

# Projektowanie efektywnych algorytmów

## Sprawozdanie

### Zadanie projektowe nr 2

Metoda Tabu Search dla problemu komiwojażera

#### 1. Wstęp

Celem projektu jest zastosowanie 3 algorytmów dla wcześniej wybranego zagadnienia, w moim przypadku problemu komiwojażera a następnie porównanie wyników otrzymanych przy użyciu każdego z nich. W drugim zadaniu projektowym zbadane zostało działanie metody Tabu Search na podstawie napisanego algorytmu, badając go w zależności od najbardziej istotnych z dostępnych parametrów dla wcześniej wybranych instancji testowych. Uzyskane wyniki zostaną potem porównane ze sobą pod względem dokładności otrzymanego wyniku, oraz jak się one mają do wcześniej przeprowadzonych testów dla metody podziału i ograniczeń.

#### 2. Metoda Tabu Search

**Tabu Search** (przeszukiwanie tabu) to metahuerystyka używana do rozwiązywania problemów optymalizacyjnych badając sąsiedztwo rozwiązania znajdującego się w przestrzeni rozwiązań problemu oraz zapamiętując wykonywane ostatnio ruchy. Podstawą tej metody jest zapamiętywanie ruchów na **liście tabu** blokuje jego wykonywanie w pewnej ilości następnych kroków w zależności od przyznanej im **kadencji**. Zapisanie ruchów niedozwolonych w tabu pozwala na odrzucenie rozwiązań niedawno sprawdzane zwiększać obszar przeszukiwania, co skutkuje większą możliwością wyjścia z minimum lokalnego, ale kosztem dokładności algorytmu.

Podczas działania może się jednak okazać że właśnie zakazany został bardzo dobry ruch przez co nie można go wykonać. Aby zapobiec takim sytuacjom używamy **kryterium aspiracji**, które sprawdza elementy w tabu pod kątem opłacalności, i w przypadku odpowiednio wysokiej poprawy rozwiązania łamie tabu używając rozwiązania zakazanego. W ten sposób nie blokujemy rozwiązań bardzo dobrych, co może poprawić uzyskiwane wyniki.

W przypadku braku poprawienia wyniku przez dłuższy czas uruchamiana jest **dywersyfikacja**, czyli metoda resetująca aktualne rozwiązanie algorytmu. Wylosowanie nowego aktualnego rozwiązania pomoże wydostać się z minimum lokalnego, w którym może utknąć algorytm.<sup>1</sup>

### 3. Implementacja algorytmu

Algorytm został napisany na podstawie poniższego pseudokodu:

```
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
     $\forall \text{element} \in \text{tabu\_list}$  do
        -- kadencjai
        if kadencjai = 0 then
            usuń element(trybut, kadencjai) z tabu_list
    if CriticalEvents() then
         $x_0 \leftarrow \text{Restart()} \text{ (Dywersyfikacja)}$ 
        if  $f(x_0) > f(x_{opt})$  then
             $x_{opt} \leftarrow x_0$ 
    until warunek zakończenia
```

Algorytm rozpoczyna działanie losując rozwiązanie początkowe które będzie podstawą do działania przeszukiwań tabu. Od tego momentu wykonane przeszukiwanie lokalne z ograniczeniem tabu, dopóki czas działania nie przekroczy tego podanego w konfiguracji.

Dla aktualnej przeglądanej ścieżki wybierane są **dwie krawędzie**, z których jedna jest wybierana w taki sposób, aby pozbyć się najdłuższej krawędzi w algorytmie (pierwszy wierzchołek najdłużej krawędzi oraz wierzchołek poprzedzający go), natomiast druga krawędź brana jest całkowicie losowo (z pominięciem wcześniej wybranych wierzchołków) dając w ten sposób sprawiedliwy wybór ruchu z otoczenia lokalnego.

Wybrane w ten sposób krawędzie są następnie sprawdzane, czy znajdują się na **liście tabu** i czy mogą zostać użyte do sprawdzania następnego ruchu. Jeśli danego ruchu nie ma w tabu, ruch ten zostaje poddany sprawdzeniu czy poprawia aktualne rozwiązanie, czy też nie.

W przypadku gdy znajduje się w tabu, może on wciąż zostać wykorzystany, pod warunkiem że przejdzie on przez **kryterium aspiracji**, które pozwoli na wykorzystanie ruchów obiecujących. Kryterium to jednak bada tylko te krawędzie, których kadencja znajduje się w przedziale od  $2/3$  , do  $2/3$  aktualnie ustawionej kadencji. Gdy ten warunek zostanie spełniony, badany zostaje jeszcze przyrost jaki daje nam algorytm. Jeśli długość drogi jest krótsza niż  $1/30$  długości aktualnie badanej ścieżki podzielonej, to kryterium aspiracji zostaje spełnione i ruch może zostać sprawdzony.

---

<sup>1</sup> Na podstawie opracowania: [www.ii.uni.wroc.pl/~prz/2011lato/ah/opracowania/szuk\\_tabu.opr.pdf](http://www.ii.uni.wroc.pl/~prz/2011lato/ah/opracowania/szuk_tabu.opr.pdf)

Gdy już wiemy że dany ruch jest dozwolony, wykonujemy go zamieniając wylosowane krawędzie otrzymując nową drogę z otoczenia poprzedniego rozwiązania. Jeżeli długość tej drogi jest mniejsza od aktualnie badanej, zostaje ona aktualnie badaną drogą z nowym otoczeniem. Analogicznie gdy droga po zamianie jest lepsza od najlepszego rozwiązania, najlepsze rozwiązanie zostanie przez nią zastąpione.

Po zbadaniu rozwiązania, nie zależnie od tego czy jest ono lepsze czy nie, **para krawędzi dodawana jest do tabu** (o ile jest miejsce) zakazując wykonywania ruchów przez podaną w konfiguracji **kadencje**. Gdy nowa droga zostanie dodana do tabu, następuje zmniejszenie dywersyfikacji dla pozostałych elementów w liście tabu, oraz usunięcie ich z tablicy w przypadku gdy wartość dywersyfikacji dla danej pary krawędzi będzie równa 0.

Ostatnim elementem algorytmu jest **dywersyfikacja**. Występuje ona, gdy przez podaną w konfiguracji ilość iteracji algorytm nie znajdzie żadnego ruchu polepszającego rozwiązanie. W takim przypadku algorytm jest resetowany, losując nowe aktualne rozwiązanie oraz czyszcząc listę zakazanych przejść, dając nam szansę przeglądania innej części zbioru rozwiązań wychodząc z minimum lokalnego, zwiększając tym samym prawdopodobieństwo znalezienia minimum globalnego w badanej instancji. Jedyne elementy zapamiętane podczas dywersyfikacji to aktualne najlepsze rozwiązanie, oraz czas trwania algorytmu.

Czas działania algorytmu jest przede wszystkim uzależniony od czasu ograniczającego podanego w konfiguracji. Z kolei jego główna część wewnętrz pętli ograniczonej czasowo opiera się co najwyżej na pojedynczych zapętlach o maksymalnej wielkości równej rozmiarowi instancji lub rozmiarowi tabu (w przypadku gdy ta wartość jest większa) dając nam **złożoność obliczeniową  $O(n)$** .

#### 4. Procedura testowania

Pomiary zostały przeprowadzone poprzez wykonanie 10 prób dla każdej testowanej instancji w każdej konfiguracji. Wszystkie testy przeprowadzone zostały z czasem działania algorytmu równemu 30 sekund.

