

Programowanie Efektywnych Algorytmów - Projekt: Sprawozdanie

Wydział Elektroniki

Kierunek: Informatyka Techniczna

Grupa zajęciowa Wt 17:05

Semestr: 2021/2022 Zima

Prowadzący:

Dr inż. Antoni Sternal

Autor

Byczko Maciej 252747

- Programowanie Efektywnych Algorytmów - Projekt: Sprawozdanie
 - Wstęp teoretyczny
 - Problem komiwojażera
 - Simulated annealing (Symulowanie wyżarzanie)
 - Tabu Search
 - Typy sąsiedztwa
 - Plan eksperymentu
 - Wyniki eksperymentów
 - Rozmiar: 17
 - Symulowane Wyżarzanie
 - Sąsiedztwo: Swap
 - Sąsiedztwo: Reverse
 - Tabu Search
 - Sąsiedztwo: Swap, Dywersyfikacja: OFF
 - Sąsiedztwo: Swap, Dywersyfikacja: ON
 - Sąsiedztwo: Reverse, Dywersyfikacja: OFF
 - Sąsiedztwo: Reverse, Dywersyfikacja: ON
 - Rozmiar: 100
 - Symulowane Wyżarzanie
 - Sąsiedztwo: Swap
 - Sąsiedztwo: Reverse
 - Tabu Search
 - Sąsiedztwo: Swap, Dywersyfikacja: OFF
 - Sąsiedztwo: Swap, Dywersyfikacja: ON
 - Sąsiedztwo: Reverse, Dywersyfikacja: OFF
 - Sąsiedztwo: Reverse, Dywersyfikacja: ON
 - Rozmiar: 443
 - Symulowane Wyżarzanie
 - Sąsiedztwo: Swap
 - Sąsiedztwo: Reverse
 - Tabu Search
 - Sąsiedztwo: Swap, Dywersyfikacja: OFF
 - Sąsiedztwo: Swap, Dywersyfikacja: ON

- [Sąsiedztwo: Reverse, Dywersyfikacja: OFF](#)
- [Sąsiedztwo: Reverse, Dywersyfikacja: ON](#)
- [Wnioski](#)
- [Bibliografia](#)

Wstęp teoretyczny

Problem komiwojażera

Problem komiwojażera (ang. travelling salesman problem, TSP) – zagadnienie optymalizacyjne, polegające na znalezieniu drogi o najmniejszym koszcie.

komiwojażer - przedstawiciel firmy podróżujący w celu zdobywania klientów i przyjmowania zamówień na towar. (definicja ze słownika)

W celu zobrazowania problemu należy wyobrazić sobie tytułowego komiwojażera, który podróżuje między miastami w celu wykonywania swojej pracy. Podróż zaczyna z siedziby swojej firmy po czym jego trasa przebiega przez każde miasto dokładnie jeden raz, aż w końcu wraca z powrotem do głównego budynku firmy. Matematycznie prezentujemy ten problem jako graf którego wierzchołki są miastami a łączące je trasy to krawędzie z odpowiednimi wagami. Jest to pełny graf ważony oraz może być skierowany, co tworzy problem asymetryczny.

Rozwiązywanie problemu komiwojażera sprowadza się do znalezienia właściwego - o najmniejszej sumie wag krawędzi - cyklu Hamiltona, czyli cyklu przechodzącego przez każdy wierzchołek grafu dokładnie jeden raz. Przeszukanie wszystkich cykli (czyli zastosowanie metody *Brute Force*(przegląd zupełny)) nie jest optymalną metodą, jako że prowadzi do wykładniczej złożoności obliczeniowej - $O(n!)$, dla której problemy o dużym n traktowane jako nieroziwiązywalne. Klasyfikuje to problem komiwojażera jako problem NP-trudny, czyli niedający rozwiązania w czasie wielomianowym. To powoduje konieczność skorzystania z tzw. algorytmów heurystycznych bądź metaheurystycznych (bardziej ogólnych), a w naszym przypadku konkretnie algorytmu *Tabu search* oraz *Simulated annealing*.

Simulated annealing (Symulowanie wyżarzanie)

Algorytm Simulated Annealing (symulowane wyżarzanie) to kolejna z metod przeszukiwania lokalnego - która podobnie jak Tabu search - bazuje na dynamicznej zmianie sąsiedztwa danego rozwiązania, ale w odróżnieniu od Tabu Search nie przeszukuje całego sąsiedztwa i zmiana zachodzi pod pewnym, ściśle określonym matematycznym równaniem prawdopodobieństwem.

Algorytm powstał na podstawie algorytmu autorstwa N. Metropolisa, służącemu do symulacji zachowań grupy atomów znajdujących się w równowadze termodynamicznej przy zadanej temperaturze.

Zamiast zmiany energii zostały wprowadzone pojęcia nowej i starej wartości funkcji celu, zaś początkowa temperatura w algorytmie zastępuje początkową energię.

Bardzo ogólnie - algorytm polega na losowym przetasowaniu ścieżki i przyjęciu nowego rozwiązania jeżeli jest lepsze, a jeżeli jest gorsze to przyjęcie go z pewnym prawdopodobieństwem.

Umożliwia to wychodzenie poza obszar minimum lokalnego, co znacznie ułatwia odnalezienie minimum globalnego.

Niezbędne jest jednak odpowiednie „nastrojenie” algorytmu, czyli ustawienie specyficznych dla niego parametrów, które i tak nie daje pewności znalezienia minimum globalnego, gdyż zawsze występuje pewien czynnik losowy - w generowaniu sąsiada i w funkcji prawdopodobieństwa.

W symulowanym wyżarzaniu możemy wyróżnić następujące parametry:

- Czas: podawany przez użytkownika,
- Temperatura początkowa: wyznaczana w programie jako iloczyn wielkości problemu oraz wartości losowo znalezionej ścieżki (w celu ustawienia wielkości temperatury aby był dostosowana do wartości ścieżek w podanym problemie),
- Długość epoki - liczba wewnętrznych iteracji dla jednej temperatury: 10-krotna wartość wielkości problemu,
- Funkcja wychładzania: stała wartość ustalona na 0.99, więc $T_{next} = 0.99 * T_{prev}$
- Funkcja prawdopodobieństwa: $e^{-\frac{\text{local cost} - \text{global cost}}{\text{temperature}}}$.

Wykonywanie algorytmu kończy się wraz z upływem podanego czasu.

Tabu Search

Algorytm Tabu search (przeszukiwanie tabu, poszukiwanie z zakazami) to jedna z metod przeszukiwania lokalnego bazująca na dynamicznej zmianie sąsiedztwa danego rozwiązania i szukaniu lokalnie najlepszych rozwiązań, przeznaczona do rozwiązywania problemów optymalizacyjnych.

Przeszukiwanie, dzięki wielu parametrom cechującym Tabu search, może (choć nie musi) doprowadzić do otrzymania globalnie najlepszego rozwiązania.

Algorytm charakteryzuje znikoma złożoność pamięciowa oraz brak jawnie zdefiniowanej czasowej złożoności obliczeniowej, gdyż algorytm kończy się wraz z pewnym warunkiem, w naszym przypadku wykonywanie go trwa określony czas.

