

Wrocław, 20.12.2019r.
Prowadzący: dr inż. Zbigniew Buchalski

Projektowanie efektywnych algorytmów

Projekt №2

Implementacja i analiza efektywności algorytmu Tabu Search i/lub
Symulowanego Wyżarzania dla problemu komiwojażera

Nikita Stepanenko
nr albumu 245816

1. Założenia projektowe

Zadaniem projektowym była implementacja dla problemu komiwojażera algorytmów Tabu Search i Symulowanego Wyżarzania oraz porównanie efektywności obu algorytmów, a także wyznaczenie błędu względnego (porównując z najlepszym znanym rozwiązaniem).

Problem komiwojażera jest jednym z najbardziej znanych i wymagających zagadnień optymalizacyjnych dla informatyka. Polega on na znalezieniu ścieżki o najmniejszej sumie wag przechodzącej przez każdy wierzchołek grafu dokładnie jeden raz i wracającej do wierzchołka początkowego. Jest to problem NP-trudny.

Istota i nazwa problemu bierze się z zagwozдки, jaką komiwojażer może mieć przy układaniu trasy, na której odwiedzi określone z góry miasta – zależy mu bowiem na jak najkrótszej ścieżce. Rozróżniamy symetryczne (*STSP* – ang. *Symmetric Traveling Salesman Problem*) i asymetryczne (*ATSP* – ang. *Asymmetric Traveling Salesman Problem*) przypadki tego problemu. *STSP* polegają na tym, że waga krawędzi od wierzchołka *A* do *B* jest taka sama, jak od wierzchołka *B* do *A*. Asymetryczna wersja nie spełnia warunku dla każdej takiej pary krawędzi.

W tym projekcie przedstawione zostaną rozwiązania za pomocą algorytmu „Tabu Search” oraz „Symulowanego Wyżarzania”.

Te algorytmy dążą do rozwiązania problemu poprzez wykorzystanie pewnych pojęć zdefiniowanych przez dany algorytm. Nie dają one gwarancji znalezienia najlepszego możliwego rozwiązania, ale wymagają znacznie mniejszej złożoności obliczeniowej, co powoduje w wielu wypadkach, że okazują się znacznie użyteczniejsze od dokładnego rozwiązania.

2. Specyfikacja techniczna

- Program zrealizowany w języku programowania C++
- Struktury przechowujące dane alokowane są dynamicznie, zależnie od rozmiaru problemu
- Program posiada możliwość wczytywania danych z pliku
- Program posiada możliwość wprowadzenia kryterium stopu
- Algorytmy zostały zaimplementowane zgodnie z paradygmatami programowania obiektowego
- Czas wykonania algorytmu mierzony był z dokładnością do nanosekund, przy wykorzystaniu bibliotek systemowych (chrono)

3. Wstęp teoretyczny

a) Algorytm Tabu Search

Tabu Search może być określona jako metoda dynamicznej zmiany sąsiedztwa danego rozwiązania, co oznacza że $N(x)$ nie jest stałe (jednorazowo określone) dla każdego x , lecz może zmieniać się w zależności od informacji na temat rozwiązywanego problemu, zgromadzonych na wcześniejszych etapach poszukiwania. Jest metaheurystyką do rozwiązywania problemów optymalizacyjnych, opartą na iteracyjnym przeszukiwaniu przestrzeni rozwiązań, wykorzystującą (zmienne) sąsiedztwo pewnych rozwiązań, zapamiętującą niektóre ruchy (transformacje rozwiązań) i częstość ich występowania, w celu unikania minimów lokalnych i poszukiwania rozwiązań globalnie optymalnych w rozsądnym czasie.

Schemat działania algorytmu Tabu Search przedstawiony został na wykładzie za pomocą poniższej procedury:

```
wybierz lub wylosuj punkt startowy  $x_0 \in X$ 
 $x_{opt} \leftarrow x_0$ 
 $tabu\_list \leftarrow \emptyset$ 
repeat
     $x_0 \leftarrow \text{AspirationPlus}(x_0)$ 
    if  $f(x_0) > f(x_{opt})$  then
         $x_{opt} \leftarrow x_0$ 
    zweryfikuj  $tabu\_list$ 
    Dla każdego element  $e$   $tabu\_list$  do
         $kadencja_i \leftarrow 0$ 
        if  $kadencja_i = 0$  then
            usuń element( $atrybut_i, kadencja_i$ ) z  $tabu\_list$ 
    if CriticalEvent() then
         $x_0 \leftarrow \text{Restart}()$  //Dywersyfikacja
    if  $f(x_0) < f(x_{opt})$  then
         $x_{opt} \leftarrow x_0$ 
until warunek zakończenia
```

Od początku jest pobierany rozmiar listy Tabu, graf do rozwiązania, oraz maksymalny czas na rozwiązanie(jeżeli czas nie jest podany, to za maksymalny czas będzie przyjęta liczba równa 10 min). Później jest wybierana początkowa ścieżka $(0,1,2,...,n-1,n,0)$.

Dalej losowane 2 różne wierzchołki które będą zamienione miejscami. Jeśli te wierzchołki nie są w liście tabu, oraz nowa ścieżka jest lepsza od poprzedniej, to ona przejmie się jak najlepsz. W innym przypadku losujemy dalej. Jeśli 100 razy pod rząd nie znaleźliśmy lepszą ścieżkę, to oczyszczymy listę Tabu.

b) Algorytm Symulowanego Wyżarzania

Algorytm Symulowanego Wyżarzania jest szczególnym przypadkiem algorytmu genetycznego. Nawiązuje do zjawiska fizycznego - zastygania cieczy tworzącej uporządkowaną, krystaliczną strukturę. W wysokiej temperaturze cząsteczki cieczy poruszają się swobodnie, ale wraz ze spadkiem temperatury możliwości ruchowe oraz prędkość poruszania się cząsteczek spada. Operacja powolnego schładzania ma na celu doprowadzenie metalu do równowagi termodynamicznej w stosunku do stanu wyjściowego oraz osiągnięcie pożądanych cech. Cały proces jest sterowany przez parametr zwany temperaturą. Schemat działania algorytmu symulowanego wyżarzania przedstawiony był na wykładzie za pomocą poniższej procedury:

```
t <- 0
inicjuj T
wybierz losowo rozwiązanie x
ocień x - wylicz g(x)
  repeat
    repeat
      wybierz nowe rozwiązanie y w otoczeniu x
      jeżeli g(y) < g(x) wtedy x <- y
      w przeciwnym wypadku, jeżeli  $p[0,1) < e^{\frac{g(y)-g(x)}{T}}$ , wtedy x <- y
    until warunek zakończenia
  T <- f(T,t)
  t <- t+1
until (kryterium zakończenia)
```

Dla każdej iteracji wyznaczana jest nowa temperatura, która jest wyliczana wzorem $T = T * s$, gdzie s to sztywno zadana przez nas stała. Jest to sposób geometryczny wyznaczania temperatury. Ważnym pytaniem, które sobie należy postawić w związku z rozważanym problemem jest jak zdefiniować sąsiedztwo. W tym przypadku będzie to rozważany na wykładzie sposób zamiany pozycji dwóch miast w permutacji. Zadanych stałych do wyliczania temperatury jest dziesięć, każda używana przy oddzielnej iteracji dla danej instancji. Użytkownik może sobie ustalać współczynnik temperatury. Temperatura początkowa jest wyliczona za pomocą wzoru: długość ścieżki*10.

