
24	141607.1	50	144487.9	76	149224.3	102	149783.6		
25	144945.1	51	149654.7	77	149679.7	103	139852		
26	142642	52	141854.2	78	148251.5	104	146640.5		

#### Kolejność miast dla najkrótszej trasy:

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

Średni wynik: 146 175.7

Dla 127 miast algorytm NN również osiągnął wyższy wynik niż Solver (125 806,44).

## 4. Algorytm symulowanego wyżarzania

**Symulowane wyżarzanie (SA)** to metaheurystyczna metoda optymalizacji inspirowana procesem wyżarzania w metalurgii. Algorytm jest szczególnie efektywny w rozwiązywaniu problemów kombinatorycznych, takich jak problem komiwojażera. Działanie algorytmu rozpoczyna się od wyboru początkowego rozwiązania, którym jest losowa kombinacja miast. Następnie ustalana jest początkowa temperatura, która kontroluje prawdopodobieństwo akceptacji gorszych rozwiązań. W każdej iteracji generowane jest nowe rozwiązanie przez modyfikację bieżącej trasy: swap, insert, reverse. Po wygenerowaniu nowego rozwiązania obliczana jest długość trasy zarówno dla bieżącego, jak i nowego rozwiązania. Jeśli nowa trasa jest krótsza, zostaje zaakceptowana jako nowe rozwiązanie. W przypadku dłuższej trasy jest ona akceptowana z pewnym prawdopodobieństwem, które zależy od różnicy długości tras oraz aktualnej temperatury. Temperatura jest redukowana po każdej iteracji według wybranej metody schładzania (wybrano metodę geometryczną). Algorytm kończy działanie, gdy osiągnięta zostanie ustalona liczba iteracji.

Zmienne parametry:

- liczba iteracji,

- rodzaj sąsiedztwa,
- temperatura początkowa,
- współczynnik schładzania ( $\alpha$ ).

Z przyjętych wartości parametrów wygenerowano 192 różne kombinacje, z których każda została powtórzona 35 razy.

### Liczba miast 48

#### Wartości parametrów:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$
1000	insert	1000	0.01
5000	reverse	1500	0.025
10000	swap	2000	0.05
15000		3000	0.1

10 najlepszych wyników:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$	wynik
5000	insert	2000	0.1	10707
10000	insert	1500	0.01	10725
15000	insert	2000	0.01	10742
15000	insert	2000	0.05	10789
10000	reverse	1000	0.025	10794
10000	insert	1000	0.1	10862
5000	insert	3000	0.1	10875
10000	reverse	2000	0.025	10878
5000	insert	2000	0.01	10892
5000	reverse	3000	0.01	10893

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

Najlepszy wynik uzyskany za pomocą algorytmu symulowanego wyżarzania wyniósł 10 707, co oznacza, że był gorszy od rezultatu osiągniętego przy użyciu solvera. Co istotne, wynik ten został osiągnięty przy użyciu zaledwie 5000 iteracji, a nie największej liczby iteracji przewidzianej w testach. Warto również zauważyć, że wśród 10 najlepszych wyników żadna z kombinacji parametrów nie wykorzystywała metody zmiany sąsiedztwa typu swap. W przypadku pozostałych parametrów nie zaobserwowano wyraźnych zależności, które mogłyby wskazywać na przewagę konkretnej ich konfiguracji.

10 najgorszych wyników:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$	wynik
1000	swap	1000	0.1	20032
1000	swap	2000	0.05	20089
1000	swap	2000	0.05	20147
1000	swap	2000	0.025	20179
10000	swap	3000	0.05	20214
1000	swap	2000	0.1	20239
15000	swap	1000	0.05	20260
10000	swap	1500	0.05	20385
5000	swap	1000	0.1	20704
1000	swap	1000	0.025	21034

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

Najgorsze rozwiązanie uzyskane w algorytmie symulowanego wyżarzania wyniosło 21 034, co jest wynikiem gorszym niż najgorsze rozwiązanie osiągnięte za pomocą solvera. Co ciekawe, wszystkie 10 najsłabszych wyników zostało uzyskanych przy użyciu metody zmiany sąsiedztwa typu swap.



Średnie długości tras dla każdej kombinacji:

	iteracje	rodzaj sąsiedztwa	temp. początkowa	$\alpha$	średni wynik	średni czas (ms)
1.	5000	reverse	3000	0.01	11585.14	858.8
2.	10000	reverse	2000	0.025	11619.80	1647.9
3.	15000	reverse	1500	0.01	11636.83	2474.9
4.	15000	reverse	2000	0.01	11641.66	2499.7
5.	15000	reverse	3000	0.01	11644.23	2477.7
6.	10000	reverse	3000	0.01	11683.09	1693.7
7.	15000	reverse	1000	0.01	11716.23	2462.8
8.	5000	reverse	1500	0.1	11720.46	809.2
9.	5000	reverse	1000	0.01	11723.46	881.5
10.	5000	reverse	2000	0.01	11725.94	846.3
11.	5000	reverse	1500	0.01	11732.86	858.1
12.	5000	reverse	3000	0.05	11734.20	827.6
13.	15000	reverse	2000	0.1	11735.74	2456.2
14.	10000	reverse	1000	0.01	11741.06	1782.1
15.	10000	reverse	2000	0.01	11745.31	1679.9
16.	15000	reverse	1500	0.025	11776.31	2499.2
17.	10000	insert	1500	0.01	11779.83	1470.0
18.	10000	reverse	3000	0.025	11781.40	1645.2
19.	15000	reverse	2000	0.05	11784.97	2418.7
20.	15000	reverse	3000	0.025	11793.60	2458.9
21.	5000	reverse	1000	0.025	11795.34	836.0
22.	10000	reverse	1500	0.01	11801.23	1654.7
23.	5000	reverse	2000	0.1	11803.14	826.4
24.	5000	reverse	1000	0.05	11813.54	824.7
25.	5000	reverse	1500	0.025	11817.57	823.7
26.	15000	reverse	2000	0.025	11818.89	2480.1
27.	10000	reverse	1000	0.025	11819.43	1671.5
28.	5000	reverse	2000	0.025	11822.66	837.9
29.	5000	reverse	2000	0.05	11826.80	825.9
30.	15000	reverse	1000	0.1	11830.49	2446.7
31.	5000	reverse	1000	0.1	11832.91	836.0
32.	15000	reverse	1500	0.1	11833.77	2452.1
33.	15000	insert	2000	0.01	11857.46	2172.0
34.	15000	reverse	1000	0.025	11864.77	2427.0
35.	10000	reverse	2000	0.1	11868.14	1644.6
36.	10000	reverse	3000	0.05	11870.83	1636.9
37.	15000	reverse	3000	0.05	11871.89	2438.9
38.	10000	reverse	1500	0.05	11871.94	1630.1
39.	10000	reverse	1000	0.1	11890.06	1653.5
40.	10000	reverse	1500	0.025	11891.57	1664.3
41.	5000	reverse	3000	0.1	11904.00	832.4
42.	5000	reverse	3000	0.025	11907.29	834.1

	iteracje	rodzaj sąsiedztwa	temp. początkowa	$\alpha$	średni wynik	średni czas (ms)
97.	5000	insert	1000	0.025	12311.51	776.4
98.	10000	insert	1500	0.05	12317.83	1444.1
99.	15000	insert	2000	0.1	12326.46	2191.6
100.	10000	insert	3000	0.025	12337.37	1446.9
101.	5000	insert	2000	0.025	12345.91	777.0
102.	5000	insert	3000	0.01	12360.54	774.4
103.	5000	insert	1000	0.1	12365.71	747.0
104.	10000	insert	2000	0.05	12368.29	1441.9
105.	10000	insert	1000	0.05	12376.57	1445.2
106.	10000	insert	2000	0.025	12382.54	1469.3
107.	5000	insert	3000	0.05	12397.49	774.4
108.	5000	insert	1500	0.05	12437.46	836.9
109.	10000	insert	1500	0.025	12444.09	1446.8
110.	10000	insert	3000	0.1	12458.77	1449.1
111.	10000	insert	2000	0.1	12461.14	1440.4
112.	5000	insert	2000	0.05	12463.37	795.6
113.	1000	insert	1000	0.01	13099.23	177.8
114.	1000	insert	1500	0.05	13140.60	202.2
115.	1000	insert	3000	0.01	13157.51	154.5
116.	1000	insert	2000	0.01	13220.03	206.6
117.	1000	insert	1500	0.01	13259.77	233.2
118.	1000	insert	3000	0.1	13313.49	174.5
119.	1000	insert	1500	0.025	13316.97	202.3
120.	1000	insert	2000	0.1	13420.80	199.8
121.	1000	insert	2000	0.025	13436.83	203.2
122.	1000	insert	3000	0.05	13472.43	150.2
123.	1000	insert	1000	0.025	13478.57	193.0
124.	1000	insert	1500	0.1	13481.00	214.3
125.	1000	insert	1000	0.05	13495.03	185.6
126.	1000	insert	2000	0.05	13495.20	217.8
127.	1000	insert	3000	0.025	13509.14	148.0
128.	1000	insert	1000	0.1	13742.63	172.6
129.	10000	swap	3000	0.01	14278.57	1462.0
130.	15000	swap	2000	0.01	14709.20	2190.7
131.	5000	swap	2000	0.01	14709.63	762.0
132.	5000	swap	3000	0.01	14782.83	807.0
133.	15000	swap	1500	0.01	14842.54	2206.4
134.	15000	swap	3000	0.01	14896.69	2239.9
135.	5000	swap	1500	0.025	14985.54	816.8
136.	15000	swap	1000	0.01	14991.03	2218.3
137.	5000	swap	1500	0.01	15054.89	759.2
138.	10000	swap	1000	0.05	15108.23	1459.2

43.	10000	reverse	3000	0.1	11908.43	1634.3
44.	5000	reverse	1500	0.05	11921.34	815.3
45.	1000	reverse	1500	0.1	11921.71	169.0
46.	15000	insert	1500	0.01	11927.31	2159.1
47.	10000	reverse	2000	0.05	11929.49	1669.2
48.	1000	reverse	2000	0.01	11932.51	176.7
49.	15000	reverse	3000	0.1	11936.51	2420.5
50.	15000	reverse	1000	0.05	11936.83	2481.9
51.	10000	reverse	1000	0.05	11937.00	1626.6
52.	10000	reverse	1500	0.1	11941.97	1643.3
53.	10000	insert	2000	0.01	11945.17	1454.1
54.	1000	reverse	2000	0.025	11957.63	169.8
55.	1000	reverse	3000	0.025	11962.80	222.7
56.	1000	reverse	3000	0.05	11968.89	218.9
57.	1000	reverse	1000	0.01	11976.51	182.4
58.	15000	reverse	1500	0.05	11976.63	2549.1
59.	10000	insert	3000	0.01	11979.46	1460.4
60.	1000	reverse	1000	0.025	11979.71	184.3
61.	1000	reverse	3000	0.01	11990.14	202.6
62.	15000	insert	1000	0.01	12011.06	2181.7
63.	1000	reverse	1000	0.05	12028.54	173.1
64.	1000	reverse	1500	0.05	12032.31	171.4
65.	1000	reverse	3000	0.1	12045.69	252.1
66.	1000	reverse	1500	0.01	12050.03	170.9
67.	15000	insert	1500	0.1	12052.29	2190.1
68.	1000	reverse	1500	0.025	12058.86	174.2
69.	5000	insert	2000	0.01	12065.97	794.0
70.	1000	reverse	1000	0.1	12071.49	165.3
71.	1000	reverse	2000	0.05	12099.06	173.1
72.	15000	insert	3000	0.01	12099.91	2175.8
73.	15000	insert	1500	0.025	12107.23	2183.1
74.	15000	insert	3000	0.025	12141.66	2147.2
75.	5000	insert	1000	0.05	12150.03	761.5
76.	1000	reverse	2000	0.1	12151.69	165.4
77.	10000	insert	1000	0.1	12166.60	1427.4
78.	5000	insert	3000	0.025	12169.46	790.2
79.	5000	insert	1500	0.1	12172.66	780.9
80.	5000	insert	1000	0.01	12174.86	757.1
81.	15000	insert	2000	0.05	12188.80	2195.7
82.	5000	insert	1500	0.01	12195.94	833.0
83.	10000	insert	1000	0.025	12196.03	1481.8
84.	5000	insert	1500	0.025	12201.09	935.8
85.	10000	insert	1000	0.01	12202.46	1461.6
86.	15000	insert	2000	0.025	12223.43	2174.2
87.	15000	insert	3000	0.05	12239.54	2157.2
88.	15000	insert	1000	0.025	12255.34	2180.9

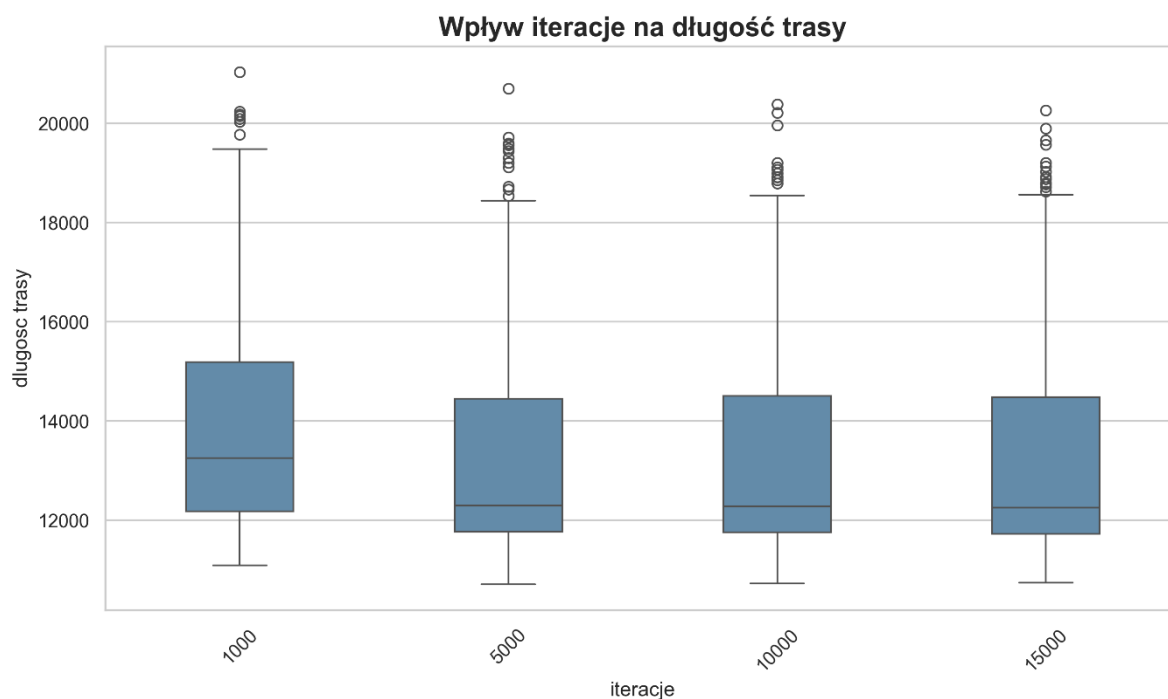
139.	10000	swap	2000	0.01	15108.60	1487.1
140.	5000	swap	1000	0.01	15110.23	838.7
141.	10000	swap	1000	0.1	15149.40	1470.3
142.	10000	swap	2000	0.1	15215.63	1459.2
143.	10000	swap	1000	0.025	15249.74	1459.0
144.	10000	swap	1000	0.01	15255.23	1467.3
145.	15000	swap	2000	0.025	15269.31	2213.4
146.	10000	swap	1500	0.01	15290.86	1480.7
147.	5000	swap	1000	0.025	15327.46	790.4
148.	5000	swap	1500	0.05	15338.09	804.6
149.	15000	swap	1500	0.025	15403.86	2173.5
150.	5000	swap	3000	0.025	15425.20	773.8
151.	5000	swap	1000	0.05	15429.37	857.0
152.	10000	swap	2000	0.025	15431.69	1476.5
153.	1000	swap	2000	0.01	15463.57	191.1
154.	1000	swap	1500	0.01	15481.57	159.0
155.	15000	swap	3000	0.025	15483.34	2188.5
156.	15000	swap	2000	0.05	15485.66	2164.4
157.	15000	swap	2000	0.1	15514.40	2195.4
158.	15000	swap	1000	0.1	15518.46	2208.2
159.	5000	swap	3000	0.05	15567.40	775.4
160.	15000	swap	1500	0.05	15590.57	2207.0
161.	5000	swap	1500	0.1	15606.97	829.2
162.	5000	swap	2000	0.025	15607.54	765.5
163.	1000	swap	1000	0.01	15615.34	156.0
164.	10000	swap	1500	0.025	15655.06	1455.0
165.	15000	swap	1000	0.05	15660.34	2179.5
166.	10000	swap	1500	0.1	15707.31	1478.6
167.	1000	swap	3000	0.01	15716.00	179.1
168.	15000	swap	3000	0.05	15720.11	2197.7
169.	10000	swap	2000	0.05	15729.34	1455.5
170.	15000	swap	1000	0.025	15731.63	2218.6
171.	10000	swap	3000	0.025	15733.74	1450.6
172.	5000	swap	1000	0.1	15743.20	805.8
173.	5000	swap	3000	0.1	15748.49	751.8
174.	10000	swap	1500	0.05	15847.29	1448.3
175.	15000	swap	3000	0.1	15897.89	2209.6
176.	1000	swap	1000	0.05	15926.43	161.0
177.	5000	swap	2000	0.05	15959.17	860.1
178.	10000	swap	3000	0.05	16021.66	1456.0
179.	1000	swap	1000	0.025	16101.54	154.7
180.	10000	swap	3000	0.1	16124.14	1446.2
181.	15000	swap	1500	0.1	16134.14	2172.1
182.	5000	swap	2000	0.1	16157.43	778.7
183.	1000	swap	2000	0.025	16158.77	197.8
184.	1000	swap	1500	0.05	16196.06	147.3

89.	10000	insert	3000	0.05	12259.57	1438.6
90.	15000	insert	1000	0.05	12263.26	2132.5
91.	15000	insert	3000	0.1	12264.31	2138.8
92.	5000	insert	3000	0.1	12266.54	727.8
93.	15000	insert	1500	0.05	12276.29	2155.4
94.	5000	insert	2000	0.1	12286.91	777.8
95.	10000	insert	1500	0.1	12293.11	1436.5
96.	15000	insert	1000	0.1	12310.60	2189.3

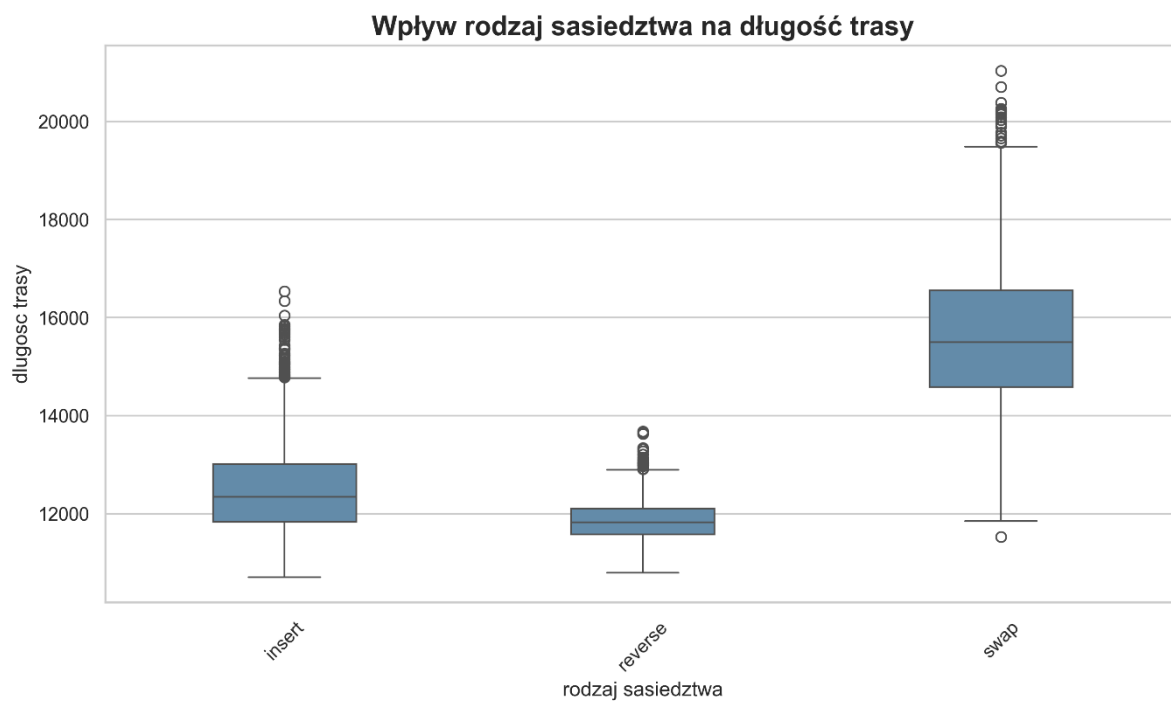
185.	1000	swap	1500	0.1	16279.09	149.8
186.	1000	swap	3000	0.05	16324.77	180.6
187.	1000	swap	2000	0.1	16371.49	184.2
188.	1000	swap	3000	0.1	16395.71	201.8
189.	1000	swap	3000	0.025	16402.20	196.3
190.	1000	swap	1000	0.1	16484.49	155.4
191.	1000	swap	2000	0.05	16545.06	194.2
192.	1000	swap	1500	0.025	16882.66	163.5

Średnia długość trasy (ogólnie): 13325.87

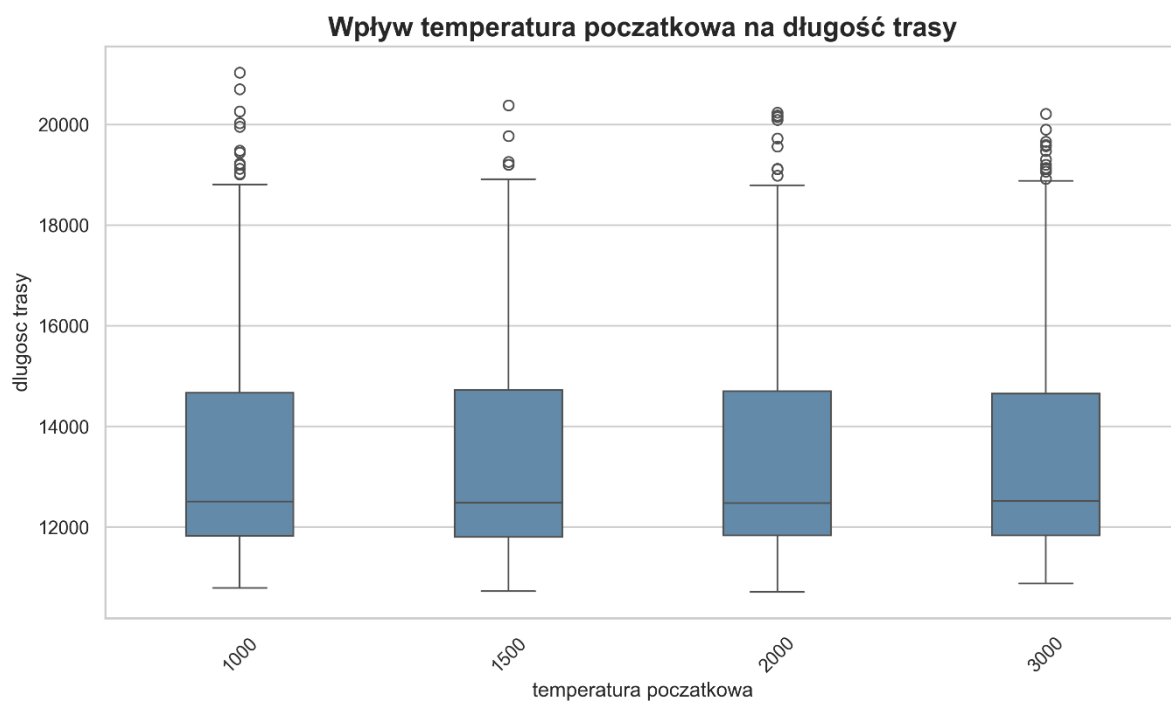
Średni wynik uzyskany za pomocą algorytmu symulowanego wyżarzania jest gorszy niż wynik osiągnięty przy użyciu solvera. Analizując średnie wyniki dla każdej z kombinacji parametrów, można zauważyć, że najlepszą metodą doboru sąsiedztwa okazała się metoda reverse, która dominowała w najlepszych średnich wynikach. Z kolei metoda swap okazała się najgorsza, gdyż jej wyniki były zdecydowanie gorsze. W przypadku pozostałych parametrów nie zaobserwowano tak wyraźnej zależności, co sugeruje, że ich wpływ na ostateczny wynik nie był tak jednoznaczny.



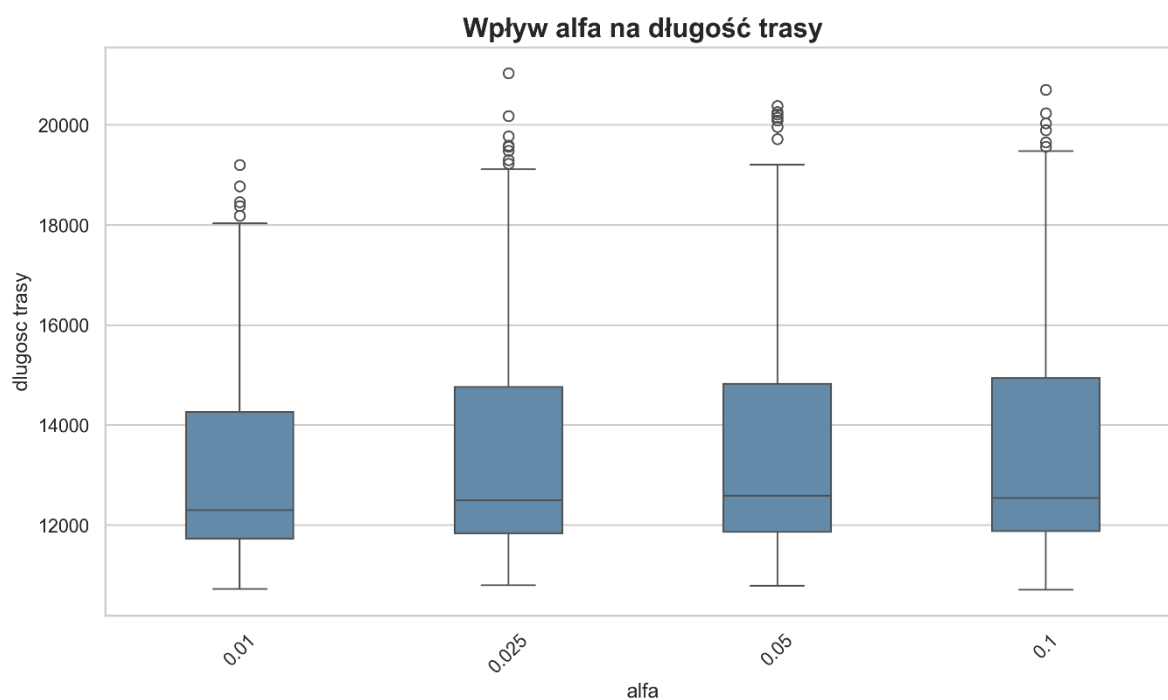
Iteracje	Średnia Najkrótsza Trasa	Min	Max
1000	13846.07	11090	21034
5000	13158.03	10707	20704
10000	13170.33	10725	20385
15000	13129.03	10742	20260



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	15603.97	11527	21034
insert	12514.35	10707	16535
reverse	11859.29	10794	13682



Temperatura początkowa	Średnia Najkrótsza Trasa	Min	Max
1000	13311.65	10794	21034
1500	13326.78	10725	20385
2000	13320.59	10707	20239
3000	13344.46	10875	20214



$\alpha$	Średnia Najkrótsza Trasa	Min	Max
0.01	13061.86	10725	19203
0.025	13361.96	10794	21034
0.05	13424.91	10789	20385
0.1	13454.74	10707	20704

## Liczba miast 76

### Wartości parametrów:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$
1000	insert	4000	0.01
5000	reverse	6000	0.025
10000	swap	8000	0.05
15000		12000	0.1

10 najlepszych wyników:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	alfa	wynik
10000	insert	8000	0.025	111125.437
10000	reverse	12000	0.05	111702.3847
15000	reverse	8000	0.05	111923.0241
15000	reverse	4000	0.05	111952.9054
5000	reverse	12000	0.01	112329.6422
15000	reverse	4000	0.05	113167.2157
15000	reverse	12000	0.05	113463.4305
15000	reverse	4000	0.01	113644.6664
10000	reverse	12000	0.025	113802.4351
5000	reverse	6000	0.01	113938.382

Trasa najlepszego wyniku:

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

Najlepszy wynik uzyskany za pomocą algorytmu symulowanego wyżarzania wyniósł 111125.437, co pozwoliło osiągnąć lepszy rezultat niż ten uzyskany przy użyciu solver. Warto zauważyć, że podobnie jak w przypadku 48 miast, wśród 10 najlepszych wyników żadna z kombinacji parametrów nie uwzględniała metody zmiany sąsiedztwa typu swap. Ponadto, w odniesieniu do pozostałych parametrów, nie zaobserwowano wyraźnych zależności, które wskazywałyby na przewagę konkretnej kombinacji.

10 najgorszych wyników:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$	wynik
1000	swap	8000	0.1	228129.4379
1000	swap	8000	0.1	228290.6438
1000	swap	12000	0.025	228885.5636
1000	swap	8000	0.1	229412.0923

1000	swap	6000	0.025	231567.9629
1000	swap	4000	0.1	233346.8386
1000	swap	6000	0.05	233771.0971
1000	swap	6000	0.1	233874.1758
1000	swap	4000	0.05	235120.3513
1000	swap	8000	0.05	236525.1806

Trasa najgorszego wyniku:

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

Najgorszy wynik uzyskany za pomocą algorytmu symulowanego wyżarzania wyniósł 236525.1806, co stanowi gorszy rezultat niż najgorsze rozwiązanie osiągnięte przy użyciu solvera. Tak jak w przypadku 48 miast, wszystkie 10 najgorszych wyników zostało osiągniętych przy zastosowaniu metody zmiany sąsiedztwa typu swap.