W zaimplementowanym Tabu Search możemy wyróżnić następujące parametry:

- Czas: podawany przez użytkownika,
- Kadencja: $\sqrt{\text{problem size}}$
- Ilość braku polepszenia ścieżki w celu zastosowania dywersyfikacji: ustaliona na stałą wartość 20.

Lista Tabu została zaimplementowana jako tablica dwuwymiarowa (macierz $n \times n$) gdzie aktualna kadencja jest oznaczana jako $tabuList[i, j]$ gdzie i, j to sąsiedztwo dla wierzchołków i oraz j .

Wykonywanie algorytmu kończy się wraz z upływem podanego czasu.

Typy sąsiedztwa

Zaimplementowane zostały 2 różne rodzaje sąsiedztwa:

- Swap - zamiana dwóch elementów
- Reverse - Zamiana kolejności elementów pomiędzy podanymi indeksami

Przykład działania sąsiedztw na przykładowej tablicy wartości:

```
Tablica tab = [0,1,2,3,4,5]
```

```
Indeks a = 1
```

```
Indeks b = 4
```

```
Wynik sąsiedztwa Swap(tab, a, b) -> [0,4,2,3,1,5]
```

```
Wynik sąsiedztwa Reverse(tab, a, b) -> [0,4,3,2,1,5]
```

Plan eksperymentu

Program został napisany w języku C#, w .NET Framework, w środowisku JetBrains Rider.

Do mierzenia czasu wykorzystano klasę Stopwatch z przestrzeni nazw System.Diagnostics.

Testowane instancje problemu były wielkości:

17, 100 oraz **443**.

W tabelach zostały umieszczone wartości gdy zachodziła zmiana w znalezionym rozwiążaniu globalnym.

Wyniki eksperymentów

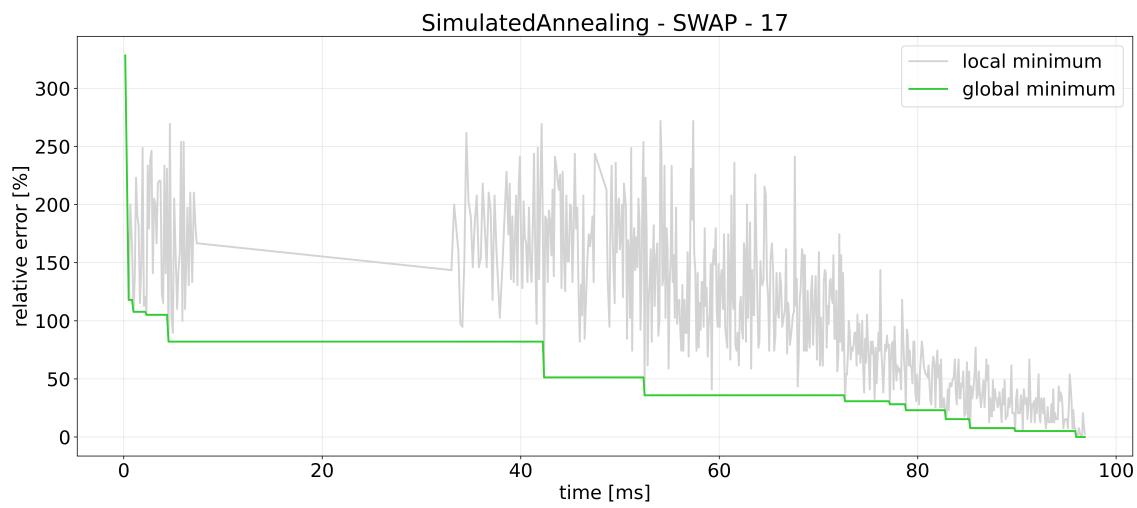
Rozmiar: 17

- **Poprawna ścieżka: 39**

Symulowane Wyżarzanie

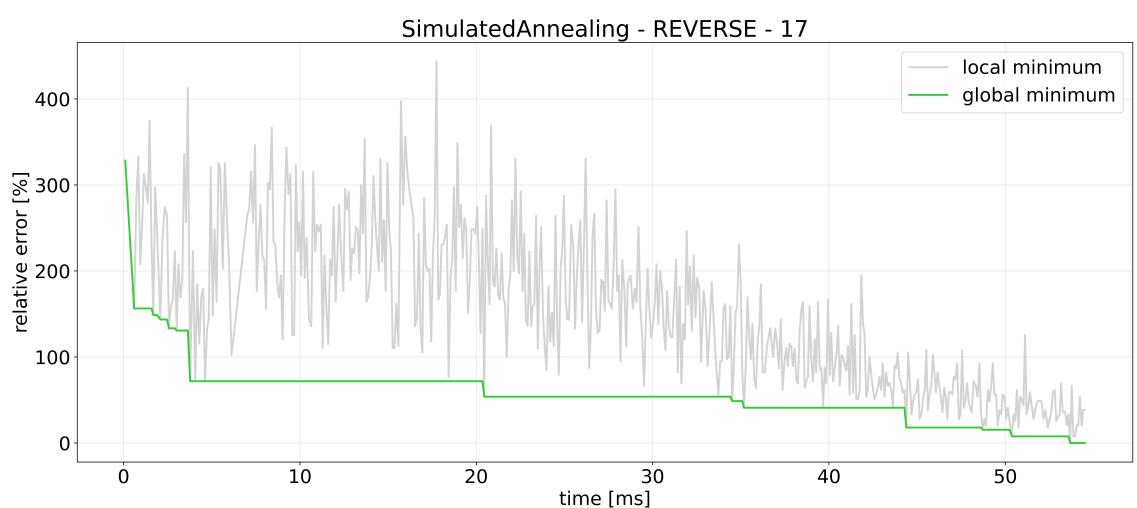
Sąsiedztwo: Swap

nr.	Czas [ms]	Ścieżka	Błąd Względny [%]
1	0.15	167	328.21
2	0.98	81	107.69
3	4.51	71	82.05
4	42.35	59	51.28
5	52.48	53	35.9
6	72.68	51	30.77
7	78.81	48	23.08
8	82.83	45	15.38
9	89.82	41	5.13
10	95.97	39	0.0



