Projektowanie Efektywnych Algorytmów
Algorytm Mrówkowy

Jakub Klawon

25 Stycznia 2024

Spis treści

1 Wstęp teoretyczny

Tematem projektu jest przygotowanie programu do badania efektywności algorytmu mrówkowego (ang. Ant Colony Optimization) w rozwiązywaniu problemu komiwojażera (ang. traveling salesman problem). Implementacja zakłada typ cykliczny, który aktualizuje liczbę feromonów po każdej iteracji. Algorytm przyjmuje na wstępie kilka parametrów: zmienna α, która wpływa na prawdopodbieństwo wyboru następnego miasta kierując się ilością feromonu na danej trasie, zmienna β, która wpływa na to prawdopodobieństwo kierując się odległością między miastami, szybkość wyparowywania feromonu oraz ilość feromonu zostawianego przez mrówkę. Teoretyczna złożoność czasowa algorytmu to $O(n^3)$.

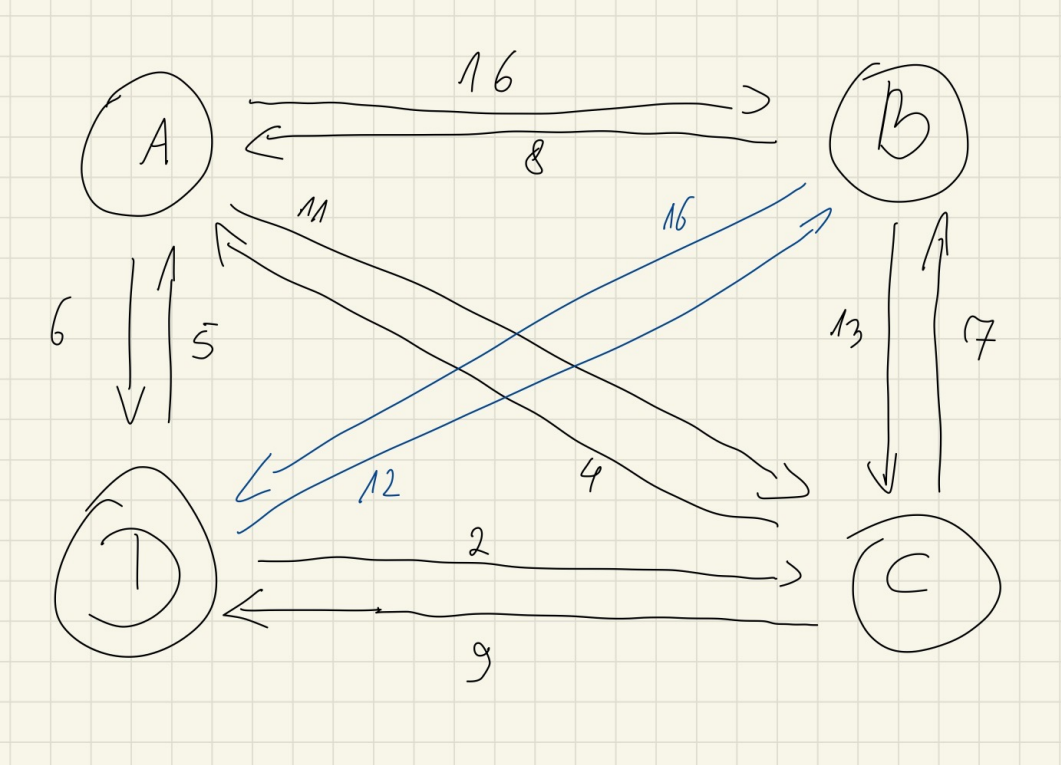
2 Opis implementacji algorytmu

1. Algorytm przyjmuje jako parametry: zmienna alfa i beta, współczynnik szybkości parowania feromonów, ilość feromonów zostawianych przez mrówkę na trasie.
2. Początkowa liczba feromonów obliczana jest na podstawie próbkowania 10 tys. różnic między sąsiadami.
3. Do przechowywania tras wykorzystywane są tablice dynamiczne, ze względu na potrzebę ciągłego dostępu do ich zawartości.
4. Do generowania liczb losowych, zastosowana została klasa uniform_real_distribution zarówno dla typu integer jak i typu double.
5. Populacja mrówek jest taka sama jak liczba wierzchołków. W każdej iteracji n mrówek szuka potencjalnej trasy z każdego wierzchołku po kolei.
6. Feromony na danej trasie aktualizowane są po każdej iteracji. Jeżeli trasa nie zmieni się na lepszą przez 10 następnych iteracji, algorytm kończy pracę.