Średnie długości tras dla każdej kombinacji:

	iteracje	rodzaj sąsiedztwa	temp. początkowa	$\alpha$	średni wynik	średni czas (ms)
1.	10000	reverse	6000	0.01	121012.6	2442.3
2.	10000	reverse	6000	0.1	121820.7	2435.5
3.	5000	reverse	8000	0.1	122240.4	1225.1
4.	15000	reverse	6000	0.05	122261.9	3602.4
5.	15000	reverse	6000	0.025	122299.8	3617.4
6.	10000	reverse	12000	0.01	122383.0	2441.6
7.	15000	reverse	8000	0.1	122466.8	3611.7
8.	15000	reverse	4000	0.05	122539.3	3655.7
9.	5000	reverse	12000	0.025	122555.5	1207.0
10.	15000	reverse	12000	0.1	122653.0	3573.5
11.	10000	reverse	8000	0.025	122665.3	2413.2
12.	15000	reverse	8000	0.01	122675.0	3634.6
13.	15000	reverse	4000	0.1	122745.1	3631.5
14.	5000	reverse	12000	0.1	122765.1	1239.1
15.	15000	reverse	12000	0.05	122771.0	3576.9
16.	5000	reverse	6000	0.1	122823.5	1220.3
17.	10000	reverse	8000	0.05	122876.6	2446.9
18.	15000	reverse	4000	0.025	122935.7	3577.0
19.	15000	reverse	8000	0.025	122940.6	3576.3
20.	10000	reverse	12000	0.05	123073.7	2442.7

	iteracje	rodzaj sąsiedztwa	temp. początkowa	$\alpha$	średni wynik	średni czas (ms)
97.	1000	reverse	8000	0.1	142685.0	243.4
98.	1000	reverse	6000	0.01	143066.3	261.8
99.	1000	reverse	4000	0.01	143662.1	251.2
100.	1000	reverse	4000	0.05	143812.1	248.6
101.	1000	reverse	6000	0.05	143930.9	241.6
102.	1000	reverse	4000	0.025	144136.6	250.9
103.	1000	reverse	6000	0.025	144274.0	247.2
104.	1000	reverse	12000	0.025	144355.6	252.5
105.	1000	reverse	6000	0.1	144643.6	246.7
106.	1000	reverse	8000	0.025	144718.0	247.9
107.	1000	reverse	4000	0.1	145152.7	239.1
108.	1000	reverse	8000	0.01	145271.4	260.7
109.	1000	reverse	12000	0.05	145654.8	246.6
110.	1000	reverse	12000	0.01	146153.1	251.6
111.	1000	reverse	8000	0.05	146355.0	243.6
112.	1000	reverse	12000	0.1	146421.2	240.1
113.	15000	swap	12000	0.01	165748.0	3270.5
114.	15000	swap	8000	0.01	167279.4	3245.2
115.	10000	swap	6000	0.05	167430.9	2228.5
116.	10000	swap	8000	0.01	167884.5	2176.9

21.	10000	reverse	12000	0.1	123134.5	2385.9
22.	5000	reverse	4000	0.05	123154.9	1227.1
23.	10000	reverse	6000	0.05	123220.8	2447.2
24.	10000	reverse	4000	0.05	123224.8	2414.2
25.	5000	reverse	12000	0.01	123236.3	1233.3
26.	15000	reverse	12000	0.01	123268.2	3615.6
27.	5000	reverse	12000	0.05	123339.1	1193.6
28.	15000	reverse	8000	0.05	123370.0	3583.1
29.	15000	reverse	6000	0.1	123385.3	3564.8
30.	5000	reverse	6000	0.01	123426.7	1243.2
31.	10000	reverse	12000	0.025	123427.7	2427.1
32.	10000	reverse	4000	0.1	123476.5	2417.9
33.	5000	reverse	4000	0.025	123537.3	1266.4
34.	10000	reverse	8000	0.01	123540.4	2424.7
35.	5000	reverse	8000	0.01	123561.5	1250.2
36.	5000	reverse	4000	0.1	123635.6	1206.9
37.	15000	reverse	6000	0.01	123653.3	3639.6
38.	5000	reverse	8000	0.025	123760.1	1214.0
39.	15000	reverse	4000	0.01	123840.4	3641.7
40.	10000	reverse	8000	0.1	123976.8	2413.7
41.	5000	reverse	8000	0.05	124068.2	1227.1
42.	10000	reverse	4000	0.025	124092.4	2418.4
43.	10000	reverse	6000	0.025	124115.4	2436.7
44.	15000	reverse	12000	0.025	124501.0	3595.4
45.	5000	reverse	6000	0.025	124516.2	1193.7
46.	10000	reverse	4000	0.01	124797.5	2468.6
47.	5000	reverse	6000	0.05	124800.6	1221.5
48.	5000	reverse	4000	0.01	124845.0	1271.9
49.	15000	insert	12000	0.025	128559.8	3237.9
50.	15000	insert	8000	0.05	128643.7	3244.5
51.	15000	insert	6000	0.01	129013.0	3277.8
52.	15000	insert	8000	0.025	129210.5	3211.3
53.	15000	insert	6000	0.05	129379.3	3215.2
54.	10000	insert	12000	0.01	129548.0	2191.1
55.	15000	insert	4000	0.05	129651.1	3227.1
56.	15000	insert	4000	0.1	129765.8	3240.8
57.	10000	insert	6000	0.01	129927.4	2221.2
58.	10000	insert	6000	0.05	129934.2	2177.9
59.	10000	insert	8000	0.01	129992.0	2171.9
60.	15000	insert	8000	0.01	130096.6	3228.9
61.	15000	insert	8000	0.1	130115.3	3239.2
62.	10000	insert	4000	0.1	130148.7	2178.2
63.	15000	insert	4000	0.01	130148.9	3224.8
64.	10000	insert	12000	0.025	130272.0	2145.5
65.	15000	insert	6000	0.025	130281.0	3237.7
66.	15000	insert	12000	0.05	130417.5	3235.3

117.	5000	swap	8000	0.01	168276.9	1113.4
118.	10000	swap	4000	0.01	168279.8	2195.3
119.	15000	swap	4000	0.01	168290.7	3193.8
120.	10000	swap	12000	0.01	168451.1	2207.2
121.	15000	swap	6000	0.1	169172.3	3145.3
122.	5000	swap	4000	0.01	169403.3	1069.2
123.	15000	swap	4000	0.1	169462.7	3250.4
124.	5000	swap	6000	0.01	169524.1	1081.9
125.	5000	swap	12000	0.01	169558.0	1082.6
126.	15000	swap	12000	0.1	169585.9	3217.4
127.	5000	swap	12000	0.05	169607.1	1064.8
128.	15000	swap	6000	0.01	169765.1	3166.4
129.	10000	swap	6000	0.025	170049.4	2223.1
130.	15000	swap	6000	0.025	170229.2	3120.3
131.	5000	swap	6000	0.1	170469.0	1067.2
132.	1000	insert	4000	0.1	170503.7	217.9
133.	10000	swap	12000	0.025	170601.5	2211.6
134.	10000	swap	4000	0.1	170787.3	2178.0
135.	5000	swap	12000	0.025	170847.7	1052.5
136.	10000	swap	4000	0.025	171072.1	2224.1
137.	10000	swap	6000	0.01	171182.8	2209.6
138.	5000	swap	8000	0.1	171197.1	1080.8
139.	1000	insert	6000	0.1	171237.5	223.2
140.	15000	swap	12000	0.05	171420.6	3268.0
141.	5000	swap	8000	0.05	171599.4	1052.4
142.	1000	insert	4000	0.025	171607.4	220.6
143.	1000	insert	12000	0.01	171815.9	226.3
144.	15000	swap	4000	0.05	171857.6	3269.7
145.	5000	swap	8000	0.025	171995.1	1055.8
146.	15000	swap	6000	0.05	172039.4	3142.2
147.	1000	insert	8000	0.01	172138.7	224.0
148.	1000	insert	12000	0.05	172224.9	208.7
149.	10000	swap	8000	0.05	172240.5	2166.2
150.	15000	swap	8000	0.1	172455.9	3249.2
151.	10000	swap	6000	0.1	172539.2	2207.6
152.	15000	swap	8000	0.05	172587.8	3282.6
153.	15000	swap	4000	0.025	172646.7	3269.5
154.	15000	swap	12000	0.025	172848.0	3281.2
155.	5000	swap	6000	0.025	172898.3	1060.4
156.	10000	swap	8000	0.025	173041.1	2207.3
157.	10000	swap	12000	0.05	173083.7	2189.8
158.	10000	swap	12000	0.1	173175.5	2202.0
159.	15000	swap	8000	0.025	173268.5	3258.1
160.	10000	swap	4000	0.05	173291.1	2209.8
161.	1000	insert	4000	0.01	173415.6	216.7
162.	1000	insert	6000	0.025	173425.0	214.7

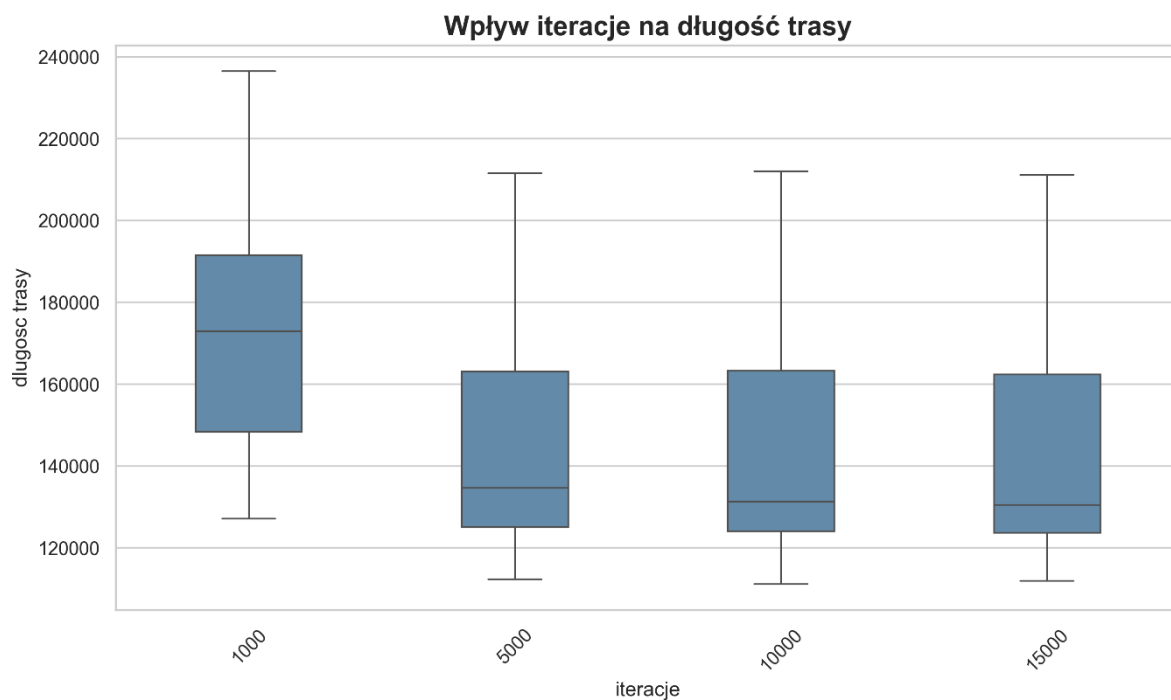


67.	10000	insert	8000	0.05	130825.2	2150.4
68.	15000	insert	6000	0.1	130831.8	3228.6
69.	15000	insert	12000	0.01	130925.6	3290.4
70.	10000	insert	4000	0.025	130936.9	2153.7
71.	10000	insert	4000	0.01	130975.2	2165.4
72.	10000	insert	12000	0.05	131035.1	2154.1
73.	10000	insert	8000	0.1	131132.0	2170.0
74.	15000	insert	4000	0.025	131308.0	3252.9
75.	15000	insert	12000	0.1	131515.1	3250.2
76.	10000	insert	8000	0.025	131737.5	2164.8
77.	10000	insert	6000	0.025	131799.2	2219.7
78.	10000	insert	4000	0.05	132019.4	2175.5
79.	10000	insert	12000	0.1	132037.6	2150.8
80.	5000	insert	12000	0.1	132820.3	1085.6
81.	5000	insert	8000	0.1	132824.2	1062.7
82.	5000	insert	6000	0.1	132975.2	1037.8
83.	5000	insert	12000	0.025	133386.2	1206.2
84.	5000	insert	8000	0.01	133623.6	1085.7
85.	5000	insert	4000	0.01	133962.6	1071.1
86.	5000	insert	8000	0.05	134149.3	1063.3
87.	10000	insert	6000	0.1	134180.7	2207.3
88.	5000	insert	6000	0.025	134795.9	1080.3
89.	5000	insert	6000	0.05	134871.9	1047.2
90.	5000	insert	4000	0.05	135213.8	1060.7
91.	5000	insert	8000	0.025	135799.0	1083.0
92.	5000	insert	4000	0.1	135807.7	1094.0
93.	5000	insert	6000	0.01	135927.4	1077.1
94.	5000	insert	4000	0.025	136288.9	1077.1
95.	5000	insert	12000	0.01	136329.3	1263.6
96.	5000	insert	12000	0.05	136899.0	1065.3

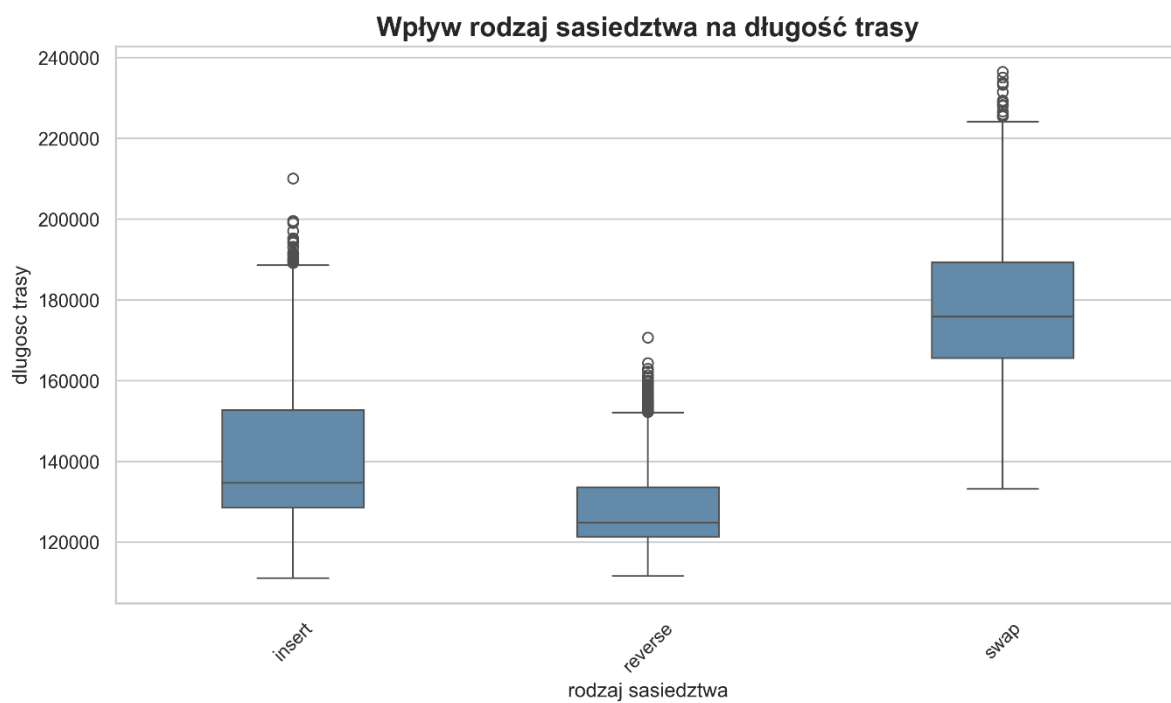
163.	5000	swap	4000	0.1	173434.6	1061.6
164.	1000	insert	8000	0.025	173550.1	223.0
165.	1000	insert	6000	0.01	173753.5	216.7
166.	5000	swap	12000	0.1	173899.3	1068.9
167.	1000	insert	4000	0.05	174005.2	216.7
168.	1000	insert	12000	0.025	174223.9	214.9
169.	1000	insert	12000	0.1	174349.9	218.2
170.	5000	swap	6000	0.05	174591.3	1077.2
171.	1000	insert	6000	0.05	174766.8	222.2
172.	1000	insert	8000	0.05	174863.1	218.3
173.	5000	swap	4000	0.05	175424.2	1042.9
174.	5000	swap	4000	0.025	175745.1	1050.0
175.	1000	insert	8000	0.1	176403.9	214.4
176.	10000	swap	8000	0.1	176828.8	2168.6
177.	1000	swap	12000	0.01	190531.2	228.1
178.	1000	swap	4000	0.01	195802.8	243.9
179.	1000	swap	8000	0.01	196186.4	222.9
180.	1000	swap	4000	0.05	196814.0	230.2
181.	1000	swap	12000	0.05	197260.6	225.6
182.	1000	swap	8000	0.025	197979.8	216.4
183.	1000	swap	12000	0.025	198199.6	236.8
184.	1000	swap	6000	0.1	198870.2	257.0
185.	1000	swap	6000	0.025	199674.4	222.6
186.	1000	swap	6000	0.01	200544.8	231.0
187.	1000	swap	6000	0.05	200648.4	222.6
188.	1000	swap	4000	0.025	200970.0	224.8
189.	1000	swap	12000	0.1	201462.4	229.6
190.	1000	swap	8000	0.05	201470.8	219.4
191.	1000	swap	4000	0.1	201516.1	234.3
192.	1000	swap	8000	0.1	202880.4	216.3

Średnia długość trasy (ogólnie): 149645.3

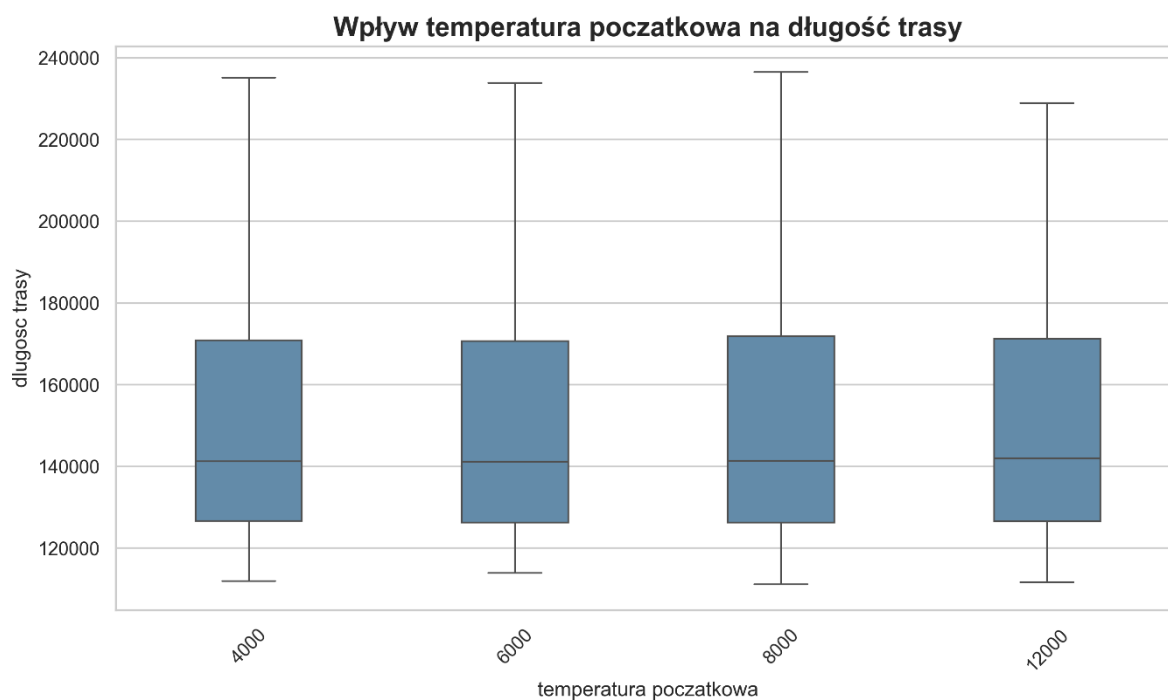
Średni wynik uzyskany za pomocą algorytmu symulowanego wyżarzania jest gorszy niż wynik osiągnięty przy użyciu solvera. Analizując wyniki średnie dla różnych kombinacji parametrów, można zauważyć, że najlepszą metodą doboru sąsiedztwa okazała się metoda reverse, która przeważała w najlepszych średnich wynikach. Podobnie jak w przypadku 48 miast, metoda swap okazała się najgorsza, ponieważ jej wyniki były zdecydowanie mniej korzystne. Kombinacje z najmniejszą liczbą iteracji (1000) wyznaczały trasy dłuższe niż kombinacje z większą liczbą iteracji. W przypadku pozostałych parametrów nie zaobserwowano tak wyraźnych zależności, co sugeruje, że ich wpływ na końcowy wynik nie był jednoznaczny.



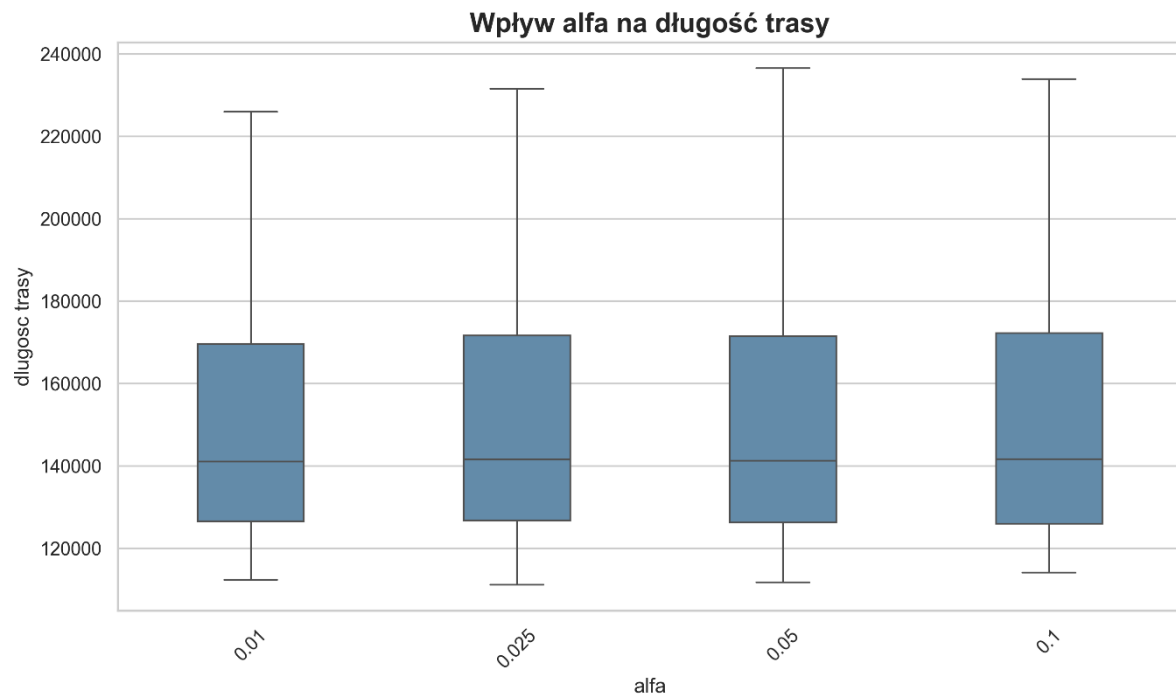
Iteracje	Średnia Najkrótsza Trasa	Min	Max
1000	172237.29	127126.19	236525.18
5000	143341.89	112329.64	211555.54
10000	141818.31	111125.44	212002.84
15000	141183.90	111923.02	211132.81



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	178091.87	133249.17	236525.18
insert	142255.05	111125.44	210135.50
reverse	128589.12	111702.38	170715.31



Temperatura początkowa	Średnia Najkrótsza Trasa	Min	Max
4000	149794.69	111952.91	235120.35
6000	149582.93	113938.38	233874.18
8000	149780.17	111125.44	236525.18
12000	149423.61	111702.38	228885.56



$\alpha$	Średnia Najkrótsza Trasa	Min	Max
0.01	148681.15	112329.64	225971.10
0.025	149959.98	111125.44	231567.96
0.05	149973.35	111702.38	236525.18
0.1	149966.91	114133.96	233874.18

**Liczba miast 127**

**Wartości parametrów:**

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$
1000	insert	5000	0.01
5000	reverse	7500	0.025
10000	swap	10000	0.05
15000		15000	0.1

10 najlepszych wyników:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	$\alpha$	wynik
15000	reverse	15000	0.01	124721.8911
10000	reverse	5000	0.025	124909.8929
15000	reverse	10000	0.05	125196.519
15000	reverse	10000	0.025	125382.512
10000	reverse	5000	0.01	126299.3337
15000	reverse	10000	0.025	126410.5109
10000	reverse	10000	0.1	126560.426
10000	reverse	15000	0.1	126587.8196
15000	reverse	15000	0.01	126621.2987
15000	reverse	5000	0.01	126830.1622

Trasa najlepszego wyniku:

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

Najlepszy wynik uzyskany przy użyciu algorytmu symulowanego wyżarzania wyniósł 124721.8911, co pozwoliło uzyskać lepszy rezultat niż ten osiągnięty za pomocą solvera. Wszystkie najlepsze trasy zostały uzyskane przy zastosowaniu metody reverse. Ponadto, w przypadku pozostałych parametrów, nie zaobserwowano wyraźnych zależności wskazujących na przewagę którejkolwiek kombinacji.

10 najgorszych wyników:

iteracje	rodzaj sąsiedztwa	temperatura początkowa	alfa	wynik
1000	swap	10000	0.05	275435.7296
1000	swap	10000	0.05	275558.151
1000	swap	15000	0.05	276318.6497

1000	swap	10000	0.1	276514.015
1000	swap	5000	0.05	276560.4149
1000	swap	7500	0.025	276647.3324
1000	swap	7500	0.025	279796.1585
1000	swap	7500	0.1	280596.9006
1000	swap	10000	0.05	281751.2562
1000	swap	10000	0.05	286797.4408

Trasa najgorszego wyniku:

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

Podobnie jak w poprzednich zbiorach danych, najgorszy wynik uzyskany za pomocą algorytmu symulowanego wyżarzania jest gorszy niż wynik osiągnięty przy użyciu solvera. Dodatkowo, podobnie jak wcześniej, dominującą metodą zmiany sąsiedztwa okazała się metoda swap.

Średnie długości tras dla każdej kombinacji:

	iteracje	rodzaj sąsiedztwa	temp. początkowa	$\alpha$	średni wynik	średni czas (ms)
1.	15000	reverse	5000	0.1	132823.0	5703.4
2.	15000	reverse	10000	0.01	133118.6	5708.5
3.	15000	reverse	5000	0.05	133628.8	5720.9
4.	15000	reverse	10000	0.025	133892.9	5723.8
5.	10000	reverse	10000	0.01	133933.8	3888.1
6.	15000	reverse	15000	0.05	133980.7	5716.7
7.	15000	reverse	7500	0.05	134072.9	6260.1
8.	15000	reverse	7500	0.025	134079.4	5829.9
9.	15000	reverse	5000	0.025	134113.9	5867.4
10.	10000	reverse	7500	0.1	134150.0	3818.4
11.	15000	reverse	15000	0.025	134184.2	5710.2
12.	10000	reverse	5000	0.01	134302.5	3900.5
13.	10000	reverse	7500	0.025	134322.0	3883.3
14.	10000	reverse	10000	0.025	134381.8	3862.6
15.	15000	reverse	7500	0.01	134576.7	5847.1
16.	15000	reverse	5000	0.01	134587.8	6365.4
17.	15000	reverse	7500	0.1	134788.6	6373.2
97.	15000	swap	15000	0.025	182900.3	5387.6
98.	15000	swap	10000	0.1	184282.3	5221.9
99.	15000	swap	15000	0.01	184882.1	5523.9
100.	15000	swap	7500	0.01	185823.0	5218.2
101.	15000	swap	7500	0.1	186163.4	5217.7
102.	15000	swap	7500	0.05	187016.1	5216.4
103.	15000	swap	15000	0.1	187078.8	5249.3
104.	15000	swap	5000	0.05	187095.1	5238.6
105.	15000	swap	10000	0.01	187505.7	5271.4
106.	15000	swap	5000	0.1	187555.4	5209.4
107.	15000	swap	10000	0.025	187598.1	5427.5
108.	15000	swap	5000	0.01	187759.6	5215.0
109.	10000	swap	15000	0.025	188146.4	3498.5
110.	15000	swap	7500	0.025	188402.2	5226.9
111.	15000	swap	15000	0.05	188451.0	5321.1
112.	15000	swap	10000	0.05	188672.1	5198.1
113.	10000	swap	5000	0.1	188851.8	3531.3

18.	15000	reverse	15000	0.1	134795.3	5672.9
19.	15000	reverse	10000	0.05	134870.0	5685.2
20.	10000	reverse	7500	0.05	134912.9	3856.3
21.	10000	reverse	10000	0.05	134917.5	3878.4
22.	10000	reverse	15000	0.05	134936.0	3866.9
23.	10000	reverse	15000	0.1	134950.7	3849.4
24.	15000	reverse	10000	0.1	135043.3	5831.9
25.	15000	reverse	15000	0.01	135331.5	5688.1
26.	10000	reverse	7500	0.01	135366.3	3883.5
27.	10000	reverse	5000	0.1	135486.9	3868.5
28.	10000	reverse	5000	0.025	135678.2	3854.4
29.	10000	reverse	10000	0.1	135864.2	3909.0
30.	10000	reverse	15000	0.01	135882.3	3817.2
31.	10000	reverse	5000	0.05	136016.1	3857.3
32.	10000	reverse	15000	0.025	136103.5	3857.4
33.	5000	reverse	10000	0.01	139640.7	1957.2
34.	5000	reverse	15000	0.01	140096.9	1972.3
35.	5000	reverse	7500	0.1	140201.4	1869.6
36.	5000	reverse	7500	0.01	140277.3	1916.5
37.	5000	reverse	10000	0.05	140478.6	1938.5
38.	5000	reverse	5000	0.1	140599.3	1887.6
39.	5000	reverse	10000	0.1	140840.7	1950.5
40.	5000	reverse	5000	0.025	140896.8	1881.9
41.	5000	reverse	7500	0.05	140913.2	1858.4
42.	5000	reverse	15000	0.05	141053.7	1946.5
43.	5000	reverse	15000	0.1	141091.8	1926.0
44.	5000	reverse	7500	0.025	141098.6	1875.9
45.	5000	reverse	5000	0.05	141237.2	1856.9
46.	5000	reverse	10000	0.025	141359.6	1956.6
47.	5000	reverse	15000	0.025	141382.4	1951.3
48.	5000	reverse	5000	0.01	141399.7	1904.9
49.	15000	insert	10000	0.025	148582.9	12054.1
50.	15000	insert	15000	0.01	150353.0	5144.7
51.	15000	insert	5000	0.05	150663.2	5211.0
52.	15000	insert	7500	0.1	150911.8	6295.3
53.	15000	insert	10000	0.01	151221.4	7278.9
54.	15000	insert	15000	0.1	151261.0	5090.0
55.	15000	insert	5000	0.1	151385.3	5247.4
56.	15000	insert	15000	0.05	151557.3	5097.1
57.	15000	insert	10000	0.05	151581.1	9704.9
58.	15000	insert	5000	0.01	151802.1	5226.4
59.	15000	insert	7500	0.01	151860.9	8337.6
60.	15000	insert	5000	0.025	152165.8	5203.1
61.	15000	insert	15000	0.025	152252.7	5093.8
62.	15000	insert	7500	0.025	152301.7	6320.9
63.	15000	insert	7500	0.05	152664.3	6676.0

