

Projektowanie Efektywnych Algorytmów

Projekt

20/12/2022

259091 Jakub Mordalski

(4) Simulated annealing

Spis treści	strona
Sformułowanie zadania	2
Metoda	3
Algorytm	4
Dane testowe	7
Procedura badawcza	8
Wyniki	10
Analiza wyników i wnioski	16
Źródła	17
Dodatek	18

1. Sformułowanie zadania

Celem zadania projektowego jest zaimplementowanie algorytmu rozwiązującego problem komiwojażera za pomocą metody symulowanego wyżarzania, a następnie zbadaniu jego efektywności poprzez pomiar czasu wykonania algorytmu.

Metoda symulowanego wyżarzania (ang. simulated annealing) to heurystyka oparta na mechanizmie termodynamicznym, która polega na losowym przeszukiwaniu przestrzeni rozwiązań przy jednoczesnym stopniowym zmniejszaniu "temperatury" systemu, co ma na celu zapewnienie lepszej jakości rozwiązania w końcowym etapie.

Dodatkowo należy zbadać wpływ przyjętych parametrów na czas wykonania algorytmu oraz dokładność wyniku.

Problem komiwojażera (TSP - Traveling Salesman Problem) to zagadnienie polegające na odnalezieniu minimalnego cyklu Hamiltona w grafie pełnym ważonym.

Graf pełny ważony – między każdą parą wierzchołków istnieje połączenie, a każda z krawędzi ma swoją wagę.

Cykl Hamiltona – Cykl przechodzący przez każdy wierzchołek dokładnie raz, z wyłączeniem wierzchołka początkowego, do niego wracamy po przejściu wszystkich wierzchołków.

Problem Komiwojażera polega zatem na odnalezieniu ścieżki zamkniętej, która przechodzi przez wszystkie wierzchołki, kończąc w miejscu początkowym. Koszt ścieżki powinien być jak najmniejszy.

TSP występuje w dwóch wersjach:

Symetryczny - opiera się na grafie nieskierowanym.

Asymetryczny – opiera się na grafie skierowanym.

2. Metoda

Zastosowana w zadaniu metoda symulowanego wyżarzania (ang. simulated annealing) to heurystyka służąca do rozwiązywania problemów optymalizacyjnych. Jest on inspirowany procesem wyżarzania metalu, w którym materiał jest stopniowo ogrzewany i schładzany, co pozwala mu uzyskać optymalną strukturę krystaliczną. W algorytmie symulowanego wyżarzania stosuje się podobne podejście, ale zamiast ogrzewać i schładzać fizyczny materiał, "ogrzewa" się i schładza rozwiązanie problemu, aby znaleźć optymalne rozwiązanie.

Przy rozwiązaniu problemu komiwojażera metodą symulowanego wyżarzania należy przyjąć pewien początkowy sposób przejścia przez instancję. Następnie algorytm symulowanego wyżarzania losowo modyfikuje to rozwiązanie, np. poprzez zamianę kolejności odwiedzania dwóch miast. Nowe rozwiązanie jest porównywane z poprzednim i jeśli jest lepsze (czyli prowadzi do krótszej ścieżki), to jest automatycznie akceptowane. Jeśli jednak nowe rozwiązanie jest gorsze, to algorytm symulowanego wyżarzania akceptuje je z pewnym prawdopodobieństwem, które maleje wraz ze spadkiem temperatury (tzw. "schładzaniem").

Algorytm symulowanego wyżarzania składa się z następujących kroków:

1. Ustawienie parametrów wykonania algorytmu, takich jak: temperatura początkowa, wielkość epoki, mnożnik schładzania (w przypadku geometrycznego schematu schładzania), oraz wybranie początkowego rozwiązania (w naszym przypadku jest to rozwiązanie losowe).
2. Losowe modyfikowanie obecnego rozwiązania, aby uzyskać nowe rozwiązanie. Można to zrobić na różne sposoby, np. poprzez zamianę kolejności odwiedzania dwóch miast lub dodanie lub usunięcie miasta z trasy. Ważne, aby zachować warunek, że każde miasto zostanie odwiedzione dokładnie raz.
3. Porównanie nowego rozwiązania z poprzednim i akceptacja go, jeśli jest lepsze, lub akceptacja z pewnym prawdopodobieństwem, jeśli jest gorsze. Aby ocenić, czy nowe rozwiązanie jest lepsze, należy obliczyć długość ścieżki dla obu rozwiązań i porównać je. Jeśli nowe rozwiązanie prowadzi do krótszej ścieżki, to jest ono automatycznie akceptowane. Jeśli jest gorsze, to należy obliczyć prawdopodobieństwo akceptacji tego rozwiązania na podstawie różnicy między nowym a poprzednim rozwiązaniem oraz aktualnej temperatury. Prawdopodobieństwo akceptacji gorszego rozwiązania jest większe, im wyższa jest temperatura, a mniejsze, im niższa.
4. Akceptacja nowego rozwiązania z prawdopodobieństwem obliczonym w poprzednim kroku, lub pozostawienie obecnego rozwiązania bez zmian. Można to zrobić poprzez wygenerowanie losowej liczby z zakresu od 0 do 1 i porównanie jej z obliczonym wcześniej prawdopodobieństwem.

Jeśli losowa liczba jest mniejsza od prawdopodobieństwa, to nowe rozwiązanie jest akceptowane, w przeciwnym razie pozostawia się obecne rozwiązanie bez zmian.

5. Schłodzenie temperatury zgodnie z ustalonym schematem. Dla schematu geometrycznego wartość temperatury mnożona jest przez a^k Gdzie a oznacza zdefiniowany wcześniej mnożnik temperatury a k – rozmiar epoki,

Natomiast dla schematu Boltzmana temperatura dzielona jest przez $1 + \log(1 + k)$ gdzie k oznacza rozmiar epoki

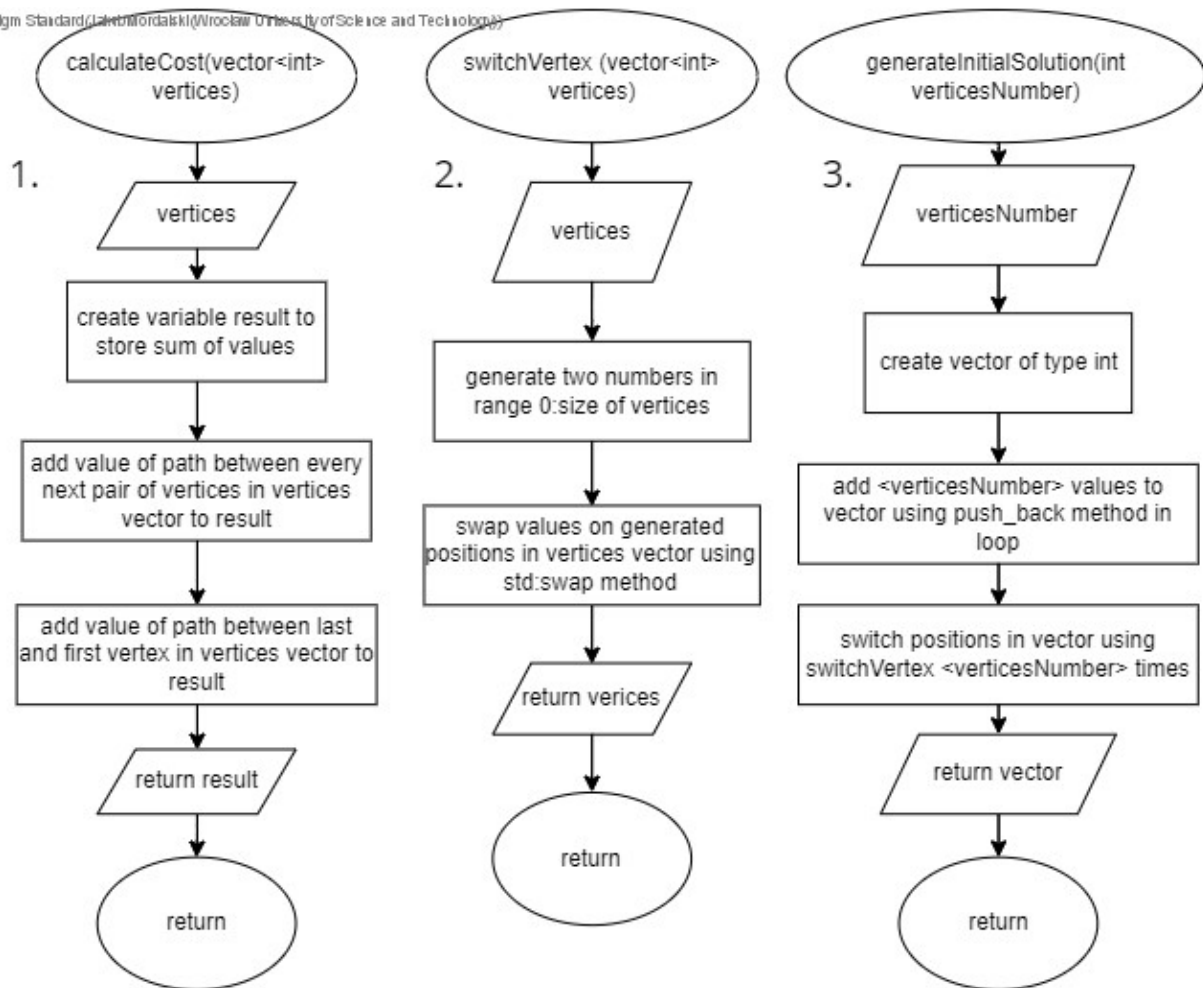
6. Powtarzanie kroków 2-5 aż do osiągnięcia zadanej temperatury (w naszym przypadku ta temperatura wynosi 1).