Testy przygotowanych zestawów zostały wykonane na prywatnym komputerze. Podczas przeprowadzania testów nie były uruchomione żadne dodatkowe aplikacje, a wszelkie procesy w tle zostały ograniczone do minimum poprzez wyłączenie aplikacji w tle oraz odłączeniu komputera od Internetu (zachowanie jedynie procesów wymaganych przez system operacyjny) w celu uzyskania jak najlepszych wyników. Konfiguracja sprzętowa maszyny testującej wygląda następująco:

**Procesor:** Intel Core i7-4700MQ 2.4 GHz

**Pamięć RAM:** 16 GB

**System operacyjny:** Windows 10 Pro N

Jako podstawy testowej użyłem zgodnie z poleceniem plików z podanego źródła<sup>2</sup>. Do testowania użyłem 6 różnych instancji problemu, 3 symetryczne oraz 3 asymetryczne:

**gr17.tsp, pa561.tsp, si1032.tsp, br17.atsp, ftv170.atsp i rbg443.atsp**

---

<sup>2</sup> <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

Każdy z plików posiada inny zestaw wierzchołków, w różnej liczbie oraz o różnej odległości względem siebie. Również do każdego z nich dołączona jest znaleziona optymalna ścieżka, co pozwoli nam porównać otrzymane wyniki podczas testów z najlepszymi znalezionymi do tej pory, dzięki czemu będziemy mogli określić czy dany algorytm działa prawidłowo. Wczytywanie plików **.tsp** i **.atsp** ograniczyłem jedynie do obsługi wybranych plików, jednak program powinien wczytać dowolne dane z wcześniej podanego źródła zapisane jako macierz zwykła bądź diagonalna.

Podczas badań sprawdzone zostały różnice pomiędzy następującymi parametrami w algorytmie: **długość listy tabu** oraz **długość kadencji**. W obydwu przypadkach zostaną przeprowadzone testy z użyciem **funkcji aspiracji**, jak i bez niej. Wartości dla badanych parametrów są następujące (gdzie **n** to rozmiar instancji):

Badanie wielkości listy tabu: **7, 10, n/4, n/2, n, 2n**

Badanie długości kadencji: **n/6, n/4, n/2, n, 2n**

Wykonane testy opierają się na domyślnym ustawieniu parametrów, oczywiście poza zmianami aktualnie testowanych parametrów możliwych do edycji w pliku konfiguracyjnym *config.txt* dołączonym do projektu. Parametry domyślne prezentują się one następująco:

**Czas:** 30 sekund

**Rozmiar tabu:**  $n/2$

**Kadencja:**  $n/4$

**Iteracje krytyczne<sup>3</sup>:** 1000

Wyniki pomiarów przedstawiają średnią wartość uzyskaną dla badanego parametru, oraz najlepszą drogę jaką udało się uzyskać. Aby dowiedzieć się jak bardzo wygenerowane rozwiązania różnią się od rozwiązania optymalnego, został obliczony błąd względem rozwiązania optymalnego na podstawie wzoru:  $B = (Droga(N) - Droga(O)) * 100 / Droga(O)$ , gdzie **Droga(O)** to optymalna droga dla danej instancji, a **Droga(N)** to uzyskana wartość w testach. Błąd ten zapisany jest w procentach.

Wyniki optymalne dla poszczególnych instancji:

**gr17.tsp:** 2085

**pa561.tsp:** 2763

**si1032.tsp:** 92650

**br17.atsp:** 39

**ftv170.atsp:** 2755

**rbg443.atsp:** 2720

---

<sup>3</sup> Ilość iteracji po których następuje dywersyfikacja algorytmu.

## 5. Wyniki

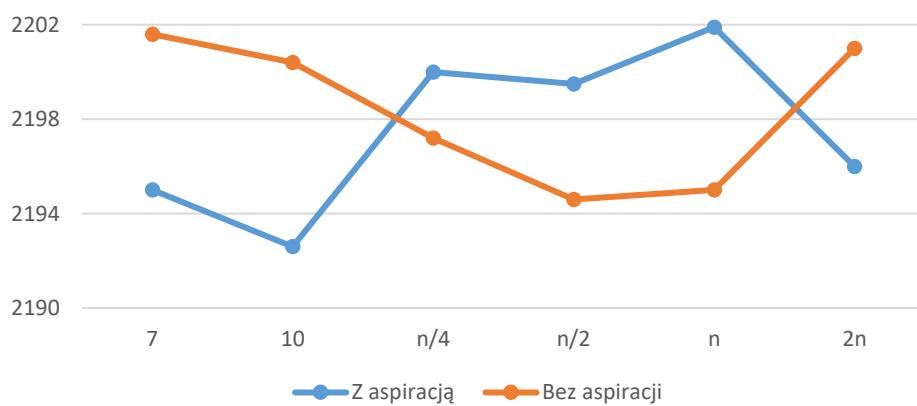
### 5.1. Badanie wielkości listy tabu

**gr17.tsp**

	Rozmiar tabu:	7	10	n/4	n/2	n	2n
Z aspiracją	Średnia droga	2195	2192,6	2200	2199,5	2201,9	2196
	Błąd średniej drogi (%)	5,28	5,16	5,52	5,49	5,61	5,32
	Najkrótsza droga	2135	2135	2135	2135	2135	2135
	Błąd najkrótszej drogi (%)	2,40	2,40	2,40	2,40	2,40	2,40
Bez aspiracji	Średnia droga	2201,6	2200,4	2197,2	2194,6	2195	2201
	Średni błąd drogi (%)	5,59	5,53	5,38	5,26	5,28	5,56
	Najkrótsza droga	2135	2135	2135	2135	2135	2135
	Błąd najkrótszej drogi (%)	2,40	2,40	2,40	2,40	2,40	2,40

Tabela 1: Wyniki testów **gr17.tsp** dla różnego rozmiaru tabu

**br17.atsp - zmiana rozmiaru tabu  
- średni wynik**



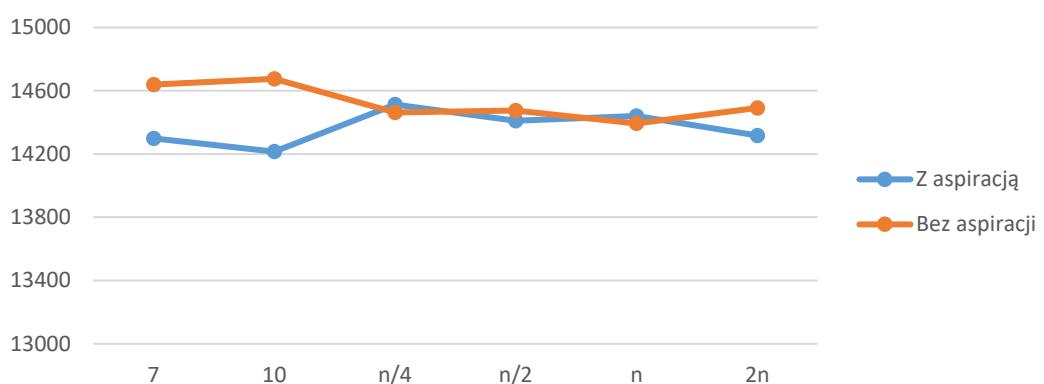
Wykres 1: Porównanie średnich wyników instancji **gr17.tsp** z i bez kryterium aspiracji

### pa561.tsp

	Rozmiar tabu:	7	10	n/4	n/2	n	2n
Z aspiracją	Średnia droga	14299,5	14214,9	14512,6	14410	14439,5	14317,9
	Błąd średniej drogi (%)	417,54	414,47	425,25	421,53	422,60	418,20
	Najkrótsza droga	13690	13440	14156	13843	13983	13390
	Błąd najkrótszej drogi (%)	395,48	386,43	412,34	401,01	406,08	384,62
<hr/>							
Bez aspiracji	Średnia droga	14638,2	14675,2	14462,3	14474,2	14392,9	14492
	Błąd średniej drogi (%)	429,79	431,13	423,43	423,86	420,92	424,50
	Najkrótsza droga	13786	14266	13377	13996	13431	13699
	Błąd najkrótszej drogi (%)	398,95	416,32	384,15	406,55	386,10	395,80