3 Przykład Praktyczny

3.1 Dane

$$\left|\begin{array}{cccc} 0 & 16 & 11 & 6 \\ 8 & 0 & 13 & 16 \\ 4 & 7 & 0 & 9 \\ 5 & 12 & 2 & 0 \end{array}\right|$$



Rysunek 1: Graficzne przedstawienie danych

3.2 Opis

1. Algorytm rozpoczyna działanie próbkowaniem 10000 losowych instancji i porównuje różnicę między nimi. Na tej podstawie oblicza średnią, przez którą dzieli liczbę wierzchołków. Tak zostaje policzona początkowa ilość feromonów, która w tym przykładzie wynosi 0.525.

2. Algorytm przypisuje wartości $\alpha = 5.3$, $\beta = 14.0$, współczynnik wyparowywania feromonu = 0.5, ilość zostawianego feromonu = 100.0.

3. Wyliczana jest chęć skorzystania z danej trasy, na podstawie wzoru $\delta[i][j] = feromon[i][j] * 1000 * \frac{1}{graf[i][j]}$.

4. Algorytm rozpoczyna pracę od wierzchołka A. Dodaje go na koniec aktualnie badanej ścieżki i zaznacza jako odwiedzony w tablicy wartości boolowskich.

5. Algorytm tworzy mianownik z nieodwiedzonych miast używając wzoru ... i zlicza ile miast zostało nieodwiedzonych.

6. Dla każdego nieodwiedzonego miasta, zostaje policzone prawdopodobieństwo jego wyboru na podstawie wzoru ...

7. Algorytm tworzy sumę częściową z prawdopodobieństw i generuje losową liczbę zmiennoprzecinkową z zakresu 0 do 1.0. Następnie szuka pierwszego miasta, od którego prawdopodobieństwa wyboru będzie ona mniejsza.

8. Algorytm wybiera wierzchołek D, zaznacza go jako odwiedzony i dodaje do ścieżki. Następnie powtarza punkty 5-7 dla kolejnych wartości.

9. Utworzona zostaje ścieżka A -> D -> C -> B -> A, dla której koszt wynosi 23. Jest to najkrótsza znaleziona ścieżka, więc algorytm zapisuje ją jako najlepszy wynik i wykonuje punkty 4 - 8 zaczynając od kolejnych trzech wierzchołków. Dla każdej iteracji, znalezionych zostaje n=4 tras.

10. Po każdej znalezionej trasie, zostaje obliczona ilość feromonu do dodania na danej odległości korzystając ze wzoru $dodaj_f eromony[i][i+1] = \frac{ilosczostawianego feromonu}{kosztsciezki}$

11. Po znalezieniu 4 ścieżek, aktualizowana jest ilość feromonu na danych odległościach, korzystając ze wzoru $feromon[i][j] = feromon[i][j] + dodaj_f eromony[i][j]$.

12. Algorytm wykonuje się jeszcze 10 razy, gdyż wynik 23 jest najlepszym możliwym w tym grafie.

13. Algorytm kończy pracę, wskazując wynik 23 jako najlepszy z trasą A -> D -> C -> B -> A.

4 Plan eksperymentu

Eksperyment składa się z 4 części:

1. Pomiary czasu dla losowo generowanych przykładów - program generuje 15 różnych losowych instancji dla przedziału 10,20, ... ,190 elementów. Dla każdej liczby miast przelicza średnią z wyników. Zakres liczb znajdujących się w grafach to od 1 do 2000. Eksperyment prowadzony jest na $\alpha = 5.3$, $\beta = 14.0$, $evaporation\_rate = 0.5$, $pheromones = 100.0$.