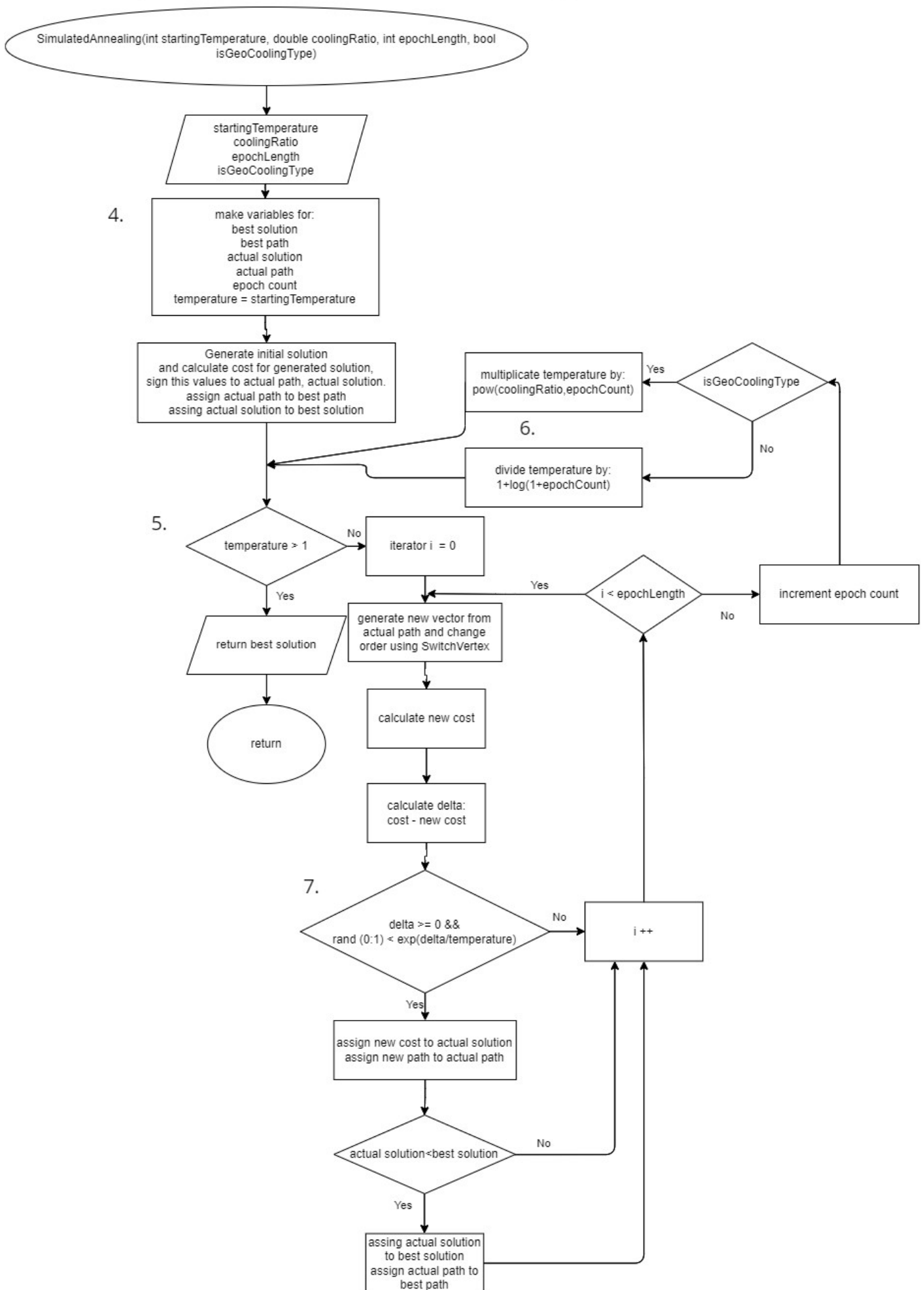
Numer epoki ustalany jest na podstawie ilości zmian przeszukiwanych ścieżek oraz określonym wcześniej rozmiarze epoki. Przykładowo, po 100 iteracjach dla rozmiaru epoki 100 znajdziemy się w epoce numer 1, po kolejnych 100 w epoce numer 2 itd.

Na koniec algorytm symulowanego wyżarzania zwraca najlepsze znalezione rozwiązanie dla podanego zestawu miast i ich odległości. Należy pamiętać, że algorytm symulowanego wyżarzania nie gwarantuje znalezienia optymalnego rozwiązania, ale zwykle daje dobre wyniki w porównaniu do innych algorytmów optymalizacyjnych.

3. Algorytm

Virtual Paradigm Standard (Lubomirski/Wrocław University of Science and Technology)





- 1.funkcja CalculateConst odpowiada za obliczanie kosztu ścieżki dla podanego w argumencie wektora wierzchołków. Wartości ścieżek między parami wierzchołków pobierane są z przechowywanej w klasie macieży sąsiedztwa.
2. Funkcja switchVertex odpowiada za zamienienie kolejności w podanym wektorze, realizowane jest to przez wylosowanie dwóch indexów z zakresu wektora a następnie zamienienie ich ze sobą miejscami.
3. Funkcja generateInitialSolution generuje wektor wierzchołków o zadanej wielkości, następnie zamienia wierzchołki podaną ilość razy w celu uzyskania losowej ścieżki.
4. Deklarowanie początkowych wartości oraz zmiennych pomocnych przy wykonaniu algorytmu.
5. Warunek stopu – temperatura mniejsza od 1.
6. Wybór sposobu chłodzenia.
7. Decyzja o przyjęciu nowoznalezionego rozwiązania.

4. Dane testowe

Do sprawdzenia poprawności działania algorytmu wybrano następujący zestaw instancji:
zestawy podano w następującym systemie:

<nazwa pliku> <optimalna długość ścieżki> <optimalna ścieżka>

W przypadku niektórych plików optimalna ścieżka nie była podana.

