

Projektowanie Efektywnych Algorytmów

Projekt

17/11/2021

252702 Bartłomiej Sawicki

(2) Held-Karp

Spis treści	Strona
Sformułowanie zadania	2
Metoda	3
Algorytm	4
Dane testowe	6
Procedura badawcza	7
Wyniki	8
Analiza wyników i wnioski	10

1 Sformułowanie zadania

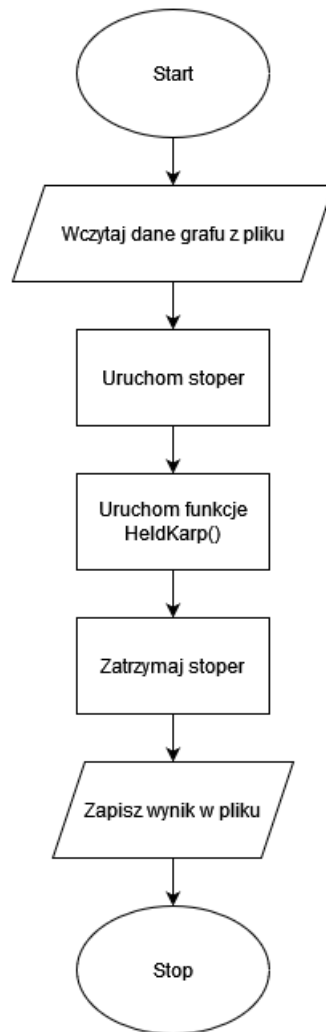
Zadanie polega na opracowaniu, implementacji i zbadaniu efektywności algorytmu Held- Karp rozwiązującego problem komiwojażera w wersji optymalizacyjnej. Dodatkowo należy znaleźć „granice możliwości” zaimplementowanego programu czyli sprawdzić dla jakiej instancji problemu program przestanie działać ze względu na brak dostępnej pamięci lub będzie wykonywał się zbyt długo.

2 Metoda

Metoda programowania dynamicznego (*eng. dynamic programming*), opiera się na podziale rozwiązywanego problemu na podproblemy względem kilku parametrów. Metodę programowania dynamicznego można stosować w przypadkach gdy podproblemy nie są zależne. Przy rozwiązywaniu problemu tą metodą dany podproblem rozwiązuje się tylko raz co prowadzi do wielomianowej liczby rozwiązań. W przypadku metody przeglądu zupełnego ilość rozwiązywanych problemów była ponad wielomianowa.

