Projektowanie Efektywnych Algorytmów
Projekt

23/12/2022

259198 Lena Kuźma

1) Simulated Annealing

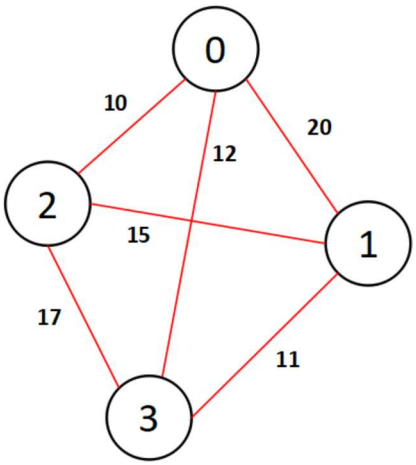
1. Sformułowanie zadania

Zadanie polega na opracowaniu i implementacji algorytmu rozwiązywana problemu komiwojażera z wykorzystaniem metody symulowanego wyżarzania (*Simulated Annealing)*. Problem komiwojażera (eng. TSP – Travelling Salesman Problem) polega na znalezieniu minimalnego cyklu Hamiltona (każdy wierzchołek grafu odwiedzany jest dokładnie raz) w pełnym grafie ważonym (*Rysunek 1*). W rozwiązaniu należy zbadać wpływ:

- sposobu wyboru temperatury początkowej,
- sposobu wyboru rozwiązania początkowego,
- schematu chłodzenia,
- długości epoki oraz
- sposobu wyboru rozwiązania w sąsiedztwie rozwiązania bieżącego

na:

- jakość uzyskiwanych rozwiązań (różnicy pomiędzy rozwiązaniem uzyskanym i optymalnym),
- czas uzyskiwania rozwiązań oraz
- zużycie pamięci.



Rysunek 1: Graf pełny ważony - symboliczna reprezentacja problemu komiwojażera

2. Metoda

Symulowane wyżarzanie to jedna z technik projektowania algorytmów heurystycznych. Cechą charakterystyczną tej metody jest występowanie parametru sterującego zwanego temperaturą, który maleje w trakcie wykonywania algorytmu.

SA należy do technik przeszukiwania sąsiedztwa. Jest modyfikacją algorytmu zachłannego. Zamiast pełnego przeglądu sąsiedztwa bieżącego rozwiązania, kolejne rozwiązanie jest wybierane w sposób losowy.

Sąsiedztwo rozwiązania x jest zbiorem rozwiązań, które mogą być osiągnięte z x drogą prostej operacji zwanej ruchem. Jeśli rozwiązanie y jest lepsze od dowolnego z jego sąsiedztwa, wtedy y jest optimum lokalnym.

Pierwszym krokiem algorytmu jest przyjęcie rozwiązania początkowego, które jest modyfikowane w kolejnych krokach. Jeśli w danym kroku uzyskamy rozwiązanie lepsze, wybieramy je zawsze. Istotną cechą symulowanego wyżarzania jest jednak to, że z pewnym prawdopodobieństwem może być również zaakceptowane rozwiązanie gorsze (ma to na celu umożliwienie wyjście z ekstremum lokalnego).

Prawdopodobieństwo przyjęcia gorszego rozwiązania wyrażone jest wzorem:

$$exp\left(-\frac{|f(y)-f(x)|}{T}\right)$$

Gdzie:

x – rozwiązanie poprzednie

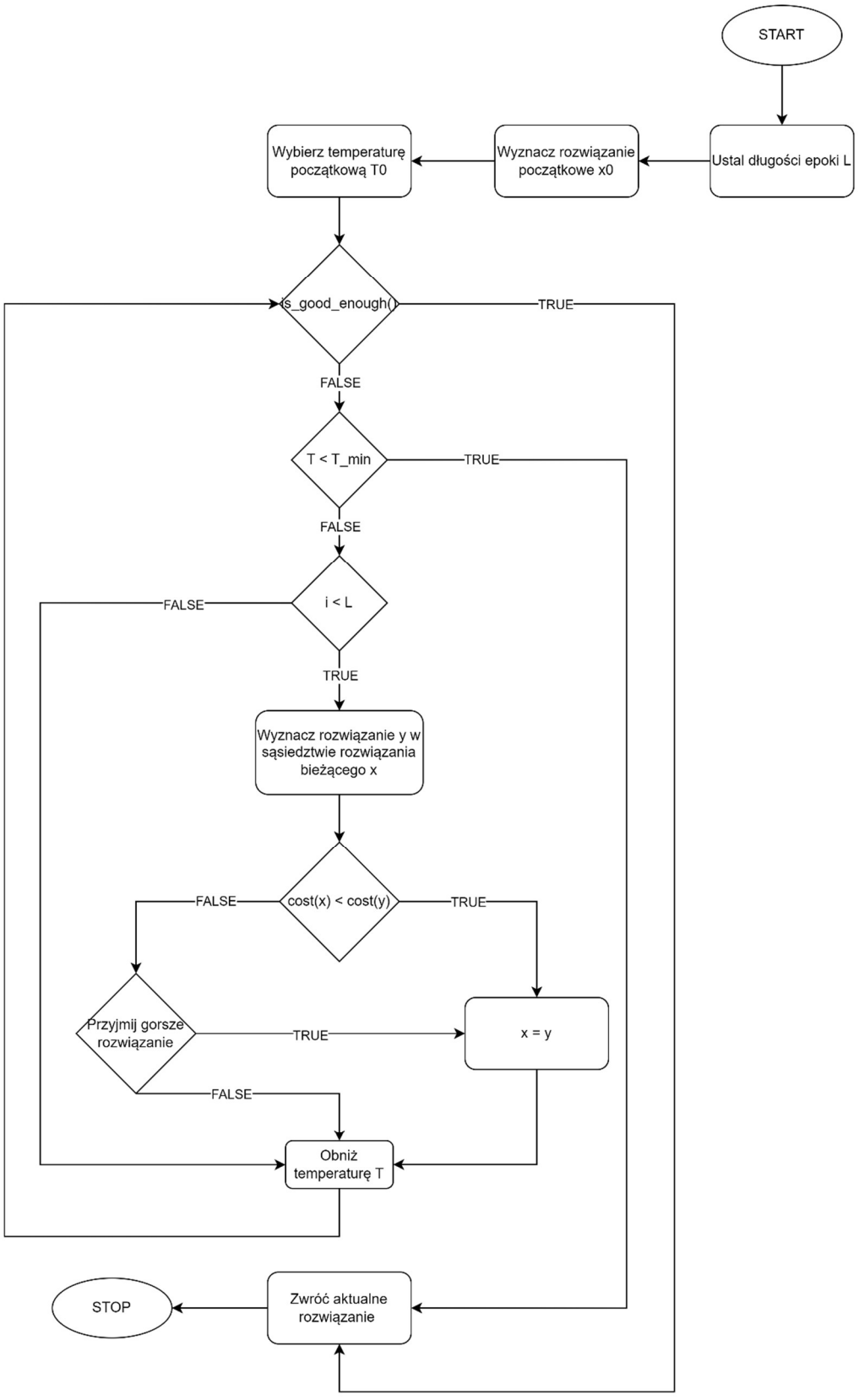
y – rozwiązanie bieżące

T – aktualna temperatura

f(x) – funkcja oceny jakości

Prawdopodobieństwo przyjęcia gorszego rozwiązania spada wraz ze spadkiem temperatury i wzrostem różnicy jakości obu rozwiązań.

3. Algorytm



Rysunek 2: Schemat blokowy opisujący działanie zastosowanego algorytmu.

- Temperatura początkowa była wybierana na dwa sposoby:
  - Zależnie od kosztu rozwiązania początkowego:

$$T\_max = f(x^0) * w$$

f(x^0) – koszt rozwiązania początkowego
w – dowolny współczynnik

  - Jako stała:

$$T\_max = C$$

C - stała

- Testy zostały przeprowadzone dla dwóch schematów chłodzenia (funkcji zmiany temperatury):
  - Schematu logarytmicznego (Boltzmanna):

$$\alpha(T) = \frac{T}{a + b \log k}$$

a, b – pewne współczynniki

k – nr bieżącej iteracji algorytmu

T - aktualna temperatura

  - Schematu geometrycznego:

$$\alpha(T) = a^k T$$

0 < a < 1

T - aktualna temperatura

- Długość epoki była wyznaczana jako % wielkości sąsiedztwa. Sąsiedztwo definiujemy jako wszystkie rozwiązania możliwe do uzyskania z rozwiązania bieżącego, po wykonaniu ruchu (np. 2-zamiana).

- Rozwiązanie w sąsiedztwie rozwiązania bieżącego było wybierane na dwa sposoby:
  - 2-zamiana – Losujemy dwa wierzchołki i zamieniamy je miejscami.
  - Zamiana łuków – Losujemy dwa wierzchołki. Następnie odwracamy kolejność łuku zaczynającego się w jednym wylosowanym wierzchołku, a kończącego się w drugim.

- Rozwiązanie początkowe było wyznaczane przy pomocy algorytmu *greedy*.

4. Dane testowe

Do sprawdzenia poprawności działania algorytmu oraz do dostrojenia (określenia wartości parametrów algorytmu) wybrano następujący zestaw instancji:

ftv47.atsp

ftv170.atsp

ftv403.atsp

Źródło: TSPLIB (uni-heidelberg.de)

Do wykonania badań wybrano następujący zestaw instancji:

br17.atsp

ftv47.atsp

ftv64.atsp

kro124.atsp

ftv170.atsp

rbg323.atsp

Źródło: TSPLIB (uni-heidelberg.de)

5. Procedura badawcza

Procedura badawcza polegała na uruchomieniu programu (napisanego w języku *python*) sterowanego plikiem inicjującym *.INI* (format pliku: nazwa instancji, liczba wykonań rozwiązania, nazwa pliku wyjściowego, najlepsze znane rozwiązanie (*Rysunek 3*)).

```
[FILE]
file1 = br17.atsp.txt
file2 = ftv47.atsp.txt
file3 = ftv64.atsp.txt
file4 = kro124p.atsp.txt
file5 = ftv170.atsp.txt
file6 = rbg323.atsp.txt

[ITERATOR]
i1 = 10
i2 = 5
i3 = 5
i4 = 3
i5 = 3
i6 = 2

[OUTPUT]
file1 = SA.xlsx

[OPTIMAL_VALUE]
opt1 = 39
opt2 = 1776
opt3 = 1839
opt4 = 36230
opt5 = 2755
opt6 = 1326
```

Rysunek 3: Przykładowa zawartość pliku inicjalizującego .INI

Każda z instancji rozwiązywana była zadaną liczbę razy, np. *br17.atsp* wykonana została 10 razy. Do pliku wyjściowego *SA.xlsx* zapisywany był czas wykonania, otrzymane rozwiązanie (koszt ścieżki) oraz jego błąd procentowy. Czas wykonywania algorytmu został zmierzony przy pomocy funkcji *perf_counter()* z biblioteki *time*. Plik wyjściowy zapisywany był w formacie *xlsx.* Na *Rysunku 4* przedstawiono fragment zawartości pliku wyjściowego.