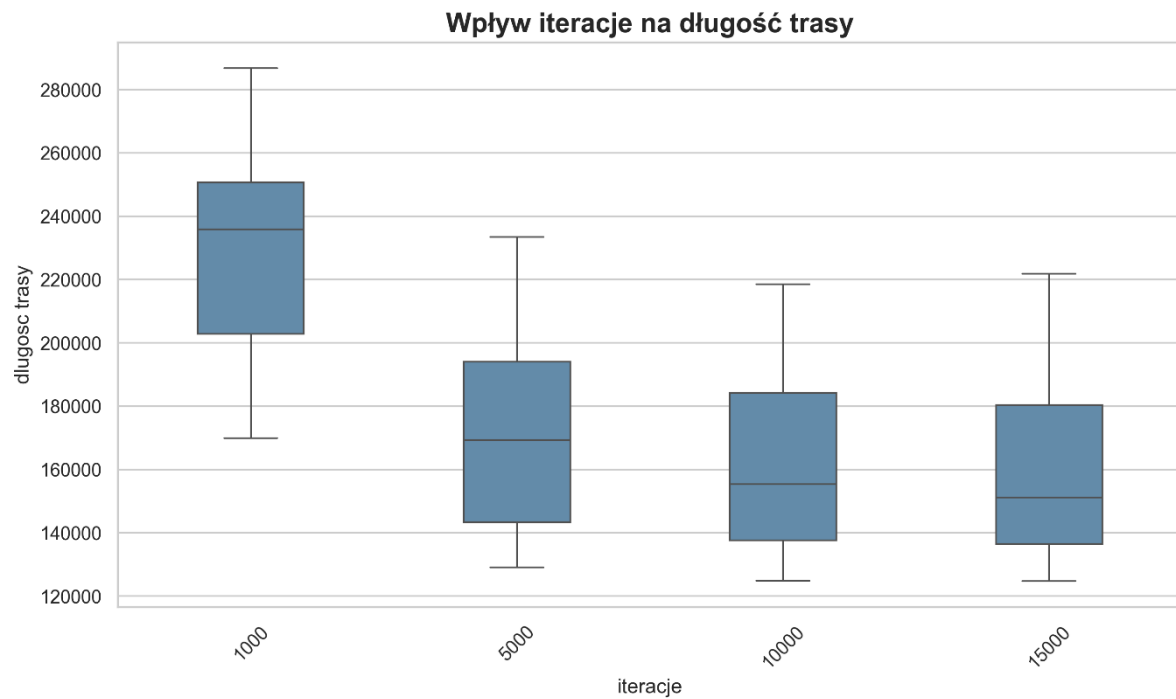
114.	10000	swap	7500	0.025	189079.0	3502.0
115.	10000	swap	10000	0.05	189323.7	3505.8
116.	10000	swap	10000	0.01	189396.2	3525.8
117.	10000	swap	7500	0.01	189655.0	3530.5
118.	10000	swap	5000	0.01	189944.0	3476.4
119.	10000	swap	5000	0.025	190139.9	3475.2
120.	10000	swap	15000	0.01	190601.4	3497.8
121.	10000	swap	15000	0.1	190617.2	3467.6
122.	10000	swap	7500	0.05	190916.7	3499.5
123.	10000	swap	5000	0.05	191216.3	3496.0
124.	15000	swap	5000	0.025	191410.4	5209.1
125.	10000	swap	7500	0.1	191616.9	3463.9
126.	10000	swap	10000	0.025	192480.4	3531.6
127.	10000	swap	15000	0.05	193376.2	3475.8
128.	10000	swap	10000	0.1	193770.2	3515.3
129.	1000	reverse	15000	0.05	195867.2	414.4
130.	1000	reverse	7500	0.025	196154.8	402.1
131.	5000	swap	10000	0.01	196838.2	1777.8
132.	1000	reverse	7500	0.05	196864.6	401.1
133.	1000	reverse	7500	0.01	197111.6	412.6
134.	1000	reverse	10000	0.1	197136.3	403.7
135.	1000	reverse	10000	0.025	197264.7	415.2
136.	1000	reverse	15000	0.1	197298.1	412.7
137.	1000	reverse	15000	0.025	197305.6	399.4
138.	1000	reverse	5000	0.1	197803.7	397.8
139.	1000	reverse	5000	0.025	197913.2	404.6
140.	1000	reverse	10000	0.01	197914.4	403.9
141.	5000	swap	7500	0.025	197998.3	1762.4
142.	1000	reverse	5000	0.01	198109.4	405.7
143.	1000	reverse	10000	0.05	198734.5	399.8
144.	1000	reverse	7500	0.1	198807.9	419.4
145.	5000	swap	7500	0.01	199077.4	1768.3
146.	5000	swap	5000	0.1	199087.1	1740.8
147.	5000	swap	15000	0.1	199203.7	1769.6
148.	1000	reverse	15000	0.01	199357.9	405.1
149.	5000	swap	7500	0.05	199453.4	1760.2
150.	1000	reverse	5000	0.05	199718.7	407.0
151.	5000	swap	5000	0.01	199844.3	1788.2
152.	5000	swap	15000	0.05	200110.7	1758.3
153.	5000	swap	5000	0.025	200438.2	1792.3
154.	5000	swap	10000	0.025	200695.5	1745.6
155.	5000	swap	15000	0.01	200785.6	1767.6
156.	5000	swap	7500	0.1	201052.1	1724.4
157.	5000	swap	10000	0.1	201158.8	1765.5
158.	5000	swap	10000	0.05	201976.5	1746.3
159.	5000	swap	5000	0.05	203488.6	1758.1

64.	15000	insert	10000	0.1	152767.4	10720.3	160.	5000	swap	15000	0.025	203751.3	1768.2
65.	10000	insert	15000	0.01	153453.3	3490.3	161.	1000	insert	5000	0.01	233136.2	360.5
66.	10000	insert	15000	0.05	154975.7	3464.7	162.	1000	insert	15000	0.05	233714.7	361.3
67.	10000	insert	15000	0.025	155075.5	3483.8	163.	1000	insert	10000	0.1	235299.7	350.9
68.	10000	insert	5000	0.025	155079.0	3462.4	164.	1000	insert	7500	0.025	235611.4	368.5
69.	10000	insert	5000	0.1	155289.7	3466.5	165.	1000	insert	15000	0.025	235709.3	360.9
70.	10000	insert	15000	0.1	155555.2	3462.8	166.	1000	insert	15000	0.01	235877.1	364.5
71.	10000	insert	7500	0.025	155626.0	3466.5	167.	1000	insert	10000	0.05	236543.6	359.6
72.	10000	insert	7500	0.01	155758.9	3467.6	168.	1000	insert	10000	0.01	236595.3	361.6
73.	10000	insert	5000	0.01	156028.2	3468.8	169.	1000	insert	5000	0.1	236777.5	351.7
74.	10000	insert	7500	0.1	156127.5	3430.3	170.	1000	insert	7500	0.1	236792.9	354.4
75.	10000	insert	10000	0.01	156486.6	3458.8	171.	1000	insert	7500	0.05	237153.0	357.2
76.	10000	insert	10000	0.1	156521.6	3475.4	172.	1000	insert	5000	0.025	237267.3	356.5
77.	10000	insert	10000	0.025	156726.7	3438.3	173.	1000	insert	10000	0.025	237282.4	354.3
78.	10000	insert	5000	0.05	156995.1	3500.1	174.	1000	insert	7500	0.01	237559.9	359.7
79.	10000	insert	7500	0.05	157323.2	3448.7	175.	1000	insert	15000	0.1	239135.0	350.3
80.	10000	insert	10000	0.05	157805.6	3501.3	176.	1000	insert	5000	0.05	241715.3	364.4
81.	5000	insert	10000	0.05	166434.6	1763.3	177.	1000	swap	7500	0.05	248861.0	361.6
82.	5000	insert	10000	0.01	167285.1	1769.5	178.	1000	swap	5000	0.1	253447.7	366.6
83.	5000	insert	15000	0.025	167797.4	1670.2	179.	1000	swap	5000	0.01	253574.9	369.0
84.	5000	insert	5000	0.1	169177.9	1739.0	180.	1000	swap	5000	0.025	254052.6	398.9
85.	5000	insert	5000	0.01	169289.8	1740.8	181.	1000	swap	10000	0.025	254186.8	366.1
86.	5000	insert	7500	0.01	169354.3	1717.0	182.	1000	swap	15000	0.025	254360.2	358.9
87.	5000	insert	7500	0.1	169404.9	1742.6	183.	1000	swap	7500	0.01	254364.2	359.6
88.	5000	insert	15000	0.01	170008.7	1709.5	184.	1000	swap	15000	0.05	254374.7	374.7
89.	5000	insert	5000	0.05	170047.8	1748.5	185.	1000	swap	15000	0.01	254443.2	373.7
90.	5000	insert	15000	0.05	170297.1	1740.8	186.	1000	swap	15000	0.1	255131.9	359.7
91.	5000	insert	15000	0.1	170427.1	1718.4	187.	1000	swap	5000	0.05	255709.1	359.1
92.	5000	insert	10000	0.025	170475.0	1702.5	188.	1000	swap	7500	0.025	255720.4	362.9
93.	5000	insert	7500	0.025	170618.5	1752.0	189.	1000	swap	7500	0.1	256178.8	370.7
94.	5000	insert	10000	0.1	171048.4	1708.9	190.	1000	swap	10000	0.01	256252.8	370.6
95.	5000	insert	5000	0.025	171442.6	1724.2	191.	1000	swap	10000	0.1	256878.1	366.4
96.	5000	insert	7500	0.05	171560.6	1746.5	192.	1000	swap	10000	0.05	259061.4	373.3

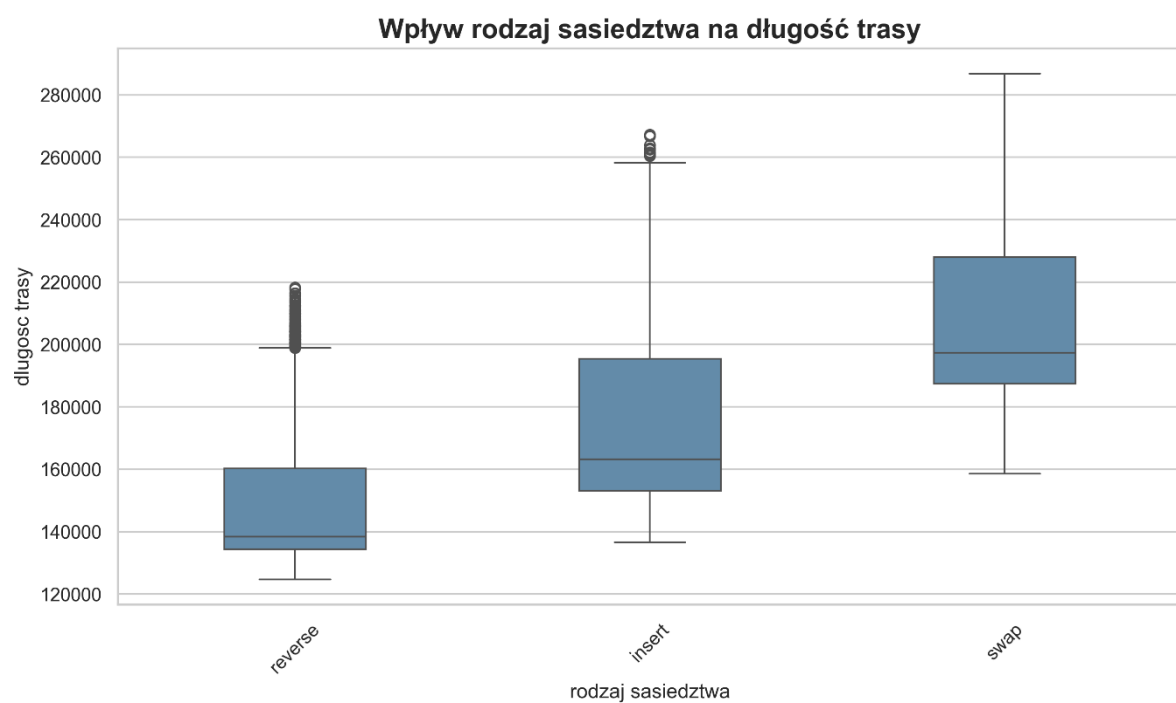
Średnia długość trasy (ogólnie): 179517.2

Dla zbioru danych zawierającego 127 punktów średnia wartość uzyskana przy użyciu algorytmu symulowanego wyżarzania jest gorsza niż wynik osiągnięty za pomocą solvera. Podobnie jak w przypadku 48 i 76 miast, najlepsze rezultaty osiągnięto przy użyciu metody reverse, natomiast najgorsze wyniki uzyskano przy zastosowaniu metody swap. Trasy z 1000 iteracji charakteryzują się większym dystansem w porównaniu do tras z większą liczbą iteracji. Pozostałe parametry nie wykazują istotnego wpływu na długość trasy.

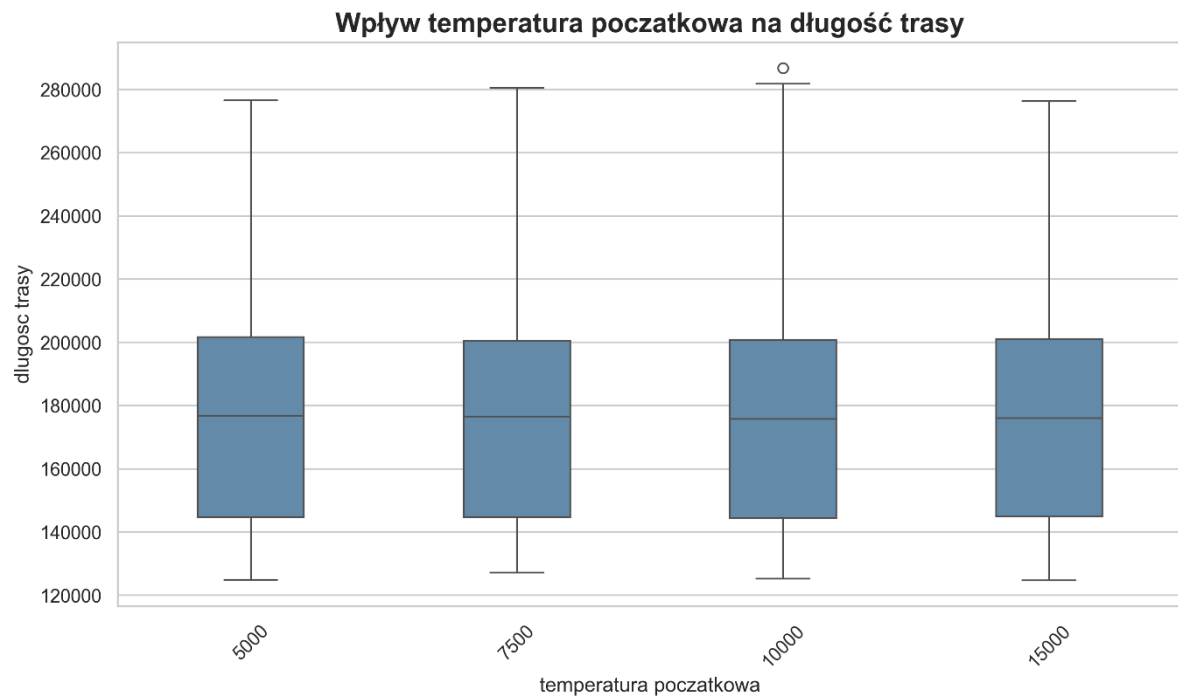




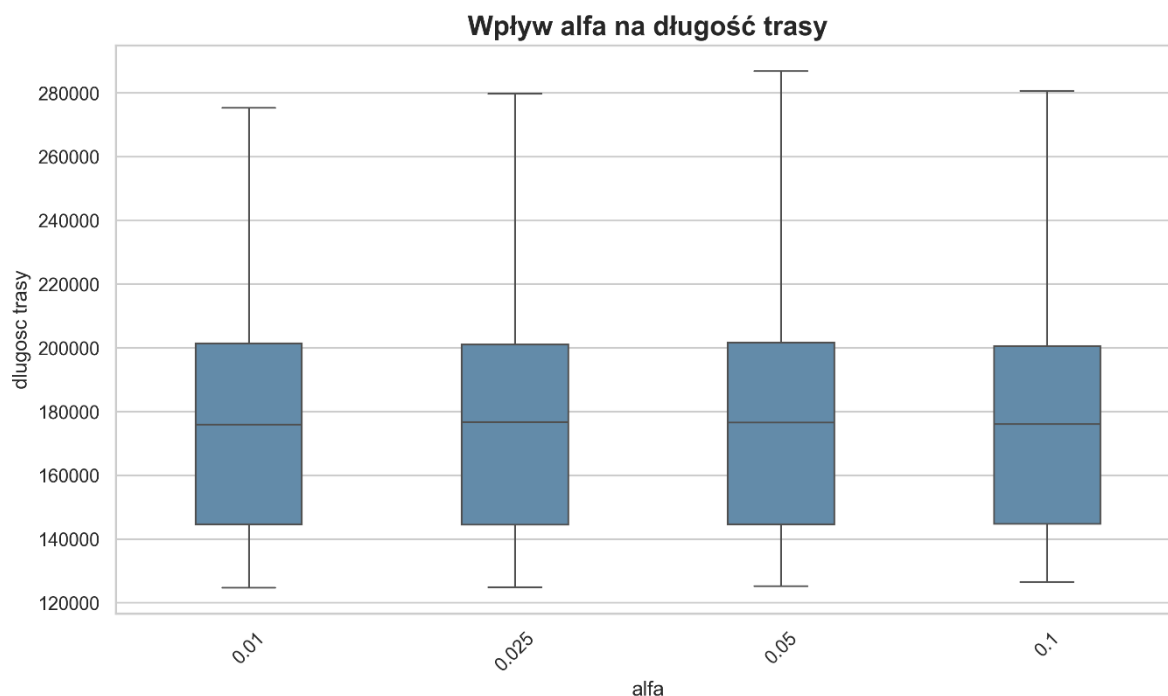
Iteracje	Średnia Najkrótsza Trasa	Min	Max
1000	229711.06	169937.05	286797.44
5000	170254.11	129107.21	233439.75
10000	160524.25	124909.89	218547.37
15000	157579.48	124721.89	221850.90



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	208176.32	158513.75	286797.44
insert	178421.88	136555.87	267345.73
reverse	151953.48	124721.89	218283.88



Temperatura początkowa	Średnia Najkrótsza Trasa	Min	Max
5000	179712.35	124909.89	276560.41
7500	179368.87	127122.94	280596.90
10000	179627.62	125196.52	286797.44
15000	179360.06	124721.89	276318.65



$\alpha$	Średnia Najkrótsza Trasa	Min	Max
0.01	179121.38	124721.89	275274.32
0.025	179489.70	124909.89	279796.16
0.05	179840.36	125196.52	286797.44
0.1	179617.46	126560.43	280596.90

## 5. Algorytm wspinaczki z multistartem

Algorytm **wspinaczki z multistartem** (IHC) to metoda optymalizacji, która polega na wielokrotnym uruchamianiu algorytmu wspinaczki górskiej z losowo wybranych punktów startowych w przestrzeni rozwiązań. Dzięki temu unika on pułapek w postaci lokalnych minimów, co pozwala na większą szansę znalezienia rozwiązania globalnego. Na początku generowane są losowe początkowe rozwiązania (w tych przypadkach 50), które określają kolejność odwiedzania miast. Dla każdego rozwiązania obliczana jest długość trasy, a następnie algorytm próbuje znaleźć lepsze rozwiązanie, zmieniając kolejność odwiedzin. Jeśli uda się znaleźć lepsze rozwiązanie, proces jest powtarzany, a gdy nie nastąpi poprawa, rozwiązanie uznawane jest za lokalnie optymalne.

Z przyjętych wartości parametrów utworzono 30 różnych kombinacji, ponieważ dodano warunek: `liczba_iteracji_bez_poprawy < liczba_iteracji`. Każda kombinacja została powtórzona 10 razy.

### Wartości parametrów:

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy
100	insert	50
500	reverse	100
1000	swap	500
5000		1000

### Liczba miast 48

10 najlepszych wyników:

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	wynik
1000	reverse	100	10628
5000	insert	100	10628
500	reverse	100	10638
500	reverse	100	10648
1000	reverse	500	10648
5000	reverse	100	10653
1000	reverse	50	10653
1000	reverse	500	10653
100	reverse	50	10663
1000	reverse	500	10663

Trasy najlepszych wyników:

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

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

Najlepszy wynik uzyskany w wyniku przeprowadzenia algorytmu wspinaczki z multistartem to trasa o długości 10 628, co oznacza, że jest ona lepsza od wyniku uzyskanego za pomocą solvera. W trakcie obliczeń znaleziono dwie różne trasy o tej samej długości, które zostały utworzone w dwóch różnych kombinacjach. Warto zauważyć, że żadna z 10 najlepszych kombinacji nie zawiera zmiany sąsiedztwa typu swap.

#### 10 najgorszych tras

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	wynik
5000	swap	500	13438
500	swap	50	13443
5000	swap	100	13453
5000	swap	100	13466
5000	swap	100	13467
500	swap	100	13501
500	swap	50	13570
500	swap	50	13591
5000	swap	1000	13674
500	swap	50	14067

Trasa najgorszego wyniku:

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

Wszystkie 10 najgorszych tras wykorzystują zmianę sąsiedztwa typu swap. W pozostałych parametrach nie zaobserwowano wyraźnych zależności.

Średnie długości tras dla każdej kombinacji:

	iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	średni wynik	średni czas (ms)
1.	1000	reverse	500	10734.0	45361.2
2.	5000	reverse	100	10746.8	17004.1
3.	500	reverse	100	10751.3	17992.2

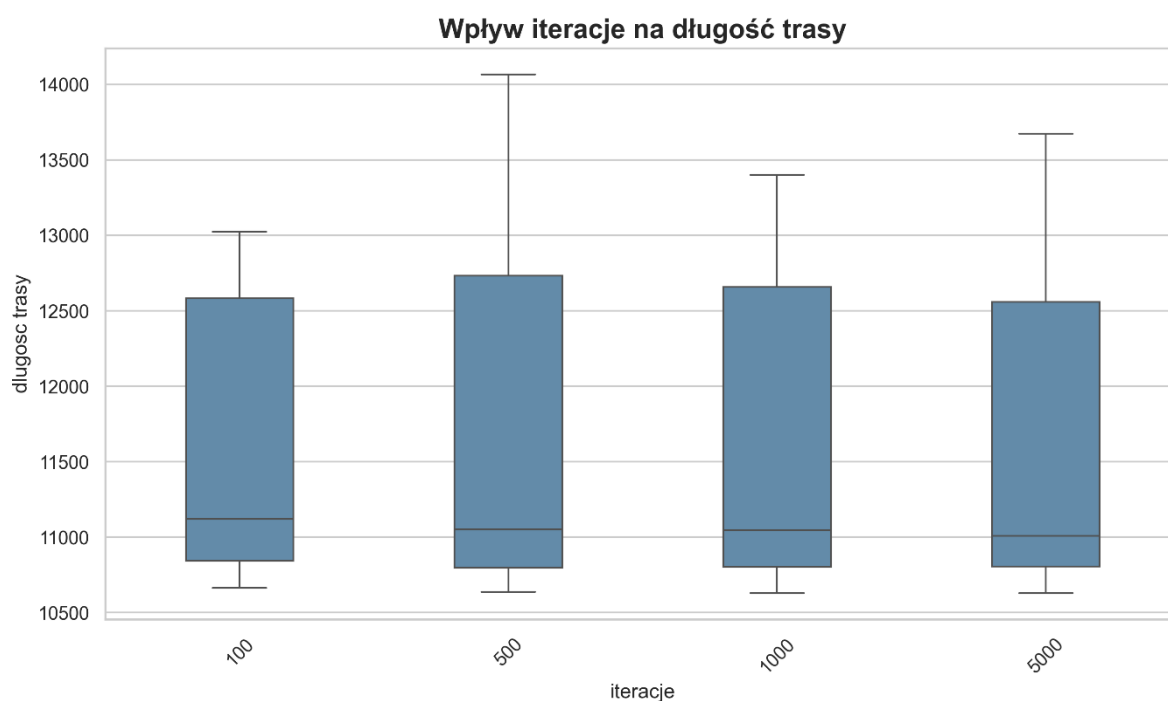
	iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	średni wynik	średni czas (ms)
16.	500	insert	50	11067.2	21057.9
17.	1000	insert	50	11092.6	19859.0
18.	1000	insert	100	11094.0	27362.3

4.	100	reverse	50	10755.2	22058.3
5.	1000	reverse	50	10765.3	13843.2
6.	5000	reverse	500	10767.2	44488.1
7.	1000	reverse	100	10768.3	17562.0
8.	5000	reverse	50	10777.1	13740.9
9.	500	reverse	50	10779.0	14287.1
10.	5000	reverse	1000	10788.0	79062.7
11.	5000	insert	100	10979.8	26045.0
12.	5000	insert	500	11017.1	57501.9
13.	5000	insert	1000	11029.3	89728.5
14.	500	insert	100	11038.5	25621.6
15.	1000	insert	500	11039.2	54706.5

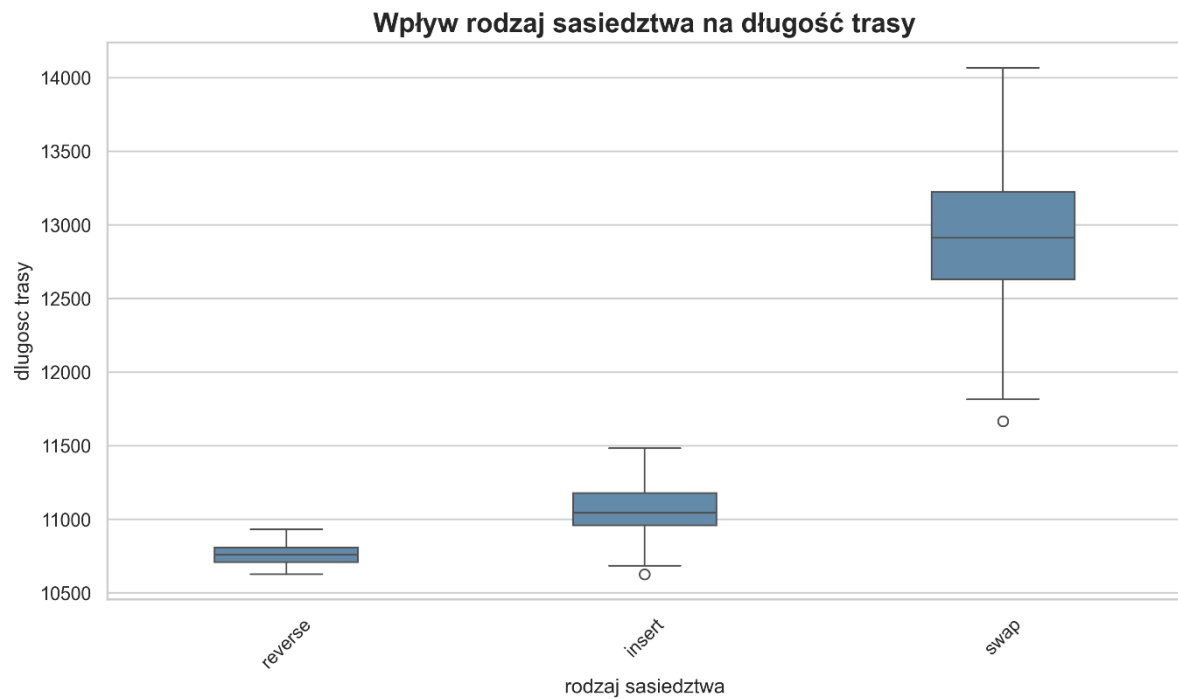
19.	5000	insert	50	11129.8	20816.4
20.	100	insert	50	11164.0	19935.3
21.	100	swap	50	12749.2	15184.9
22.	1000	swap	50	12758.9	15892.2
23.	5000	swap	50	12810.2	15429.2
24.	500	swap	100	12812.0	20291.2
25.	1000	swap	500	12813.9	48541.4
26.	5000	swap	500	12839.5	47337.8
27.	5000	swap	100	12920.3	19767.9
28.	1000	swap	100	12929.7	20051.9
29.	5000	swap	1000	12965.6	81460.7
30.	500	swap	50	13338.0	20196.7

Średnia długość trasy (ogólnie): 11574.03

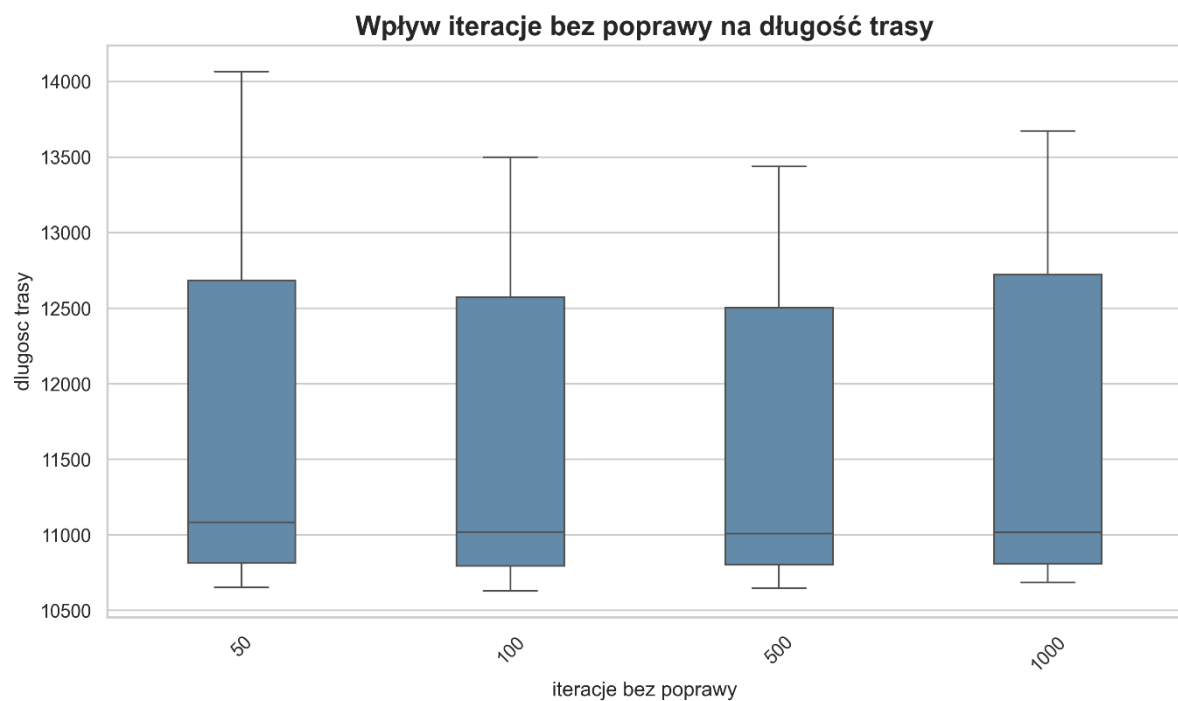
Średni wynik dla wszystkich tras jest gorszy niż ten uzyskany za pomocą solvera. Jak wcześniej zauważono, najlepsze wyniki osiągnięto dzięki zmianie sąsiedztwa metodą reverse, podczas gdy metoda swap dała najsłabsze rezultaty.