Sąsiedztwo: Reverse

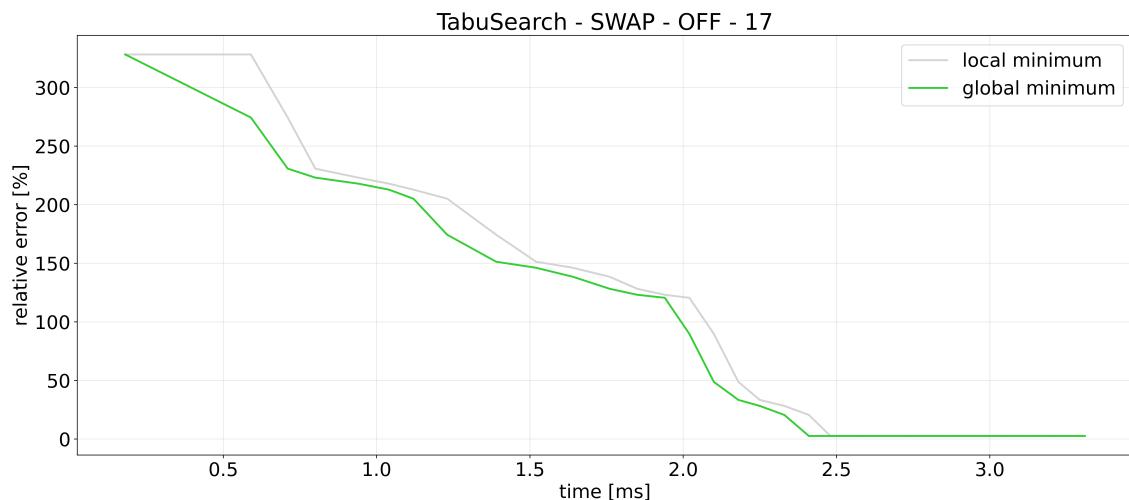
nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.09	167	328.21
2	2.08	95	143.59
3	3.01	90	130.77
4	3.77	67	71.79
5	20.45	60	53.85
6	35.18	55	41.03
7	44.39	46	17.95
8	48.72	45	15.38
9	50.38	42	7.69
10	53.68	39	0.0



Tabu Search

Sąsiedztwo: Swap, Dywersyfikacja: OFF

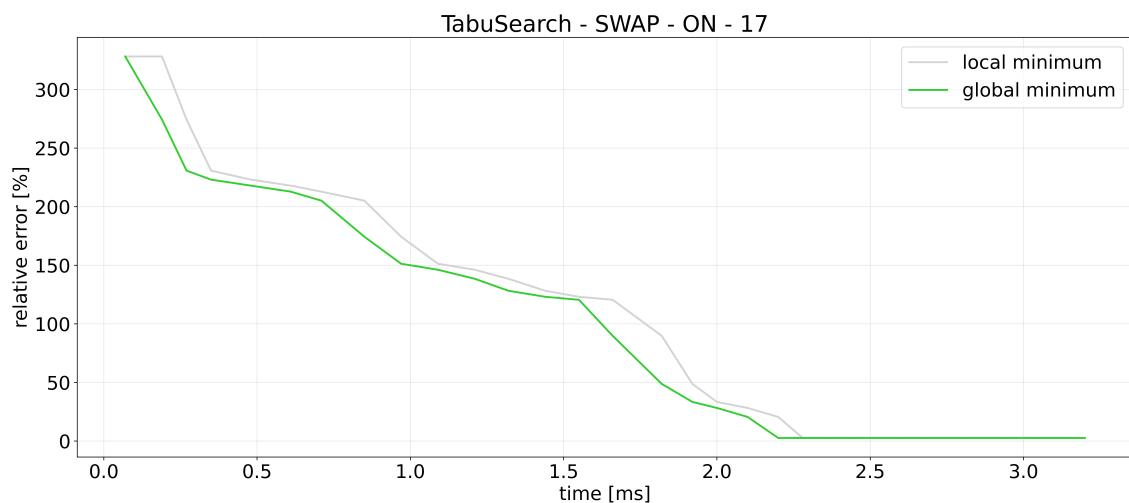
nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.18	167	328.21
2	0.94	124	217.95
3	1.23	107	174.36
4	1.76	89	128.21
5	1.94	86	120.51
6	2.02	74	89.74
7	2.1	58	48.72
8	2.18	52	33.33
9	2.33	47	20.51
10	2.41	40	2.56



Sąsiedztwo: Swap, Dywersyfikacja: ON

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.07	167	328.21
2	0.19	146	274.36
3	0.27	129	230.77
4	0.85	107	174.36
5	1.55	86	120.51
6	1.66	74	89.74

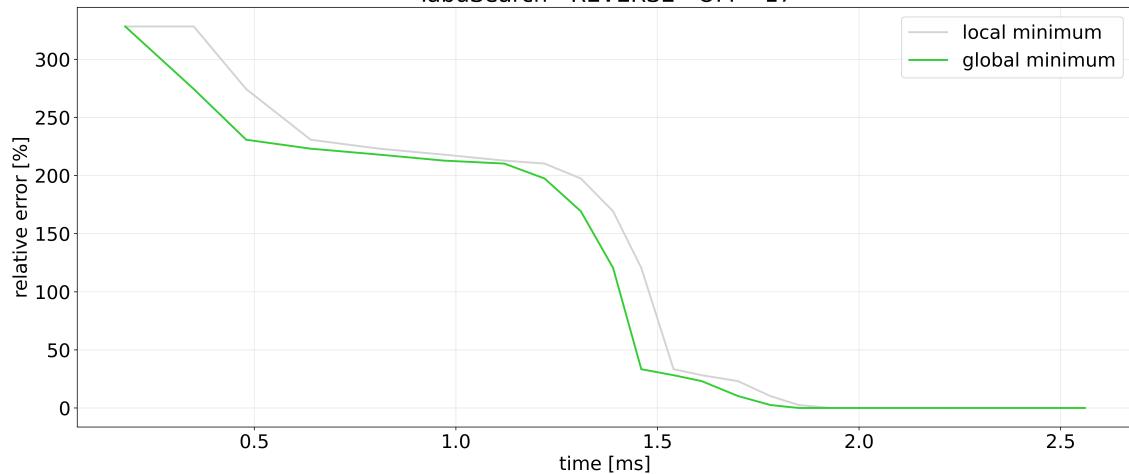
nr.	Czas [ms]	Ścieżka	Błąd Względny
7	1.82	58	48.72
8	1.92	52	33.33
9	2.1	47	20.51
10	2.2	40	2.56



Sąsiedztwo: Reverse, Dywersyfikacja: OFF

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.18	167	328.21
2	0.35	146	274.36
3	0.48	129	230.77
4	1.22	116	197.44
5	1.31	105	169.23
6	1.39	86	120.51
7	1.61	48	23.08
8	1.7	43	10.26
9	1.78	40	2.56
10	1.85	39	0.0

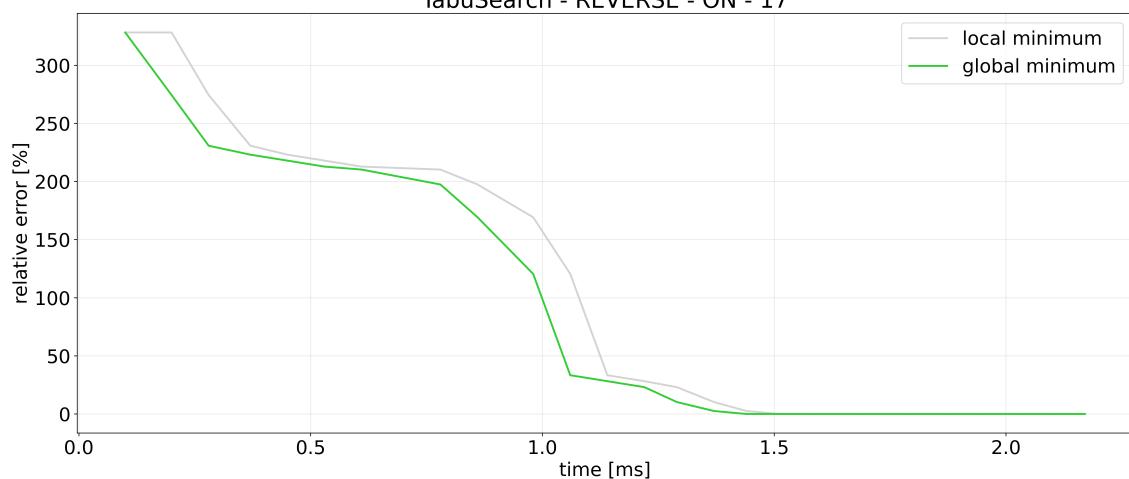
TabuSearch - REVERSE - OFF - 17



Sąsiedztwo: Reverse, Dywersyfikacja: ON

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.1	167	328.21
2	0.2	146	274.36
3	0.28	129	230.77
4	0.78	116	197.44
5	0.86	105	169.23
6	0.98	86	120.51
7	1.06	52	33.33
8	1.14	50	28.21
9	1.22	48	23.08
10	1.44	39	0.0