	A	B	C	D	E	F
1	Nazwa instancji		Liczba powtórzeń		Wartość optymalna	
2	ftv47.atsp.txt		5		1776	
3						
4	Czas wykonywania [s]:		Uzyskany koszt:		Błąd [%]:	
5	0,633277		2380		34,00901	
6	0,004551		2365		33,16441	
7	0,004846		2230		25,56306	
8	0,04077		2277		28,20946	
9	0,051232		2368		33,33333	

Rysunek 4: fragment pliku wyjściowego w formacie .xlsx

Wyniki opracowane zostały w programie *MS Excel*.

6. Wyniki

❖ WYBÓR TEMPERATURY POCZĄTKOWEJ

Założenia wstępne:

- Sposób wyboru rozwiązania początkowego – *algorytm greedy*
- Schemat chłodzenia – *geometryczny*
- Sposób wyboru rozwiązania w sąsiedztwie rozwiązania bieżącego – *2-zamiana*
- Długość epoki – *30% wielkości sąsiedztwa*

Badanie wpływu wyboru temperatury początkowej na:

- czas uzyskiwania rozwiązań (z błędem <= 35%):

1. *T_max = C*
   C - stała

Wielkość instancji:	T_max = 1	T_max = 5	T_max = 20	T_max = 70
17	0,002	0,005	0,173	0,976
47	4,218	0,026	0,114	4,975
64	0,059	0,047	0,589	13,168
124	0,011	0,012	0,017	0,016
170	*425,391**	7,786	125,367	302,275
323	0,141	0,134	0,172	0,131

Tabela 1: Czasy wykonywania algorytmu dla różnych wartości temperatury początkowej.

Z otrzymanych wyników (*Tabela 1*) można wywnioskować, że dla małych instancji (np. 17) lepsza jest mniejsza temperatura, ponieważ szybciej uzyskujemy oczekiwane rozwiązanie.

Zbyt duża temperatura może być niekorzystna, ponieważ częściej będzie przyjmowane gorsze rozwiązanie, co może wydłużyć czas wykonywania algorytmu. Widać to na przykładzie instancji o wielkości 64.

Oprócz rozmiaru instancji należy zwrócić również uwagę na jej złożoność. Widać to na przykładzie pliku *ftv170.atsp* oraz *rbg323.atsp*. Rozwiązanie dla instancji o wielkości 323 zostaje znalezione dużo szybciej (~100 razy) niż dla instancji o wielkości 170.

Wyniki uzyskane dla instancji *ftv64.atsp* oraz *ftv170.atsp* (podkreślone na szaro w *Tabeli 1)* sugerują, że może istnieć najlepsza temperatura dla danej instancji i zadanych warunków algorytmu (np. schemat chłodzenia). Wynika to ze znaczącej redukcji czasu wykonywania algorytmu dla wybranej temperatury początkowej (w tym przypadku T = 5), w porównaniu z pozostałymi.

** Dla instancji ftv170.atsp i T_max = 1 w dwóch na trzy uzyskane rozwiązania nie udało uzyskać się wyniku z błędem mniejszym niż 35%. Temperatura spadła do temperatury minimalnej (T_min = 0.01).*

2. *T_max = f(x^0) * w*
   f(x^0) – koszt rozwiązania początkowego
   w – wybrany współczynnik

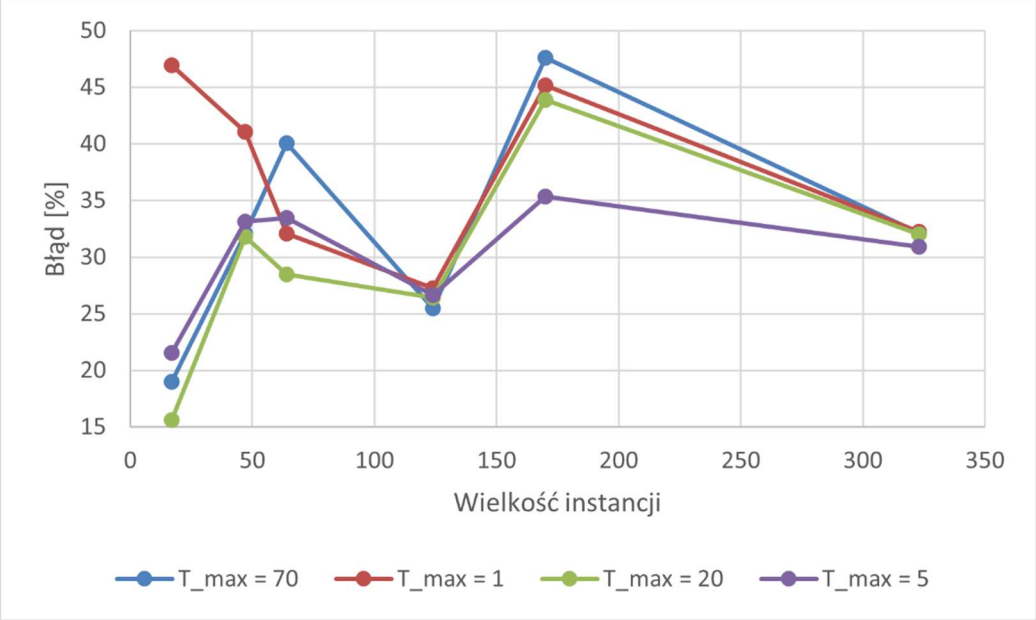
Wielkość instancji:	T = *f(x^0)* * 0.01	T = *f(x^0)* * 0.1	T = *f(x^0)* * 1	T = *f(x^0)* * 1.5
17	0,033	0,002	0,515	0,695
47	8,589	0,353	12,547	23,858
64	13,499	4,117	28,45	42,086
124	0,016	0,038	0,01	0,014
170	1275,277	260,685	921,455	1079,701
323	0,165	0,157	0,146	0,167

Tabela 2: Czasy wykonywania algorytmu dla różnych wartości temperatury początkowej.