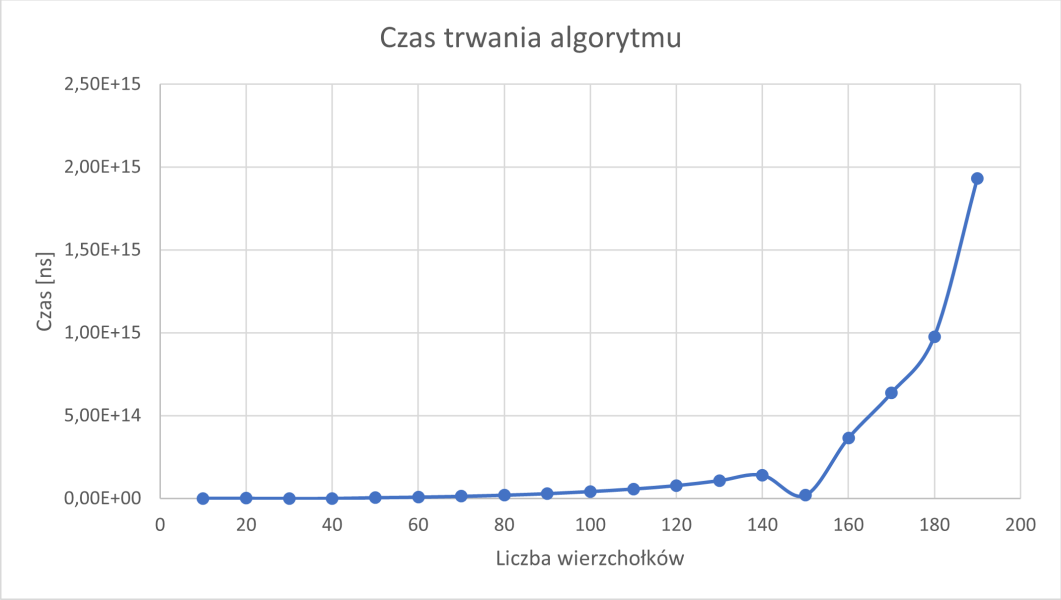
2. Wyniki poziomu błędu dla konkretnych przykładów - program działa na 15 plikach do 170 miast, pobranych ze strony Uniwersytetu w Heidelbergu, dla których znane są optymalne wyniki. Korzystając ze wzoru $\frac{wynik-optymalny}{optymalny} * 100$ przekazuje błąd wyrażony w procentach. Dla każdego zestawu wykonuje 15 pomiarów, z których wylicza średnią. Pomiary wykonywane będą dla różnych parametrów, które będą przedstawiane przed każdymi danymi.

5 Wyniki eksperymentu

5.1 Pomiary czasu dla losowo generowanych przykładów

Liczba wierzchołków	Czas [ns]
10	6,58E+11
20	8,93E+11
30	1,55E+10
40	2,76E+11
50	4,88E+12
60	8,20E+12
70	1,30E+13
80	2,00E+13
90	2,93E+13
100	4,19E+13
110	5,74E+13
120	7,77E+13
130	1,08E+14
140	1,41E+14
150	1,90E+13
160	3,63E+14
170	6,36E+14
180	9,77E+14
190	1,93E+15

Tabela 1: Pomiary czasu dla $\alpha = 5.3$, $\beta = 14.0$ itd.



