

Programowanie Efektywnych Algorytmów - Projekt: Sprawozdanie

Wydział Elektroniki

Kierunek: Informatyka Techniczna

Grupa zajęciowa Wt 17:05

Semestr: 2021/2022 Zima

Prowadzący:

Dr inż. Antoni Sterna

Autor

Byczko Maciej 252747

- Programowanie Efektywnych Algorytmów - Projekt: Sprawozdanie
 - Wstęp teoretyczny
 - Problem komiwojażera
 - Simulated annealing (Symulowanie wyżarzanie)
 - Tabu Search
 - Typy sąsiedztwa
 - Plan eksperymentu

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 nieroziwiązalne. 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 po-dobnie 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.

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.

Typy sąsiedztwa

Zaimplementowane zostały 2 różne rodzaje sąsiedztwa:

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

Plan eksperymentu

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

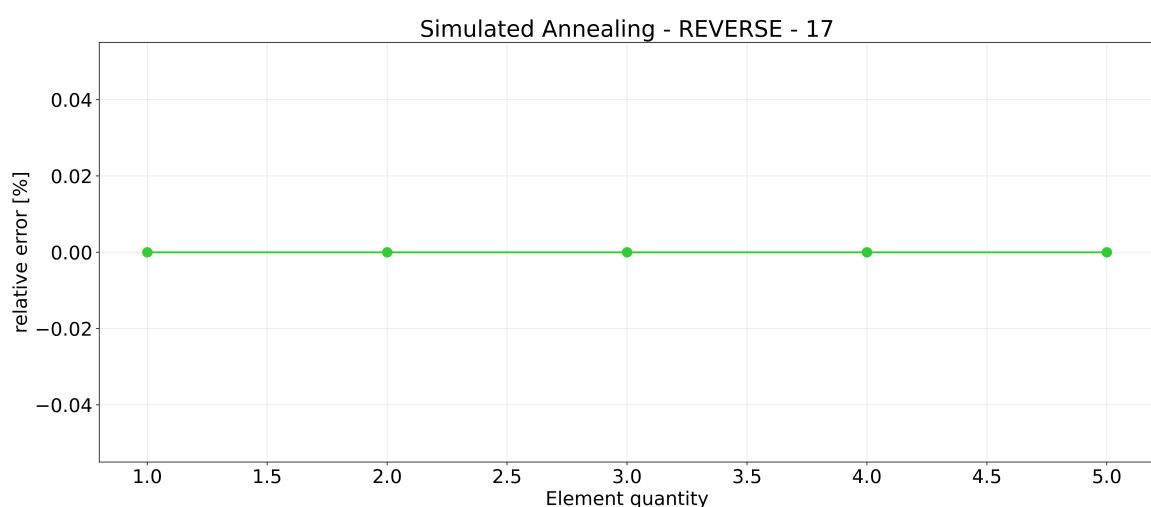
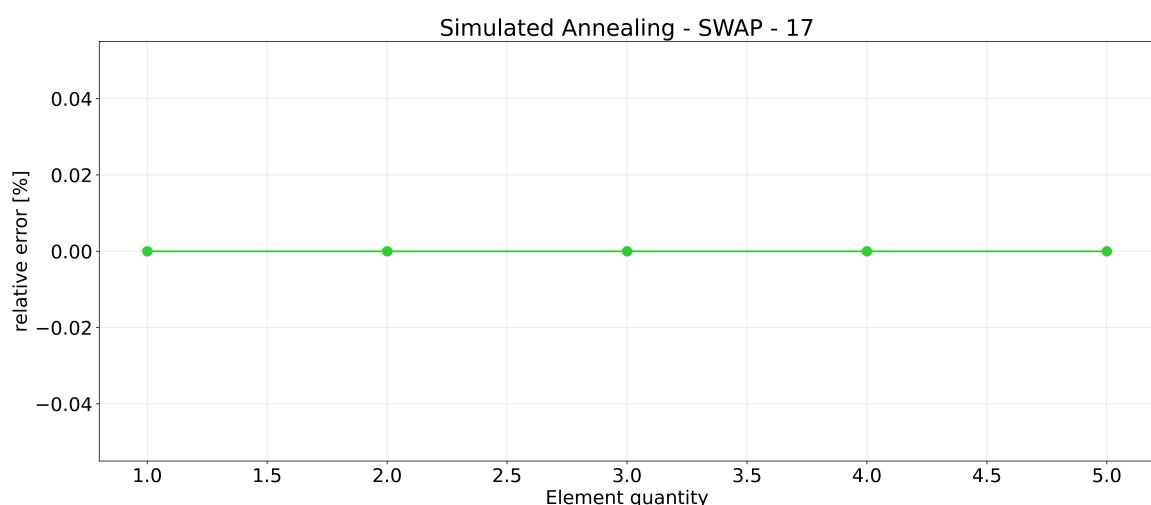
Do mierzenia czasu wykorzystano klasę `Stopwatch` z przestrzeni nazw `System.Diagnostics`.