- tsp_6_2.txt 80 {0 5 1 2 3 4 0}
- tsp_10.txt 212 {0 3 4 2 8 7 6 9 1 5 0}
- tsp_12.txt 264 {0 1 8 4 6 2 11 9 7 5 3 10 0}
- tsp_13.txt 269 {0 10 3 5 7 9 11 2 6 4 8 1 12 0}
- tsp_14.txt 282 {0 10 3 5 7 9 13 11 2 6 4 8 1 12 0}
- tsp_15.txt 291 {0 10 3 5 7 9 13 11 2 6 4 8 14 1 12 0}
- tsp_17.txt 39 {0 11 13 2 9 10 1 12 15 14 5 6 3 4 7 8 16 0}
- gr17.tsp 2085
- gr21.tsp 2707
- gr48.tsp 5046
- gr120.tsp 6942
- si175.tsp 21407
- gr96.txt 55209
- kroA100.txt 21282
- kroB150.txt 26130
- pr152.txt 73682
- ftv170.txt 2755
- kroB200.txt 29368
- rbg323.txt 1326
- pa561.tsp 2763

źródła :

<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

<http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>

5. Procedura badawcza

Procedura badawcza polegała na uruchomieniu algorytmu dla instancji o różnych wielkościach, różnych parametrów a następnie zmierzeniu czasu działania algorytmu dla tych wartości. Pomiar czasu zrealizowany został za pomocą odczytywania różnicy czasu systemowego przed rozpoczęciem algorytmu oraz po jego zakończeniu a następnie obliczeniu różnicy. Parametry występujące w algorytmie to :

coolingRatio :[0,9 ; 0,99 ; 0,999 ; 0,9999]

initialTemperature:[10 ; 100 ; 1000 ; 10000]

initialTemperature:[10 ; 100 ; 1000 ; 10000]

Działanie algorytmu zostało przetestowane dla wszystkich kombinacji parametrów. Przy uruchomieniu używany jest plik sterujący conf.ini, zawierający następujące dane <nazwa pliku> <ilość testów> <wielkość instancji> <optimalna długość ścieżki> <optimalna ścieżka>. optymalna ścieżka o wartości -1 oznacza brak tego parametru

Zawartość pliku użytego podczas testów :

tsp_6_2.txt 10 6 80 0 5 1 2 3 4

tsp_10.txt 10 10 212 0 3 4 2 8 7 6 9 1 5

tsp_12.txt 10 12 264 0 1 8 4 6 2 11 9 7 5 3 10

tsp_13.txt 10 13 269 0 10 3 5 7 9 11 2 6 4 8 1 12

tsp_14.txt 10 14 282 0 10 3 5 7 9 13 11 2 6 4 8 1 12

tsp_15.txt 10 15 291 0 10 3 5 7 9 13 11 2 6 4 8 14 1 12

tsp_17.txt 10 17 39 0 11 13 2 9 10 1 12 15 14 5 6 3 4 7 8 16

gr17.tsp 10 17 2085 -1

gr21.tsp 10 21 2707 -1

gr48.tsp 10 48 5046 -1

gr120.tsp 10 120 6942 -1

si175.tsp 10 175 21407 -1

gr96.txt 10 96 55209 -1

kroA100.txt 10 100 21282 -1

kroB150.txt 10 150 26130 -1

pr152.txt 10 152 73682 -1

ftv170.txt 10 171 2755 -1

kroB200.txt 10 200 29368 -1

rbg323.txt 10 323 1326 -1

pa561.tsp 10 561 2763 -1

Plik wyjściowy

Wyniki Powyższego testu zostały zapisane w plikach resultAll.csv oraz resultAvg.csv, które zawierały następujące dane :

<długość ścieżki> <procent błędu> <czas wykonania> <temperatura początkowa>
<mnożnik chłodzenia><rozmiar epoki>,

Testy dla poszczególnych plików zostały oddzielone linią zawierającą:

<nazwa pliku> <ilość testów> <wielkość instancji> <optimalna długość ścieżki> <optimalna ścieżka>

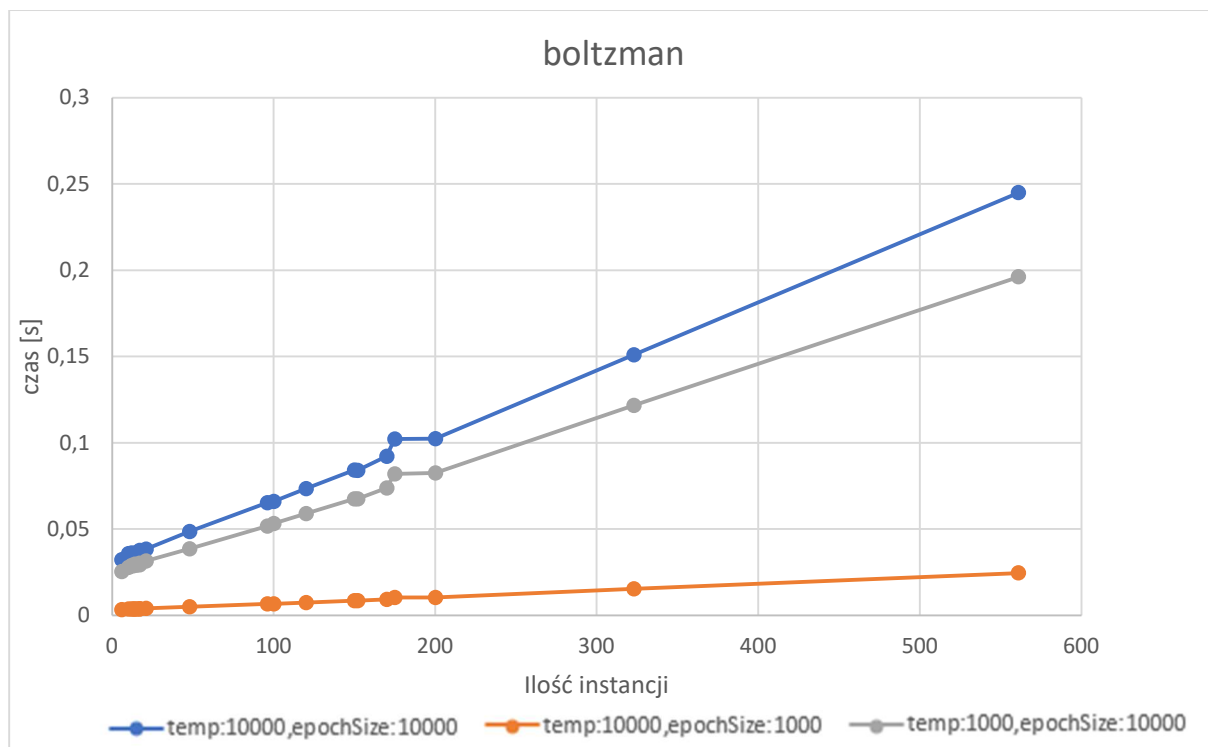
Test był przeprowadzany na platformie

- AMD Ryzen 5 3500X 4.1 GHz
- 16GB RAM DDR4 3400MHz CL16
- Samsung 500GB M.2 PCIe NVMe 970 EVO Plus

