Projektowanie efektywnych algorytmów
Projekt – zadanie 2

Algorytm symulowanego wyżarzania oraz tabu search

Problem komiwojażera

Autor: Filip Przygoński, 248892
Prowadzący: mgr inż. Antoni Sterna
Grupa zajęciowa: śr. 13:15

Wstęp teoretyczny

Problem komiwojażera (Travelling Salesman Problem - TSP)

W problemie komiwojażera dane jest n miast. Komiwojażer zaczyna w pierwszym mieście, odwiedza każde inne miasto jeden raz i wraca do pierwszego miasta. Rozwiązaniem problemu jest najkrótsza droga spełniająca powyższy warunek.

Patrząc informatycznie, problem można zobrazować jako ważony, skierowany (jeśli 'koszt' przejazdu z miasta 1 do miasta 2 jest różny od przejazdu z miasta 2 do miasta 1), graf pełny, gdzie wierzchołkami grafu są miasta.

Symulowane wyżarzanie (SA)

Generujemy pierwsze rozwiązanie, np. losowo. Ustalamy początkowe parametry takie jak temperatura początkowa czy funkcja zmiany temperatury. Następnie, dopóki nie zostanie osiągnięty warunek zatrzymania (wystarczająco niska temperatura lub przekroczony maksymalny czas), generujemy losowe rozwiązanie z sąsiedztwa obecnego rozwiązania i sprawdzamy czy wygenerowane rozwiązanie jest lepsze od obecnego. Jeśli tak, przyjmujemy je jako obecne. Jeśli nie, przyjmujemy je jako obecne z prawdopodobieństwem zależnym od temperatury – im wyższa temperatura, tym większe prawdopodobieństwo przyjęcia gorszego rozwiązania. Po sprawdzeniu rozwiązania zmniejszamy temperaturę ustaloną funkcją.

Wybrana funkcja obniżania temperatury to redukcja geometryczna: $T_{next} = 0{,}99 \cdot T_{prev}$

Temperatura początkowa to: $liczba\ miast \cdot długość\ drogi\ pierwszego\ rozwiązania$

W danej temperaturze testowane jest wiele rozwiązań: $liczba\ miast \cdot 5$, chyba że znaleziono najlepsze rozwiązanie, wtedy nie zmniejszamy temperatury.

Warunkiem zakończenia algorytmu jest przekroczenie maksymalnego czasu, jeśli podano maksymalny czas, w przeciwnym wypadku warunkiem końca jest temperatura mniejsza od 10^{-9}.

Tabu search (TS)

Generujemy pierwsze rozwiązanie, np. losowo. Następnie, szukamy w sąsiedztwie obecnego rozwiązania najlepszego sąsiada. Jeśli sąsiednie rozwiązanie jest najlepszym dotychczas znalezionym, przyjmujemy je jako obecne nawet jeśli jest zakazane. W przeciwnym wypadku jeśli sąsiednie rozwiązanie jest lepsze od obecnego i nie wymaga zakazanego ruchu, to przyjmujemy je jako obecne. Po przeszukaniu sąsiedztwa, ruch który wykonaliśmy aby przejść z poprzedniego rozwiązania do obecnego zapisujemy na liście tabu z kadencją o pewnej długości. Co przeszukiwanie sąsiedztwa każdemu elementowi na liście tabu zmniejszamy kadencję o 1.

Lista tabu: macierz $n \times n$, gdzie $tabuList[i, j]$ to aktualna długość kadencji ruchu dla sąsiedztwa i, j.

Długość kadencji: 10

Dywersyfikacja: jeżeli 10 razy z rzędu nie udało się znaleźć rozwiązania lepszego od obecnie najlepszego, generujemy nowe losowe rozwiązanie i zerujemy listę tabu.

Algorytm jest wykonywany dopóki nie przekroczy maksymalnego czasu.

Typy sąsiedztwa

Zaimplementowane zostały 2 różne rodzaje sąsiedztwa:

- Swap – zamiana elementów na pozycjach i, j
- Reverse – zamiana kolejności elementów między elementami na pozycjach i, j

Przykłady działania sąsiedztw dla $\{0, 1, 2, 3, 4, 5\}, i = 1, j = 4$

- Swap: $\{0, 4, 2, 3, 1, 5\}$
- Reverse: $\{0, 4, 3, 2, 1, 5\}$

Opis klas

- MatrixGraph – klasa reprezentująca graf jako macierz sąsiedztwa.
- Algorithms – klasa statyczna – metody algorytmów, sąsiedztwa oraz metody pomocnicze dla algorytmów.
- Program – klasa statyczna – metody interfejsu użytkownika

Plan eksperymentu

Program został napisany w języku C#, w .NET Framework, w środowisku Visual Studio 2019.

Do mierzenia czasu wykorzystano klasę *Stopwatch* z przestrzeni nazw *System.Diagnostics*:
https://docs.microsoft.com/en-gb/dotnet/api/system.diagnostics.stopwatch

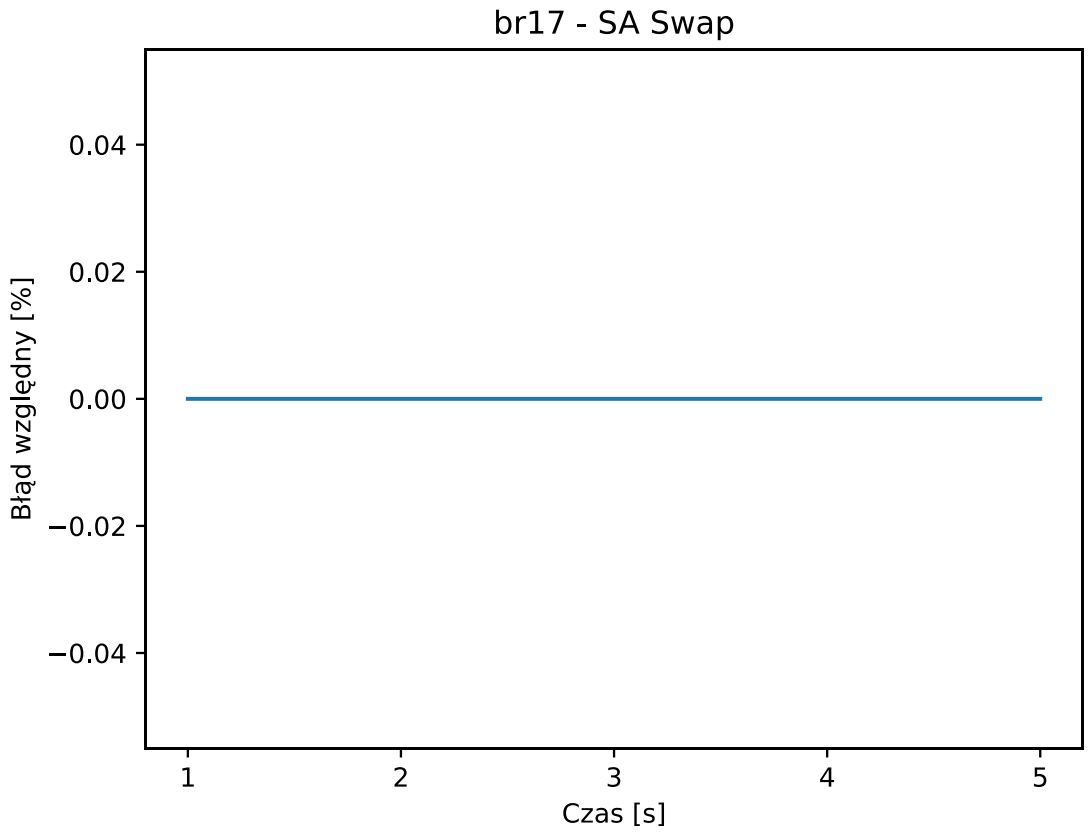
Testowane instancje problemu to *br17, ft70, rbg323.*

