Proiectarea algoritmilor
Lucrare de laborator nr. 13
Paradigma programării dinamice
Problema drumurilor minime între oricare două vârfuri ale unui digraf ponderat

Cuprins

1 Descriere

Se consideră un digraf ponderat $D = (\langle V, A\rangle, \ell)$.

Problema constă în a determina, pentru oricare două vârfuri i, j, un drum de lungime minimă de la vârful i la vârful j (dacă există).

Metoda utilizată este programarea dinamică.

2 Modelul matematic

Extindem funcţia ℓ la $\ell : V \times V \to \mathscr{R}$, prin asignarea $\ell_{ij} = \infty$ pentru acele perechi de vârfuri distincte cu $\langle i, j\rangle \notin E$ şi $\ell_{ii} = 0$ pentru orice $i = 0, \ldots, n-1$.

Definim starea problemei ca fiind subproblema corespunzătoare determinării drumurilor de lungime minimă cu vârfuri intermediare din mulţimea $X \subseteq V$, DM2VD(X) (Drum Minim între oricare două Vârfuri ale unui Digraf). Evident, DM2VD(V) este chiar problema iniţială.

Notăm cu ℓ_{ij}^{X} lungimea drumului minim de la i la j construit cu vârfuri intermediare din X. Dacă $X = \emptyset$, atunci $\ell_{ij}^{\emptyset} = \ell_{ij}$.

Considerăm decizia optimă care transformă starea DM2VD$(X \cup \{k\})$ în DM2VD(X).

Presupunem că (G, ℓ) *este un digraf ponderat fără circuite negative*.

Fie ρ un drum optim de la i la j ce conţine vârfuri intermediare din mulţimea $X \cup \{k\}$. Avem lung$(\rho) = \ell_{ij}^{X\cup\{k\}}$, unde lung$(\rho)$ este lungimea drumului ρ.

Dacă vârful k nu aparţine lui ρ, atunci politica obţinerii lui ρ corespunde stării DM2VD(X) şi, aplicând principiul de optim, obţinem:

$$\ell_{ij}^{X} = \text{lung}(\rho) = \ell_{ij}^{X\cup\{k\}}$$

În cazul în care k aparţine drumului ρ, notăm cu ρ_1 subdrumul lui ρ de la i la k şi cu ρ_2 subdrumul de la k la j. Aceste două subdrumuri au vârfuri intermediare numai din X.

Conform principiului de optim, politica optimă corespunzătoare stării DM2VD(X) este subpolitică a politicii optime corespunzătoare stării DM2VD$(X \cup \{k\})$. Rezultă că ρ_1 şi ρ_2 sunt optime în DM2VD(X). De aici rezultă:

$$\ell_{ij}^{X\cup\{k\}} = \text{lung}(\rho) = \text{lung}(\rho_1) + \text{lung}(\rho_2) = \ell_{ik}^X + \ell_{kj}^X$$

Acum, ecuaţia funcţională analitică pentru valorile optime ℓ_{ij}^X are următoarea formă:

$$\ell_{ij}^{X\cup\{k\}} = \min\{\ell_{ij}^X, \ell_{ik}^X + \ell_{kj}^X\}$$

Corolar 2.1 *Dacă $\langle D, \ell\rangle$ nu are circuite de lungime negativă, atunci au loc următoarele relaţii:*
$\ell_{kk}^{X\cup\{k\}} = 0$
$\ell_{ik}^{X\cup\{k\}} = \ell_{ik}^X$
$\ell_{kj}^{X\cup\{k\}} = \ell_{kj}^X$ *pentru orice $i, j, k \in V$.*

Calculul valorilor optime rezultă din rezolvarea subproblemelor

$$\text{DM2VD}(\emptyset), \text{DM2VD}(\{0\}), \text{DM2VD}(\{0,1\}), \ldots, \text{DM2VD}(\{0,1,\ldots,n-1\}) =$$

$$\text{DM2VD}(V)$$

Convenim să notăm ℓ_{ij}^k în loc de $\ell_{ij}^{\{0,\ldots,k\}}$. Pe baza corolarului rezultă că valorile optime pot fi memorate într-un acelaşi tablou.

Maniera de determinare a acestora este asemănătoare cu cea utilizată la determinarea matricei drumurilor de către algoritmul Floyd–Warshall.

Pe baza ecuaţiilor anterioare, proprietatea de substructură optimă se caracterizează prin proprietatea următoare:

> Un drum optim de la i la j include drumurile optime de la i la k şi de la k la j, pentru orice vârf intermediar k al său.

Astfel, drumurile minime din DM2VD$(X \cup \{k\})$ pot fi determinate utilizând drumurile minime din DM2VD(X).

În continuare considerăvem numai cazurile $X = \{0,1,\ldots,k-1\}$ şi $X \cup \{k\} = \{0,1,\ldots,k-1,k\}$.

Determinarea drumurilor optime poate fi efectuată cu ajutorul unor matrice $P^k = (P_{ij}^k)$, care au semnificaţia următoare: P_{ij}^k este penultimul vârf din drumul optim de la i la j. Iniţial, avem $P_{ij}^{init} = i$, dacă $\langle i, j\rangle \in E$ şi $P_{ij}^{init} = -1$, în celelalte cazuri.

Decizia k determină matricele $\ell^k = (\ell_{ij}^k)$ şi $P^k = (P_{ij}^k)$.

