Projektowanie Efektywnych Algorytmów
Projekt

18/11/2022

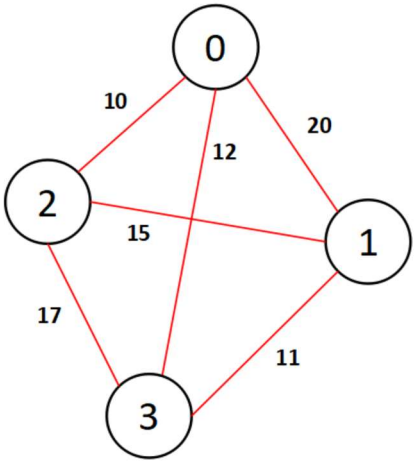
259198 Lena Kuźma

2) Algorytm Helda-Karpa

spis treści	*strona*

1. Sformułowanie zadania

Zadane polega na implementacji i zbadaniu efektywności algorytmu Helda-Karpa rozwiązującego problem komiwojażera. 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*). Należy zbadać złożoność czasową i pamięciową algorytmu.



Rysunek 1: Graf pełny ważony - symboliczna reprezentacja problemu komiwojażera

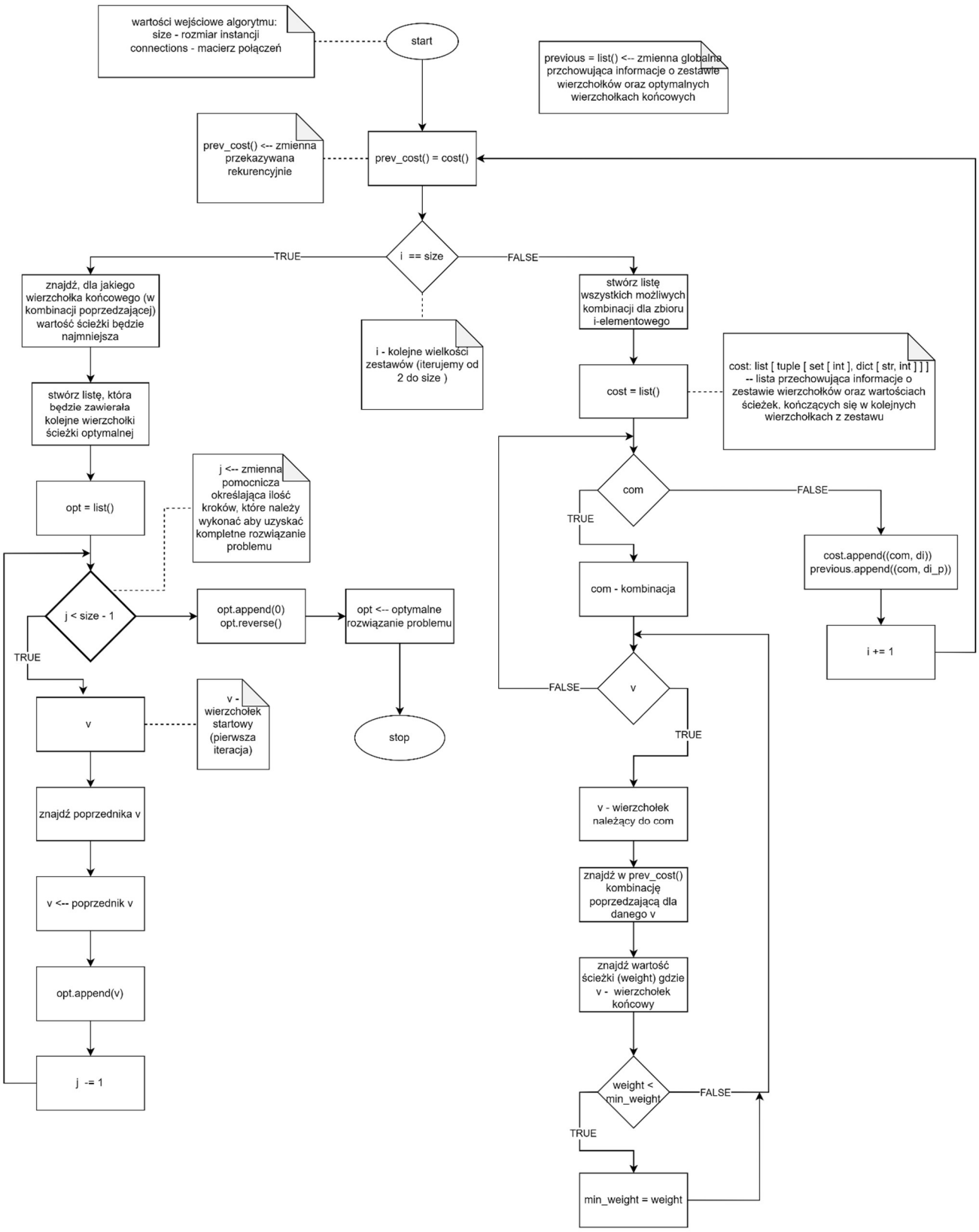
2. Metoda

Algorytm Helda-Karpa to algorytm dokładny (zwracający optymalne rozwiązanie), który został stworzony do rozwiązywania problemu komiwojażera. Opiera się na programowaniu dynamicznym, które wykorzystuje podział rozwiązywanego problemy na podproblemy. Algorytm ma złożoność czasową $O(n^2 2^n)$ i złożoność pamięciową $O(n2^n)$. Wprowadza ulepszenie względem algorytmu *Brute Force,* które polega na zmniejszeniu liczby rozpatrywanych problemów z liczby wszystkich permutacji do liczby wszystkich kombinacji wierzchołków.

Działanie algorytmu (Dokładny opis przedstawiony został na *Rysunku 2*):

1. Wybór wierzchołka startowego
2. Wyznaczenie wartości ścieżek bazowych: zaczynających się w wybranym wierzchołku startowym i kończących się w dowolnym innym wierzchołku.
3. Zwiększenie zbioru wierzchołków, przez które przechodzi ścieżka o 1.