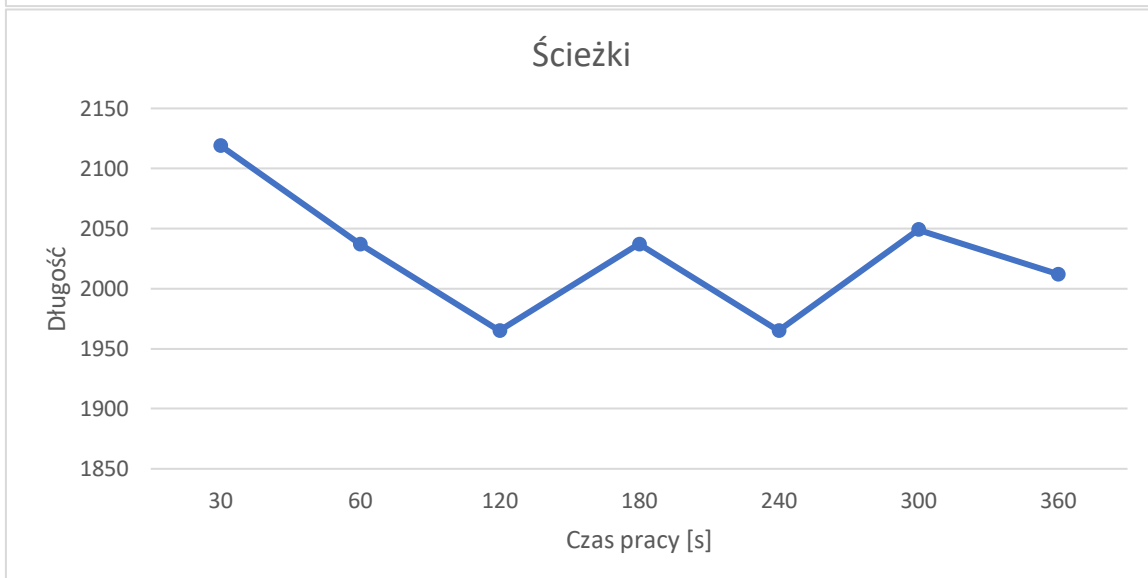
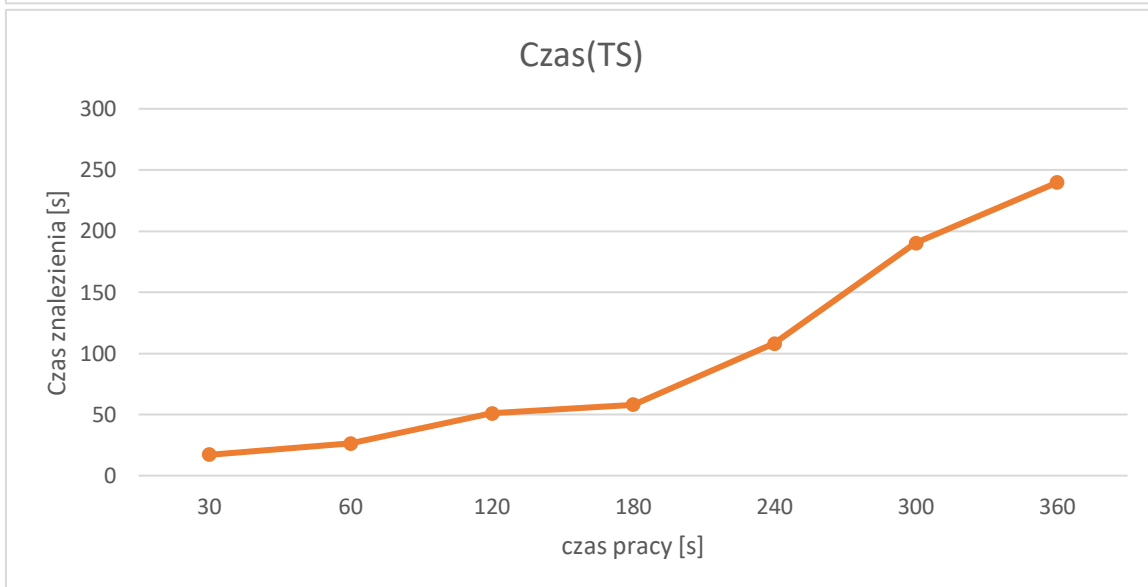
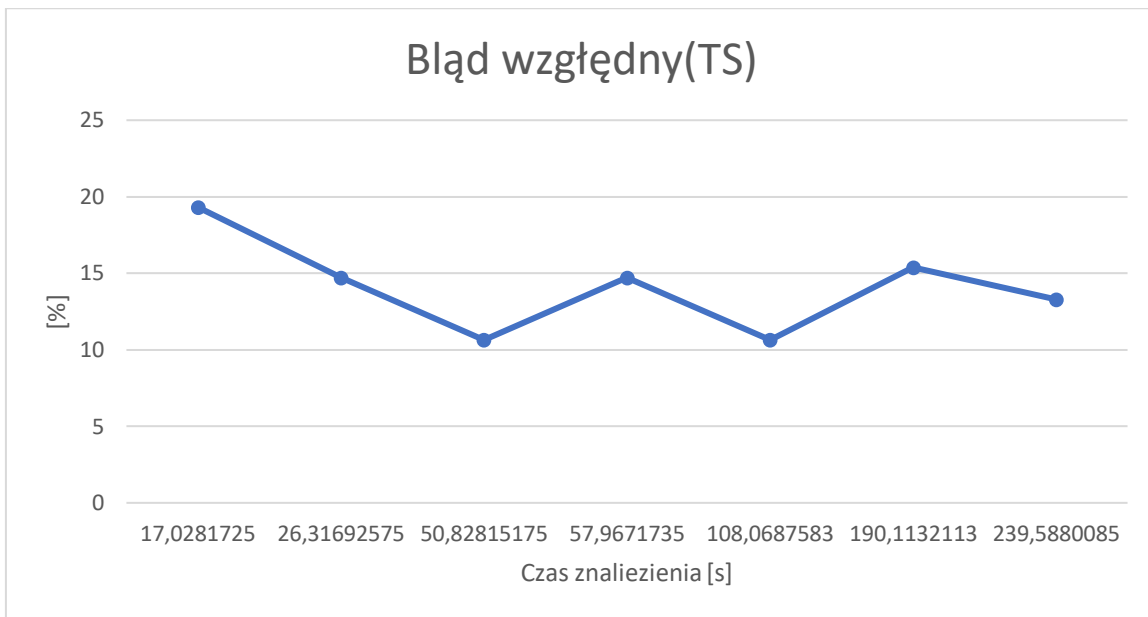
4. Testy i wykresy

Po każdej zmianie współczynnika byli przeprowadzone testy 10 raz.

a) 47 miast („ftv47.atsp”)

Algorytm Tabu Search

max czas [s]	Sr czas [s]	najkr znaleziona	bład %
30	17,02817	2119	19,31306
60	26,31693	2037	14,69595
120	50,82815	1965	10,64189
180	57,96717	2037	14,69595
240	108,0688	1965	10,64189
300	190,1132	2049	15,37162
360	239,588	2012	13,28829

