Projektowanie efektywnych algorytmów

**Projekt 1**

**Implementacja algorytmu podziału i ograniczeń**

**dla problemu komiwojażera**

wt 7:30

Prowadzący: Dr inż. Łukasz Jeleń

Autor: Jakub Sanecki

1. Teoria

Problem komiwojażera polega na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. jest problemem typu NP-zupełnego, jeżeli mówimy o czasie znalezienia najkrótszej trasy pomiędzy miastami.

Metoda podziału i ograniczeń służy właśnie do rozwiązywania problemów optymalizacyjnych. Polega ona na analizie drzewa przestrzeni stanów.

Drzewo to przedstawia możliwe ścieżki, w których może poruszać się algorytm rozwiązując dany problem. Algorytm idzie od korzenia po liście w każdym węźle obliczając ograniczenie, które mówi nam, czy rozwiązanie jest obiecujące, czy też nie. Tylko rozwiązania obiecujące są dalej sprawdzane, aby nie tworzyć przeglądu zupełnego.

Metoda podziału i ograniczeń jest zależna od danych wejściowych, dlatego może się zdarzyć, iż wystąpi dla niej przegląd zupełny – kolejne sprawdzane rozwiązanie będzie uznawane za obiecujące.

1. Algorytm.

Zdecydowałem się na użycie metody podziału i ograniczeń „Najpierw Najlepszy”. Opisana szerzej w opracowaniu Mateusza Łyczka (patrz. Bibliografia). Bazuje ona na kolejce priorytetowej i na niejawnie tworzonym drzewie przestrzeni stanów.

Poruszając się od korzenia, dla każdego z dzieci węzła, z których analizujemy możliwe rozwiązania, którymi są wierzchołki dodawane do trasy, liczymy dolne ograniczenie czy jest lepsze od dotychczasowego. Analizujemy dalej dla kolejnych dzieci. Pseudokod znajduje się w pierwszej pozycji bibliografii.

1. Implementacja.

Graf Hamiltona jest zaimplementowany za pomocą klasy Graph w reprezentacji listowej. Kolejka priorytetowa jest stworzona z wykorzystaniem biblioteki <algorithm>. Typ danych kolejki i kontener danych ma typ vector. Niejawnie stworzone drzewo również jest vectorem.

1. Wyniki.

Testy były przeprowadzone dla losowych ilości miast wraz z losowymi wagami. Wyniki są średnią z 10 pomiarów dla każdej liczby miast, natomiast wagi były wybierane losowo. Pomiary zostały przeprowadzone na komputerze MacBook Pro (Late 2016), procesor 2GHz, Intel Core i5 z pamięcią 8GB. Użyte zostało IDE XCode 9.

|  |  |
| --- | --- |
| Liczba miast | Czas wykonania algorytmu [s] |
| 5 | 0.0000057 |
| 10 | 0.009396 |
| 20 | 4.3421 |
| 25 | 90.9447 |
| 30 | 650.276 |

1. Wnioski.

Podczas testów z losowymi danymi widoczne było, że program dla tej samej ilości miast i losowych danych przyjmuje bardzo różniące się wyniki. Przy próbie wykonania algorytmu dla tych samych danych, czas był znacznie krótszy, co jest zapewne skutkiem zapisywania poprzednich wyników w pamięci. Można powiedzieć, że z praktycznego punktu widzenia czas wykonania dla 20 miast jest jeszcze akceptowalny przy wykorzystywaniu tej implementacji, natomiast powyżej, na wyniki czeka się zbyt długo.

1. Bibliografia.

* <http://www.ii.uni.wroc.pl/~prz/2011lato/ah/opracowania/met_podz_ogr.opr.pdf>
* <https://pl.wikipedia.org/wiki/Problem_komiwojażera>