Iteracje	Średnia Najkrótsza Trasa	Min	Max
100	11556.13	10663	13024
500	11631	10638	14067
1000	11555.1	10628	13402
5000	11564.23	10628	13674



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	12893.73	11666	14067
insert	11065.15	10628	11484
reverse	10763.22	10628	10931



Iteracje bez poprawy	Średnia Najkrótsza Trasa	Min	Max
50	11598.88	10653	14067
100	11560.08	10628	13501
500	11535.15	10648	13438
1000	11594.3	10684	13674

## Liczba miast 76

10 najlepszych wyników:

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	wynik
1000	reverse	100	108644.9727
1000	reverse	50	108844.9421
5000	reverse	500	108869.1305
5000	reverse	100	108951.3004
5000	reverse	100	109091.8629
1000	reverse	100	109094.471
1000	reverse	50	109132.6551
1000	reverse	50	109138.0625
5000	reverse	50	109237.5408
500	reverse	100	109245.5804

Trasa najlepszego wyniku:

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

Najkrótsza trasa dla 76 miast, znaleziona za pomocą algorytmu wspinaczki z multistartem, wynosi 108 644,9727. Osiągnięto tym samym lepszy wynik niż uzyskany za pomocą solvera. Ponownie można zauważyć zależność – kombinacje wykorzystujące metodę zmiany sąsiedztwa reverse dają najlepsze rezultaty. W przypadku pozostałych parametrów, nie zaobserwowano wyraźnych zależności wskazujących na przewagę którejkolwiek kombinacji.



## 10 najgorszych tras

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	wynik
5000	swap	100	153423.1614
5000	swap	50	153834.0506
500	swap	50	153857.9061
1000	swap	50	154329.9677
100	swap	50	154472.1169
5000	swap	500	154481.9927
100	swap	50	154816.3236
1000	swap	100	154820.6988
500	swap	50	157859.083
5000	swap	50	158037.0818

Trasa najgorszego wyniku:

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

Najsłabsza trasa uzyskana za pomocą algorytmu wspinaczki z multistartem wynosi 158037,0818. Ponownie najsłabsze wyniki pochodzą z kombinacji wykorzystujących metodę zmiany sąsiedztwa swap.

Średnie długości tras dla każdej kombinacji:

	iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	średni wynik	średni czas (ms)
1.	1000	reverse	50	109904.1	49033.6
2.	1000	reverse	100	110085.1	58003.8
3.	5000	reverse	500	110159.4	126657.6
4.	5000	reverse	100	110187.2	57585.6
5.	1000	reverse	500	110295.4	160831.9
6.	5000	reverse	50	110297.4	47943.5
7.	5000	reverse	1000	110388.4	205370.6
8.	500	reverse	100	110446.4	57896.4
9.	100	reverse	50	110481.4	51036.2

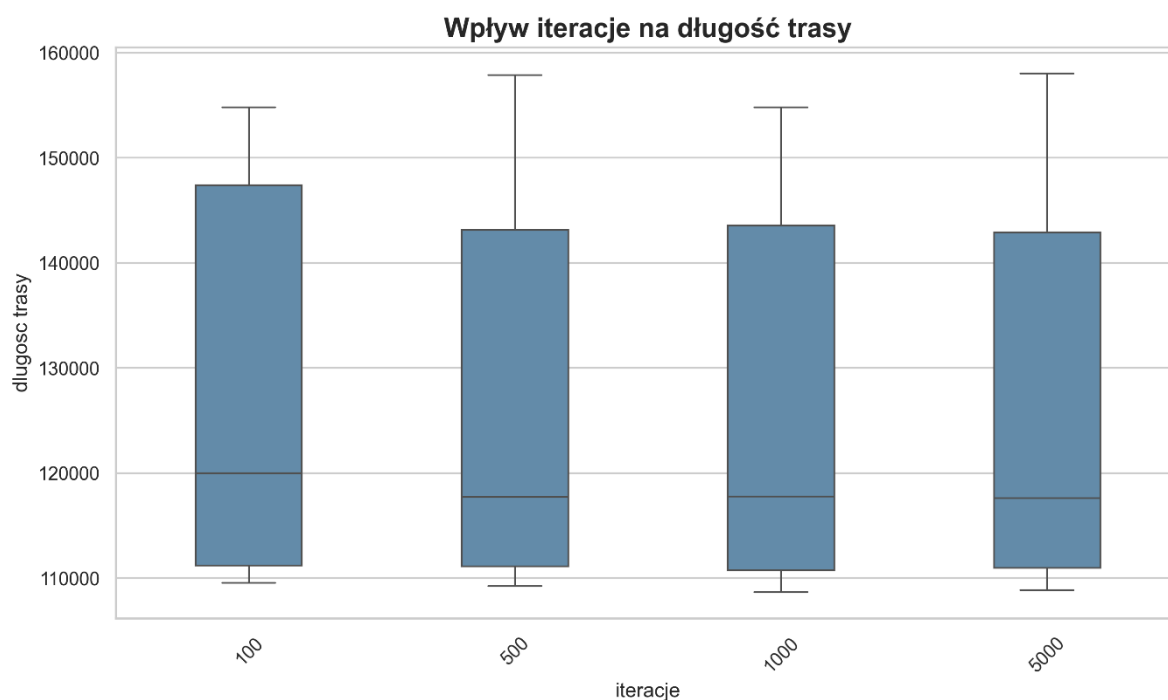
	iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	średni wynik	średni czas (ms)
16.	5000	insert	50	117963.1	80469.1
17.	1000	insert	100	118295.0	104117.2
18.	1000	insert	50	118675.7	78619.9
19.	500	insert	50	119195.0	77247.8
20.	100	insert	50	119770.5	78338.3
21.	5000	swap	100	145335.6	74555.0
22.	5000	swap	1000	145388.6	218659.1
23.	1000	swap	500	145448.1	141906.0
24.	500	swap	100	145775.2	77293.5

10.	500	reverse	50	110611.3	46843.9
11.	500	insert	100	115808.4	115248.7
12.	5000	insert	100	116137.8	104793.0
13.	1000	insert	500	116319.6	194031.5
14.	5000	insert	1000	116793.3	260803.5
15.	5000	insert	500	117665.2	178947.6

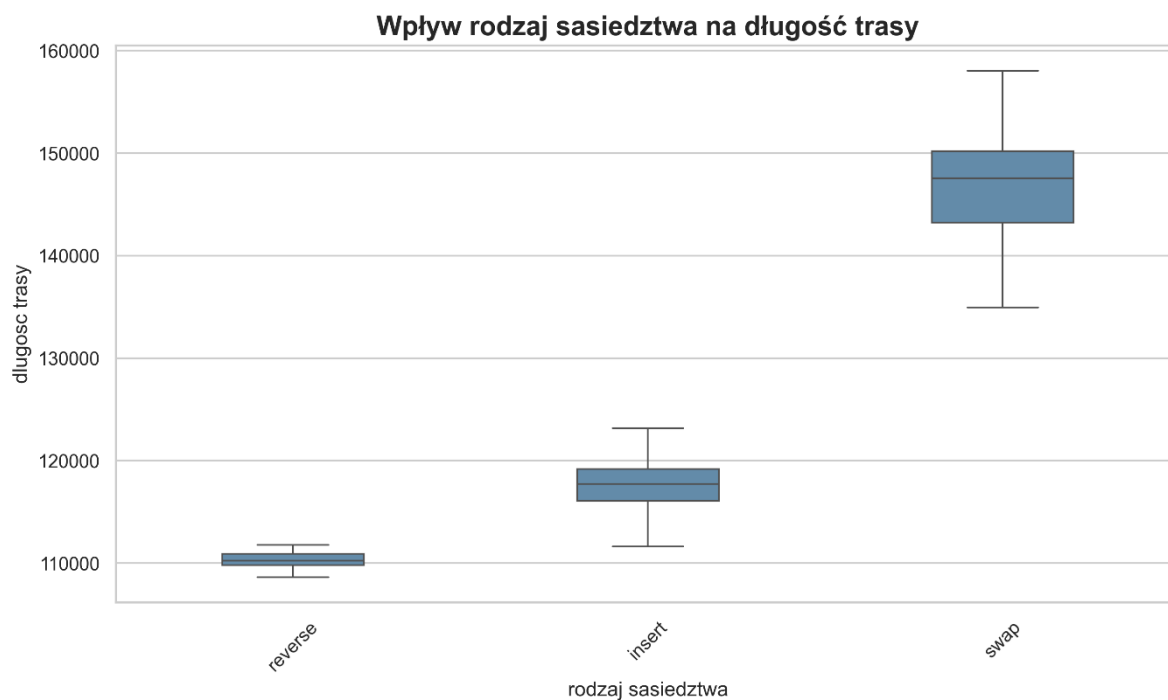
25.	5000	swap	500	145973.5	141014.4
26.	1000	swap	100	146749.7	75922.6
27.	500	swap	50	147576.6	57410.4
28.	1000	swap	50	148040.3	61850.4
29.	5000	swap	50	149543.0	60719.7
30.	100	swap	50	149743.2	62731.1

Średnia długość trasy (ogólnie): 124968.4

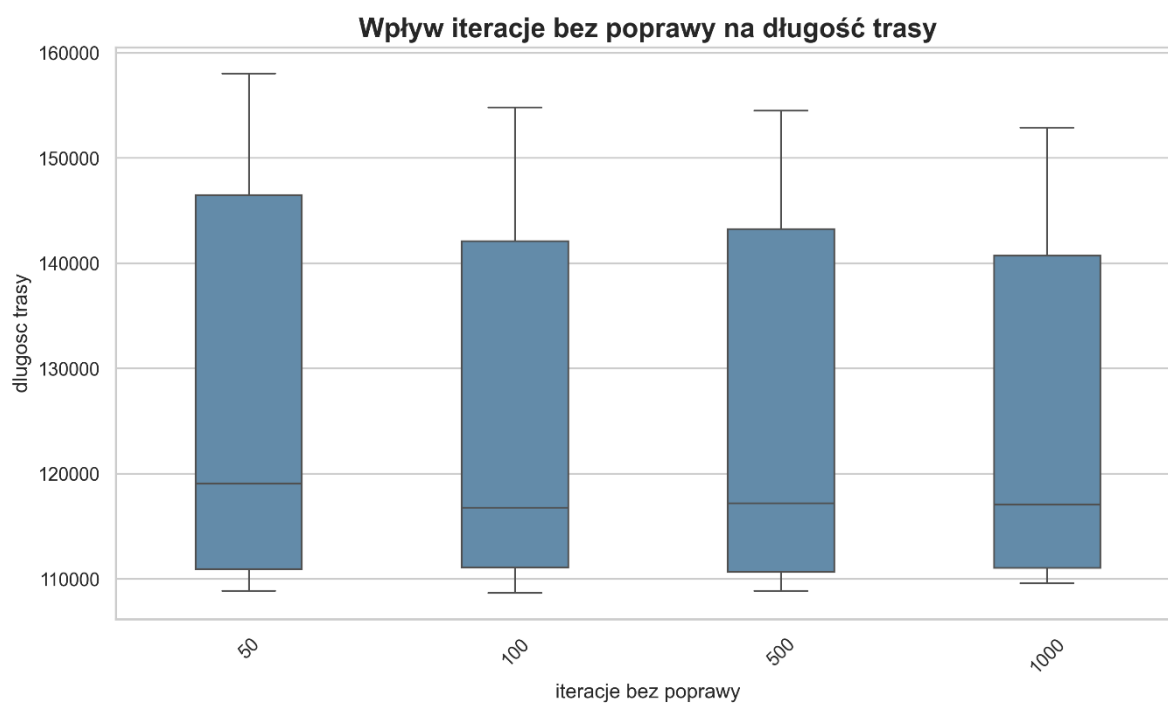
Przeciętna długość wszystkich tras jest gorsza od wyniku uzyskanego za pomocą solvera. Ponownie można zauważyć zależność – najlepsze wyniki pochodzą z kombinacji wykorzystujących metodę reverse, natomiast najsłabsze z tych, które stosują metodę swap.



Iteracje	Średnia Najkrótsza Trasa	Min	Max
100	126665	109536.2	154816.3
500	124902.1	109245.6	157859.1
1000	124868.1	108645	154820.7
5000	124652.7	108869.1	158037.1



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	146957.4	134940.88	158037.08
insert	117662.4	111623.86	123161.53
reverse	110285.6	108644.97	111792.48



Iteracje bez poprawy	Średnia Najkrótsza Trasa	Min	Max
50	125983.5	108844.94	158037.08
100	124313.4	108644.97	154820.70
500	124310.2	108869.13	154481.99
1000	124190.1	109589.73	152879.82

## Liczba miast 127

10 najlepszych wyników:

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	wynik
1000	reverse	500	120640.4326
5000	reverse	50	120782.8397
1000	reverse	500	121690.7155
1000	reverse	50	121828.032
5000	reverse	500	122221.8505
5000	reverse	100	122294.2332
5000	reverse	1000	122408.2547
500	reverse	100	122496.045
1000	reverse	500	122503.5938
5000	reverse	100	122507.8987

Trasa najlepszego wyniku:

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

Ponownie, rezultat uzyskany za pomocą algorytmu wspinaczki z multistartem dla 127 miast jest lepszy niż wynik osiągnięty w solverze. Również w tym przypadku powtarza się zależność, że wszystkie najlepsze wyniki uzyskano w kombinacjach wykorzystujących metodę reverse. Dla pozostałych parametrów nie zaobserwowano wyraźnych zależności.

### 10 najgorszych tras

iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	wynik
5000	swap	50	170218.4836
500	swap	100	170243.6671
1000	swap	500	170476.2361
5000	swap	500	170892.5871
5000	swap	500	171292.4177
100	swap	50	171979.3941
1000	swap	100	172786.2136
5000	swap	100	172948.1779
5000	swap	50	173785.8424
1000	swap	50	178106.7159

Trasa najgorszego wyniku:

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

Najgorsza trasa dla 127 miast wynosi 178 106,7159. Ponownie, wszystkie 10 najsłabszych wyników uzyskano przy wykorzystaniu metody zmiany sąsiedztwa swap.

Średnie długości tras dla każdej kombinacji:

	iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	średni wynik	średni czas (ms)
1.	1000	reverse	500	122276.6	510073.9
2.	5000	reverse	100	122942.7	289159.3
3.	5000	reverse	500	123451.2	511635.7
4.	500	reverse	100	123530.4	300542.1
5.	5000	reverse	1000	123654.3	745278.1
6.	500	reverse	50	123821.4	227780.6
7.	1000	reverse	50	123840.1	249900.9
8.	5000	reverse	50	123908.4	232228.2

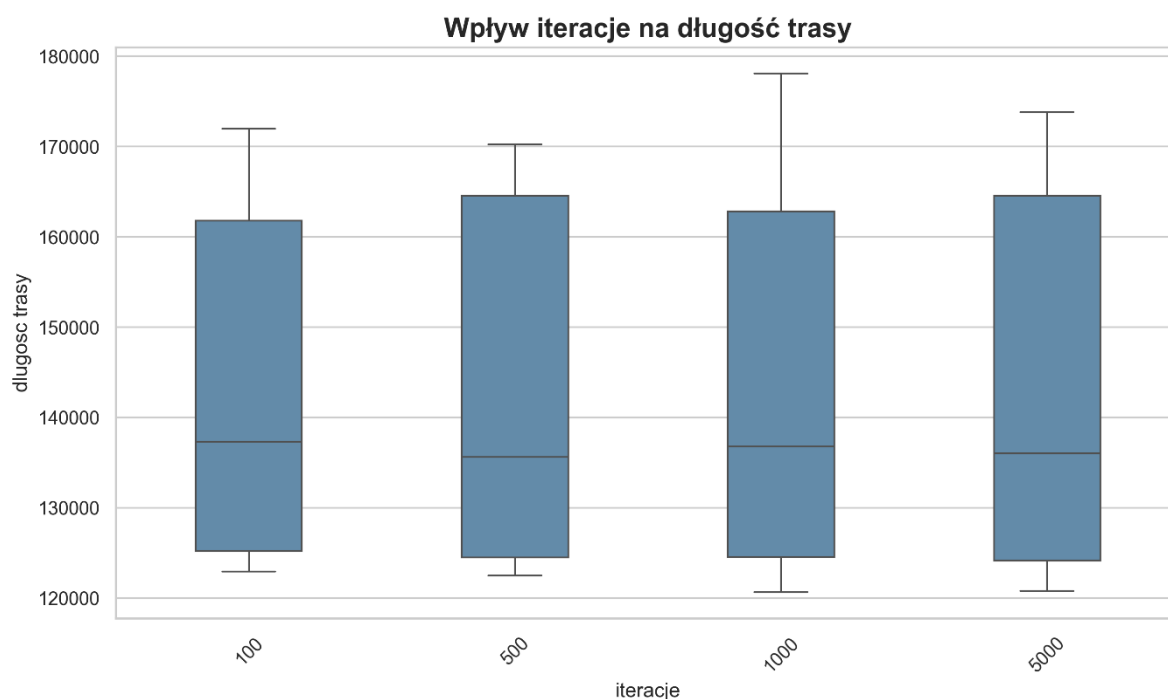
	iteracje	rodzaj sąsiedztwa	liczba iteracji bez poprawy	średni wynik	średni czas (ms)
16.	1000	insert	50	137076.5	429781.2
17.	100	insert	50	137083.2	446497.8
18.	5000	insert	100	137089.1	549301.2
19.	500	insert	50	137771.4	458302.6
20.	5000	insert	50	138798.4	426925.3
21.	1000	swap	500	164171.3	586735.2
22.	500	swap	50	165370.3	311683.2
23.	5000	swap	1000	166157.5	906554.9

9.	1000	reverse	100	123925.9	355891.9
10.	100	reverse	50	124280.9	234822.6
11.	500	insert	100	134235.8	598609.1
12.	5000	insert	500	134873.5	925796.2
13.	5000	insert	1000	135788.0	1157876.0
14.	1000	insert	100	136322.7	570714.2
15.	1000	insert	500	136654.7	937784.1

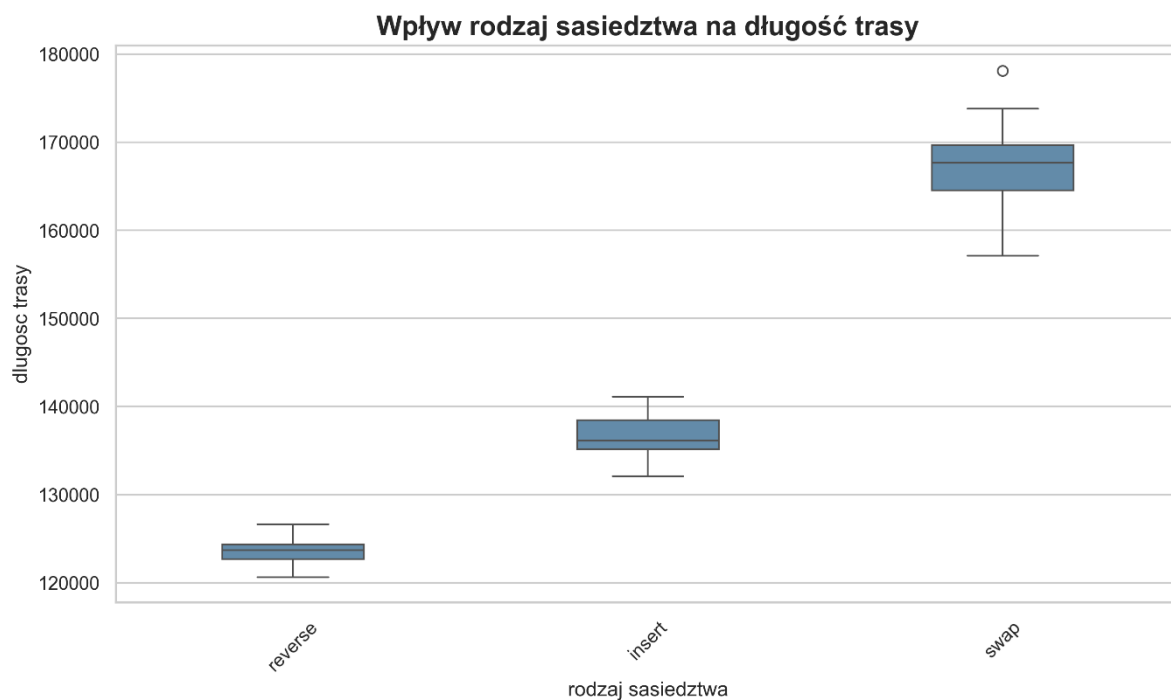
24.	100	swap	50	166367.3	304342.2
25.	1000	swap	100	166517.4	367758.9
26.	5000	swap	500	167480.4	590274.4
27.	5000	swap	100	167730.4	353519.5
28.	500	swap	100	168464.7	384651.6
29.	5000	swap	50	168867.5	296886.0
30.	1000	swap	50	169520.3	283865.9

Średnia długość trasy (ogólnie): 142399.1

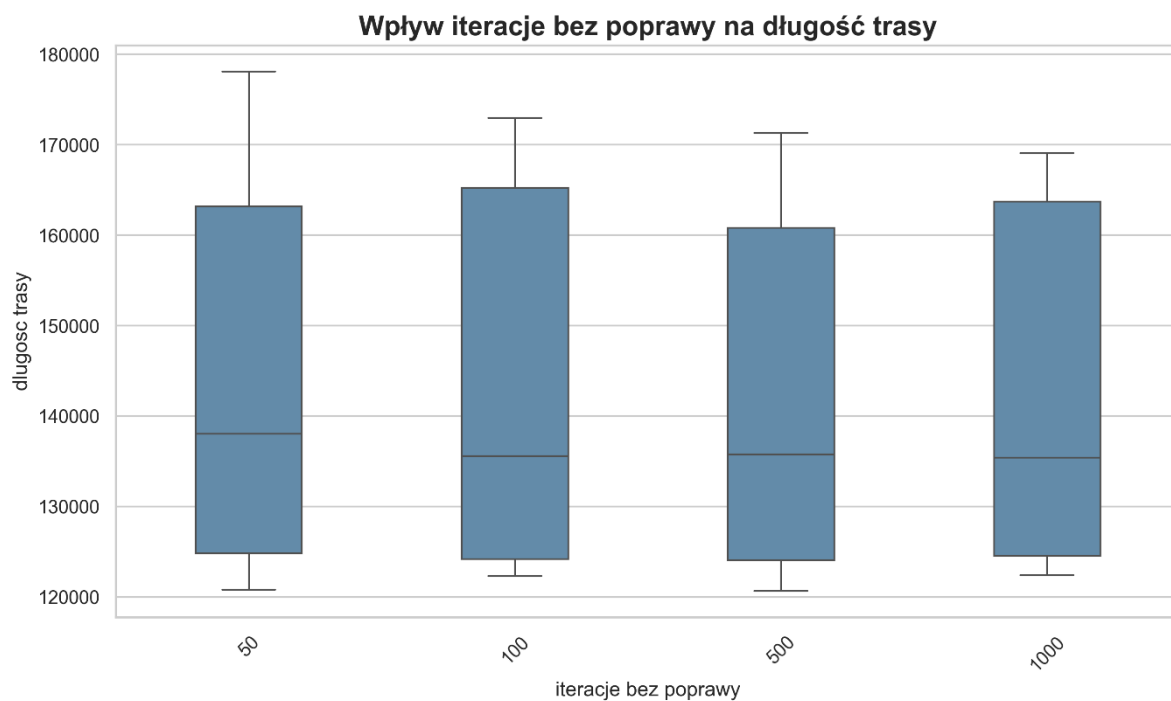
Średni wynik dla wszystkich tras znalezionych za pomocą algorytmu wspinaczki z multistartem jest gorszy niż ten uzyskany w solverze. Również dla 127 miast można zauważyć zależność – najlepsze wyniki pochodzą z metody reverse, podczas gdy najslabsze z metody swap. Dla pozostałych parametrów nie zaobserwowano wyraźnych zależności



Iteracje	Średnia Najkrótsza Trasa	Min	Max
100	142577.12	122917.66	171979.39
500	142199.01	122496.04	170243.67
1000	142256.15	120640.43	178106.72
5000	142561.79	120782.84	173785.84



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	167064.7	157117.84	178106.72
insert	136569.3	132084.89	141106.90
reverse	123563.2	120640.43	126615.50



Iteracje bez poprawy	Średnia Najkrótsza Trasa	Min	Max
50	143058.8	120782.84	178106.72
100	142306.6	122294.23	172948.18
500	141484.6	120640.43	171292.42
1000	141866.6	122408.25	169064.40

## 6. Algorytm genetyczny

**Algorytm genetyczny** (GA) to technika optymalizacji, która wykorzystuje proces iteracyjnego ulepszania rozwiązań na podstawie mechanizmów selekcji, krzyżowania i mutacji. Działa na zasadzie przetwarzania populacji potencjalnych rozwiązań, wybierając najlepsze z nich i tworząc nowe kombinacje, które stopniowo zbliżają się do optymalnego wyniku.

GA jest skuteczny w problemach wymagających przeszukiwania dużych przestrzeni rozwiązań, takich jak optymalizacja tras, harmonogramowanie czy projektowanie systemów. Dzięki swojej uniwersalności znajduje zastosowanie w różnych dziedzinach technicznych i inżynierskich.

### Działanie algorytmu:

1. Inicjalizacja populacji – Tworzymy początkową populację z losowych rozwiązań problemu.
2. Ewaluacja – Każde rozwiązanie oceniamy na podstawie funkcji celu, która mierzy jego jakość.
3. Selekcja rodziców – Wybieramy rozwiązania, które będą "rodzicami" i wezmą udział w tworzeniu nowych rozwiązań.
4. Krzyżowanie – Łączymy cechy wybranych rodziców, aby stworzyć nowe rozwiązania (potomków).
5. Mutacja – Wprowadzamy losowe zmiany w niektórych rozwiązaniach, aby zwiększyć różnorodność.
6. Tworzenie nowej populacji – Zastępujemy starą populację nowymi rozwiązaniami, które powstały z krzyżowania i mutacji.
7. Sprawdzenie warunku zakończenia – Kończymy działanie algorytmu, gdy osiągniemy maksymalną liczbę pokoleń lub gdy rozwiązania przestaną się poprawiać przez określoną liczbę iteracji.

### Opis parametrów:

#### Wielkość populacji:

Wielkość populacji odnosi się do liczby indywidualnych rozwiązań (osobników) w danym pokoleniu algorytmu genetycznego. Na początku algorytmu populacja jest generowana losowo, tworząc różnorodne możliwe rozwiązania problemu. Z każdą kolejną generacją algorytm przetwarza te rozwiązania w celu ich optymalizacji.

#### Liczba pokoleń:

Liczba pokoleń określa, ile razy algorytm przeprowadzi cały cykl operacji, takich jak selekcja, krzyżowanie, mutacja i ewaluacja rozwiązań. Z każdym pokoleniem populacja rozwiązań jest



modyfikowana, a nowe pokolenie zawiera lepsze (lub bardziej dopasowane) rozwiązania w porównaniu do poprzedniego.

### Rodzaj krzyżowania:

Rodzaj krzyżowania opisuje sposób tworzenia nowych rozwiązań (potomstwa) poprzez połączenie cech dwóch wybranych rodziców. W projekcie zastosowane zostały dwie techniki krzyżowania:

- **OX (Order Crossover)** - W tej metodzie wybrany fragment genów z pierwszego rodzica jest kopiowany do potomka w tej samej pozycji. Pozostałe miejsca są wypełniane elementami z drugiego rodzica, zachowując ich kolejność w jakich występują i unikając powtórzeń.
- **PMX (Partially Matched Crossover)** - Fragment genów z pierwszego rodzica jest kopiowany do potomka, a brakujące miejsca wypełnia się elementami z drugiego rodzica. W celu uniknięcia powtórzeń stosuje się mapowanie, które pozwala dopasować elementy do właściwych pozycji.

### Metoda doboru rodziców:

- **Skalowanie dopasowania**- Metoda ta dostosowuje wartości funkcji dopasowania, aby stosunek szans najlepszego osobnika do średniej w populacji wynosił 2:1. Osiąga się to przez odjęcie od funkcji dopasowania każdego osobnika różnicy  $2 * \text{średnia} - \text{max}$ . Skalowanie zmniejsza różnice między osobnikami, zapobiegając przedwczesnej zbieżności, ale w przypadku jednego wybitnego osobnika może zbyt spłaszczyć wartości dopasowania.
- **Ranking** - Osobniki są sortowane według funkcji dopasowania, a wybór rodziców zależy od pozycji w rankingu, nie od samej wartości dopasowania. Wyższe pozycje mają większe szanse na selekcję. Metoda zapobiega dominacji najlepszych osobników, wspierając różnorodność populacji.
- **Turniejowa** - W metodzie turniejowej losowo wybiera się grupę osobników, którzy konkurują ze sobą na podstawie wartości funkcji dopasowania. Zwycięzca tego turnieju zostaje wybrany jako rodzic. Selekcja odbywa się na podstawie bezpośredniej rywalizacji, gdzie lepsze osobniki mają większe szanse na wybór.

### Mutacje:

Mutacja w algorytmach genetycznych to technika wprowadzania losowych modyfikacji do rozwiązań w celu zwiększenia różnorodności populacji oraz uniknięcia szybkiej zbieżności do lokalnych optymalnych rozwiązań. Jednym z rodzajów mutacji jest mutacja przez zamianę (swap), która polega na wymianie miejscami dwóch losowo wybranych elementów w sekwencji rozwiązania.

### Wartości parametrów:

W poniższej tabeli przedstawiono parametry, które były zmieniane, oraz ich wartości w poszczególnych przypadkach. Wszystkich kombinacji parametrów było 384, a cały proces został powtórzony 10 razy dla każdego przypadku, aby wyeliminować wpływ losowości.

Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców
100	500	150	PMX	Skalowanie
500	1000	500	OX	Ranking
1000	2500	750		Turniejowa
2500	5000	1000		

Pozostałe stałe parametry:

- Prawdopodobieństwo krzyżowania: 0.8
- Prawdopodobieństwo mutacji: 0.2

### Liczba miast 48

#### 10 najlepszych wyników:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Długość trasy	Czas wykonania (ms)
1.	1000	2500	750	OX	Turniejowa	10648	8252
2.	2500	2500	1000	PMX	Turniejowa	10648	28154
3.	2500	1000	150	OX	Turniejowa	10648	4787
4.	2500	1000	150	OX	Turniejowa	10648	5944
5.	2500	5000	750	OX	Turniejowa	10648	12732
6.	2500	2500	1000	OX	Turniejowa	10648	48729
7.	1000	2500	500	OX	Turniejowa	10683	5653
8.	500	1000	500	OX	Turniejowa	10683	1796
9.	1000	2500	1000	OX	Turniejowa	10683	4995
10.	1000	1000	750	OX	Turniejowa	10684	3490

#### Kolejność miast dla najkrótszych tras:

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

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

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

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

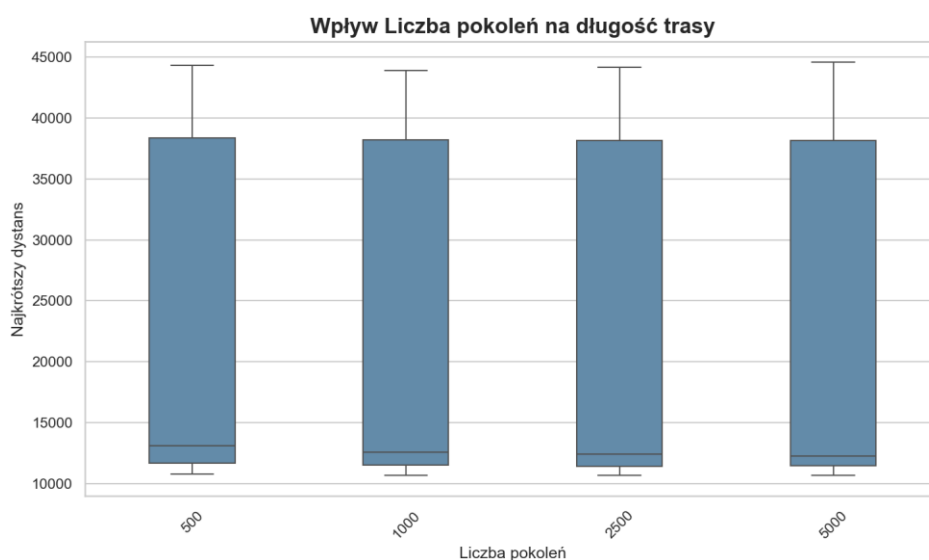
1, 8, 38, 31, 44, 18, 7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 46, 33, 15, 12, 20, 47, 11, 23, 14, 25, 13, 21, 32, 39, 48, 5, 42, 24, 10, 45, 35, 4, 26, 2, 29, 34, 41, 16, 22, 3, 40, 9

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

Najniższą wartością znaną dla macierzy 48 wynosi 10648, co jest dokładnie tą samą wartością, jaką udało się uzyskać za pomocą Solvera.