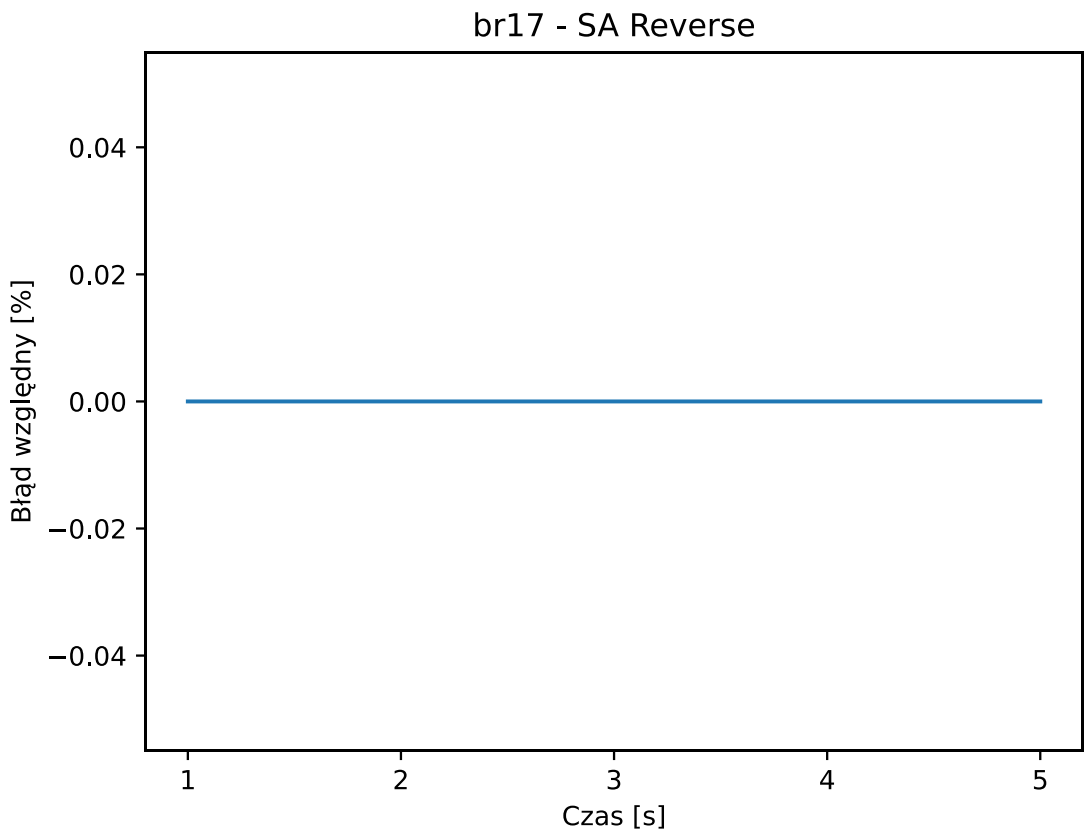
Wyniki eksperymentu

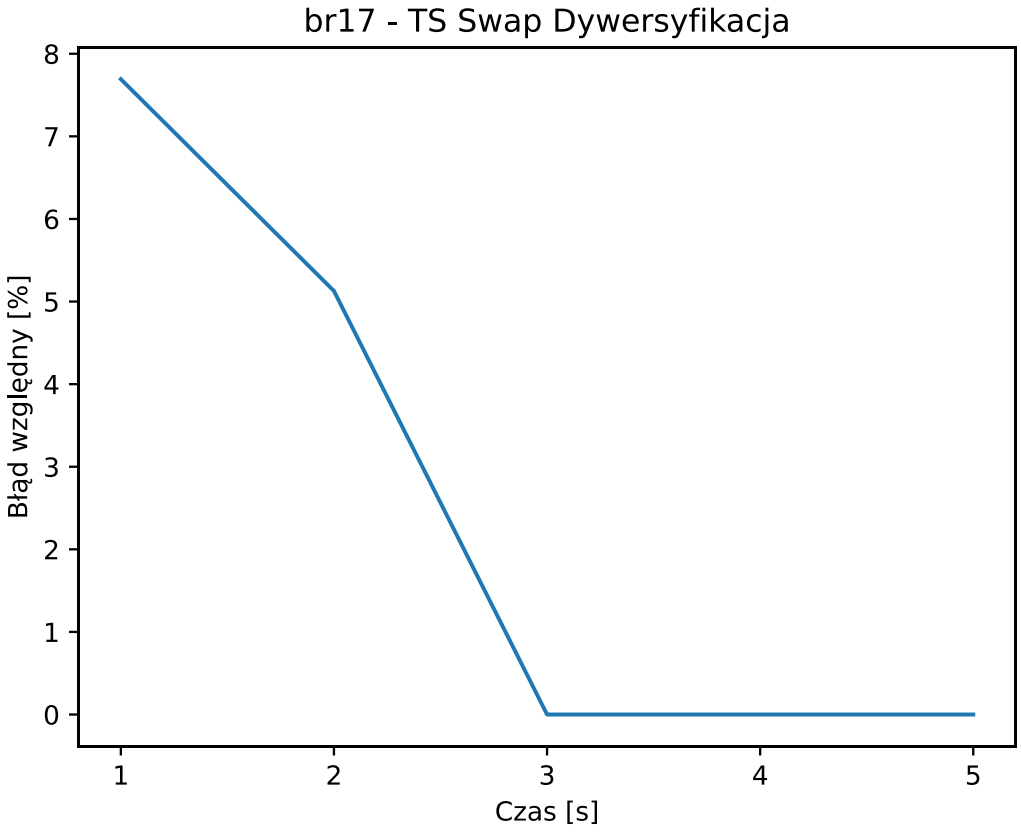
Tytuł wykresu: instancja – algorytm, typ sąsiedztwa, dywersyfikacja (dla tabu)