Testowane instancje problemu były wielkości: 17,65,171,443

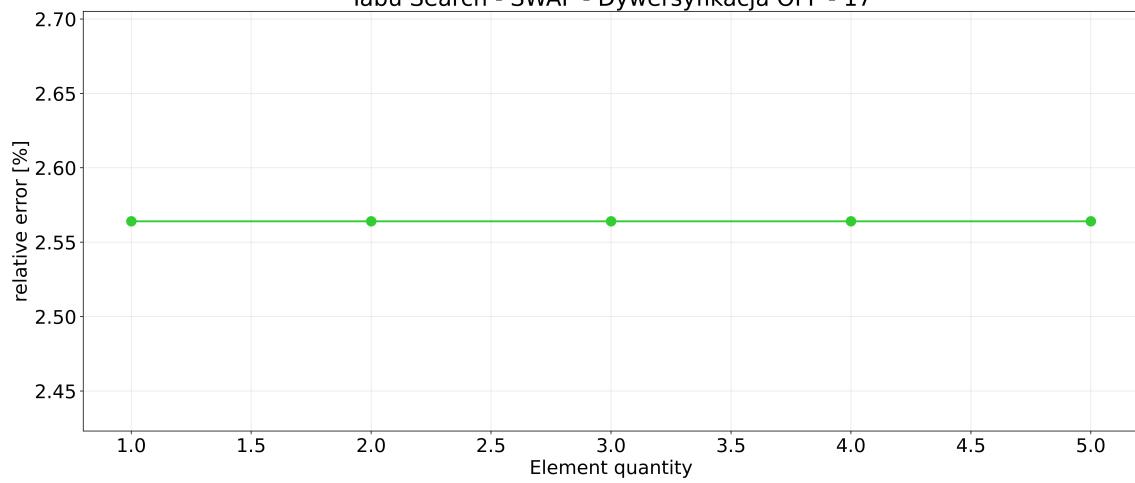
Rozmiar: 17

czas [s]	Simulated Annealing - SWAP	Simulated Annealing -	Tabu Search - SWAP - Dyweryfikacja	Tabu Search - SWAP - Dyweryfikacja	Tabu Search - REVERSE - Dyweryfikacja	Tabu Search - REVERSE - Dyweryfikacja
1	0.0	0.0	2.564	2.564	0.0	0.0

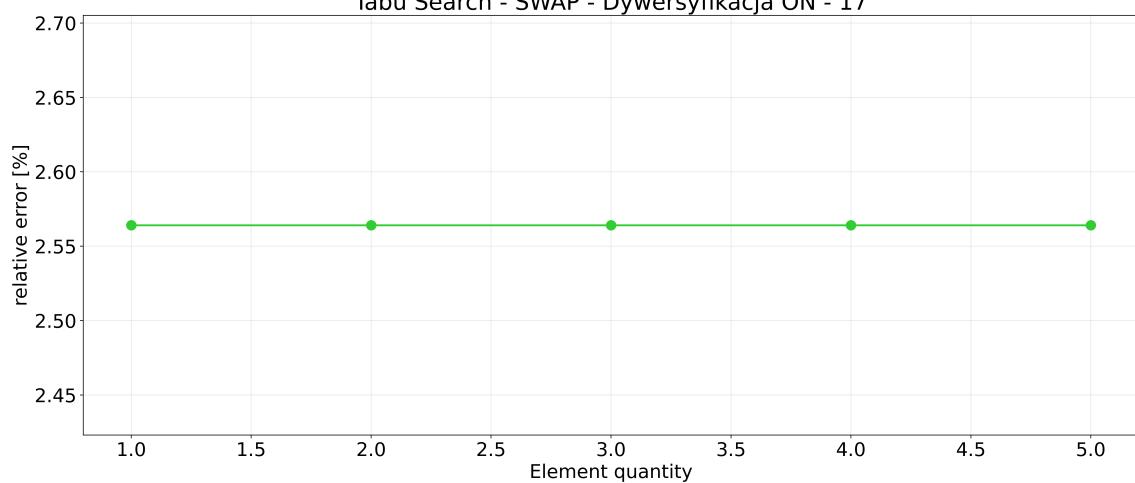
czas [s]	Simulated Annealing - SWAP	Simulated Annealing - REVERSE	Tabu Search - SWAP - Dywersyfikacja OFF	Tabu Search - SWAP - Dywersyfikacja ON	Tabu Search - REVERSE - Dywersyfikacja OFF	Tabu Search - REVERSE - Dywersyfikacja ON
2	0.0	0.0	2.564	2.564	0.0	0.0
3	0.0	0.0	2.564	2.564	0.0	0.0
4	0.0	0.0	2.564	2.564	0.0	0.0
5	0.0	0.0	2.564	2.564	0.0	0.0



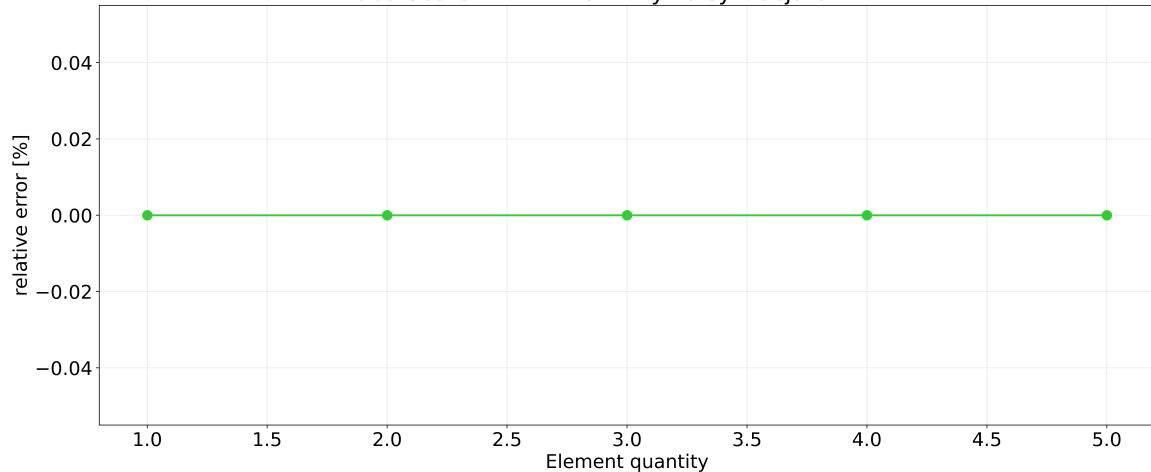
Tabu Search - SWAP - Dywersyfikacja OFF - 17



