Sprawozdanie lista1

Autorzy: Tomasz Hołub (261718) & Piotr Piotrowski (261723)

Wstęp:

Sprawozdanie zawiera opis i analizę badań prowadzonych nad algorytmem metaheurystycznym Tabu Search, rozwiązującym problem komiwojażera. Rozpatrywanymi czynnikami są:

- Wpływ wielkości instancji na czas wykonywania algorytów
- Wpływ parametrów na dokładność uzyskanego wyniku.

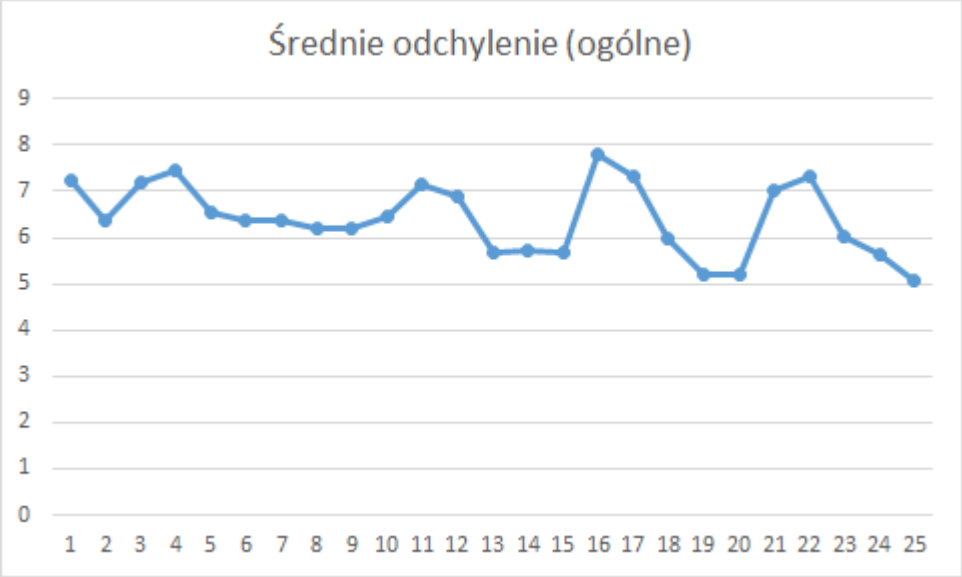
Algorytmy zostały napisane w języku Python, pełny kod znajduję się w repozytorium :

https://github.com/holubek01/Tabu_2

Zaimplementowane algorytmy:

- Tabu Search

1. Badania



Wykres 1.1 opisujący procentową wartość odchylenia w zależności od wprowadzonych parametrów dla algorytmu Tabu Search

Legenda:

	log(n)	sqrt(n)	n/2	n	nlog(n)	queue_size
log(n)	1	2	3	4	5	
sqrt(n)	6	7	8	9	10	
n/2	11	12	13	14	15	
n	16	17	18	19	20	
nlog(n)	21	22	23	24	25	
max_iteration						

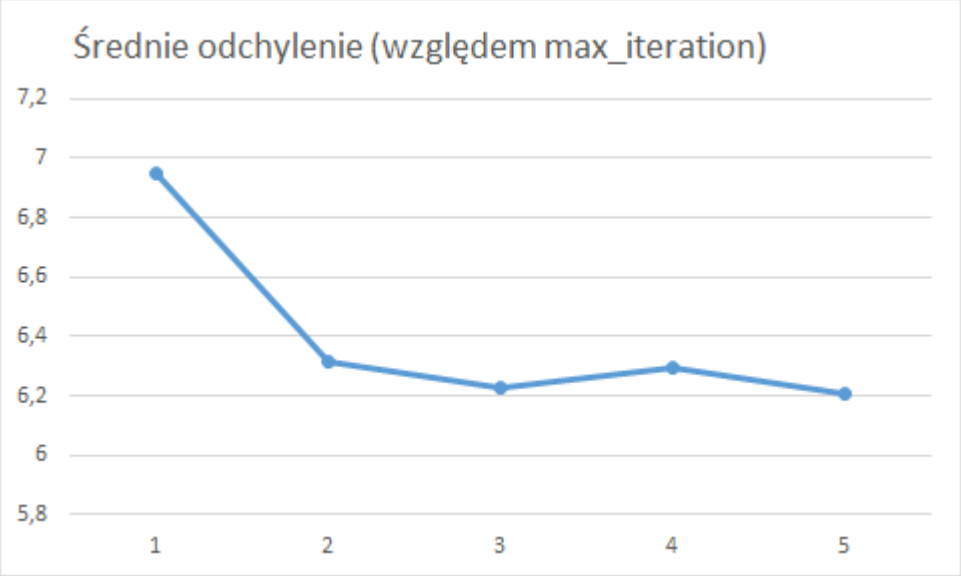
Wartość odchylenia liczona jest według poniższego wzoru:

$$\frac{(x-y)}{y} * 100\ \%$$

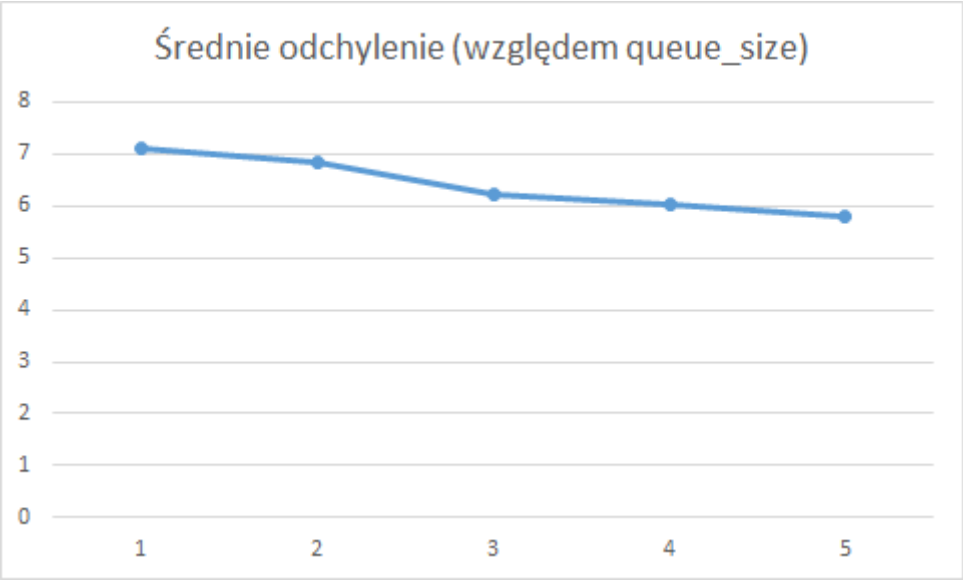
Gdzie:

$$x - wartość\ funkcji\ celu\ dla\ danej\ instancji\ (algorytm\ Tabu\ Search)$$

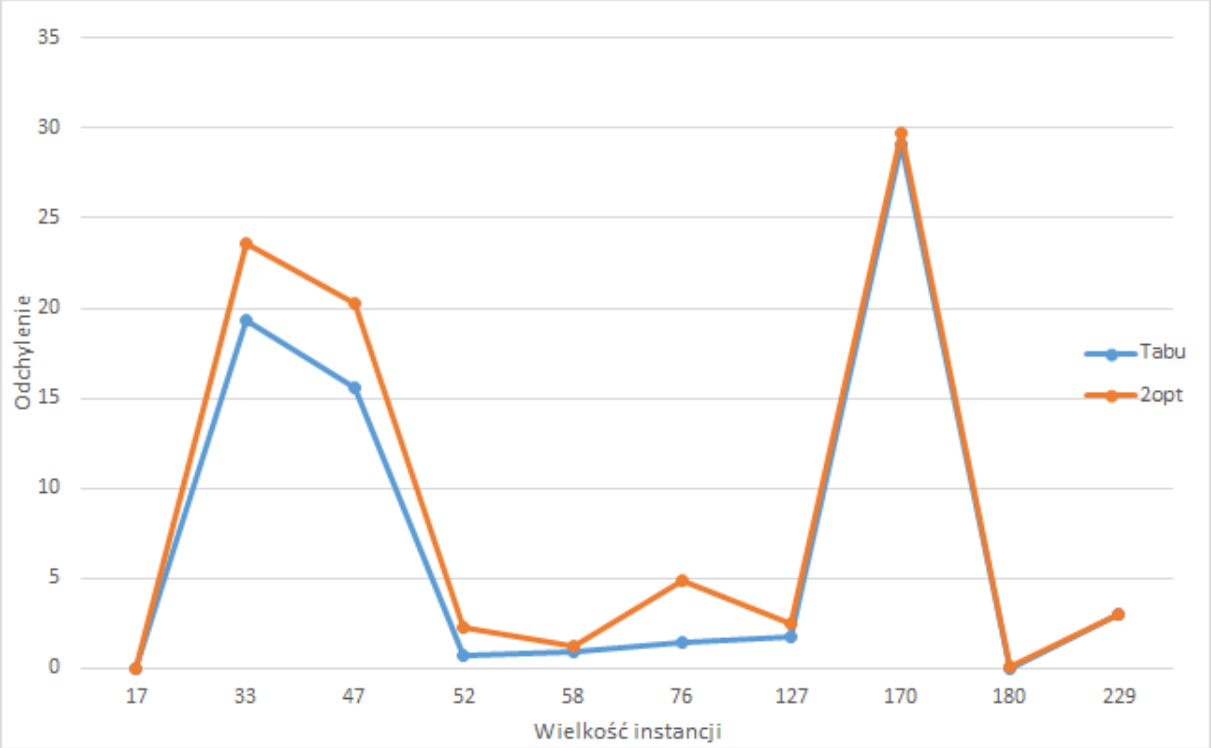
$$y - wartość\ funkcji\ celu\ dla\ najlepszego\ znanego\ rozwiązania$$



Wykres 1.2 opisujący procentową wartość odchylenia w zależności od maksymalnej liczby iteracji określonej jako warunek stopu



