

Dominika Dolik, 235853
Prowadzący: dr inż. Dariusz Banasiak
Termin: środa, 17:25

Projektowanie Efektywnych Algorytmów

Zadanie projektowe nr 2

Rozwiązanie problemu komiwojażera przy użyciu metody
symulowanego wyżarzania

Wstęp teoretyczny

Algorytmy metaheurystyczne służą do rozwiązywania problemów optymalizacyjnych, bazują na analogiach do procesów ze świata rzeczywistego. Prowadzą do uzyskania wyników bliskich (lub równych) wynikom optymalnym.

Algorytm symulowanego wyżarzania nawiązuje do zjawiska fizycznego, polegającego na nagrzaniu materiału do określonej temperatury, wytrzymaniu przy tej temperaturze oraz ponownym studzeniu. Celem takiej obróbki w metalurgii jest przybliżenie stanu materiału do warunków równowagi termodynamicznej w stosunku do stanu wyjściowego oraz osiągnięcie pożądaných cech (np. twardości).

Różnicą między symulowanym wyżarzaniem a pierwotnymi metodami iteracyjnymi jest możliwość w wyborze przez niego gorszego rozwiązania. Wybór taki jest dokonywany z pewnym prawdopodobieństwem. Dzięki temu algorytm symulowanego wyżarzania może w określonych warunkach wyjść ze znalezionej minimum lokalnego i dalej podążać w kierunku rozwiązania optymalnego.

Parametrem, który ma wpływ na to prawdopodobieństwo, jest temperatura. Im wyższa, tym prawdopodobieństwo wyboru gorszego rozwiązania jest większe. Wraz z kolejnymi iteracjami temperatura spada i wybierane są częściej rozwiązania lepsze. Pod koniec pracy algorytmu temperatura jest na tyle niska, że prawdopodobieństwo wyboru gorszego rozwiązania jest bliskie zeru.

Opis algorytmu dla problemu TSP w krokach

1. Wybór rozwiązania początkowego R i obliczenie wartości ścieżki początkowej W
2. Dopóki $T_A > T_K$:
 - 2.1. Dopóki $i < \text{limit_iteracji}$:
 - 2.1.1. Stworzenie nowego rozwiązania R_T przez zamianę dwóch losowych elementów w R
 - 2.1.2. Obliczenie wartości ścieżki dla nowego rozwiązania W_T
 - 2.1.3. Jeśli $W_T < W$ lub prawdopodobieństwo $> \text{random}()$ to $W = W_T$, $R = R_T$
 - 2.2. Obniżenie temperatury

Implementacja algorytmu

Implementacja algorytmu powstała w środowisku Microsoft Visual Studio w języku C++.

Do opisu algorytmu została utworzona klasa `SimulatedAnnealing`, która w konstruktorze przyjmuje rozmiar wczytanej instancji jako parametr. Wczytywanie grafu odbywa się za pomocą klasy `Reader`. Na tej podstawie tworzone są dwie tablice zawierające kolejne wierzchołki grafu (permutowana ścieżka i przechowująca najlepsze rozwiązanie), co jest też rozważane jako początkowa ścieżka.

Metoda `execute()` jest metodą główną klasy i to ona odpowiada za przebieg algorytmu. Wywołuje w czasie działania inne zaimplementowane metody, odpowiadające za wyliczanie aktualnych wartości parametrów. Są to: `acceptanceProb()` – obliczająca prawdopodobieństwo akceptacji gorszego rozwiązania, `sumCosts()` – obliczająca wartość ścieżki. `RandomSwap()`

odpowiada za zamianę kolejności dwóch losowych wierzchołków. Pozostałe metody `copyArray()` i `printResult()` odpowiadają kolejno za kopiowanie wartości z jednej tablicy do drugiej oraz wydrukowanie wyniku na ekran.

Wartości parametrów charakterystycznych dla algorytmu (temperatura, współczynnik studzenia α) są podane bezpośrednio w kodzie. Największym problemem przy implementacji tej metody, było odpowiednie ich dobranie.