### 10 najgorszych wyników:

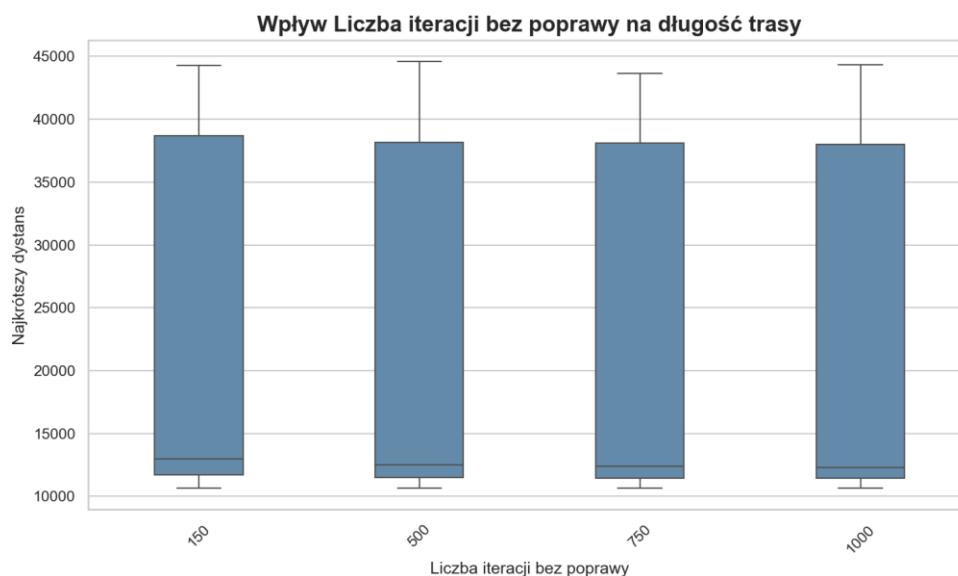
	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Długość trasy	Czas wykonania (ms)
1.	100	5000	500	PMX	Scaling	44604	196
2.	100	500	1000	OX	Scaling	44339	145
3.	100	500	1000	PMX	Scaling	44296	294
4.	100	500	150	OX	Scaling	44286	57
5.	100	2500	500	PMX	Scaling	44197	298
6.	100	500	500	PMX	Scaling	44001	264
7.	100	500	150	OX	Scaling	43957	49
8.	100	2500	500	PMX	Scaling	43928	238
9.	100	500	150	OX	Scaling	43926	45
10.	100	1000	150	OX	Scaling	43917	116



Liczba pokoleń	Średnia długość trasy	Minimum	Maksimum
5000	21045.34	10648	44604
2500	21122.99	10648	44197
1000	21232.36	10648	43917
500	21506.90	10761	44339

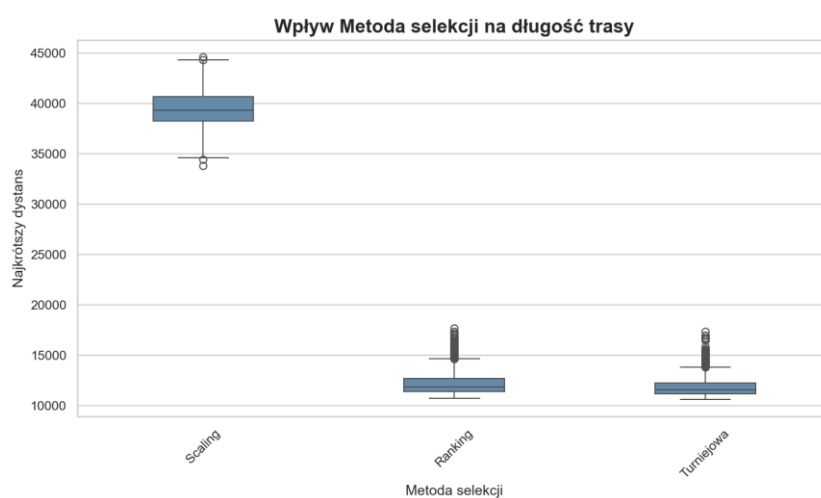
Wraz z malejącą liczbą pokoleń, średnia długość trasy rośnie, co sugeruje, że mniejsza liczba iteracji utrudnia osiągnięcie optymalnych rozwiązań. Z kolei większa liczba pokoleń umożliwia lepsze dopasowanie, prowadząc do krótszych tras. Mimo to, w przypadku liczby pokoleń 5000,

2500 i 1000, wartości minimalne długości trasy są na takim samym poziomie, co sugeruje, że nawet przy mniejszej liczbie pokoleń udało się osiągnąć podobnie optymalne rozwiązania.



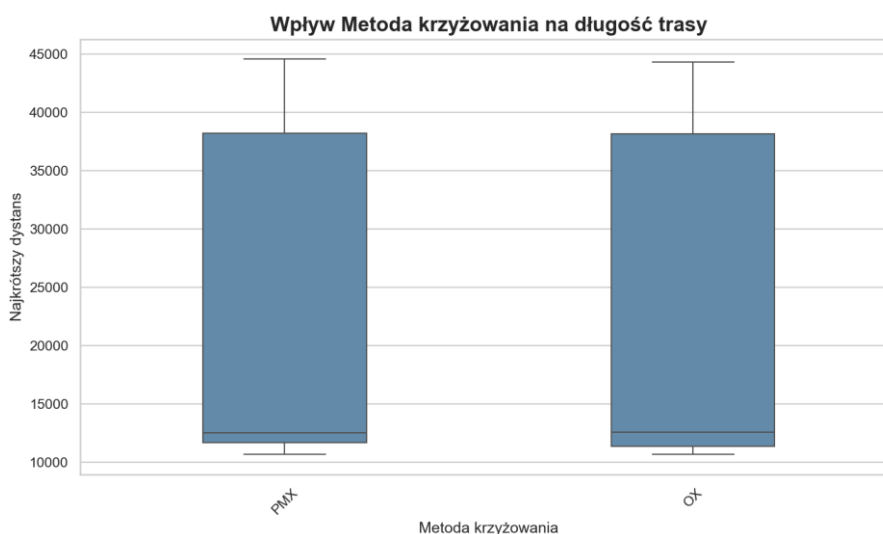
Liczba iteracji bez poprawy	Średnia długość trasy	Minimum	Maksimum
150	21547.41458	10648	44286
500	21161.97604	10683	44604
750	21133.36563	10648	43645
1000	21064.83333	10648	44339

Zmiana liczby iteracji bez poprawy nie ma istotnego wpływu na długość trasy. Widać tylko nieznaczne różnice w średnich, które mogą być wynikiem losowości. Minimalne wartości długości trasy pozostają na tym samym poziomie, a różnice w maksymalnych wartościach są marginalne.



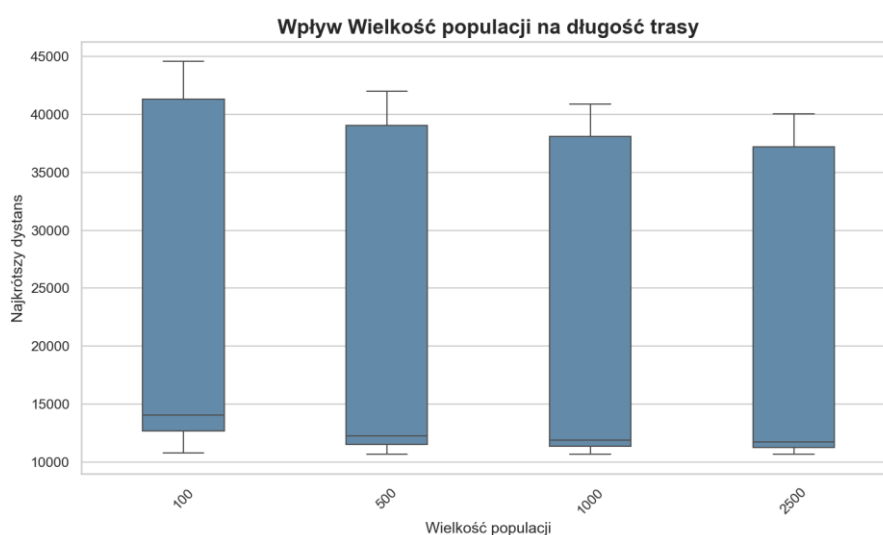
Metoda selekcji	Średnia długość trasy	Minimum	Maksimum
Ranking	12233.73125	10725	17707
Scaling	39538.11875	33840	44604
Turniejowa	11908.84219	10648	17367

Różne metody selekcji mają istotny wpływ na długość trasy. Metoda rankingowa prowadzi do najkrótszej średniej długości trasy, z minimalną wartością wynoszącą 10725, co wskazuje na skuteczną optymalizację. Metoda turniejowa daje bardzo podobne wyniki, z minimalną wartością 10648, co sugeruje jej efektywność w optymalizacji. Z kolei metoda skalowania generuje znacznie dłuższe trasy, z minimalną wartością 33840, co wskazuje na gorszą skuteczność tej metody w porównaniu do pozostałych.



Metoda krzyżowania	Średnia długość trasy	Minimum	Maksimum
OX	21164.94063	10648	44339
PMX	21288.85417	10648	44604

Obie metody krzyżowania – OX i PMX – dają bardzo podobne wyniki pod względem długości trasy. Różnice w średnich długościach tras są niewielkie, a minimalne i maksymalne wartości są zbliżone, co sugeruje, że obie metody działają równie skutecznie, a wszelkie różnice mogą wynikać z losowości.



Wielkość populacji	Średnia długość trasy	Minimum	Maksimum
100	22845.91	10762	44604
500	21132.76	10683	42034
1000	20660.53	10648	40884
2500	20268.39	10648	40059

Wraz ze wzrostem wielkości populacji, średnia długość trasy maleje, co wskazuje na lepszą jakość optymalizacji przy większej liczbie osobników. Minimalne wartości długości trasy pozostają na podobnym poziomie, natomiast różnice pojawiają się głównie w wartościach maksymalnych, które są mniejsze w przypadku większej populacji. Oznacza to, że większe populacje prowadzą do bardziej efektywnych rozwiązań, generując krótsze trasy.

#### Najlepsze średnie wyniki dla danych konfiguracji parametrów:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Średnia długość trasy	Czas wykonania (ms)
1.	2500	2500	1000	OX	Turniejowa	10818.6	23012
2.	2500	5000	1000	OX	Turniejowa	10887.5	22936
3.	2500	2500	750	OX	Turniejowa	10905.5	16366.7
4.	2500	5000	500	OX	Turniejowa	10909.8	15574.9
5.	1000	2500	1000	OX	Turniejowa	10923.4	7799.4
6.	2500	5000	750	OX	Turniejowa	10948.7	18382.3
7.	2500	1000	1000	OX	Turniejowa	10969.8	10297
8.	2500	1000	500	OX	Turniejowa	11015.2	10480
9.	1000	2500	750	OX	Turniejowa	11041.2	7766.1
10.	2500	2500	500	OX	Turniejowa	11090	10306

W przeprowadzonych eksperymentach przetestowano 384 różne kombinacje parametrów algorytmu genetycznego, obejmujące wielkość populacji, liczbę pokoleń, liczbę iteracji bez poprawy, rodzaj krzyżowania oraz metodę doboru rodziców. Spośród tych kombinacji, 10 najlepszych średnich wyników uzyskano przy zastosowaniu krzyżowania OX oraz doboru rodziców metodą turniejową. Średnie wyniki były zazwyczaj lepsze przy najwyższych wartościach pozostałych parametrów, takich jak większa liczba pokoleń, większa liczba iteracji bez poprawy czy większa wielkość populacji.

## Liczba miast 76

### 10 najlepszych wyników:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Długość trasy	Czas wykonania (ms)
1.	2500	2500	750	OX	Turniejowa	108159.4383	33009
2.	2500	2500	500	OX	Turniejowa	108233.9876	36336
3.	1000	5000	1000	OX	Turniejowa	108280.4566	26543
4.	2500	5000	750	OX	Turniejowa	108332.9335	20798
5.	2500	1000	500	OX	Turniejowa	108347.1611	13600
6.	2500	2500	1000	OX	Turniejowa	108428.9081	34438
7.	2500	5000	1000	OX	Turniejowa	108444.0472	50680
8.	2500	2500	500	OX	Turniejowa	108468.0331	33143
9.	2500	2500	750	OX	Turniejowa	108728.5848	41315
10.	2500	1000	1000	OX	Turniejowa	108833.6343	14508

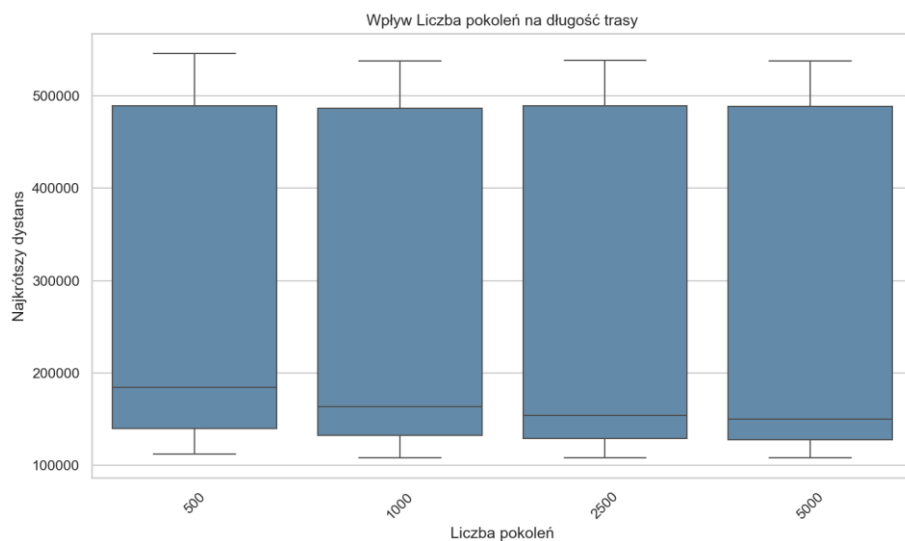
### Kolejność miast dla najkrótszej trasy:

5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 74, 15, 16, 17, 18, 37, 36, 38, 39, 40, 34, 35, 33, 32, 29, 30, 31, 19, 20, 26, 27, 28, 43, 42, 54, 53, 52, 55, 56, 57, 58, 59, 60, 41, 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

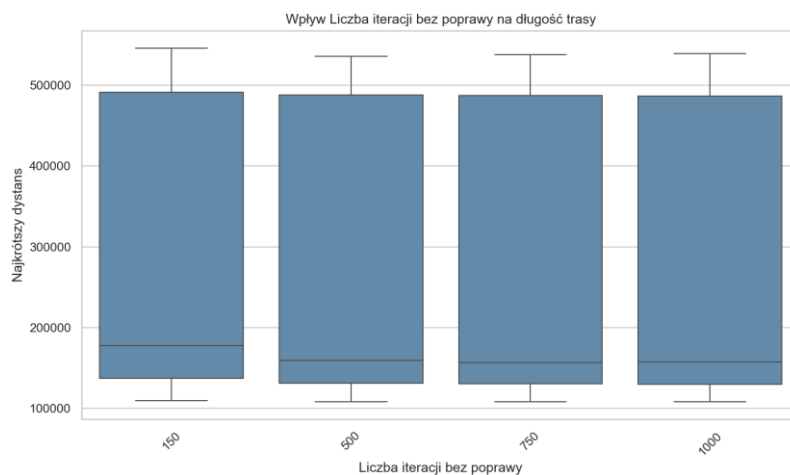
W przypadku macierzy 76 udało się osiągnąć wynik lepszy niż ten uzyskany za pomocą Solvera, który wynosił 111228.

### 10 najgorszych wyników:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Długość trasy	Czas wykonania (ms)
1.	100	500	150	OX	Skalowanie	546007.9144	68
2.	100	500	1000	PMX	Skalowanie	539358.6783	365
3.	100	2500	150	OX	Skalowanie	538712.6869	105
4.	100	5000	750	PMX	Skalowanie	537978.0388	518
5.	100	1000	150	PMX	Skalowanie	537681.5165	110
6.	100	2500	150	OX	Skalowanie	536885.6579	62
7.	100	5000	500	PMX	Skalowanie	536183.5882	358
8.	100	5000	1000	OX	Skalowanie	535606.8822	600
9.	100	1000	1000	OX	Skalowanie	535490.4446	419
10.	100	1000	500	PMX	Skalowanie	535434.8872	356

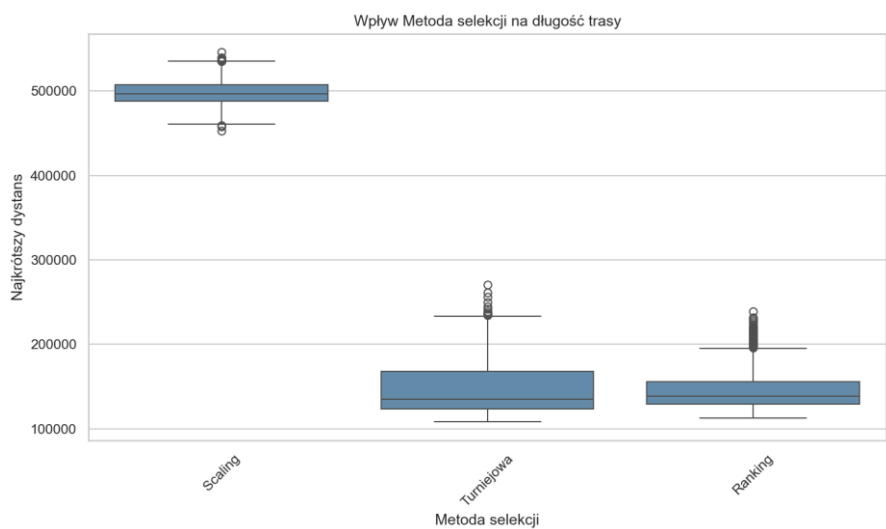


Liczba pokoleń	Średnia długość trasy	Minimum	Maksimum
5000	259594.0541	108280.4566	537978.0388
2500	260405.9095	108159.4383	538712.6869
1000	263751.2022	108347.1611	537681.5165
500	272410.5359	112446.1086	546007.9144

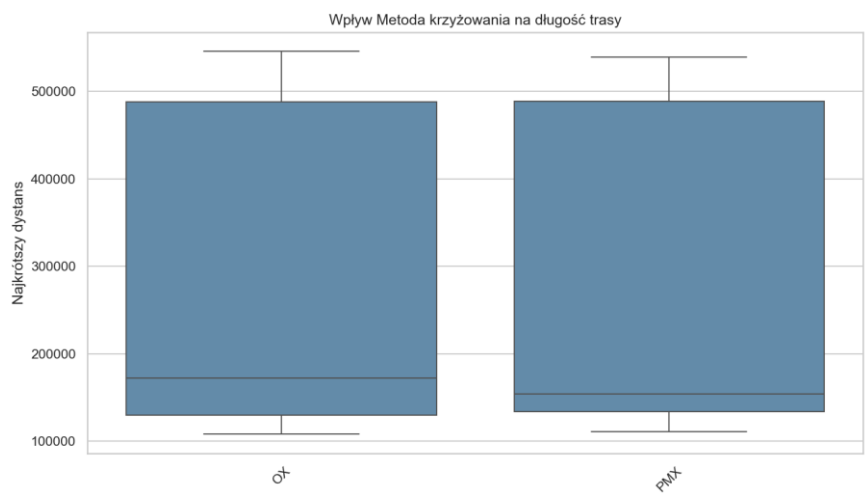


Liczba iteracji bez poprawy	Średnia długość trasy	Minimum	Maksimum
150	270360.672	109153.0424	546007.9144
500	262588.3429	108233.9876	536183.5882
750	261755.0318	108159.4383	537978.0388
1000	261457.6549	108280.4566	539358.6783

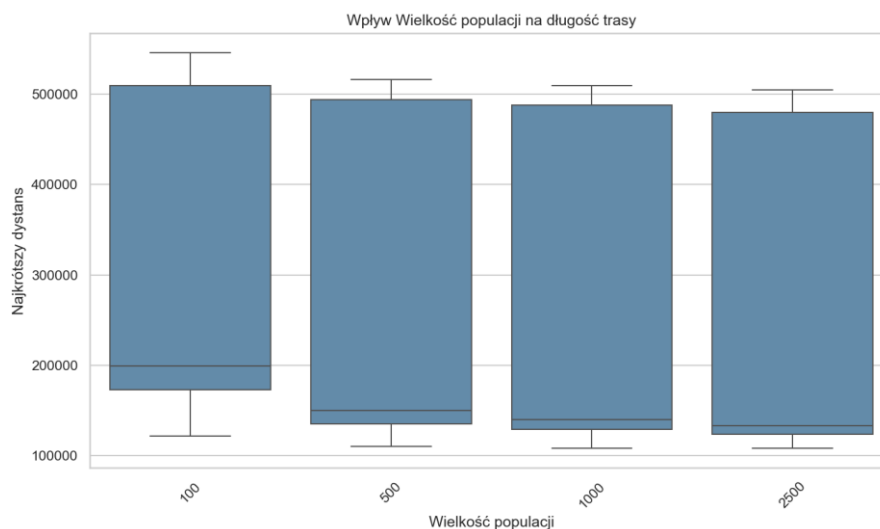




Metoda selekcji	Średnia długość trasy	Minimum	Maksimum
Ranking	146640.5028	112446.1086	239119.7362
Scaling	498507.647	452977.8868	546007.9144
Turniejowa	146973.1265	108159.4383	270596.1543



Metoda krzyżowania	Średnia długość trasy	Minimum	Maksimum
OX	265169.3204	108159.4383	546007.9144
PMX	262911.5304	110670.7926	539358.6783



Wielkość populacji	Średnia długość trasy	Minimum	Maksimum
100	293992.4602	121351.5731	546007.9144
500	261563.6167	110047.8597	516428.1277
1000	253911.7337	108280.4566	509588.8659
2500	246693.8911	108159.4383	504990.2981

Dla macierzy 78 miast wyniki wpływu parametrów na długość trasy są podobne do wniosków dotyczących macierzy 48 miast. Zwiększenie liczby pokoleń, liczby iteracji bez poprawy oraz wielkości populacji prowadzi do poprawy jakości rozwiązań, ale efekty stają się minimalne po osiągnięciu pewnego punktu. Na przykład, w przypadku liczby pokoleń, różnice między 5000, 2500 i 1000 pokoleniami są niewielkie, a dalsze zwiększanie liczby pokoleń przestaje znacząco wpływać na poprawę wyników. Podobnie, liczba iteracji bez poprawy ma również mniejszy wpływ, szczególnie po 500 iteracjach.

W przypadku metod selekcji, rankingowa daje najlepsze wyniki, a skalowanie prowadzi do gorszych rezultatów. Selekcja turniejowa, choć nieco gorsza od rankingowej, również daje dobre wyniki. Jeśli chodzi o metodę krzyżowania, różnica między OX a PMX jest minimalna, gdzie OX daje lepsze rezultaty, ale różnice te nie są duże.

Wielkość populacji ma natomiast wyraźny wpływ na jakość rozwiązań. Zwiększenie liczby osobników prowadzi do lepszych wyników, a największe korzyści daje populacja rzędu 2500 osobników, chociaż różnice między 1000 i 2500 osobnikami są stosunkowo niewielkie.

Podsumowując, podobnie jak w przypadku mniejszej macierzy, zwiększenie liczby pokoleń, iteracji oraz wielkości populacji wpływa na jakość rozwiązywania, jednak po pewnym punkcie poprawy stają się coraz mniejsze.

#### Najlepsze średnie wyniki dla danych konfiguracji parametrów:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Średnia długość trasy	Czas wykonania (ms)
1.	2500	5000	750	OX	Turniejowa	111149.0932	33344.3
2.	2500	2500	500	OX	Turniejowa	111364.1059	27865.4
3.	2500	2500	750	OX	Turniejowa	111371.7112	31100.9

4.	2500	5000	1000	OX	Turniejowa	112050.6422	40237.5
5.	2500	2500	1000	OX	Turniejowa	113257.779	33513.7
6.	1000	5000	750	OX	Turniejowa	113868.9549	12668.5
7.	2500	1000	750	OX	Turniejowa	114524.2898	13955.8
8.	2500	5000	500	OX	Turniejowa	114615.0621	23832.9
9.	1000	2500	1000	OX	Turniejowa	114999.2426	13105.1
10.	2500	1000	1000	OX	Turniejowa	115596.0294	14007.1

Najwyższe 10 średnich wyników dla konkretnych kombinacji parametrów uzyskano przy zastosowaniu metody krzyżowania OX oraz turniejowej metody doboru rodziców. Dla pozostałych parametrów, najwyższe wyniki osiągnięto przy wyższych wartościach tych zmiennych.

### Liczba miast 127

#### 10 najlepszych wyników:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Długość trasy	Czas wykonania (ms)
1.	2500	5000	500	PMX	Turniejowa	125582.2275	151935
2.	2500	5000	750	PMX	Turniejowa	126561.9543	147462
3.	2500	2500	500	PMX	Turniejowa	126964.9436	79602
4.	2500	5000	150	PMX	Turniejowa	128273.6989	39507
5.	2500	2500	1000	PMX	Turniejowa	128365.2578	111823
6.	2500	2500	1000	PMX	Turniejowa	128456.0933	115201
7.	2500	5000	1000	PMX	Turniejowa	129063.9195	128063
8.	2500	5000	1000	PMX	Turniejowa	129066.9788	225576
9.	2500	1000	500	PMX	Turniejowa	129964.8865	47618
10.	1000	5000	750	PMX	Turniejowa	130556.302	62607

#### Kolejność miast dla najkrótszej trasy:

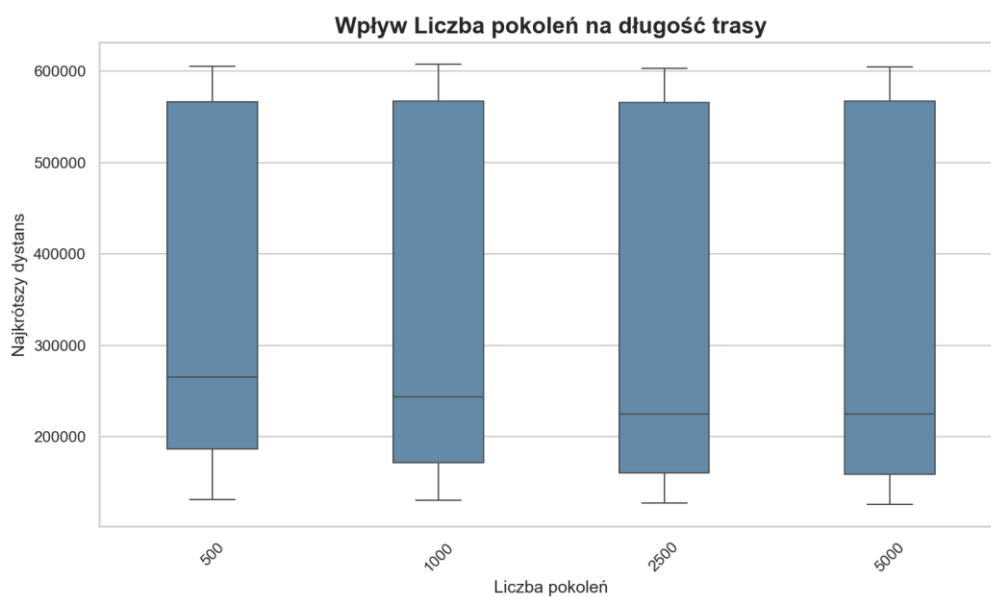
69, 96, 109, 110, 85, 86, 87, 88, 119, 63, 126, 82, 83, 102, 101, 98, 97, 123, 95, 93, 127, 107, 111, 112, 94, 46, 49, 47, 53, 118, 48, 45, 103, 44, 54, 57, 121, 56, 5, 100, 58, 64, 113, 125, 104, 89, 92, 99, 65, 66, 55, 124, 52, 115, 13, 50, 51, 2, 35, 40, 43, 34, 42, 39, 38, 25, 29, 32, 28, 122, 33,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

Dla macierzy 127 algorytm genetyczny osiągnął lepszy wynik niż Solver, choć różnica ta jest minimalna i wynosi zaledwie 224,21.

