

Determinarea clicilor

Definiție 1

Fie $G=(V, U)$ un graf neorientat. Se numește **clică** un subgraf complet maximal cu această proprietate.

Problema determinării tuturor clicilor într-un graf sau a clicii cu număr maxim de vârfuri este **NP-completă**.

Notății:

CS (Complete Subgraph) – mulțimea vârfurilor din CS induce un subgraf complet în G

CA (Candidates) – mulțimea vârfurilor utilizate pentru a extinde CS astfel încât să devină clică

NOT – mulțimea vârfurilor care au fost utilizate până acum pentru extinderea CS și sunt în continuare excluse.

Mulțimile CA și NOT conțin toate vârfurile grafului, care nu aparțin mulțimii CS dar sunt adiacente cu toate vârfurile din CS.

Pe parcursul algoritmului vom nota NOT_i și CA_i mulțimile NOT și CA de la pasul i .

Generarea recursivă prin backtracking a clicilor

Algoritmul generează recursiv toate extensiile mulțimii CS, adăugând vârfuri din CA, nu din NOT. Vârfurile din NOT sunt excluse, deoarece într-o etapă anterioară au fost construite toate extensiile mulțimii CS care conțin acele vârfuri.

Inițial:

$CS \leftarrow \emptyset$

$NOT_0 \leftarrow \emptyset$

$CA_0 \leftarrow V$

generare (i)

{Dacă $CA_i \neq \emptyset$

 {Extrage primul candidat din mulțimea CA_i (c)

$CS \leftarrow CS \cup \{c\}$

$CA_{i+1} = CA_i - \{x \mid x \in CA_i, [x, c] \notin U\}$

$NOT_{i+1} = NOT_i - \{x \mid x \in NOT_i, [x, c] \notin U\}$

 generare (i+1)

$CS \leftarrow CS - \{c\}$

$NOT_i \leftarrow NOT_i \cup \{c\}$

 }

}

Pentru ca CS să fie clică trebuie ca atât CA, cât și NOT să fie vide.

Condiție de marginire:

Dacă NOT conține un vârf (să-l notăm v) adiacent cu toate vârfurile din CA, nu se va obține o nouă clică (pentru că în orice apel ulterior v va rămâne în NOT și subgraful complet obținut nu e maximal).

Algoritmul Bron-Kerbosch

Algoritmul precedent generează toate clicile, în ordine lexicografică.

Algoritmul Bron-Kerbosch selectează un nou vârf din mulțimea de candidați astfel încât condiția de marginire să devină adevărată cât mai repede cu putință.

Pentru aceasta:

1. Se determină un vârf v din $\text{NOT} \cup \text{CA}$ cu număr maxim de vârfuri adiacente în CA.
2. Dacă $v \in \text{CA}$, v va fi noul candidat. La revenirea din apelul recursiv, v va fi plasat în NOT. În acest moment $v \in \text{NOT}$ și vom continua selectând mai întâi acei candidați care nu sunt adiacenți cu v . Dacă nu mai există candidați în CA care nu sunt adiacenți cu v , aplicăm condiția de marginire și revenim.

Algoritmul BK nu generează clicile în ordine lexicografică, ci are tendința de a genera mai întâi clicile de dimensiune mai mare.

Determinarea tuturor clicilor maxime într-o familie de grafuri induse

Problema

Să considerăm M o mulțime de obiecte, fiecare obiect având asociate o mulțime de proprietăți P (mulțimea de proprietăți fiind reprezentată prin vectore caracteristic).

Să considerăm că între obiectele din M există o relație simetrică și nereflexivă R .

Se cere să se determine o submulțime maximală de obiecte, astfel încât:

- oricare două obiecte din submulțime să fie în relația R ;
- toate obiectele din submulțime au cel puțin o proprietate comună.

Transpunerea problemei în limbaj de grafuri.

Fie $G=(V, U)$ un graf neorientat (vârfurile sunt obiectele; dacă două obiecte sunt în relația R , atunci există muchie între vârfurile asociate obiectelor respective).

Fiecare proprietate P_i din P induce un subgraf al lui G astfel:

1. Mulțimea vârfurilor X_i este formată din vârfurile corespunzătoare obiectelor care au proprietatea P_i . Mai exact, $X_i = \{x \in V \mid P_i(x) = 1\}$
2. Mulțimea muchiilor este formată din muchiile din U care au ambele extremități în X_i .
Mai exact, $U_i = \{[x, y] \in U \mid x, y \in X_i\}$

Observație

Mulțimile de vârfuri din subgrafurile induse de proprietățile din P nu sunt în neapărat disjuncte.

Problema se reduce la a determina toate clicile maxime ale subgrafurilor induse de proprietățile din mulțimea P .

Definiție 2

Fie $G_1, G_2, \dots, G_m$ subgrafurile induse de proprietățile din P .

Fie $C_i =$ mulțimea tuturor clicilor din graful G_i .

O clică $C \in C_i$ este clică maximală în G dacă nu există o clică $D \in C_j$ astfel încât $C \subset D$.

Mulțimea clicilor maxime ale grafului G o vom nota MC .

Definiție 3

Fie $G=(V, U)$ un graf neorientat și $P=(P_1, P_2, \dots, P_m)$ o mulțime de proprietăți asociate vârfurilor. Pentru orice $i \in V$, definim $A_i = \{j \mid P_j(i) = 1\}$ (mulțimea proprietăților vârfului i).

Definim un graf cu atribute G_A ca un triplet (V, U, A) , unde $A = \{A_1, A_2, \dots, A_n\}$.

Definiție 4

Fie G_A un graf cu atribute. O submulțime $C = \{i_1, i_2, \dots, i_k\} \subset V$ se numește clică în G_A dacă îndeplinește următoarele 3 proprietăți:

1. C induce un subgraf complet al lui G
2. $\bigcap_{i \in C} A_i \neq \emptyset$
3. C este maximală (nu există o clică $D \subset V$, astfel încât $C \subset D$).

Teoremă

Fie $G=(V, U)$ un graf neorientat și $P=(P_1, P_2, \dots, P_m)$ o mulțime de proprietăți asociate vârfurilor. Submulțimea $C \subset V$ este clică în familia de grafuri induse dacă și numai dacă C este clică în G_A .

Modificarea algoritmului Bron-Kerbosch

Vom adapta definițiile mulțimilor CS , CA și NOT astfel încât să se adapteze definiției 4.

Mai exact, CS va fi o mulțime de vârfuri care induce un subgraf complet și care în plus respectă condiția 2. din definiția 4 ($\bigcap_{i \in CS} A_i \neq \emptyset$).

În mod analog, mulțimile CA și NOT conțin vârfuri adiacente cu toate vârfurile din CS și în plus, $\forall j \in CA \cup NOT, A_j \cap (\bigcap_{i \in CS} A_i) \neq \emptyset$

Definiție 5

Fie $i, j \in V$. Spunem că vârfurile i și j sunt adiacente relativ la proprietatea k dacă $[i, j] \in U$ și $k \in A_i \cap A_j$.

Condiția de marginire

Dacă există $v \in NOT$, astfel încât v este adiacent cu toate vârfurile din CA relativ la un atribut $k \in \bigcap_{i \in CS} A_i$, atunci orice extensie a mulțimii CS din acest punct nu va conduce la obținerea unei clici.