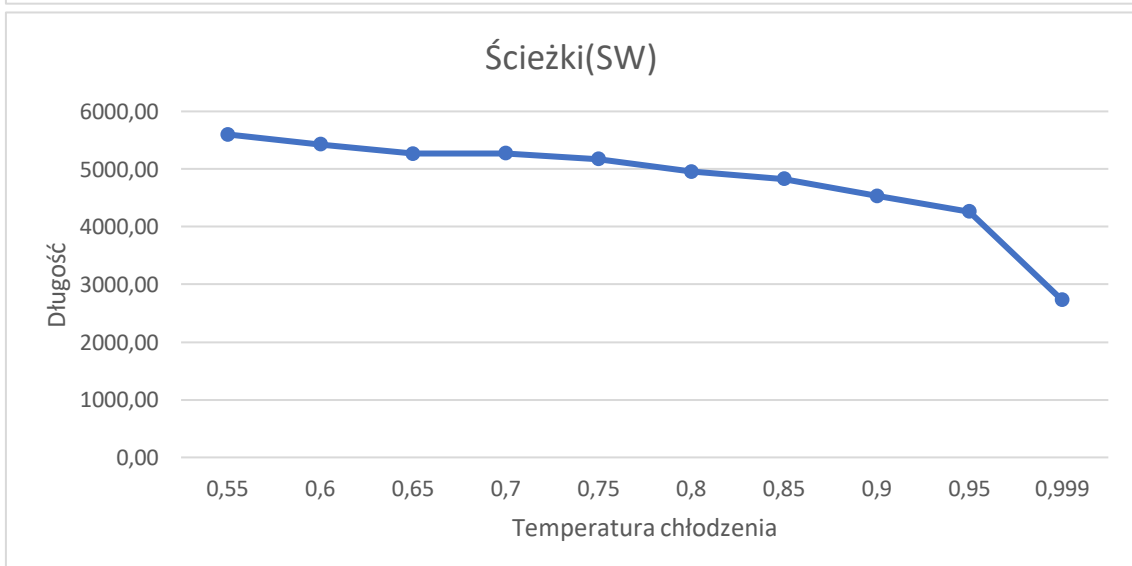
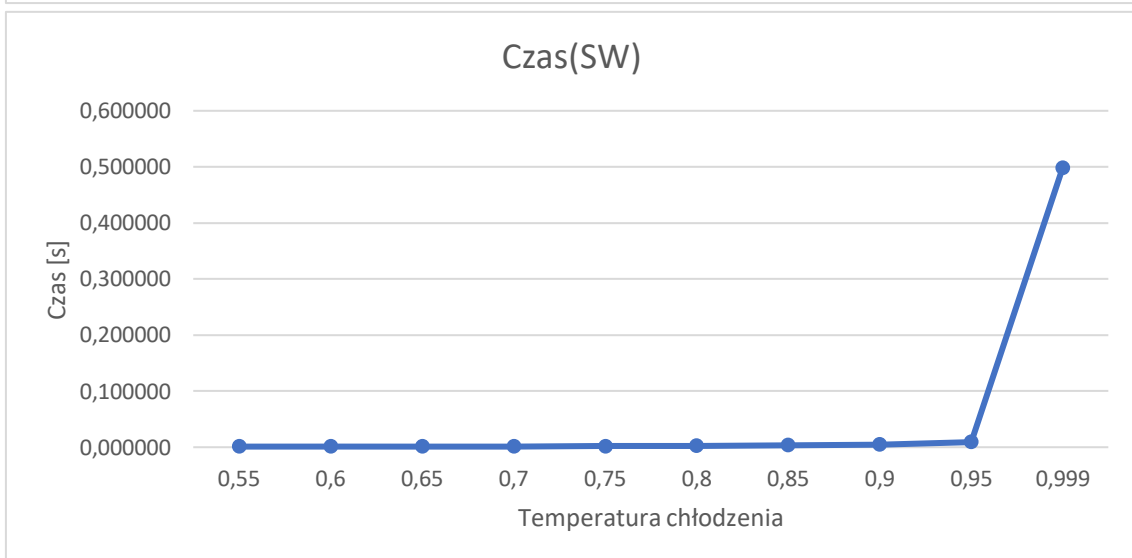
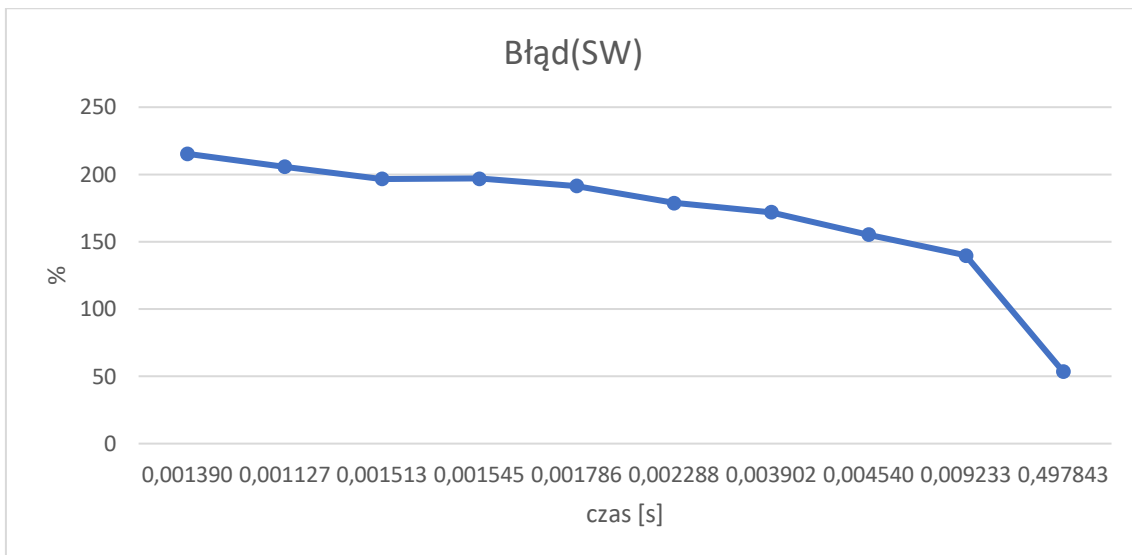


Algorytm Symulowanego Wyżarzania

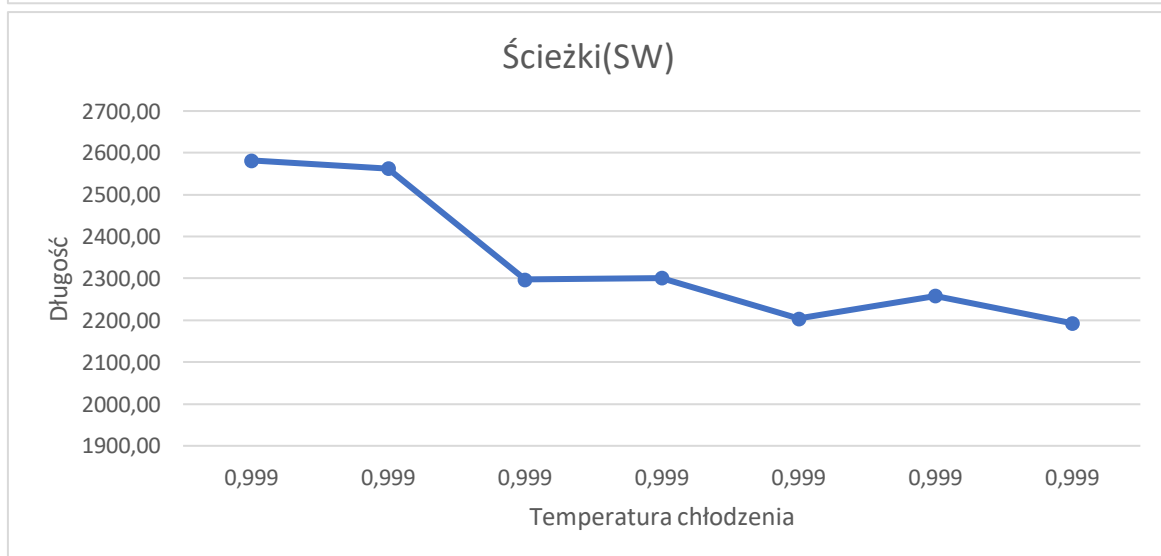
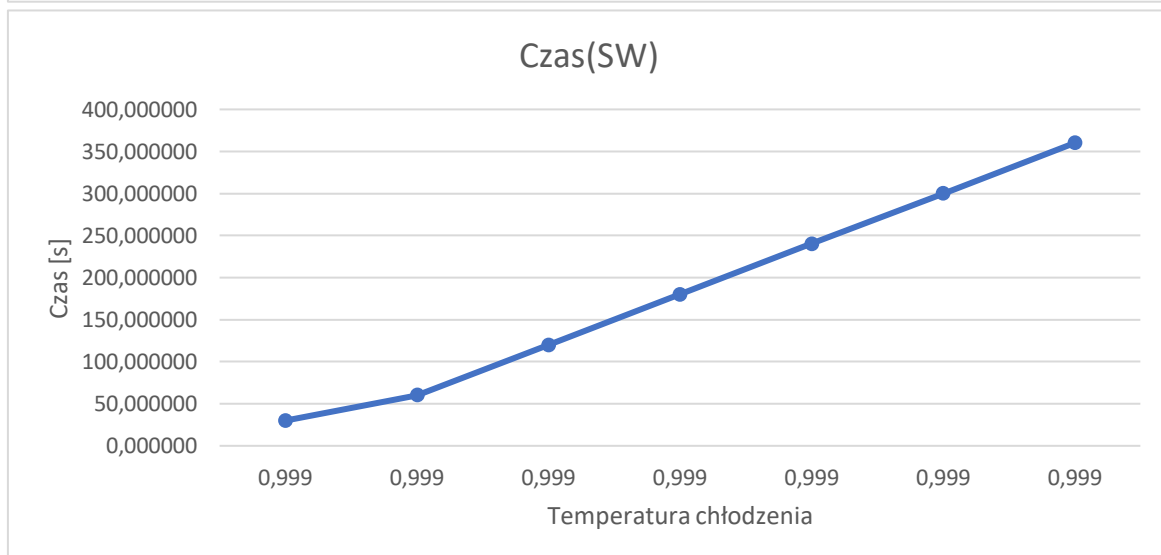
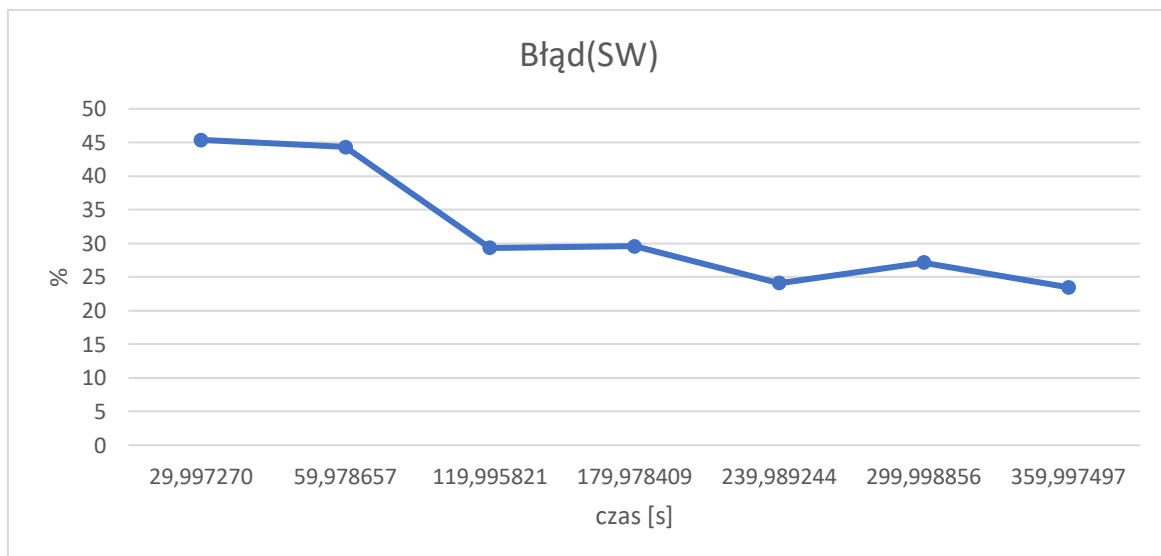
Temperatura początkowa	Temperatura chłodzenia	Temperatura minimalna	Długość najkrótszej trasy	Czas pracy [s]	blad %
42890	0,55	1.92936e-05	5601,00	0,001390	215,37162
42890	0,6	2.06399e-05	5430,00	0,001127	205,74324
42890	0,65	2.91817e-05	5269,00	0,001513	196,67793
42890	0,7	2.17891e-05	5273,00	0,001545	196,90315
42890	0,75	2.43719e-05	5175,00	0,001786	191,38514
42890	0,8	2.66627e-05	4955,00	0,002288	178,99775
42890	0,85	2.86297e-05	4831	0,003902	172,01577
42890	0,9	3.02592e-05	4534	0,004540	155,29279
42890	0,95	1.92936e-05	4262	0,009233	139,97748
42890	0,999	1.92936e-05	2731	0,497843	53,772523