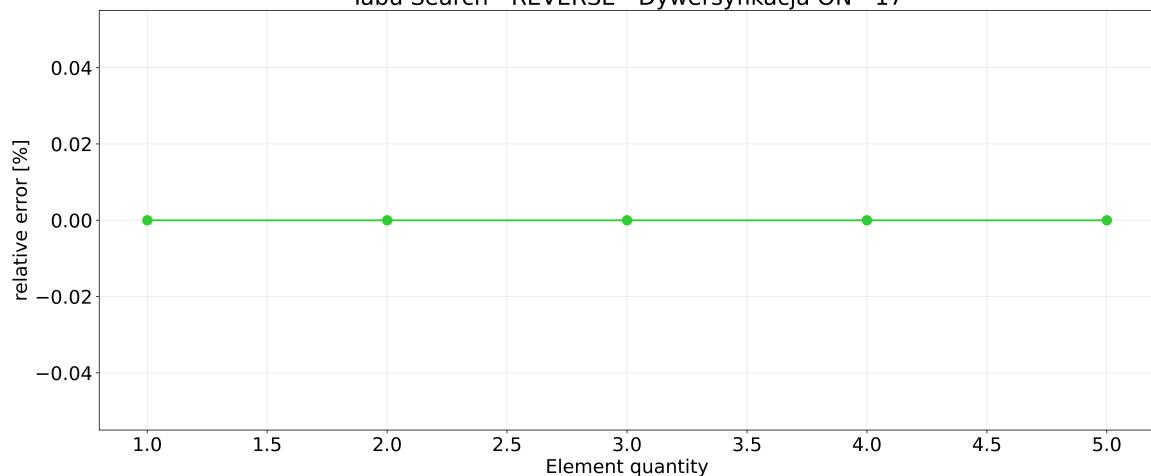
Tabu Search - SWAP - Dywersyfikacja ON - 17



Tabu Search - REVERSE - Dywersyfikacja OFF - 17



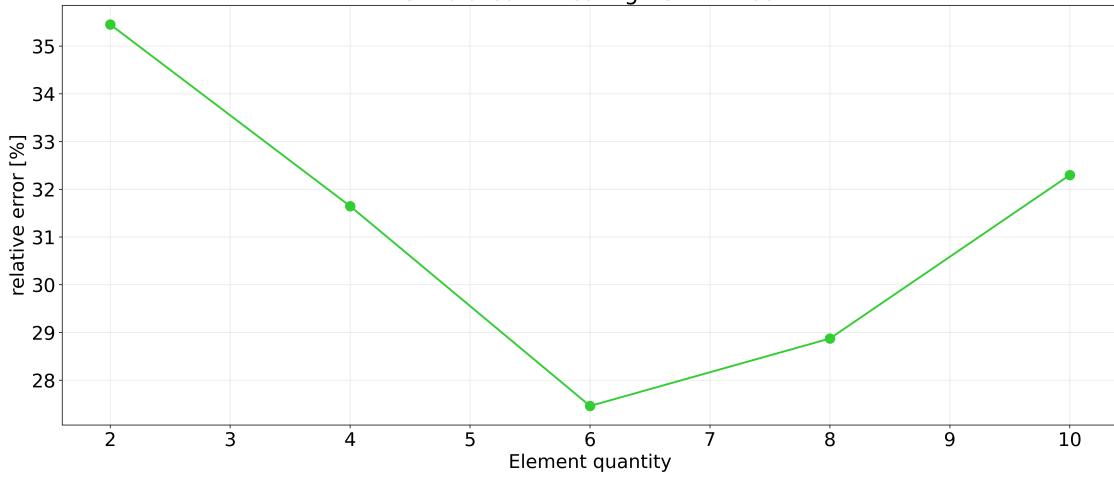
Tabu Search - REVERSE - Dywersyfikacja ON - 17



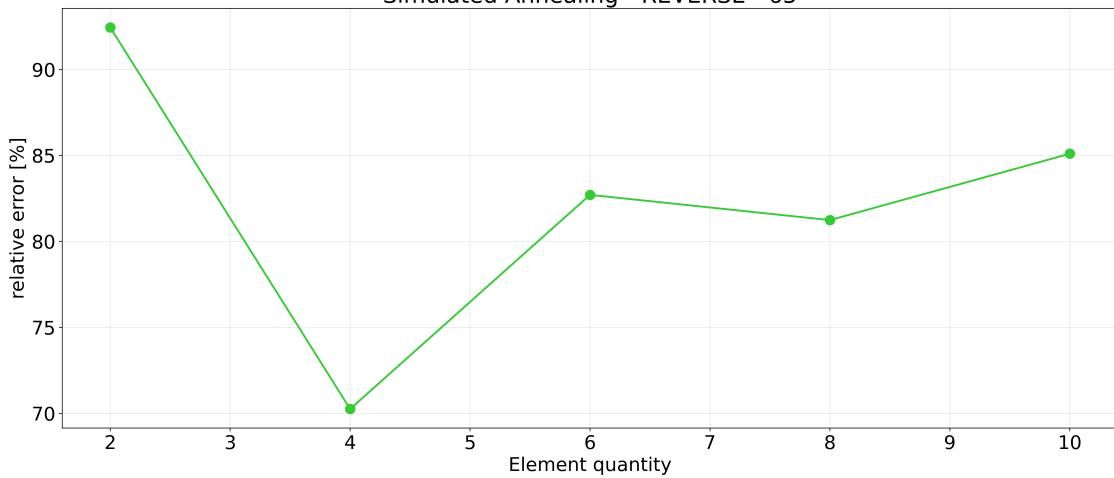
Rozmiar: 65

czas [s]	Simulated Annealing - SWAP	Simulated Annealing - REVERSE	Tabu Search - SWAP - Dywersyfikacja OFF	Tabu Search - SWAP - Dywersyfikacja ON	Tabu Search - REVERSE - Dywersyfikacja OFF	Tabu Search - REVERSE - Dywersyfikacja ON
2	35.454	92.442	46.112	46.112	115.661	115.661
4	31.648	70.256	46.112	46.112	115.661	115.661
6	27.461	82.708	46.112	46.112	115.661	115.661
8	28.874	81.240	46.112	46.112	115.661	115.661
10	32.3	85.101	46.112	46.112	115.661	115.661

