

Projektowanie Efektywnych Algorytmów

Projekt

17/11/2021

252702 Bartłomiej Sawicki

(3) Branch & Bound

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 opartego o metodę podziału i ograniczeń (*Branch & Bound*) 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

Działanie metody podziału i ograniczeń opiera się o analizę drzewa przestrzeni rozwiązań. W przypadku problemu komiwojażera drzewo to zawiera możliwe ścieżki. Przeglądanie drzewa rozpoczyna się od korzenia a następnie przechodzi się do kolejnych liści tworząc rozwiązanie. Przechodzenie całego drzewa rozwiązań jest zbyt kosztowne czasowo i pamięciowo dlatego dla każdego węzła oblicza się granicę. Obliczona granica pozwala ustalić, który węzeł jest obiecujący i umożliwi szybkie rozwiązanie zadania.

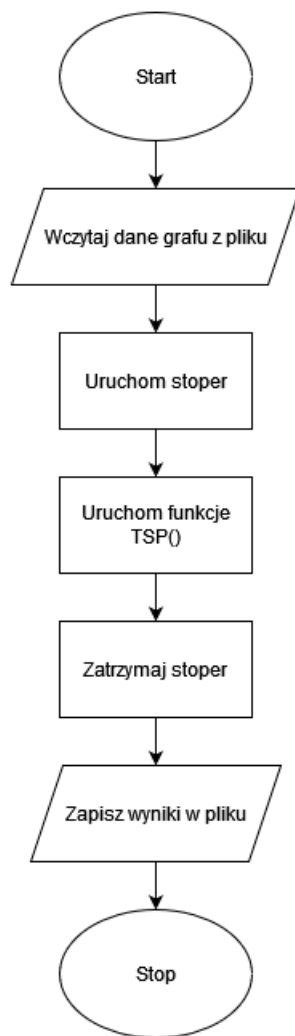
Przy przeszukiwaniu drzewa przestrzeni rozwiązań należy wybrać strategię odwiedzania kolejnych wierzchołków. Na potrzeby tego projektu wyróżniamy następujące strategie:

- Best search
- Breadth search
- Depth search

Do implementacji algorytmu wykozystałem tylko strategię *Best search*. Strategia ta polega na wybieraniu wierzchołka, do którego ścieżka prowadząca ma najmniejszą wagę.