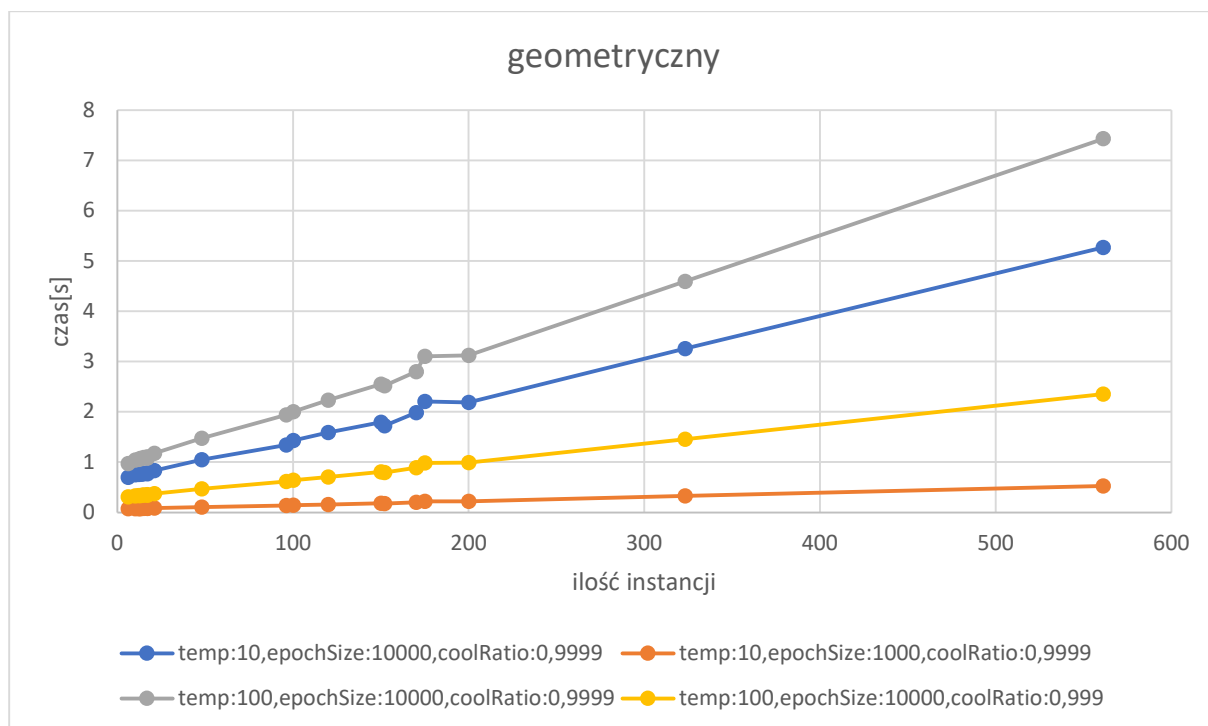
6. Wyniki

Wyniki działania programu zostały zapisane w plikach resultAll.csv, resultAvg.csv, zostaną dołączone do sprawozdania.

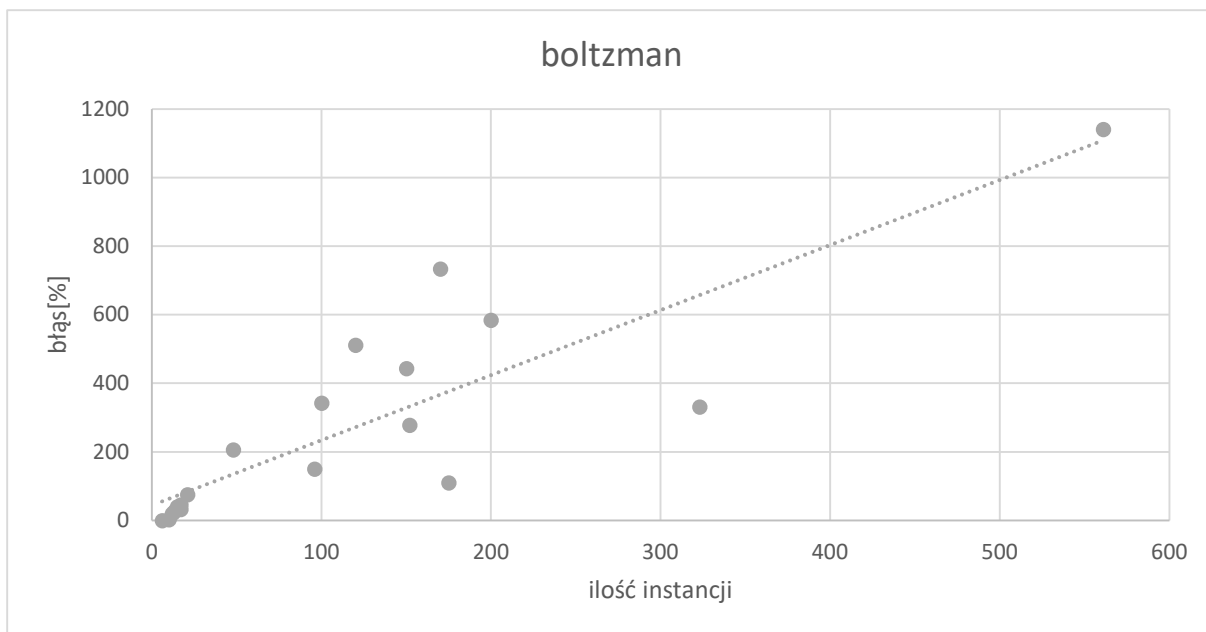
Wpływ metody wychładzania na czas oraz błędy.



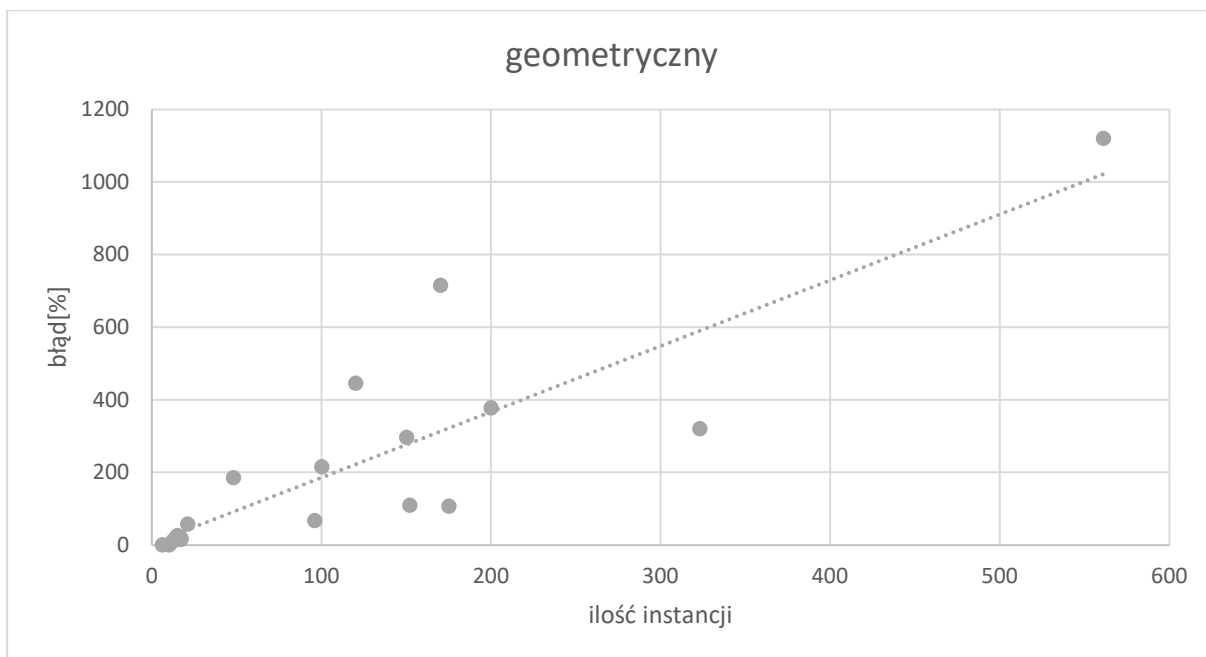
Rysunek 1: Wykres czasu działania algorytmu Simulated annealing schemat chłodzenia boltzman względem ilości instancji



Rysunek 2: Wykres czasu działania algorytmu Simulated annealing schemat chłodzenia geometryczny względem ilości instancji

