Wojciech Krzaczek 184035 24.04.2012r

**Projektowanie algorytmów i metody sztucznej inteligencji**

**Lista nr 4: analiza efektywności algorytmów Dijkstry i Bellmana Forda.**

1. Wprowadzenie

Celem laboratorium było zbadanie złożoności obliczeniowej algorytmów szukających najkrótszej drogi do z danego punktu do pozostałych. Testowane były dwa algorytmy Dijkstry i Bellmana-Forda dla 50, 70, 100, 200 i 300 wierzchołków oraz o gęstości równej 100%, 75%, 50% i 25%.

Czas działania algorytmu jest podany w milisekundach i był mierzony funkcją czas() z biblioteki time.h.

1. Badane algorytmy

* Algorytm Dijkstry

Mając dany graf z wyróżnionym wierzchołkiem (*źródłem*) algorytm znajduje odległości od źródła do wszystkich pozostałych wierzchołków.

Złożoność obliczeniowa:

Wyniki szukania najkrótszej drogi prezentują się następująco:

1. Dla Macierzy Wag

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ilość  wierzchołków | 100% | 75% | 50% | 25% |
| 50 | 6 | 4 | 3 | 2 |
| 70 | 19 | 14 | 10 | 6 |
| 100 | 77 | 46 | 37 | 22 |
| 200 | 2162 | 1239 | 578 | 220 |
| 300 | 12745 | 7059 | 3481 | 871 |

1. Dla Połączonych List Sąsiadów

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ilość  wierzchołków | 100% | 75% | 50% | 25% |
| 50 | 5 | 5 | 3 | 7 |
| 70 | 20 | 15 | 12 | 9 |
| 100 | 78 | 57 | 43 | 35 |
| 200 | 1506 | 921 | 534 | 357 |
| 300 | 9845 | 5794 | 3073 | 1472 |

* Algorytm Bellmana-Forda

**Algorytm Bellmana-Forda** rozwiązuje problem najkrótszej ścieżki, tj. pozwala znaleźć ścieżkę o najmniejszej wadze pomiędzy dwoma wierzchołkami w grafie ważonym. Idea algorytmu opiera się na metodzie relaksacji (dokładniej następuje relaksacja  razy każdej z krawędzi). W odróżnieniu od algorytmu Dijkstry, poprawność algorytmu Bellmana-Forda nie opiera się na założeniu, że wagi w grafie są nieujemne (nie może jednak występować cykl o łącznej ujemnej wadze osiągalny ze źródła). Za tę ogólność płaci się jednak wyższą złożonością czasową.

Złożoność obliczeniowa:

Wyniki szukania najkrótszej drogi prezentują się następująco:

1. Dla Macierzy Wag

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ilość  wierzchołków | 100% | 75% | 50% | 25% |
| 50 | 3 | 2 | 1 | 1 |
| 70 | 13 | 8 | 4 | 3 |
| 100 | 60 | 31 | 14 | 4 |
| 200 | 1849 | 984 | 385 | 49 |
| 300 | 11650 | 6082 | 2641 | 278 |

1. Dla Połączonych List Sąsiadów

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ilość  wierzchołków | 100% | 75% | 50% | 25% |
| 50 | 3 | 2 | 2 | 1 |
| 70 | 13 | 7 | 4 | 1 |
| 100 | 53 | 30 | 18 | 6 |
| 200 | 1209 | 609 | 260 | 72 |
| 300 | 8705 | 4651 | 1775 | 384 |

1. Wnioski

Jak widać na wykresach algorytm Djikstry jest szybszy od Bellmana-Forda, gdyż nie był on zaimplementowany na kopcu, który zwiększa jego efektywność.

Jak można zauważyć oba algorytmy działają o wiele szybciej, gdy graf na którym pracują jest przedstawiony w postaci połączonych list sąsiadów.

1. Literatura

* Slajdy z wykładu Dr Krysiaka.
* http://pl.wikipedia.org/wiki/Algorytm\_Dijkstry