4. Wyznaczenie optymalnych wartości ścieżek wychodzących z wierzchołka startowego i przechodzących przez wierzchołki z danego zbioru. Powrót do kroku 3, chyba, że zbiór zawiera już wszystkie wierzchołki oprócz startowego.
5. Wyznaczenie rozwiązania optymalnego poprzez prześledzenie rozwiązań podproblemów rozwiązywanych w kroku 4.

3. Algorytm



Rysunek 2: Schemat blokowy opisujący działanie zastosowanego algorytmu

4. Dane testowe

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

tsp_6_1.txt

tsp_12.txt

tsp_13.txt

tsp_17.txt

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

Do wykonania badań wybrano następujący zestaw instancji:

tsp_6-2.txt

tsp_10.txt

tsp_12.txt

tsp_15.txt

gr12.tsp.txt

gr21.tsp.txt

Źródło: Teaching (uni-heidelberg.de)

5. Procedura badawcza

Należało zbadać zależność czasu rozwiązania problemu od wielkości instancji (złożoność czasową) oraz zależność zużycia pamięci od wielkości instancji (złożoność pamięciową). Procedura badawcza polegała na uruchomieniu programu (napisanego w języku *python*) sterowanego plikiem inicjującym *.INI* (format pliku: nazwa instancji, liczba wykonań rozwiązania, nazwa pliku wyjściowego, wartość optymalna, ścieżka optymalna (*Rysunek 3*)).

```
[FILE]
file1 = tsp_6_2.txt
file2 = tsp_10.txt

[ITERATOR]
i1 = 100
i2 = 50

[OUTPUT]
file1 = heldKepler.xlsx

[OPTIMAL_VALUE]
opt1 = 80
opt2 = 212

[OPTIMAL_PATH]
path1 = [0 5 1 2 3 4]
path2 = [0 3 4 2 8 7 6 9 1 5]
```