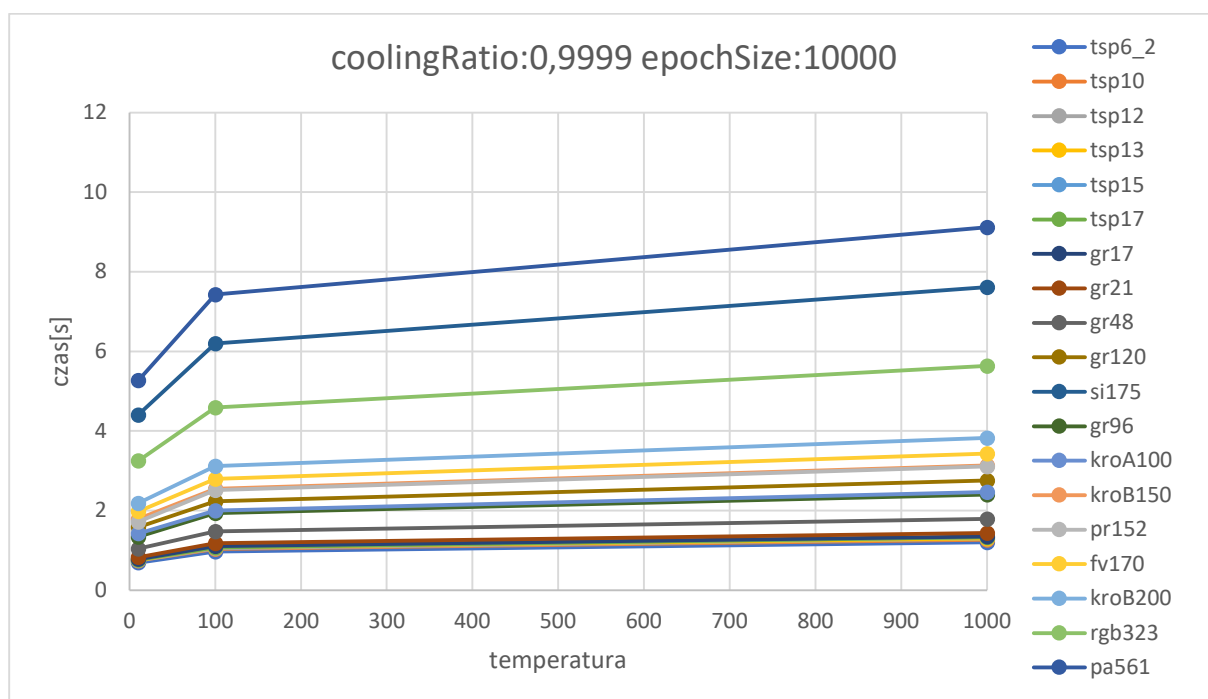


Rysunek 3: Wykres błędu algorytmu Simulated annealing schemat chłodzenia boltzman względem ilości instancji

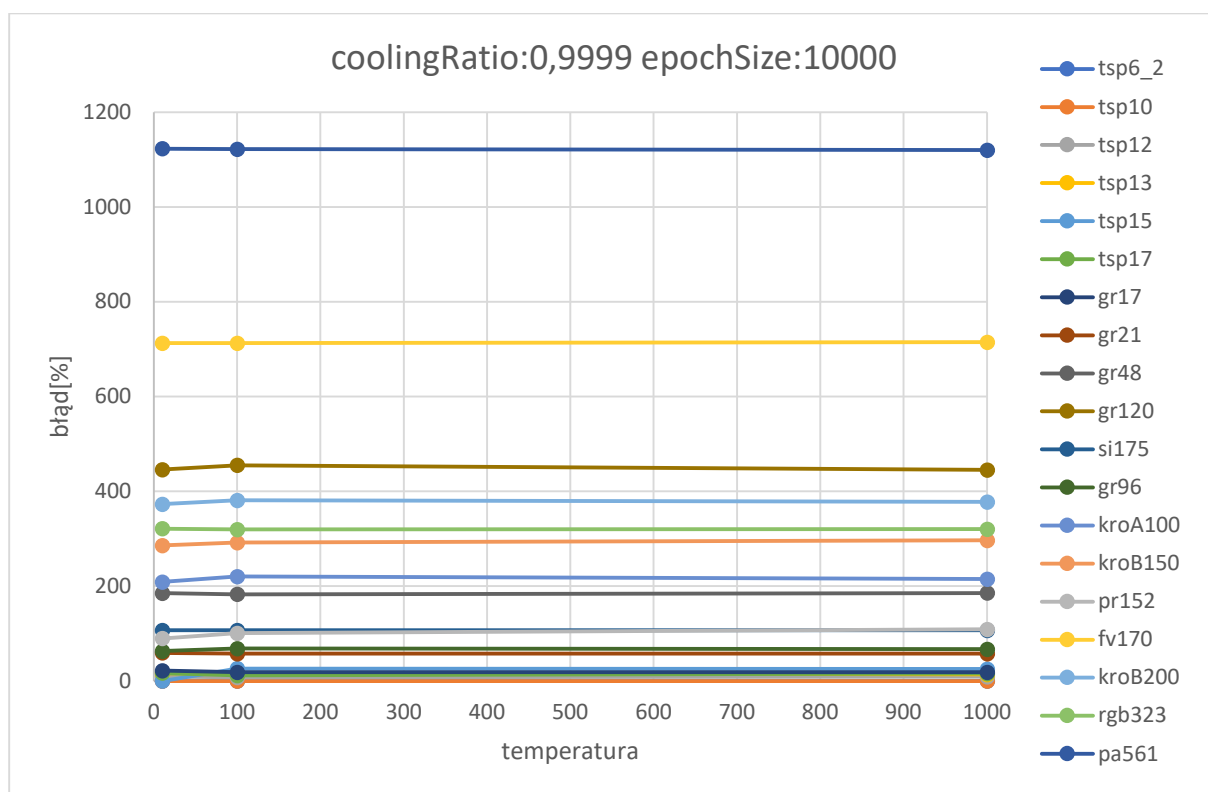


Rysunek 4: Wykres błędu algorytmu Simulated annealing schemat chłodzenia geometryczny względem ilości instancji

Wpływ temperatury na czas oraz błąd – geometryczny

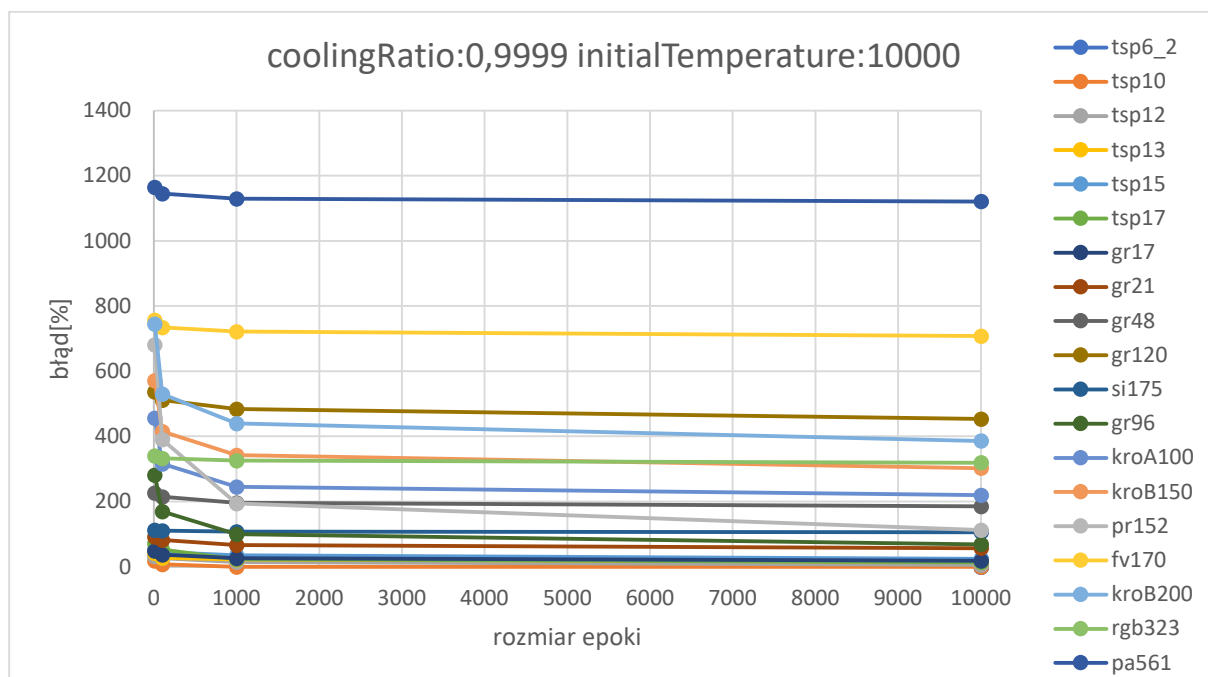


Rysunek 5: Wykres czasu wykonania algorytmu Simulated annealing schemat chłodzenia geometryczny względem temperatury