Badanie najlepszego przypadku:

Czas pracy []	Temperatura początkowa	Temperatura chłodzenia	Temperatura minimalna	Długość najkrótszej trasy	Czas znalezienia [s]	blad %
30	42890	0,999	2.46539e-321	2582,00	29,997270	45,382883
60	42890	0,999	2.46539e-321	2563,00	59,978657	44,313063
120	42890	0,999	2.46539e-321	2297,00	119,995821	29,335586
180	42890	0,999	2.46539e-322	2301,00	179,978409	29,560811
240	42890	0,999	2.46539e-323	2204,00	239,989244	24,099099
300	42890	0,999	2.46539e-324	2258,00	299,998856	27,13964
360	42890	0,999	2.46539e-325	2193	359,997497	23,47973



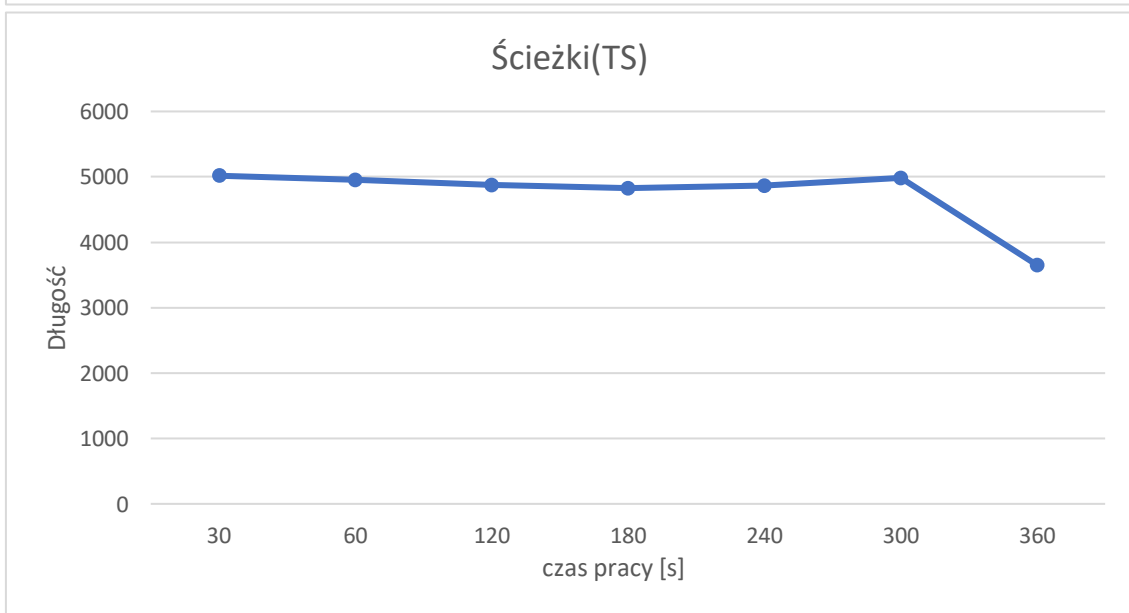
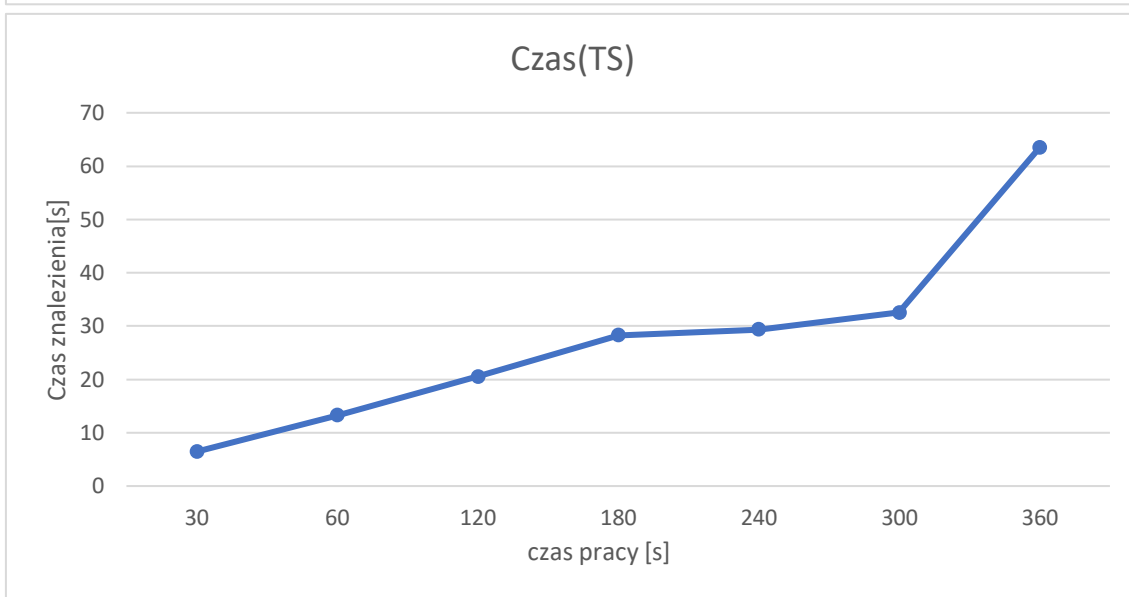
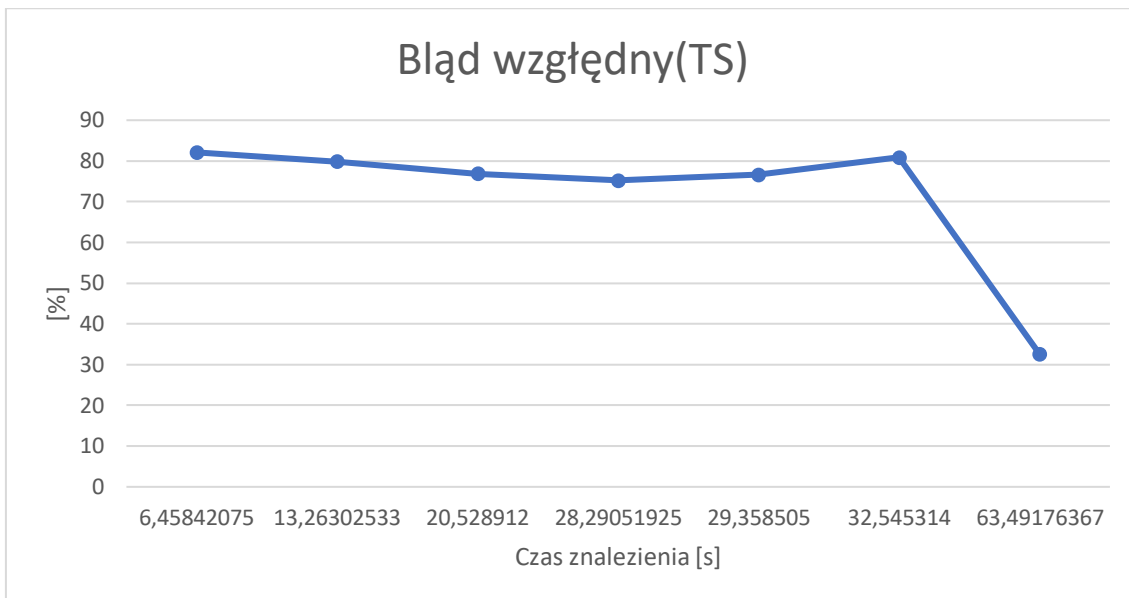
Wykresy najlepszego przypadku:



b) 170 miast („ftv170.atsp“)

