

Projektowanie Efektywnych Algorytmów

Projekt

18/10/2022

259126 Maciej Fras

(1) Brute Force

[illegible]

1. Sformułowanie zadania

Zadanie polega na opracowaniu, implementacji i zbadaniu efektywności algorytmu przeglądu zupełnego rozwiązującego problem komiwojażera w wersji optymalizacyjnej. Problem komiwojażera (ang. Travelling salesman problem, TSP) to zagadnienie optymalizacyjne, polegające na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym, gdzie:

Cykl Hamiltona – to taki cykl w grafie, w którym każdy wierzchołek grafu odwiedzany jest dokładnie raz (oprócz pierwszego wierzchołka)

Graf Ważony Pełny – struktura składająca się ze zbioru wierzchołków oraz zbioru krawędzi. Każdej krawędzi przypisana jest pewna wartość liczbową.

Jednym z praktycznych zastosowań rozwiązania problemu komiwojażera jest wyznaczenie najkrótszej trasy pozwalającej na odwiedzenie wszystkich zadanych miast (w tym celu musimy znać odległości pomiędzy poszczególnymi miastami).



Rysunek 1: Rozwiązanie problemu komiwojażera na przykładzie wyznaczenia najkrótszej drogi łączącej 49 wybranych miast Polski