Dane testowe

Do testów użyto gotowych plików tekstowych, zawierających liczbę wierzchołków grafu i macierz z wartościami odległości między wierzchołkami. Dane pochodzą ze strony uniwersytetu w Heidelbergu. Pomiary zostały wykonane na siedmiu rozmiarów instancji ($n = 6, 17, 33, 47, 64, 70, 170$). Każda instancja ma wyznaczone optymalne rozwiązanie, co umożliwiło ocenę jakości rozwiązań proponowanych przez program. Parametry (temperatura początkowa, temperatura końcowa, współczynnik studzenia α) zostały wyznaczone metodą „prób i błędów”. Do testów zostały wykorzystane następujące wartości:

temperatura początkowa	temperatura końcowa	współczynnik studzenia α
10000.0	0.0001	$\alpha \in \{0.99, 0.999, 0.9999\}$

Do weryfikacji odchyleń rozwiązań zwróconych przez program od rzeczywistej wartości wyliczono procentowy błąd korzystając ze wzoru:

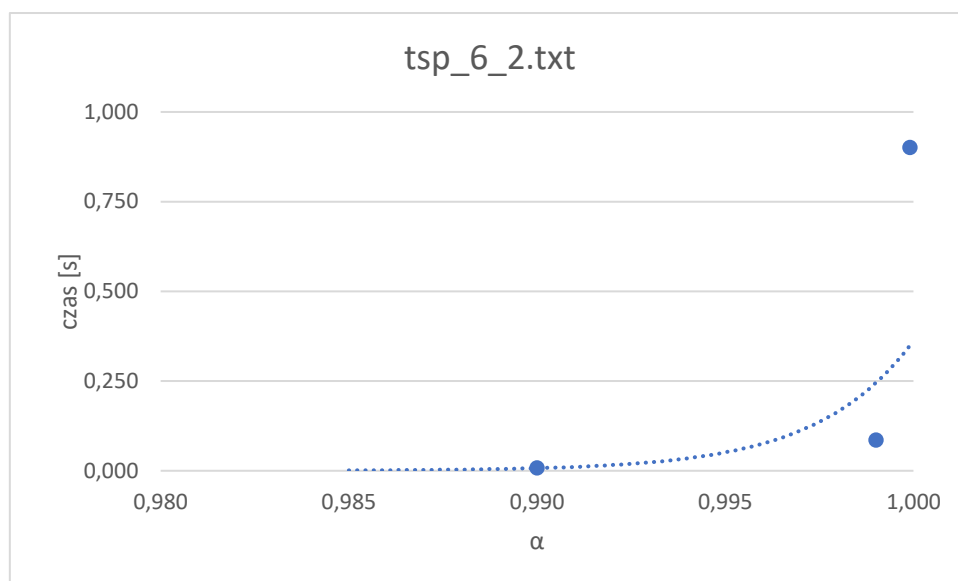
$$B = \frac{(DrogaWyliczona - DrogaOptymalna)}{DrogaOptymalna} * 100$$

Dla każdej wielkości parametru alfa testy zostały przeprowadzone 10 razy, a wyniki uśrednione.

