

Projektowanie Efektywnych Algorytmów

Projekt

20/10/2021

252702 Bartłomiej Sawicki

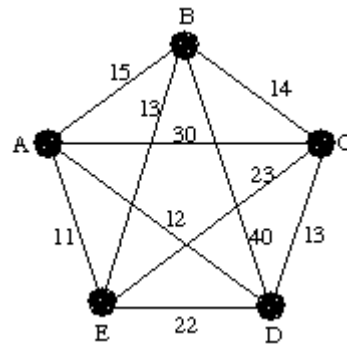
(1) Brute force

<i>Spis treści</i>	<i>strona</i>
<i>Sformułowanie zadania</i>	2
<i>Metoda</i>	3
<i>Algorytm</i>	4
<i>Dane testowe</i>	6
<i>Procedura badawcza</i>	7
<i>Wyniki</i>	8
<i>Analiza wyników i wnioski</i>	9

1. Sformułowanie zadania

Zadanie polega na opracowaniu, implementacji i zbadaniu efektywności algorytmu przeglądu zupełnego rozwiązującego problem komiwojażera w wersji optymalizacyjnej.

Problem komiwojażera polega na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. Cykl Hamiltona to taki cykl w grafie, w którym każdy wierzchołek grafu odwiedzany jest dokładnie raz (poza wierzchołkiem startowym).



Rysunek 1: Przykładowy graf pełny ważony

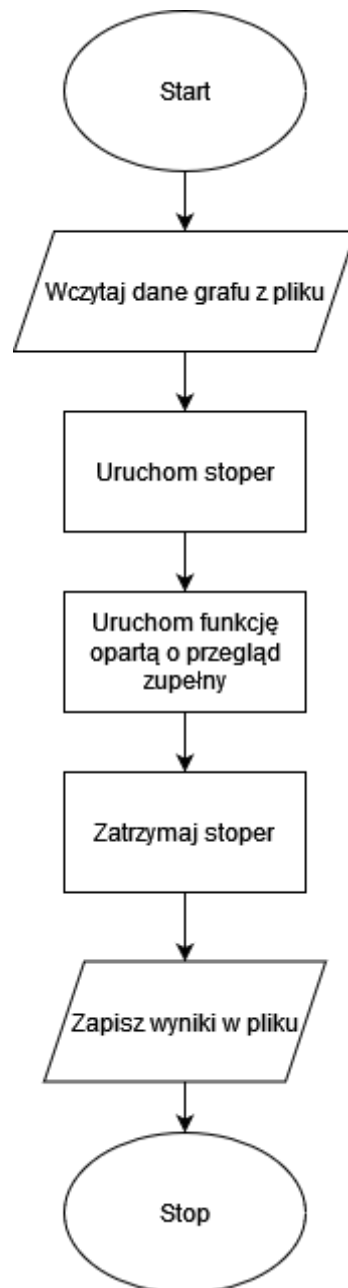
2. Metoda

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). Wadą algorytmu opartego na metodzie przeglądu zupełnego jest duża złożoność obliczeniowa natomiast zaletą jest łatwość implementacji.

Metoda przeglądu zupełnego w kontekście problemu komiwojażera sprowadza się do sprawdzenia wszystkich ścieżek, które spełniają kryteria problemu. Następnie ze zbioru znalezionych ścieżek należy wybrać taką, która ma najmniejszą sumę wag.