Rysunek 2: Wykres czasu od liczby wierzchołków

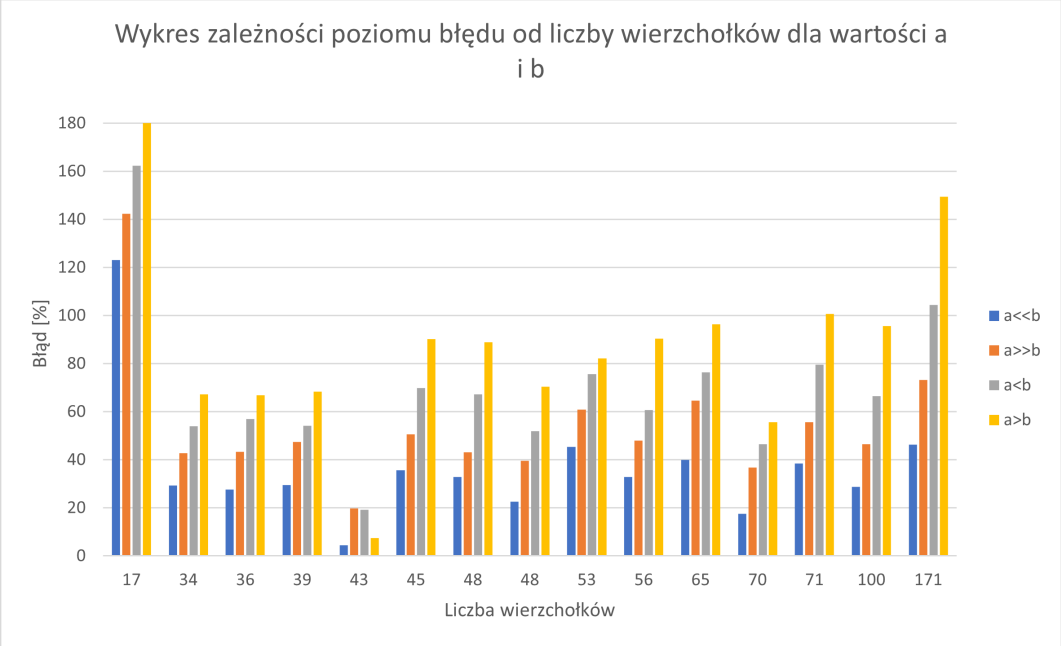
5.2 Wyniki poziomu błędu dla konkretnych przykładów

Liczba wierzchołków	Błąd $\alpha << \beta$ [%]	Błąd $\alpha >> \beta$ [%]	Błąd $\alpha < \beta$ [%]	Błąd $\alpha > \beta$
17	123,077	142,222	162,222	180
34	29,3209	42,7372	53,8932	67,2369
36	27,6805	43,2994	56,9269	66,9201
39	29,4118	47,3551	54,1786	68,2832
43	4,51601	19,7817	19,2278	7,36892
45	35,5528	50,5063	69,7747	90,1178
48	32,8529	43,1794	67,1922	88,8964
48	22,4583	39,5969	51,9655	70,3721
53	45,3903	60,8351	75,5781	82,099
56	32,7695	47,9892	60,597	90,3275
65	39,8659	64,5641	76,4292	96,4183
70	17,4933	36,7043	46,3998	55,5931
71	38,506	55,6308	79,4496	100,694
100	28,7254	46,4852	66,5287	95,6333
171	46,323	73,1543	104,469	149,423

Tabela 2: Pomiary błędu dla danych wartości α i β, opisanych poniżej.

$\alpha << \beta$	$\alpha >> \beta$	$\alpha < \beta$	$\alpha > \beta$
5.3 i 14.0	14.0 i 5.3	3.2 i 4.0	4.0 i 3.2