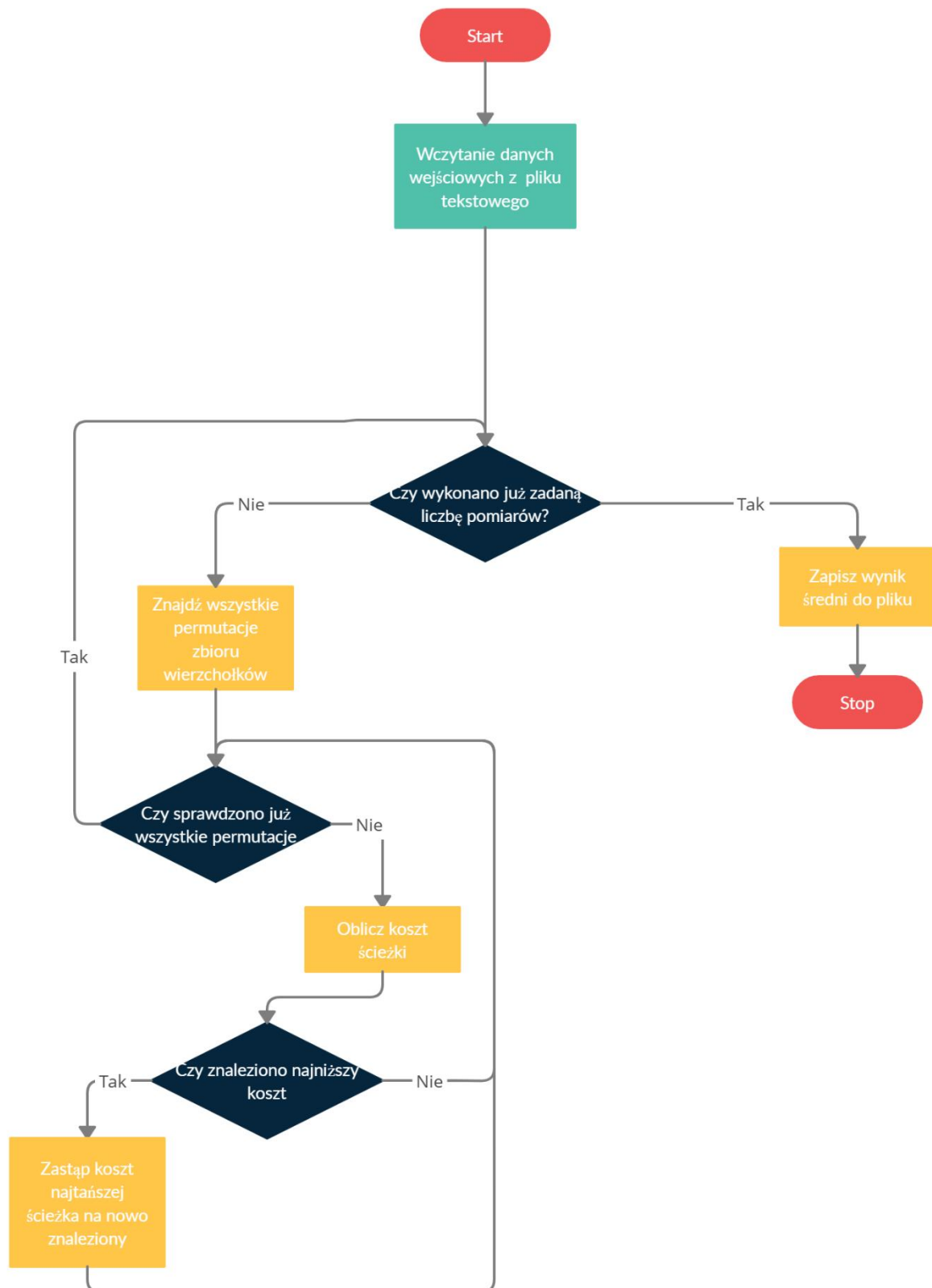
2. Opis metody

Metoda przeglądu zupełnego, tzw. przeszukiwanie wyczerpujące (eng. exhaustive search) bądź metoda siłowa (eng. brute force), polega na znalezieniu i sprawdzeniu wszystkich rozwiązań dopuszczalnych problemu, wyliczeniu dla nich wartości funkcji celu i wyborze rozwiązania o ekstremalnej wartości funkcji celu – najniższej (problem minimalizacyjny) bądź najwyższej (problem maksymalizacyjny).

W przypadku rozwiązywania problemu komiwojażera rozwiązujemy problem minimalizacyjny, ponieważ naszym celem jest znalezienie ścieżki o jak najmniejszym koszcie jej pokonania.

Metoda przeglądu zupełnego powinna za każdym razem zwrócić rozwiązanie optymalne, gdyż mamy pewność, że nie pominiemy żadnego z możliwych rozwiązań problemu. Jest to duża zaleta, jednak implikuje ona jeszcze bardziej istotną wadę tego algorytmu – bardzo wysoką złożoność czasową, która ogranicza maksymalną ilość instancji dla których można rozwiązać problem komiwojażera w rozsądnym czasie do ok. kilkunastu węzłów.

3. Opis algorytmu



Rysunek 2 - schemat blokowy algorytmu brute force rozwiązującego problem komiwojażera

4. Dane testowe

Do sprawdzenia poprawności działania algorytmu oraz do wykonania badań wybrano następujący zestaw instancji:

tsp_6_1.txt, tsp_10.txt, tsp_12.txt, tsp_13.txt [Index of /pea-stud/tsp \(pwr.wroc.pl\)](https://pwr.wroc.pl/pea-stud/tsp)

5. Procedura badawcza

Należało zbadać zależność czasu rozwiązania problemu od wielkości instancji. W przypadku algorytmu realizującego przegląd zupełny przestrzeni rozwiązań dopuszczalnych nie występowały parametry programu, które mogły mieć wpływ na czas i jakość uzyskanego wyniku. W związku z tym procedura badawcza polegała na uruchomieniu programu sterowanego plikiem konfiguracyjnym *ini.env* (plik korzysta ze zmiennych środowiskowych, które za pomocą nazwy sugerują swoje zastosowanie, są to: GRAPH_PATH – ścieżka do grafu, RESULT_PATH – ścieżka do pliku, gdzie znajdzie się wynik działania algorytmu, MEASURES_AMOUNT – liczba pomiarów które chcemy wykonać dla wskazanego grafu).

Przykładowa zawartość gotowego pliku:

```
GRAPH_PATH='tsp_6_1.txt'  
RESULT_PATH='pomiar_6_1.txt'  
MEASURES_AMOUNT=1000
```

Do pliku wyjściowego zapisywany był czas wykonania oraz otrzymane rozwiązanie (koszt ścieżki) oraz ścieżka (numery kolejnych węzłów). Plik wyjściowy zapisywany był w formacie txt. Poniżej przedstawiono fragment zawartości pliku wyjściowego dla przykładowego pomiaru.

0.4419724941253662 212 [0, 3, 4, 2, 8, 7, 6, 9, 1, 5, 0]

0.40535879135131836 212 [0, 3, 4, 2, 8, 7, 6, 9, 1, 5, 0]

0.4812455177307129 212 [0, 3, 4, 2, 8, 7, 6, 9, 1, 5, 0]

...

Uśredniony wynik pomiaru dla 10 instancji: 0.4628057241439819

Pomiar czasu został wykonany za pomocą funkcji `time()` z modułu `time` języka Python. Procedura pomiaru czasu składa się z dwukrotnego wywołania tejże funkcji – przed rozpoczęciem wykonywania algorytmu oraz zaraz po jego ukończeniu. Wynikiem pomiaru czasu jest różnica tych wywołań.