Wyniki pomiarów

$n = 6$ (tsp_6_2.txt), rozwiązanie optymalne: 80

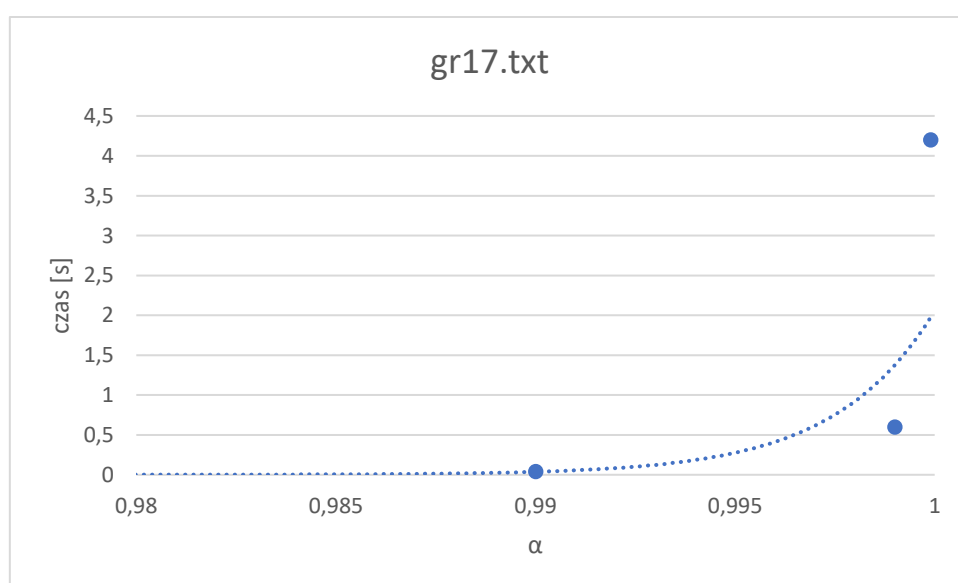
α	0,99	0,999	0,9999
	wynik		
	80	80	80
	czas [s]		
	0,008	0,086	0,901
	błąd [%]		
	0	0	0

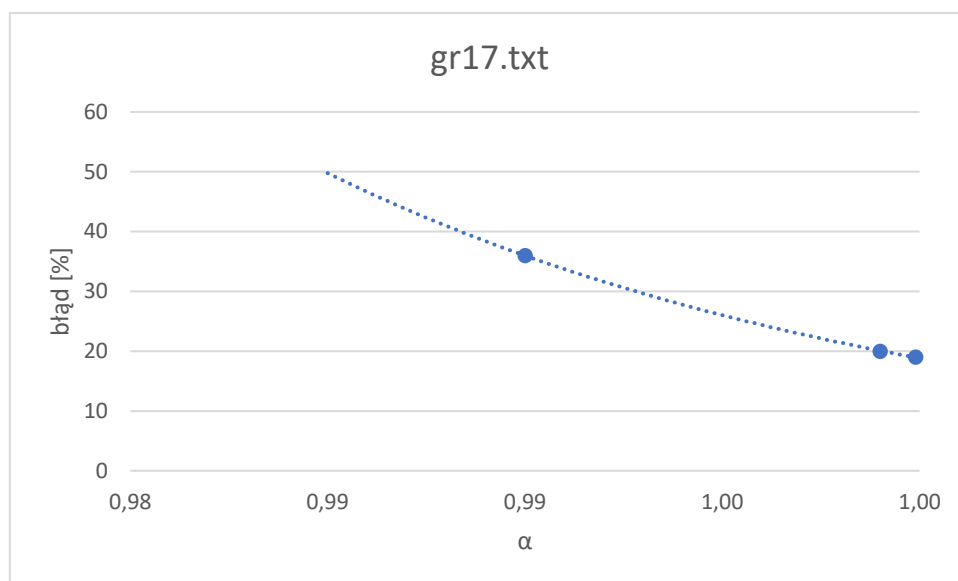


Wykres zależności błędów od wielkości parametru α został pominięty.