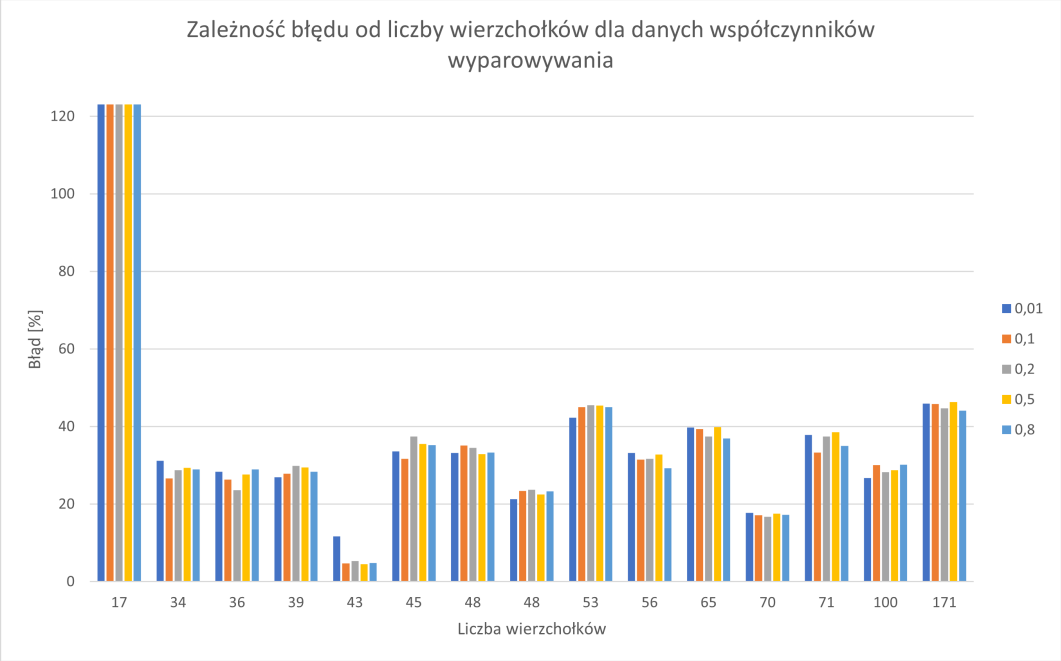
Tabela 3: Wartości α i β



Rysunek 3: Wykres zależności poziomu błędu od liczby wierzchołków dla danych współczynników α i β.

Liczba wierzchołków	Błąd $evap = 0.01$ [%]	Błąd 0.1 [%]	Błąd 0.2 [%]	Błąd 0.5 [%]	Błąd 0.8 [%]
17	123,077	123,077	123,077	123,077	123,077
34	31,1405	26,6511	28,7921	29,3209	28,9684
36	28,391	26,377	23,5574	27,6805	28,9749
39	26,9804	27,8606	29,8693	29,4118	28,3355
43	11,6572	4,74377	5,33571	4,51601	4,79359
45	33,5937	31,6801	37,3962	35,5528	35,2594
48	33,1682	35,1126	34,5045	32,8529	33,262
48	21,3054	23,4434	23,7336	22,4583	23,3417
53	42,2969	45,0456	45,5284	45,3903	45,0311
56	33,2131	31,4303	31,6459	32,7695	29,291
65	39,7245	39,3148	37,4769	39,8659	36,9114
70	17,7414	17,1655	16,7483	17,4933	17,2613
71	37,8496	33,3094	37,3915	38,506	35,0256
100	26,697	30,0311	28,2928	28,7254	30,129
171	45,9528	45,8125	44,7211	46,323	44,1186

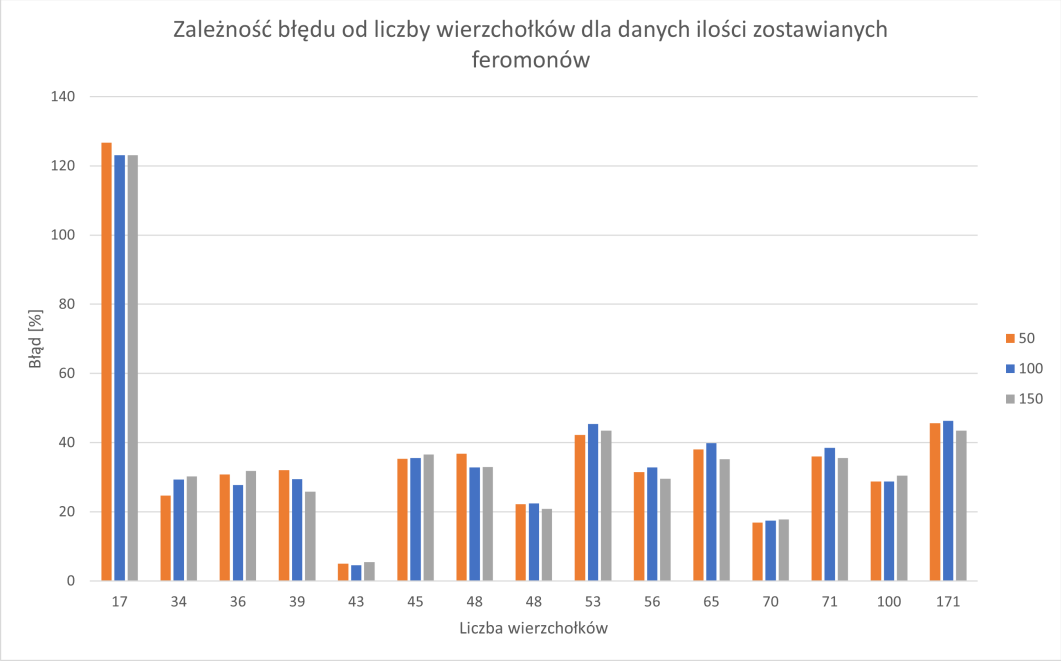
Tabela 4: Pomiary błędu dla danych współczynników wyparowywania feromonów, $\alpha = 5.3, \beta = 14.0, ph = 100.0$