TabuSearch - REVERSE - ON - 17



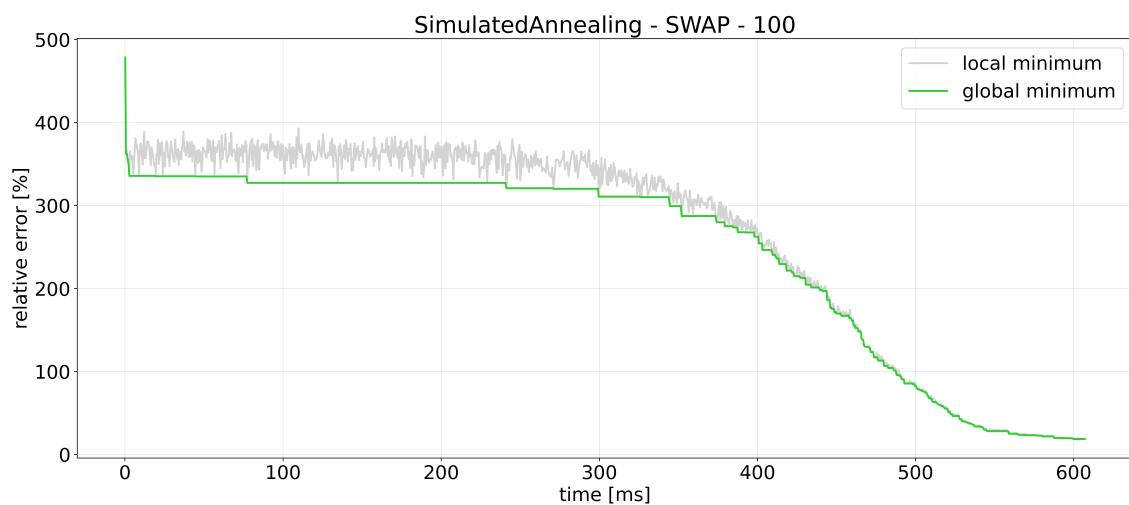
Rozmiar: 100

- **Poprawna ścieżka: 36230**

Symulowane Wyżarzanie

Sąsiedztwo: Swap

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.15	209567	478.43
2	1.74	164991	355.4
3	384.79	135361	273.62
4	430.62	110374	204.65
5	468.2	83238	129.75
6	486.31	72936	101.31
7	487.81	71653	97.77
8	491.14	69179	90.94
9	492.97	67221	85.54
10	500.69	65314	80.28
11	507.69	61844	70.7
12	510.62	59307	63.7
13	514.17	58295	60.9
14	516.55	57265	58.06
15	519.51	56095	54.83
16	528.23	51556	42.3
17	537.39	48468	33.78
18	543.45	47102	30.01
19	580.68	44142	21.84
20	599.65	43023	18.75

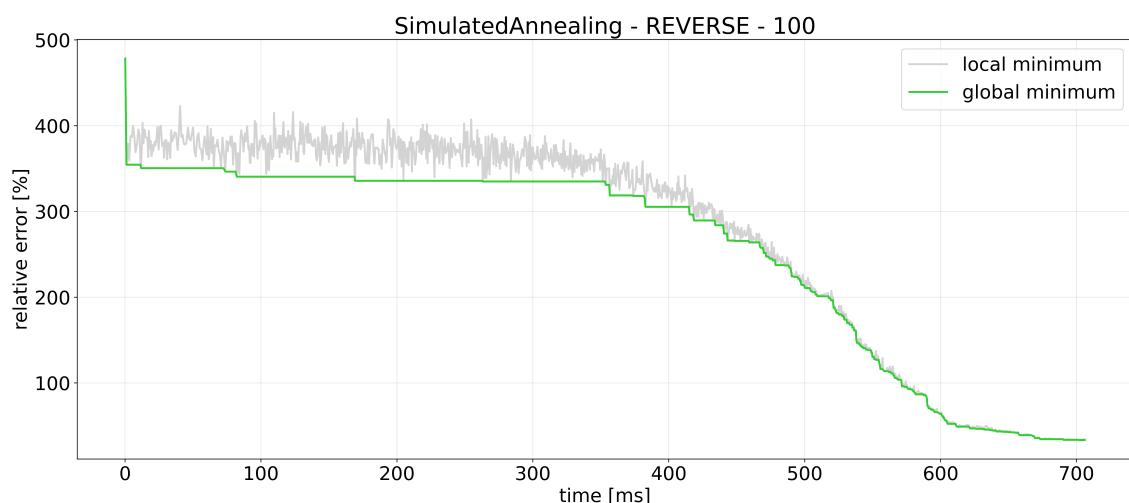


Sąsiedztwo: Reverse

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.13	209567	478.43
2	169.25	157894	335.81
3	415.29	143652	296.5
4	448.46	132503	265.73
5	469.59	128549	254.81
6	490.02	119696	230.38
7	509.31	109119	201.18
8	522.43	103454	185.55
9	529.14	99263	173.98
10	538.02	90535	149.89
11	542.97	87254	140.83
12	557.74	78018	115.34
13	566.35	74557	105.79
14	580.86	68345	88.64
15	590.33	62777	73.27
16	602.16	57642	59.1
17	625.95	53101	46.57
18	655.69	51406	41.89
19	669.34	49228	35.88

nr.	Czas [ms]	Ścieżka	Błąd Względny
------------	------------------	----------------	----------------------

20 690.35 48479 33.81



Tabu Search

Sąsiedztwo: Swap, Dywersyfikacja: OFF

nr.	Czas [ms]	Ścieżka	Błąd Względny
------------	------------------	----------------	----------------------