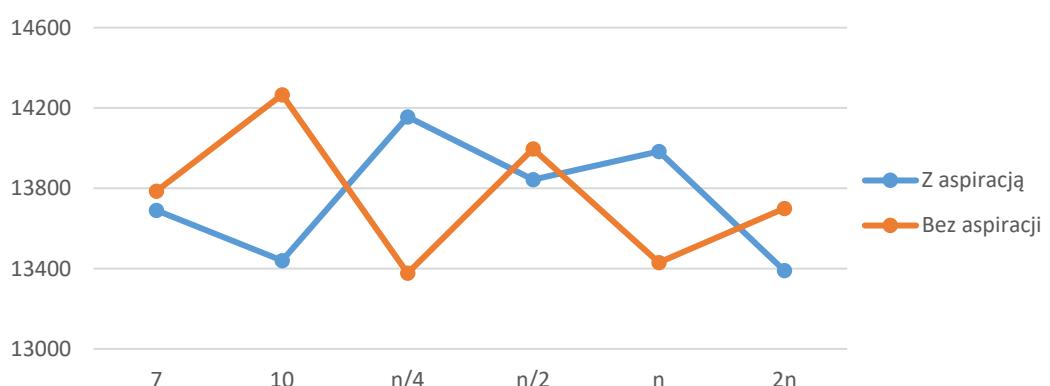
Tabela 2: Wyniki testów **pa561.tsp** dla różnego rozmiaru tabu

### pa561.tsp - zmiana rozmiaru tabu - średni wynik



Wykres 2: Porównanie średnich wyników instancji **pa561.tsp** z i bez kryterium aspiracji

### pa561.tsp - zmiana rozmiaru tabu - najlepszy wynik



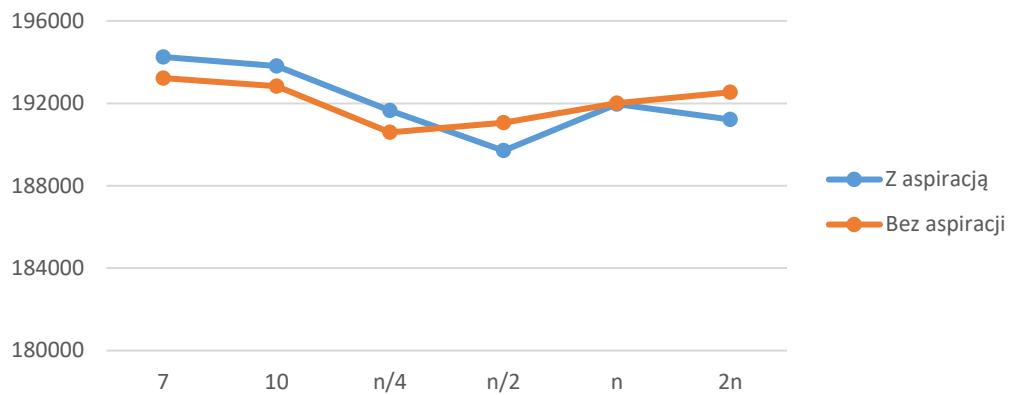
Wykres 3: Porównanie najlepszych wyników instancji **pa561.tsp** z i bez kryterium aspiracji

### si1032.tsp

	Rozmiar tabu:	7	10	n/4	n/2	n	2n
Z aspiracją	Średnia droga	194250	193808,8	191652,8	189702,1	191965,3	191210,3
	Średni błąd (%)	109,66	109,18	106,86	104,75	107,19	106,38
	Najkrótsza droga	187614	190432	183758	182990	183176	181953
	Błąd najkrótszej drogi (%)	102,50	105,54	98,34	97,51	97,71	96,39
<hr/>							
Bez aspiracji	Średnia droga	193226,4	192836,4	190590,2	191052,5	192010,4	192537
	Średni błąd (%)	108,56	108,13	105,71	106,21	107,24	107,81
	Najkrótsza droga	186497	191135	184061	187308	186578	184180
	Błąd najkrótszej drogi (%)	101,29	106,30	98,66	102,17	101,38	98,79

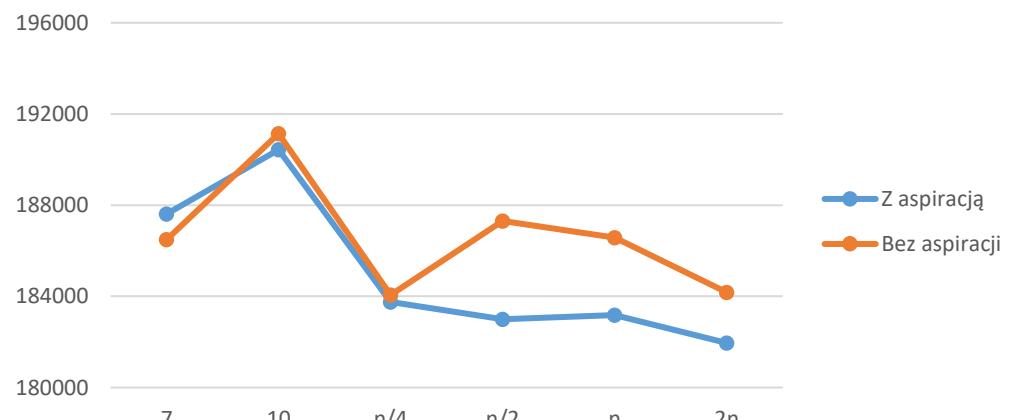
Tabela 3: Wyniki testów **si1032.tsp** dla różnego rozmiaru tabu

### si1032.tsp - zmiana rozmiaru tabu - średni wynik



Wykres 4: Porównanie średnich wyników instancji **si1032.tsp** z i bez kryterium aspiracji

### si1032.tsp - zmiana rozmiaru tabu - najlepszy wynik



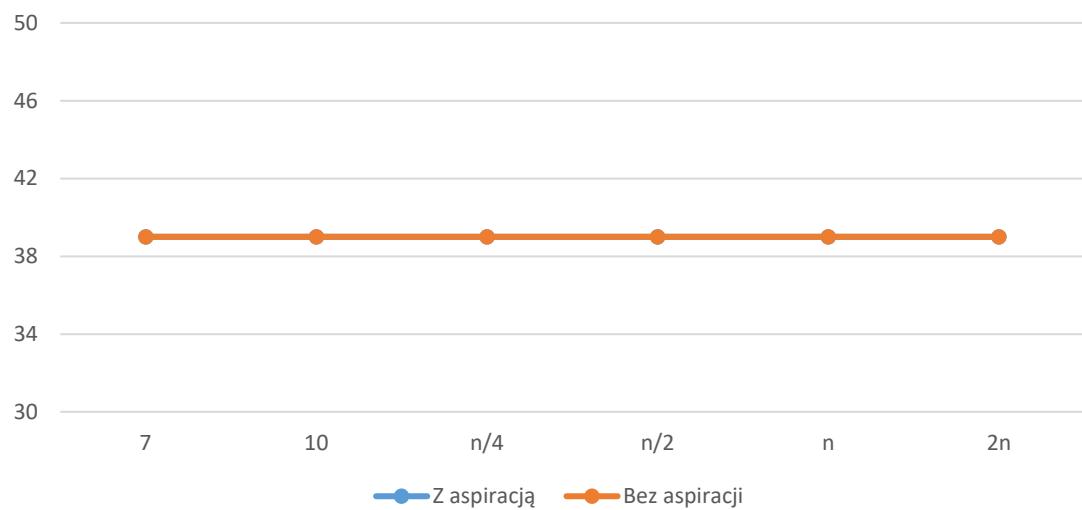
Wykres 5: Porównanie najlepszych wyników instancji **si1032.tsp** z i bez kryterium aspiracji