Rysunek 4: Zależność błędu od liczby wierzchołków dla danych współczynników wyparowywania feromonów, $\alpha = 5.3, \beta = 14.0, ph = 100.0$

Liczba wierzchołków	Błąd $pher = 50.0$ [%]	Błąd $pher = 100.0$ [%]	Błąd $pher = 150.0$ [%]
17	126,667	123,077	123,077
34	24,7227	29,3209	30,2333
36	30,7717	27,6805	31,7809
39	31,9913	29,4118	25,8431
43	5,05101	4,51601	5,46738
45	35,28	35,5528	36,5613
48	36,7793	32,8529	32,958
48	22,1943	22,4583	20,8284
53	42,209	45,3903	43,4149
56	31,476	32,7695	29,5274
65	37,9953	39,8659	35,1677
70	16,9302	17,4933	17,7943
71	35,959	38,506	35,5385
100	28,78	28,7254	30,4403
171	45,5608	46,323	43,4531

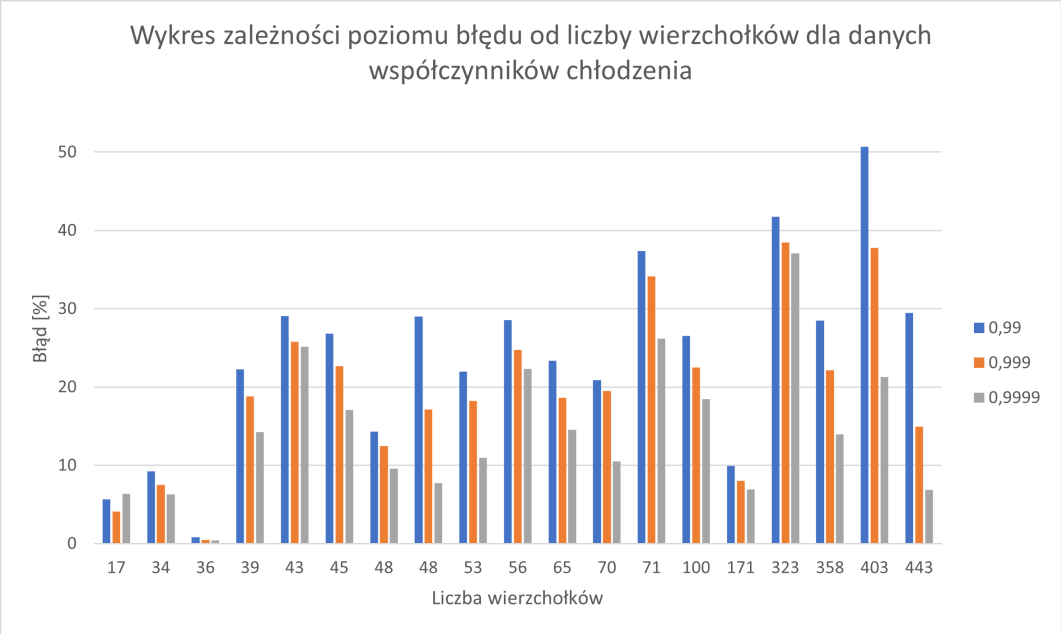
Tabela 5: Pomiary błędu dla danych ilości pozostawianych feromonów na danej trasie, $\alpha = 5.3, \beta = 14.0, evaporation\_rate = 0.5$