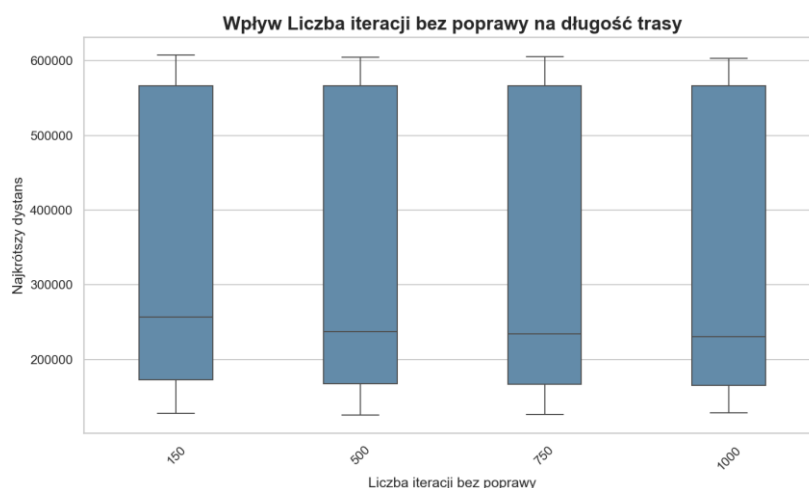
#### 10 najgorszych wyników:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Długość trasy	Czas wykonania (ms)
1.	100	1000	150	OX	Scaling	607684.5982	128
2.	100	500	750	OX	Scaling	605262.4964	373
3.	100	5000	500	OX	Scaling	604515.7452	399

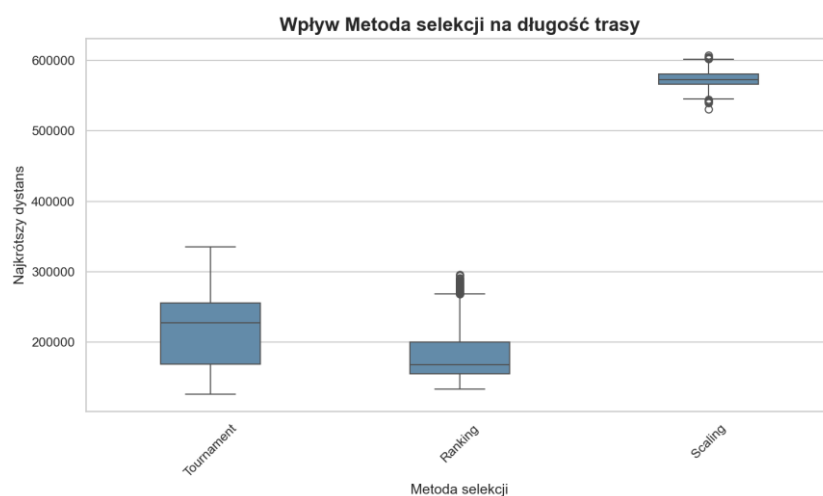
4.	100	2500	1000	OX	Scaling	603323.3111	743
5.	100	500	150	OX	Scaling	602853.2883	115
6.	100	500	1000	OX	Scaling	602345.3132	416
7.	100	5000	1000	PMX	Scaling	602304.9288	1561
8.	100	2500	750	OX	Scaling	601802.039	601
9.	100	5000	500	OX	Scaling	601143.3781	392
10.	100	1000	1000	PMX	Scaling	601083.7054	1532



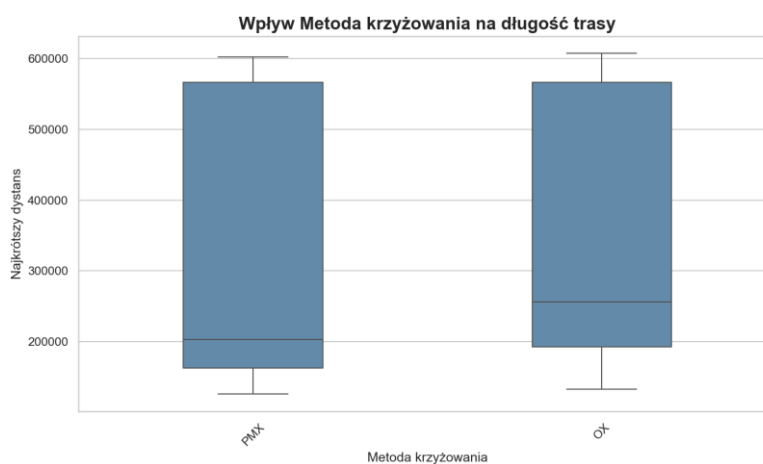
Liczba pokoleń	Średnia długość trasy	Minimum	Maksimum
5000	316213.6408	125582.2275	604515.7452
2500	316840.484	126964.9436	603323.3111
1000	325471.6763	129964.8865	607684.5982
500	336209.8778	131174.3556	605262.4964



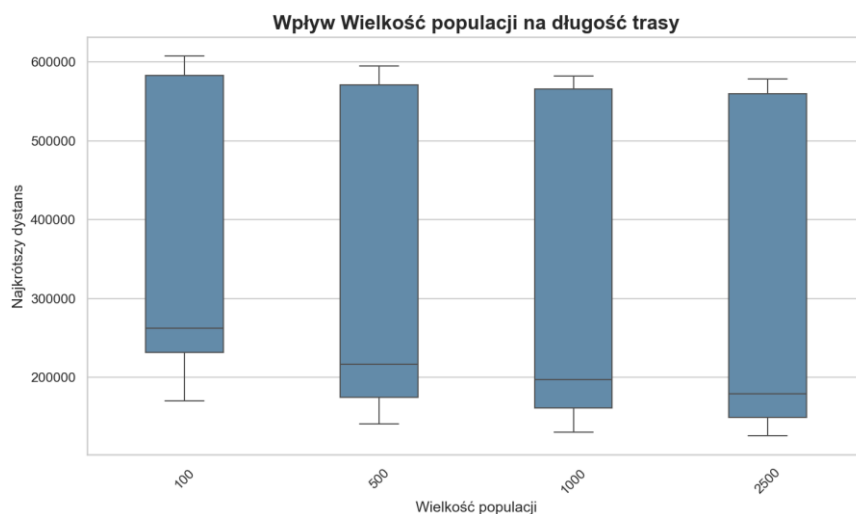
Liczba iteracji bez poprawy	Średnia długość trasy	Minimum	Maksimum
150	329896.8753	128273.6989	607684.5982
500	322590.087	125582.2275	604515.7452
750	321591.3631	126561.9543	605262.4964
1000	320657.3534	128365.2578	603323.3111



Metoda selekcji	Średnia długość trasy	Minimum	Maksimum
Ranking	181417.6546	132740.5415	295542.5254
Scaling	573684.5661	530564.4035	607684.5982
Turniejowa	215949.5385	125582.2275	335385.7909



Metoda krzyżowania	Średnia długość trasy	Minimum	Maksimum
OX	334026.2131	132740.5415	607684.5982
PMX	313341.6264	125582.2275	602304.9288



Wielkość populacji	Średnia długość trasy	Minimum	Maksimum
100	356342.037	170062.7933	607684.5982
500	322283.4543	140828.9717	594601.1996
1000	312501.9299	130556.302	582145.4101
2500	303608.2576	125582.2275	578015.614

Dla macierzy 127 miast, wpływ parametrów na długość trasy jest podobny do mniejszych macierzy, jednak zmiany w większym problemie mają bardziej wyraźny wpływ.

Zwiększenie liczby pokoleń poprawia wyniki, ale różnice między 5000, 2500 i 1000 pokoleniami są minimalne. Liczba iteracji bez poprawy również ma mniejszy wpływ po osiągnięciu pewnego progu.

Metoda selekcji rankingowej daje najniższą średnią długość trasy, jednak selekcja turniejowa prowadzi do najlepszego wyniku, co sugeruje, że chociaż rankingowa ma lepszą średnią, turniejowa może osiągnąć lepsze optymalne rozwiązania. Selekcja poprzez skalowanie prowadzi do gorszych efektów. W przypadku krzyżowania, PMX daje lepsze wyniki niż OX, szczególnie przy większych problemach.

Wielkość populacji ma istotny wpływ na jakość rozwiązań. Zwiększenie liczby osobników poprawia wyniki, przy czym największe korzyści uzyskano przy 2500 osobnikach, gdzie wyniki były najlepsze. Większa populacja umożliwia lepszą eksplorację przestrzeni rozwiązań, co prowadzi do bardziej optymalnych wyników, szczególnie w większych problemach, jak ten z 127 miastami.

#### Najlepsze średnie wyniki dla danych konfiguracji parametrów:

	Wielkość populacji	Liczba pokoleń	Liczba iteracji bez poprawy	Rodzaj krzyżowania	Metoda doboru rodziców	Średnia długość trasy	Czas wykonania (ms)
1.	2500	2500	1000	PMX	Turniejowa	135808.8157	120048.3
2.	2500	5000	1000	PMX	Turniejowa	137608.4781	191416.9
3.	2500	5000	500	PMX	Turniejowa	140043.0779	139194
4.	2500	2500	150	PMX	Turniejowa	141202.1736	55329.3
5.	2500	2500	500	PMX	Turniejowa	141841.2838	102701.4
6.	2500	2500	750	OX	Ranking	142067.564	204975.4

7.	2500	5000	750	OX	Ranking	142241.6097	252785
8.	2500	5000	500	OX	Ranking	142482.8915	224631.7
9.	2500	5000	750	PMX	Turniejowa	142529.0645	134845.9
10.	2500	1000	150	PMX	Turniejowa	143297.2879	47095.1

Dla macierzy 127 nie zaobserwowano wyraźnej dominacji konkretnej metody krzyżowania ani metody doboru rodziców. W tym przypadku widoczna jest większa różnorodność kombinacji parametrów prowadzących do najlepszych średnich wyników.

## 7. Algorytm Tabu Search

Przeszukiwanie tabu (*Tabu search*, *TS*) – metaheurystyka stosowana do rozwiązywania problemów optymalizacyjnych. Podstawową ideą algorytmu jest przeszukiwanie przestrzeni, stworzonej ze wszystkich możliwych rozwiązań, za pomocą sekwencji ruchów. W sekwencji ruchów istnieją ruchy niedozwolone, ruchy tabu. Algorytm unika oscylacji wokół optimum lokalnego dzięki przechowywaniu informacji o sprawdzonych już rozwiązaniach w postaci listy tabu (TL). Rozwiązanie jest zastępowane przez najlepszego sąsiada, nawet gdy prowadzi do gorszego rozwiązania.

Parametry algorytmu:

**Lista tabu** zawiera ruchy zabronione, czyli takie które zostały już ostatnio wykonane. Parametr nazywany tutaj „Długość listy tabu” określa **kadencję**, czyli liczbę iteracji, przez które dany ruch pozostaje na liście tabu.

Czasami można łamać tabu – wykonywać ruchy niedozwolone (tzw. **kryterium aspiracji**). Dzieje się to głównie wtedy, kiedy ruch "tabu" da rozwiązanie lepsze od znalezionej do tej pory.

**Rodzaj sąsiedztwa** określa sposób ruchu, który modyfikuje ścieżkę w celu eksploracji przestrzeni rozwiązań.

Wartości parametrów:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium
10	100	swap	liczba iteracji
50	500	insert	liczba iteracji bez poprawy
100	1000	inverse	
500	5000		

Z przyjętych wartości parametrów wygenerowano 96 różnych kombinacji, z których każda została powtórzona 5 razy.

## Liczba miast 48

10 najlepszych wyników:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Najkrótszy dystans	Czas (milisekundy)
50	1000	inverse	liczba iteracji	10628	906,4201
50	5000	inverse	liczba iteracji bez poprawy	10628	2641,0521
50	5000	inverse	liczba iteracji	10628	4893,9621
50	5000	inverse	liczba iteracji	10628	6433,5173
100	1000	inverse	liczba iteracji bez poprawy	10628	691,3327
100	1000	inverse	liczba iteracji bez poprawy	10628	1138,5434
100	1000	inverse	liczba iteracji bez poprawy	10628	581,9027
100	1000	inverse	liczba iteracji	10628	1119,8247
100	5000	inverse	liczba iteracji bez poprawy	10628	3328,5807
100	5000	inverse	liczba iteracji bez poprawy	10628	2918,8013

Trasa pierwszego spośród najlepszych rozwiązań: 42, 5, 48, 39, 32, 21, 47, 20, 33, 46, 36, 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

Wszystkie 10 pierwszych miejsc, a nawet aż o 9 więcej niż widać w powyższej tabeli, zajmują wyniki o długości trasy wynoszącej dokładnie 10628 (< solver). Jednak trasy tych rozwiązań często się powtarzają, lub czasami kolejność występujących w nich miast jest taka sama, ale trasy te rozpoczynają się od innych punktów albo podążają w przeciwnym kierunku. Można by zatem stwierdzić, że jest to jedno i to samo rozwiązanie, a nie kilkanaście innych najlepszych rozwiązań. Jednak warto pokazać wiele tych rozwiązań, choćby dla przedstawienia skali, czyli ile razy na 480 (96 kombinacji \* 5 powtórzeń) dostać można takowy najkrótszy dystans. Każdy z 19 najlepszych wyników powstał wskutek wykonania algorytmu za pomocą parametru „inverse” określającego rodzaj przeszukiwania sąsiedztwa. Sam czas wykonania algorytmu dla poszczególnych kombinacji z jednej strony pokazuje różnicę między długością wykonywania kodu dla 1000 a dla 5000 iteracji, jednak warto pamiętać, że na tę prędkość składa się wiele czynników, na które programista kodu nie zawsze ma wpływ, jak parametry używanych do obliczeń komputerów, procesy działające (lub nie) w tle, czy wpływ na ten czas podłączenia (lub nie) urządzenia do źródła energii.

10 najgorszych wyników:

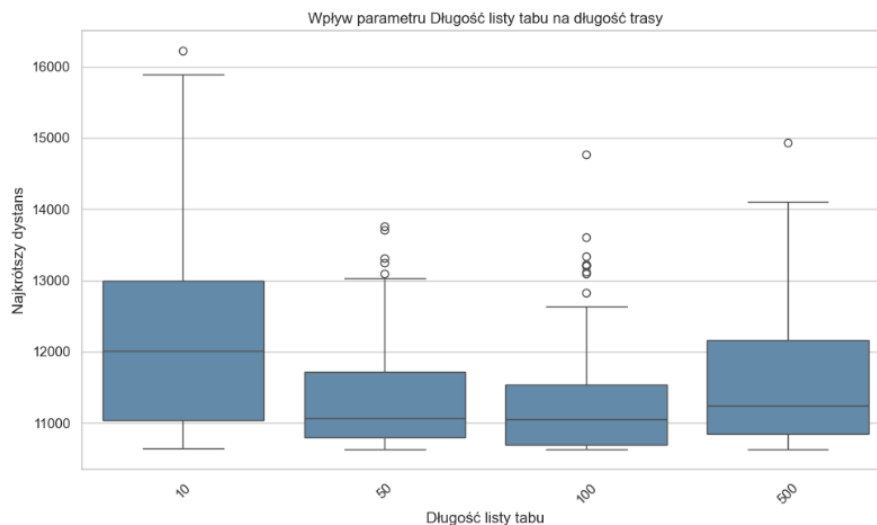
Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Najkrótszy dystans	Czas (milisekundy)
10	100	swap	liczba iteracji	16222	126,4005
10	1000	swap	liczba iteracji bez poprawy	15889	420,4272
10	100	swap	liczba iteracji	15737	194,4058
10	1000	insert	liczba iteracji bez poprawy	15511	527,5405
10	5000	swap	liczba iteracji bez poprawy	15389	2166,0201
10	500	insert	liczba iteracji	15205	284,3644
500	100	swap	liczba iteracji	14942	108,7279
10	500	swap	liczba iteracji	14934	503,6146
10	1000	swap	liczba iteracji	14916	466,1241
100	100	swap	liczba iteracji	14773	212,3672



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

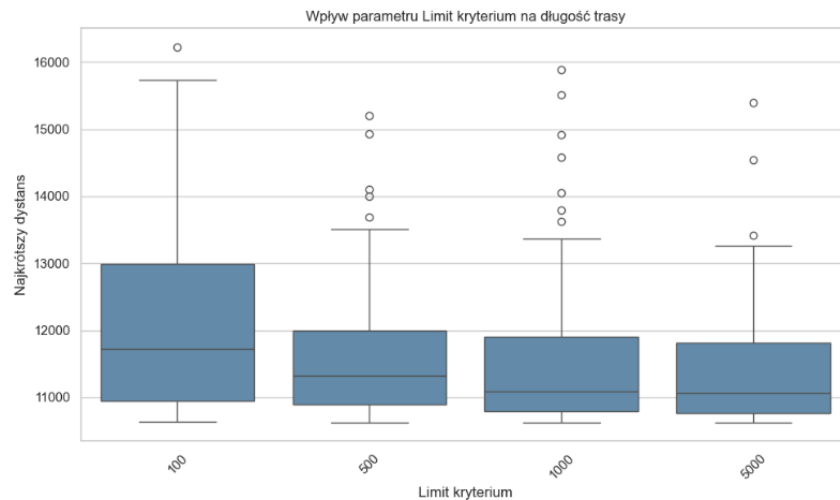
Najbardziej rzucającym się w oczy wnioskiem wynikającym z tabeli najgorszych rozwiązań jest każdorazowe wystąpienie rodzaju przeszukiwania sąsiedztwa typu swap. Najczęściej to również zbyt krótka lista tabu przyczynia się do tak słabego wyniku.

### Wykresy pudełkowe wpływu kolejnych parametrów na długość trasy:



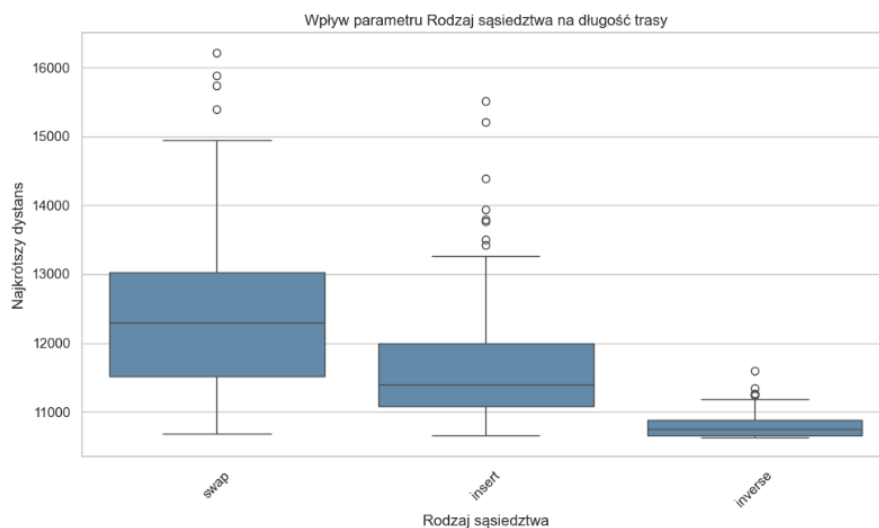
Długość listy tabu	Średnia Najkrótsza Trasa	Min	Max
10	12259,025	10648	16222
50	11359,41667	10628	13767
100	11331,025	10628	14773
500	11570,575	10628	14942

Wykres pudełkowy oraz parametry w tabeli z najlepszymi wynikami świadczą o tym, że dla 48 danych optymalną długością listy tabu będzie 50 lub 100. Zbyt krótka lista tabu powoduje mniejszą kontrolę nad przeszukiwaniem przestrzeni rozwiązań i może powodować trudności w wychodzeniu z optimów lokalnych. Zbyt długa jak widać też nie wpływa dobrze na skuteczność algorytmu, gdyż być może wiele potencjalnych ruchów zostaje zablokowanych, nawet jeśli mogłyby prowadzić do lepszych rozwiązań.



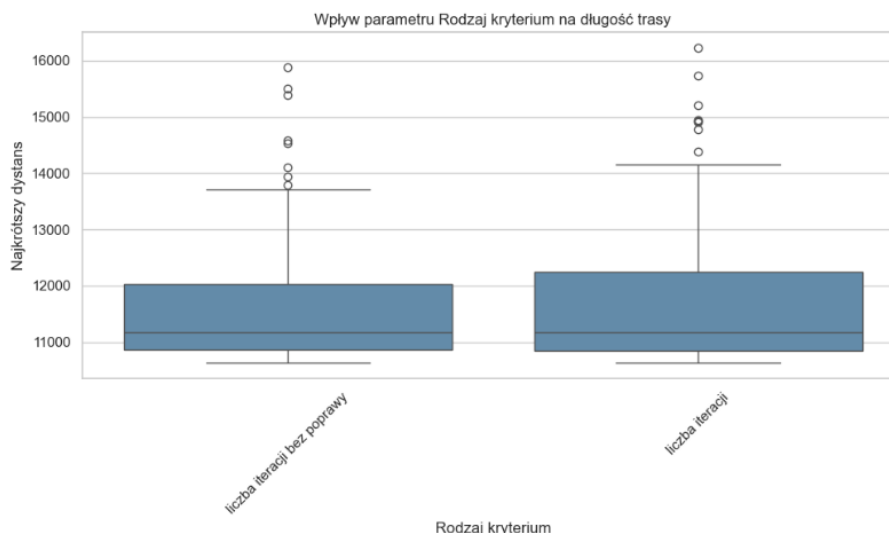
Limit kryterium	Średnia Najkrótsza Trasa	Min	Max
100	12031,66667	10638	16222
500	11601,275	10628	15205
1000	11486,53333	10628	15889
5000	11400,56667	10628	15389

Zgodnie z logiką, im więcej iteracji zostanie wykonanych, tym większe prawdopodobieństwo znalezienia lepszego rozwiązania, gdyż mamy do czynienia z algorytmem, który z zasady radzi sobie z wychodzeniem z optimów lokalnych. Jednak największy przeskok widać przy zmianie ze 100 iteracji do 500, a dalszy wzrost iteracji nie pomniejsza wyników aż tak znacząco. Wydłuża jednak czas działania algorytmu, więc jeśli dla kogoś czas ten jest ważny i 5000 iteracji trwa za długo, może spokojnie ograniczyć wielkość tego parametru.



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	12389,75	10688	16222
insert	11705,79375	10653	15511
inverse	10794,4875	10628	11600

W tym przypadku wyraźnie widać przeskok w wartości związane z zastosowaniem innej metody przeszukiwania sąsiedztwa. Tak samo jak w przypadku innych algorytmów, swap wypada najslabiej, a inverse – najkorzystniej. Można to również wywnioskować z tabel 10 najlepszych i najgorszych wyników.



Rodzaj kryterium	Średnia Najkrótsza Trasa	Min	Max
liczba iteracji	11683,10417	10628	16222
liczba iteracji bez poprawy	11576,91667	10628	15889

W przypadku rodzaju kryterium zakończenia algorytmu, trochę lepsze wyniki daje liczba iteracji bez poprawy, bo po prostu jest to tyle samo lub więcej powtórzeń niż sama liczba iteracji.

#### Najlepsze średnie wyniki dla danych konfiguracji parametrów dla 48 miast:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Średnia Najkrótsza Trasa	Średni Czas Wykonania
100	1000	inverse	liczba iteracji bez poprawy	10639	755,93058
100	5000	inverse	liczba iteracji	10639,2	4466,95182
100	5000	inverse	liczba iteracji bez poprawy	10639,2	4358,96868
500	5000	inverse	liczba iteracji	10646	3152,08654
100	1000	inverse	liczba iteracji	10653	995,1346
50	5000	inverse	liczba iteracji bez poprawy	10667,2	2979,54132
50	5000	inverse	liczba iteracji	10670,8	4740,18202
500	500	inverse	liczba iteracji	10680,2	398,91404
100	500	inverse	liczba iteracji	10690,2	629,6461
500	1000	inverse	liczba iteracji	10695	784,61746

## Liczba miast 76

10 najlepszych wyników:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Najkrótszy dystans	Czas (milisekundy)
50	5000	inverse	liczba iteracji bez poprawy	108159,4383	20005,9333
50	5000	inverse	liczba iteracji bez poprawy	108159,4383	12525,2
50	5000	inverse	liczba iteracji bez poprawy	108159,4383	12100,2878
100	1000	inverse	liczba iteracji bez poprawy	108159,4383	3127,8866
100	5000	inverse	liczba iteracji bez poprawy	108159,4383	10125,5032
100	5000	inverse	liczba iteracji	108159,4383	10339,9436
50	100	inverse	liczba iteracji	108159,4383	640,2317
50	500	inverse	liczba iteracji	108159,4383	1592,0218
50	1000	inverse	liczba iteracji	108159,4383	4943,7606
50	1000	inverse	liczba iteracji	108159,4383	2390,1743

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

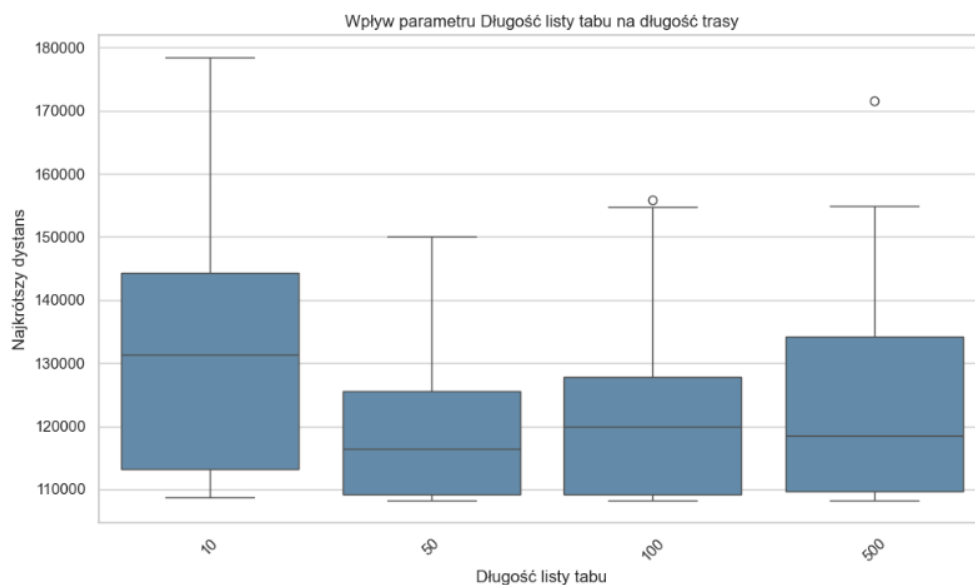
Tutaj sytuacja jest podobna, jak w przypadku najlepszych wyników dla 48 danych – najkrótsza trasa występuje więcej niż 10 razy (konkretnie są to 23 wyniki), lecz niektóre trasy wśród nich się powtarzają. Ponownie inverse jest najefektywniejszym typem przeszukiwania sąsiedztwa. Wynik jest korzystniejszy niż ten z solvera.

10 najgorszych wyników:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Najkrótszy dystans	Czas (milisekundy)
10	5000	swap	liczba iteracji bez poprawy	178419,2877	9988,8967
10	1000	swap	liczba iteracji bez poprawy	174372,2681	2056,6453
500	100	swap	liczba iteracji	171578,7784	501,4734
10	100	swap	liczba iteracji bez poprawy	170919,174	645,4501
10	500	swap	liczba iteracji bez poprawy	170098,6247	1233,2062
10	500	swap	liczba iteracji	168015,4461	1431,338
10	1000	swap	liczba iteracji	165149,2168	2208,2177
10	5000	swap	liczba iteracji bez poprawy	161162,9917	11879,1564
10	500	swap	liczba iteracji	160158,5386	1568,9086
10	5000	swap	liczba iteracji bez poprawy	159671,5537	9909,1963

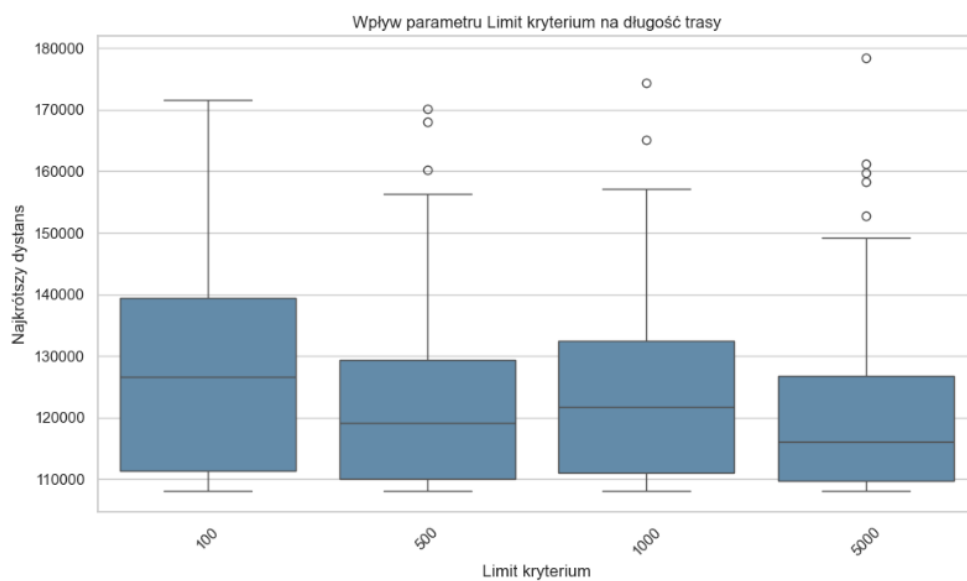
Trasa najgorszego rozwiązania: 23, 22, 21, 25, 43, 42, 53, 67, 70, 68, 69, 47, 45, 46, 24, 26, 27, 49, 50, 51, 56, 58, 59, 60, 33, 32, 29, 30, 76, 75, 3, 4, 20, 31, 34, 40, 41, 61, 57, 55, 52, 44, 48, 66, 65, 54, 28, 12, 13, 14, 74, 9, 10, 19, 35, 39, 38, 36, 37, 18, 17, 11, 15, 16, 62, 63, 64, 73, 72,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

Wnioski dotyczące najgorszych wyników wśród 48 miast można powtórzyć dla 76 miast.



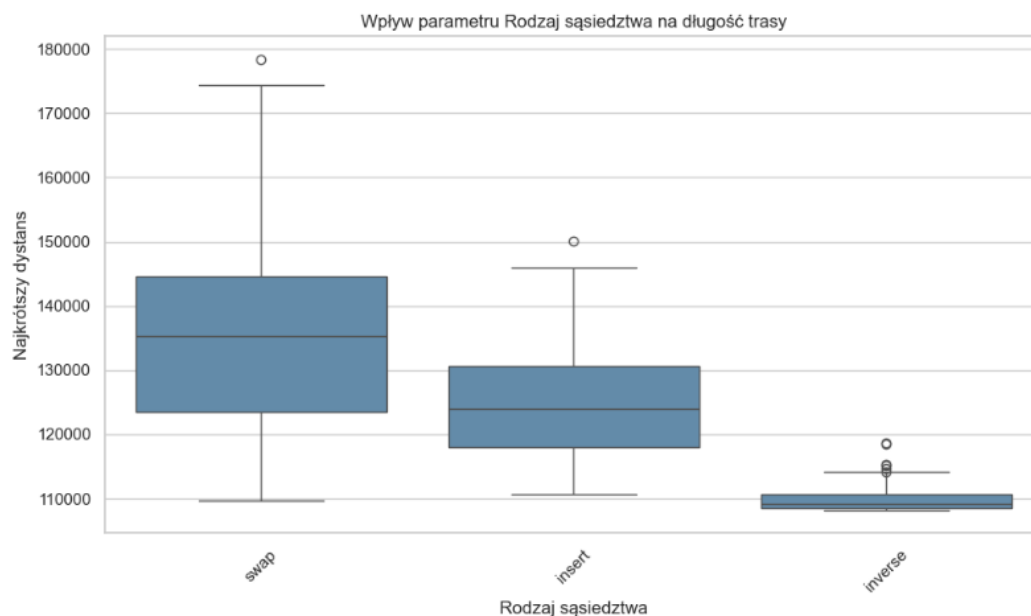
Długość listy tabu	Średnia Najkrótsza Trasa	Min	Max
10	131942,6588	108724,24	178419,3
50	119047,5998	108159,44	150125,9
100	120641,87	108159,44	155773,9
500	122618,4126	108159,44	171578,8

Tak samo, zbyt krótka i zbyt długa kadencja tabu źle służy długości trasy.