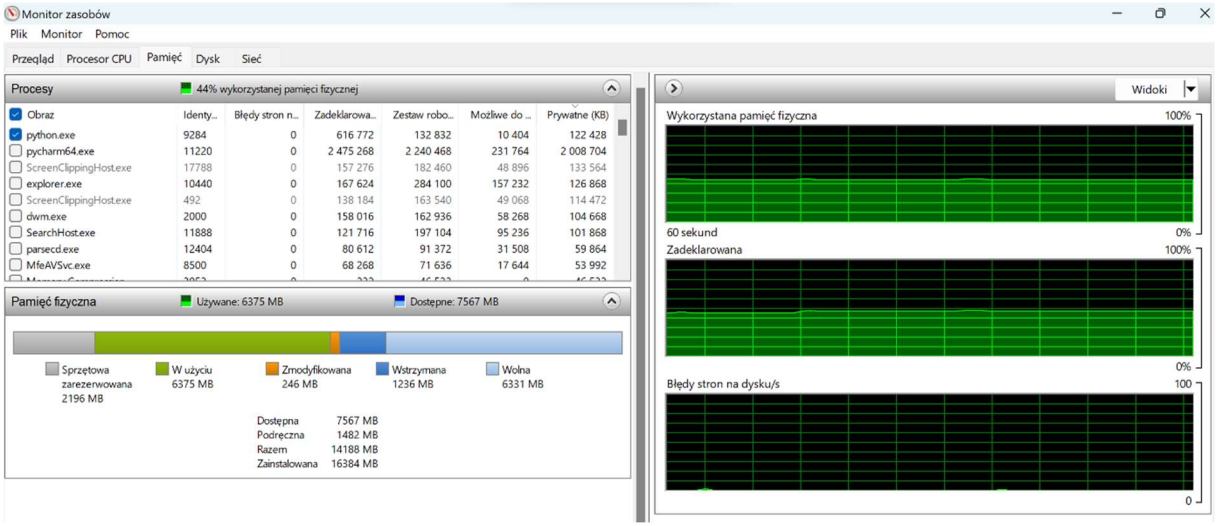
Rysunek 3: Struktura zawartości pliku inicjalizującego .INI

Każda z instancji rozwiązywana była zgodnie z liczbą jej wykonań, np. *tsp_6_2.txt* wykonana została 100 razy. Do pliku wyjściowego *heldKarp.xlsx* zapisywany był czas wykonania, otrzymane rozwiązanie (koszt ścieżki) oraz ścieżka (numery kolejnych węzłów). 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	G	H	I	J
1	Nazwa instancji		Liczba powtorzen		Wartosc optymalna		Sciezka optymalna			
2	tsp_15.txt		5		291		[0, 10, 3, 5, 7, 9, 13, 11, 2, 6, 4, 8, 14, 1, 12]			
3										
4	Czasy wykonywania algorytmu [s]:									
5	0,063104									
6	0,057178									
7	0,061983									
8	0,061156									
9	0,093735									

Rysunek 4: Fragment pliku wyjściowego w formacie .xlsx

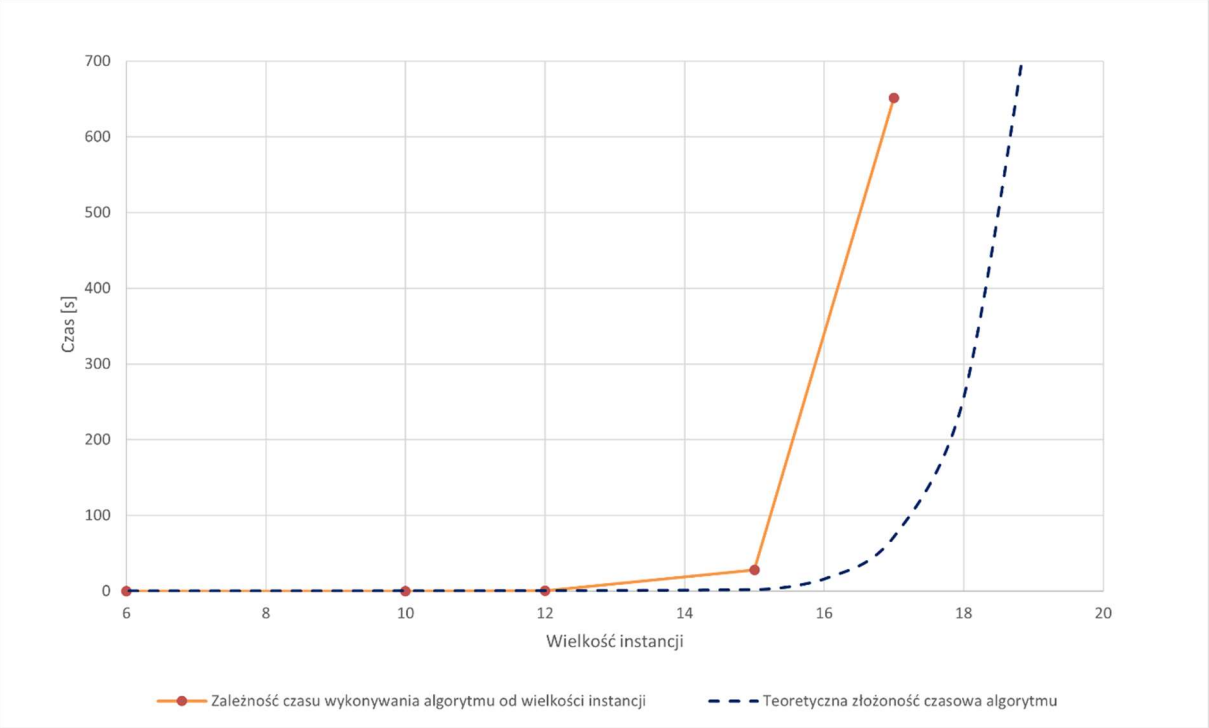
Zależność wykorzystania pamięci komputera od wielkości instancji była monitorowana na bieżąco przy pomocy systemowego *Monitora zasobów (Rysunek 5).* Ilość wykorzystywanej pamięci dla kolejnych instancji została oszacowana na podstawie aktualnego etapu wykonywania programu i odczytów z *Monitora zasobów*.