$n = 17$ (gr17.txt), rozwiązanie optymalne: 2085

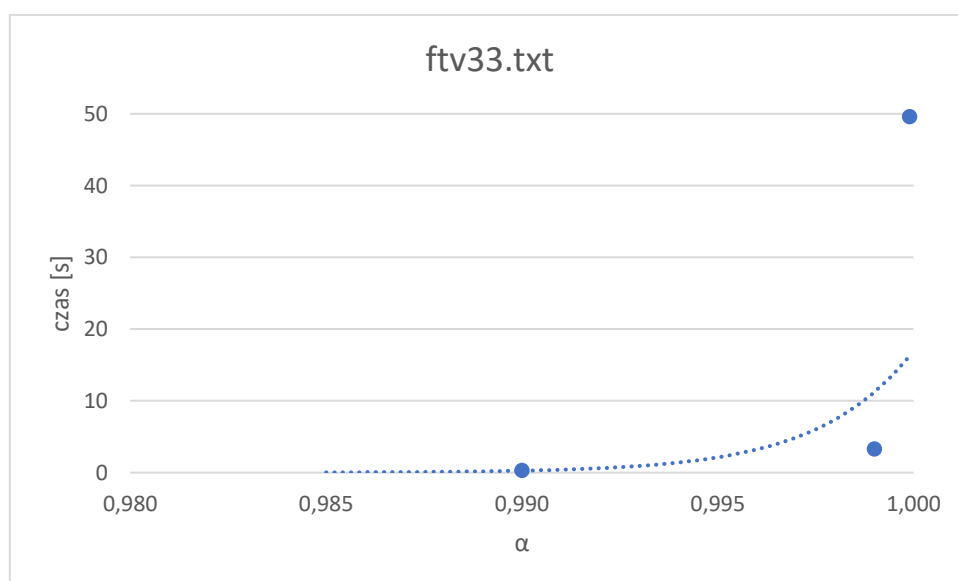
α	0.99	0.999	0.9999
	wynik		
	2853	2519	2495
	czas [s]		
	0,04	0,6	4,2
	błąd [%]		
	36	20	19

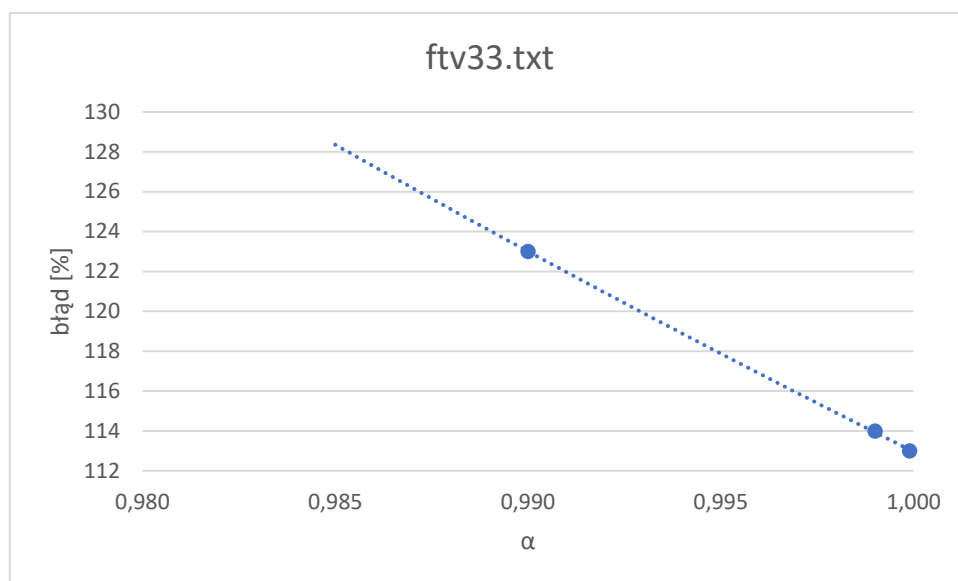




n = 33 (ftv33.txt), rozwiązanie optymalne: 1286

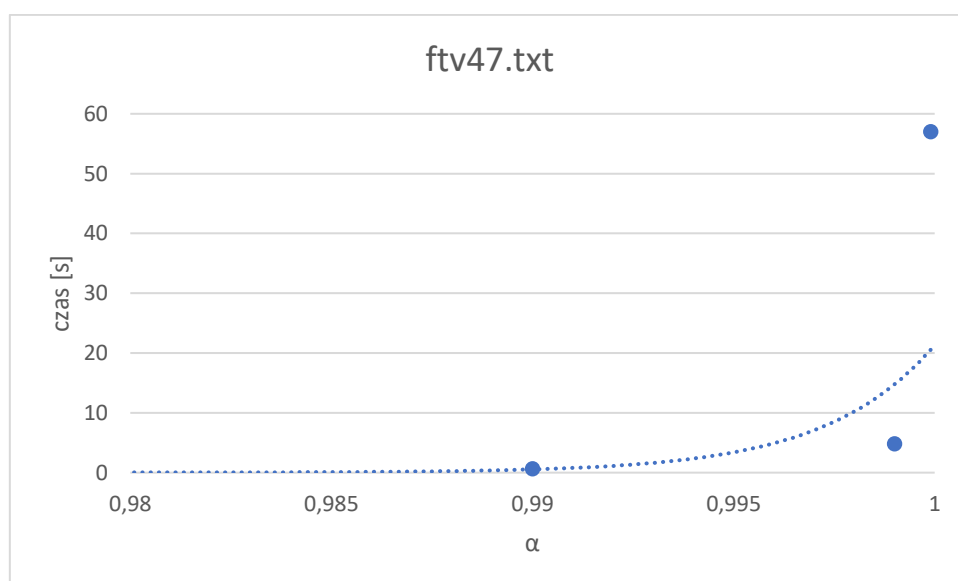
α	0,99	0,999	0,9999
	wynik		
	2878	2754	2745
	czas [s]		
	0,3	3,3	49,6
	błąd [%]		
	123	114	113