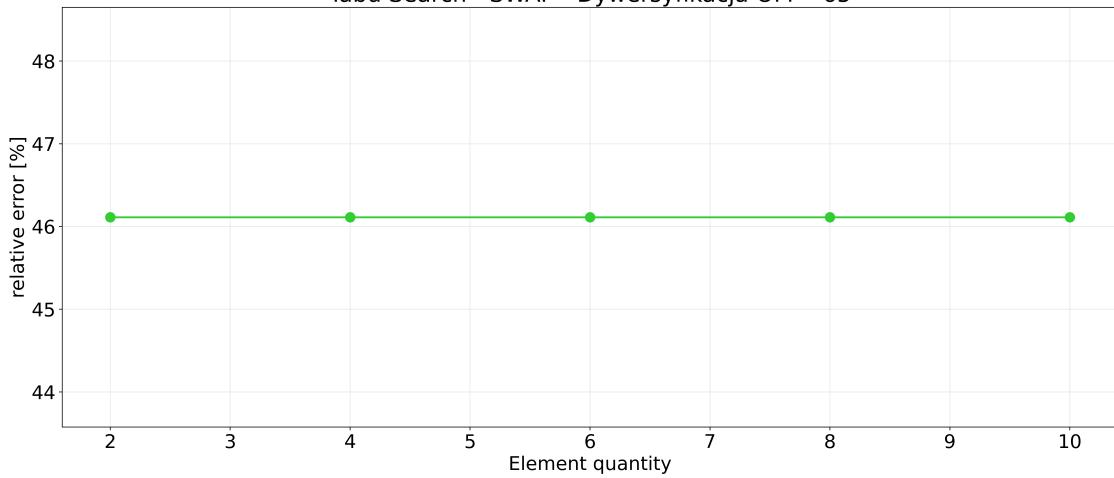
Simulated Annealing - SWAP - 65



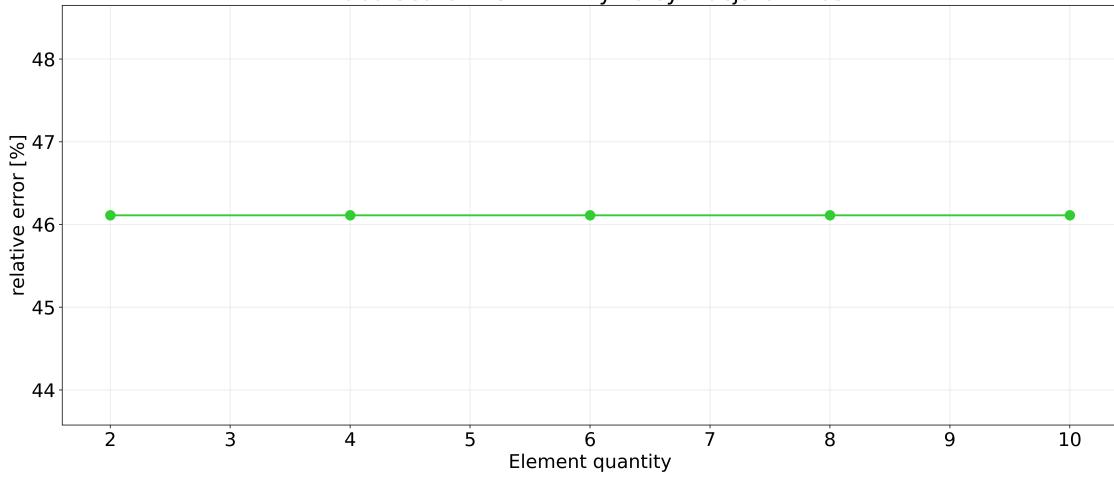
Simulated Annealing - REVERSE - 65



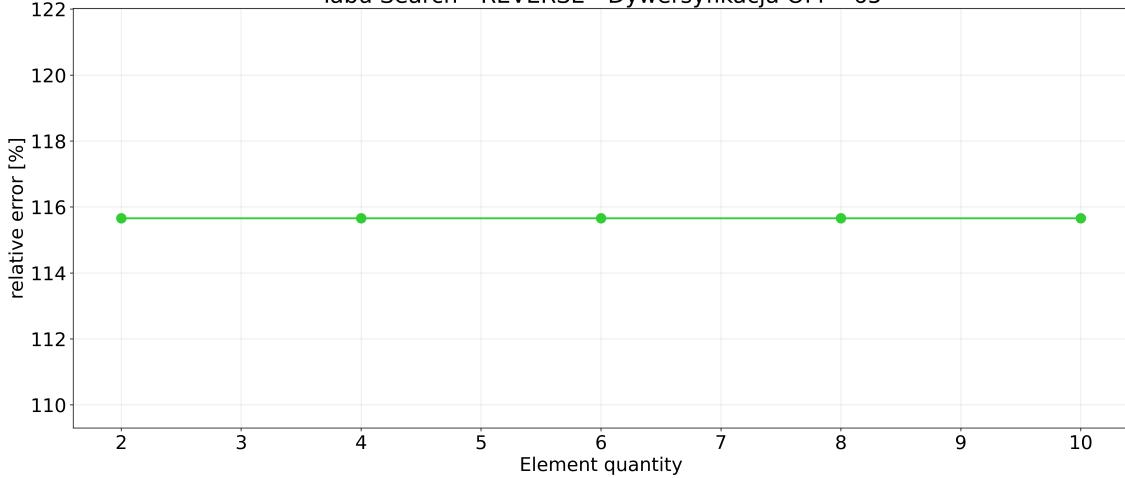
Tabu Search - SWAP - Dywersyfikacja OFF - 65