Rysunek 5: Zrzut ekranu przedstawiający wykorzystaną metodę pomiaru wykorzystywanej pamięci

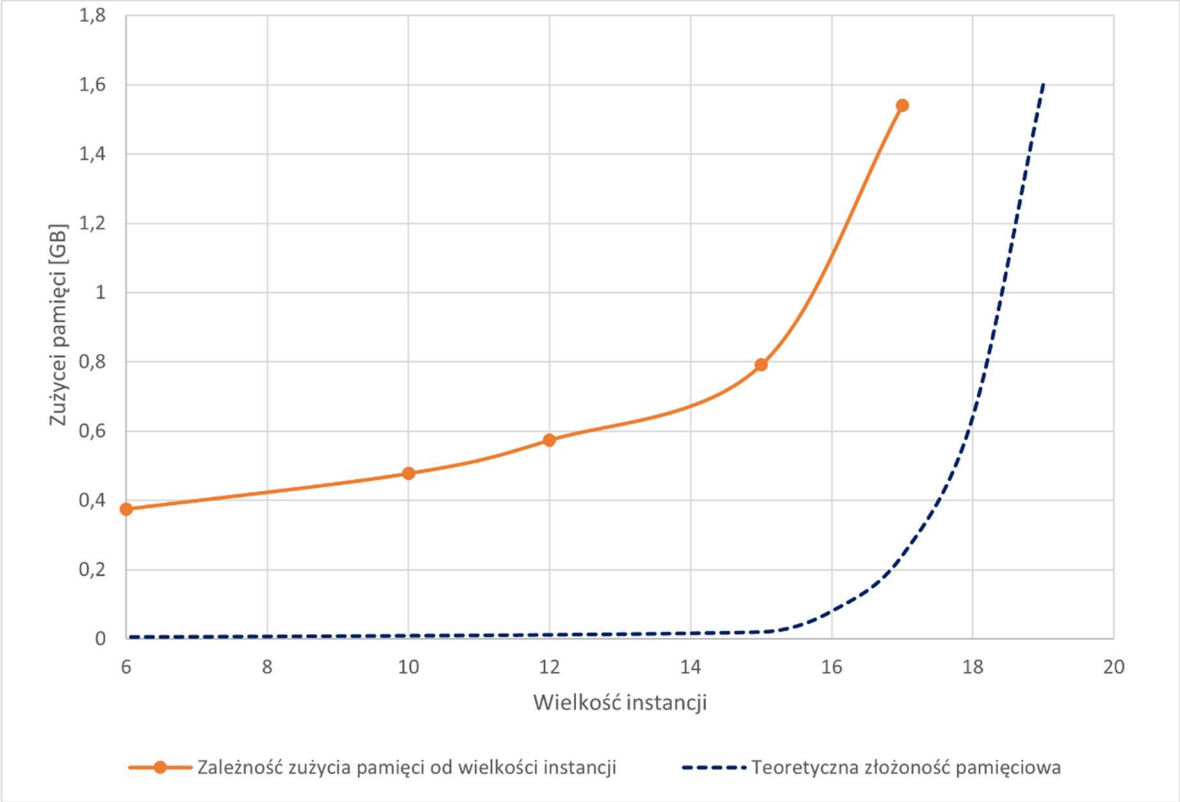
6. Wyniki

Wyniki zostały zgromadzone w pliku *heldKarp.xlsx* i opracowane w programie *MS Excel*. Przedstawione zostały w postaci wykresu zależności czasu uzyskania rozwiązania problemu od wielkości instancji (*Rysunek 6*). Pomiar czasu dla instancji o wielkości 21 z pliku *gr21.tsp* przekroczył założone 30 min. Pomiar nie został uzyskany, natomiast sugeruje duży wzrost czasu wykonywania algorytmu. Dlatego możemy przyjąć, że funkcja ma charakter wykładniczy.