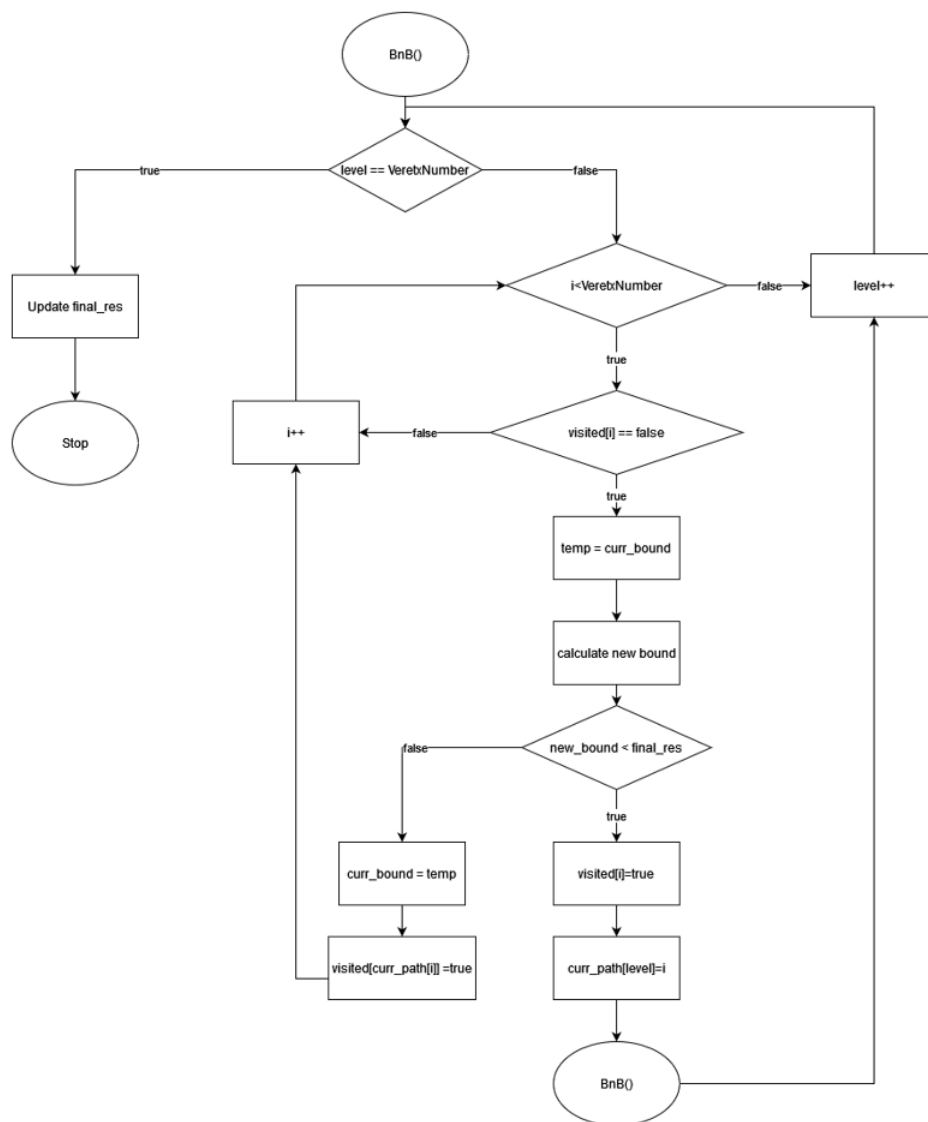
3 Algorytm

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 części algorytmu

4 Dane testowe

Do sprawdzenia poprawności działania wybrano następujący zestaw instancji:

1. tsp_6_1.txt
2. tsp_10.txt
3. 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_10.txt
3. tsp_12.txt
4. tsp_13.txt
5. tsp_14.txt
6. tsp_15.txt <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>
7. gr_17.txt
8. gr_21.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 opartego o metodę podziału i ograniczeń 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

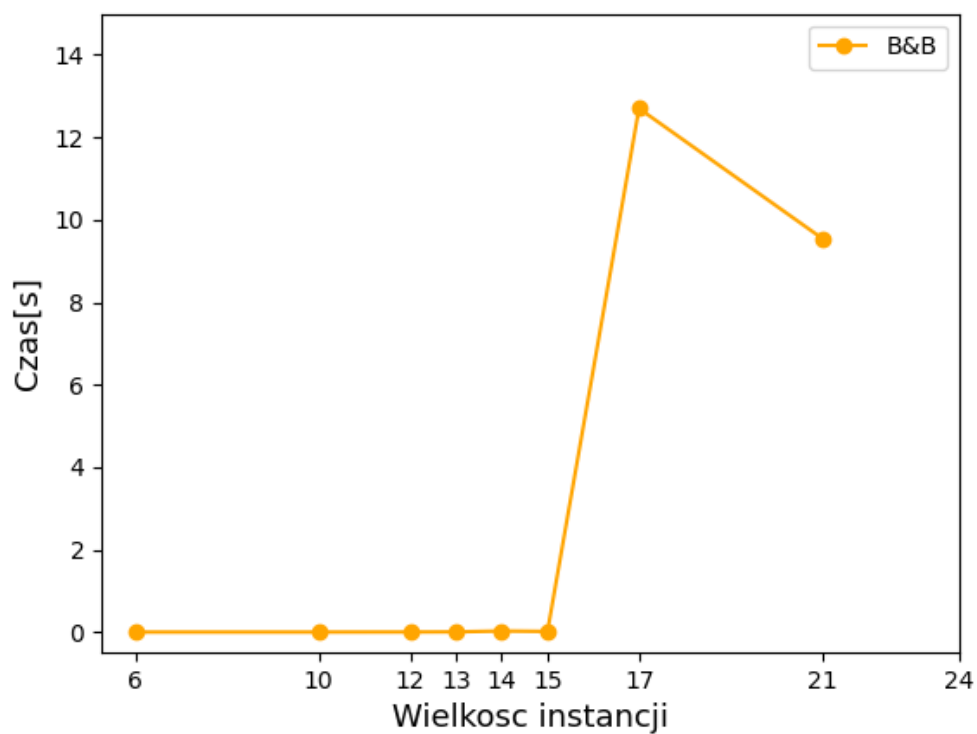
Procedura mierzenia czasu polega na pobraniu wartości odpowiedniego licznika przed i po wywołaniu funkcji rozwiązującej problem komiwojażera. Do uzyskania pomiaru w jednostkach czasu potrzebna jest częstotliwość impulsów. Rezultat końcowy jest uzyskany przez różnicę między stanami liczników przed i po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



