

Drumuri minime in graf

Olteanu Eduard-Florin

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București, România
edioltean99@yahoo.com

Rezumat. Drumul minim într-un graf reprezintă costul minim de la un nod la toate celelalte sau costul minim între oricare două noduri. Un graf este o structură care corespunde unui grup de obiecte, în care unele perechi de obiecte sunt "legate" reciproc de muchii orientate sau nu. Muchiile din graf pot avea sau nu cost. Pentru drumuri minime în graf există diferiți algoritmi, cum ar fi: Dijkstra, Bellman-Ford și Dijkstra adaptat (cunoscut și sub numele de Algoritmul lui Dial), pentru costul minim de la un nod la toate celelalte, și Floyd-Warshall, Johnson și Dijkstra sau Bellman-Ford (aplicate între oricare două noduri) pentru costul minim între oricare două noduri. În acest studiu vom analiza algoritmi pentru costul minim de la un nod la toate celelalte.

Cuvinte-cheie: graf, cost, Dijkstra, Bellman-Ford, Dial

1 Descrierea problemei rezolvate

1.1 Descrierea problemei rezolvate

Fie un graf ponderat orientat, $G(V, E)$, cu o funcție pentru costurile muchiilor, w definită pe E cu valori în \mathbb{R} . Problema își propune să găsească, pentru un nod u , drumul minim de la u la toate celelalte noduri, unde costul unui drum este suma tuturor costurilor muchiilor ce alcătuiesc drumul. Deși, în cele mai multe cazuri, costul este o funcție cu valori pozitive, există situații în care graful cu muchii de cost negativ are relevanță practică. O parte dintre algoritmi pot determina drumul corect de cost minim, inclusive pe aceste grafuri. Totuși, căutarea drumului minim nu are sens în cazul în care graful conține cicluri de cost negativ - un drum minim va avea lungime infinită, întrucât costul său s-ar reduce la fiecare parcurgere a ciclului.

1.2 Aplicații practice

Algoritmii pentru determinarea drumurilor minime au multiple aplicații practice:

- Găsirea drumului minim dintre două locații (Google Maps, GPS etc.)
- Stabilirea unei agende de zbor, în vederea asigurării unor conexiuni optime
- Rutare în cadrul unei rețele (telefonice, de calculatoare etc.)

2 Specificarea algoritmilor aleși

2.1 Algoritmul lui Dijkstra

Algoritmul lui Dijkstra determină pentru un nod dat într-un graf orientat cu costuri costurile minime ale drumurilor care au acel nod ca extremitate inițială.

Mai precis, pentru un nod u – sursă, algoritmul determină pentru orice nod v costul minim al unui drum de la u la v . [1]

2.2 Algoritmul Bellman-Ford

Algoritmul Bellman Ford poate fi folosit și pentru grafuri ce conțin muchii de cost negativ, dar nu poate fi folosit pentru grafuri ce conțin cicluri de cost negativ (când căutarea unui drum minim nu are sens). Cu ajutorul său putem afla dacă un graf conține cicluri.

Algoritmul folosește același mecanism de relaxare ca și Dijkstra, dar, spre deosebire de acesta, nu optimizează o soluție folosind un criteriu de optim local, ci parcurge fiecare muchie de un număr de ori egal cu numărul de noduri și încearcă să o relaxeze de fiecare dată, pentru a îmbunătăți distanța până la nodul destinație al muchiei curente.

2.3 Algoritmul lui Dial – Dijkstra adaptat

Algoritmul lui Dial este un algoritm Dijkstra adaptat pentru costuri mici. Astfel, algoritmul lui Dijkstra este modificat, folosind altă structură de date numită buckets, iar algoritmul alege diferit și nodul cu costul minim la fiecare pas. Fie C lungimea maximă a unei muchii în G . Algoritmul menține $nC + 1$ buckets, în bucket-ul k se află toate nodurile cu costul egal cu k (orice cost finit fiind cel mult nC).

3 Criteriile de evaluare pentru soluția propusă

Voi implementa un algoritm care va crea matricea de adiacență a grafului de dimensiune $n \times n$, unde n este numărul de noduri din graf, folosind valori generate random. Eficiența va fi testată folosind un număr mare de noduri. ($n \in [100, 1000]$).

References

1. <https://www.pbinfo.ro/?pagina=articole&subpagina=afisare&id=6135>
2. <https://www.geeksforgeeks.org/dials-algorithm-optimized-dijkstra-for-small-range-weights/>
3. <http://elf.cs.pub.ro/pa/wiki/laboratoare/laborator-08#algoritmul-bellman-ford>
4. <https://courses.csail.mit.edu/6.006/spring11/lectures/lec15.pdf>
5. <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>