1 0.13 209567 478.43

2 30.7 197531 445.21

3 83.66 187556 417.68

4 122.67 180016 396.87

5 155.9 170229 369.86

6 214.84 160026 341.69

7 251.7 153215 322.9

8 299.73 145887 302.67

9 389.5 133293 267.91

10 449.08 125084 245.25

11 496.61 118028 225.77

12 541.0 110065 203.8

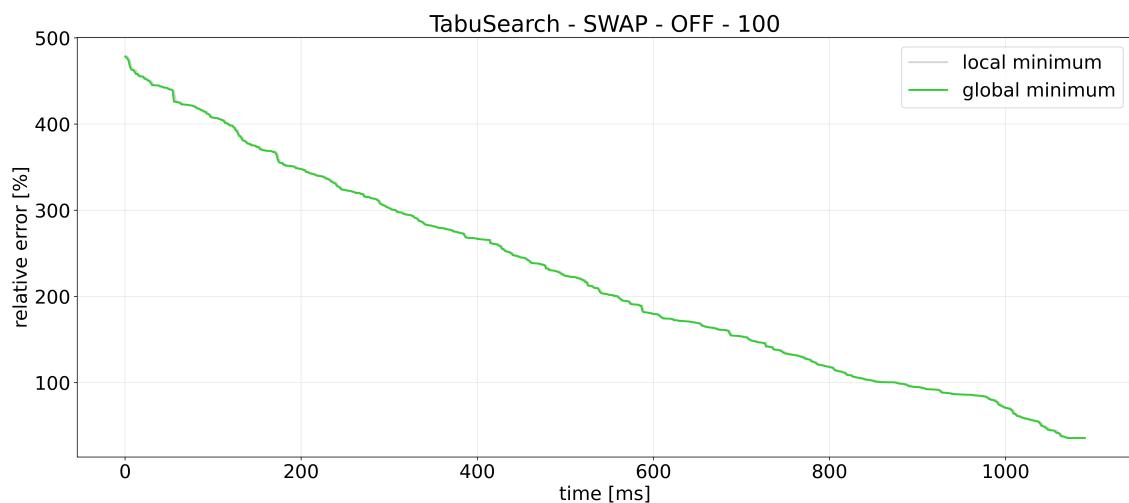
13 595.47 101692 180.68

14 658.87 95963 164.87

15 727.93 87683 142.02

16 784.96 80585 122.43

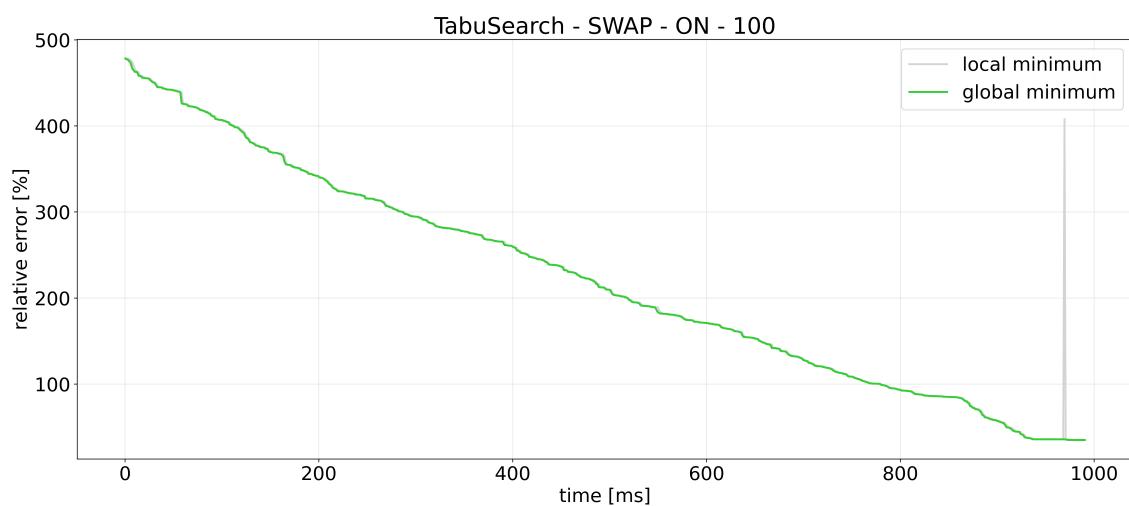
nr.	Czas [ms]	Ścieżka	Błąd Względny
17	926.46	68713	89.66
18	998.97	61927	70.93
19	1045.19	53613	47.98
20	1070.24	49217	35.85



Sąsiedztwo: Swap, Dywersyfikacja: ON

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.19	209567	478.43
2	33.2	197531	445.21
3	80.46	187556	417.68
4	117.33	180016	396.87
5	149.31	170229	369.86
6	199.06	160026	341.69
7	227.53	153215	322.9
8	279.43	145887	302.67
9	373.46	133293	267.91
10	425.43	125084	245.25
11	468.72	118028	225.77
12	503.82	110065	203.8
13	562.31	101692	180.68
14	618.03	95963	164.87

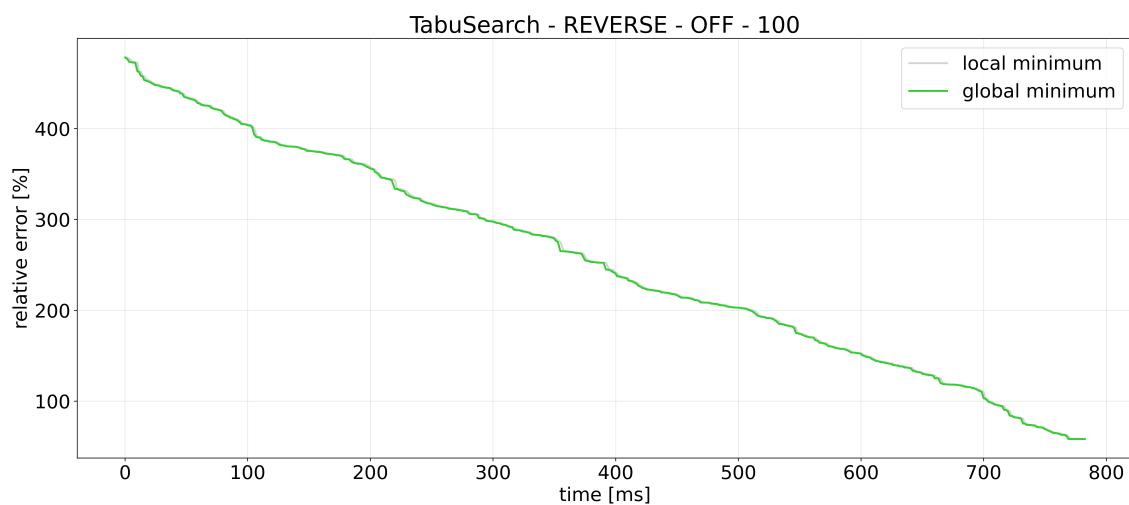
nr.	Czas [ms]	Ścieżka	Błąd Względny
15	667.29	87683	142.02
16	710.59	80585	122.43
17	755.8	74622	105.97
18	813.5	68713	89.66
19	877.32	61927	70.93
20	937.09	49217	35.85



Sąsiedztwo: Reverse, Dywersyfikacja: OFF

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.14	209567	478.43
2	26.26	198480	447.83
3	82.67	186349	414.35
4	108.27	177718	390.53
5	199.28	165433	356.62
6	232.14	154136	325.44
7	271.17	148737	310.54
8	312.6	142165	292.4
9	351.17	136463	276.66
10	399.57	123513	240.91
11	435.7	116294	220.99
12	482.34	111202	206.93