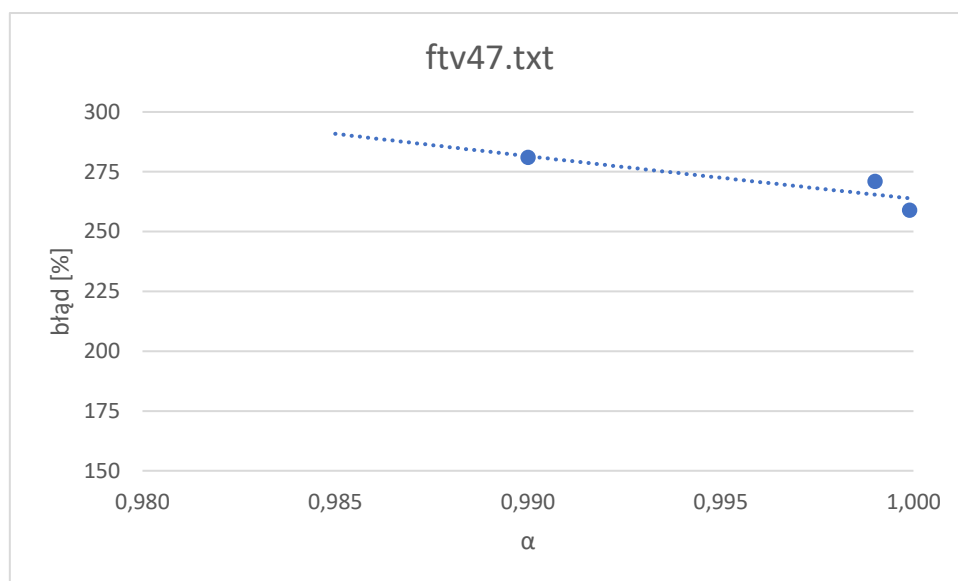




n = 47 (ftv47.txt), rozwiązanie optymalne: 1776

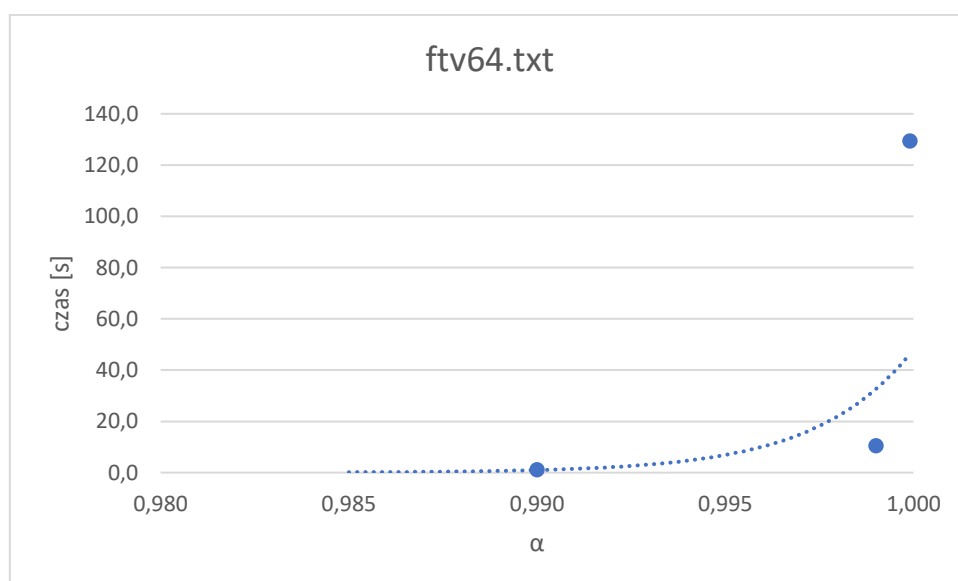
α	0.99	0.999	0.9999
	wynik		
	4749	4723	4620
	czas [s]		
	0,6	4,8	57
	błąd [%]		
	167	165	160

