

Drumuri minime in graf : Costul minim intre oricare doua noduri

Grama Nicolae

Grupa 322CA, Facultatea de Automatica si Calculatoare
Universitatea Politehnica București
nicolae.grama@stud.acs.upb.ro

Abstract. Aceasta lucrare își propune analizarea principalelor metode de determinare a costului minim de la un nod la altul, pentru un graf ponderat, prin evidențierea avantajelor, cat si a dezavantajelor acestora.

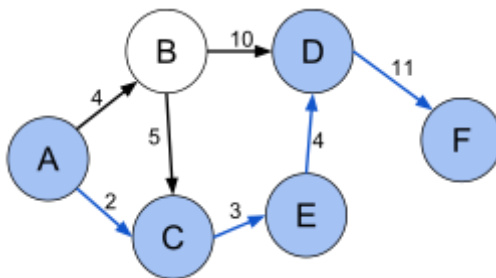
Cuvinte cheie : Grafuri, Dijkstra, Johnson, Floyd-Warshall

1 Introducere

1.1 Descrierea problemei rezolvate

În viață de zi cu zi, ne confruntăm cu o problemă : „Cum ajung cel mai repede din punctul A în punctul B ?”. Pentru a rezolvă această problemă, putem reprezenta harta , punctele în care vrem să ajungem cu ajutorul unui graf. Cum fiecare muchie din graf are un cost, problema inițială se transforma în „ Care este costul minim ca să ajung din nodul A în nodul B?”.

Soluțiile pentru această problemă („Shortest Path Problem) sunt algoritmi ce calculează costul minim între oricare două noduri : Dijkstra, Johnson, Floyd-Warshall, A*, Viterbi, Bellman-Ford, etc.



1.2 Exemple de aplicații practice pentru problema aleasa

Cum am menționat și anterior, acești algoritmi își găsesc aplicabilitatea în probleme de determinare a celui mai scurt drum, de exemplu, GPS. În momentul de față, costul muchiilor nu mai reprezintă neapărat distanța între noduri, ci timpul, deoarece GPS-ul va lua în calcul mai mulți parametrii pentru a determina cea mai eficientă rută din punct de vedere al timpului.

Acești algoritmi se pot folosi, de asemenea, pentru a rezolva Cubul Rubik: nodurile reprezintă stările cubului și muchiile reprezintă o transformare (rotație).

Alte aplicații sunt „operation research” (planul general al fabricilor), robotică, transporturi și design VLSI. .

1.3 Specificarea soluțiilor alese

În continuare, am ales să prezint 3 algoritmi care rezolvă problema propusă: Dijkstra, Johnson și Floyd-Warshall. O parte din aceștia calculează costul minim de la un nod la toate celelalte, dar pot fi adaptați pentru a calcula doar costul minim între oricare două noduri. Fiecare din aceștia prezintă anumite avantaje și dezavantaje:

- Dijkstra
 - Are o complexitate minimă $O(|V| * |E|)$ folosind heapuri Fibonacci
 - Nu se poate folosi dacă există muchii cu cost negativ
- Floyd-Warshall
 - Este ușor de implementat
 - Are o complexitate $O(|V|^3)$
- Johnson
 - Poate fi folosit pe grafuri cu cost negativ și are o complexitate mai bună decât Floyd-Warshall ($O(|V|^2 \log|V| + |V||E|)$) pe grafuri rare
 - Pentru grafuri complete, are aceeași complexitate ca Floyd-Warshall

1.4 Specificarea criteriilor de evaluare alese pentru validarea soluțiilor

Pentru a testa corectitudinea, eficiența și performanța acestor algoritmi, voi folosi un generator ce va construi teste cu inputuri aleatorii. Utilizatorul va avea posibilitatea să aleagă numărul de noduri din graf, numărul de muchii, intervalul de valori pentru costuri și eventual, dacă graful conține cicluri negative.

Vor exista astfel mai multe tipuri de teste, împărțite în două categorii: teste de acceptanță și teste de eficiență/performanță.

Testele de acceptanță vor verifica implementarea algoritmilor. Verificăm dacă algoritmul găsește un drum, dacă calculează costul corect (și implicit, drumul minim) și dacă algoritmul funcționează în situațiile în care „nu are voie să funcționeze” (ex.: Dijkstra pe un graf cu muchii cu cost negativ)

Testele de performanță compară cei trei algoritmi între ei pentru diferite inputuri, în special pentru cazurile nefavorabile specifice fiecărui algoritm, din punct de vedere al vitezei de execuție.

Referințe

1. OCW: PROIECTAREA ALGORITMILOR <https://ocw.cs.pub.ro/courses/pa>
2. GeeksForGeeks <https://www.geeksforgeeks.org/>