Proiectarea algoritmilor

Paradigma programării dinamice

Lucrare de laborator nr. 11

Cuprins

Problema drumurilor minime între oricare două vârfuri

- 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.

Problema drumurilor minime - modelul matematic

- Extindem funcția ℓ la $\ell : V \times V \to \mathscr{R}$, prin asignarea $\ell_{ij} = \infty$ pentru acele perechi de vârfuri distincte cu $\langle i,j \rangle \notin E$ și $\ell_{ii} = 0$ pentru orice $i = 0, \ldots, n-1$.
- Definim starea problemei ca fiind subproblema corespunzătoare determinării drumurilor de lungime minimă cu vârfuri intermediare din mulțimea $X \subseteq V$, $\mathrm{DM2VD}(X)$ (Drum Minim între oricare două Vârfuri ale unui Digraf).
- Evident, $\mathrm{DM2VD}(V)$ este chiar problema inițială.
- 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 $\mathrm{DM2VD}(X \cup \{k\})$ în $\mathrm{DM2VD}(X)$.
- Presupunem că (G, ℓ) *este un digraf ponderat fără circuite negative*.
- Fie ρ un drum optim de la i la j ce conține vârfuri intermediare din mulțimea $X \cup \{k\}$.
- Avem $\mathrm{lung}(\rho) = \ell_{ij}^{X \cup \{k\}}$, unde $\mathrm{lung}(\rho)$ este lungimea drumului ρ.
- Dacă vârful k nu aparține lui ρ, atunci politica obținerii lui ρ corespunde stării $\mathrm{DM2VD}(X)$ și, aplicând principiul de optim, obținem:

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

Problema drumurilor minime - modelul matematic (continuare)

- În cazul în care k aparține drumului ρ, notăm cu ρ_1 subdrumul lui ρ de la i la k și 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 $\mathrm{DM2VD}(X)$ este subpolitică a politicii optime corespunzătoare stării $\mathrm{DM2VD}(X \cup \{k\})$.
- Rezultă că ρ_1 și ρ_2 sunt optime în $\mathrm{DM2VD}(X)$.
- De aici rezultă:

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

- Acum, ecuația funcțională analitică pentru valorile optime ℓ_{ij}^{X} are următoarea formă:

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

Problema drumurilor minime - modelul matematic (continuare)

- **Corolar:** Dacă $\langle D, \ell \rangle$ nu are circuite de lungime negativă, atunci au loc următoarele relații:

$$\ell_{kk}^{X \cup \{k\}} = 0$$
$$\ell_{ik}^{X \cup \{k\}} = \ell_{ik}^{X}$$
$$\ell_{kj}^{X \cup \{k\}} = \ell_{kj}^{X}$$

pentru orice $i, j, k \in V$.

- Calculul valorilor optime rezultă din rezolvarea subproblemelor

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

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

Problema drumurilor minime - modelul matematic (continuare)

- Pe baza ecuațiilor 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 și de la k la j, pentru orice vârf intermediar k al său.
- Astfel, drumurile minime din $\mathrm{DM2VD}(X \cup \{k\})$ pot fi determinate utilizând drumurile minime din $\mathrm{DM2VD}(X)$.

Problema drumurilor minime - modelul matematic (continuare)

- În continuare considerăvem numai cazurile $X = \{0, 1, \ldots, k-1\}$ și $X \cup \{k\} = \{0, 1, \ldots, k-1, k\}$
- Determinarea drumurilor optime poate fi efectuată cu ajutorul unor matrice $P^k = (P^k_{ij})$, care au semnificația următoare: P^k_{ij} este penultimul vârf din drumul optim de la i la j.
- Inițial, avem $P^{init}_{ij} = i$, dacă $\langle i, j \rangle \in E$ și $P^{init}_{ij} = -1$, în celelalte cazuri.
- Decizia k determină matricele $\ell^k = (\ell^k_{ij})$ și $P^k = (P^k_{ij})$.
  - Dacă $\ell^{k-1}_{ik} + \ell^{k-1}_{kj} < \ell^{k-1}_{ij}$, 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 și penultimul vârf din drumul de la i la j coincide cu penultimul vârf din drumul de la k la j: $P^k_{ij} = P^{k-1}_{kj}$.
  - În caz contrar, avem $P^k_{ij} = P^{k-1}_{ij}$.
- Cu ajutorul matricei P^{n-1}_{ij} pot fi determinate drumurile optime: ultimul vârf pe drumul de la i la j este $j_t = j$, penultimul vârf este $j_{t-1} = P^{n-1}_{ij_t}$, antipenultimul este $j_{t-2} = P^{n-1}_{ij_{t-1}}$ ș.a.m.d.
- În acest mod, toate drumurile pot fi memorate utilizând numai $O(n^2)$ spațiu.

Algoritmul Floyd-Warshall - pseudocod

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

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

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

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

$$\ell_{ij}^{0} = \min\{\ell_{ij}^{init}, \ell_{i0}^{init} + \ell_{0j}^{init}\}$$

$$P_{ij}^{0} = \begin{cases} P_{ij}^{init} & ,\ell_{ij}^{0} = \ell_{ij}^{init} \\ P_{0j}^{init} & ,\ell_{ij}^{0} = \ell_{i0}^{init} + \ell_{0j}^{init} \end{cases}$$

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

$$\ell_{ij}^{k} = \min\{\ell_{ij}^{k-1}, \ell_{ik}^{k-1} + \ell_{kj}^{k-1}\}$$

$$P_{ij}^{k} = \begin{cases} P_{ij}^{k-1} & ,\ell_{ij}^{k} = \ell_{ij}^{k-1} \\ P_{kj}^{k-1} & ,\ell_{ij}^{k} = \ell_{ik}^{k-1} + \ell_{kj}^{k-1} \end{cases}$$

```
end
```

Algoritmul Floyd-Warshall - implementare (descriere)

- 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 ușor de văzut că matricea ponderilor include și reprezentarea lui A).
- Datorită corolarului (1), matricele ℓ^k și ℓ^{k-1} pot fi memorate de același 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 obține $G.L[i, i] < 0$, pentru un i oarecare, atunci există un circuit de lungime negativă care trece prin i.
- Funcția `Floyd-Warshall` întoarce valoarea *true* dacă digraful ponderat reprezentat de matricea `G.L` nu are circuite negative:
  - `G.L` conține la ieșire ponderile (lungimile) drumurilor minime între oricare două vârfuri;
  - `G.P` conține la ieșire reprezentarea drumurilor minime.

Algoritmul Floyd-Warshall - implementare (pseudocod)

```
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
```

Algoritmul Floyd-Warshall - evaluare

Se verifică ușor că execuția algoritmului `Floyd-Warshall` necesită $O(n^3)$ timp și utilizează $O(n^2)$ spațiu.

Sarcini de lucru și barem de notare

Sarcini de lucru:

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

Barem de notare:

1. Implementarea algoritmului Floyd-Warshall: 7p
2. Afișarea drumurilor minime între oricare două vârfuri: 2p
3. Baza: 1p

Bibliografie

 Lucanu, D. și Craus, M., *Proiectarea algoritmilor*, Editura Polirom, 2008.