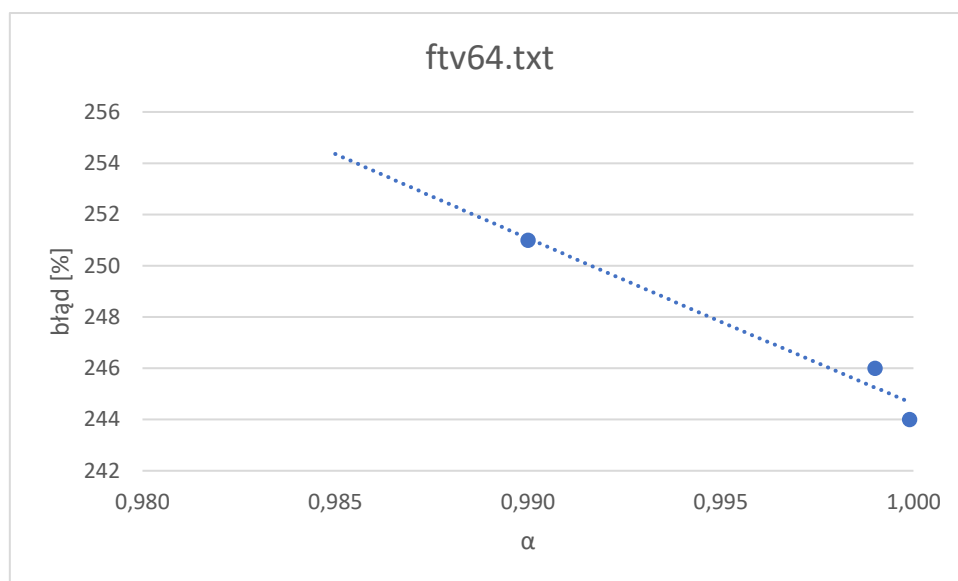




n = 64 (ftv64.txt), rozwiązanie optymalne: 1839

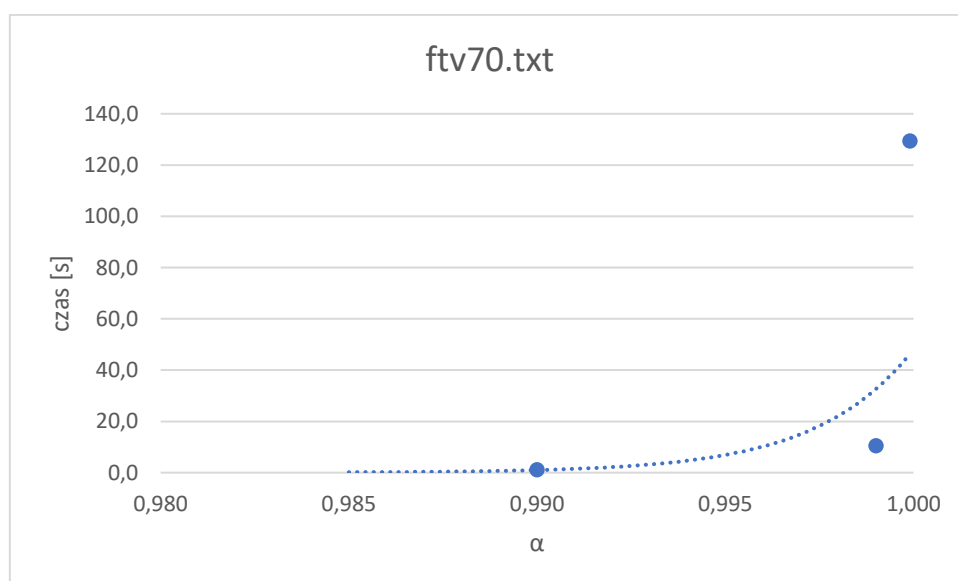
α	0.99	0.999	0.9999
	wynik		
	6460	6374	6334
	czas [s]		
	1,1	10,5	129,3
	błąd [%]		
	251	246	244