Algorytm Tabu Search

max czas [s]	Sr czas [s]	droga		blad %
30	6,4584208	5017		82,10526
60	13,263025	4954		79,81851
120	20,528912	4873		76,8784
180	28,290519	4826		75,17241
240	29,358505	4865		76,58802
300	32,545314	4982		80,83485
360	63,491764	3652		32,55898

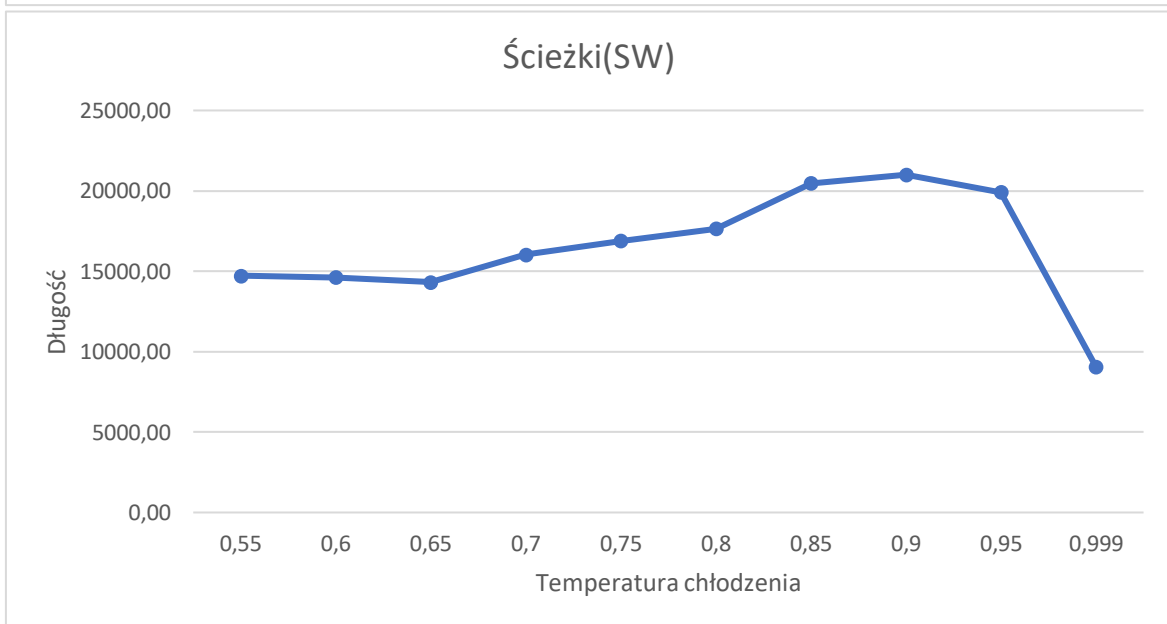
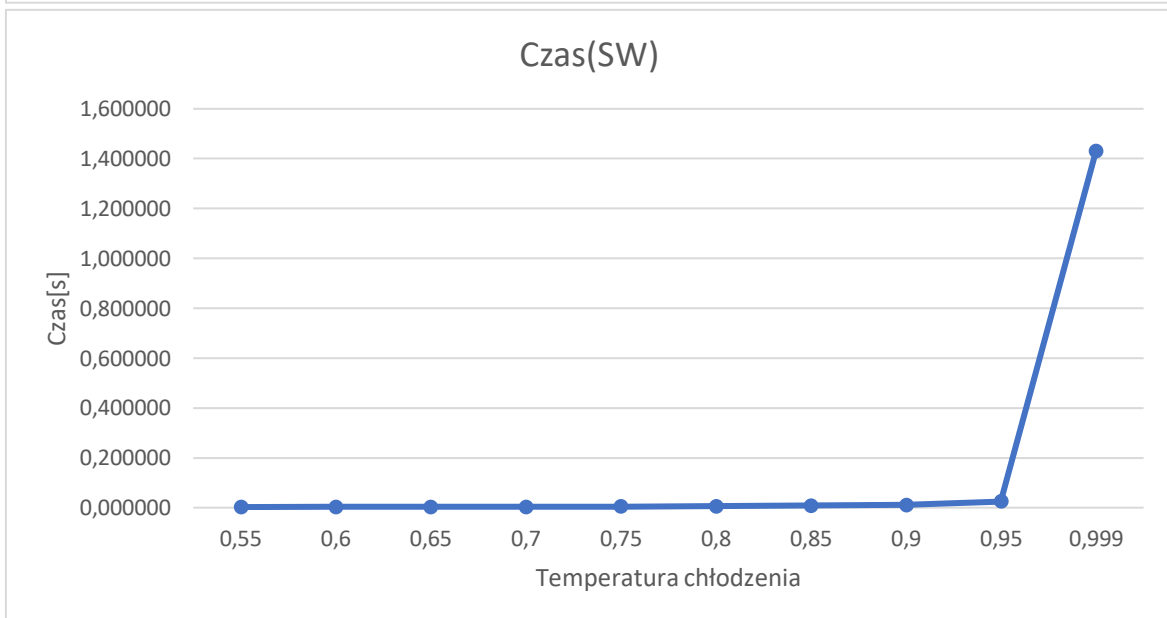
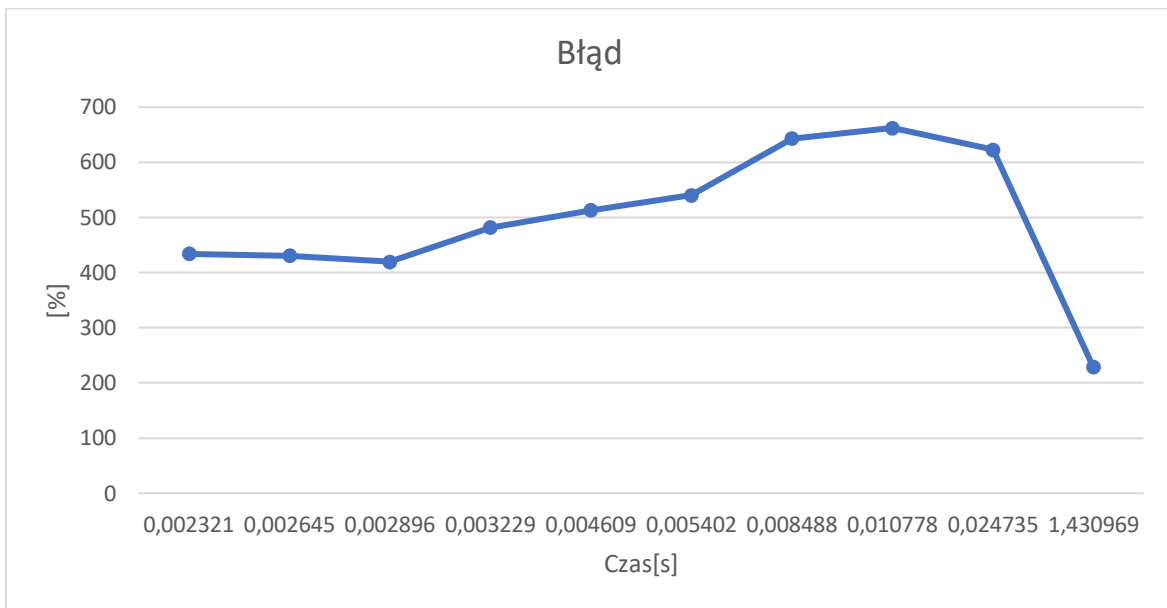