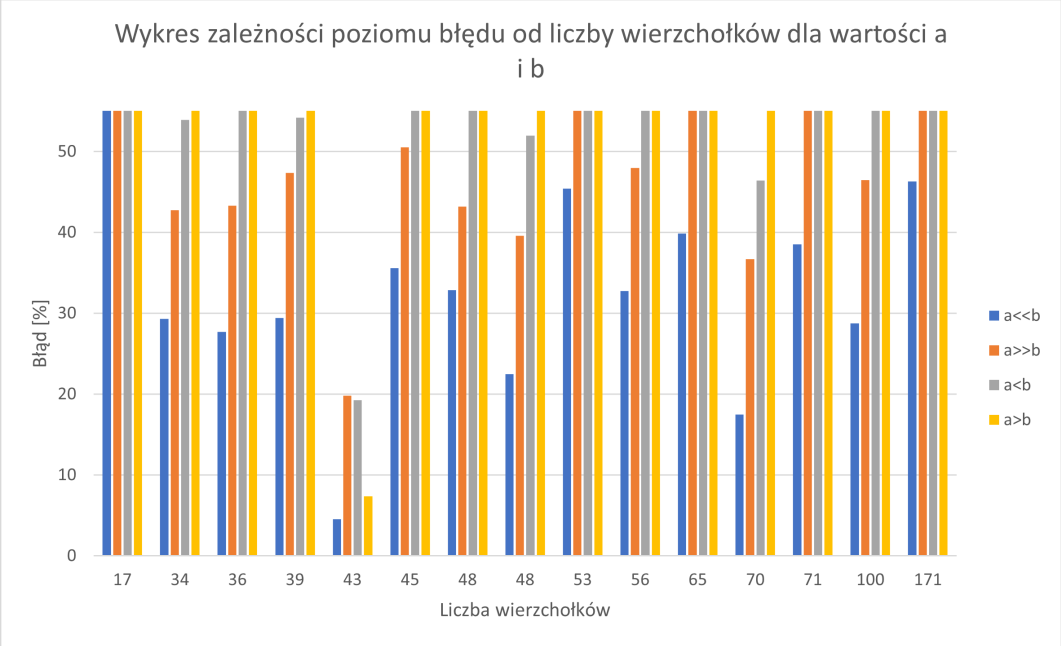
Rysunek 5: Zależność błędu od liczby wierzchołków dla danych ilości pozostawianych feromonów na danej trasie, $\alpha = 5.3, \beta = 14.0, evaporation\_rate = 0.5$

6 Porównanie z algorytmem symulowanego wyżarzania

6.1 Poziom błędu

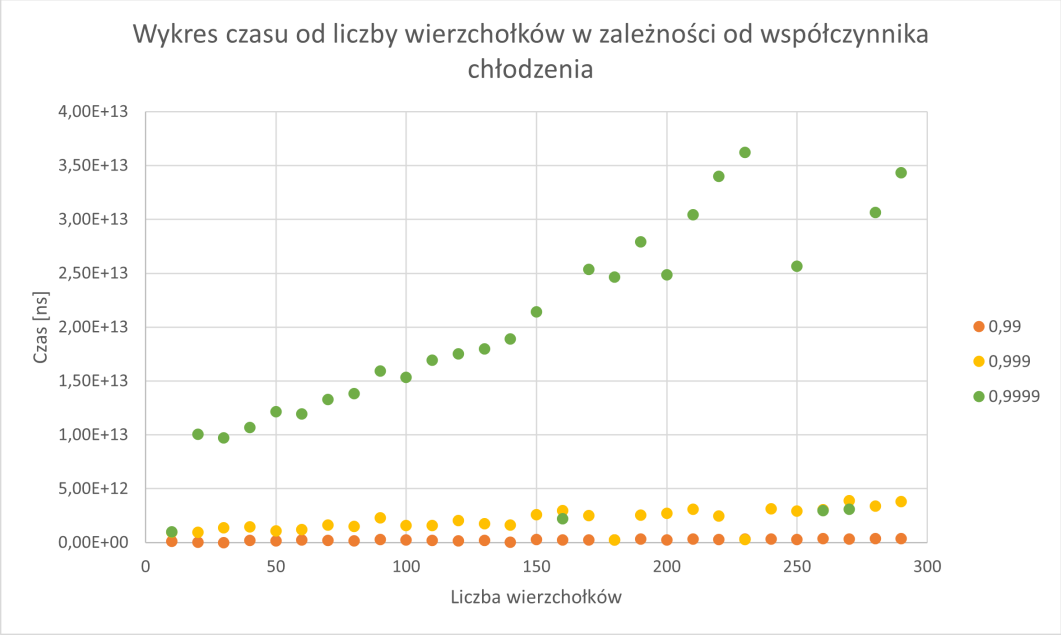


Rysunek 6: Błąd wyznaczony dla danych instancji i danych współczynników algorytmu symulowanego wyżarzania