- Dacă $\ell_{ik}^{k-1} + \ell_{kj}^{k-1} < \ell_{ij}^{k-1}$, atunci drumul optim de la i la j este format din concatenarea drumului optim de la i la k cu drumul optim de la k la j şi penultimul vârf din drumul de la i la j coincide cu penultimul vârf din drumul de la k la j: $P_{ij}^k = P_{kj}^{k-1}$.

- În caz contrar, avem $P_{ij}^k = P_{ij}^{k-1}$.

Cu ajutorul matricei P_{ij}^{n-1} pot fi determinate drumurile optime: ultimul vârf pe drumul de la i la j este $j_t = j$, penultimul vârf este $j_{t-1} = P_{ij_t}^{n-1}$, antipenultimul este $j_{t-2} = P_{ij_{t-1}}^{n-1}$ ş.a.m.d.

În acest mod, toate drumurile pot fi memorate utilizând numai $O(n^2)$ spaţiu.

3 Algoritmul Floyd-Warshall

```
procedure Floyd-Warshall(G, ℓ, P)
   for i ← 0 to n-1 do
       for j ← 0 to n-1 do
```

$$\ell^{\mathtt{init}}_{\mathtt{ij}} = \begin{cases} 0 & ,i=j \\ \ell_{ij} & ,\langle i,j\rangle \in A \\ \infty & ,\text{altfel} \end{cases}$$

$$\mathtt{P}^{\mathtt{init}}_{\mathtt{ij}} = \begin{cases} i & ,i \neq j, \langle i,j\rangle \in A \\ -1 & ,\text{altfel} \end{cases}$$

```
   for i ← 0 to n-1 do
       for j ← 0 to n-1 do
```

$$\ell^{0}_{\mathtt{ij}} = \min\{\ell^{\mathtt{init}}_{\mathtt{ij}}, \ell^{\mathtt{init}}_{\mathtt{i0}} + \ell^{\mathtt{init}}_{\mathtt{0j}}\}$$

$$\mathtt{P}^{0}_{\mathtt{ij}} = \begin{cases} P^{init}_{ij} & ,\ell^{0}_{ij} = \ell^{init}_{ij} \\ P^{init}_{0j} & ,\ell^{0}_{ij} = \ell^{init}_{i0} + \ell^{init}_{0j} \end{cases}$$

```
   for k ← 1 to n-1 do
       for i ← 0 to n-1 do
           for j ← 0 to n-1 do
```

$$\ell^{\mathtt{k}}_{\mathtt{ij}} = \min\{\ell^{\mathtt{k-1}}_{\mathtt{ij}}, \ell^{\mathtt{k-1}}_{\mathtt{ik}} + \ell^{\mathtt{k-1}}_{\mathtt{kj}}\}$$

$$\mathtt{P}^{\mathtt{k}}_{\mathtt{ij}} = \begin{cases} P^{k-1}_{ij} & ,\ell^{k}_{ij} = \ell^{k-1}_{ij} \\ P^{k-1}_{kj} & ,\ell^{k}_{ij} = \ell^{k-1}_{ik} + \ell^{k-1}_{kj} \end{cases}$$

```
end
```

Presupunem că digraful $G = (V,A)$ este reprezentat prin matricea de ponderilor (lungimilor) arcelor, pe care convenim să o notăm aici cu `G.L` (este uşor de văzut că matricea ponderilor include şi reprezentarea lui A).

Datorită corolarului (1), matricele ℓ^k şi ℓ^{k-1} pot fi memorate de acelaşi tablou bidimensional `G.L`.

Simbolul ∞ este reprezentat de o constantă `plusInf` cu valoare foarte mare.

Dacă digraful are circuite negative, atunci acest lucru poate fi depistat:

Dacă la un moment dat se obţine $G.L[i,i] < 0$, pentru un i oarecare, atunci există un circuit de lungime negativă care trece prin i.

Funcţia `Floyd-Warshall` întoarce valoarea *true* dacă digraful ponderat reprezentat de matricea `G.L` nu are circuite negative:

`G.L` conţine la ieşire ponderile (lungimile) drumurilor minime între oricare două vârfuri;

`G.P` conţine la ieşire reprezentarea drumurilor minime.

```
procedure Floyd-Warshall(G, P)
   for i ← 0 to n-1 do
       for j ← 0 to n-1 do
           if ((i ≠ j) and (L[i,j] ≠ plusInf))
              then P[i,j] ← i
              else P[i,j] ← -1
   for k ← 0 to n-1 do
       for i ← 0 to n-1 do
           for j ← 1 to n do
               if ((L[i,k] = PlusInf) or (L[k,j] = PlusInf))
                  then temp ← plusInf
                  else temp ← L[i,k]+L[k,j]
               if (temp < L[i,j])
                  then L[i,j] ← temp
                       P[i,j] ← P[k,j]
               if ((i = j) and (L[i,j] < 0))
                  then throw '(di)graful are circuite negative'
end
```

Se verifică uşor că execuţia algoritmului `Floyd-Warshall` necesită $O(n^3)$ timp şi utilizează $O(n^2)$ spaţiu.

4 Sarcini de lucru

1. Scrieţi o funcţie C/C++ care implementează un algoritmul Floyd-Warshall.
2. Dat fiind un graf $G = (V,E)$, scrieţi un program care să afişeze drumurile minime între oricare două vârfuri

Bibliografie

[1] Lucanu, D. şi Craus, M., *Proiectarea algoritmilor*, Editura Polirom, 2008.