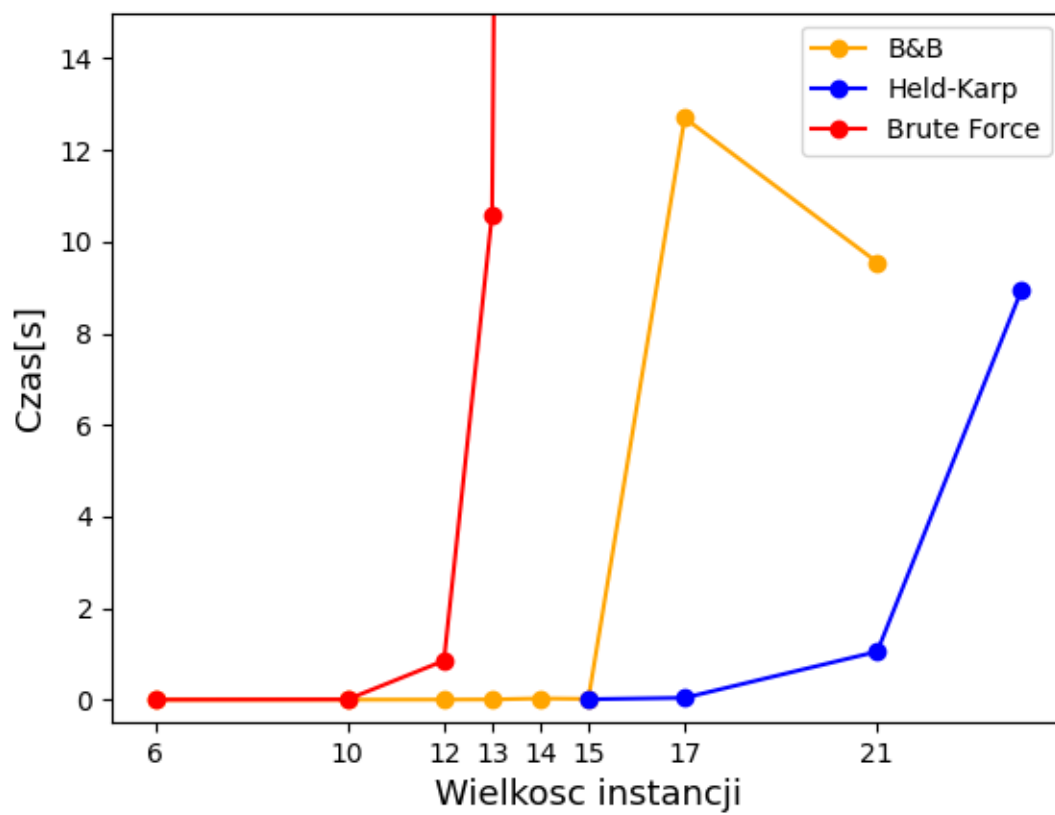
Rysunek 3: Specyfikacja sprzętu, na którym przeprowadzono badania

6 Wyniki

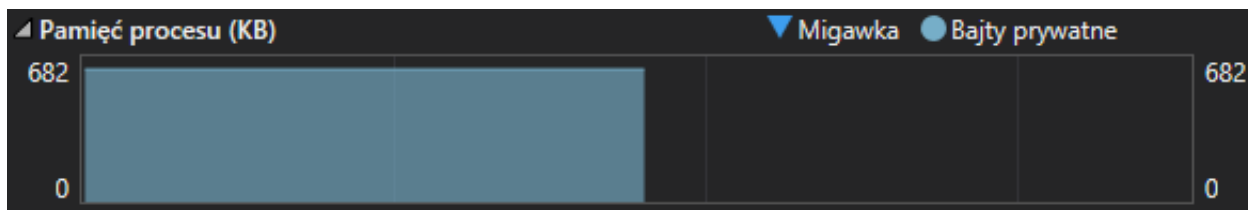
Wyniki zostały zgromadzone w pliku wyniki.csv oraz opracowane przy pomocy programu MS Excel. Dane przedstawione zostały w postaci wykresu zależności czasu od wielkości instancji problemu.



Rysunek 4: Wykres zależności czasu od liczby wierzchołków grafu



Rysunek 5: Porównanie czasu działania algorytmów w zależności od wielkości instancji



Rysunek 6: Wykres zużycia pamięci procesu od czasu

7 Analiza wyników i wnioski

Krzywa wzrostu czasu (pomarańczowa) względem wielkości instancji ma charakter trudny do ustalenia (Rysunek 4). Wynika to z faktu, że algorytm oparty o metodę podziału i ograniczeń może przyjmować złożoność czasową od $O(n)$ do $O(n!)$ gdzie n to ilość wierzchołków rozpatrywanego grafu. Złożoność $O(n)$ można otrzymać gdy w rozpatrywanym drzewie od razu napotkamy ścieżkę której koszt jest mniejszy niż koszt dojścia do innego wierzchołka na pierwszym poziomie. Najgorszy możliwy scenariusz przy rozwiązywaniu problemu komiwojażera tą metodą to sprawdzenie wszystkich ścieżek w drzewie rozwiązań (przegląd zupełny).

Na podstawie porównania zrealizowanych algorytmów (Rysunek 5) widać jak dużą przewagę ma algorytm oparty o programowanie dynamiczne nad metodą siłową. Niska złożoność obliczeniowa powoduje w tym przypadku bardzo duże zużycie pamięci. Na wykresie widać również przewagę algorytmu opartego na metodzie podziału i ograniczeń nad metodą siłową. Niestety metoda podziału i ograniczeń nie daje gwarancji stałej złożoności obliczeniowej. Czas rozwiązania problemu komiwojażera tą metodą nie zależy tylko od wielkości instancji ale również od tego w jaki sposób ułoży się drzewo rozwiązań. Na Rysunku 4 wyraźnie widać, że problem dla 21 wierzchołków grafu został rozwiązany szybciej niż problem dla 17 wierzchołków.