Wykres 1.3 opisujący procentową wartość odchylenia w zależności od wielkości listy zabronień tabu



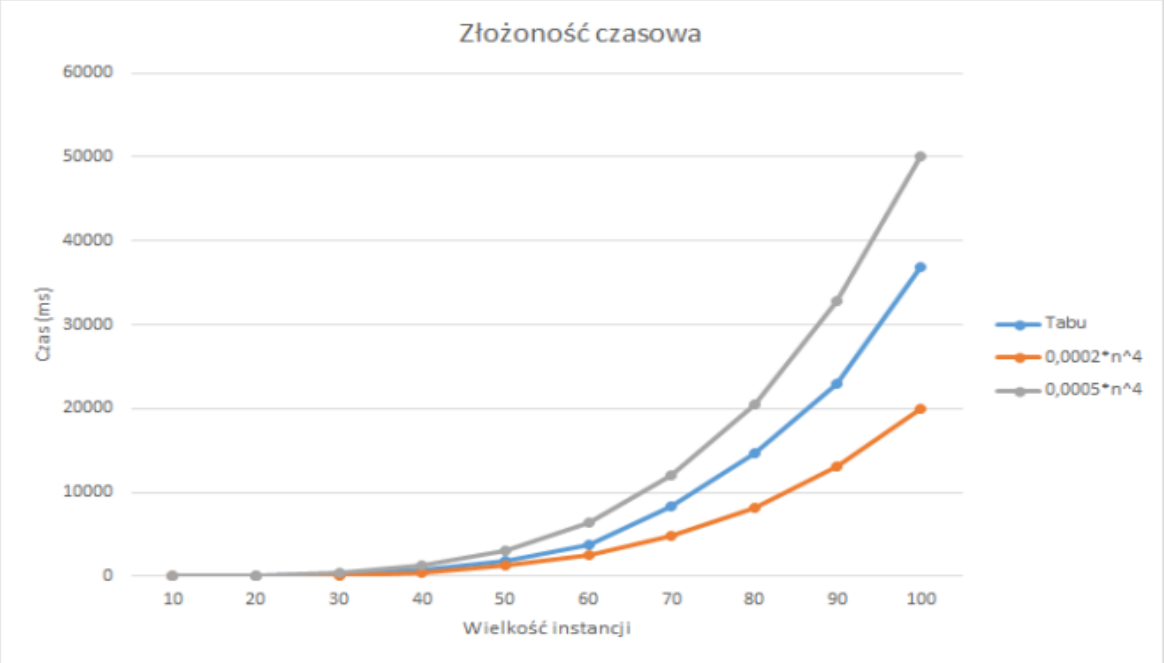
Wykres 1.4 wykres opisujący procentową wartość odchylenia dla wybranych instancji z TSPLIP dla algorytmu Tabu Search oraz 2opt.

2. Złożoność czasowa Tabu Search

W tym punkcie analizować będziemy złożoność czasową algorytmu Tabu Search.

Przed stworzeniem wykresu poddaliśmy analizie nasz kod aby oszacować złożoność czasową, co pozwoliło nam dobrać odpowienie ograniczenia.

Samo wnętrzne „koks funkcji” jest złożoności O(n) ale potem jest rekurencyjnie wywoływana, ale nie możemy określić ile dokładnie razy wykona się rekurencja. Przyjmujemy natomiast liczbe rekurencji jako n. Zatem złożoność czasowa algorytmu Tabu Search to O(n^4)



Wykres 3.1 opisujący złożoność algorytmu Tabu Search

Wykres czasu w zależności od wielkości instancji dla algorytmu „Tabu Search” ograniczyliśmy z góry funkcją daną wzorem $f(x) = \frac{5}{10000}x^4$ natomiast z dołu funkcją daną wzorem $f(x) = \frac{2}{10000}x^4$. Zatem można wywnioskować, że złożoność czasowa algorytmu „Tabu Search” wynosi n^4.

3. Analiza algorytmu Tabu Search

1.Przeszukiwanie sąsiedztwa zostało zaimplementowane z użyciem „invert”. Operacja ta dodatkowo została poddana akceleracji.

2. Parametry (rozmiar listy tabu oraz maksymalna liczba iteracji) zostały dobrane na podstawie przeprowadzonych badań. Dla instancji o rozmiarze size oba parametry przyjmują wartość jako size/2.

3. Lista tabu ma odgórnie określony rozmiar, jednak po zapełnieniu, gdy chcemy dodać kolejny element, najstarszy element zostaje usunięty, natomiast element, który chcemy dodać zostaje umieszczony na końcu kojeki (Struktura FIFO).

4. Podczas przeszukiwania sąsiedztwa dana droga powstała przez operację z wykorzystaniem parametrów znajdujących się w liście tabu, jest brana pod uwagę, tylko wtedy gdy jej waga jest mniejsza od najmniejszej do tej pory znanej wagi.

5. Warunkiem stopu jest przekroczenie maksymalnej liczby iteracji bez poprawy wyniku. Po pierwszym takim przekroczeniu program się nie kończy, ale funkcja zostaje wywołana od początku, z argumentem będącym losową spośród 10 ostatnich najlepszych dróg.