Algoritmi pentru grafuri și aplicații

Gălbiniță Sebastian

September 6, 2020

Cuprins

Drumuri minime de sursă unică

Fie un graf orientat ponderat $G = (V, E)$ și funcția cost $f : E \longrightarrow \mathbb{R}$. **Costul** drumului $p = [\alpha_0, \alpha_1, ..., \alpha_k]$ este dat de

$$f(p) = \sum_{i=1}^{k} f(\alpha_{i-1}, \alpha_i)$$

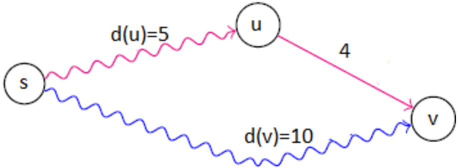
Astfel putem defini **costul unui drum minim**

$$\delta(u, v) = \begin{cases} min\left\{f(p) : u \leadsto v\right\}, & \text{dacă există drum de la } u \text{ la } v \\ \infty, & \text{altfel} \end{cases}$$

Relaxare

Pentru fiecare nod $v \in V$, se va reține un **predecesor** $\omega[v]$ și conservăm un atribut $d[v]$.
Procesul de relaxare a unei muchii (u, v) este dat de următoarea inegalititate:

$$d[v] > d[u] + f(u, v)$$



Algoritmul lui Dijkstra

Algoritmul lui Dijkstra rezolvă problema drumurilor minime de sursă unică într-un graf orientat ponderat $G = (V, E)$ pentru care toate costurile muchiilor sunt nenegative. Vom presupune că pentru fiecare muchie $(u, v) \in E$, $f(u, v) \geq 0$.

DIJKSTRA (G, f, s)

INIȚIALIZEAZĂ-SURSĂ-UNICĂ(G, s)
$S \longleftarrow \emptyset$
$Q \longleftarrow V(G)$
while $Q \neq \emptyset$
    $u \longleftarrow$ EXTRAGE-MIN(Q)
    $S \longleftarrow S \cup \{u\}$
    for fiecare vârf $v \in Adj[u]$
        RELAXEAZĂ(u, v, f)

Drumuri minime între toate perechile de vârfuri

Fie un graf orientat ponderat $G = (V, E)$ și o funcție de costuri $f : E \longrightarrow \mathbb{R}$ aplicată arcelor grafului. Pentru fiecare $u, v \in V$, determinăm un **drum de cost minim** de la u la v. Ca date de intrare avem o matrice A, având dimensiunea $n \times n$.

$$a_{ij} = \begin{cases} 0, & \text{dacă } i = j, \\ f(i,j), & \text{dacă } i \neq j \text{ și } (i,j) \in E, \\ \infty, & \text{dacă } i \neq j \text{ și } (i,j) \notin E. \end{cases}$$

Iar ca date de ieșire o matrice $D = (d_{ij})$ de dimensiune $n \times n$.

Structura unui drum minim

Presupunem că graful este reprezentat printr-o matrice de adiacență $A = (a_{ij})$. Considerăm un **drum minim** p de la nodul i la j și presunem că are m arce. Presupunând că nu sunt cicluri de cost negativ, atunci m este finit. Dacă $i = j$, atunci p are costul 0 și nu conține nici un arc. Dacă nodurile sunt distincte, atunci descompunem drumul p în $i \overset{p'}{\rightsquigarrow} k \to j$, unde drumul p' conține cel mult $m - 1$ arce. Mai mult, p' este un drum minim de la i la k. Deci avem următoarea egalitate:

$$\delta(i,j) = \delta(i,k) + f(k,j).$$

Determinarea drumurilor minime

Presupunem ca date de intrare matricea $A = (a_{ij})$, determinăm o serie de matrici $D^{(1)}, D^{(2)}, ..., D^{(n-1)}$, unde, pentru $m = 1, 2, ..., n-1$ avem $D^{(m)} = (d_{ij}^{(m)})$. Matricea finală $D^{(n-1)}$ va conține costurile drumurilor minime.

EXTINDE(D, A)

$n \leftarrow linii[D]$
fie $B = (b_{ij})$ matrice cu dimensiunea $n \times n$
for $i \leftarrow 1, n$
    for $j \leftarrow 1, n$
        $b_{ij} \leftarrow \infty$
        for $k \leftarrow 1, n$
            $b_{ij} \leftarrow \min(b_{ij}, d_{ik} + a_{kj})$
return B

Determinarea drumurilor minime

$$d_{ij}^{(m)} = \min\left(d_{ij}^{(m-1)}, \min_{1\le k\le n}\left\{d_{ik}^{(m-1)} + a_{kj}\right\}\right) = \min_{1\le k\le n}\left\{d_{ik}^{(m-1)} + a_{kj}\right\}$$

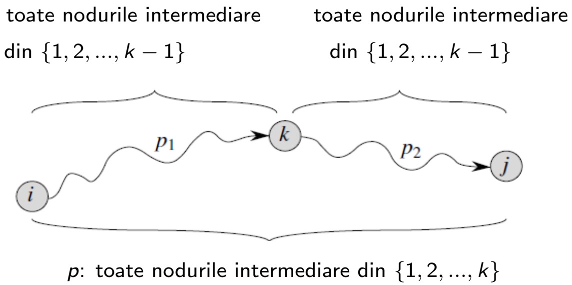
Deoarece am determinat șirul de $n-1$ matrice, putem transpune tot ce scris într-o funcție.