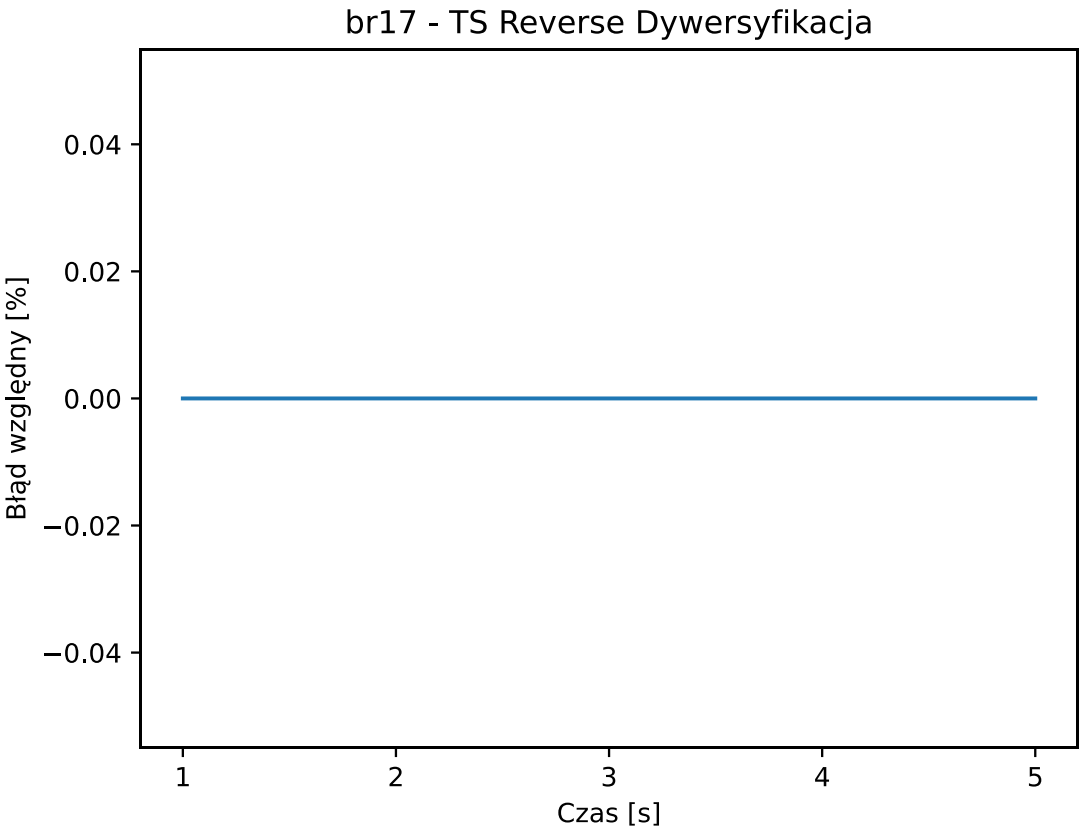
Instancja br17, optymalny wynik: 39

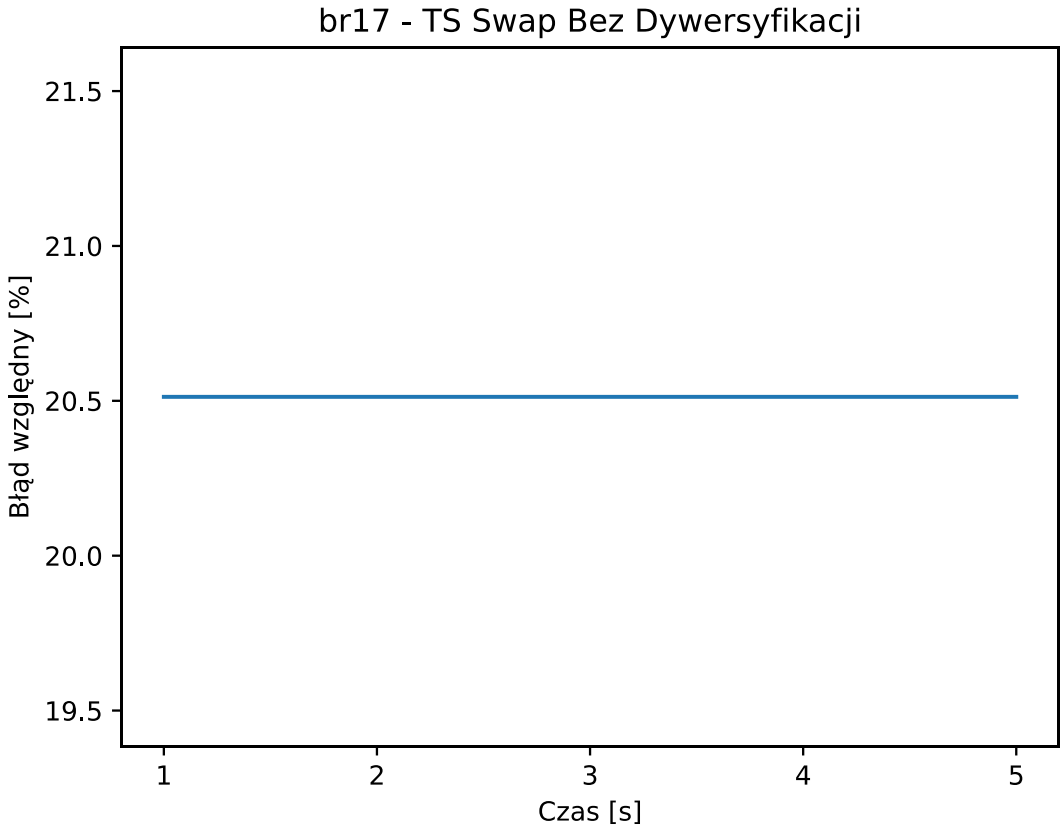
SA Swap	39	39	39	39	39
Błąd względny [%]	0	0	0	0	0
Czas [s]	1	2	3	4	5
SA Reverse	39	39	39	39	39
Błąd względny [%]	0	0	0	0	0
Czas [s]	1	2	3	4	5
TS Swap Dywersyfikacja	42	41	39	39	39
Błąd względny [%]	7,692308	5,128205	0	0	0
Czas [s]	1	2	3	4	5
TS Reverse Dywersyfikacja	39	39	39	39	39
Błąd względny [%]	0	0	0	0	0
Czas [s]	1	2	3	4	5
TS Swap Bez Dywersyfikacji	47	47	47	47	47
Błąd względny [%]	20,51282	20,51282	20,51282	20,51282	20,51282
Czas [s]	1	2	3	4	5
TS Reverse Bez Dywersyfikacji	39	39	39	39	39
Błąd względny [%]	0	0	0	0	0
Czas [s]	1	2	3	4	5

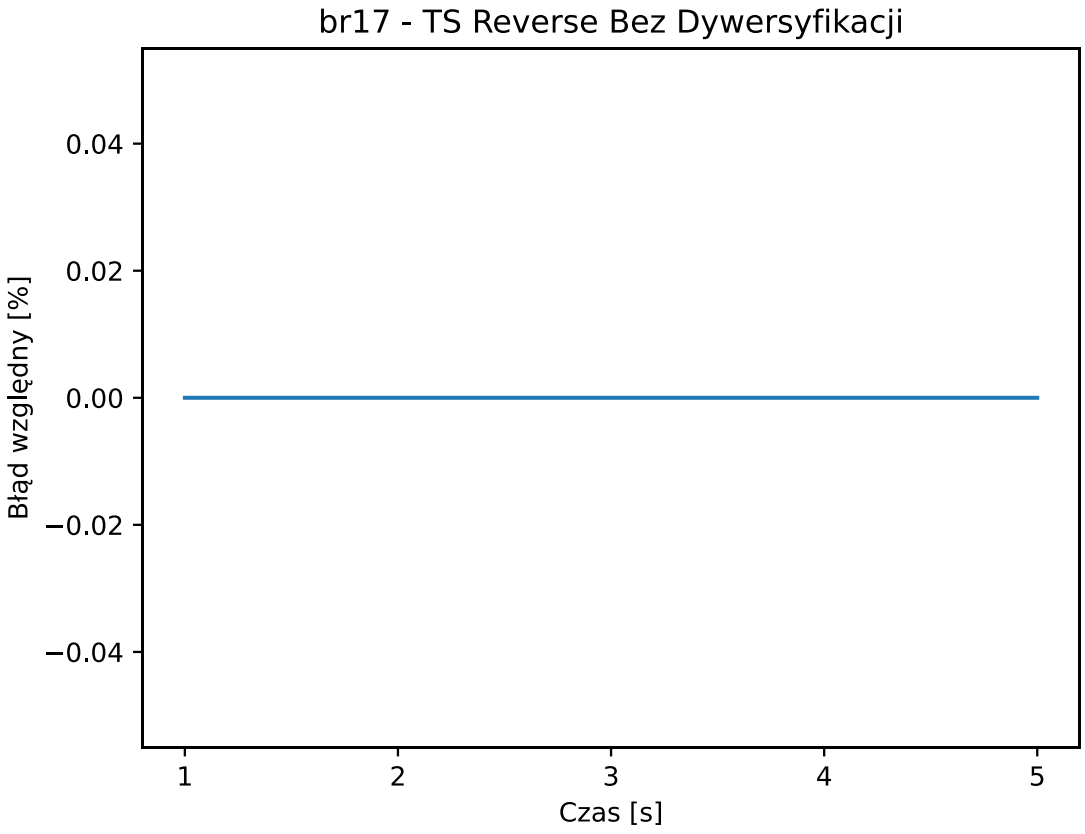

br17 - SA Swap
0.04
0.02
0.00
−0.02
−0.04
Błąd względny [%]
1
2
3
4
5
Czas [s]


br17 - SA Reverse
0.04
0.02
0.00
−0.02
−0.04
Błąd względny [%]
1
2
3
4
5
Czas [s]


br17 - TS Swap Dywersyfikacja
Błąd względny [%]
Czas [s]
0
1
2
3
4
5
6
7
8
1
2
3
4
5