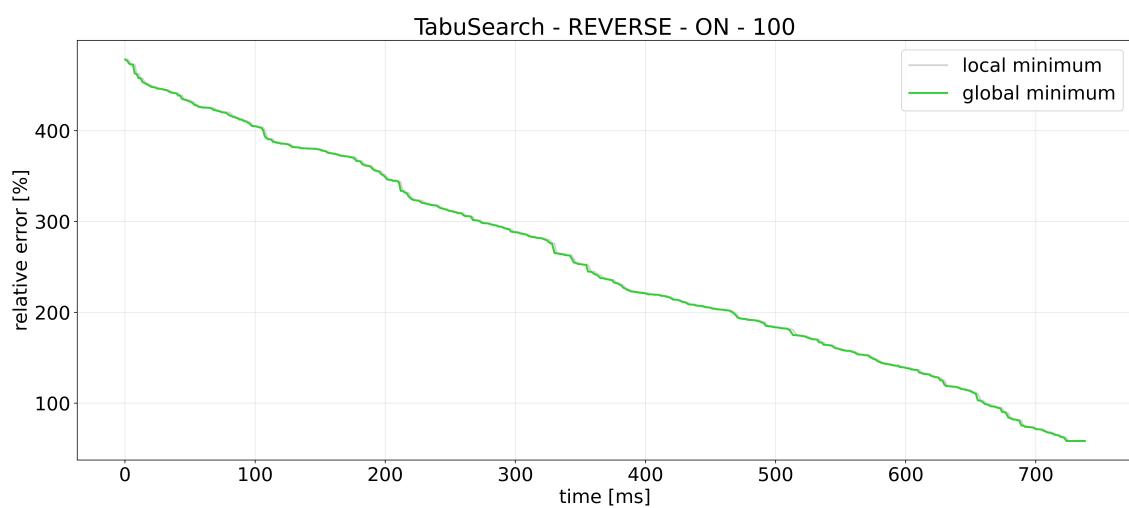
nr.	Czas [ms]	Ścieżka	Błąd Względny
13	522.34	105680	191.69
14	554.22	98450	171.74
15	588.75	92734	155.96
16	658.81	82566	127.89
17	699.97	73612	103.18
18	724.54	66086	82.41
19	756.45	59796	65.05
20	769.0	57355	58.31



Sąsiedztwo: Reverse, Dywersyfikacja: ON

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.18	209567	478.43
2	22.08	198480	447.83
3	85.01	186349	414.35
4	110.6	177718	390.53
5	160.51	171891	374.44
6	191.34	165433	356.62
7	219.49	154136	325.44
8	252.74	148737	310.54
9	292.15	142165	292.4
10	326.25	136463	276.66

nr.	Czas [ms]	Ścieżka	Błąd Względny
11	362.89	123513	240.91
12	443.41	111202	206.93
13	478.64	105680	191.69
14	525.19	98450	171.74
15	560.23	92734	155.96
16	625.16	82566	127.89
17	655.35	73612	103.18
18	681.73	66086	82.41
19	714.72	59796	65.05
20	723.44	57355	58.31



Rozmiar: 443

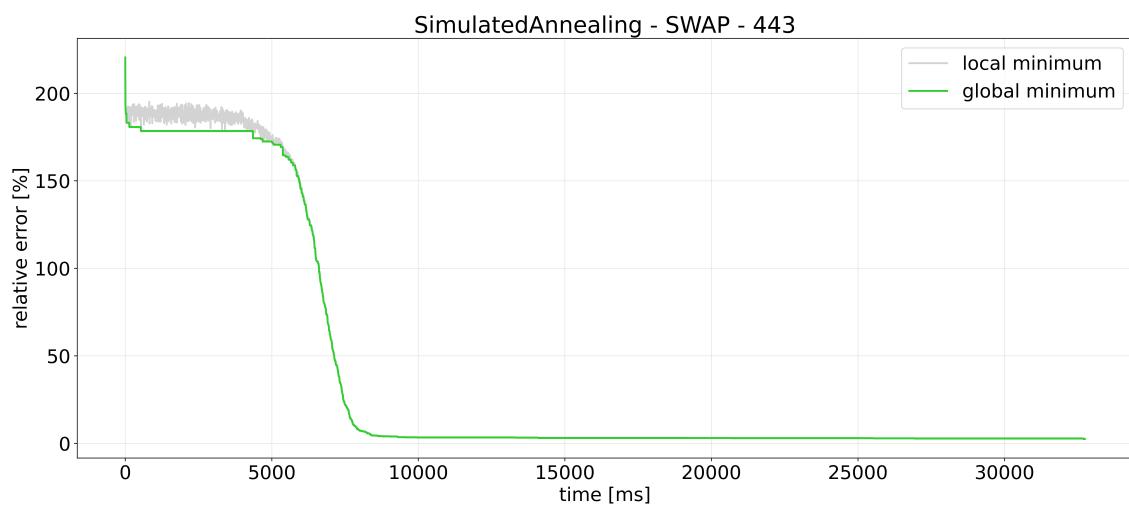
- Poprawna ścieżka: 2720

Symulowane Wyżarzanie

Sąsiedztwo: Swap

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.13	8717	220.48
2	4360.48	7462	174.34
3	5487.28	7174	163.75
4	5836.24	6957	155.77

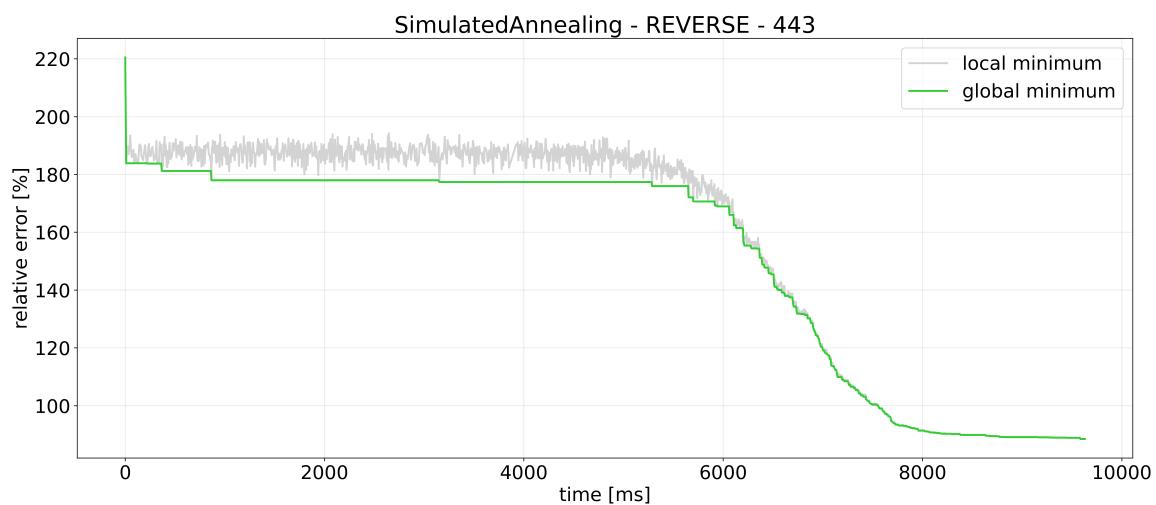
nr.	Czas [ms]	Ścieżka	Błąd Względny
5	5979.48	6677	145.48
6	6204.55	6251	129.82
7	6389.1	6008	120.88
8	6457.61	5763	111.88
9	6579.39	5515	102.76
10	6672.07	5214	91.69
11	6740.55	5014	84.34
12	6841.33	4809	76.8
13	6930.52	4564	67.79
14	7070.06	4182	53.75
15	7182.29	3958	45.51
16	7279.08	3771	38.64
17	7457.54	3371	23.93
18	7640.01	3174	16.69
19	7841.85	2990	9.93
20	25526.59	2800	2.94