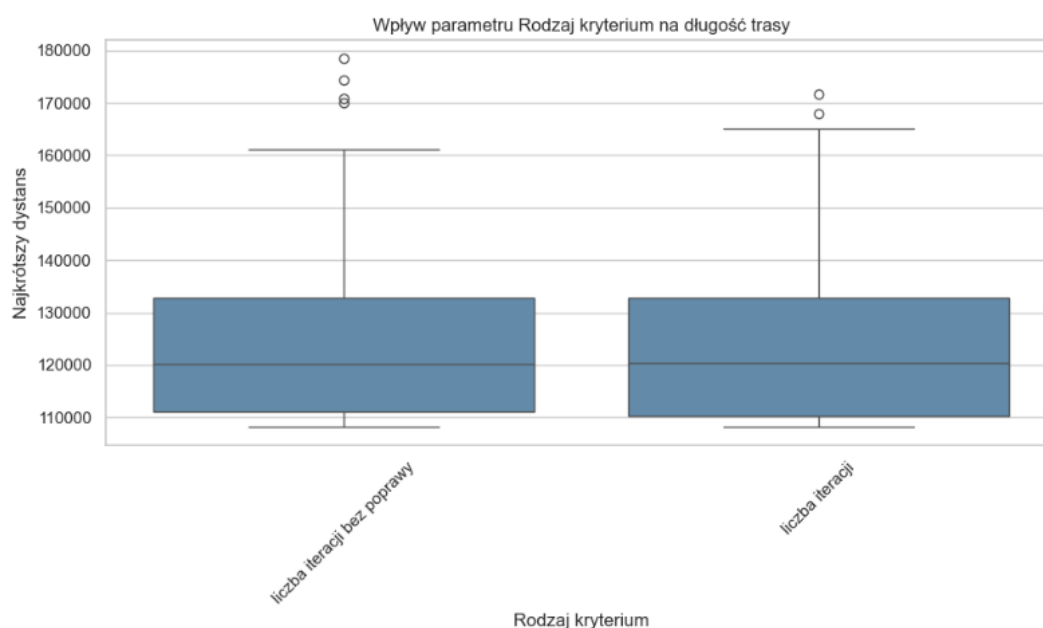
Limit kryterium	Średnia Najkrótsza Trasa	Min	Max
100	127556,919	108159,44	171578,8
500	122377,7889	108159,44	170098,6
1000	123494,369	108159,44	174372,3
5000	123494,369	108159,44	174372,3

Na powyższym wykresie już trudniej może być widoczny spadek długości najkrótszej znalezionej trasy. Jednak tak samo najbardziej odznacza się słaba skuteczność najkrótszej listy tabu.



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	135770,8327	109683,48	178419,3
insert	125025,4713	110672,57	150125,9
inverse	109891,6019	108159,44	118737,6

Niezmiennie, wybór typu insert znacząco poprawia wynik algorytmu.



Rodzaj kryterium	Średnia Najkrótsza Trasa	Min	Max
liczba iteracji	123532,5834	108159,44	171578,8
liczba iteracji bez poprawy	123592,6871	108159,44	178419,3

Oba rodzaje kryterium wpływają podobnie na długość trasy.

#### Najlepsze średnie wyniki dla danych konfiguracji parametrów dla 76 miast:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Średnia Najkrótsza Trasa	Średni Czas Wykonania
100	5000	inverse	liczba iteracji	108337,4121	15334,55758
100	1000	inverse	liczba iteracji bez poprawy	108396,6636	3964,04538
100	1000	inverse	liczba iteracji	108402,8502	3814,39008
50	1000	inverse	liczba iteracji	108482,9307	3604,74054
100	5000	inverse	liczba iteracji bez poprawy	108529,855	14558,2102
50	5000	inverse	liczba iteracji bez poprawy	108567,6869	13943,4034
50	500	inverse	liczba iteracji	108653,8246	2072,74904
50	1000	inverse	liczba iteracji bez poprawy	108708,5858	2903,96988
50	5000	inverse	liczba iteracji	108878,1665	13244,83674
500	500	inverse	liczba iteracji bez poprawy	108910,9974	1453,67836

#### Liczba miast 127

10 najlepszych wyników:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Najkrótszy dystans	Czas (milisekundy)
50	100	inverse	liczba iteracji bez poprawy	118293,5238	2970,7251
500	5000	inverse	liczba iteracji	118370,8283	119435,871
500	5000	inverse	liczba iteracji	118479,2898	53623,75
100	1000	inverse	liczba iteracji bez poprawy	118513,2041	16711,3594
500	500	inverse	liczba iteracji bez poprawy	118651,3444	7072,8254
500	1000	inverse	liczba iteracji bez poprawy	118673,8969	13362,4198
100	500	inverse	liczba iteracji bez poprawy	118712,949	8423,4596
100	100	inverse	liczba iteracji	118736,3118	3770,6132
500	1000	inverse	liczba iteracji	118775,2623	16558,1977
500	1000	inverse	liczba iteracji bez poprawy	118780,0295	15754,3025

Trasa najlepszego rozwiązania: 103, 45, 54, 57, 121, 56, 124, 52, 5, 50, 115, 13, 120, 10, 100, 64, 58, 91, 61, 62, 59, 60, 116, 90, 3, 11, 9, 24, 23, 4, 22, 19, 72, 8, 67, 73, 74, 68, 71, 70, 69,

75, 76, 78, 117, 84, 81, 126, 82, 83, 101, 102, 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, 95, 123, 97, 98, 32, 29, 28, 122, 33, 25, 26, 38, 39, 42, 34, 43, 40, 35, 37, 36, 41, 14, 12, 30, 27, 31, 80, 79, 77, 18, 21, 17, 20, 108, 15, 106, 6, 114, 105, 7, 1, 16, 2, 51, 44

W przypadku 127 danych tylko raz udało się uzyskać najkrótszą trasę, a jej długość jest lepsza niż ta z solvera. Udało się to przy pomocy takich parametrów jak długość listy tabu wynosząca 50 (jako jedyny przypadek wśród wymienionych 10 pierwszych miejsc) i, co ciekawe, przy zastosowaniu tylko 100 iteracji z rzędu bez poprawy – to znaczy, że albo trasa startowa wylosowała się bardzo korzystnie, albo następowało sporo popraw wyników i iteracje bez poprawy często się zerowały. Poza tym, tak jak zwykle, typ inverse jest częścią każdego z najlepszych wyników.

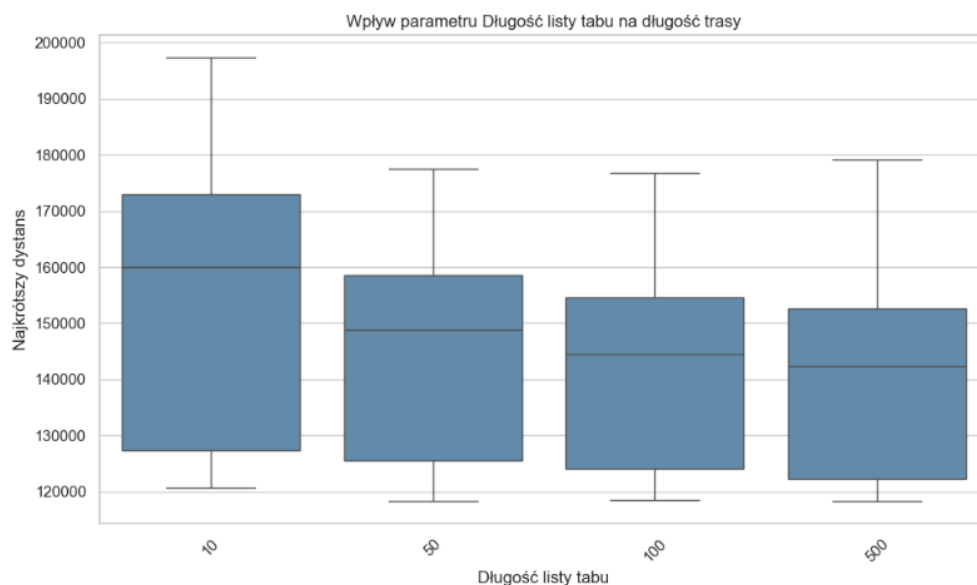
10 najgorszych wyników:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Najkrótszy dystans	Czas (milisekundy)
10	1000	swap	liczba iteracji bez poprawy	197271,0515	10296,1146
10	500	swap	liczba iteracji	190767,6476	7170,2194
10	1000	swap	liczba iteracji	190418,4327	10206,3848
10	5000	swap	liczba iteracji	190418,373	44581,0497
10	1000	swap	liczba iteracji	190142,5089	12101,2792
10	5000	swap	liczba iteracji bez poprawy	189990,8639	44162,4436
10	5000	swap	liczba iteracji	186939,5103	53343,1918
10	100	swap	liczba iteracji bez poprawy	185581,6783	2810,0825
10	500	swap	liczba iteracji	184701,7142	7157,3481
10	5000	swap	liczba iteracji bez poprawy	183756,8852	46233,557

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

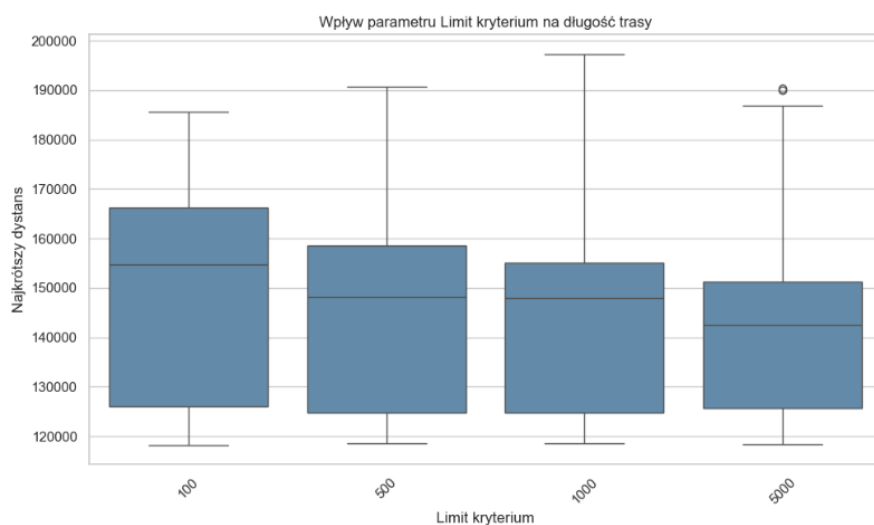
Wszystkie najgorsze wyniki mają związek z ruchem typu swap oraz ze zbyt krótką listą tabu, nawet pomimo 1000 czy 5000 iteracji lub iteracji bez poprawy.





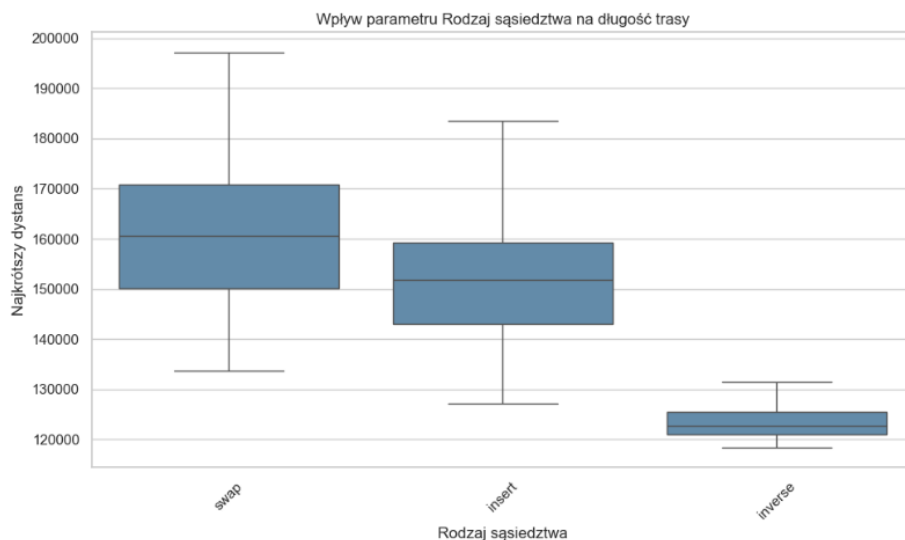
Długość listy tabu	Średnia Najkrótsza Trasa	Min	Max
10	154109,9859	120643,2094	197271,0515
50	144712,1849	118293,5238	177532,5677
100	142093,3141	118513,2041	176730,0212
500	140514,2726	118370,8283	179108,0337

Jak można zauważyć i porównać wykres z tymi dla poprzednich danych, 10 to wciąż zbyt krótka kadencja tabu, ale za to 500 nabiera większego sensu dla większej ilości miast, gdyż opcji wykonania ruchu w celu przeszukania sąsiedztwa jest już dużo więcej.



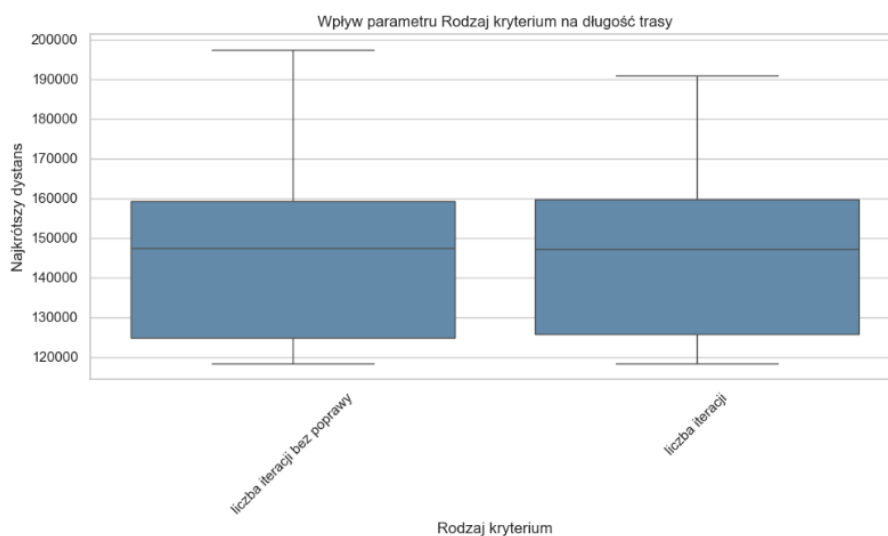
Limit kryterium	Średnia Najkrótsza Trasa	Min	Max
100	149356,0088	118293,5238	185581,6783
500	145240,1446	118651,3444	190767,6476
1000	144489,1857	118513,2041	197271,0515
5000	142344,4184	118370,8283	190418,3730

Im więcej iteracji, tym generalnie lepiej dla wyników algorytmu. Tu jedynie przeskok między 100 a kolejnymi wartościami tego parametru nie jest aż tak duży.



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	160927,7001	133722,5522	197271,0515
insert	151935,3343	127171,7775	183558,1866
inverse	123209,2837	118293,5238	131466,1262

Niezmiennie inverse wygrywa w skuteczności z pozostałymi ruchami, a swap jest najgłupszy.



Rodzaj kryterium	Średnia Najkrótsza Trasa	Min	Max
liczba iteracji	145602,5068	118370,8283	190767,6476
liczba iteracji bez poprawy	145112,3720	118293,5238	197271,0515

Dla tych miast również wybór tego kryterium nie robi szczególnie dużej różnicy.

### Najlepsze średnie wyniki dla danych konfiguracji parametrów dla 127 miast:

Długość listy tabu	Limit kryterium	Rodzaj sąsiedztwa	Rodzaj kryterium	Średnia Najkrótsza Trasa	Średni Czas Wykonania
500	5000	inverse	liczba iteracji bez poprawy	120143,734	66776,24254
500	5000	inverse	liczba iteracji	120466,5732	78562,11638
500	1000	inverse	liczba iteracji bez poprawy	120605,5765	19430,46034
500	500	inverse	liczba iteracji bez poprawy	120629,8291	8164,07344
500	1000	inverse	liczba iteracji	120926,315	13780,11888
100	1000	inverse	liczba iteracji bez poprawy	121383,9377	22378,24624
100	5000	inverse	liczba iteracji bez poprawy	121508,1462	89610,60378
100	500	inverse	liczba iteracji bez poprawy	121731,059	8673,96428
500	500	inverse	liczba iteracji	122019,1744	8187,87586
50	1000	inverse	liczba iteracji bez poprawy	122333,245	14562,19798

W przypadku podsumowań 10 najlepszych średnich wyników dla trzech różnych ilości miast, można zauważyć pewne schematy – brakuje w nich najkrótszej, wynoszącej 10, długości listy tabu, nie występuje limit kryterium iteracji o wartości najniższej, czyli 100 oraz każda kombinacja przeszukuje sąsiedztwo za pomocą ruchu typu inverse.

## 8. Algorytm kolonii pszczół

Jest to iteracyjny algorytm optymalizacji odwzorowujący zachowanie pszczół podczas poszukiwania pokarmu. Polega na podziale roju pszczół na kilka grup i przydzielenie każdej z nich konkretnych zadań. Jedno przejście procesu można streścić w kilku krokach:

- 1) Inicjalizacja populacji początkowej.
- 2) Ocena jakości znalezionych rozwiązań.
- 3) Eksploracja najlepszych rozwiązań.
- 4) Wybór elitarnych rozwiązań spośród najlepszych przeszukiwanych.
- 5) Eksploracja przestrzeni wyznaczonymi osobnikami.
- 6) Ocena jakości znalezionych rozwiązań.
- 7) Powrót do pkt. 3) lub zakończenie.

Krótki opis kluczowych parametrów:

- Liczba pszczół zwiadowczych – ilość losowych kombinacji tras generowanych początku algorytmu, co pozwala na szeroką eksplorację przestrzeni rozwiązań; każda pszczoła zwiadowcza reprezentuje jedno potencjalne rozwiązanie problemu komiwojażera (układ miast), który pszczoła "bada", obliczając długość trasy dla tej kolejności miast.
- Liczba najlepszych stanowisk – ilość miejsc o najlepszej jakości spośród wszystkich

ocenionych przez pszczoły zwiadowcze.

- Liczba elitarnych stanowisk – ilość miejsc najlepszych spośród wcześniej już wybranych najlepszych stanowisk.
- Liczba pszczoł najlepszych – ilość pszczoł przypisana do przeszukiwania każdego z najlepszych (ale jeszcze nie elitarnych) stanowisk w każdej iteracji algorytmu.
- Liczba pszczoł elitarnych – ilość pszczoł przypisanych do przeszukiwania każdego z elitarnych stanowisk w każdej iteracji algorytmu.

Badane wartości parametrów:

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa
2	4	4	2	swap
5	10	10	5	insert
15	30	30	15	inverse
25	50	50	25	

Pozostałe stałe parametry:

- Liczba pszczoł zwiadowczych = 500
- Maksymalna liczba iteracji = 1000
- Rodzaj kryterium = „liczba iteracji”

Z przyjętych wartości parametrów wygenerowano 768 różnych kombinacji, z których każda została powtórzona 2 razy.

### Liczba miast 48

10 najlepszych wyników:

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Najkrótszy dystans	Czas (milisekundy)
2	10	50	15	inverse	10628	2251,5622
2	10	50	25	insert	10628	1654,7136
2	30	30	25	inverse	10628	991,8455
5	30	50	25	inverse	10628	1334,1257
5	50	30	15	inverse	10628	1141,301
5	50	30	15	inverse	10628	1189,322
5	50	50	2	inverse	10628	1561,0815
15	4	30	15	inverse	10628	678,712
15	30	30	25	inverse	10628	1722,1367
15	50	30	2	inverse	10628	802,1442

Trasa pierwszego z najlepszych rozwiązań: 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, 43, 17, 27, 19, 37, 6, 28, 7, 18, 44, 31, 38, 8, 1, 9, 40, 15, 12, 11

W wynikach algorytmu pszczelego znaleziono 12 najkrótszych tras o wartości 10628 (mniejszej niż najlepszy wynik solvera), gdzie również połączenia kolejnych miast często się

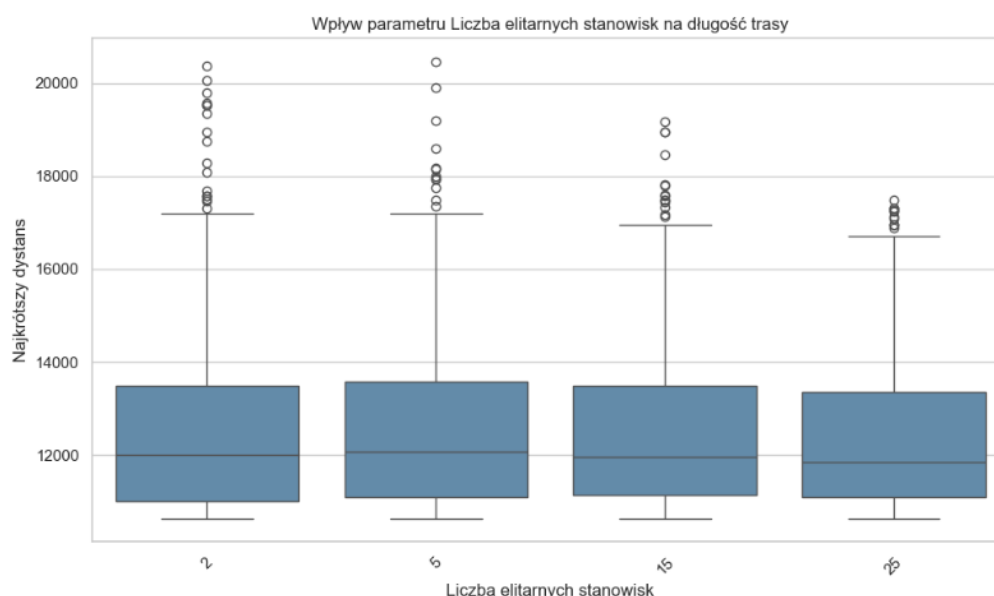
powtarzają. Wyjątkowo wśród tych kombinacji wystąpiła jedna, której typ ruchu to nie inverse, a insert. Dominują tu także kombinacje, w których występuje mała liczba elitarnych stanowisk – tylko nieliczne miejsca trafiają na listę elitarnych i są bardzo dokładnie przeszukiwane przez pszczoły wyznaczone do tego zadania, co ostatecznie daje świetne wyniki. Tylko raz w tabeli wystąpił przypadek, gdy najlepszych stanowisk jest mniej niż elitarnych.

10 najgorszych wyników:

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Najkrótszy dystans	Czas (milisekundy)
5	4	4	2	swap	20469	4029,0265
2	50	4	2	swap	20360	3376,9588
2	10	4	2	swap	20054	3687,463
5	10	4	2	swap	19901	3999,5595
2	30	4	2	swap	19785	3695,1978
2	50	4	2	swap	19563	3437,4955
2	4	4	2	swap	19528	2191,1018
2	4	4	2	swap	19518	2640,122
2	30	4	2	swap	19345	3833,305
5	30	4	2	swap	19189	3698,6377

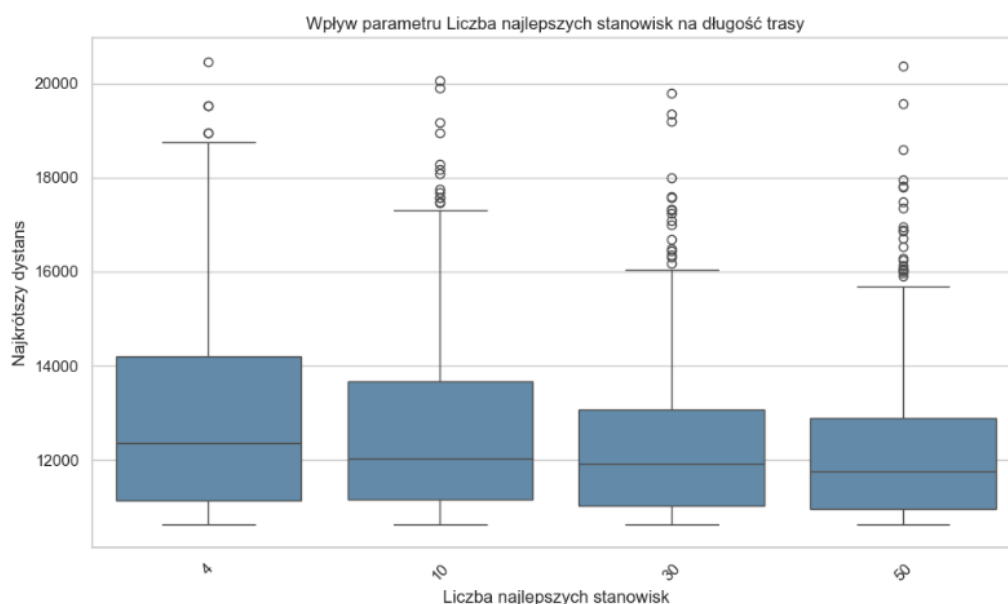
Trasa najgorszego rozwiązania: 3, 23, 13, 11, 9, 7, 44, 30, 18, 6, 43, 28, 46, 41, 29, 24, 10, 35, 45, 48, 32, 20, 47, 12, 40, 38, 22, 16, 31, 17, 19, 27, 37, 36, 33, 15, 14, 5, 2, 42, 4, 26, 39, 25, 21, 34, 1, 8

Oczywistym wnioskiem wynikającym z rankingu najgorszych wyników jest fakt, że zbyt mała ilość pszczoł przeszukujących elitarne i najlepsze stanowiska do niczego nie prowadzi. Najgorsze wyniki są bardzo wysokie w stosunku do najniższych występujących w tej pierwszej tabeli. Również swap jako typ ruchu przyczynia się do tak słabych rezultatów.



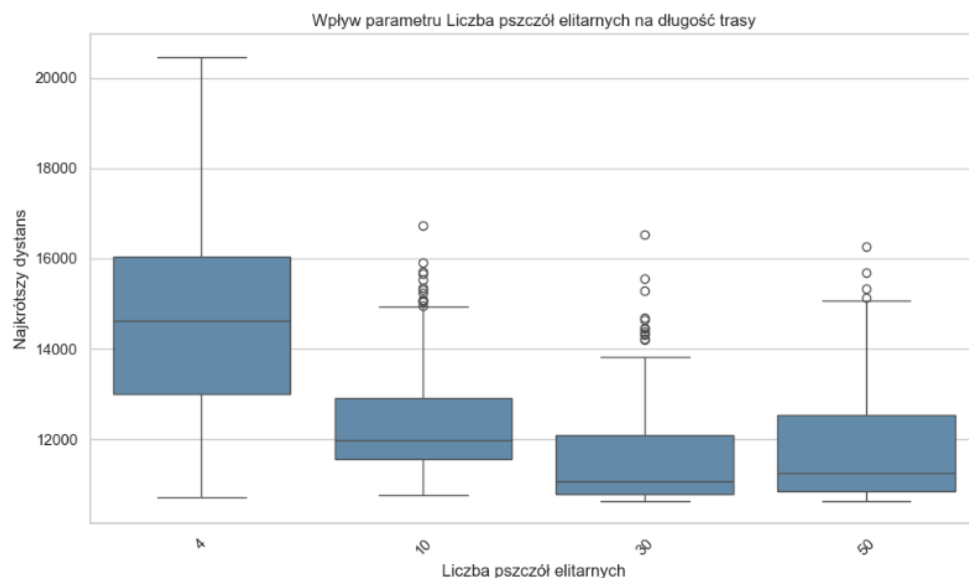
Liczba elitarnych stanowisk	Średnia Najkrótsza Trasa	Min	Max
2	12605,86979	10628	20360
5	12627,91927	10628	20469
15	12564,47396	10628	19163
25	12483,27604	10628	17480

Pomimo występowania głównie niskich wartości parametru liczby elitarnych stanowisk w tabeli najlepszych wyników, wykres wskazuje na to, że wartość tego parametru nie jest jednak najistotniejsza dla długości trasy.



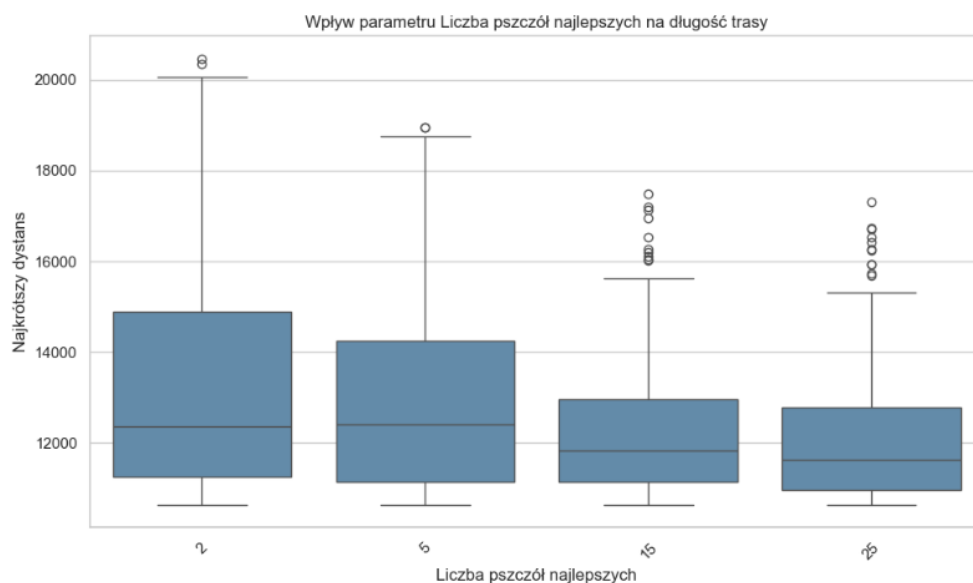
Liczba najlepszych stanowisk	Średnia Najkrótsza Trasa	Min	Max
4	12926,55469	10628	20469
10	12689,23698	10628	20054
30	12404,66667	10628	19785
50	12261,08073	10628	20360

Większa liczba najlepszych stanowisk ewidentnie pomaga polepszeniu wyników, ponieważ wśród najlepszych stanowisk wyszukiwane są stanowiska elitarne, które później są dokładnie zgłębiane.



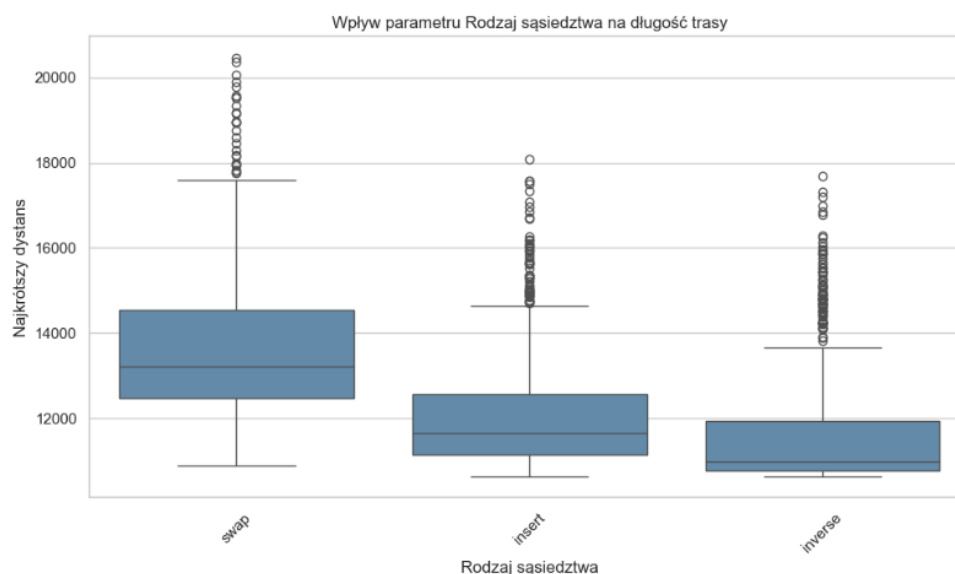
Liczba pszczół elitarnych	Średnia Najkrótsza Trasa	Min	Max
4	14641,17708	10708	20469
10	12311,32813	10757	16738
30	11543,27083	10628	16528
50	11785,76302	10628	16264

Widoczne jest znaczne polepszenie wyników wraz ze wzrostem liczby pszczół elitarnych, jednak dla 48 miast, zbyt duża ich ilość może być już trochę mniej korzystnym wyjściem.



Liczba pszczół najlepszych	Średnia Najkrótsza Trasa	Min	Max
2	13182,89323	10628	20469
5	12861,29948	10638	18958
15	12218,89323	10628	17490
25	12018,45313	10628	17315

Duża ilość „najlepszych pszczół” sprzyja lepszemu przeszukiwaniu najlepszych stanowisk, więc i lepszemu wyborowi stanowisk elitarnych.