Czas wykonywania algorytmu był dłuższy przy zastosowaniu temperatury maksymalnej zależnej od kosztu rozwiązania początkowego.

– jakość uzyskiwanych rozwiązań:

W badaniu wpływu temperatury początkowej na jakość rozwiązania został dodany kolejny warunek końcowy: czas = 10s.



Rysunek 5: Wykres zależności wielkości błędu od wielkości instancji dla zadanych wartości temperatury początkowej.

Z wykresu (*Rysunek 5*) wynika, że najmniejszy błąd miały rozwiązania, dla których temperatura początkowa wynosiła 5 lub 20. Sugeruje to, że istnieje przedział *T_max*, dla którego rozwiązanie jest znajdywane najszybciej. Im temperatura jest niższa tym prawdopodobieństwo przeszukania innej przestrzeni rozwiązań maleje. Z kolei im *T_max* jest wyższe tym przeszukiwanie odbywa się bardziej chaotycznie. W obydwu przypadkach możemy przegapić rozwiązanie spełniające zadany warunek.

❖ WYBÓR SCHEMATU CHŁODZENIA

Założenia wstępne:

- Sposób wyboru rozwiązania początkowego – *algorytm greedy*
- Temperatura początkowa – *T_max = 5*
- Sposób wyboru rozwiązania w sąsiedztwie rozwiązania bieżącego – *2-zamiana*
- Długość epoki – *30% wielkości sąsiedztwa*

Schemat logarytmiczny (Boltzmanna):

$$\alpha(T) = \frac{T}{a + b \log k}$$

a = 0.85, b = 0.95

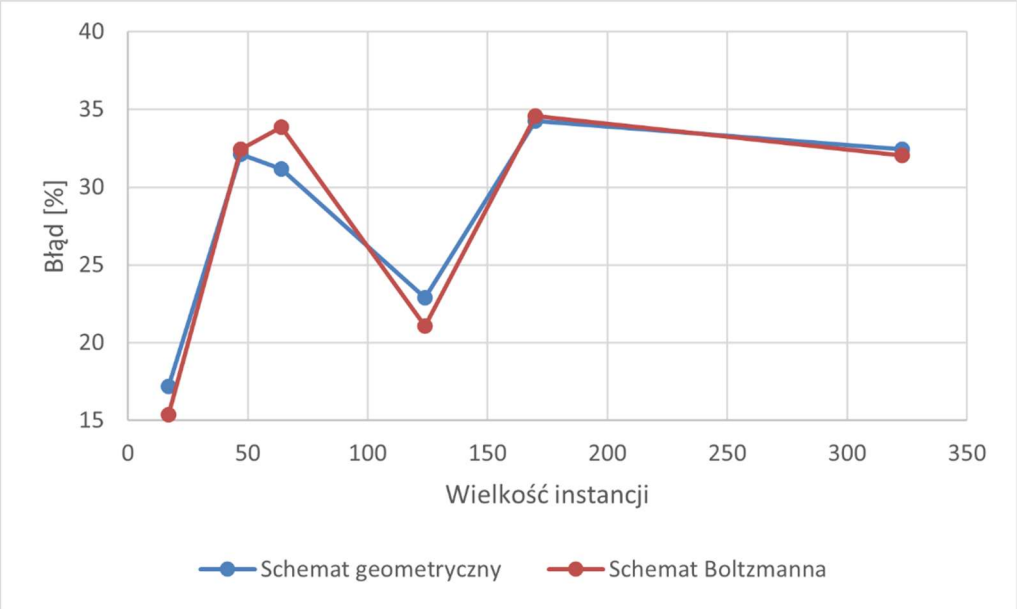
Schemat geometryczny:

$$\alpha(T) = a^k T$$

a = 0.999

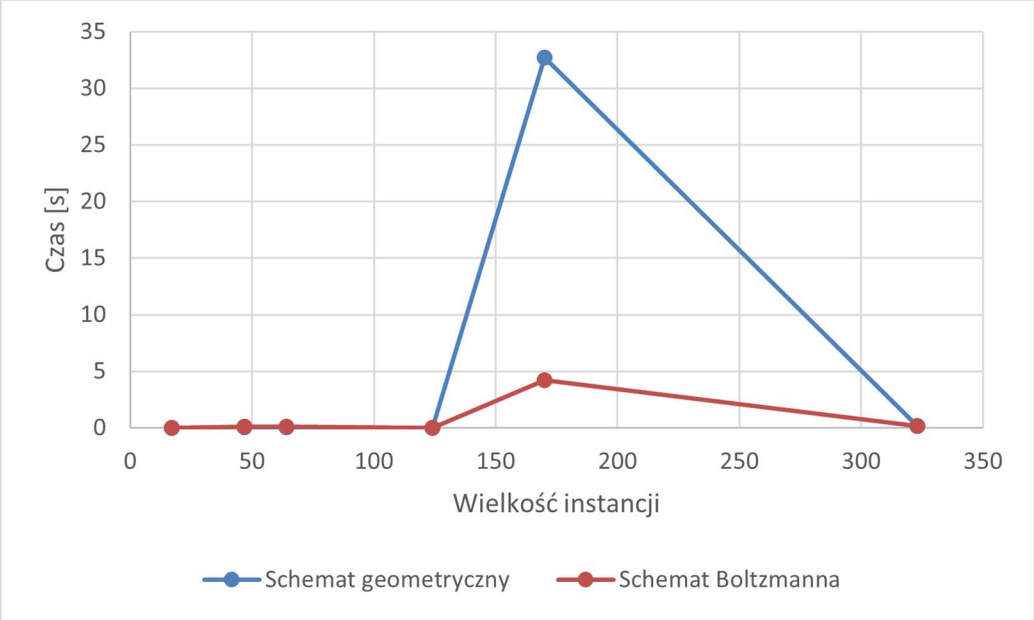
Badanie wpływu wyboru schematu chłodzenia na:

- jakość uzyskiwanych rozwiązań:



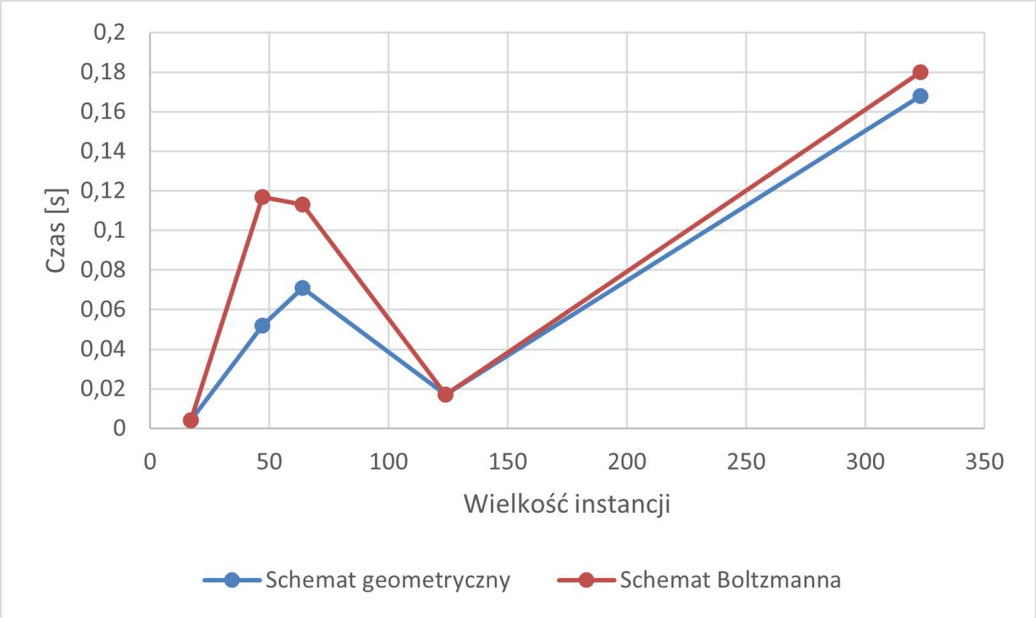
Rysunek 6: Wykres zależności błędu uzyskanego rozwiązania od wielkości instancji dla zadanego schematu chłodzenia.

– czas uzyskiwania rozwiązań (dla błędu 35%):



Rysunek 7: Wykres zależności czasu wykonywania algorytmu od wielkości instancji dla zadanych schematów chłodzenia.

Wykres z wyłączeniem instancji ftv170.atsp:



Rysunek 8: Wykres zależności czasu wykonywania algorytmu od wielkości instancji dla zadanych schematów chłodzenia.

Przy przyjętych założeniach, rozwiązania zostały uzyskane szybciej dla schematu geometrycznego (*Rysunek 7*). Wyjątkiem jest instancja *ftv170.atsp*, dla której schemat Boltzmanna sprawdził się zdecydowanie lepiej (*Rysunek 6*). Dokładność rozwiązania w obydwu przypadkach była zbliżona (*Rysunek 8*).

❖ WYBÓR ROZWIĄZANIA W SĄSIEDZTWIE ROZWIĄZANIA BIEŻĄCEGO

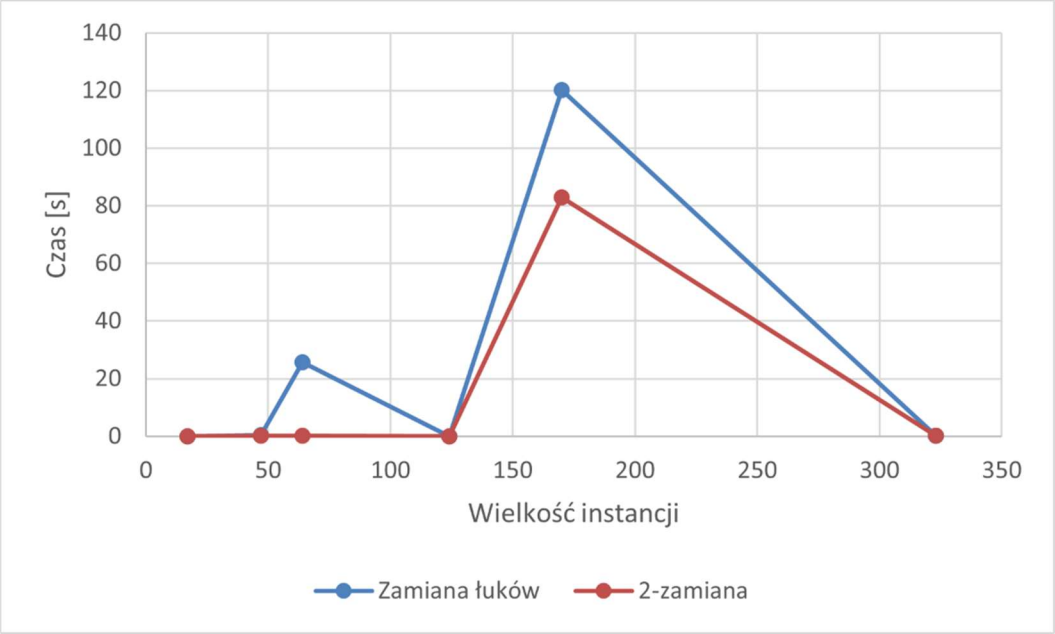
Założenia wstępne:

- Temperatura początkowa -20
- Schemat chłodzenia – *logarytmiczny*
- Sposób wyboru rozwiązania początkowego – *algorytm greedy*
- Długość epoki – *30% wielkości sąsiedztwa*

Został dodany kolejny warunek końca wykonywania algorytmu: czas = 120s.

Badanie wpływu wyboru rozwiązania w sąsiedztwie rozwiązania bieżącego na:

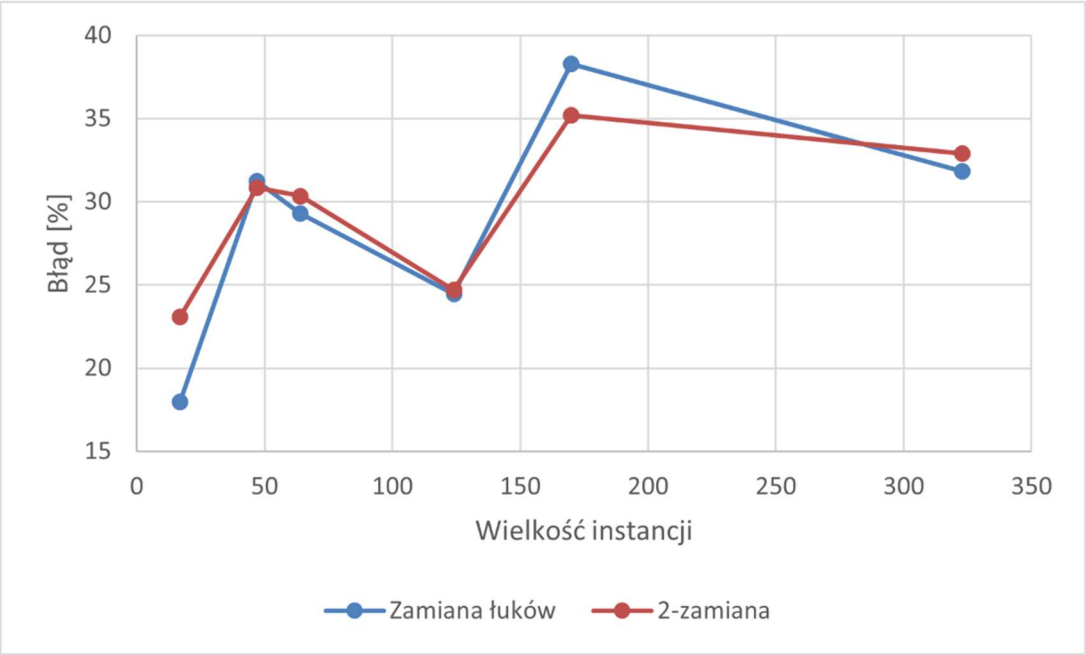
- czas uzyskiwania rozwiązań (dla błędu 35%*):



Rysunek 9: Wykres zależności czasu wykonywania algorytmu od wielkości instancji dla zadanego sposobu wyboru rozwiązania.

**Dla instancji ftv170.atsp średni błąd w zadanym czasie wynosił 38,29% - dla zamiany łuków i 35,21% - dla 2-zamiany.*

– jakość uzyskiwanych rozwiązań:



Rysunek 10: Wykres zależności wielkości błędu uzyskanego rozwiązania od wielkości instancji dla zadanego sposobu wyboru rozwiązania.

Z uzyskanych wyników (*Rysunek 9*) widać, że algorytm wykonuje się szybciej przy zastosowaniu 2-zamiany. Ponieważ obydwie metody wykorzystują losowość, a różnica w błędzie przy zastosowaniu badanych metod jest niewielka (*Rysunek 10*), bardziej korzystną wydaje się być 2-zamiana.

❖ WYBÓR DŁUGOŚCI EPOKI

Założenia wstępne:

- Temperatura początkowa – 20
- Schemat chłodzenia – *logarytmiczny*
- Sposób wyboru rozwiązania w sąsiedztwie rozwiązania bieżącego – *2-zamiana*
- Sposób wyboru rozwiązania początkowego – *algorytm greedy*

Długość epoki była wyznaczana jako % wielkości sąsiedztwa:

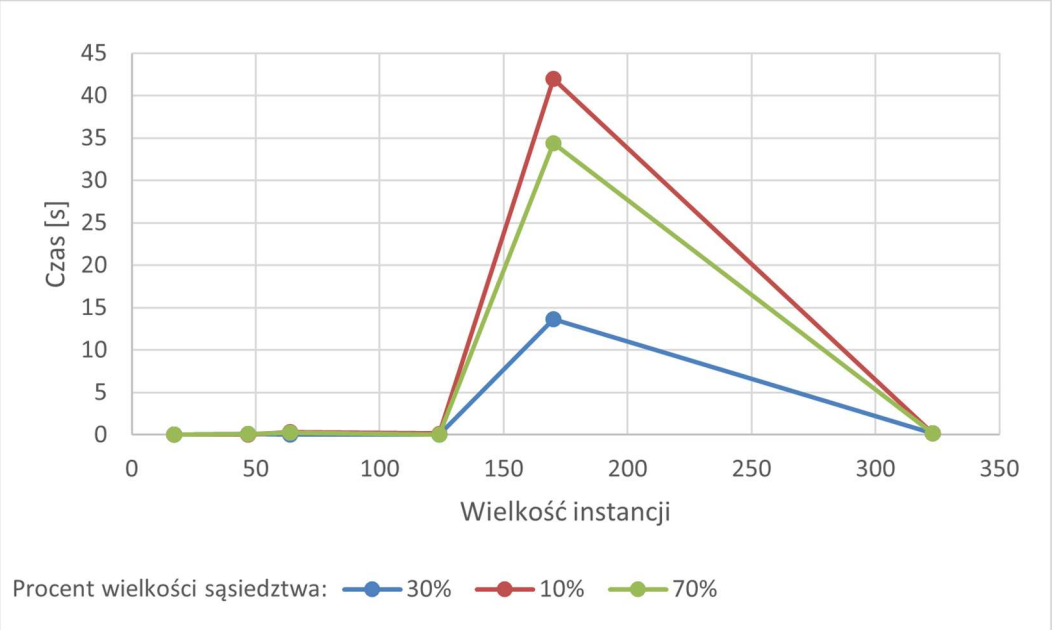
$$L = x * s$$

x – wartość [%], *s – wielkość sąsiedztwa*

Został dodany kolejny warunek końca wykonywania algorytmu: czas = 120s.

Badanie wpływu długości epoki na:

- czas uzyskiwania rozwiązań (dla błędu 35%*):

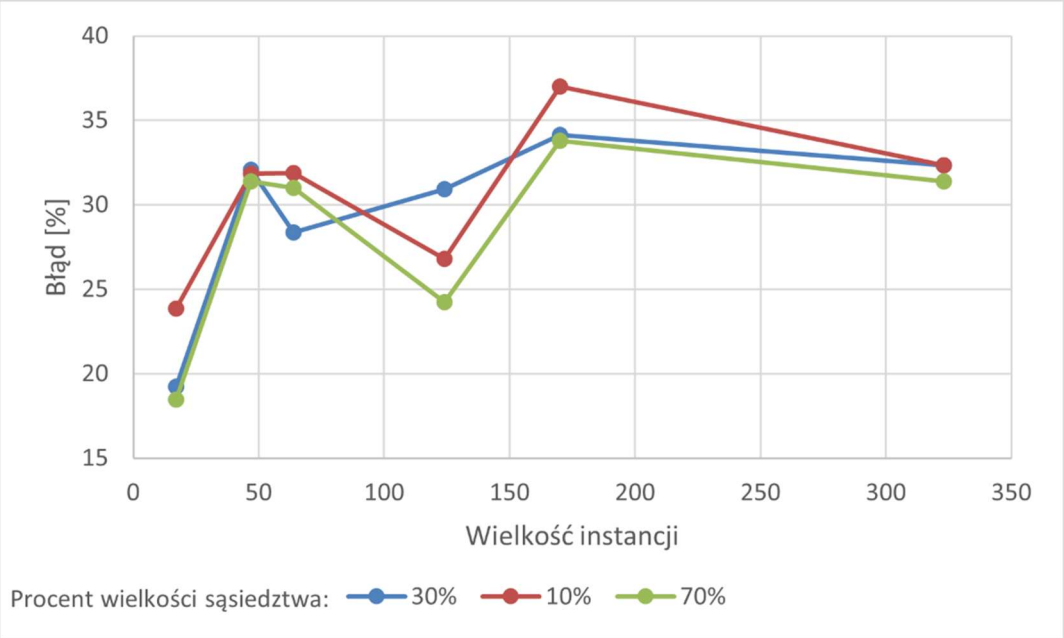


Rysunek 11: Wykres zależności czasu wykonywania algorytmu od wielkości instancji dla zadanej długości epoki, określonej jako procent sąsiedztwa.

Z wykresu (*Rysunek 11*) wynika, że algorytm wykonywał się najkrócej dla długości epoki równej 30% wielkości sąsiedztwa. Dłuższy czas wykonywania dla wartości 70% może wynikać z powtarzalności losowanych rozwiązań (algorytm losuje rozwiązanie, które już sprawdzał). W przypadku 30% może być to spowodowane niewystarczającą eksploracją sąsiedztwa.

**Dla instancji ftv170.atsp średni błąd w zadanym czasie przekroczył 35%.*

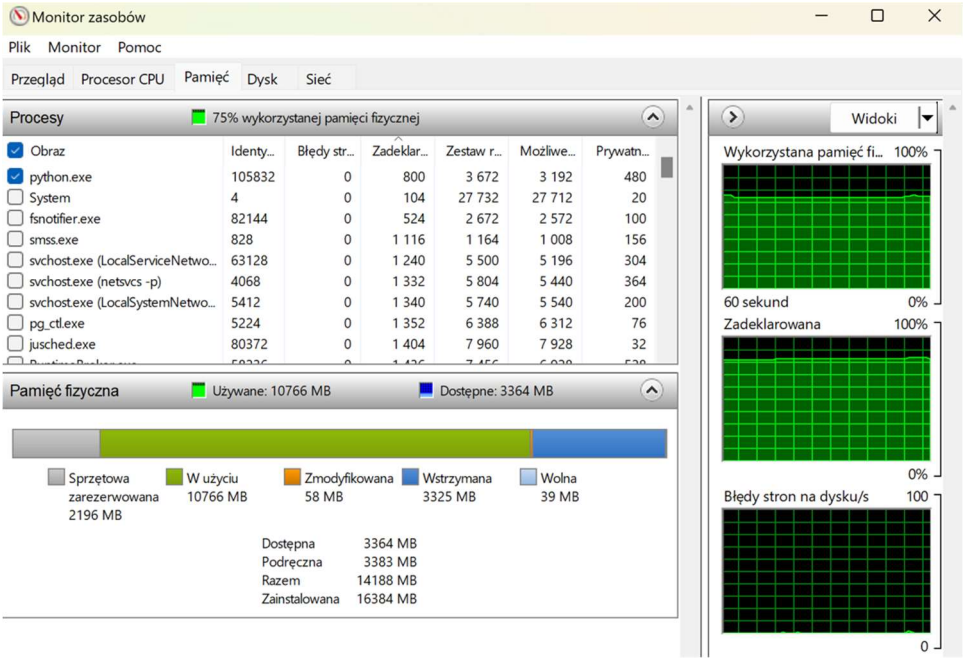
– jakość uzyskiwanych rozwiązań:



Rysunek 12: Wykres zależności wielkości błędu otrzymanego rozwiązania od wielkości instancji dla zadanej długości epoki, określonej jako procent sąsiedztwa.

Z wykresu (*Rysunek 12*) wynika, że błąd uzyskiwanego rozwiązania maleje wraz z wydłużeniem epoki.

Wykorzystanie pamięci komputera było monitorowana na bieżąco przy pomocy systemowego *Monitora zasobów* (*Rysunek 13*). Różnice w zużyciu pamięci w trakcie wykonywania algorytmu dla badanych parametrów wahały się w granicach 2MB.



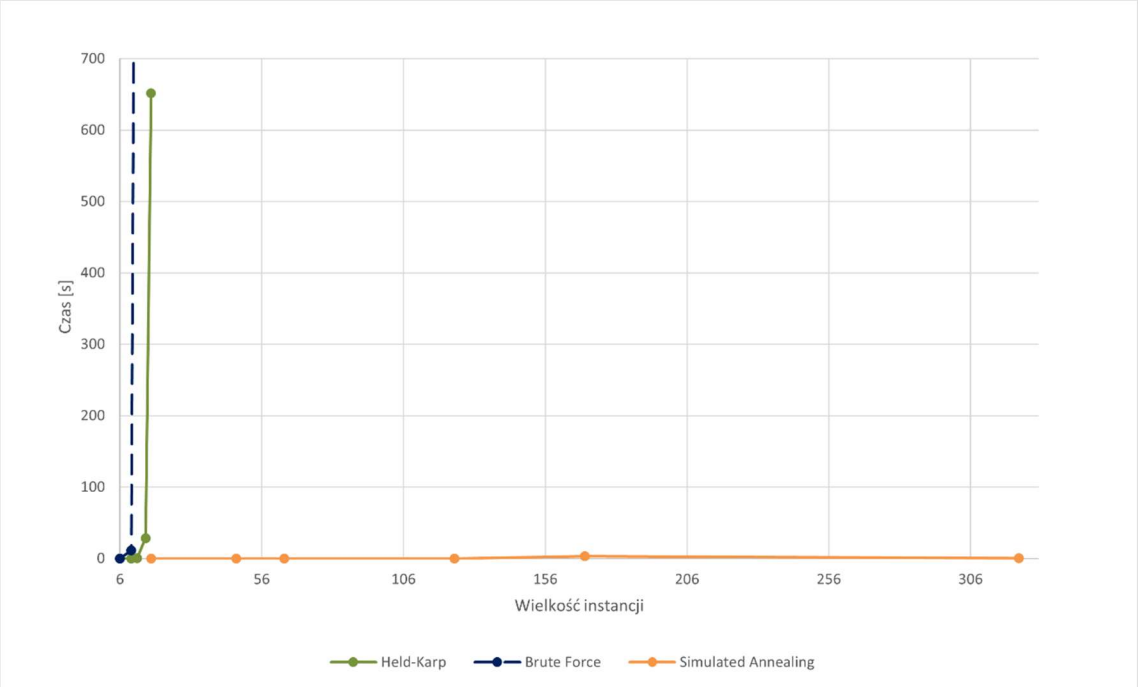
Rysunek 13: Zrzut ekranu przedstawiający wykorzystaną metodę pomiaru wykorzystywanej pamięci.

7. Analiza wyników i wnioski

Działanie algorytmu SA opiera się na doborze parametrów badanych w tym zadaniu. Aby zapewnić jego jak najlepsze działanie (szybkie i dokładne) należy znać specyfikę badanej instancji. Dobrze obrazują to wyniki uzyskane dla badanej instancji *ftv170.atsp*. Trzeba wziąć również pod uwagę, że wspomniane parametry wpływają na siebie nawzajem podczas działania algorytmu. Przykładowo, temperatura początkowa i wybór schematu chłodzenia muszą pozwolić na uzyskanie rozwiązania przed obniżeniem temperatury do wartości minimalnej.

Porównanie algorytmów:

- Brute Force
- Helda-Karpa
- Symulowanego wyżarzania (z błędem < 40%)



Rysunek 14: Porównanie czasów wykonywania algorytmów rozwiązujących problem komiwojażera.

Simulated Annealing nie jest algorytmem dokładnym, jednak pozwala znaleźć satysfakcjonujące rozwiązanie problemu w rozsądnym czasie (nawet dla dużych instancji – największą przetestowaną w tym zadaniu była *rbg323.atsp*), na co nie pozwalają algorytmy takie jak *Brute Force* czy *algorytm Helda-Karpa*.