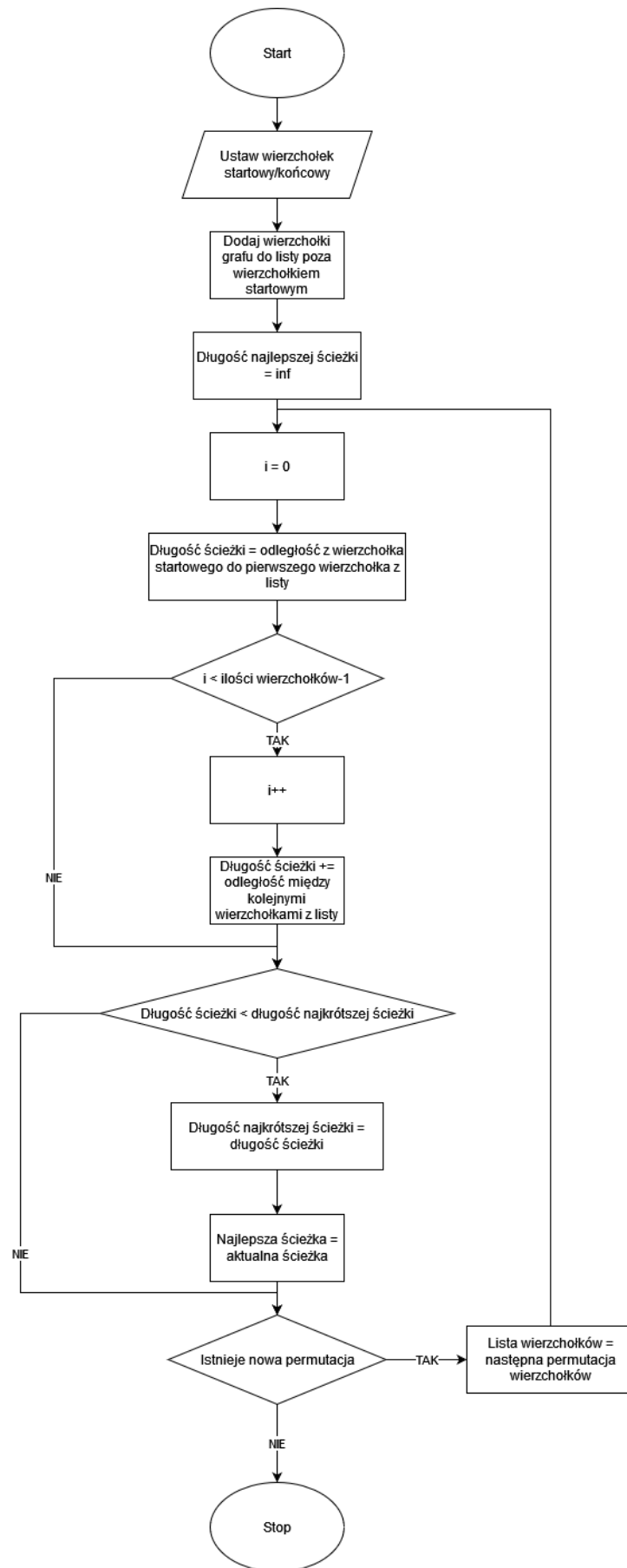
3. Algorytm

Informacje o grafie wczytywane są z pliku tekstowego. Po wczytaniu danych program tworzy macierz wag grafu.



Rysunek 2: Poglądowy schemat blokowy

Właściwy algorytm polega na przygotowaniu listy wierzchołków grafu oraz wybraniu wierzchołka początkowego. Dopóki nie zostały sprawdzone wszystkie permutacje listy wierzchołków porównywana jest aktualna permutacja listy z najlepszą (taka, w której suma wag ścieżki jest minimalna) uzyskaną do tej pory.



Rysunek 3: Schemat blokowy właściwej części algorytmu

4. Dane testowe

Do sprawdzenia poprawności działania wybrano następujący zestaw instancji:

- 1) tsp_6_1.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
- 2) tsp_6_2.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
- 3) tsp_10.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
- 4) tsp_12.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
- 5) tsp_13.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
- 6) tsp_14.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
- 7) tsp_15.txt <http://jaroslaw.mierzwa.staff.iiar.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 inicjującym .INI (format pliku: nazwa_instancji liczba_wykonań rozwiązanie_optymalne [ścieżka optymalna]).