### br17.atsp

	Rozmiar tabu:	7	10	n/4	n/2	n	2n
Z aspiracją	Średnia droga	39	39	39	39	39	39
	Średni błąd (%)	0	0	0	0	0	0
	Najkrótsza droga	39	39	39	39	39	39
	Błąd najkrótszej drogi (%)	0	0	0	0	0	0
Bez aspiracji	Średnia droga	39	39	39	39	39	39
	Średni błąd (%)	0	0	0	0	0	0
	Najkrótsza droga	39	39	39	39	39	39
	Błąd najkrótszej drogi (%)	0	0	0	0	0	0

Tabela 4: Wyniki testów **br17.atsp** dla różnego rozmiaru tabu

### gr17.atsp - zmiana rozmiaru tabu - średni wynik



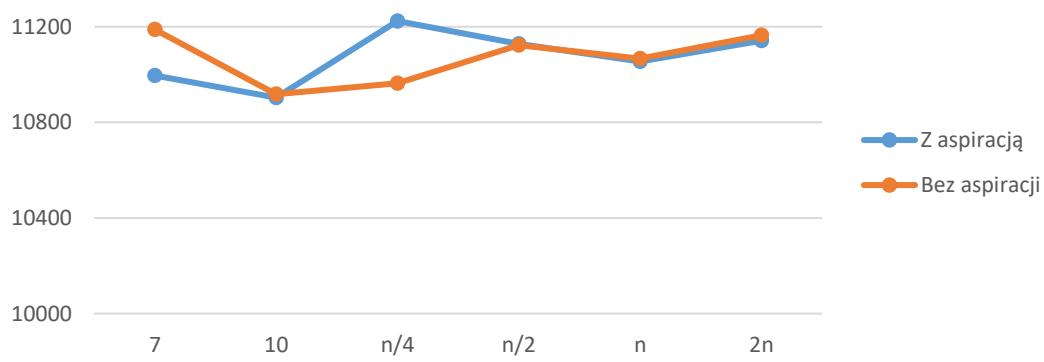
Wykres 6: Porównanie średnich wyników instancji **br17.atsp** z i bez kryterium aspiracji

### ftv170.atsp

	Rozmiar tabu:	7	10	n/4	n/2	n	2n
Z aspiracją	Średnia droga	10995,3	10903,6	11223,8	11127,7	11054,1	11141,2
	Średni błąd (%)	299,10	295,77	307,40	303,91	301,24	304,40
	Najkrótsza droga	10171	10702	10923	10790	10650	10650
	Błąd najkrótszej drogi (%)	269,18	288,46	296,48	291,65	286,57	286,57
Bez aspiracji	Średnia droga	11188,1	10917,8	10964,2	11123,5	11066,5	11164,9
	Średni błąd (%)	306,10	296,29	297,97	303,76	301,69	305,26
	Najkrótsza droga	10511	10465	10676	10715	10858	10920
	Błąd najkrótszej drogi (%)	281,52	279,85	287,51	288,93	294,12	296,37

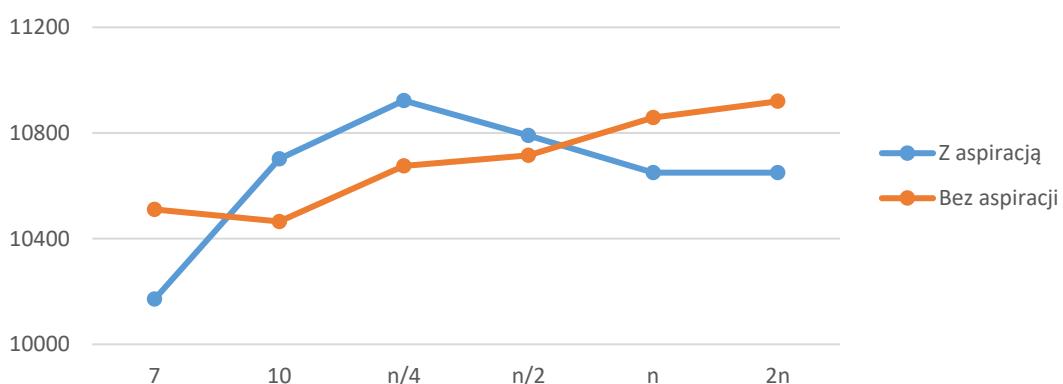
Tabela 5: Wyniki testów **ftv170.atsp** dla różnego rozmiaru tabu

### ftv170.atsp - zmiana rozmiaru tabu - średni wynik



Wykres 7: Porównanie średnich wyników instancji **ftv170.atsp** z i bez kryterium aspiracji

### ftv170.atsp - zmiana rozmiaru tabu - najlepszy wynik



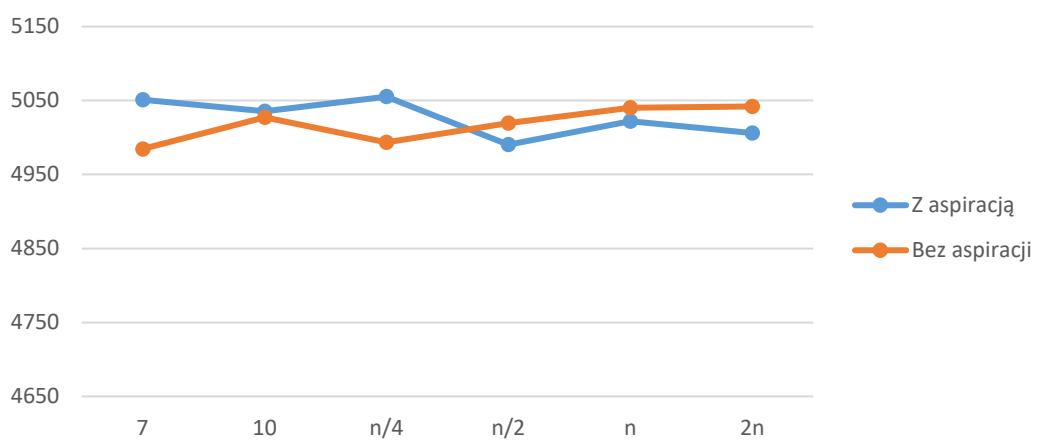
Wykres 8: Porównanie najlepszych wyników instancji **ftv170.atsp** z i bez kryterium aspiracji

### rbg443.atsp

	Rozmiar tabu:	7	10	n/4	n/2	n	2n
Z aspiracją	Średnia droga	5050,9	5035,5	5055,2	4990,3	5022,2	5006,1
	Średni błąd (%)	85,69	85,13	85,85	83,47	84,64	84,05
	Najkrótsza droga	4970	4926	4944	4891	4895	4789
	Błąd najkrótszej drogi (%)	82,72	81,10	81,76	79,82	79,96	76,07
Bez aspiracji	Średnia droga	4984,6	5027,2	4993,7	5019,3	5040,4	5042
	Średni błąd (%)	83,26	84,82	83,59	84,53	85,31	85,37
	Najkrótsza droga	4754	4931	4719	4897	4918	4815
	Błąd najkrótszej drogi (%)	74,78	81,29	73,49	80,04	80,81	77,02

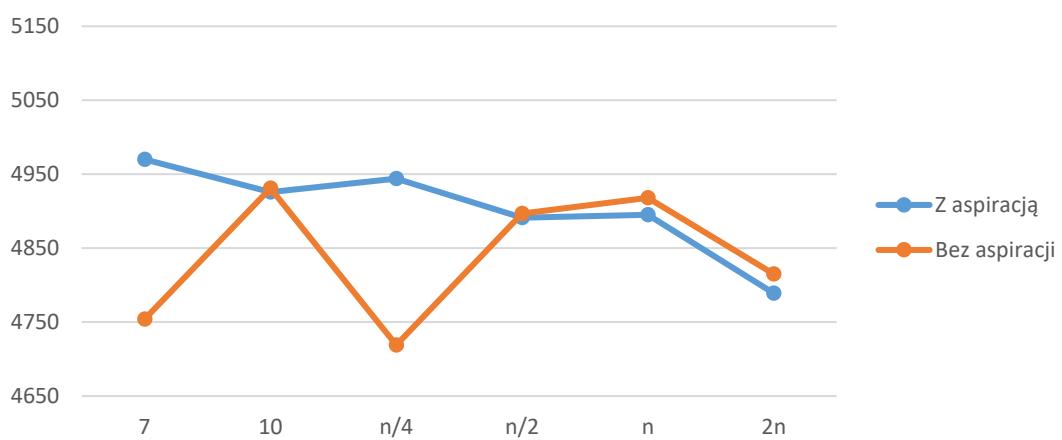
Tabela 6: Wyniki testów **rbg443.atsp** dla różnego rozmiaru tabu