Sąsiedztwo: Reverse

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.15	8717	220.48
2	865.15	7562	178.01

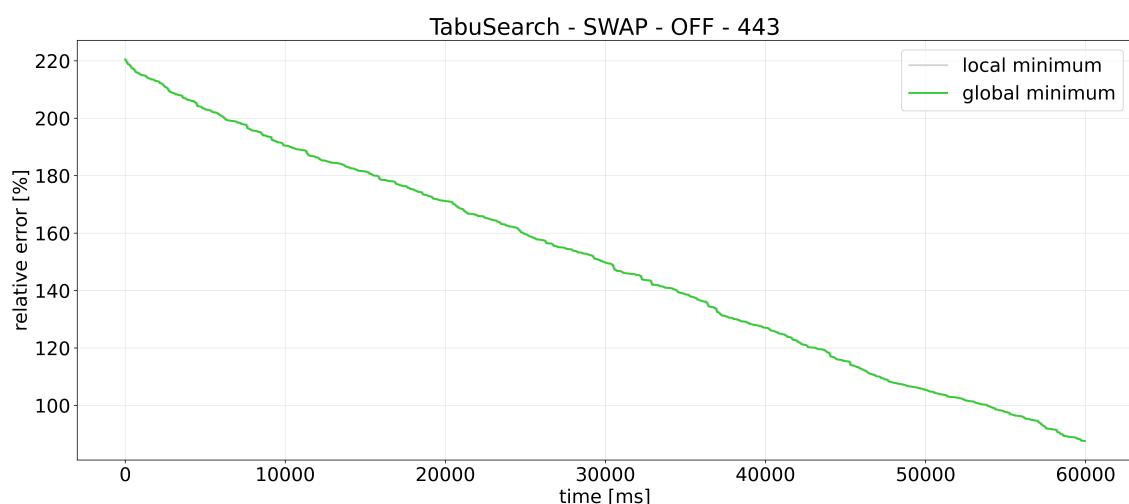
nr.	Czas [ms]	Ścieżka	Błąd Względny
3	5652.44	7401	172.1
4	5940.62	7316	168.97
5	6131.04	7113	161.51
6	6283.2	6921	154.45
7	6392.62	6768	148.82
8	6588.43	6506	139.19
9	6740.81	6307	131.88
10	6880.66	6218	128.6
11	6968.1	6028	121.62
12	7018.12	5942	118.46
13	7139.43	5755	111.58
14	7206.65	5677	108.71
15	7433.81	5502	102.28
16	7557.26	5434	99.78
17	7643.16	5357	96.95
18	7713.99	5275	93.93
19	8068.24	5191	90.85
20	9309.6	5140	88.97



Tabu Search

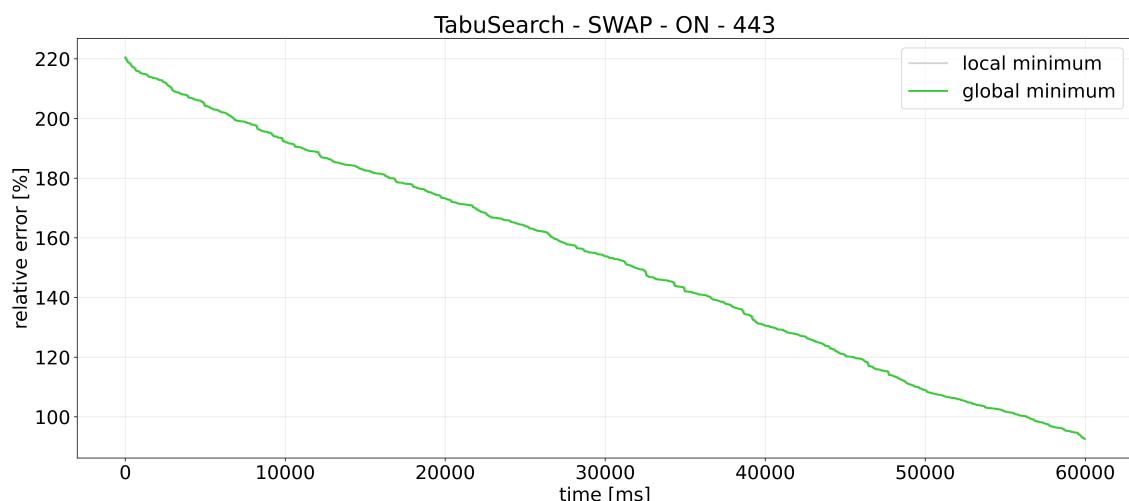
Sąsiedztwo: Swap, Dywersyfikacja: OFF

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.1	8717	220.48
2	1128.2	8567	214.96
3	2890.37	8403	208.93
4	5147.05	8240	202.94
5	7640.59	8065	196.51
6	9882.22	7903	190.55
7	12626.68	7750	184.93
8	15887.41	7585	178.86
9	18980.33	7421	172.83
10	21403.0	7257	166.8
11	24645.57	7095	160.85
12	27399.0	6935	154.96
13	30469.31	6764	148.68
14	33064.4	6582	141.99
15	36350.75	6408	135.59
16	38232.35	6251	129.82
17	44053.71	5903	117.02
18	49612.24	5602	105.96
19	53860.97	5436	99.85
20	59711.26	5107	87.76