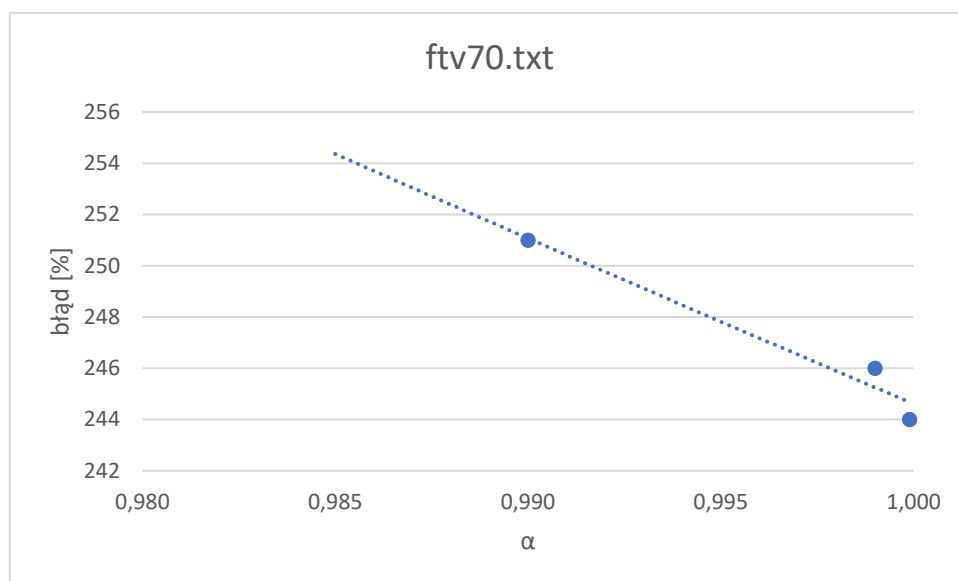




n = 70 (ftv70.txt), rozwiązanie optymalne: 1950

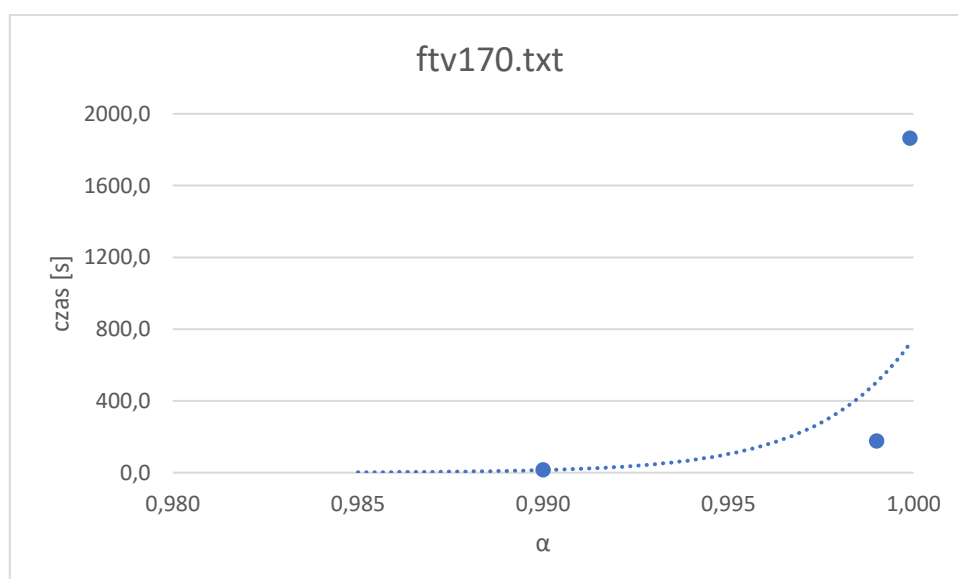
α	0.99	0.999	0.9999
	wynik		
	7433	7246	7015
	czas [s]		
	0,9	13,8	133,9
	błąd [%]		
	281	271	259