Algorytm Symulowanego Wyżarzania

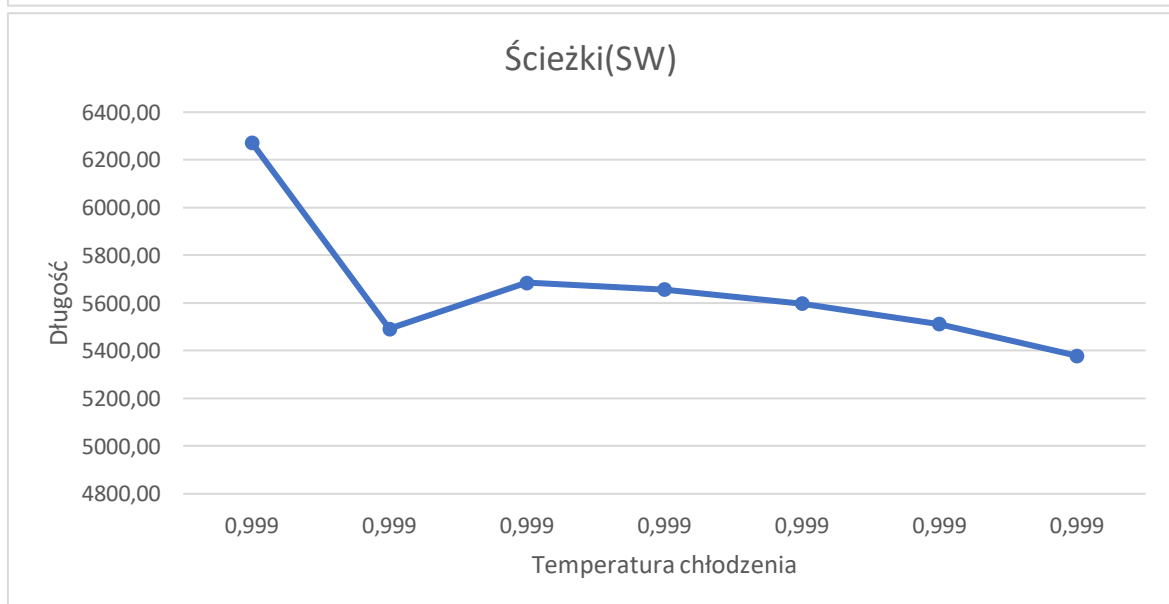
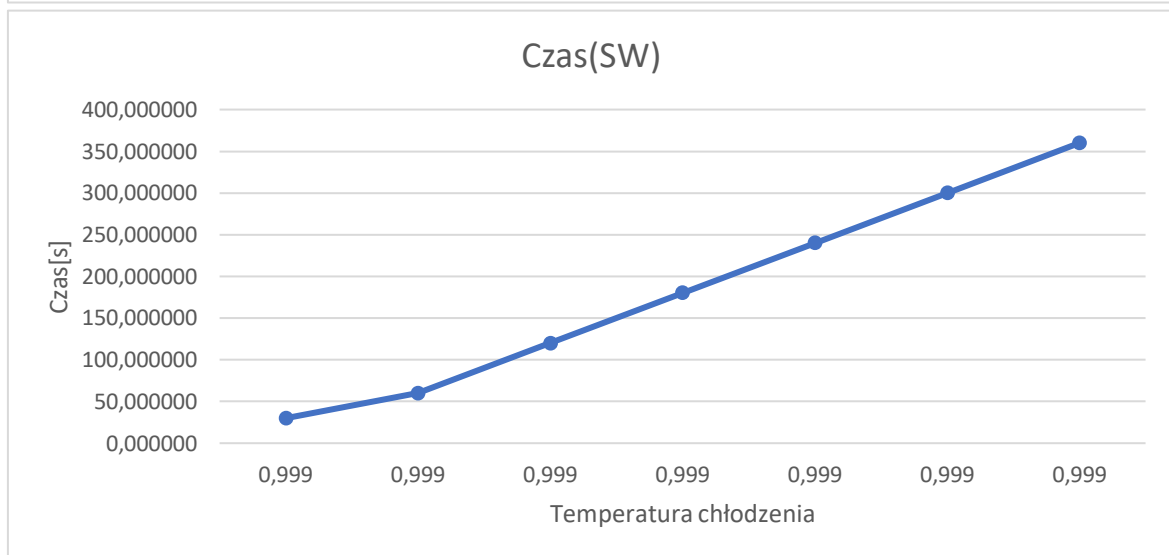
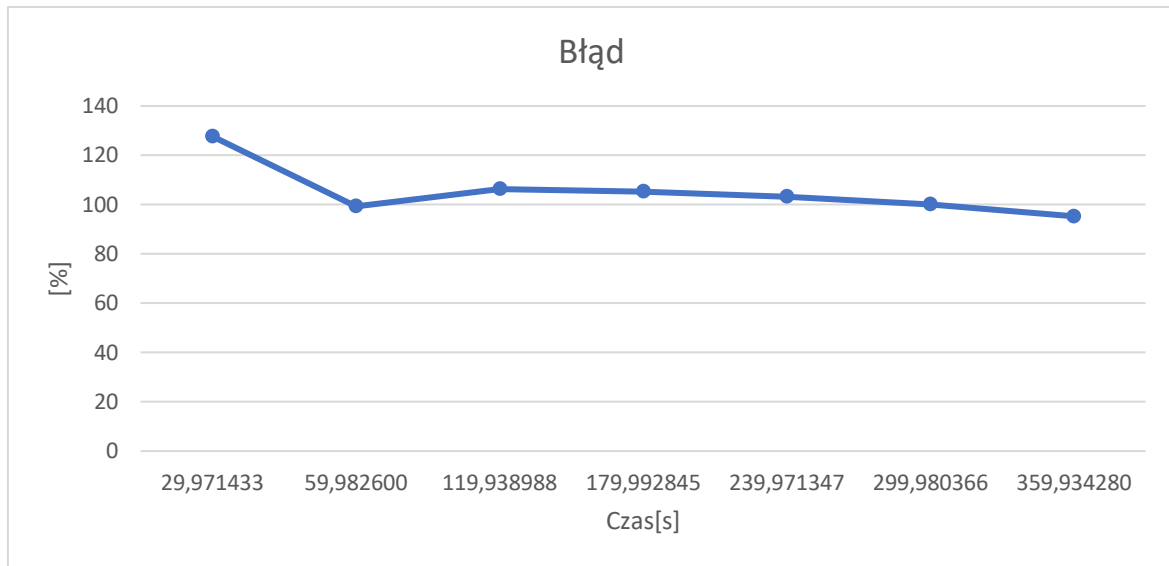
Temperatura początkowa	Temperatura chłodzenia	Temperatura minimalna	Długość najkrótszej trasy	Czas pracy [s]	Błąd %
71460	0,55	1.768e-05	14715,00	0,002321	434,1198
71460	0,6	2.06332e-05	14623,00	0,002645	430,7804
71460	0,65	2.05421e-05	14324,00	0,002896	419,9274
71460	0,7	2.54123e-05	16027,00	0,003229	481,7423
71460	0,75	3.04549e-05	16895,00	0,004609	513,2486
71460	0,8	2.8431e-05	17637,00	0,005402	540,1815
71460	0,85	2.92941e-05	20478	0,008488	643,3031
71460	0,9	2.97699e-05	21004	0,010778	662,3956
71460	0,95	2.9902e-05	19910	0,024735	622,686
71460	0,999	3.04966e-05	9056	1,430969	228,7114

Badanie najlepszego przypadku:

Czas pracy [s]	Temperatura początkowa	Temperatura chłodzenia	Temperatura minimalna	Długość najkrótszej trasy	Czas pracy[s]	Błąd %
30	71460	0,999	1.18576e-322	6272,00	29,971433	127,6588
60	71460	0,999	1.18576e-322	5492,00	59,982600	99,34664
120	71460	0,999	1.18576e-322	5685,00	119,938988	106,3521
180	71460	0,999	1.18576e-322	5657,00	179,992845	105,3358
240	71460	0,999	1.18576e-322	5598,00	239,971347	103,1942
300	71460	0,999	1.18576e-322	5512,00	299,980366	100,0726
360	71460	0,999	1.18576e-322	5379	359,934280	95,24501



