Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 1

20.03.2019r.

Nazwa kursu: Projektowanie algorytmów i metod sztucznej inteligencji

Prowadzący: dr inż. Łukasz Jeleń

Wykonał: Szymon Korczyński 24155

Termin: Środa 11-13

1. Wprowadzenie

Problem najkrótszej ścieżki polega na znalezieniu najkrótszej ścieżki pomiędzy dwoma wierzchołkami grafu.

Graf – sposób reprezentacji połączeń pomiędzy obiektami np. mapa, gdzie skrzyżowanie to wierzchołek grafu, natomiast droga to krawędzie. W grafie mogą występować ujemne wagi, jednak, aby było możliwe rozwiązanie problemu najkrótszej ścieżki musi zostać spełniony warunek braku cyklu o łącznej ujemnej wadze tzn. takiego połączenia pomiędzy wierzchołkami, dzięki którym całkowita droga będzie mniejsza od 0.

Również istnieją grafy skierowane, gdzie krawędzie łączą wierzchołki tylko w jedną stronę.

Do rozwiązania problemu najkrótszej ścieżki stosuje się takie algorytmy jak: Djikstry, Bellmana-Forda, A\*, Floyda-Warshalla.

Oznaczenia:

* V – ilość wierzchołków
* E – ilość krawędzi

1. Opis algorytmów
   1. Algorytm Bellmana-Forda

Algorytm może być stosowany dla grafów z wagami ujemnymi pod warunkiem, że nie występują cykle o łącznej ujemnej wadze.

Działanie algorytmu:

* Stworzenie tablicy odległości od wierzchołka początkowego oraz zainicjalizowanie jej jako nieskończoność dla wierzchołków niebędących początkowym.
* Sprawdzeniu V – 1 razy wszystkich krawędzi. Początkowo wybiera się krawędzie z pierwszego wierzchołka itd.
* Gdy suma wagi krawędzi i odległośc od wierzchołka początkowego jest mniejsza od odległości do wierzchołka docelowego
* Gdy w 1 iteracji nie zostanie zmieniona żadna wartość odległości, wówczas można przerwać działanie algorytmu.

Złożoność czasowa algorytmu to .

* 1. Algorytm Dijkstry

Algorytm może być stosowany tylko dla grafów z dodatnimi wagami.

Działanie algorytmu:

* Stworzenie tablicy odległości od wierzchołka początkowego
* Utworzenie kolejki priorytetowej wszystkich wierzchołków grafu. Priorytetem jest najkrótsza wyliczona odległość od wierzchołka wyróżnionego
* Dopóki kolejka nie jest pusta:
  + Usuwanie wierzchołka v z kolejki priorytetowej, mającego najmniejszą odległość od wierzchołka początkowego
  + Dla każdego sąsiadującego wierzchołka z v sprawdzić czy jest możliwa krótsza droga z wierzchołka początkowego.

Złożoność czasowa to algorytmu to , przy użyciu kopca Fibonacciego. Złożoność czasowa zależy od metody wyznaczenia wierzchołka grafu o najmniejszej odległości od wierzchołka początkowego. W przedstawionym algorytmie złożoność czasowa to

1. Wyniki

Algorytm Dijkstry

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Macierz czas działania w zależności od ilości elementów | | | | |
| Gęstość grafu | 10 [ms] | 50 [ms] | 100 [ms] | 500 [ms] | 1000 [ms] |
| 0,25 | 0,00154 | 0,0200 | 0,0708 | 3,77 | 23,99 |
| 0,5 | 0,00164 | 0,0247 | 0,0886 | 9,66 | 45,79 |
| 0,75 | 0,00182 | 0,0294 | 0,1241 | 14,76 | 68,28 |
| 1 | 0,00241 | 0,0315 | 0,1312 | 11,61 | 31,52 |
|  | Lista czas działania w zależności od ilości elementów | | | | |
| 0.25 | 0,00241 | 0,053 | 0,17 | 3,88 | 15,43 |
| 0,5 | 0,00292 | 0,062 | 0,22 | 5,03 | 21,47 |
| 0,75 | 0,00306 | 0,056 | 0,22 | 4,86 | 19,90 |
| 1 | 0,00276 | 0,047 | 0,17 | 4,32 | 18,52 |

Algorytm Bellmana-Forda

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Macierz czas działania w zależności od ilości elementów | | | | |
| Gęstość grafu | 10 [ms] | 50 [ms] | 100 [ms] | 500 [ms] | 1000 [ms] |
| 0,25 | 0,00150 | 0,0260 | 0,159 | 9,63 | 70,69 |
| 0,5 | 0,00191 | 0,0559 | 0,248 | 26,93 | 122,63 |
| 0,75 | 0,00242 | 0,0839 | 0,363 | 40,91 | 193,18 |
| 1 | 0,00269 | 0,0871 | 0,322 | 31,58 | 61,10 |
|  | Lista czas działania w zależności od ilości elementów | | | | |
| 0.25 | 0,00478 | 0,1789 | 0,676 | 14,64 | 48,01 |
| 0,5 | 0,00624 | 0,2122 | 0,849 | 17,73 | 60,89 |
| 0,75 | 0,00657 | 0,1859 | 0,762 | 13,74 | 51,81 |
| 1 | 0,00508 | 0,1470 | 0,561 | 10,57 | 48,21 |

1. Wnioski

* Czas działania algorytmu Dijkstry jest szybszy niż Bellmana-Forda
* Czas działania algorytmów jest krótszy z użyciem macierzy niż listy, co może być spowodowane złą optymalizacją listy.
* Działanie algorytmów z użyciem macierzy jest najszybszy dla gęstości grafu równej 0.5.
* Dla listy czas działania z użyciem obu algorytmów jest najkrótszy dla gęstości 0.75. Dla algorytmu Dijkstry może być spowodowane brakiem użycia kolejki priorytetowej.

1. Literatura

* Algorytm Dijkstry - https://pl.wikipedia.org/wiki/Algorytm\_Dijkstry, [dostęp: 7.05.2019r.]
* Algorytm Bellmana-Forda - https://pl.wikipedia.org/wiki/Algorytm\_Bellmana-Forda, [dostęp: 7.05.2019r.]