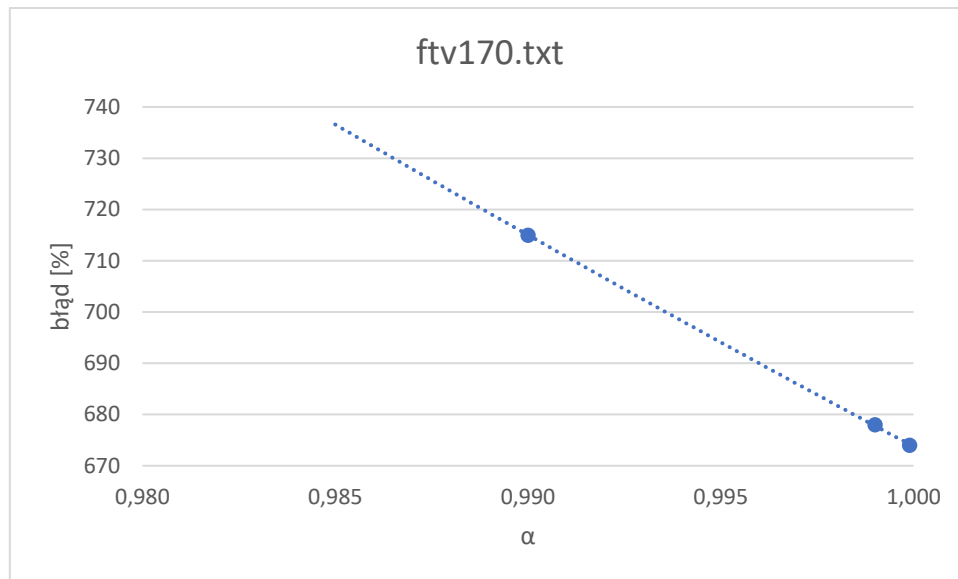




n = 170 (ftv170.txt), rozwiązanie optymalne: 2755

α	0.99	0.999	0.9999
	wynik		
	22453	21452	21326
	czas [s]		
	15,8	176,7	1865
	błąd [%]		
	715	678	674





Wnioski

- Na podstawie wykresów można zaobserwować, że czas obliczeń zwiększa się wykładniczo w funkcji temperatury chłodzenia. Algorytm wykonuje się dłużej, im wyższe są obie wartości. Wraz ze wzrostem czasu działania algorytmu, odpowiedzi zbliżają się coraz bardziej do wyników optymalnych. Im większą wartość ma współczynnik chłodzenia, tym uzyskiwany wynik jest bliższy optymalnemu.
- Algorytm symulowanego wyżarzania umożliwia znalezienie w krótkim czasie rozwiązania tworzącego cykl Hamiltona. Nie daje jednak gwarancji, że odnaleziona ścieżka będzie rozwiązaniem optymalnym.
- Zawyżone wartości błędu dla poszczególnych instancji użytych do testów mogą wynikać z tego, że wciąż przyjęto za małą wartość parametru α . Można dojść zatem do wniosku, że ustawianie parametrów metodą prób i błędów jest żmudne i nieefektywne – istnieje dużo możliwości ustawienia liczby z przedziału od 0 do 1 i nadal ciężko stwierdzić ile jeszcze trzeba zmienić parametr aby zbliżyć się do optimum.

Źródła

- http://155.158.112.25/~algorytmyewolucyjne/materialy/algorytm_symulowanego_wyżarzania.pdf
- https://pl.wikipedia.org/wiki/Symulowane_wyżarzanie