### rbg443.atsp - zmiana rozmiaru tabu - średni wynik



Wykres 9: Porównanie średnich wyników instancji **rbg443.atsp** z i bez kryterium aspiracji

### rbg443.atsp - zmiana rozmiaru tabu - najlepszy wynik



Wykres 10: Porównanie najlepszych wyników instancji **rbg443.atsp** z i bez kryterium aspiracji

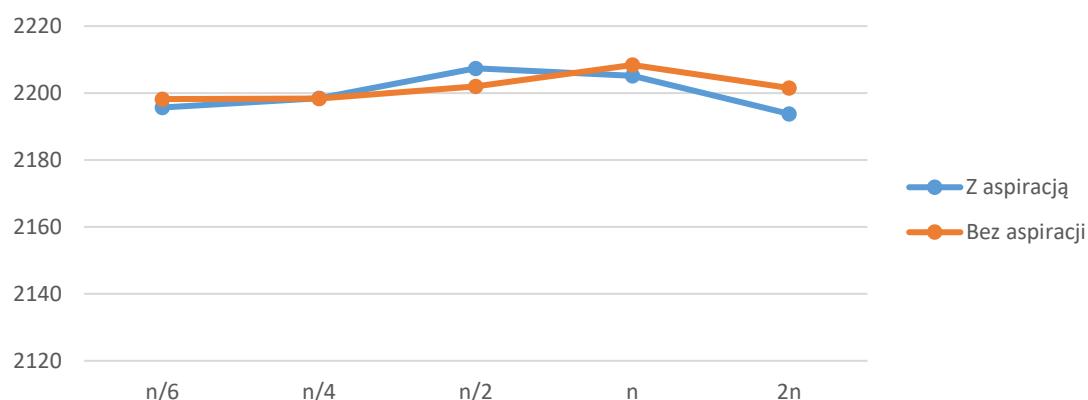
## 5.2. Badanie długości kadencji:

**gr17.tsp**

Długość kadencji:		n/6	n/4	n/2	n	2n
Z aspiracją	Średnia droga	2195,7	2198,5	2207,4	2205,2	2193,8
	Średni błąd (%)	5,31	5,44	5,87	5,76	5,22
	Najkrótsza droga	2153	2135	2164	2177	2161
	Błąd najkrótszej drogi (%)	3,26	2,40	3,79	4,41	3,65
Bez aspiracji	Średnia droga	2198,2	2198,4	2202	2208,4	2201,5
	Średni błąd (%)	5,43	5,44	5,61	5,92	5,59
	Najkrótsza droga	2153	2135	2164	2177	2161
	Błąd najkrótszej drogi (%)	3,26	2,40	3,79	4,41	3,65

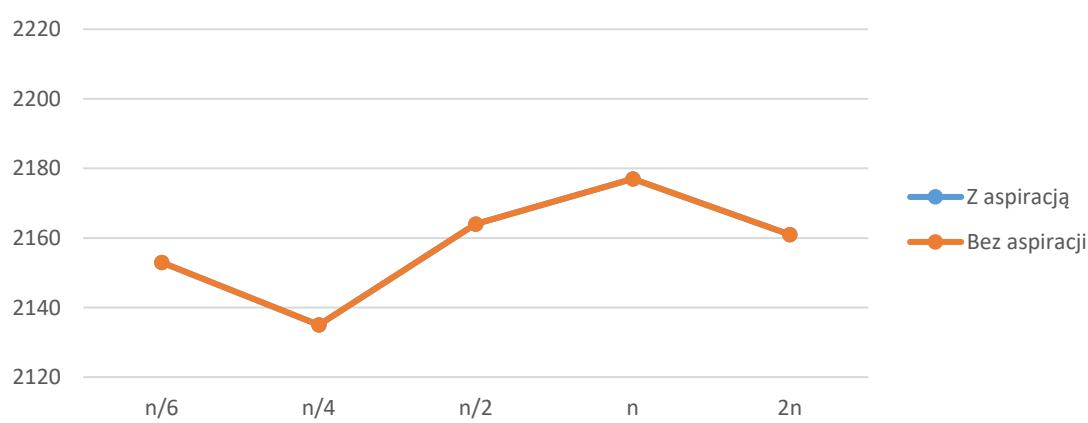
Tabela 7: Wyniki testów **gr17.tsp** dla różnej długości kadencji

**gr17.tsp - zmiana kadencji  
- średni wynik**



Wykres 11: Porównanie średnich wyników instancji **gr17.tsp** z i bez kryterium aspiracji

**gr17.tsp - zmiana kadencji  
- najlepszy wynik**



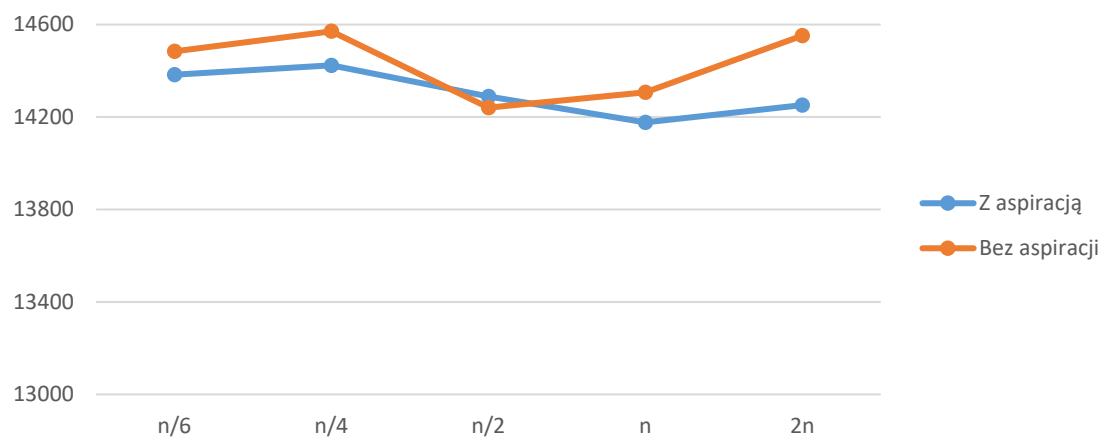
Wykres 12: Porównanie najlepszych wyników instancji **gr17.tsp** z i bez kryterium aspiracji

### pa561.tsp

	Długość kadencji:	n/6	n/4	n/2	n	2n
Z aspiracją	Średnia droga	14383,8	14423,7	14288,6	14176,9	14251,7
	Średni błąd (%)	420,59	422,03	417,14	413,10	415,81
	Najkrótsza droga	13322	13909	13486	13623	13233
	Błąd najkrótszej drogi (%)	382,16	403,40	388,09	393,05	378,94
Bez aspiracji	Średnia droga	14483,9	14570,8	14240,7	14307,6	14552,6
	Średni błąd (%)	424,21	427,35	415,41	417,83	426,70
	Najkrótsza droga	13973	14008	13091	13731	13708
	Błąd najkrótszej drogi (%)	405,72	406,99	373,80	396,96	396,13