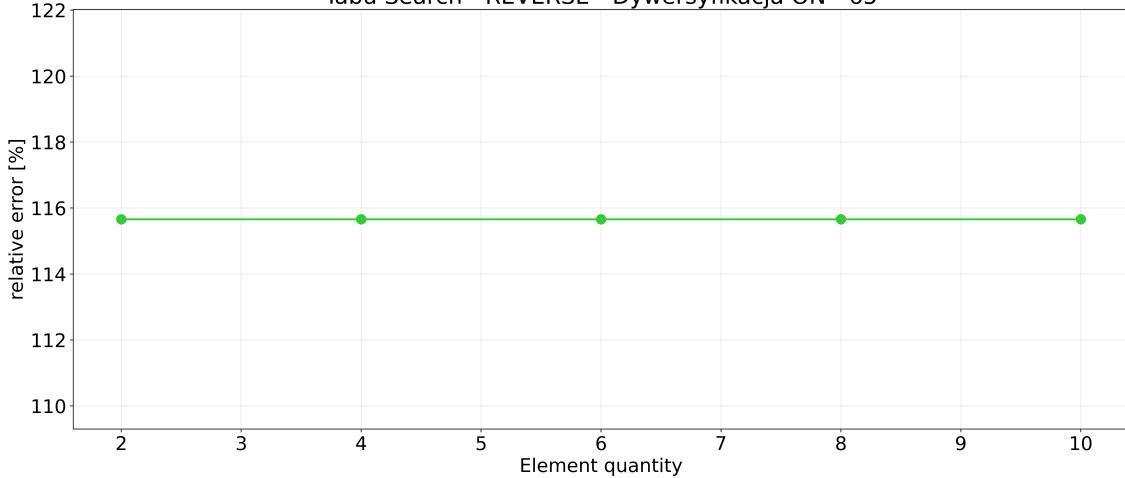
Tabu Search - SWAP - Dywersyfikacja ON - 65



Tabu Search - REVERSE - Dywersyfikacja OFF - 65



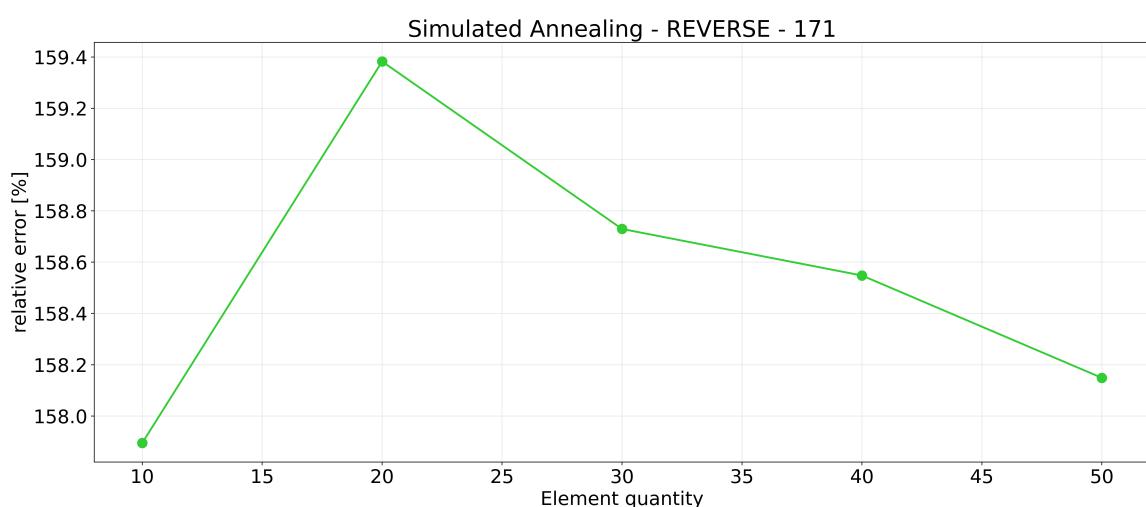
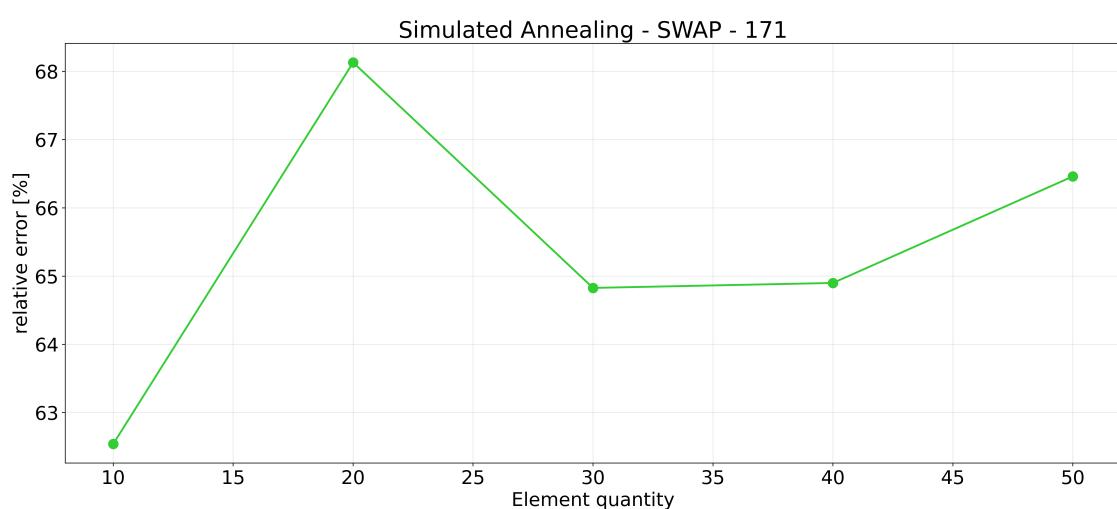
Tabu Search - REVERSE - Dywersyfikacja ON - 65