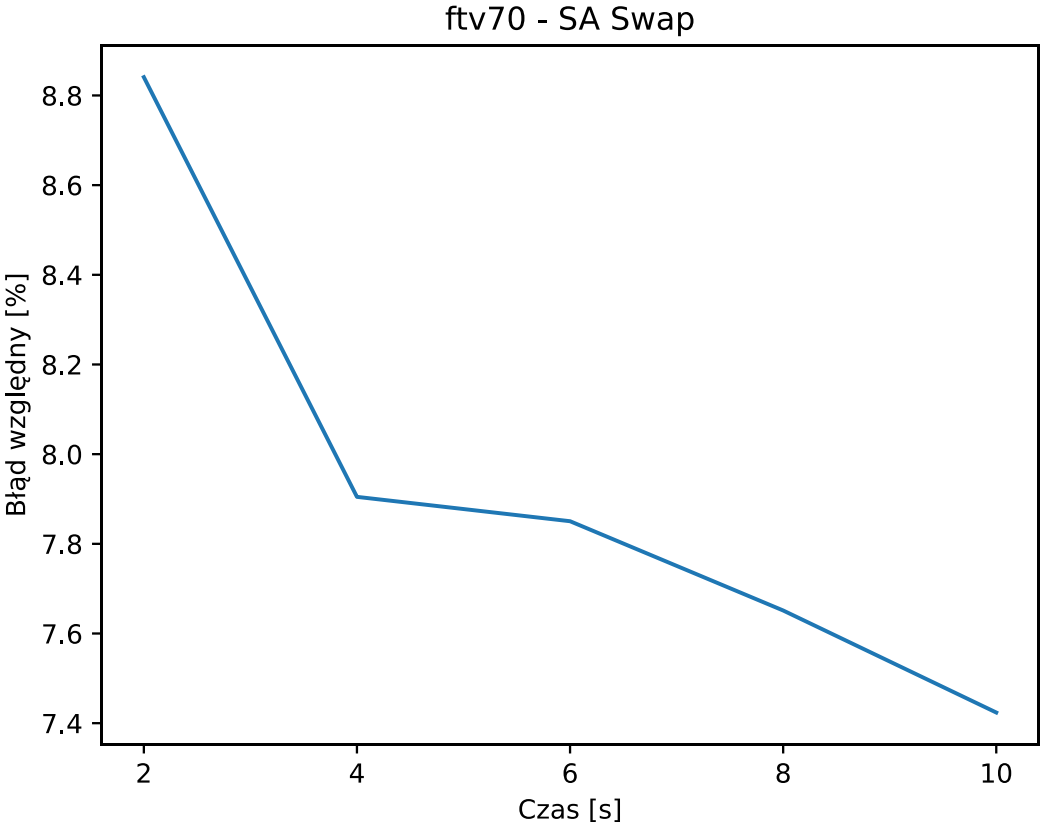

br17 - TS Reverse Dywersyfikacja
Błąd względny [%]
Czas [s]
0.04
0.02
0.00
−0.02
−0.04
1
2
3
4
5

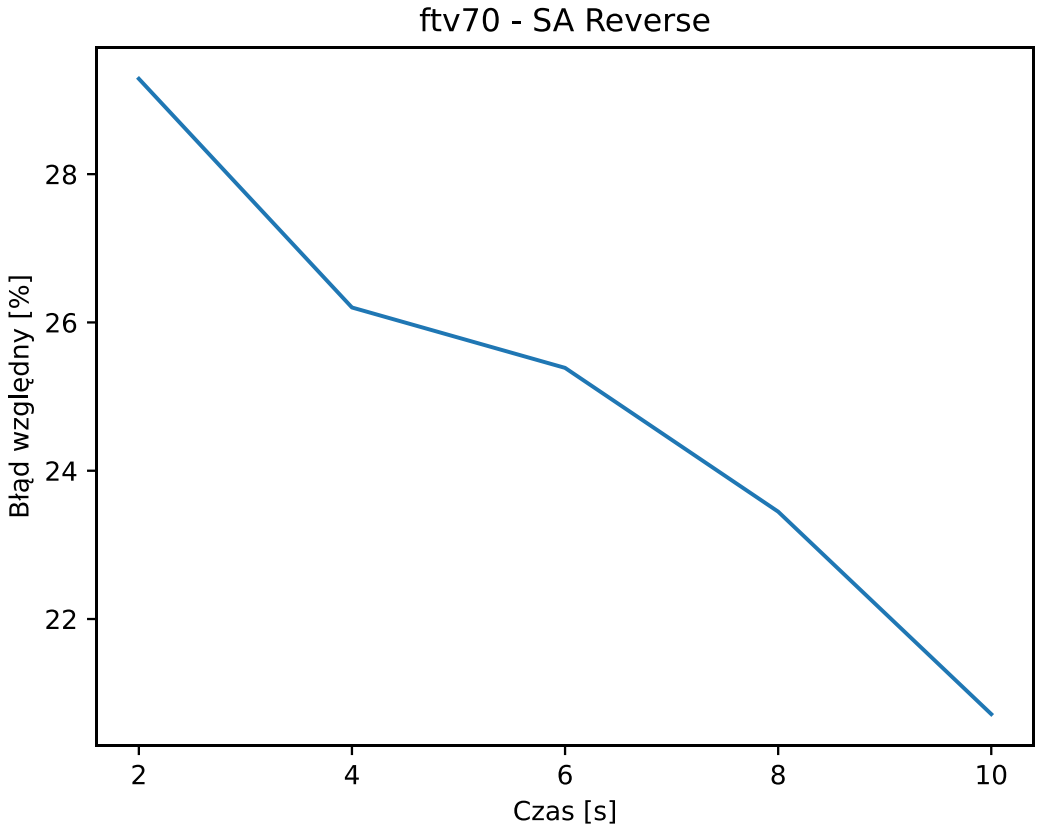

br17 - TS Swap Bez Dywersyfikacji
Błąd względny [%]
21.5
21.0
20.5
20.0
19.5
1
2
3
4
5
Czas [s]

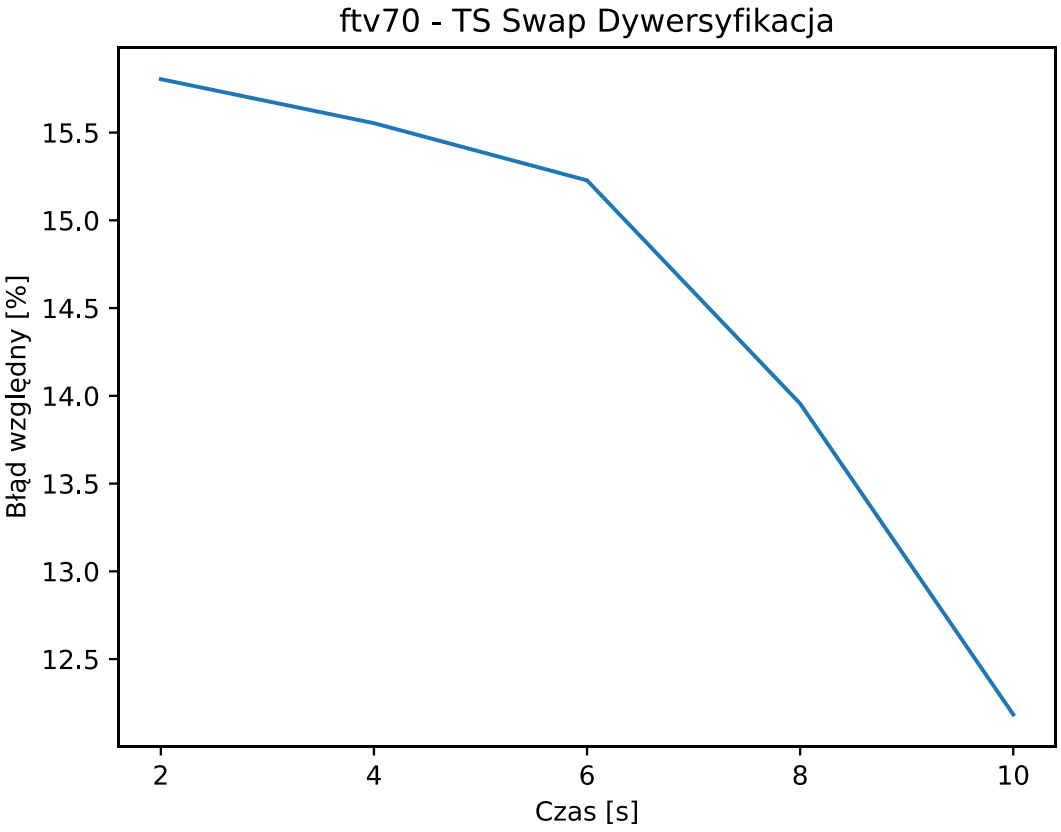

br17 - TS Reverse Bez Dywersyfikacji
Błąd względny [%]
0.04
0.02
0.00
−0.02
−0.04
1
2
3
4
5
Czas [s]