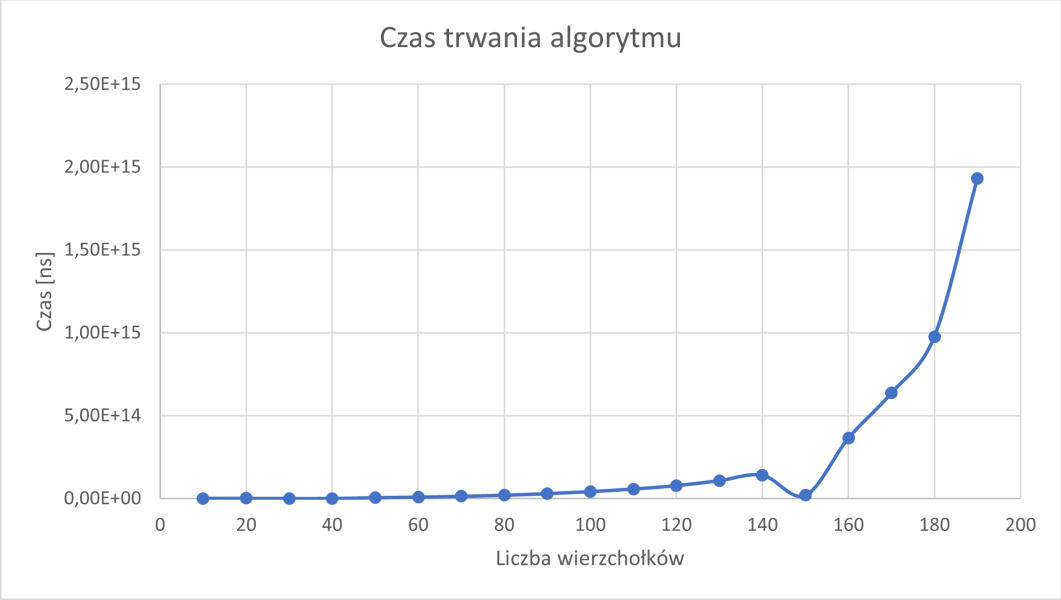


Rysunek 7: Wykres zależności poziomu błędu od liczby wierzchołków dla danych współczynników α i β. dla algorytmu mrówkowego

6.2 Czas wykonania algorytmu



Rysunek 8: Wykres czasu od liczby wierzchołków dla algorytmu symulowanego wyżarzania



Rysunek 9: Wykres czasu od liczby wierzchołków dla algorytmu mrówkowego

7 Wnioski

Czas wykonywania się algorytmu mrówkowego pokrywa się z teoretyczną złożonością $O(n^3)$. W porównaniu do algorytmu symulowanego wyżarzania, wypada on jednak gorzej w tym aspekcie. Ponadto, mimo, że algorytm kończy działanie po zaledwie 10 iteracjach bez poprawy wyniku, nie jest on w stanie poradzić sobie z liczbą wierzchołków większą niż 280. Zmiana parametrów nie wpływała znacząco na czas trwania algorytmu, więc nie prowadziłem więcej badań czasowych, niż zależność od liczby wierzchołków.

Dokładność algorytmu wydaje się niezależna od liczby wierzchołków. Najbardziej dokładne wyniki otrzymywałem od parametrów $\alpha = 5.3$, $\beta = 14.0$, współczynnik wyparowywania feromonów $= 0.1$ i ilość zostawianych feromonów na danej trasie $= 100.0$. W porównaniu do algorytmu symulowanego wyżarzania, mrówkowy osiąga w większości przypadków gorsze wyniki. Różnica między nimi jest na ogół niewielka, poza instancją 17 wierzchołków.

W celu osiągania lepszych wyników, próbowałem rozpoczynać działanie algorytmu od wyniku zachłannego. O ile w bardzo małych instancjach (<30) dawało to nieznacznie lepsze wyniki, tak jednak w tych większych sprawiało, że algorytm zbyt szybko poddawał się z poszukiwaniem lepszych rozwiązań, lub robił to na tyle wolno, że dawał kilku procentowo gorsze wyniki.