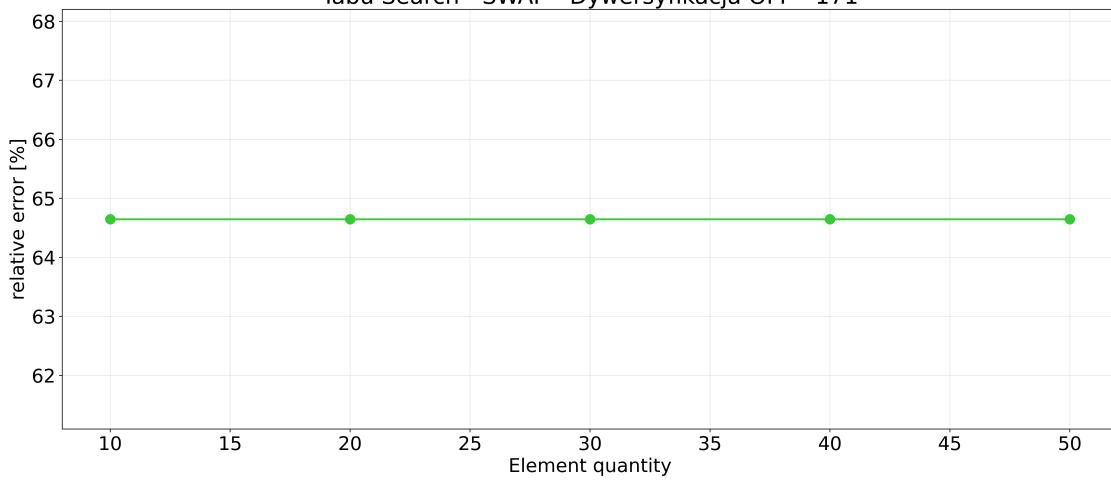
Rozmiar: 171

czas [s]	Simulated Annealing - SWAP	Simulated Annealing - REVERSE	Tabu Search - SWAP - Dywersyfikacja OFF	Tabu Search - SWAP - Dywersyfikacja ON	Tabu Search - REVERSE - Dywersyfikacja OFF	Tabu Search - REVERSE - Dywersyfikacja ON
10	62.541	157.895	64.646	64.646	86.715	86.715

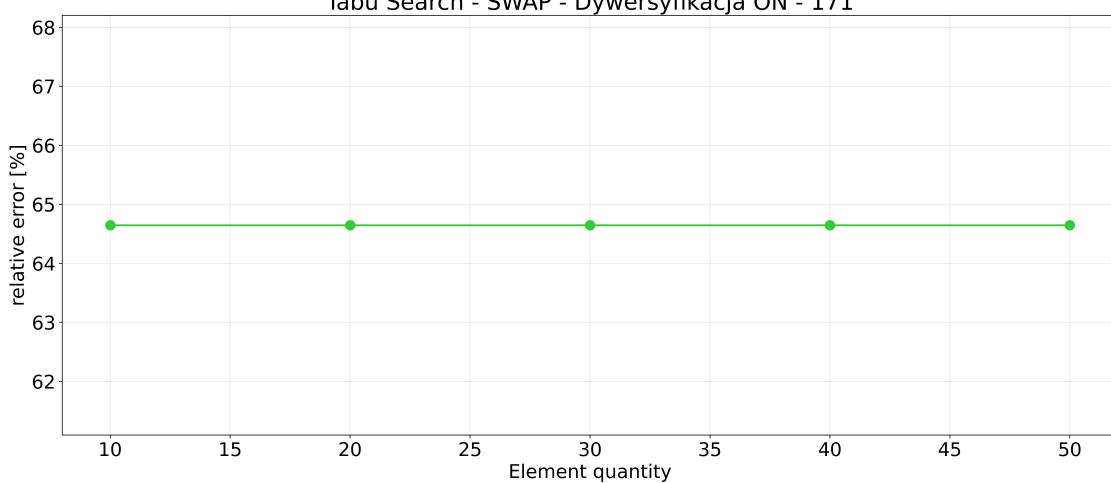
czas [s]	Simulated Annealing - SWAP	Simulated Annealing - REVERSE	Tabu Search - SWAP - Dyweryfikacja OFF	Tabu Search - SWAP - Dyweryfikacja ON	Tabu Search - REVERSE - Dyweryfikacja OFF	Tabu Search - REVERSE - Dyweryfikacja ON
20	68.131	159.383	64.646	64.646	86.715	86.715
30	64.828	158.730	64.646	64.646	86.715	86.715
40	64.9	158.548	64.646	64.646	86.715	86.715
50	66.461	158.149	64.646	64.646	86.715	86.715



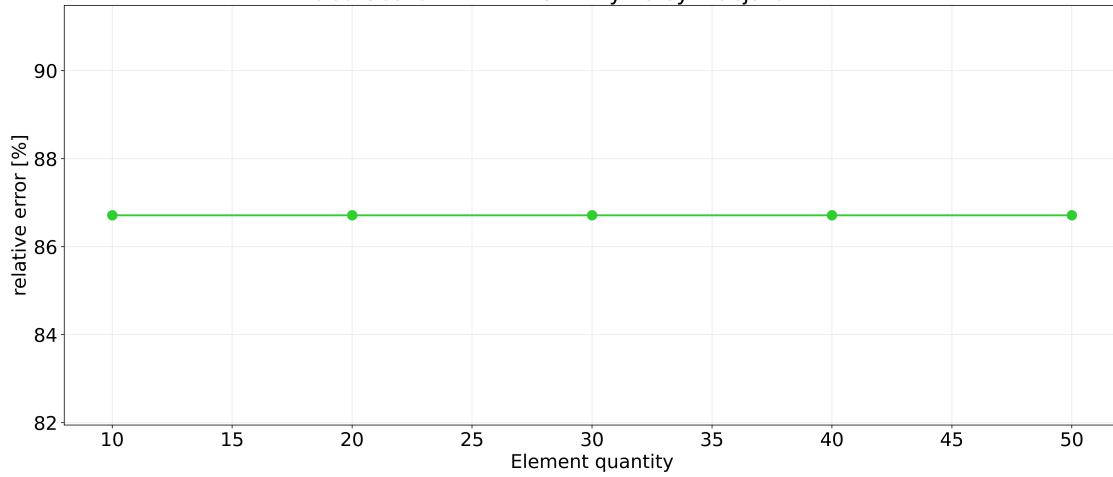
Tabu Search - SWAP - Dywersyfikacja OFF - 171