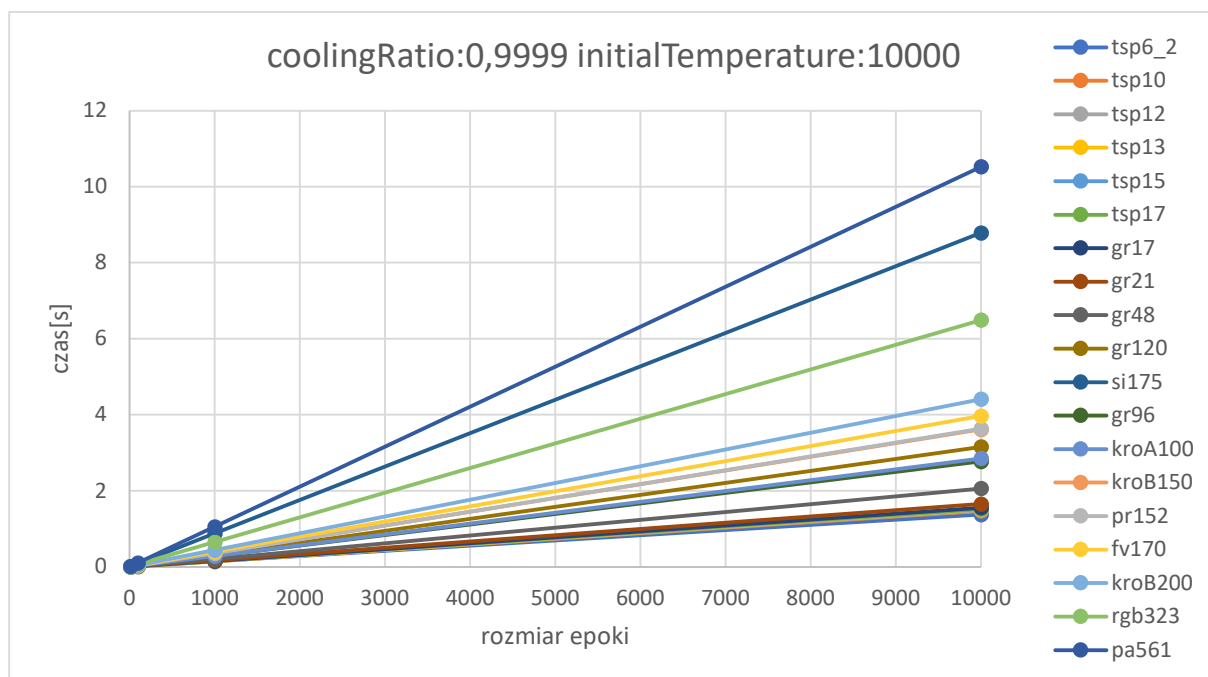


Rysunek 6: Wykres błędu algorytmu Simulated annealing schemat chłodzenia geometryczny względem temperatury

Wpływ rozmiaru epoki na czas oraz błąd – geometryczny

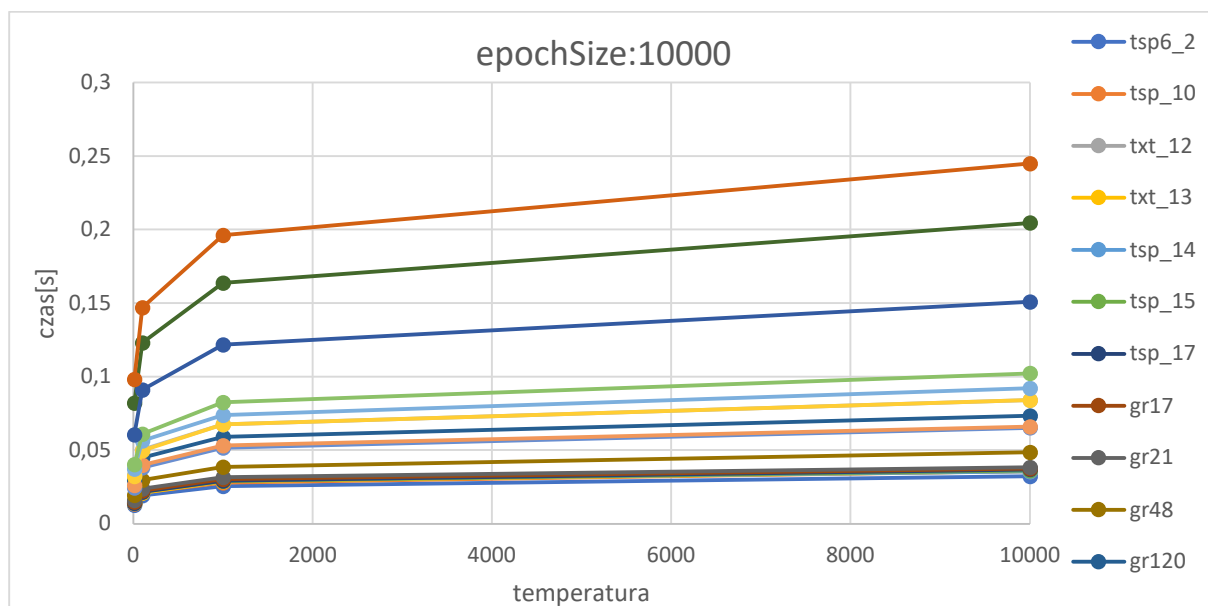


Rysunek 7: Wykres błędów algorytmu Simulated annealing schemat chłodzenia geometryczny względem rozmiaru epoki