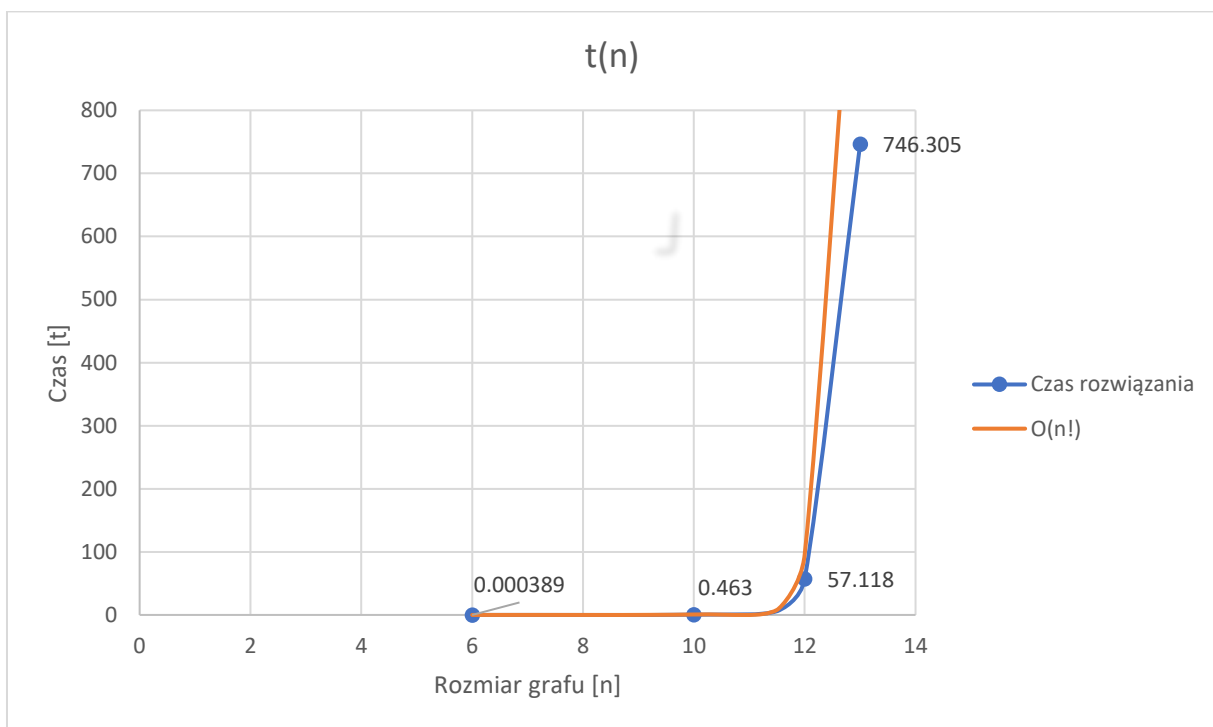
6. Wyniki

Wyniki zgromadzone zostały w plikach: pomiar_6_1.txt, pomiar_10.txt, pomiar_12.txt, pomiar_13.txt. Wszystkie ww. pliki zostały dołączone do raportu i znajdują się na dysku Google pod adresem:

<https://drive.google.com/drive/folders/1zIU4jVTnuRpuhP6YasprQ2WTyBvfFJWS?usp=sharing>

Wyniki przedstawione zostały w postaci wykresu zależności czasu uzyskania rozwiązania problemu od wielkości instancji (rysunek 2). W celu oceny poprawności wykonania algorytmu, obok wyników pomiaru naniesiony został również wykres $O(n!)$, skorygowany o współczynnik równy $2.00E-07$, tak aby móc czytelnie porównać tę zależność z otrzymanym rezultatem.

Wyniki opracowane zostały w programie MS Excel.



Rysunek 3: Wpływ wielkości instancji na czas uzyskania rozwiązania problemu komiwojażera metodą brute force oraz porównanie charakterystyki wykresu ze złożonością $O(n!)$

7. Analiza wyników i wnioski

Krzywa wzrostu czasu względem wielkości instancji ma charakter wykładniczy (rysunek 2) – obserwujemy bardzo gwałtowny wzrost czasu wykonania algorytmu przy przejściach między kolejnymi instancjami. Nałożenie krzywej $O(n!)$ potwierdza, że badany algorytm wyznacza rozwiązania problemu komiwojażera dla badanych instancji w czasie $n!$ zależnym względem wielkości instancji (obie krzywe są zgodne co do kształtu). Złożoność czasowa opracowanego algorytmu wynosi $O(n!)$