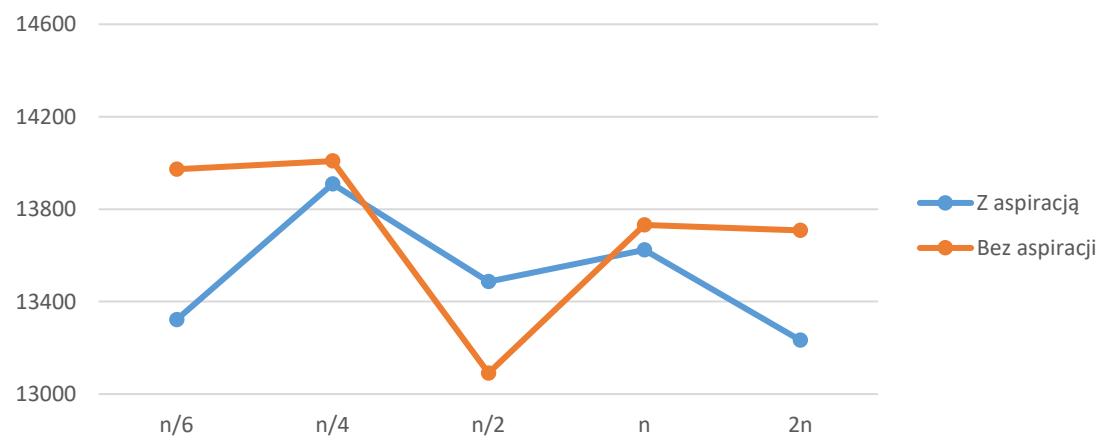
Tabela 8: Wyniki testów **pa561.tsp** dla różnej długości kadencji

### pa561.tsp - zmiana kadencji - średni wynik



Wykres 13: Porównanie średnich wyników instancji **pa561.tsp** z i bez kryterium aspiracji

### pa561.tsp - zmiana kadencji - najlepszy wynik



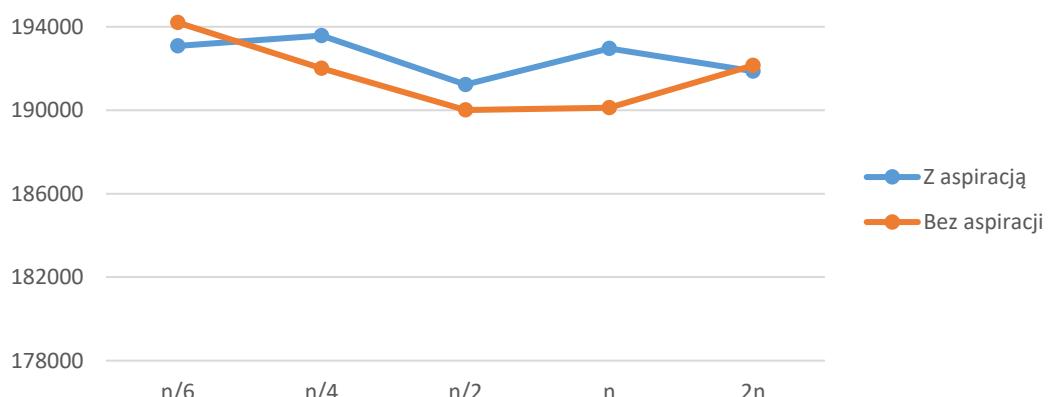
Wykres 14: Porównanie najlepszych wyników instancji **pa561.tsp** z i bez kryterium aspiracji

### si1032.tsp

	Długość kadencji:	n/6	n/4	n/2	n	2n
Z aspiracją	Średnia droga	193090,4	193580	191230	192959	191874,6
	Średni błąd (%)	108,41	108,94	106,40	108,27	107,10
	Najkrótsza droga	186690	190987	187567	190598	186698
	Błąd najkrótszej drogi (%)	101,50	106,14	102,45	105,72	101,51
Bez aspiracji	Średnia droga	194198,7	192007,9	190009,8	190118,4	192155,1
	Średni błąd (%)	109,60	107,24	105,08	105,20	107,40
	Najkrótsza droga	190377	188521	187474	181679	180766
	Błąd najkrótszej drogi (%)	105,48	103,48	102,35	96,09	95,11

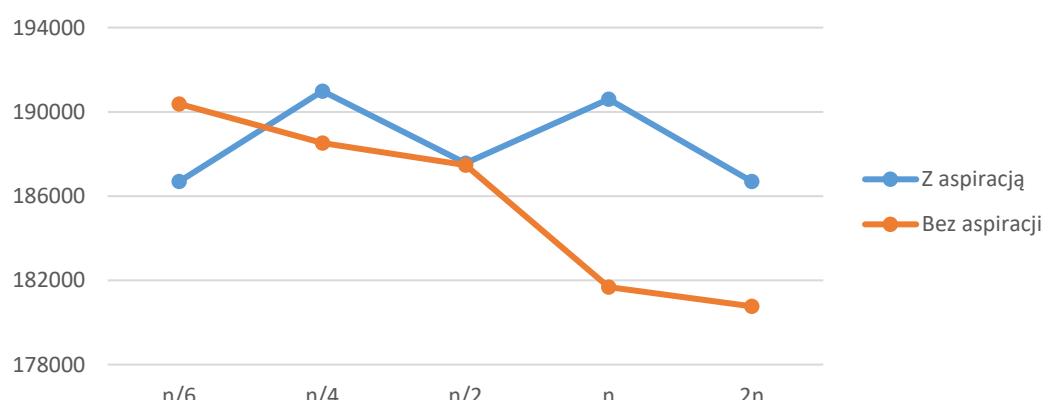
Tabela 9: Wyniki testów **si1032.tsp** dla różnej długości kadencji

### si1032.tsp - zmiana kadencji - średni wynik



Wykres 15: Porównanie średnich wyników instancji **si1032.tsp** z i bez kryterium aspiracji

### si1032.tsp - zmiana kadencji - najlepszy wynik



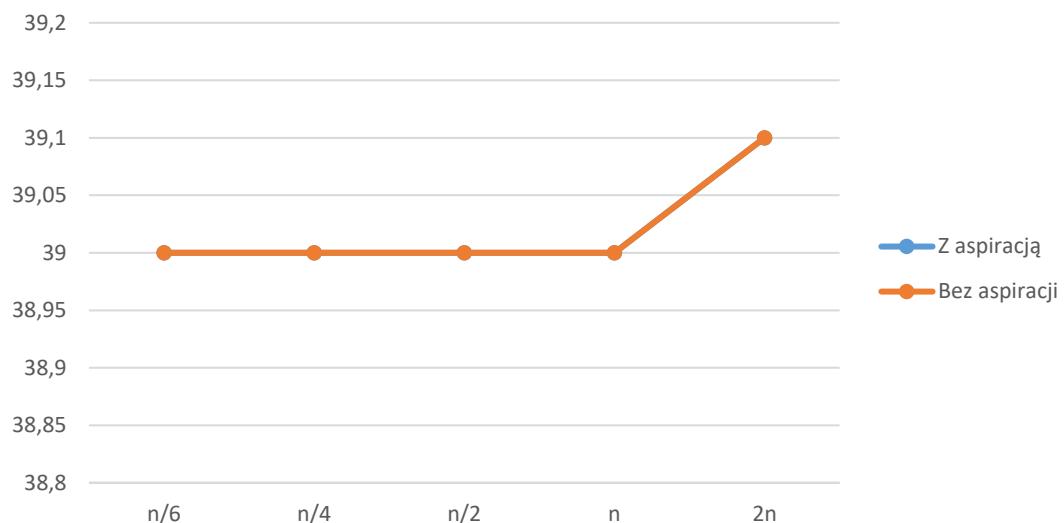
Wykres 16: Porównanie najlepszych wyników instancji **si1032.tsp** z i bez kryterium aspiracji