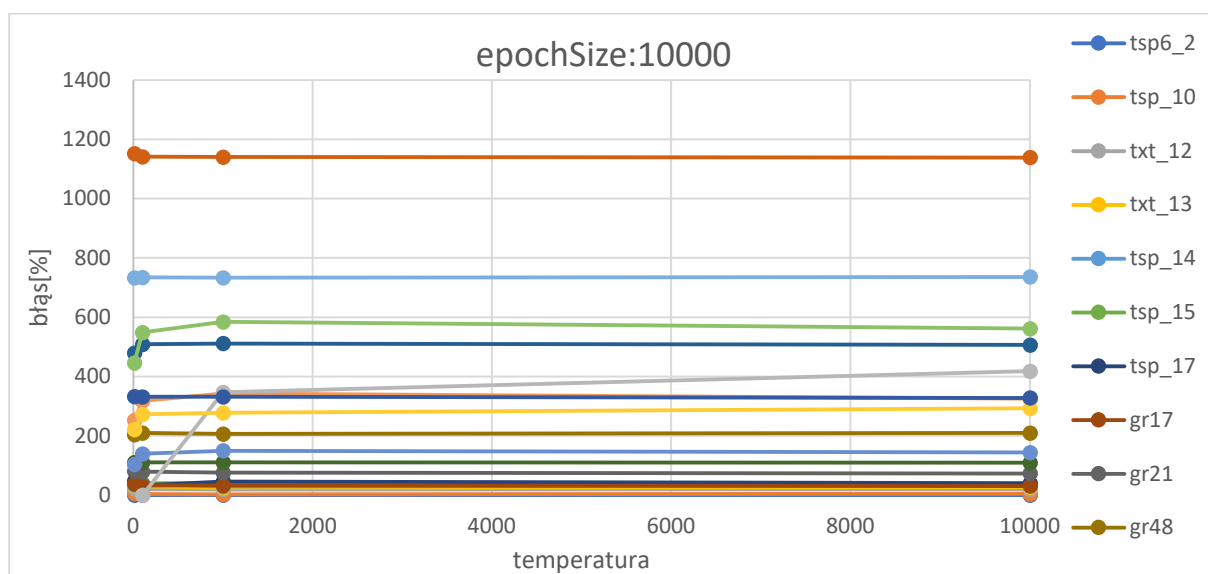


Rysunek 8: Wykres czasu wykonania algorytmu Simulated annealing schemat chłodzenia geometryczny względem rozmiaru epoki

Wpływ temperatury na czas oraz błąd – boltzman

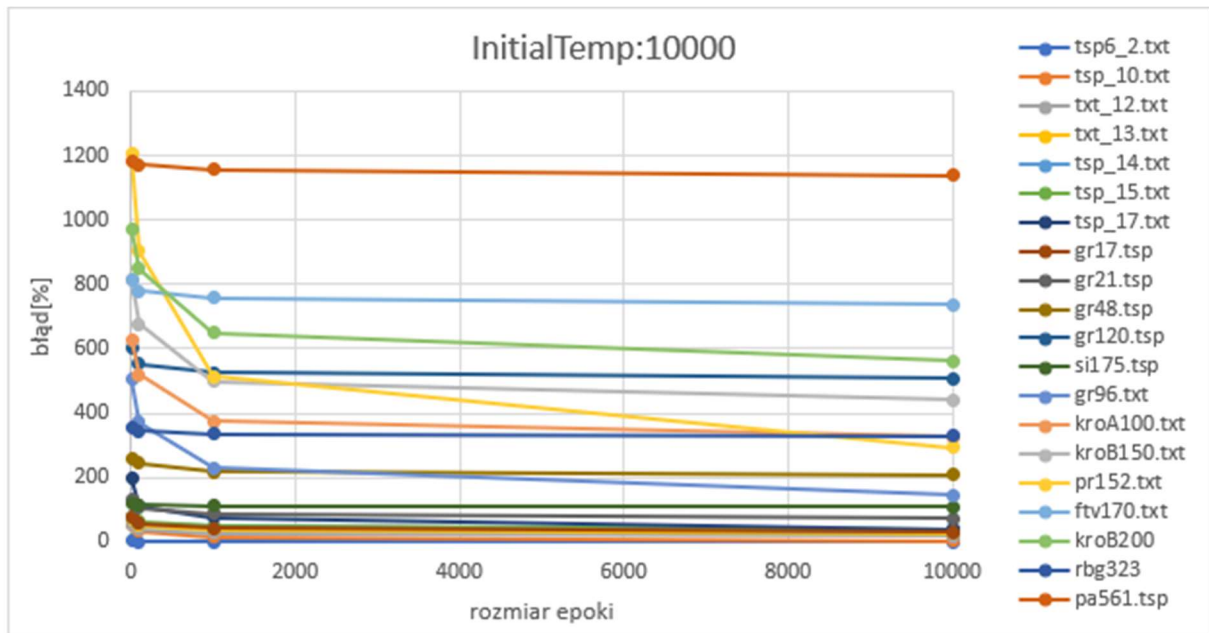


Rysunek 9: Wykres czasu wykonania algorytmu Simulated annealing schemat chłodzenia boltzmana względem temperatury

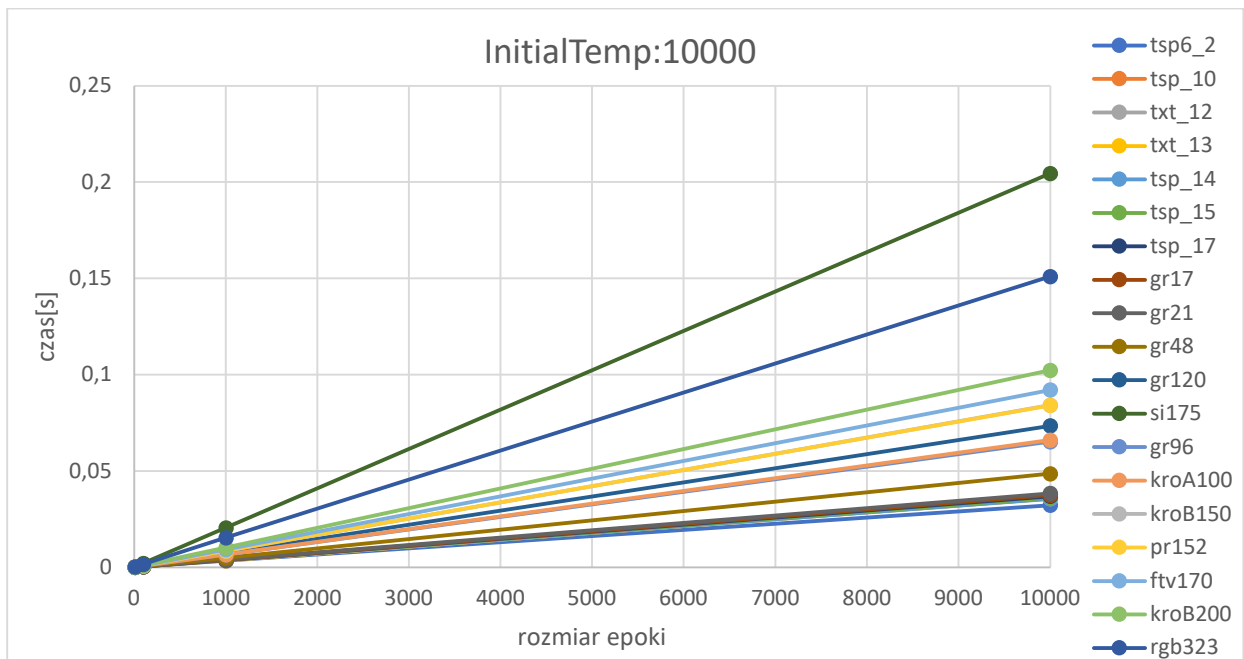


Rysunek 10: Wykres błędów algorytmu Simulated annealing schemat chłodzenia boltzmana względem temperatury

Wpływ rozmiaru epoki na czas oraz błąd – boltzman



Rysunek 11: Wykres błędów algorytmu Simulated annealing schemat chłodzenia boltzmana względem rozmiaru epoki



Rysunek 12: Wykres czasu wykonania algorytmu Simulated annealing schemat chłodzenia boltzmana względem rozmiaru epoki

7. Analiza wyników i wnioski

Przedstawione powyżej wykresy sugerują różne złożoności obliczeniowe dla obydwu schematów chłodzenia.

Stosowane oznaczenia :

n – wielkość instancji,

To – temperatura początkowa,

L – rozmiar epoki,

a – współczynnik chłodzenia

Z wykresów można odczytać :

Wpływ L na czas wykonania jest liniowy – im większa epoka tym większy czas wykonania (rys.8,12)

Wpływ L na błąd jest logarytmiczny – im większe L tym mniejsza poprawa efektywności (rys.7,11)

Wpływ To na błąd jest znikomy (rys.6,10)

Wpływ To na czas wykonania jest logarytmiczny (rys.5,9).