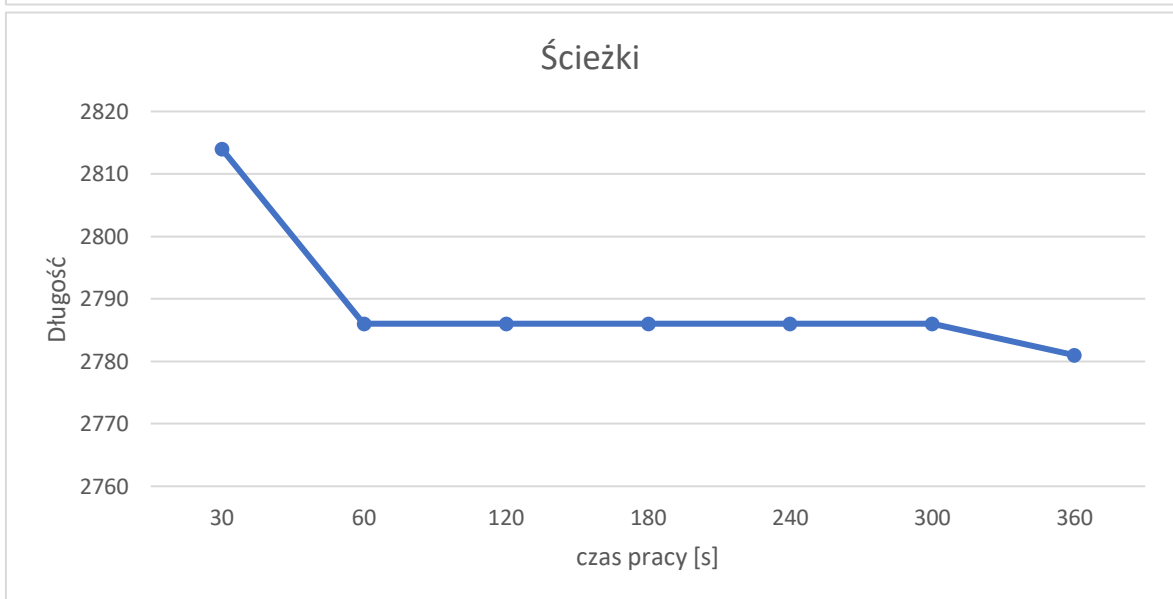
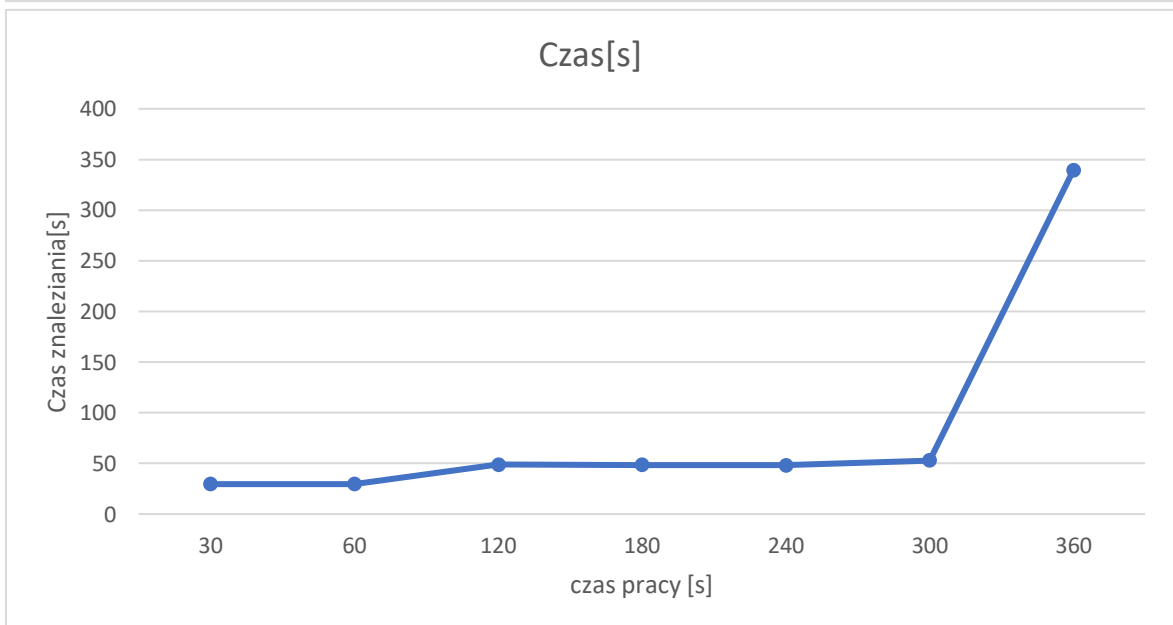
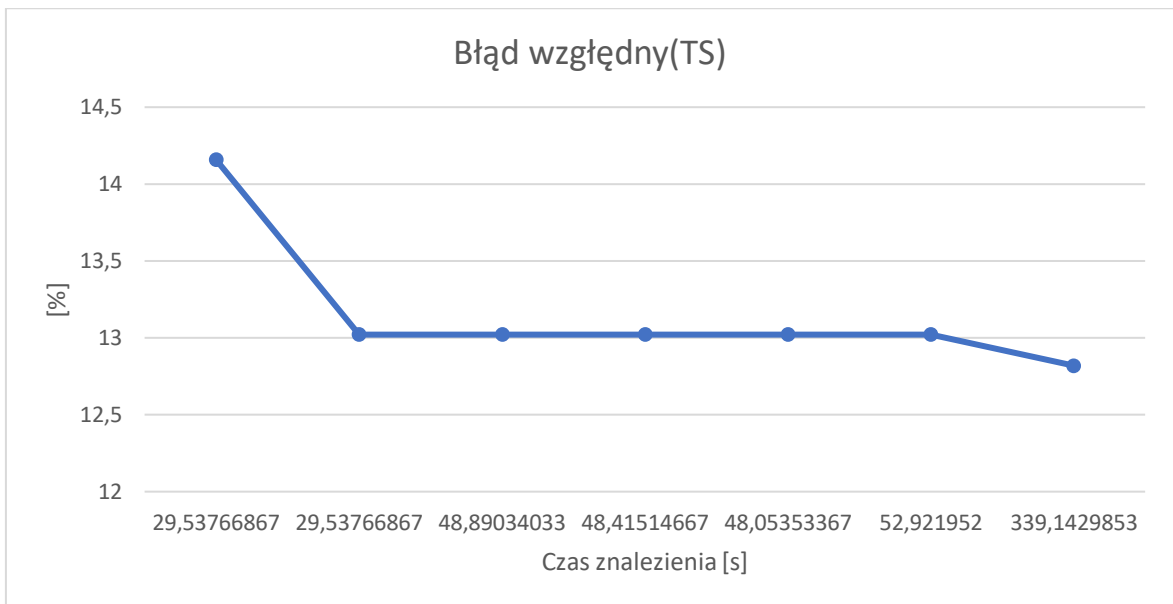
Wykresy najlepszego przypadku:



c) 403 miasta („rbg403.atsp“)

Algorytm Tabu Search

max czas [s]	Sr czas	droga		blad
30	29,5376687	2814		14,15822
60	29,5376687	2786		13,02231
120	48,8903403	2786		13,02231
180	48,4151467	2786		13,02231
240	48,0535337	2786		13,02231
300	52,921952	2786		13,02231
360	339,142985	2781		12,81947

