module – descrierea soluţiei

Asist. dr. ing. Mugurel Ionuţ Andreica

Vom aborda problema recursiv, definind o funcţie *num_modules(S)* ce calculează numărul de module, considerând că graful conţine doar nodurile din submulţimea *S*. *num_modules({1,...,N})* va întoarce rezultatul dorit.

Dacă subgraful format doar din nodurile din *S* (sau complementul acestuia) este format din mai multe componente conexe, atunci vom considera aceste componente independent şi vom calcula pentru fiecare numărul de module. În plus, putem forma şi module din combinaţii ale acestor componente (de ex., orice submulţime de componente împreună cu toate nodurile din acestea formează un modul).

Dacă subgraful format din nodurile din *S* este conex (şi la fel este şi complementul acestuia), atunci vom determina o mulţime de module maximale. Fie nodurile din *S* ordonate într-o ordine oarecare: *v(1), ..., v(|S|)*. Vom determina, pe rând, modulele maximale care nu conţin nodul *v(1)*, apoi pe cele ce îl conţin pe nodul *v(1)* dar nu îl conţin pe *v(2)*, apoi pe cele ce conţin nodurile *v(1)* şi *v(2)*, dar nu îl conţin pe *v(3)* ş.a.m.d. Pentru a determina modulele maximale ce conţin nodurile *v(1), ..., v(k-1)*, dar nu îl conţin pe nodul *v(k)*, vom împărţi toate nodurile diferite de *v(k)* în două submulţimi, în funcţie de adiacenţa (sau neadiacenţa) la *v(k)*. Aceste două submulţimi vor fi împărţite mai departe în mai multe submulţimi, în funcţie de adiacenţa (sau neadiacenţa) la nodurile din cealaltă submulţime ş.a.m.d. De fiecare dată vom continua doar cu submulţimile ce conţin nodurile *v(1), ..., v(k-1)* (pentru *k>1* există cel mult o astfel de submulţime cu care continuăm ; pentru *k=1* continuăm cu toate submulţimile obţinute). La final obţinem nişte submulţimi care sunt module maximale pentru condiţiile date (adică nu mai putem adăuga alte noduri pentru a forma un modul mai mare).

După considerarea tuturor cazurilor (*v(1), ..., v(|S|)*), este posibil ca unele module obţinute astfel să fie incluse în altele – vom verifica acest lucru şi vom elimina modulele incluse în alte module. Modulele rămase sunt disjuncte şi partiţionează complet nodurile grafului. Pentru fiecare modul *Smo* obţinut astfel vom calcula numărul de module inclus în acesta (utilizând apelul *num_modules(Smo)*). Modulele maximale astfel obţinute şi modulele care sunt submulţimi ale acestora nu pot fi combinate între ele pentru a forma module mai mari (ca în cazul componentelor conexe, prezentat mai sus).

O implementare directă a algoritmului prezentat (şi suficientă pentru obţinerea punctajului maxim) are complexitatea $O(N^4)$. Fiecare apel *num_modules(S)* poate considera $O(N)$ cazuri (pentru fiecare nod), iar fiecare caz poate fi tratat în timp $O(N^2)$ (deoarece fiecare pereche de noduri din graf este considerată o singură dată în tratarea cazului). Întrucât putem ajunge la $O(N)$ apeluri ale funcţiei *num_modules*, obţinem complexitatea menţionată.

Trebuie menţionat că există şi algoritmi optimi, de complexitate $O(N+M)$, care rezolvă această problemă.