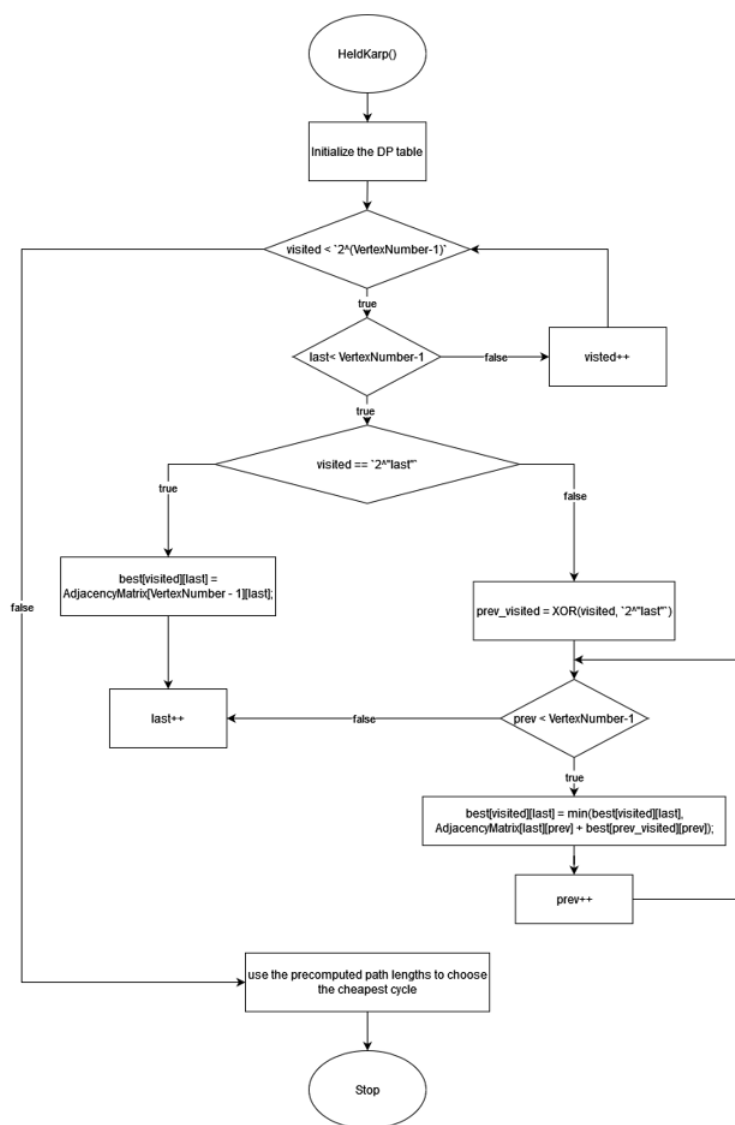
3 Algorytm

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



Rysunek 1: Poglądowy schemat blokowy

Właściwy algorytm jest przedstawiony na poniższym schemacie blokowym



Rysunek 2: Schemat blokowy właściwej część algorytmu

4 Dane testowe

Do sprawdzenia poprawności działania wybrano następujący zestaw instancji:

1. tsp_6_1.txt
2. tsp_6_2.txt
3. tsp_10.txt
4. tsp_12.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>

Do wykonania badań wybrano następujący zestaw instancji:

1. tsp_6_1.txt
2. tsp_6_2.txt
3. tsp_10.txt
4. tsp_12.txt
5. tsp_13.txt
6. tsp_14.txt
7. tsp_15.txt
8. tsp_17.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
9. gr_21.txt
10. gr_24.txt
11. bays_29.txt <http://jaroslaw.rudy.staff.iiar.pwr.wroc.pl/pea.php>

5 Procedura badawcza

Należało zbadać zależność czasu rozwiązania problemu od wielkości instancji. W przypadku algorytmu realizującego programowanie dynamiczne 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. 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

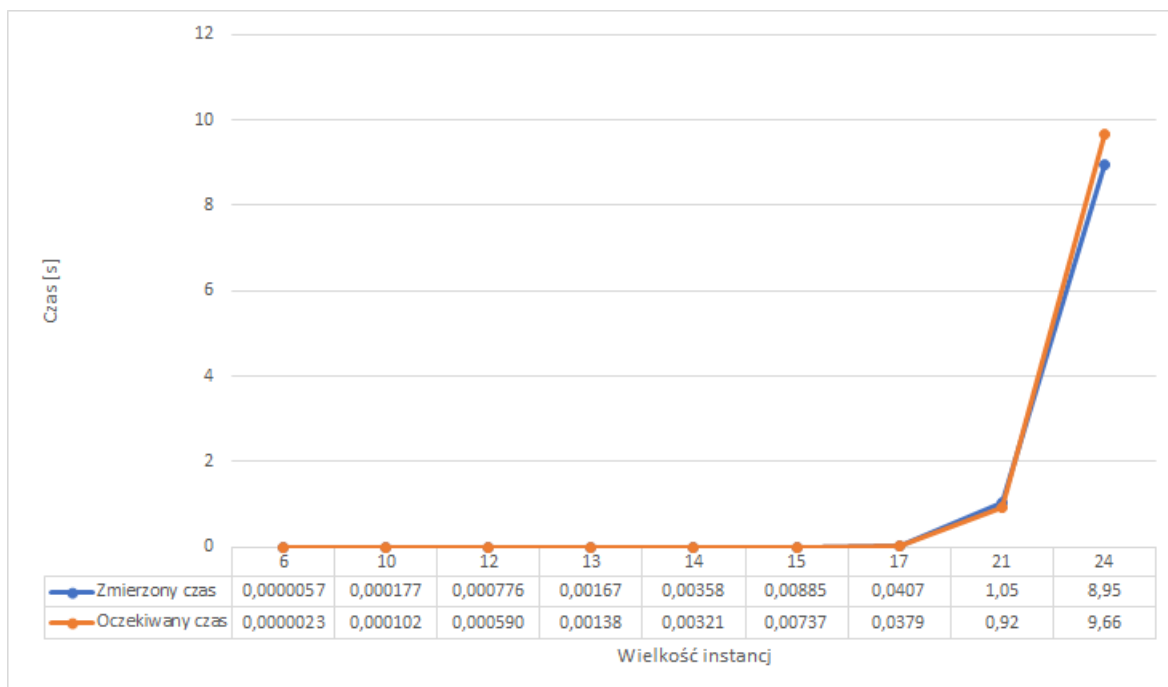
Procedura mierzenia czasu polega na pobraniu wartości odpowiedniego licznika przed i po wywołaniu funkcji rozwiązującej problem komiwojażera. Do uzyskania pomiaru w jednostkach czasu potrzebna jest częstotliwość impulsów. Rezultat końcowy jest uzyskany przez różnicę między stanami liczników przed i po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