Rysunek 6: Wpływ wielkości instancji n na czas uzyskania rozwiązania problemu komiwojażera dla algorytmu Helda-Karpa

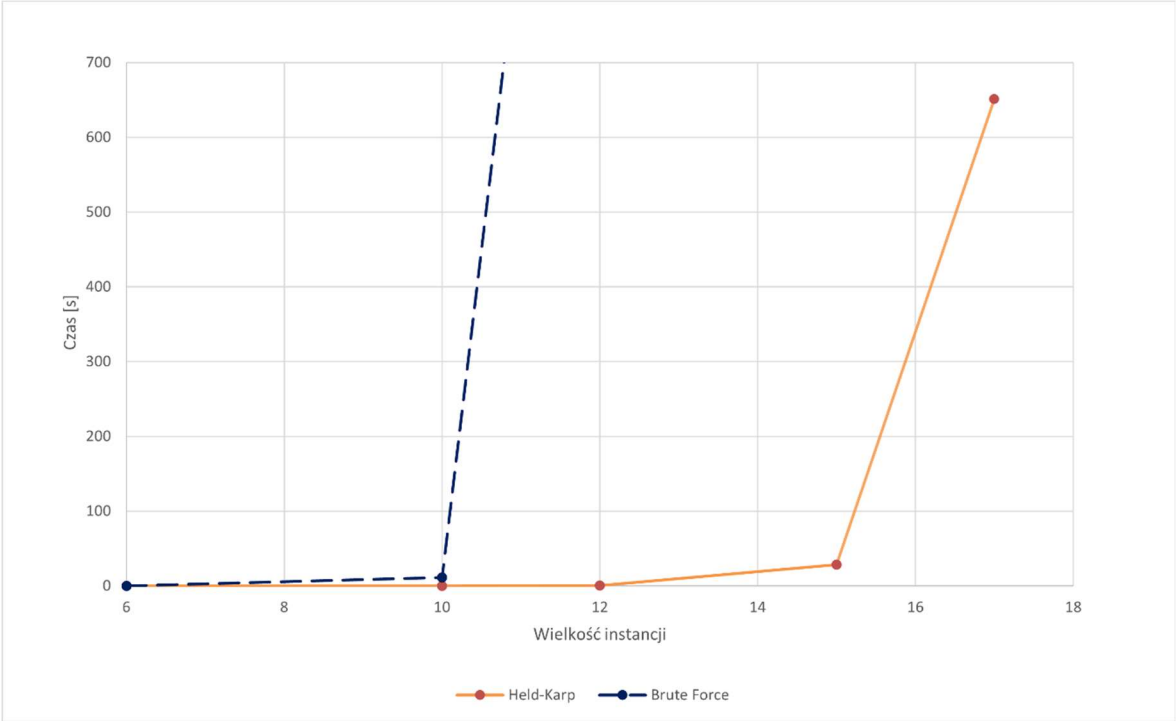
Na *Rysunku 7* zostały przedstawione wyniki badania dotyczące ilości używanej przez algorytm pamięci.



Rysunek 7: Wpływ wielkości instancji n na wykorzystanie pamięci komputera

7. Analiza wyników i wnioski

Na *Rysunku 5* została przedstawiona przybliżona krzywa przebiegu czasu wykonywania testowanego algorytmu względem wielkości instancji. Można stwierdzić, że wyniki badania potwierdzają teoretyczną złożoność czasową algorytmu Helda-Karpa, która wynosi $O(n^2 2^n)$. Zależność wykorzystania pamięci od wielkości instancji (*Rysunek 7*) ma charakter wykładniczy. W porównaniu z teoretyczną złożonością pamięciową, która wynosi $O(n2^n)$ zaobserwowany wzrost zużycia pamięci jest wolniejszy. Wynika to prawdopodobnie z przyjętego zakresu wielkości instancji, dla których zostało przeprowadzone badanie. Na *Rysunku 8* przedstawione zostało porównanie algorytmu Helda-Karpa z wcześniej badanym algorytmem Brute Force. Wynika z niego jednoznacznie, że algorytm Helda-Karpa pozwala na rozwiązanie problemu dla większych instancji w krótszym czasie.



Rysunek 8: Porównanie złożoności czasowej algorytmów: Helda-Karpa oraz Brute Force