### br17.atsp

	Długość kadencji:	n/6	n/4	n/2	n	2n
Z aspiracją	Średnia droga	39	39	39	39	39,1
	Średni błąd (%)	0	0	0	0	0,26
	Najkrótsza droga	39	39	39	39	39
	Błąd najkrótszej drogi (%)	0	0	0	0	0
Bez aspiracji	Średnia droga	39	39	39	39	39,1
	Średni błąd (%)	0	0	0	0	0,26
	Najkrótsza droga	39	39	39	39	39
	Błąd najkrótszej drogi (%)	0	0	0	0	0

Tabela 10: Wyniki testów **br17.atsp** dla różnej długości kadencji

### br17.atsp - zmiana kadencji - średni wynik



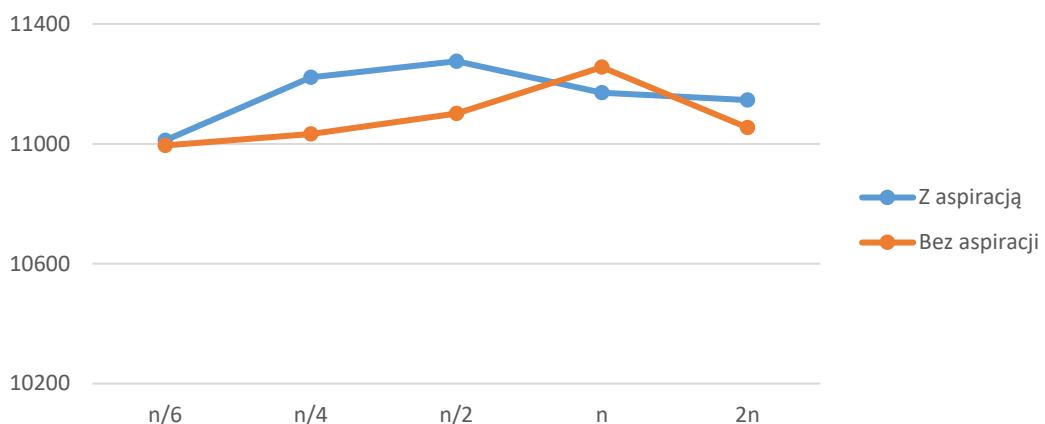
Wykres 17: Porównanie średnich wyników instancji **br17.atsp** z i bez kryterium aspiracji

### ftv170.atsp

	Długość kadencji:	n/6	n/4	n/2	n	2n
Z aspiracją	Średnia droga	11011,6	11221,3	11275,5	11170,9	11145,6
	Średni błąd (%)	299,70	307,31	309,27	305,48	304,56
	Najkrótsza droga	10537	10782	10942	10695	10639
	Błąd najkrótszej drogi (%)	282,47	291,36	297,17	288,20	286,17
Bez aspiracji	Średnia droga	10994,6	11033	11100,7	11255,6	11054,1
	Średni błąd (%)	299,08	300,47	302,93	308,55	301,24
	Najkrótsza droga	10319	10710	10732	10937	10578
	Błąd najkrótszej drogi (%)	274,56	288,75	289,55	296,99	283,96

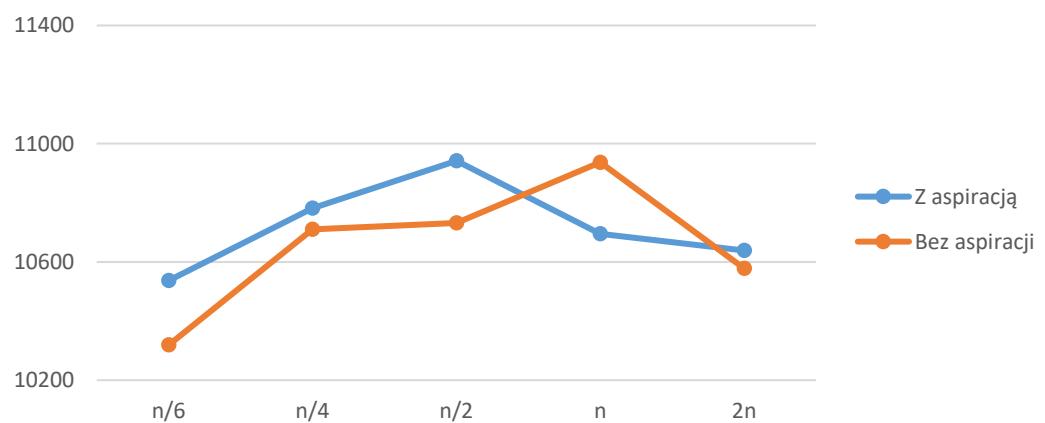
Tabela 11: Wyniki testów **ftv170.atsp** dla różnej długości kadencji

### ftv170.atsp - zmiana kadencji - średni wynik



Wykres 18: Porównanie średnich wyników instancji **ftv170.atsp** z i bez kryterium aspiracji

### ftv170.atsp - zmiana kadencji - najlepszy wynik

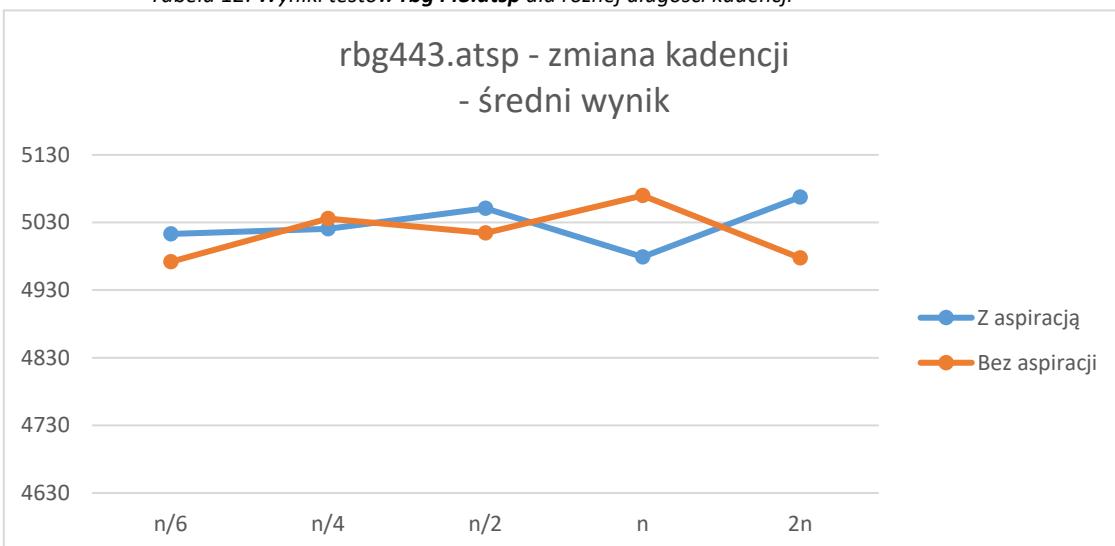


Wykres 19: Porównanie najlepszych wyników instancji **ftv170.atsp** z i bez kryterium aspiracji