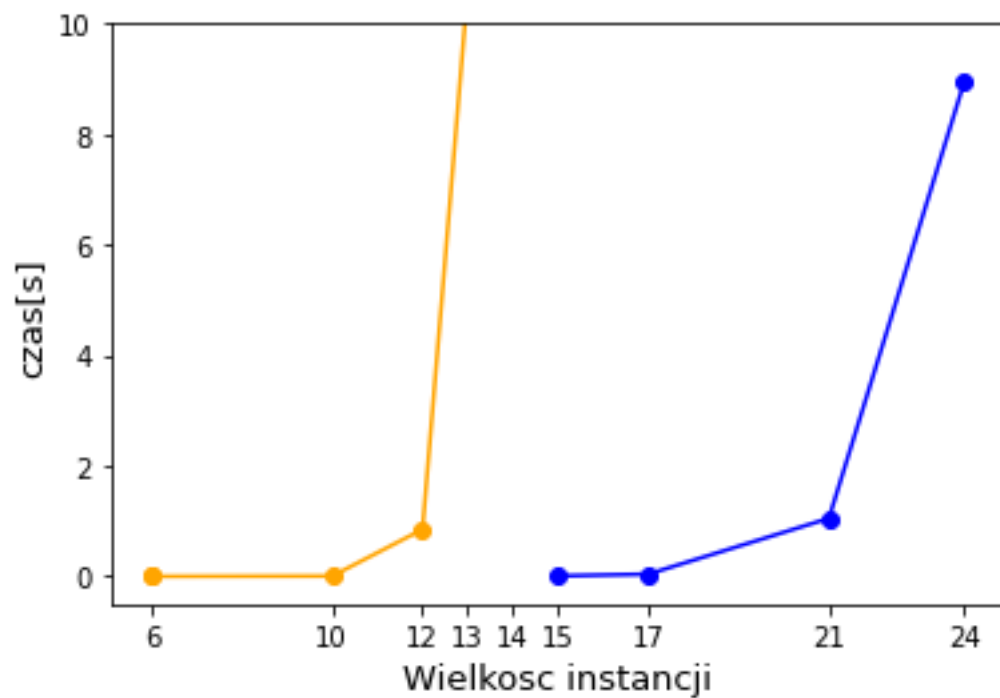
Rysunek 3: Specyfikacja sprzętu, na którym przeprowadzono badania

6 Wyniki

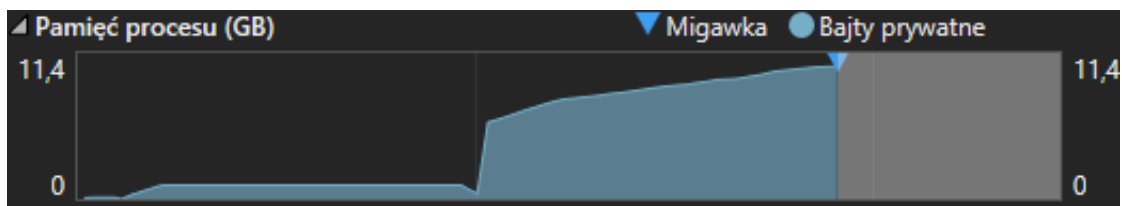
Wyniki zostały zgromadzone w pliku wyniki.csv oraz opracowane przy pomocy programu MS Excel. Dane przedstawione zostały w postaci wykresu zależności czasu od wielkości instancji problemu.



Rysunek 4: Wykres zależności czasu od liczby wierzchołków grafu



Rysunek 5: Porównanie czasu działania algorytmu oprtego o metodę przeglądu zupełnego(pomarańczowa) i algorytmu Helda-Karpa (niebieska)



Rysunek 6: Wykres zużycia pamięci procesu od czasu

7 Analiza wyników i wnioski

Krzywa wzrostu czasu(niebieska) względem wielkości instacji ma charakter wykładniczy (Rysunek 4). Nałożenie krzywej opisanej równaniem $f(x) = x^2 * 2^x$ (linia pomarańczowa) potwierdza, że zaimplementowany algorytm ma złożoność czasową $O(x^2 * 2^x)$. Algorytm Helda-Karpa dla większości badanych instancji (poza tsp61 oraz tsp62) znajdował optymalną ścieżkę w czasie znacznie krótszym niż algorytm oparty na metodzie przeglądu zupełnego (Rysunek 5).

Pomimo faktu, że algorytm Helda-Karpa jest w stanie rozwiązać problem komiwojażera dla instacji większych niż algorytm oparty o metodę przeglądu zupełnego, nie był w stanie rozwiązać instacji większej niż gr26. Przy próbie uruchomienia programu dla instacji bays29 zużycie pamięci procesu bardzo szybko przekroczyło 8GB (Rysunek 6) co spowodowało niewłaściwą pracę komputera.