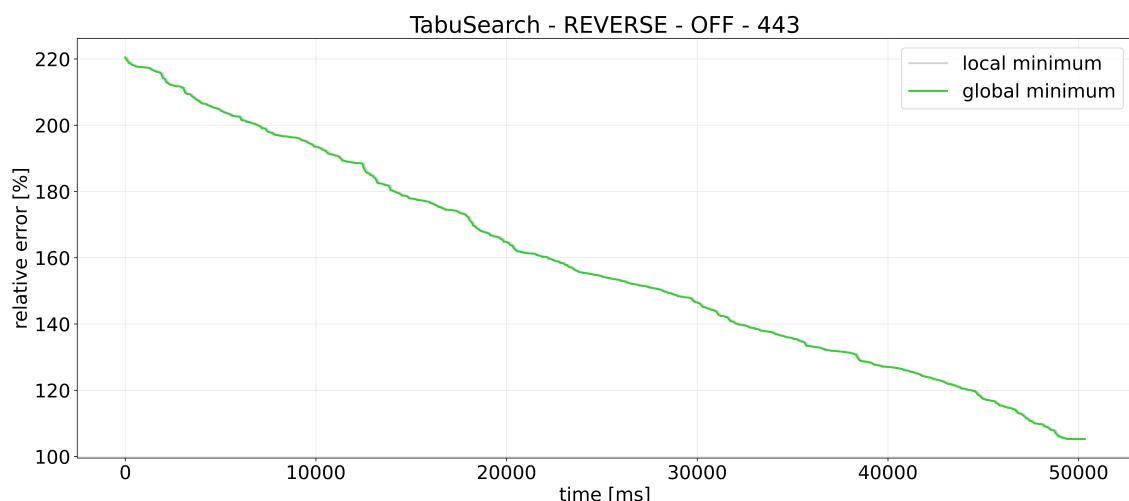
Sąsiedztwo: Swap, Dywersyfikacja: ON

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.12	8717	220.48
2	1176.52	8567	214.96
3	3131.1	8403	208.93
4	5653.38	8240	202.94
5	8268.22	8065	196.51
6	10624.41	7903	190.55
7	13449.07	7750	184.93
8	16917.75	7585	178.86
9	20154.48	7421	172.83
10	22889.55	7257	166.8
11	26535.58	7095	160.85
12	29355.14	6935	154.96
13	32503.17	6764	148.68
14	35151.25	6582	141.99
15	38586.37	6408	135.59
16	40537.07	6251	129.82
17	46458.73	5903	117.02
18	52023.7	5602	105.96
19	56355.0	5436	99.85
20	59896.48	5245	92.83



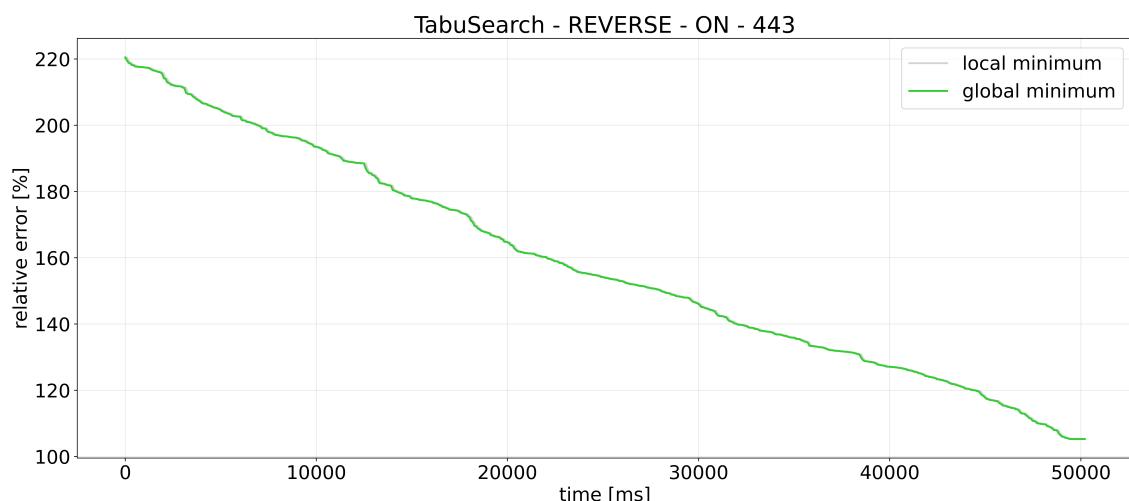
Sąsiedztwo: Reverse, **Dyweryfikacja:** OFF

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.3	8717	220.48
2	1792.33	8595	215.99
3	5204.67	8265	203.86
4	7435.89	8112	198.24
5	9908.51	7985	193.57
6	11734.1	7860	188.97
7	14873.92	7561	177.98
8	17902.85	7415	172.61
9	18792.93	7285	167.83
10	23070.43	7014	157.87
11	26081.45	6881	152.98
12	29510.17	6745	147.98
13	31154.51	6598	142.57
14	36319.43	6337	132.98
15	39269.23	6194	127.72
16	42737.41	6064	122.94
17	44892.41	5927	117.9
18	46903.49	5793	112.98
19	48626.69	5657	107.98
20	49059.38	5602	105.96



Sąsiedztwo: Reverse, Dywersyfikacja: ON

nr.	Czas [ms]	Ścieżka	Błąd Względny
1	0.09	8717	220.48
2	1833.37	8595	215.99
3	3530.98	8398	208.75
4	5215.31	8265	203.86
5	7420.2	8112	198.24
6	9876.95	7985	193.57
7	11778.06	7860	188.97
8	13285.36	7689	182.68
9	14965.81	7561	177.98
10	17925.35	7415	172.61
11	18791.93	7285	167.83
12	20346.98	7151	162.9
13	22968.37	7014	157.87
14	29347.08	6745	147.98
15	31007.21	6598	142.57
16	33288.31	6473	137.98
17	42814.62	6064	122.94
18	44965.19	5927	117.9
19	46950.04	5793	112.98
20	49090.64	5602	105.96



Wnioski

Zaprezentowane metody metaheurystyczne, dostosowane pod problem komiwojażera mogą być bardzo efektywne pod warunkiem dobrej, przemyślanej implementacji i odpowiedniego "nastrojenia" (dobrania parametrów) pod odpowiednie wielkości zadanych instancji.

Projekt można wyraźnie podzielić na dwie części:

- Projektowanie, implementacja zadanych algorytmów, przemyślenie przekazywania sąsiedztwa
- Testowanie, zapisanie interesujących nas wyników działania algorytmów

Z tych dwóch etapów zdecydowanie więcej zajął etap testowania algorytmów, implementacja metod testujących w kodzie aby nie obniżyło one znacząco efektywności działania algorytmów.

Na załączonych wykresach można zauważyc że w każdym przypadku funkcja błędu względnego od czasu jest nierosnąca, co nie powinno być zaskoczeniem gdyż dążymy do najmniejszego błędu względnego.

Dla małych instancji bardzo szybko sobie radził algorytm Tabu Search przy odpowiednich parametrach (dla Swap nie potrafił znaleźć rozwiązania przy wielkości 17), dla większych znacznie bliżej rozwiązania był algorytm symulowanego wyżarzania.

Na podanych przypadkach niestety nie możemy wyraźnie zauważyc aktywacji dywersyfikacji dla Tabu Search, gdyż nigdy nie znaleźliśmy się w wystarczająco głębokim minimum lokalnym, lecz gdyby była taka sytuacja to dywersyfikacja jest niezastąpiona.

Przy sąsiedztwie Swap wyniki były bliższe prawidłowemu rozwiązaniu gdyż za pomocą Swap jest mniejsza szansa na pominięcie rozwiązania gdyż jest łagodniejsze przeszukiwanie, w przypadku Reverse mogliśmy w skrajnych przypadkach odwrócić nawet całą permutację.

Podsumowując najlepiej wypadł algorytm symulowanego wyżarzania z sąsiedztwem Swap, uzyskane wyniki były najbliższe optymalnemu rozwiązaniu.

Bibliografia

1. https://en.wikipedia.org/wiki/Fisher–Yates_shuffle
2. <https://cs.pwr.edu.pl/zielinski/lectures/om/localsearch.pdf>
3. http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w2.pdf
4. http://www.cs.put.poznan.pl/mradom/teaching/laboratories/OptKomb/dziamski_4.pdf
5. <http://www2.imm.dtu.dk/courses/02719/tabu/4tabu2.pdf>
6. <https://sandipanweb.wordpress.com/2020/12/08/travelling-salesman-problem-tsp-with-python/>