DRUMURI-MINIME(A)

$n \leftarrow linii[A]$
$D^{(1)} \leftarrow A$
for $i \leftarrow 2, n-1$
    $D^{(i)} \leftarrow$EXTINDE $(D^{(i-1)}, A)$
return $D^{(n-1)}$

Algoritmul Floyd-Warshall

Algoritmul lui Floyd-Warshall este un algoritm pentru găsirea celor mai scurte drumuri într-un graf orientat ponderat cu cost pozitiv sau negativ. Acest algoritm se bazează pe următoarea observație. Fie $V = \{1, 2, ..., n\}$ mulțimea nodurilor lui G. Considerăm submulțimea $\{1, 2, ..., k\}$ pentru un anumit k. Pentru orice pereche de noduri $i, j \in V$, considerăm toate drumurile de la i la j ale căror noduri intermediare fac parte din mulțimea $\{1, 2, ..., k\}$. Fie p drumul de cost minim dintre aceste drumuri. Algoritmul Floyd-Warshall exploatează o relație între drumul p și drumul minim de la i la j cu toate nodurile intermediare.



Dacă k nu este nod intermediar al drumului p, un drum minim de la nodul i la j cu toate nodurile intermediare din mulțimea $\{1, 2, ..., k-1\}$ este, de asemenea, un drum minim de la i la j cu toate nodurile intermediare din mulțimea $\{1, 2, ..., k\}$.
Dacă k este nod intermediar al drumului p, atunci împărțim p în două alte drumuri. Deoarece p este drum minim rezultă că și p_1 este drum minim de la i la k cu toate nodurile intermediare din mulțimea $\{1, 2, ..., k-1\}$. Analog pentru p_2.

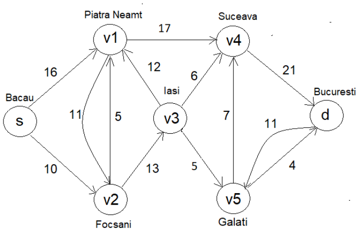
Intrarea este o matrice A de dimensiune $n \times n$. Funcția returnează matricea $D^{(n)}$ a costurilor drumurilor minime.

FLOYD-WARSHALL(A)

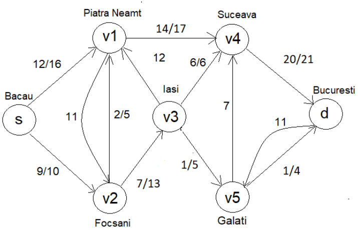
$n \leftarrow linii[A]$
$D^{(0)} \leftarrow A$
for $k \leftarrow 1, n$
    for $i \leftarrow 1, n$
        for $j \leftarrow 1, n$
            $d_{ij}^{(k)} \leftarrow min\big(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\big)$
return $D^{(n)}$

Flux maxim

Problema **fluxului maxim** este aceea de a determina cantitatea cea mai mare de material care poate fi transportată pornind de la sursă și ajungând la destinație ținând cont de restricțiile de capacitate.



a)



b)

O **rețea de transport** este un graf orientat $G = (V, E)$ în care fiecărei muchii $(u, v) \in E$ îi este atașată o **capacitate** nenegativă $c(u, v) \geq 0$. Dacă $(u, v) \notin E$ atunci considerăm $c(u, v) = 0$. Fixăm nodul sursă s și nodul destinație d. Denumim **fluxul** G ca fiind o funcție $f : V \times V \to \mathbb{R}$ care satisface următoarele condiții:

1. **Restricția de capacitate**: Pentru orice $u, v \in V$, $f(u, v) \leq c(u, v)$.
2. **Antisimetria**: Pentru orice $u, v \in V, f(u, v) = -f(u, v)$.
3. **Conservarea fluxului**: Pentru orice $u \in V \setminus \{s, d\}$ avem

$$\sum_{v \in V} f(u, v) = 0$$

Denumim **capacitatea reziduală** a arcului (u, v) ca fiind cantitatea de flux adițională care poate fi transportată de la u la v, fără a depăși $c(u, v)$.

Metoda lui Ford-Fulkerson

În fiecare iterație a metodei lui Ford-Fulkerson căutăm un drum *oarecare* de ameliorare p și mărim fluxul f de-a lungul drumului p cu capacitatea reziduală $c_f(p)$.

METODA-FORD-FULKERSON(s, d, G)

for fiecare arc $(u, v) \in E[G]$
    $f(u, v) \leftarrow 0$
    $f(v, u) \leftarrow 0$
while există un drum de la s la d în rețeaua reziduală G_f
    $c_f(p) \leftarrow min\{c_f(u, v) | (u, v) \in p\}$
    for fiecare (u, v) din p
        $f(u, v) \leftarrow f(u, v) + c_f(p)$
        $f(u, v) \leftarrow -f(u, v)$