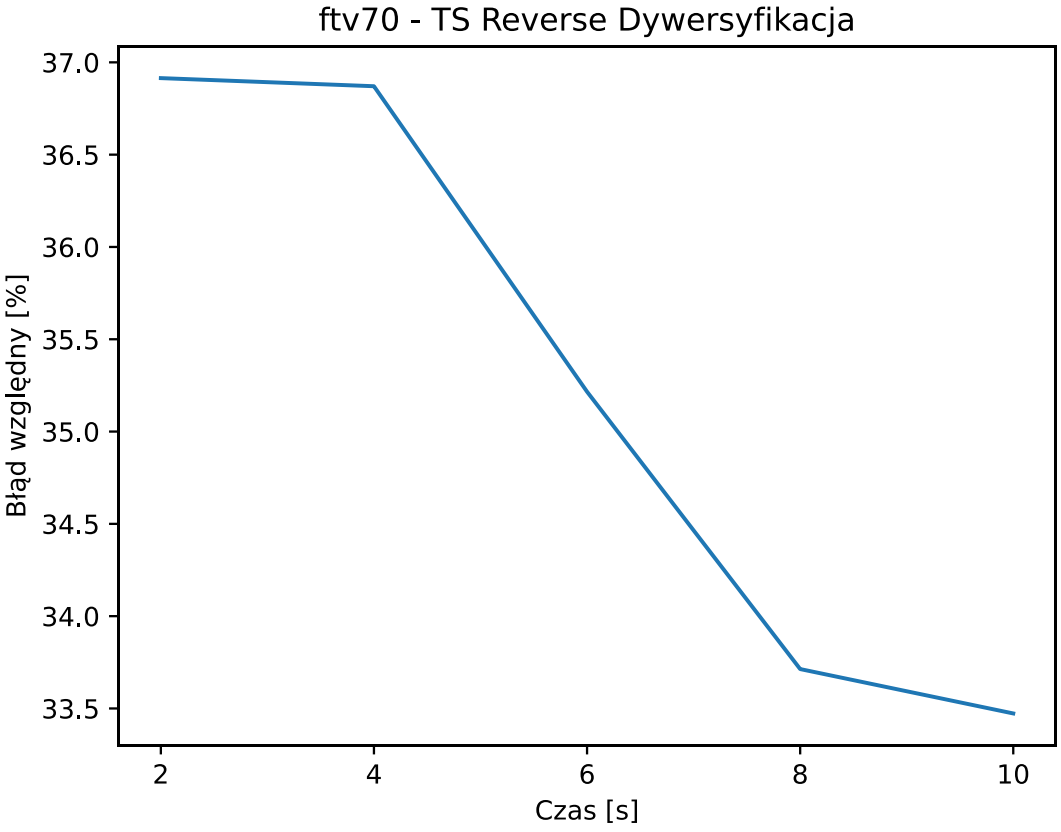
Instancja ftv70, optymalny wynik: 38673

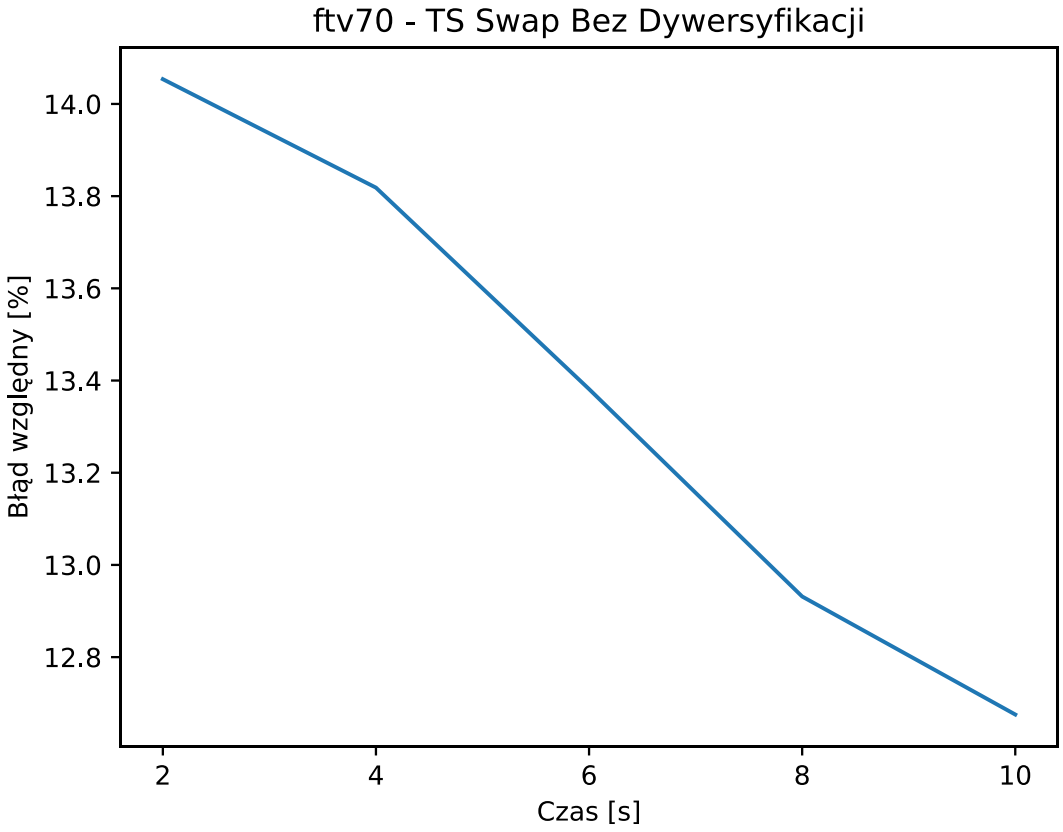
SA Swap	42092	41730	41709	41632	41544
Błąd względny [%]	8,840793	7,90474	7,850438	7,651333	7,423784
Czas [s]	2	4	6	8	10
SA Reverse	49999	48806	48491	47741	46685
Błąd względny [%]	29,28658	26,20174	25,38722	23,44788	20,7173
Czas [s]	2	4	6	8	10
TS Swap Dywersyfikacja	44785	44688	44562	44070	43385
Błąd względny [%]	15,80431	15,55349	15,22768	13,95547	12,18421
Czas [s]	2	4	6	8	10
TS Reverse Dywersyfikacja	52949	52932	52292	51711	51618
Błąd względny [%]	36,91464	36,87068	35,21578	33,71344	33,47297
Czas [s]	2	4	6	8	10
TS Swap Bez Dywersyfikacji	44108	44017	43848	43674	43575
Błąd względny [%]	14,05373	13,81843	13,38143	12,9315	12,67551
Czas [s]	2	4	6	8	10
TS Reverse Bez Dywersyfikacji	54897	53865	53821	53705	52887
Błąd względny [%]	41,95175	39,28322	39,16945	38,8695	36,75432
Czas [s]	2	4	6	8	10