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	13773,14063	10890	20469
insert	12175,18359	10628	18075
inverse	11762,83008	10628	17687

Tak samo jak w przypadku innych algorytmów swap to najmniej skuteczne wyjście. Pozostałe dwa mają między swoimi wykresami pudełkowymi mniejsze różnice, lecz i tak ciągle przoduje podejście typu inverse.

**Najlepsze średnie wyniki dla danych konfiguracji parametrów dla 48 miast:**

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczół elitarnych	Liczba pszczół najlepszych	Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Średni Czas Wykonania
5	50	30	15	inverse	10628	1165,3115
2	30	30	25	inverse	10638	1081,83095
15	30	30	25	inverse	10638	1671,65795
15	50	30	5	inverse	10643	1002,8377
2	30	50	15	inverse	10650,5	1072,55225
25	4	30	15	inverse	10650,5	1065,36385
25	4	30	25	inverse	10650,5	1127,44895
5	50	50	2	inverse	10658	1547,10575



2	50	50	25	inverse	10668,5	1805,1483
25	50	30	15	inverse	10668,5	2068,77225

### Liczba miast 76

10 najlepszych wyników:

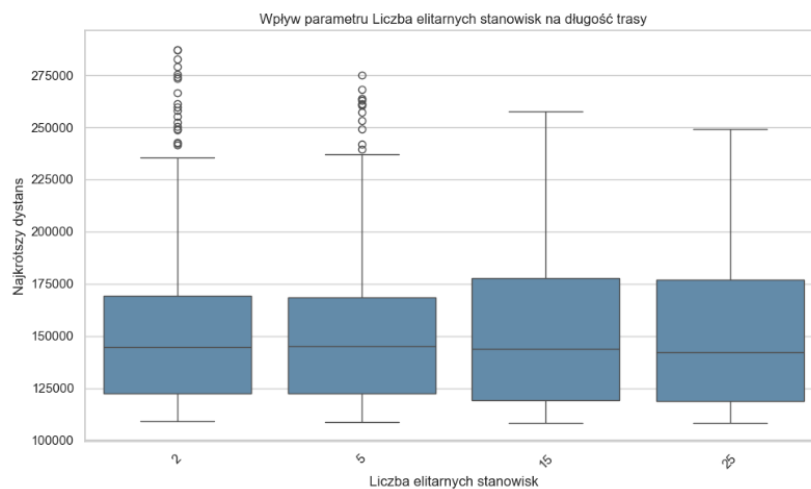
Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Najkrótszy dystans	Czas (milisekundy)
25	30	50	25	inverse	108226,1428	5127,5854
25	4	50	5	inverse	108237,5917	2997,8961
15	4	50	15	inverse	108263,9534	1739,3818
25	30	50	2	inverse	108306,9844	3244,2012
25	10	50	15	inverse	108307,8898	3826,4177
25	10	50	15	inverse	108307,8898	3400,3965
25	10	50	25	inverse	108308,5014	3772,1281
25	50	50	5	inverse	108312,7369	4472,634
25	50	50	15	inverse	108332,9335	5246,8099
25	4	50	25	inverse	108351,22	3204,2857

Dla 76 miast już tylko raz udało się uzyskać najlepszy wynik. Mimo to, różnice między dystansami kolejnych miejsc w rankingu są niewielkie, a każda długość trasy jest krótsza, niż ta z solvera. Proporcje pszczoł do stanowisk są dla najlepszej kombinacji poprawne. Podejście typu inverse jest jedynym występującym w tabeli. Warto też zauważyć, że tutaj (inaczej niż w tabeli najlepszych kombinacji dla 48 miast) występują wyższe wartości parametru liczby elitarnych stanowisk.

10 najgorszych wyników:

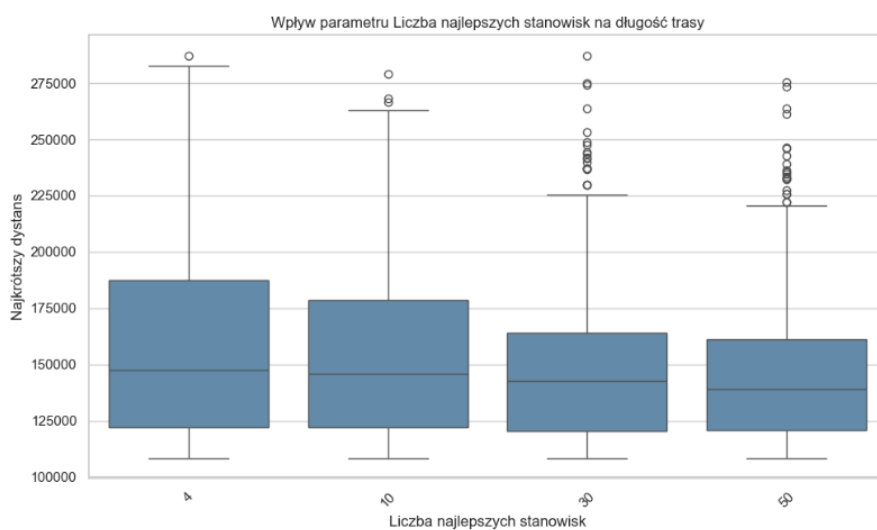
Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Najkrótszy dystans	Czas (milisekundy)
2	30	4	2	swap	287332,0424	6840,816
2	4	4	2	swap	287140,927	6740,8996
2	4	4	2	swap	282834,6277	6757,3834
2	10	4	2	swap	279046,1695	6787,4135
2	50	4	2	swap	275644,2008	6477,0973
5	30	4	2	swap	274871,3519	7064,6129
2	30	4	2	swap	274333,3852	7009,4049
2	50	4	2	swap	273380,4427	6852,8769
5	10	4	2	swap	268278,15	8533,9278
2	10	4	2	swap	266715,0123	6661,4736

Swap ciągle wypada jako najgorsza opcja, poza tym okazuje się, że w skali tylu miast, tak mało pszczoł kompletnie nie radzi sobie z wyszukiwaniem najlepszych rozwiązań.



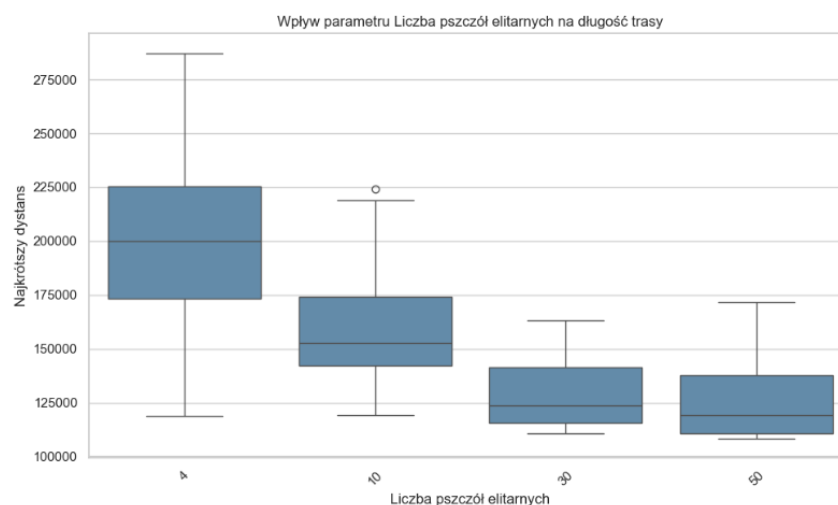
Liczba elitarnych stanowisk	Średnia Najkrótsza Trasa	Min	Max
2	154187,4426	109176,1252	287332,0424
5	152617,2215	108636,4883	274871,3519
15	151737,4016	108263,9534	257569,4498
25	150837,7014	108226,1428	249266,9766

Mniejsza liczba elitarnych stanowisk jest ciągle trochę korzystniejsza. Ważne jednak, żeby w parze z tym szła wystarczająca ilość pszczoł przeszukujących te miejsca.



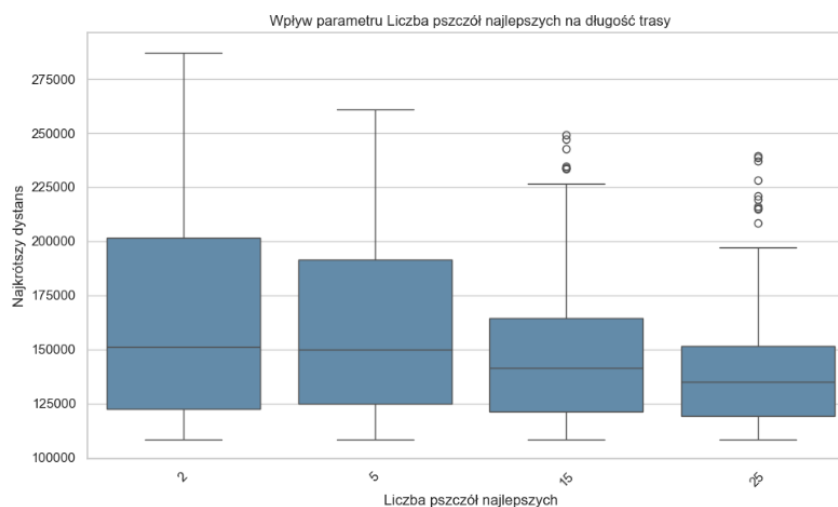
Liczba najlepszych stanowisk	Średnia Najkrótsza Trasa	Min	Max
4	158246,5501	108237,5917	287140,927
10	154434,5572	108307,8898	279046,1695
30	149426,4518	108226,1428	287332,0424
50	147272,208	108312,7369	275644,2008

Dużo najlepszych stanowisk jest pożądane z tego samego powodu, który przytaczany był przy okazji przypadku 48 miast.



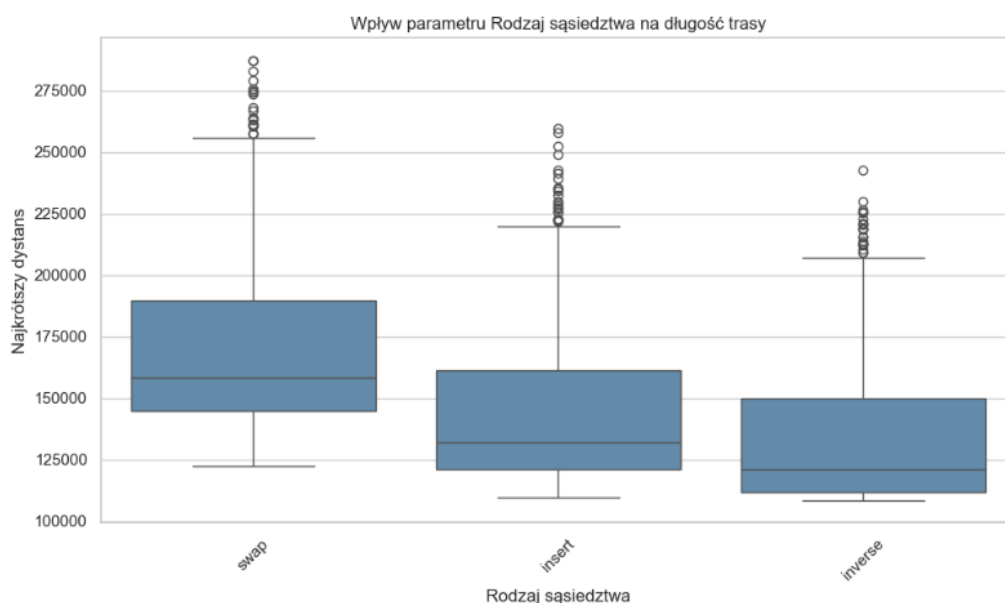
Liczba pszczół elitarnych	Średnia Najkrótsza Trasa	Min	Max
4	198668,9817	118885,8248	287332,0424
10	157742,203	119258,8226	224327,9009
30	127927,0842	110596,8816	163187,5192
50	125041,4982	108226,1428	171834,4757

Wzrost ilości pszczół przeszukujących elitarne stanowiska jest najwyraźniej bardzo ważnym elementem układanki. Daje to pewność jak najdokładniejszego przeszukania miejsc potencjalnie najbardziej obiecujących w oczekiwane minimalne rozwiązania.



Liczba pszczoł najlepszych	Średnia Najkrótsza Trasa	Min	Max
2	164422,6058	108306,9844	287332,0424
5	158872,6797	108237,5917	261120,7919
15	146186,7806	108263,9534	249266,9766
25	139897,7009	108226,1428	239537,2526

Pożądana duża ilość pszczoł wydelegowanych do znajdowania najlepszych stanowisk jest niezmiennie taka sama jak we wnioskach dla 48 miast.



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	173307,1602	122284,6949	287332,0424
insert	146687,9001	109642,8719	259677,0598
inverse	137039,765	108226,1428	242471,5766

W przypadku przeszukiwania sąsiedztwa również nic nie uległo zmianie przy powiększeniu problemu o prawie 30 miast.

**Najlepsze średnie wyniki dla danych konfiguracji parametrów dla 76 miast:**

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Średni Czas Wykonania
25	10	50	15	inverse	108307,9	3613,407
25	30	50	2	inverse	108329,9	3240,66
25	50	50	5	inverse	108455	4158,028
25	50	50	15	inverse	108489,4	5262,584
25	30	50	25	inverse	108711,1	5207,779

25	4	50	5	inverse	108755,3	2911,664
15	10	50	5	inverse	108782,7	1699,974
5	50	50	15	inverse	108799,5	2207,269
25	4	50	15	inverse	108802,7	3149,314
15	50	50	25	inverse	108808	5252,318

### Liczba miast 127

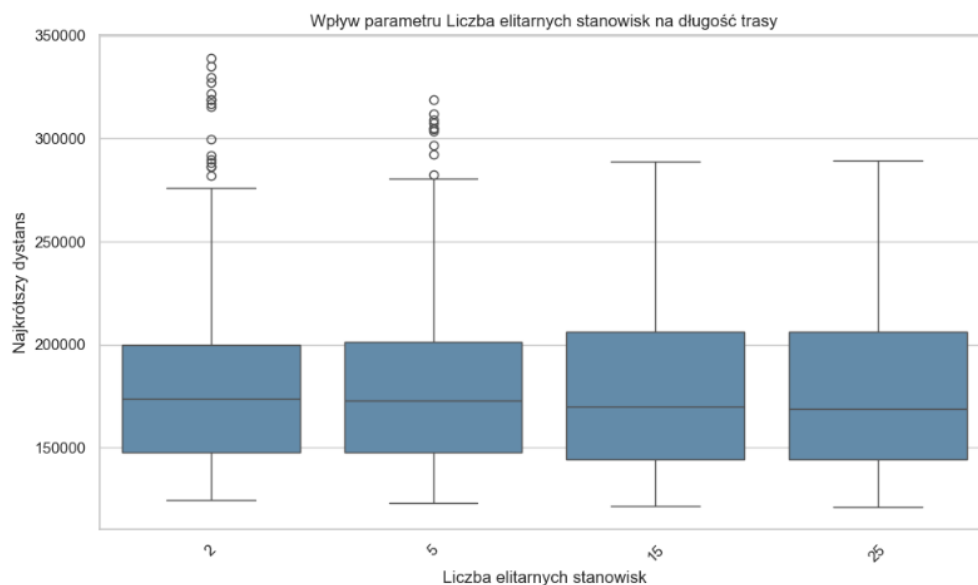
10 najlepszych wyników:

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Najkrótszy dystans	Czas (milisekundy)
25	4	50	25	inverse	121123,795	5795,5423
25	30	50	25	inverse	121341,2805	8603,4104
15	30	50	2	inverse	121837,2378	2227,3445
25	30	50	15	inverse	121864,8966	7328,6102
25	10	50	5	inverse	121937,8034	5379,8393
25	50	50	15	inverse	122013,3032	6845,1032
25	50	50	15	inverse	122158,5426	7491,53
25	4	50	15	inverse	122490,6329	5174,6852
25	4	50	15	inverse	122528,1409	5403,6083
15	50	50	25	inverse	122702,7499	7736,8494

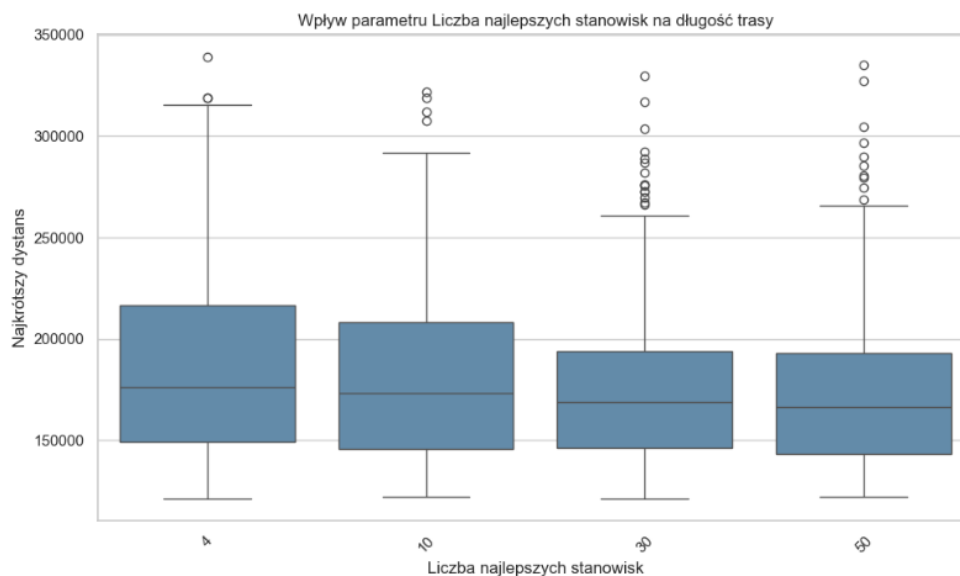
Po przeanalizowaniu najkorzystniejszych kombinacji dla 127 miast można zauważyć, że tak samo jak dla 48 i 76 miast wygrywają rozwiązania, gdzie występuje największa możliwa ilość pszczoł elitarnych, których zadaniem jest jak najdokładniejsze przeszukanie najbardziej obiecujących miejsc. Większość wyników dystansów występujących z tej tabeli jest lepszych niż soolverowe rozwiązanie.

10 najgorszych wyników:

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Najkrótszy dystans	Czas (milisekundy)
2	4	4	2	swap	339249,4762	13397,3958
2	50	4	2	swap	335243,0864	11403,8854
2	30	4	2	swap	329473,312	12062,3418
2	50	4	2	swap	327090,2111	11320,4651
2	10	4	2	swap	321891,2459	13098,61
2	10	4	2	swap	319093,4283	12968,5845
2	4	4	5	swap	319047,9655	13744,4502
5	4	4	2	swap	318996,4774	13125,7569
2	30	4	2	swap	317023,7223	12126,6392
2	4	4	2	swap	315532,2643	13300,058

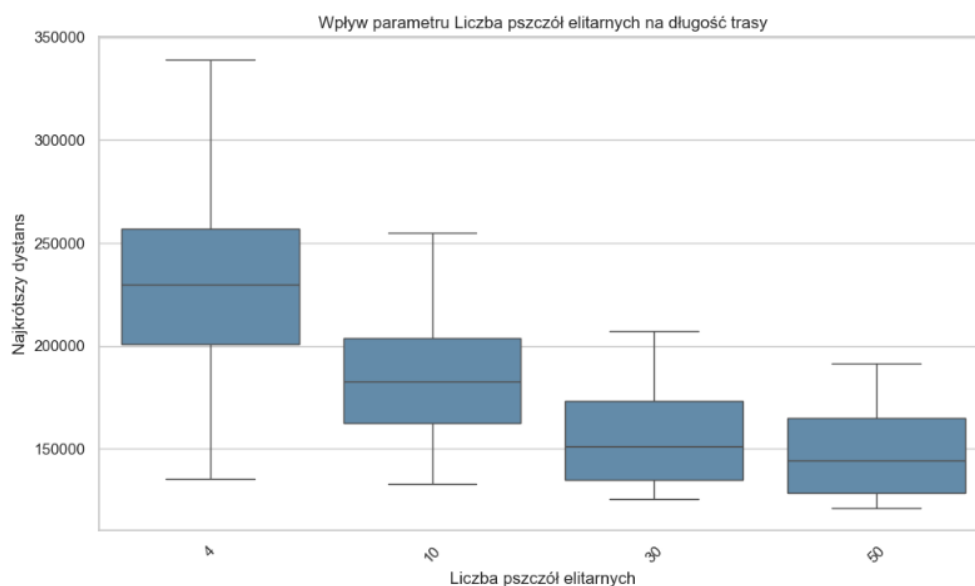


Liczba elitarnych stanowisk	Średnia Najkrótsza Trasa	Min	Max
2	182350,3894	124807,213	339249,4762
5	179295,1148	123278,813	318996,4774
15	177377,7525	121837,238	288634,9389
25	176328,5362	121123,795	289538,3367



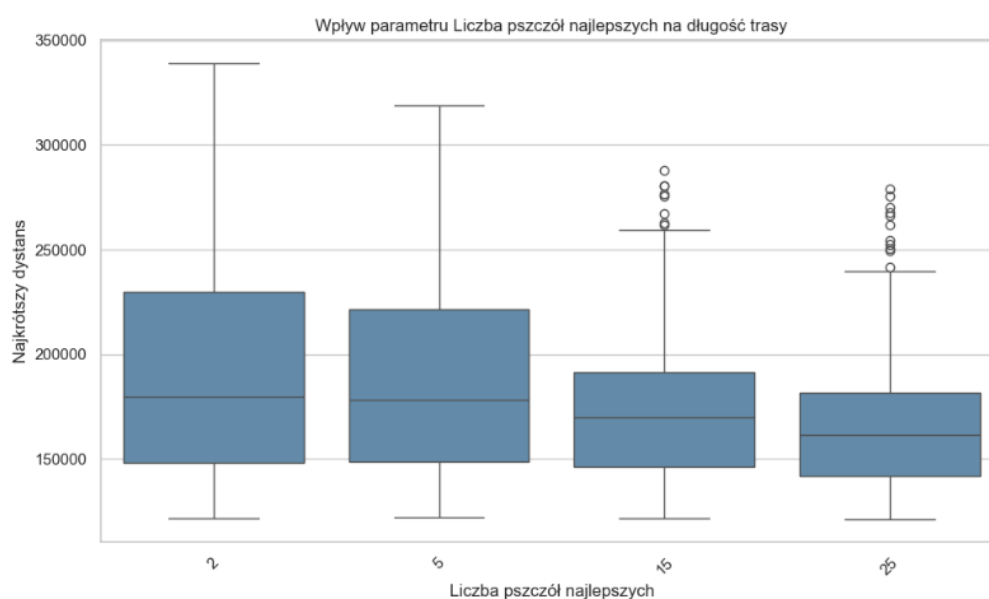
Widać lekki wzrost najlepszej znalezionej długości trasy wraz ze wzrostem ilości elitarnych stanowisk, lecz nie jest on drastyczny.

Liczba najlepszych stanowisk	Średnia Najkrótsza Trasa	Min	Max
4	185860,2697	121123,795	339249,4762
10	180946,0835	121937,803	321891,2459
30	175384,1454	121341,281	329473,312
50	173161,2943	122013,303	335243,0864



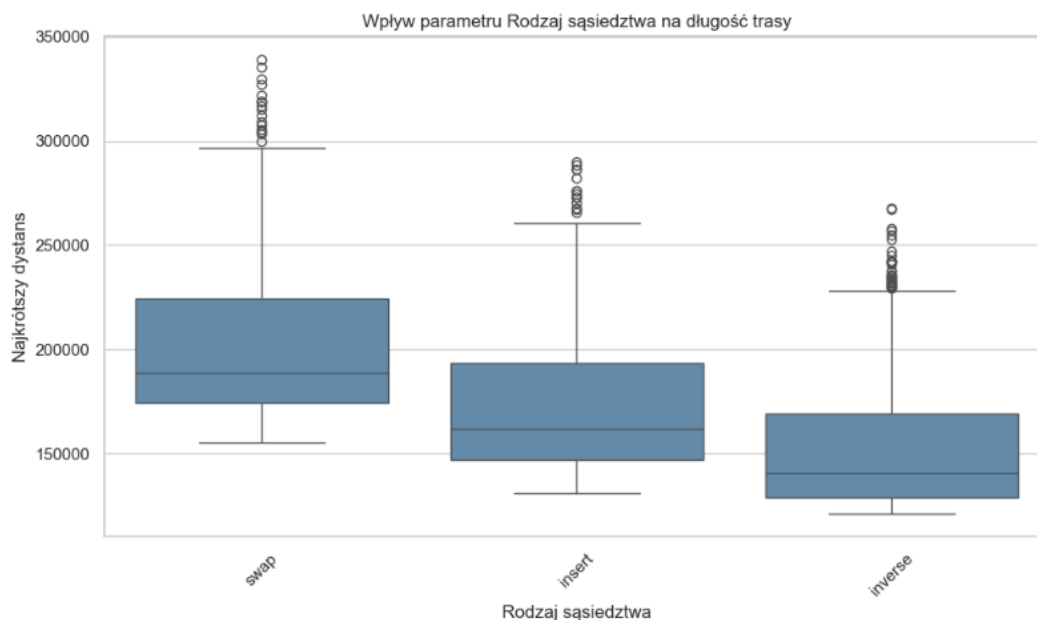
Liczba pszczół elitarnych	Średnia Najkrótsza Trasa	Min	Max
4	228501,4182	135284,195	339249,4762
10	184641,2861	133065,991	255170,9358
30	154723,0755	125435,758	207418,3575
50	147486,0132	121123,795	191391,2426

Ponownie, parametr ten jest widocznie najmocniej wpływającym na skuteczność poszukiwań optimum. Pokrywa się to z wnioskami z tabel rankingów dla wszystkich trzech zbiorów miast.



Liczba pszczół najlepszych	Średnia Najkrótsza Trasa	Min	Max
2	192185,9114	121837,238	339249,4762
5	331166,6487	121937,803	319047,9655
15	172114,0791	121864,897	287692,4797
25	165361,8414	121123,795	279204,8553

Pszczółki najlepsze również są tak ważne, że wzrost ich ilości widocznie wpływa na skrócenie wyniku. W końcu spośród miejsc najlepszych wyszukiwane są miejsca elitarne.



Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Min	Max
swap	205034,0064	155356,629	339249,4762
insert	175950,2827	131022,034	289869,8955
inverse	155529,5556	121123,795	267798,446

Wykres wpływu sposobu przeszukiwania sąsiedztwa nie zaskakuje – wnioski z niego są identyczne, jak w każdym opisywanym algorytmie i dla każdej ilości miast.

**Najlepsze średnie wyniki dla danych konfiguracji parametrów dla 127 miast:**

Liczba elitarnych stanowisk	Liczba najlepszych stanowisk	Liczba pszczoł elitarnych	Liczba pszczoł najlepszych	Rodzaj sąsiedztwa	Średnia Najkrótsza Trasa	Średni Czas Wykonania
25	50	50	15	inverse	122085,9229	7168,3166
25	30	50	25	inverse	122280,0899	8580,18825
25	4	50	15	inverse	122509,3869	5289,14675
25	10	50	5	inverse	122588,7278	5388,02465
25	30	50	15	inverse	122651,5953	7303,6088
25	4	50	25	inverse	122970,8467	5306,92135
15	30	50	2	inverse	123156,0538	2584,5564
15	50	50	2	inverse	123630,047	3297,43875
15	4	50	5	inverse	123735,7058	2389,67795
15	10	50	2	inverse	123787,6357	2782,061



## 9. Podsumowanie

- 48 miast

Algorytm	Najlepszy wynik	Średni wynik
solver	10 648	10 877.5
najbliższego sąsiada	12 012	13 018.04
symulowanego wyżarzania	10 707	13 325.87
wspinaczki z multistartem	10 628	11 574.03
genetyczny	10 648	21 226.897
tabu	10 628	11 630.01
pszczeli	10 628	12 570.38

Dla 48 miast, prawie wszystkie algorytmy osiągnęły podobną długość trasy, co świadczy o ich wysokiej efektywności w znajdowaniu rozwiązań o zbliżonej jakości. Najwyższą wartość osiągnął algorytm Najbliższego Sąsiada (NN), co jest skutkiem prostoty tego algorytmu, który, mimo że szybko generuje rozwiązania, nie zawsze znajduje najkorzystniejsze trasy, zwłaszcza w przypadku bardziej złożonych problemów.

- 76 miast

Algorytm	Najlepszy wynik	Średni wynik
solver	111 228	113 611.533
najbliższego sąsiada	130 921	147 180.1
symulowanego wyżarzania	111 125.437	149 645.3
wspinaczki z multistartem	108 644.973	124 968.4
genetyczny	108 159.438	264 040.425
tabu	108 159.438	123 562.635
pszczeli	108 226.143	152 344.942

Dla 76 miast, najlepsze wyniki uzyskały algorytmy Tabu oraz Genetyczny, które osiągnęły identyczną długość tras. Pozostałe algorytmy nie różnią się znacząco pod względem wyników, natomiast największa różnica w długości tras występuje w przypadku algorytmu Najbliższego Sąsiada (NN), który osiągnął znacznie gorszy wynik niż pozostałe metody.

- 127 miast

Algorytm	Najlepszy wynik	Średni wynik
solver	125 806.44	132 104.429
najbliższego sąsiada	133 970.6	146 175.7
symulowanego wyżarzania	124 721.891	179 517.2
wspinaczki z multistartem	120 640.433	142 399.1
genetyczny	125 582.228	323 683.92
tabu	118 293.524	145 357.439
pszczeli	121 123.795	178 837.948

Dla 127 miast, najlepsze wyniki uzyskały algorytmy Tabu i Symulowanego wyżarzania, które osiągnęły najkrótsze trasy. Pozostałe algorytmy, takie jak Solver, Genetyczny i Wspinaczki z multistartem, również uzyskały dobre wyniki, ale były nieco gorsze pod względem długości tras. Największą różnicę w długości tras ponownie widać w przypadku algorytmu Najbliższego

Sąsiada (NN), który wygenerował znacznie dłuższą trasę w porównaniu do innych metod.

## **10.Bibliografia**

[http://algorytmy.ency.pl/artukul/algorytm\\_najblizszego\\_sasiada](http://algorytmy.ency.pl/artukul/algorytm_najblizszego_sasiada)  
<https://mitu.co.in/wp-content/uploads/2022/04/8.-Hill-Climbing-Algorithm.pdf>  
[https://en.wikipedia.org/wiki/Simulated\\_annealing](https://en.wikipedia.org/wiki/Simulated_annealing)  
[https://pl.wikipedia.org/wiki/Przeszukiwanie\\_tabu](https://pl.wikipedia.org/wiki/Przeszukiwanie_tabu)  
[https://ii.uni.wroc.pl/~prz/2011lato/ah/opracowania/szuk\\_tabu.ppt](https://ii.uni.wroc.pl/~prz/2011lato/ah/opracowania/szuk_tabu.ppt)  
[https://repozytorium.ukw.edu.pl/bitstream/handle/item/3505/Wieslaw%20Popielarski%20Algorytm%20pszczeli%20w%20optymalizacji%20modelu%20przeplywowego%20szeregowania%20zadan.pdf?utm\\_source=chatgpt.com](https://repozytorium.ukw.edu.pl/bitstream/handle/item/3505/Wieslaw%20Popielarski%20Algorytm%20pszczeli%20w%20optymalizacji%20modelu%20przeplywowego%20szeregowania%20zadan.pdf?utm_source=chatgpt.com)