Sabina Miśkiewicz200577

Wt 18.00

**Sprawozdanie z laboratorium projektowanie algorytmów i metod sztucznej inteligencji**

**Algorytmy wyznaczania najkrótszej ścieżki**

Doświadczenia zostały przeprowadzone na komputerze o następujących parametrach:

* System: 64-bitowy Windows 7
* Procesor: Pentium® Dual-Core CPU T4500 @ 2.30GHz 2.30GHz
* Pamięć RAM: 3,00 GB

**Omówienie algorytmów:**

**Dijkstry**

Algorytm jest oparty o strategię zachłanną. Służy do wyznaczania najmniejszej odległości od ustalonego wierzchołka  do wszystkich pozostałych w skierowanym [grafie](http://www.algorytm.org/klasyczne/grafy-i-ich-reprezentacje.html), w odróżnieniu jednak od [Algorytmu Forda-Bellmana](http://www.algorytm.org/algorytmy-grafowe/algorytm-forda-bellmana.html), graf wejściowy nie może zawierać krawędzi o ujemnych wagach. Co jest ograniczaniem.

W algorytmie tym pamiętany jest zbiór Q wierzchołków, dla których nie obliczono jeszcze najkrótszych ścieżek, oraz wektor D[i] odległości od wierzchołka *startowego* do *i*. Na samym początku zbiór Q zawiera wszystkie wierzchołki a wektor D jest pierwszym wierszem macierzy wag krawędzi.

**Bellmana Forda**

Algorytm rozwiązuje [problem najkrótszej ścieżki](http://pl.wikipedia.org/wiki/Problem_najkr%C3%B3tszej_%C5%9Bcie%C5%BCki). Pozwala znaleźć ścieżkę o najmniejszej wadze pomiędzy dwoma [wierzchołkami](http://pl.wikipedia.org/wiki/Wierzcho%C5%82ek) w [grafie](http://pl.wikipedia.org/wiki/Graf_(matematyka)) wag. Idea [algorytmu](http://pl.wikipedia.org/wiki/Algorytm) opiera się na [metodzie relaksacji](http://pl.wikipedia.org/wiki/Metoda_relaksacji) (dokładniej następuje relaksacja *V* − 1 razy każdej z [krawędzi](http://pl.wikipedia.org/wiki/Kraw%C4%99d%C5%BA_grafu)).

W odróżnieniu od [algorytmu Dijkstry](http://pl.wikipedia.org/wiki/Algorytm_Dijkstry), poprawność algorytmu Bellmana-Forda nie opiera się na założeniu, że wagi w grafie są nieujemne. Za tę ogólność płaci się jednak wyższą [złożonością czasową](http://pl.wikipedia.org/wiki/Z%C5%82o%C5%BCono%C5%9B%C4%87_obliczeniowa).

**Dijkstry**

Złożoność czasowa wynosi O(n2).

**Bellmana Forda**

Wykres przedstawia zależność liczby wierzchołków od czasu w jakim wykonał się algorytm. Jest on w skali podwójnej logarytmicznej.

Złożoność czasową szacuje się na O(n3).

**Wady i zalety**

**Dijkstry**

**Zalety:** Szybszy od Bellmana Forda.

**Wady:** Nie może zawierać krawędzi o ujemnych wagach.

**Bellmana Forda**

**Zalety:** Może zawierać krawędzie o ujemnych wagach.

**Wady:** W grafie nie może być cyklu o ujemnej sumie wag krawędzi (przechodząc przez niego dostawalibyśmy coraz lżejsze wagi). Wolniejszy od Dijkstry.

**Wioski i posumowanie**

Każdy algorytm ma swoje wady i zalety. Dla grafów z wagami o nieujemnych wartościach lepiej jest stosować [Algorytm Dijkstry](http://www.algorytm.org/algorytmy-grafowe/algorytm-dijkstry.html).

**Wykorzystane materiały i źródła:**

* Thomas H. Cormes, Charles E. Leiserson, Ronald L. Rivest “Wprowadzenie do algorytmów” wydanie czwarte
* Adam Drozdek “C++ algorytmy I struktury danych” wydawnistwa Helion