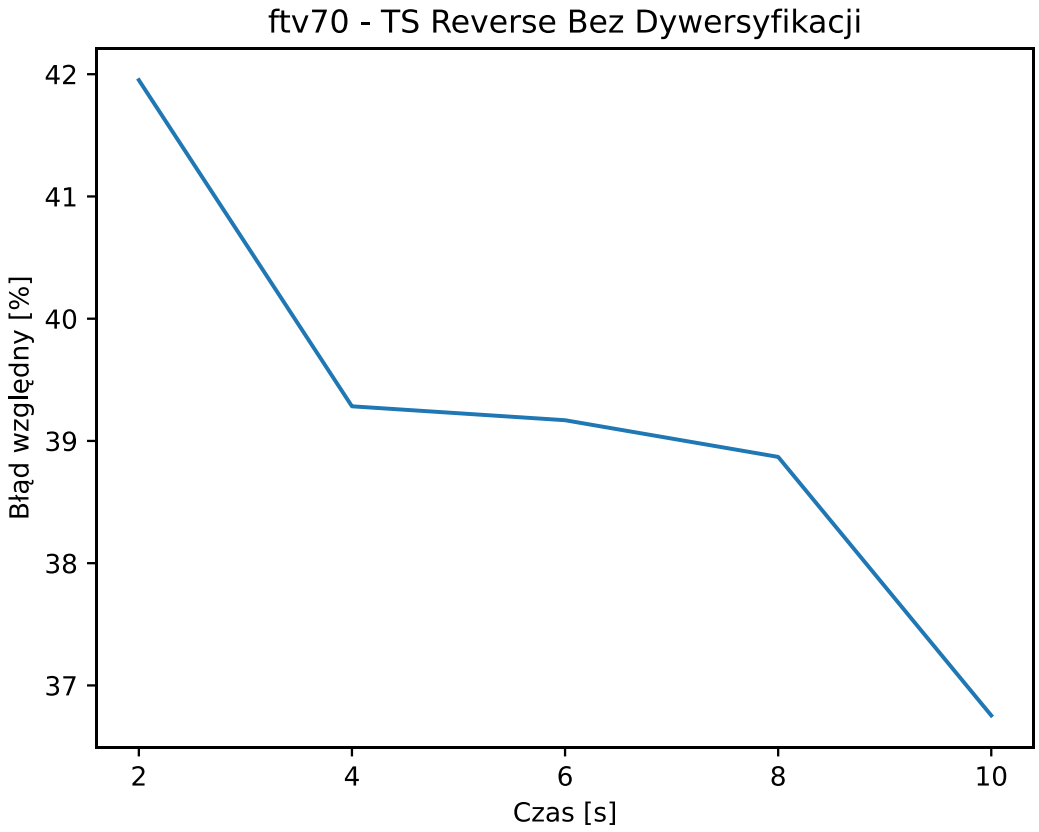

ftv70 - SA Swap
Błąd względny [%]
8.8
8.6
8.4
8.2
8.0
7.8
7.6
7.4
2
4
6
8
10
Czas [s]


ftv70 - SA Reverse
Błąd względny [%]
28
26
24
22
2
4
6
8
10
Czas [s]


ftv70 - TS Swap Dywersyfikacja
Błąd względny [%]
15.5
15.0
14.5
14.0
13.5
13.0
12.5
2
4
6
8
10
Czas [s]

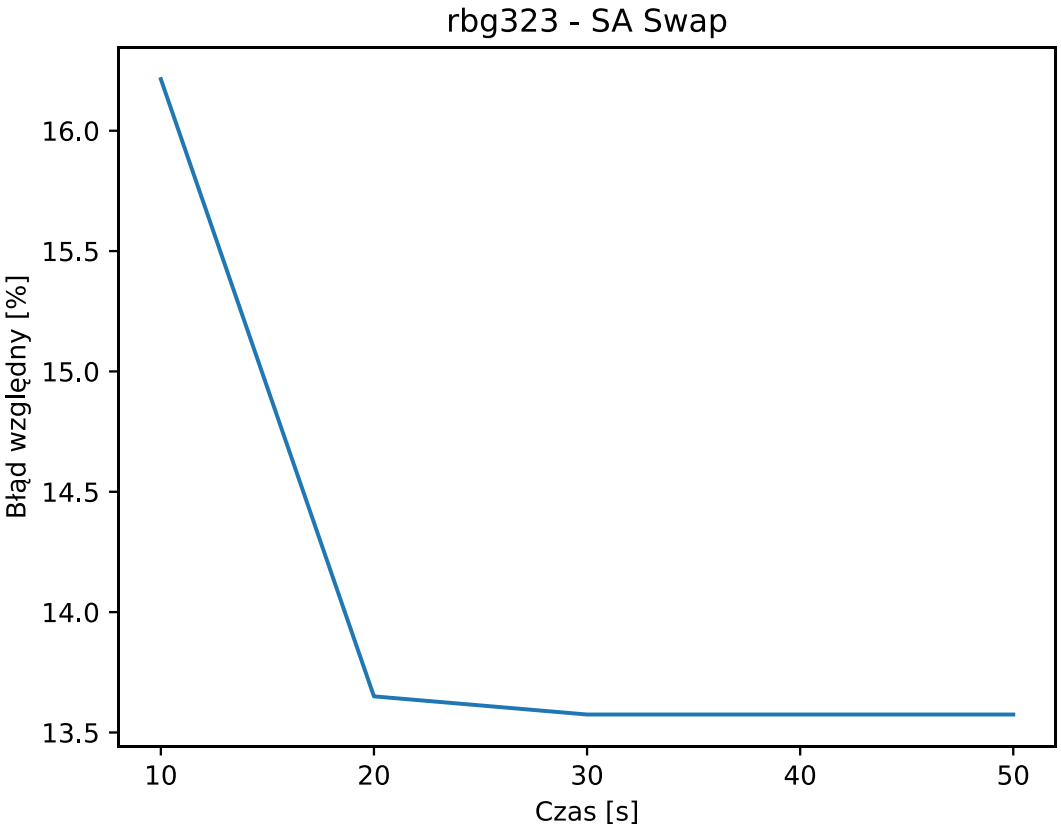

ftv70 - TS Reverse Dywersyfikacja
Błąd względny [%]
37.0
36.5
36.0
35.5
35.0
34.5
34.0
33.5
2
4
6
8
10
Czas [s]