Złożoność czasowa dla zmiennej ilości instancji jest liniowa (rys.1,2) a więc wynosi $O(n)$

Zależności dotyczą zarówno schematu chłodzenia boltzmana i geometrycznego. Główną różnicą między algorytmami jest czas działania który dla schematu boltzmana jest ogólnie mniejszy.

Podsumowując zebrane informację dochodzimy do wniosku:

Złożoność czasowa algorytmu z geometrycznym schematem chłodzenia wynosi $O(n \log(T_o)L)$

Złożoność czasowa algorytmu z schematem chłodzenia boltzmana wynosi $O(nL)$

Złożoność pamięciowa jest trudna do przetestowania ponieważ trudno jest odizolować zużycie pamięci przez sam algorytm od reszty procesów towarzyszących wykonywaniu testów. Wiele natomiast zależy od zaimplementowanych struktur, sposobu przechowywania macierzy sąsiedztwa itp.

Błąd był badany dla wszystkich kombinacji parametrów, ciężko wyciągnąć jednolite wnioski ponieważ rozstrzał przy porównaniu ilości błędów do liczby instancji jest duży, rysując jednak linię trendu można dostrzec liniowy wzrost liczby błędów wraz ze wzrostem liczby instancji. Przy założeniu że skuteczność ma wynosić co najmniej 30%, dla testowanych parametrów algorytm traci skuteczność w okolicach $n=17$, w przypadku schematu schładzania boltzmana, oraz w okolicach $n=20$ dla geometrycznego schematu.

Ogólny wniosek z porównania dwóch schematów jest następujący : Schemat boltzmana dla podobnych parametrów oferuje mniejszą dokładność w mniejszym czasie w porównaniu ze schematem geometrycznym.

8. Źródła

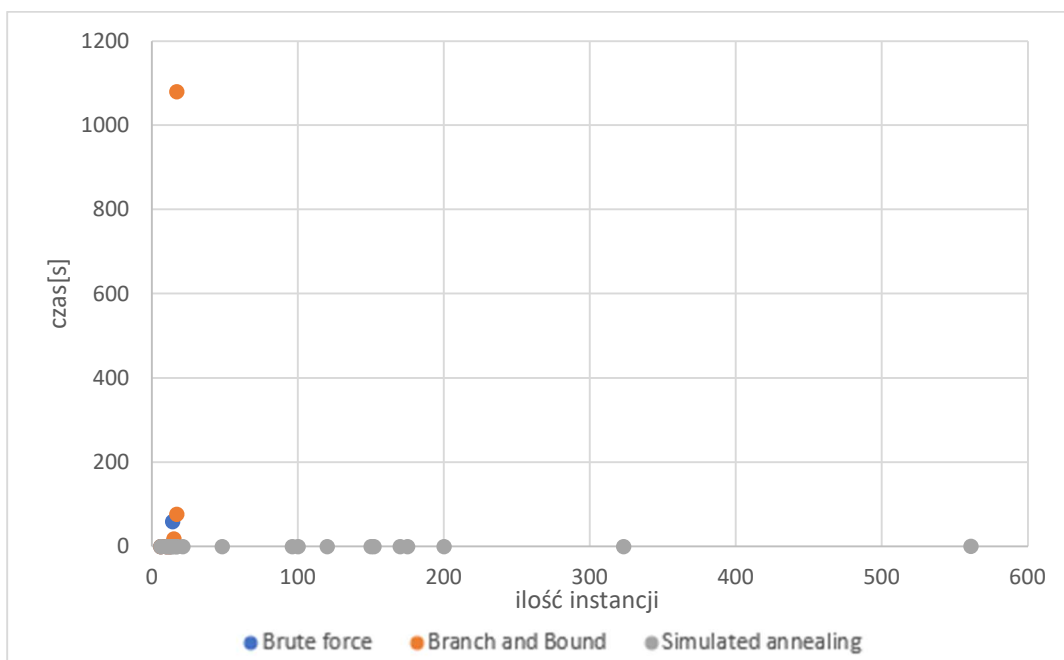
https://pl.wikipedia.org/wiki/Problem_komiwoja%C5%BCera
https://pl.wikipedia.org/wiki/Cykl_Hamiltona
<http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>
https://en.wikipedia.org/wiki/Simulated_annealing

9. Dodatek

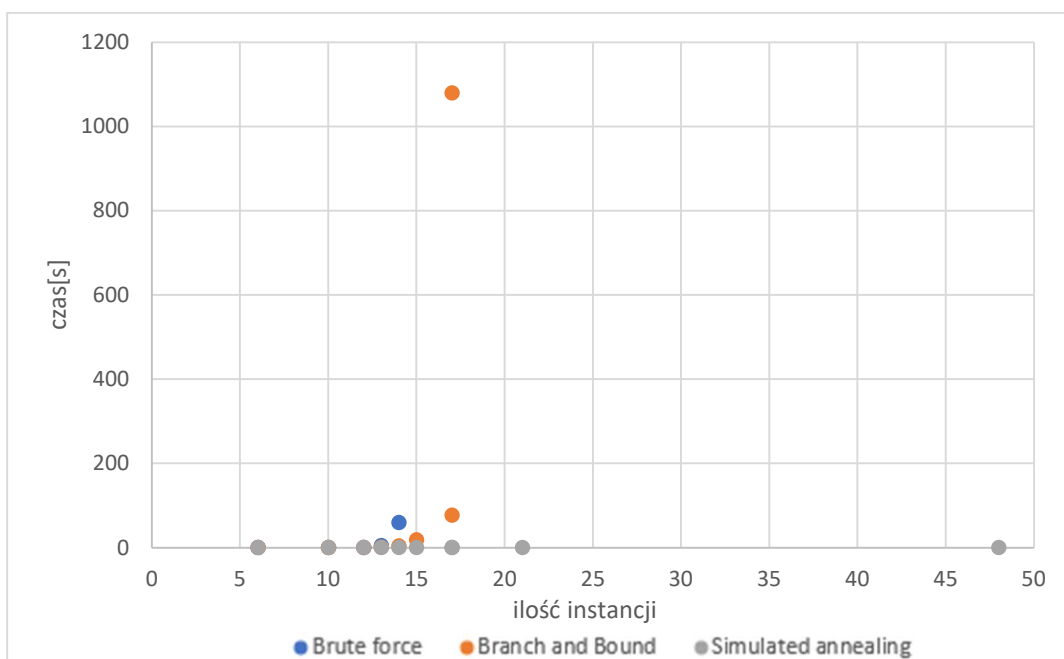
Porównując dotychczas analizowane algorytmy dochodzimy do następującego wniosku:

Algorytmy dokładne – gwarantujące 100 % poprawność rozwiązania są mało efektywne w przypadku większych instancji.

Mamy jednak opcję zrezygnowania z 100% dokładności oszczędzając przy tym znaczną ilość czasu korzystając z innych algorytmów takich jak chociażby symulowane wyżarzanie.



Rysunek 13: Wykres zawierający czas wykonania algorytmu dla dotychczas zaimplementowanych zadań



Rysunek 14: Wykres zawierający czas wykonania algorytmu dla dotychczas zaimplementowanych zadań - przybliżenie

Na powyższych wykresach umieszczono czasy wykonania wszystkich implementowanych algorytmów, doskonale widać na nich jak szybko rośnie czas wykonania algorytmów dokładnych w porównaniu do symulowanego wyżarzania.

Dzieje się tak głównie z powodu ilości wymaganych operacji.

W przypadku algorytmów dokładnych przeglądana jest znaczna część lub wszystkie kombinacje występujące w instancji co powoduje wykładniczy wzrost czasu wykonania.

W algorytmie opartym na mechanizmie termodynamicznym na czas wykonania główny wpływ mają przyjęte parametry początkowe, sposób wychładzania itp. Zwiększanie rozmiaru instancji liniowo wpływa na czas wykonania algorytmu.