Paweł Biel

nr indeksu 225949

Sprawozdanie

Problem komiwojażera metodą podziału i ograniczeń

1. Informacje teoretyczne

Problem klasy NP jest problemem, dla którego rozwiązanie można zweryfikować w czasie wielomianowym – natomiast samo znalezienie rozwiązania problemu prawie na pewno wymaga czasu ponad wielomianowego – nie znaleziono dotąd algorytmu o wielomianowej złożoności czasowej. Dla rozwiązania problemów klasy NP można zastosować przegląd zupełny – otrzymamy wtedy dokładne rozwiązanie, jednak ze względu na bardzo dużą złożoność czasową, nadaje się on jedynie dla bardzo małych rozmiarów problemu. Dla nieco większych instancji problemów również możemy otrzymać dokładny wynik stosując algorytm oparty na programowaniu dynamicznym lub metodzie podziału i ograniczeń (branch and bound). Do implementacji w ramach projektu wybrałem metodę podziału i ograniczeń dla problemu komiwojażera.

2. Problem komiwojażera

Zagadnienie optymalizacyjne, polegające na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. Często można się spotkać że problem jest dedykowany dla miast o odległość między tymi miastami to koszt podróży czy czasu, w grafie pełnym ten koszt odpowiada krawędziom. Problem komiwojażera można rozdzielić na dwa przypadki asymetryczny i symetryczny. Ten pierwszy polega na tym, że np. z miasta A do miasta B mamy 2 drogi jednostronne o różnych kosztach przejazdu (mogą być też takie same), natomiast symetryczny polega na tym że droga jest jedna w jedną stronę z jednego do drugiego miasta. Złożoność obliczeniowa:

- złożoność czasowa algorytmu wykorzystującego metodę podziału i ograniczeń jest bardzo mocno zależna od danych; w pesymistycznym przypadku wynosi O(n!) czyli tyle samo co złożoność przeglądu zupełnego, gdzie n – to ilość miast, lecz jest wysoka szansa na znalezienie rozwiązania szybciej[1].

[1] http://www.geeksforgeeks.org/branch-bound-set-5-traveling-salesman-problem/

3. Opis algorytmu

W moim programie zaimplementowałem jest algorytm Little'a oparty na metodzie podziału i ograniczeń, stosuję strategię przeszukiwania w głąb.

W pierwszej kolejności zostaje wyliczone górne ograniczenie na zasadzie połączenia drogi kolejno z pierwszego do ostatniego miasta łącznie z powrotem do pierwszego. Następnie algorytm rekurencyjnie wyszukuje rozwiązanie, jeśli to rozwiązanie jest lepsze od górnego ograniczenia to górne ograniczenie jest równe temu rozwiązaniu. Następnie nasze nowe ograniczenie górne jest porównywane po kolei wchodząc „ w górę" drzewa z innymi dolnymi granicami dolnymi , jeśli istnieje szansa że inna droga może dać lepszy wynik to wykonywana jest procedura znalezienia wyniku dla tej drogi. Jeśli dana droga po paru ruchach staje się nie korzystna algorytm rezygnuje z tej drogi.