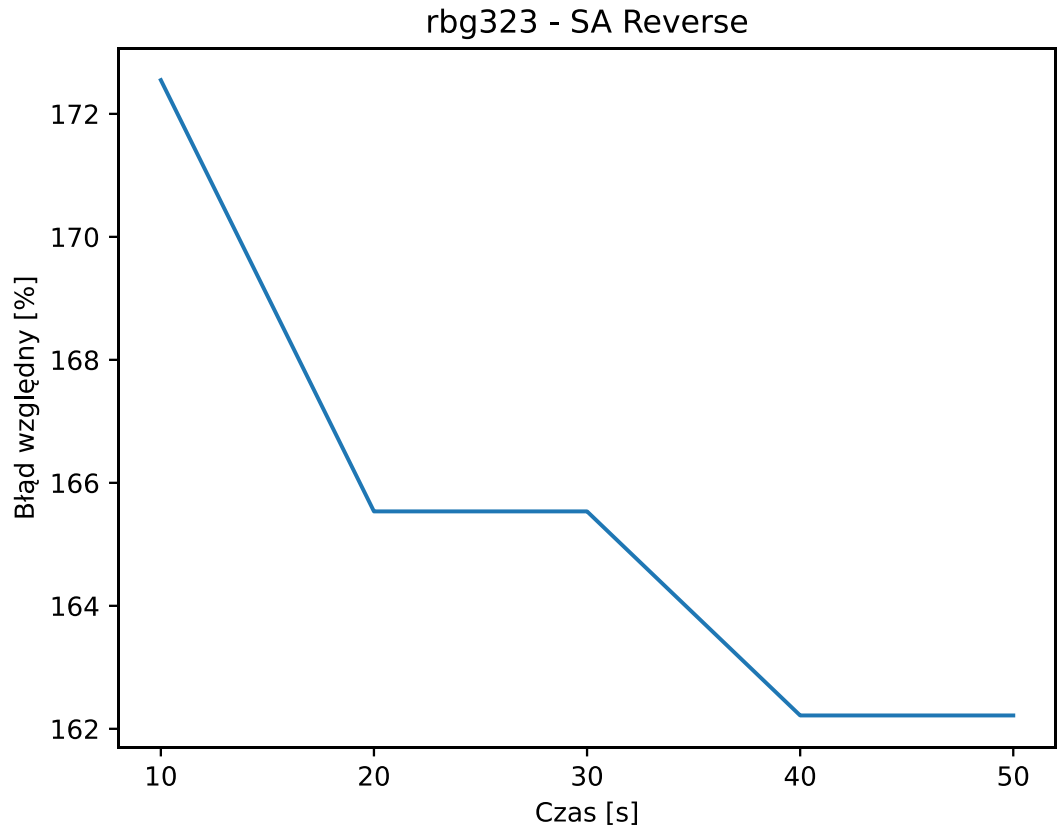

ftv70 - TS Swap Bez Dywersyfikacji
Błąd względny [%]
14.0
13.8
13.6
13.4
13.2
13.0
12.8
2
4
6
8
10
Czas [s]

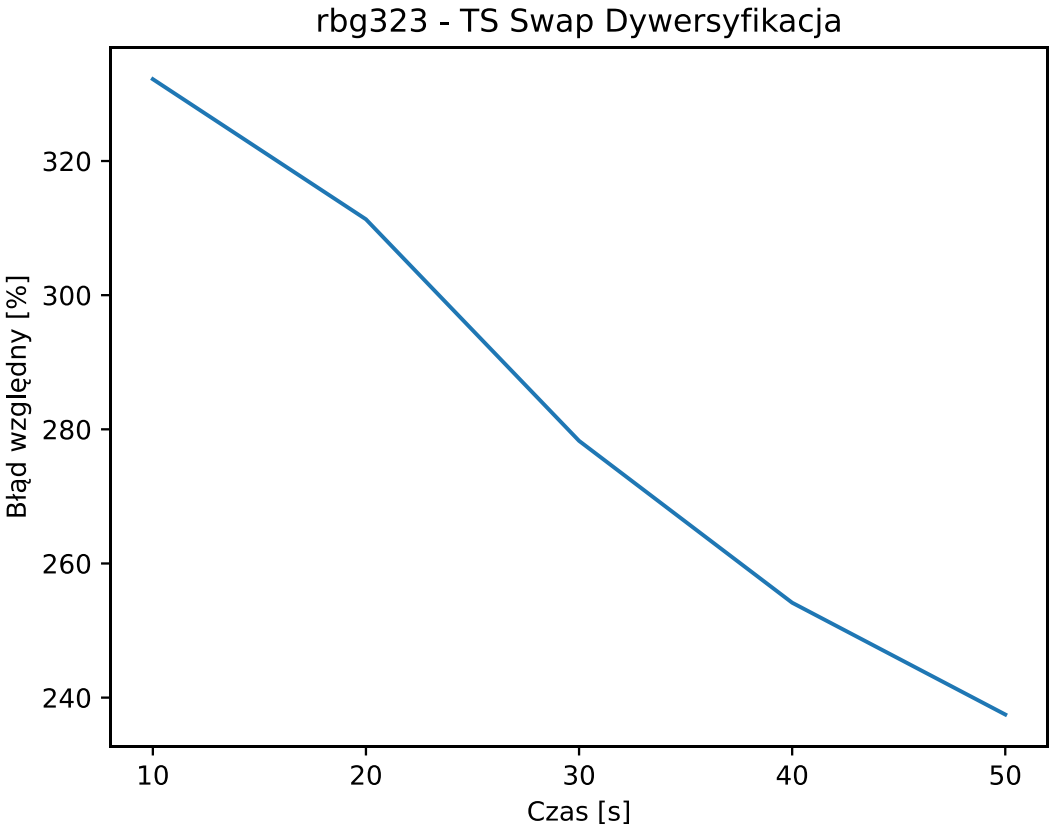

ftv70 - TS Reverse Bez Dywersyfikacji
Błąd względny [%]
42
41
40
39
38
37
2
4
6
8
10
Czas [s]