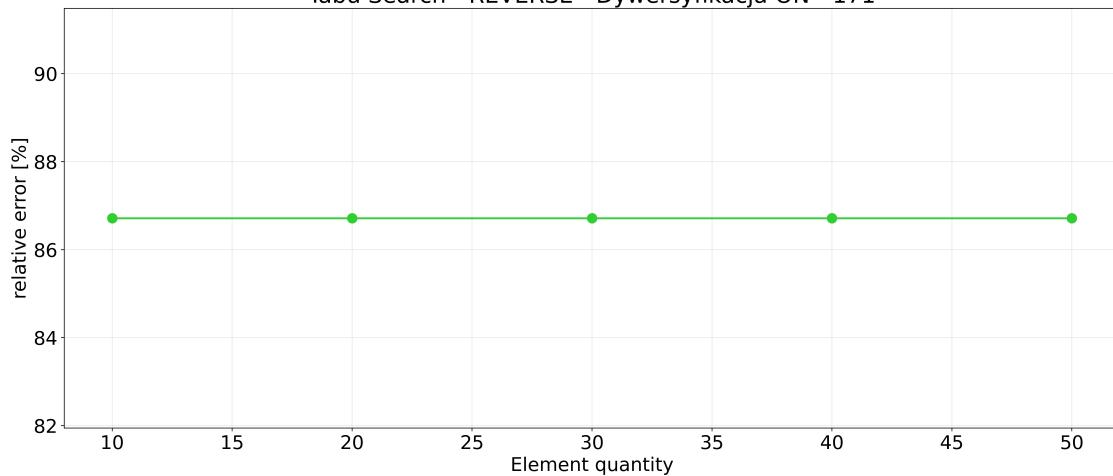
Tabu Search - SWAP - Dywersyfikacja ON - 171



Tabu Search - REVERSE - Dywersyfikacja OFF - 171



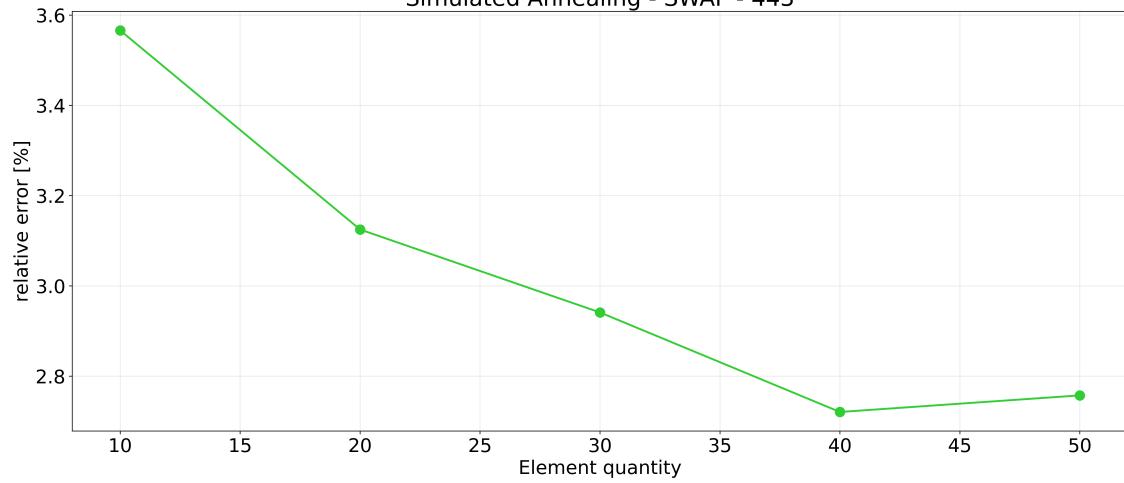
Tabu Search - REVERSE - Dywersyfikacja ON - 171



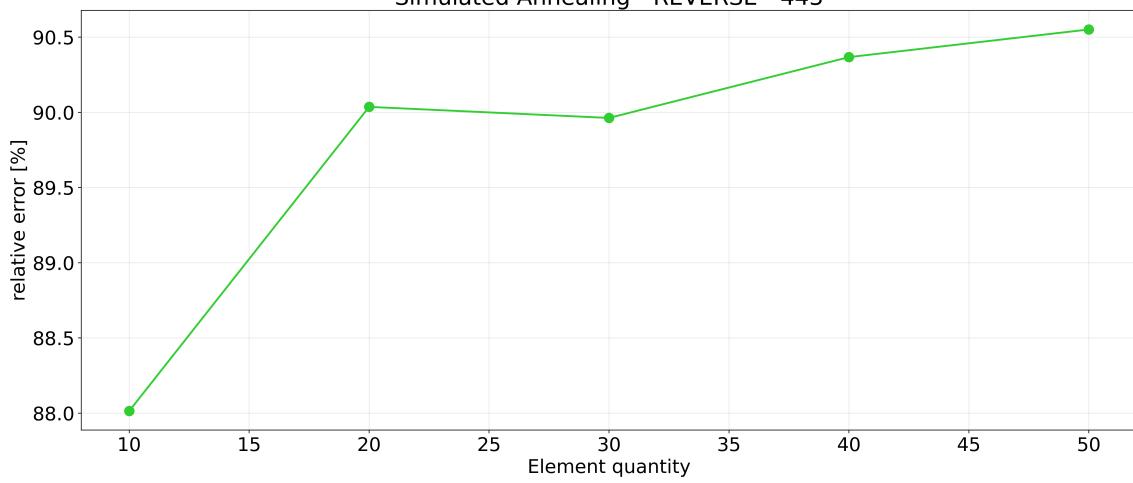
Rozmiar: 443

czas [s]	Simulated Annealing - SWAP	Simulated Annealing - REVERSE	Tabu Search - SWAP - Dywersyfikacja OFF	Tabu Search - SWAP - Dywersyfikacja ON	Tabu Search - REVERSE - Dywersyfikacja OFF	Tabu Search - REVERSE - Dywersyfikacja ON
10	3.566	88.015	195.037	195.257	196.838	196.654
20	3.125	90.037	177.647	177.243	176.029	174.154
30	2.941	89.963	159.154	159.118	153.750	153.051
40	2.721	90.368	141.544	141.581	137.574	140.662
50	2.757	90.551	128.199	126.066	123.529	123.309