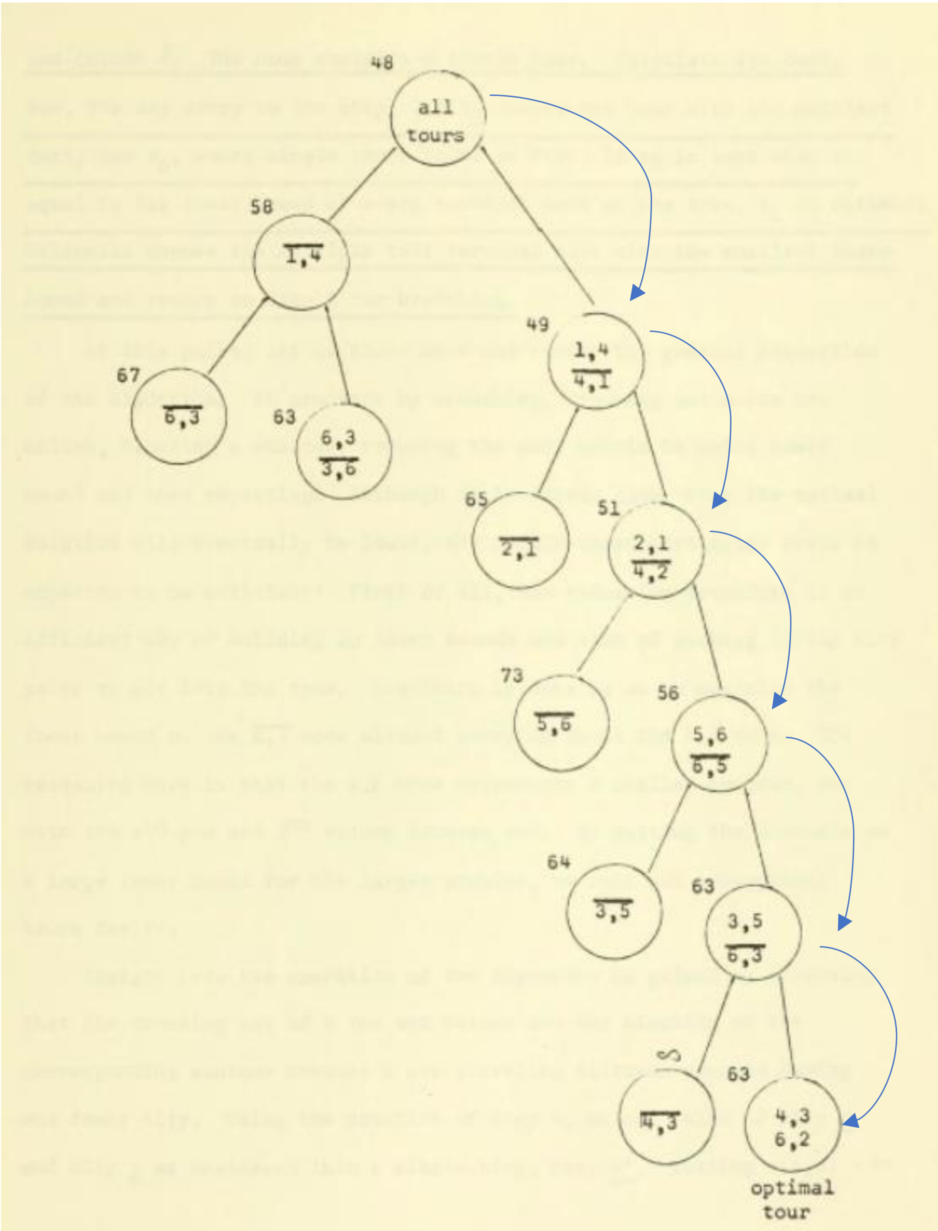
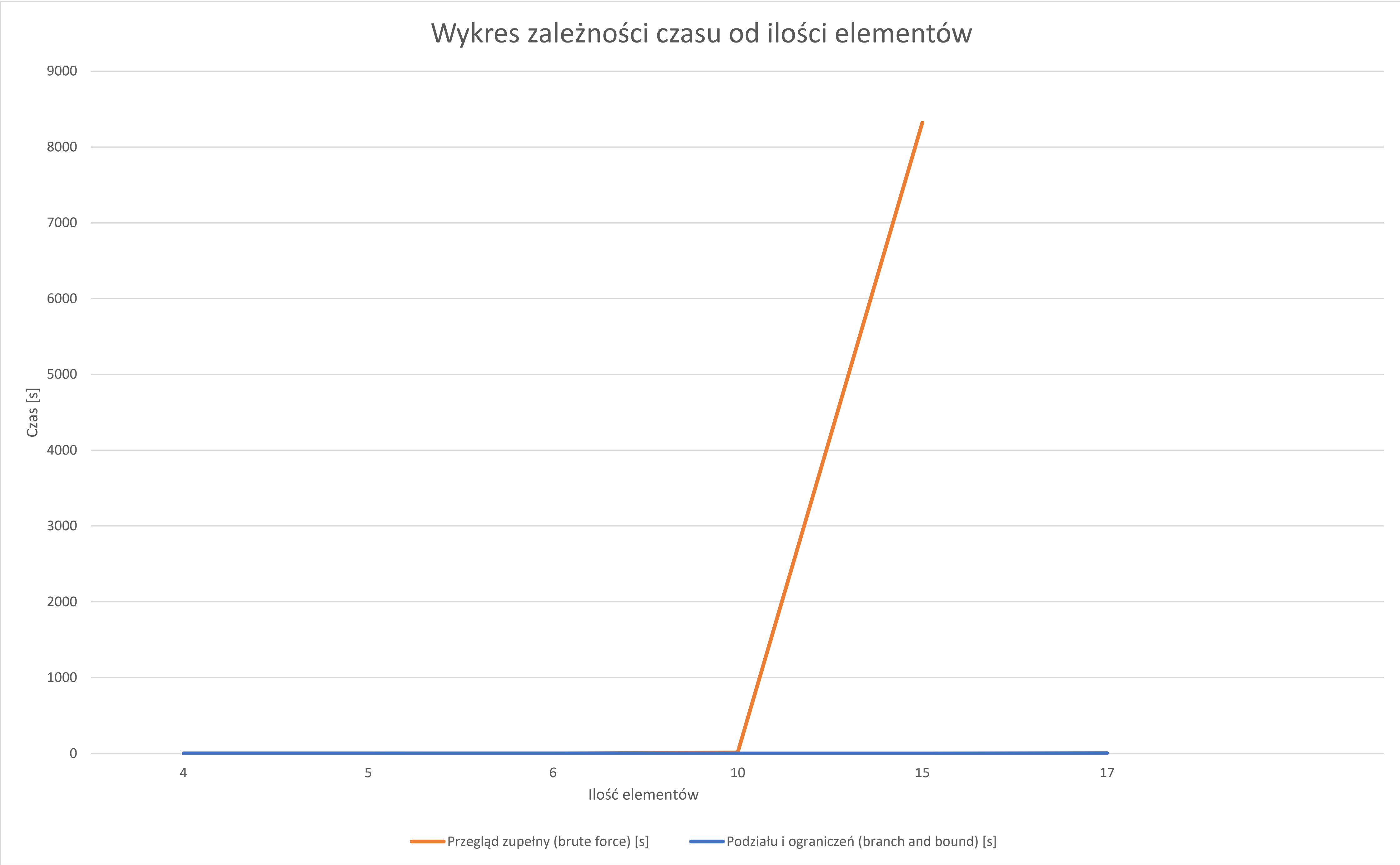