Wyniki działania programu zostały zapisane w pliku .csv

Pomiar czasu wykonano zgodnie z instrukcją zawartą na stronie:
http://staff.iiar.pwr.wroc.pl/antoni.sterna/sdizo/SDiZO_time.pdf

Procedura mierzenia czasu polega na pobraniu wartości odpowiedniego licznika przed i po wywołaniu funkcji rozwiązującej problem komiwożacza. Do uzyskania pomiaru w jednostkach czasu potrzebna jest częstotliwość impulsów. Rezultat końcowy jest uzyskany przez różnicę między stanami liczników przed i po uruchomieniu funkcji podzieloną przez częstotliwość impulsów.

Specyfikacja sprzętu, na którym przeprowadzono badania:

Wersja systemu Windows

Windows 10 Pro

© Microsoft Corporation. Wszelkie prawa zastrzeżone.

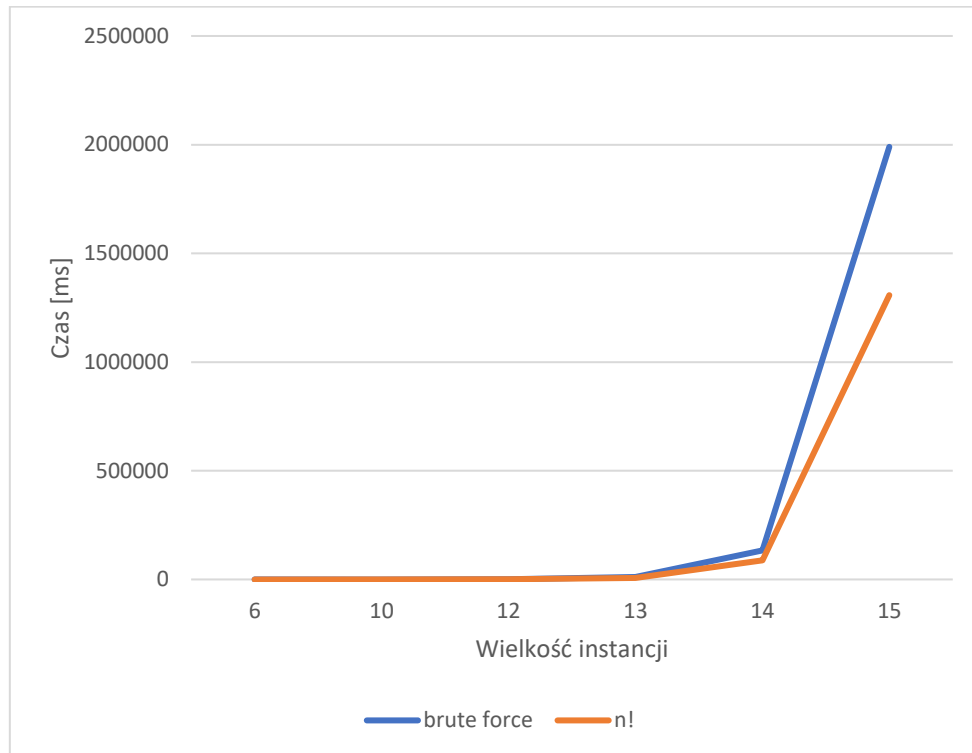


System

Procesor:	Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz 3.20 GHz
Zainstalowana pamięć (RAM):	8,00 GB
Typ systemu:	64-bitowy system operacyjny, procesor x64

6. Wyniki

Wyniki zostały zgromadzone w pliku wyniki.csv oraz opracowane przy pomocy programu MS Excel. Dane przedstawione zostały w postaci wykresu zależności czasu od wielkości instancji problemu.



Rysunek 4: Wykres zależności czasu od liczby wierzchołków grafu

7. Analiza wyników i wnioski

Krzywa wzrostu czasu względem wielkości instancji ma charakter wykładniczy (rysunek 4). 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 $O(n!)$. Ze względu na wykładniczą złożoność algorytmu maksymalna instancja grafu jaką udało mi się rozwiązać miała 14 wierzchołków.