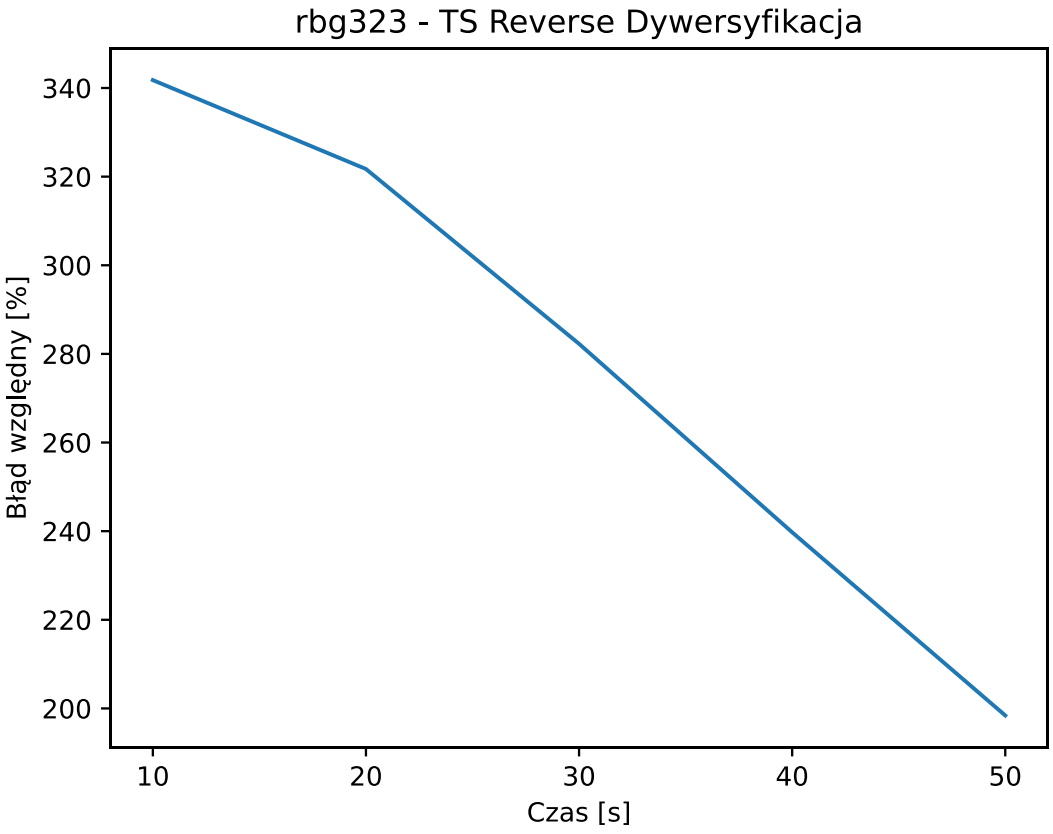
Instancja rbg323, optymalny wynik: 1326

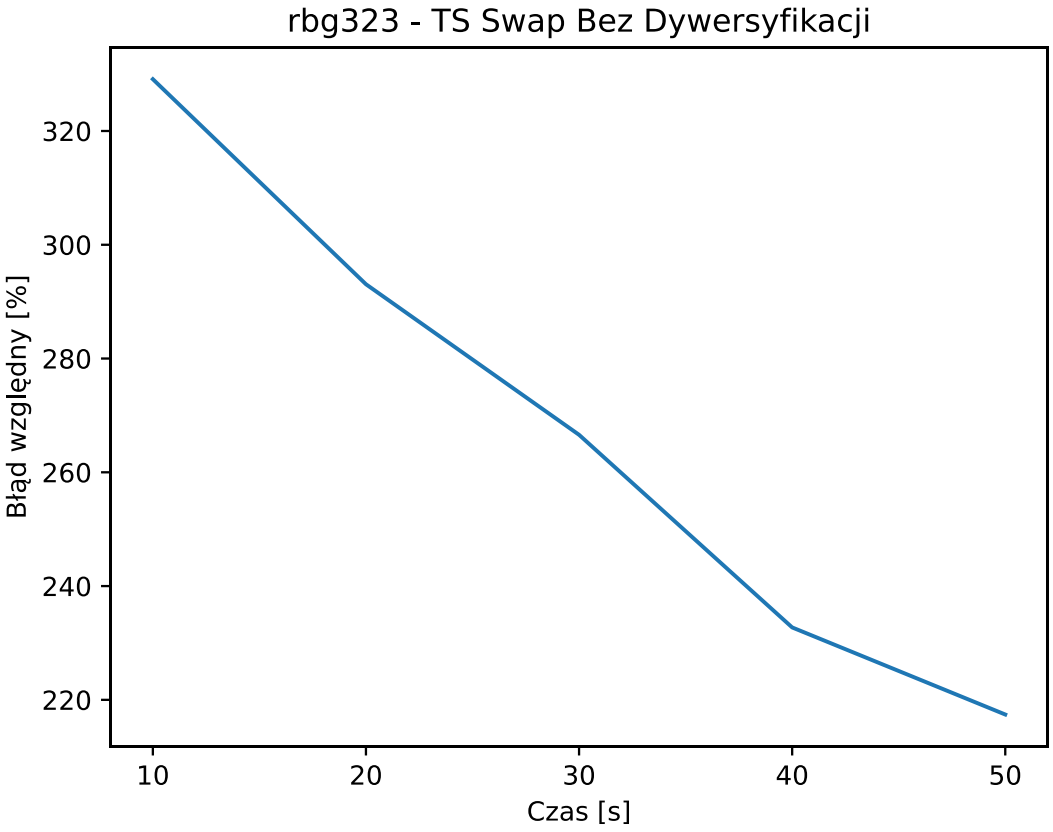
SA Swap	1541	1507	1506	1506	1506
Błąd względny [%]	16,21418	13,65008	13,57466	13,57466	13,57466
Czas [s]	10	20	30	40	50
SA Reverse	3614	3521	3521	3477	3477
Błąd względny [%]	172,549	165,5354	165,5354	162,2172	162,2172
Czas [s]	10	20	30	40	50
TS Swap Dywersyfikacja	5731	5454	5016	4696	4475
Błąd względny [%]	332,2021	311,3122	278,2805	254,1478	237,4811
Czas [s]	10	20	30	40	50
TS Reverse Dywersyfikacja	5858	5592	5069	4505	3957
Błąd względny [%]	341,7798	321,7195	282,2775	239,7436	198,4163
Czas [s]	10	20	30	40	50
TS Swap Bez Dywersyfikacji	5690	5212	4861	4412	4209
Błąd względny [%]	329,1101	293,0618	266,5913	232,73	217,4208
Czas [s]	10	20	30	40	50
TS Reverse Bez Dywersyfikacji	5929	5704	4947	4552	4250
Błąd względny [%]	347,1342	330,1659	273,0769	243,2881	220,5128
Czas [s]	10	20	30	40	50

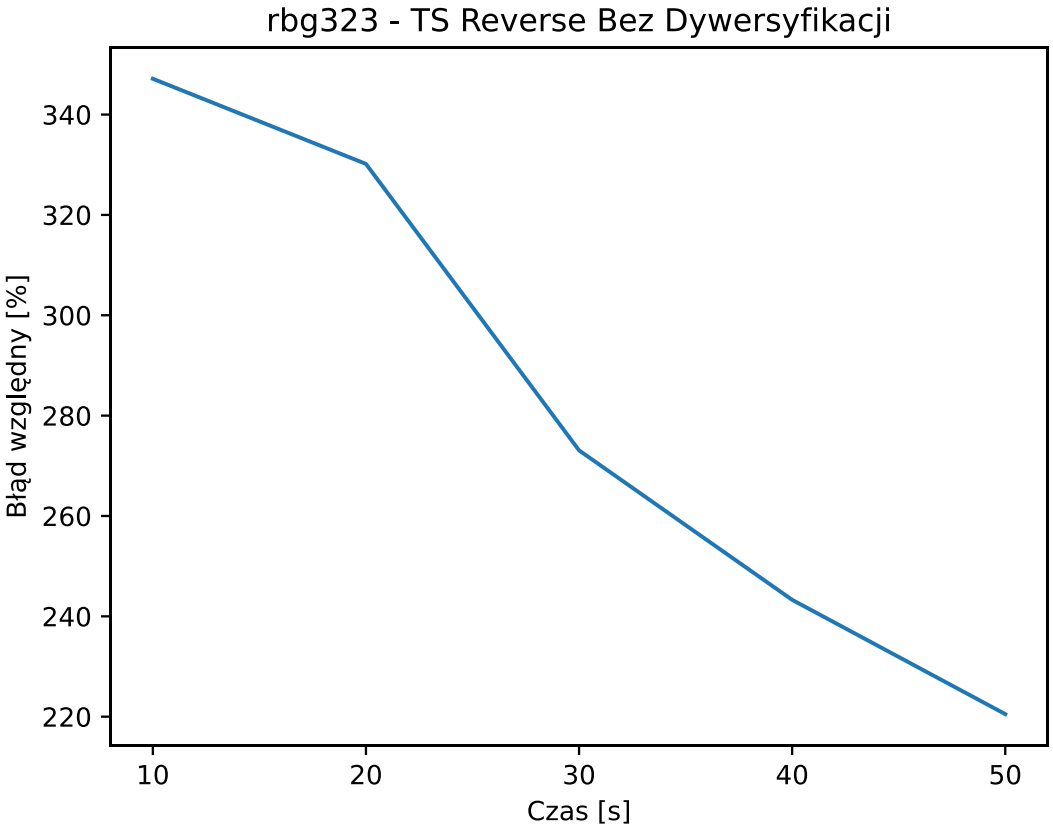

rbg323 - SA Swap
Błąd względny [%]
16.0
15.5
15.0
14.5
14.0
13.5
10
20
30
40
50
Czas [s]


rbg323 - SA Reverse
Błąd względny [%]
172
170
168
166
164
162
10
20
30
40
50
Czas [s]


rbg323 - TS Swap Dywersyfikacja
Błąd względny [%]
320
300
280
260
240
10
20
30
40
50
Czas [s]


rbg323 - TS Reverse Dywersyfikacja
Błąd względny [%]
340
320
300
280
260
240
220
200
10
20
30
40
50
Czas [s]


rbg323 - TS Swap Bez Dywersyfikacji
Błąd względny [%]
320
300
280
260
240
220
10
20
30
40
50
Czas [s]


rbg323 - TS Reverse Bez Dywersyfikacji
Błąd względny [%]
340
320
300
280
260
240
220
10
20
30
40
50
Czas [s]

Wnioski

Funkcja błędu względnego od czasu jest nierosnąca, co nie powinno dziwić.

Lepszym algorytmem okazało się symulowane wyżarzanie.

Lepszym sąsiedztwem zazwyczaj był Swap, choć dla największej instancji dla algorytmu Tabu Search efektywniejszy był Reverse.

Brak dywersyfikacji dla Tabu Search może sprawić, że algorytm nie jest w stanie wyjść z minimum lokalnego.

Brak dywersyfikacji mniej wpływał na sąsiedztwo Reverse. Prawdopodobnie dlatego, że Reverse „bardziej zmienia ścieżkę" od Swap. (Swap zamienia tylko 2 elementy ze sobą, podczas gdy Reverse może odwrócić kolejność nawet całej permutacji).

Źródła

1. https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_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.zio.iiar.pwr.wroc.pl/pea/w5_ts.pdf