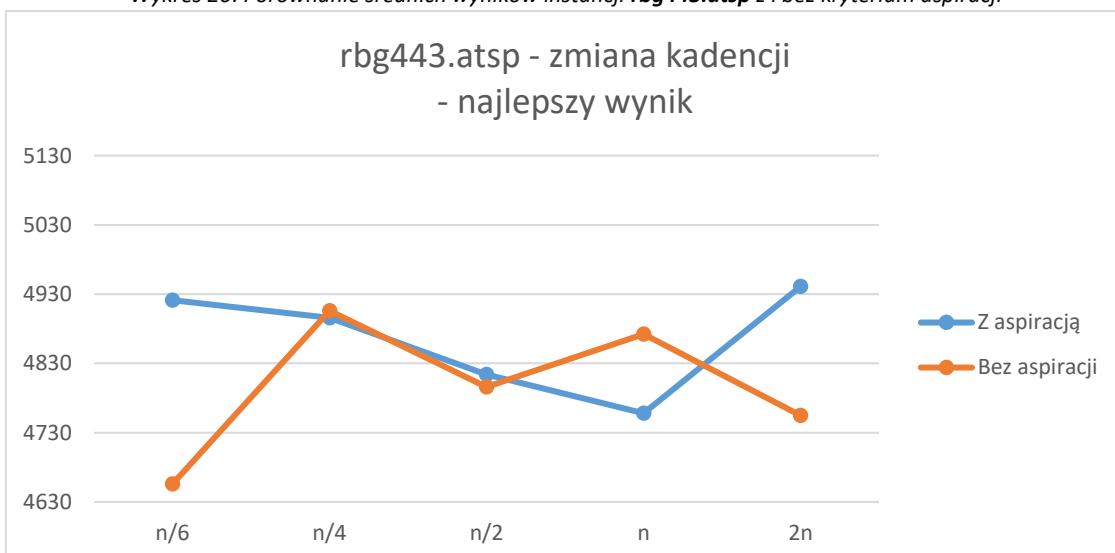
### rbg443.atsp

	Długość kadencji:	n/6	n/4	n/2	n	2n
Z aspiracją	Średnia droga	5013,3	5020,6	5051,2	4979,1	5067,6
	Średni błąd (%)	84,31	84,58	85,71	83,06	86,31
	Najkrótsza droga	4921	4896	4814	4758	4941
	Błąd najkrótszej drogi (%)	80,92	80,00	76,99	74,93	81,65
<hr/>						
Bez aspiracji	Średnia droga	4971,9	5035,6	5014,3	5069,9	4977,4
	Średni błąd (%)	82,79	85,13	84,35	86,39	82,99
	Najkrótsza droga	4656	4906	4796	4872	4755
	Błąd najkrótszej drogi (%)	71,18	80,37	76,32	79,12	74,82

Tabela 12: Wyniki testów rbg443.atsp dla różnej długości kadencji



Wykres 20: Porównanie średnich wyników instancji rbg443.atsp z i bez kryterium aspiracji



Wykres 21: Porównanie największych wyników instancji rbg443.atsp z i bez kryterium aspiracji

## 6. Wnioski

Testowany algorytm w większości dawał poprawne wyniki, których dokładność zgodnie z oczekiwaniemi różniła się w zależności od wielkości instancji. Pierwszą rzeczą rzucającą się w oczy jest to, jak wyniki są uzależnione nie tylko od samej wielkości, ale od tego jaka jest struktura konkretnej instancji, co doskonale widać na przykładzie plików **gr17.tsp** oraz **br17.tsp**. Chodź w obydwu przypadkach mamy tą samą liczbę wierzchołków, w przypadku **br17** dostajemy niemal zawsze wynik dokładny (z wyjątkiem dużego rozmiaru kadencji - tabela 10), natomiast w **gr17** wynik ten ani razu nie osiągnął optymalnej wartości, zatrzymując się w minimum lokalnym które jest bliskie optymalnego rozwiązania (tabela 1).

Chodź dla większości błędów sięgały okolic dwukrotności optymalnego wyniku, inaczej sytuacja wyglądała dla dwóch badanych instancji: **pa561.tsp** oraz **f1v170.atsp**. Wymienione instancje dawały wyniki najbardziej oddalone od optymalnej wartości, mimo że nie były to największe instancje dla danych problemów, to w przypadku **pa561** wynik był gorszy ponad czterokrotnie od optymalnej wartości (tabela 2 i 8). Najprawdopodobniej jest to spowodowane tym, że pliki te (w szczególności pa561) mają specyficzne rozłożone danych utrudniające trafienie w lepsze minimum lokalne, oraz w konsekwencji minimum globalne. Jedyną możliwością poprawienia tych wyników byłoby przebadanie algorytmu pod kątem konkretnej instancji i dobraniu odpowiednich dla niej parametrów.

Tutaj dochodzimy do największego problemu związanego z metodą Tabu Search, jaką jest ustawianie jego parametrów. Zastosowane parametry uniwersalne które sprawdziły się dla małych instancji, jak i nawet dużych dając dla **si1032.tsp** błąd w okolicach dwukrotności optymalnego wyniku (tabela 3 i 9), nie sprawdzają się dobrze dla wcześniej wspomnianych **pa561** i **f1v170**. Widać więc, że każda instancja reaguje na zmianę parametrów inaczej. Przykładowo w badaniu długości kadencji, **gr17** średnio najlepsze rozwiązanie otrzymuje dla kadencji o długości  $2n$  (gdzie  $n$  to wielkość instancji), oraz dobre w przypadku kadencji równej  $n/6$ , dając jednak najgorsze wyniki dla kadencji równej  $n$  (wykres 7). Inną z kolei tendencję wykazuje instancja o największej liczebności - **pa561** w której to wyniki są lepsze w przypadku kadencji równej właśnie  $n$ , lub  $n/2$  (tabela 8), czyli przeciwnie do tego co zaobserwowaliśmy w **gr17**. Widać więc że do każdego problemu dokładność wyniku zależy od dobranych parametrów, których wartości mają różny skutek w zależności od budowy instancji.

Podczas testowania badaniu zostały poddane jedynie trzy parametry: wielkość tabu, długość kadencji oraz dla obydwu tych przypadków działanie z i bez funkcji aspiracji. Mimo że parametrów wpływających na działanie algorytmu jest więcej, zostały wybrane te najbardziej znaczące ze względu na dużą trudność w dostosowaniu ich wszystkich do poszczególnych instancji. Z testów przeprowadzonych na badanych parametrach wynika że większy wpływ na wyniki ma długość kadencji elementu w tabu. Warto jednak zaznaczyć, że kadencja z wielkością tabu bardzo współpracują ze sobą, i mogą dawać zupełnie inne wyniki w przypadku zmiany obydwu w tym samym czasie. Niestety testy takiej zależności były by o wiele bardziej czasochłonne.

Ciekawie wygląda sytuacja w przypadku zastosowania funkcji aspiracji. Zarówno podczas badania wielkości tabu jak i kadencji wszelkie testy zostały wykonane zarówno z użyciem jak i bez użycia tej funkcji, i chodź w każdej instancji widać inne efekty jego zastosowania, to najczęściej poprawiała ona wynik w przypadku małych oraz dużych wielkości tabu. Co więcej, podobnie wygląda sytuacja w przypadku długości kadencji, co możemy zaobserwować na niemal każdym wykresie. Największym odstępstwem od tego spostrzeżenia jest plik **rbg443**, szczególnie w przypadku testowania długości kadencji (wykres 20). Bardzo prawdopodobne, że dobranie innego kryterium aspiracji pozwoliło by częściej uzyskiwać lepsze wyniki dla innych parametrów.

Ostatnią rzeczą wartą zauważenia jest porównanie wyników dla najmniejszych plików, z badanym w pierwszym etapie projektu algorytmem Branch and Bound. Chodź poprzedni algorytm oferował nam rozwiązanie optymalne, to jednak czasowo o wiele lepiej wypada metoda metaheurystyczna jaką jest Tabu Search. W przypadku **gr17**, który to daje błęd wyniku w okolicach 5% w 30 sekund, gdzie rozwiązanie optymalne zostało znalezione przy pomocy BnB w 64 sekundy różnica między błędem a czasem nie jest bardzo duża. Jednak w przypadku **br17** znalezienie optymalnego rozwiązania w 30 sekund przy pomocy Tabu Search w porównaniu z ponad 16 minutami dla BnB świetnie pokazuje przewagę metaheurystyki w rozwiązywaniu problemu, gdyż dla małych instancji przy dobrych parametrach mamy dużą szansę otrzymania wyniku optymalnego w dużo krótszym czasie. Widać więc, że jeśli nie potrzebujemy optymalnego rozwiązania, metoda Tabu Search jest bardziej sensowna nawet w przypadku niewielkich instancji jak np. wspomniane **br17**.