Figure 5. Final Tree.

Rysunek 1


Wykres zależności czasu od ilości elementów
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Czas [s]
4
5
6
10
15
17
Ilość elementów
Przegląd zupełny (brute force) [s]
Podziału i ograniczeń (branch and bound) [s]

4. Pomiary czasu

Procedura badawcza polega na wielokrotnym uruchomieniu problemu dla wybranego zestawu danych oraz pobrania czasu dla każdego wykonania. Następnie z otrzymanych czasów liczę średnią arytmetyczną i jest to średni czas wykonania problemu dla wybranego zestawu danych

n	Przegląd zupełny (brute force) [s]	Podziału i ograniczeń (branch and bound) [s]
4	0,000242	0,000986
5	0,000423	0,001042
6	0,0017	0,001483
10	10,776357	0,003619
15	8324,0023	0,006063
17	-	2,693156

5. Wnioski

Czasy wykonywania algorytmu przeglądu zupełnego dla instancji mniejszych od 6 jest porównywalny z czasem metody podziału i ograniczeń, natomiast można zauważyć że bardzo szybko rośnie wraz ze wzrostem ilości elementów, przegląd zupełny dla instancji o wielkości 17 został przerwany po 5h, z racji tego że 5h to nie jest rozsądny czas wykonywania takiego algorytmu.

Czas wykonania algorytmu opartego na metodzie podziału i ograniczeń dla tych samych rozmiarów problemu jest wyraźnie mniejszy, ale bardzo mocno zależy rodzaju i wielkości instancji. Istnieją bardzo duże szanse na zakończenie algorytmu szybciej, ponieważ algorytm przeszukuje tylko te gałęzie, które dają szanse na znalezienie lepszego wyniku. Można wyraźnie stwierdzić że jest to metoda, która jest w stanie rozwiązać dany problem w czasie rozsądnym oraz jest metodą, które jest lepsza pod względem pamięci. Na podstawie zebranych danych oraz wykresu można odczytać, że złożoność dla tych danych testowych jest O(logn),lecz w najgorszym przypadku złożoność może dojść do złożoności wykładniczej.

Nie udało się wykonać pomiarów dla macierzy symetrycznych, ponieważ algorytm zwracał złe wyniki, dlatego też nie zostały one umieszczone w sprawozdaniu.