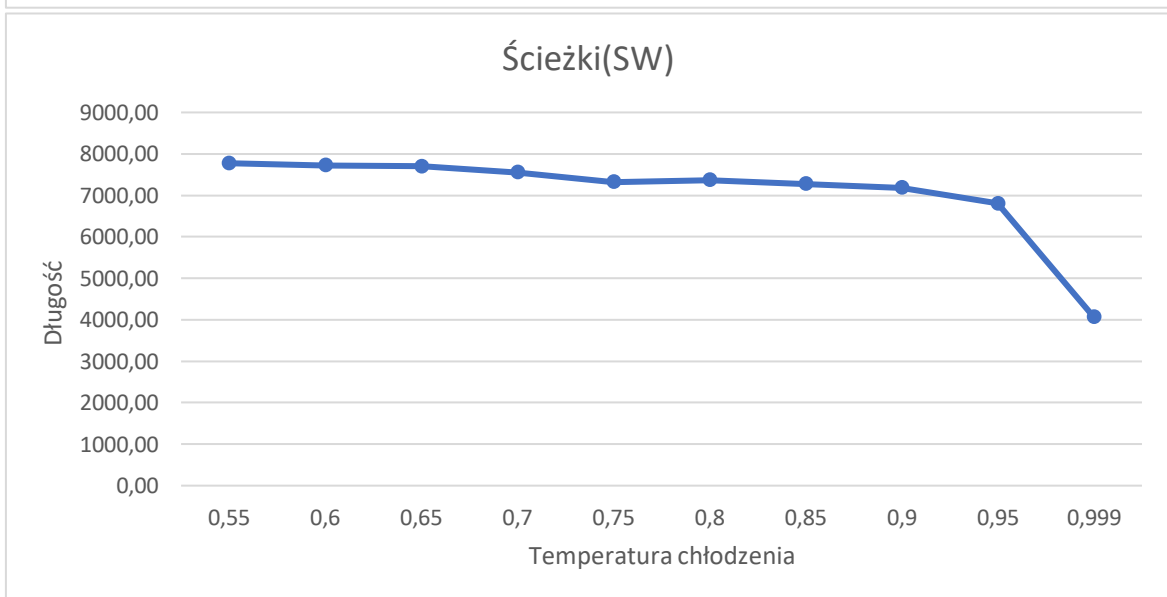
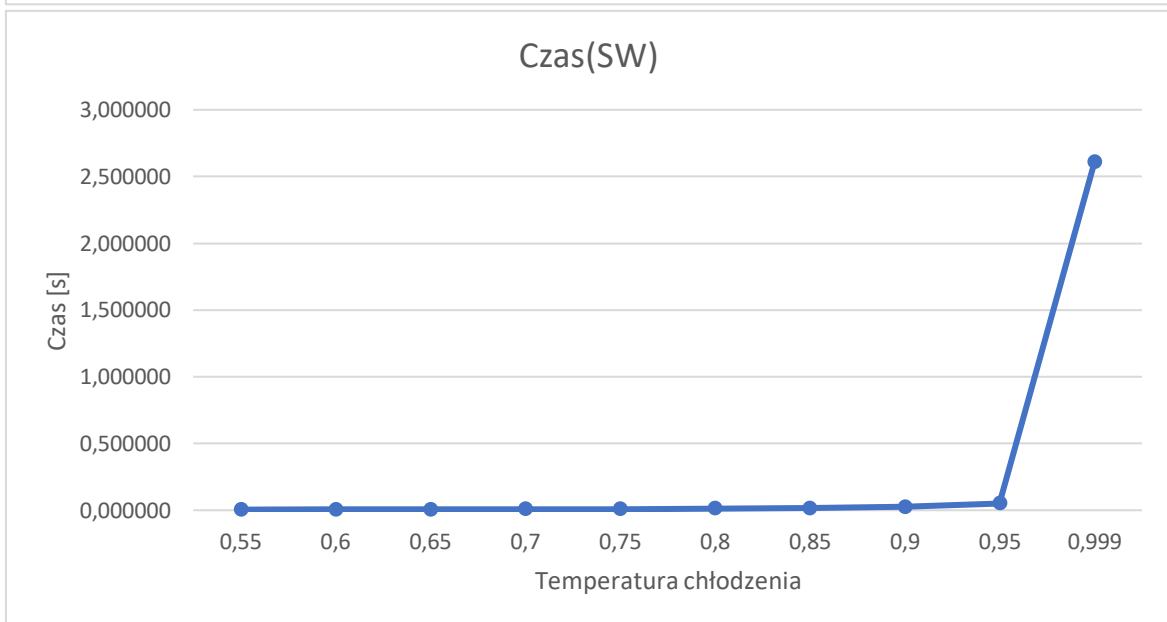
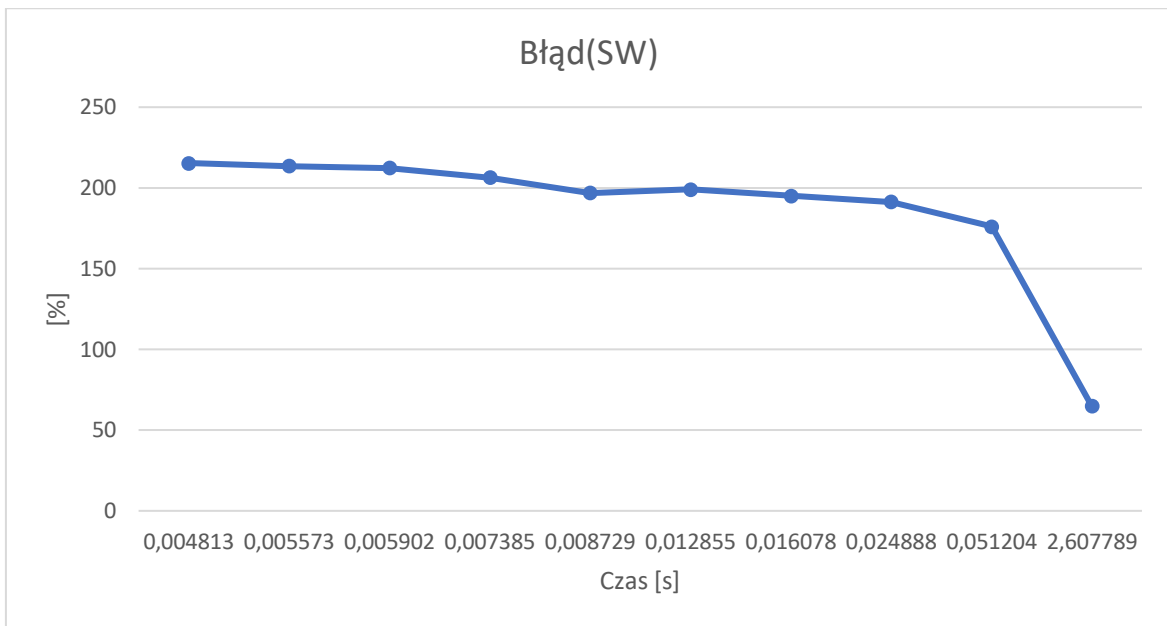


Algorytm Symulowanego Wyżarzania

Temperatura początkowa	Temperatura chłodzenia	Temperatura minimalna	Długość najkrótszej trasy	Czas pracy[s]	Bład %
79560	0,55	1.9684e-05	7775,00	0,004813	215,4158
79560	0,6	2.2972e-05	7727,00	0,005573	213,4686
79560	0,65	2.28705e-05	7700,00	0,005902	212,3732
79560	0,7	2.82928e-05	7551,00	0,007385	206,3286
79560	0,75	2.54303e-05	7320,00	0,008729	196,9574
79560	0,8	2.53229e-05	7370,00	0,012855	198,9858
79560	0,85	2.77224e-05	7276	0,016078	195,1724
79560	0,9	2.98299e-05	7181	0,024888	191,3185
79560	0,95	3.00455e-05	6803	0,051204	175,9838
79560	0,999	3.05063e-05	4063	2,607789	64,82759

Badanie najlepszego przypadku:

Czas pracy []	Temperatura początkowa	Temperatura chłodzenia	Temperatura minimalna	Długość najkrótszej trasy	Czas pracy	blad
30	79560	0,999	1.18576e-322	2696,00	29,977153	9,371197
60	79560	0,999	1.18576e-323	2678,00	59,984848	8,640974
120	79560	0,999	1.18576e-324	2586,00	119,946301	4,908722
180	79560	0,999	1.18576e-325	2609,00	179,999420	5,841785
240	79560	0,999	1.18576e-326	2568,00	239,970524	4,178499
300	79560	0,999	1.18576e-327	2569,00	299,989576	4,219067
360	79560	0,999	1.18576e-328	2562	359,962360	3,935091



Wykresy najlepszego przypadku:



5. Wstęp teoretyczny

Powyrzszse algorytmy wyjaśniają problem komiwożazera w przybliżeniu. W zależności od parametrów można mieć mniej lub więcej dokładne odpowiedzi. Z powyższych wykresów można wywnioskować to, że parametrem, który zwiększał efektywność algorytmu Symulowanego Wyżarzania, była przyjęta temperatura chłodzenia (stała im bliższa jedynce, tym lepsze efekty) oraz im dłużej działa algorytm tym bardziej dokładną drogę otrzymujemy, ale nie zawsze się opłaca. Natomiast w przypadku algorytmu Tabu Search korzystnie wpływała manipulacja liczbą iteracji (im więcej, tym lepiej, ale nie zawsze ta zasada działała). Niestety parametry te wydłużały czas wykonywania algorytmu (np. w przypadku algorytmu SW, przy zmianie wartości stałej z 0,55 do wartości 0,999, zanotowano około 10-krotny wzrost czasu).

Otrzymywany błąd względny między średnim uzyskiwanym wynikiem oraz wynikiem optymalnym nie zależy w żaden sposób od wielkości instancji.

Napisanie tego projektu pomogło poznać nowe metody rozwiązania problemu komiwożazera oraz zwiększyć swoje umiejętności w programowaniu.