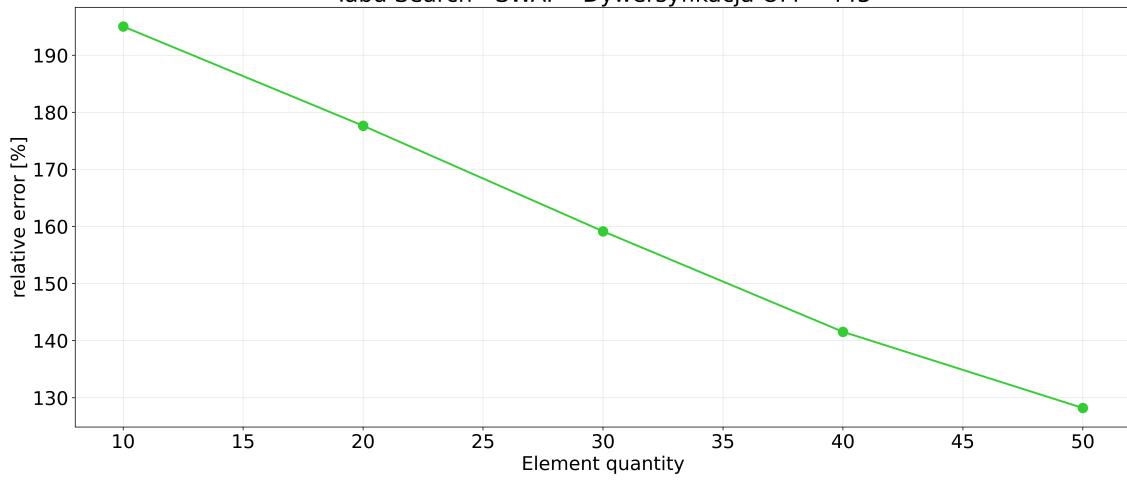
Simulated Annealing - SWAP - 443



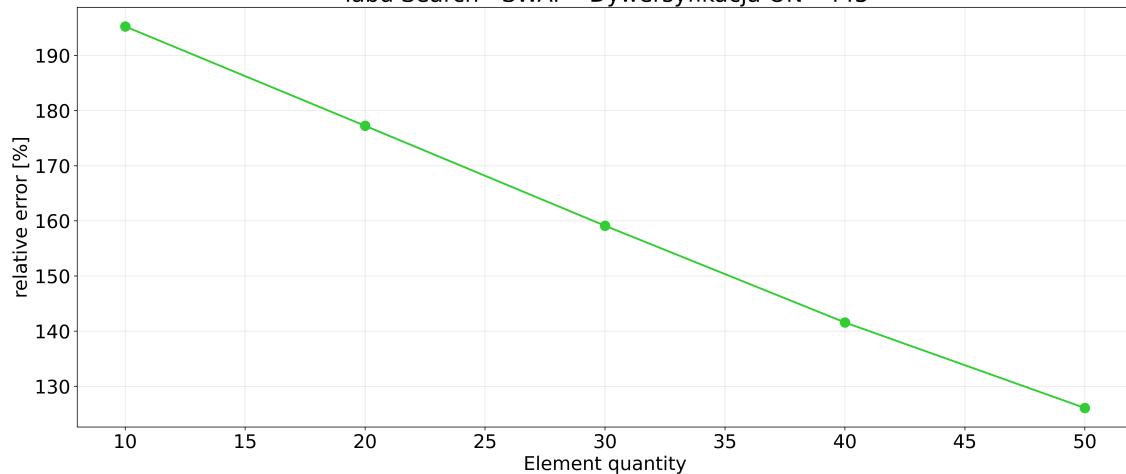
Simulated Annealing - REVERSE - 443



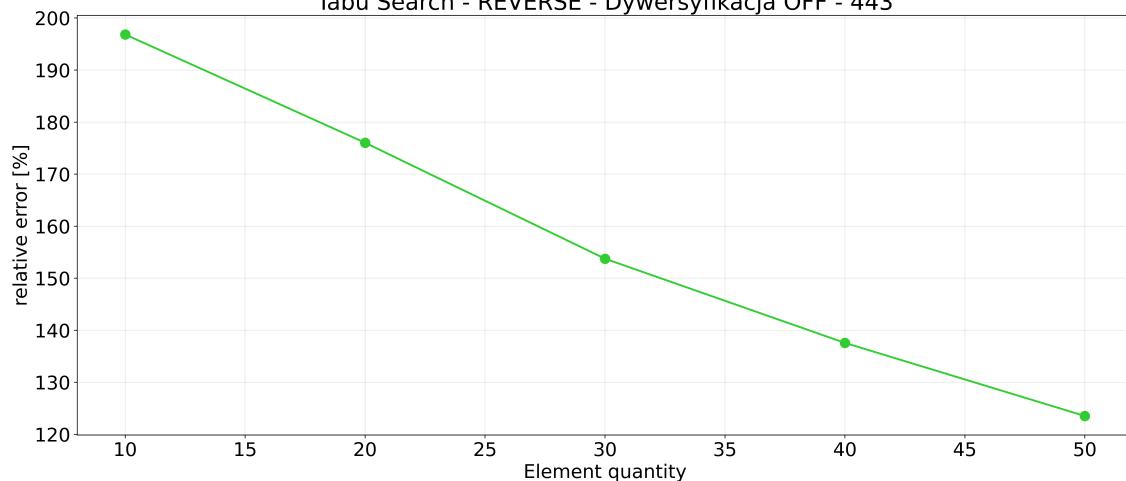
Tabu Search - SWAP - Dywersyfikacja OFF - 443



Tabu Search - SWAP - Dywersyfikacja ON - 443



Tabu Search - REVERSE - Dywersyfikacja OFF - 443



Tabu Search - REVERSE - Dywersyfikacja ON - 443

