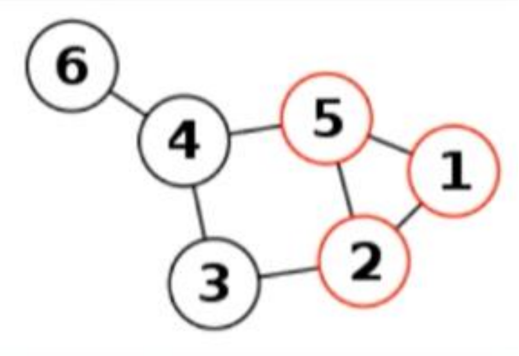
AA - Tema2 - Reducere k-Clique ≤p SAT

Student: Radu Sabina

Grupa:331CB

A.

Problema K-Clique este definita printr-un graf cu n noduri si m muchii si un nr k ce reprezinta dimensiunea subgrafului complet. Scopul acestei probleme este de a demonstra existenta unui astfel de subgraf in graful nostru.



Ex: In stanga avem un graf format din 6 noduri si 7 muchii. Putem observa faptul ca nodurile 1, 2 si 5 formeaza un subgraf total, adica o clica de dimensiune 3, deoarece toate nodurile sunt interconectate.

B.

Problema SAT este definita printr-o formulă booleană în formă normal conjunctivă. Scopul acestei probleme este de a verifica dacă există o interpretare care satisface formula.

Ex: Fie formula $(x1 \vee x2) \wedge (x2 \vee x3) \wedge (x1 \vee x3)$.

Putem observa ca exista o combinatie de valori booleene pentru x1, x2 si x3 astfel incat formula sa returneze true (x1 = True, x2 = True si x3 = False). De aceea, formula este satisfiabila.

C. Reducerea KClique la SAT

Ca pentru orice problema NP-Completa, se poate gasi o formula booleana prin care sa reducem KClique la SAT. Pentru generarea acesteia, vom impune 4 constrangeri:

1) Sunt alese k noduri dintre cele n ale grafului;
2) Un nod nu poate fi ales de doua ori;
3) Oricare 2 noduri dintre cele alese trebuie sa fie conectate;
4) O pozitie din clica nu poate fi ocupata de 2 noduri in acelasi timp;

1) Modul de gandire ales este de a asocia valoarea True unor k noduri. Astfel, fiecare nod poate fi scris sub forma a k codificari, ce reprezinta al catelea nod a fost ales. Se obtine apoi, pentru fiecare nod n, $x = n * k – i$, unde:
k = dimensiunea clicii si nr de elemente care alese;
i = indice de la 0 la n – 1, ce reprezinta cele k alegeri;

In primul rand, vom testa daca nu se aleg mai putin de k numere. Astfel, daca ne luam dupa ideea de mai sus, vom avea pentru fiecare nod k variabile x_{ij}, unde i reprezinta numarul nodului, iar j reprezinta al catelea este ales nodul respectiv. In final, pentru a ne asigura ca pentru fiecare din cele k pozitii exista un nod ales, vom avea formula:

$$(x_{10} \vee x_{20} \vee \ldots \vee x_{n0}) \wedge (x_{11} \vee x_{21} \vee \ldots \vee x_{n1}) \text{ etc.}$$

Pe scurt, fiecare paranteza reprezinta o alegere si testam daca fiecare dintre acestea are cel putin un reprezentant (vom constrange alegerea doar a unui singur nod mai incolo).

2) Cum un nod poate fi ales doar o singura data, obtinem formula: $(\sim x_{i0} \vee \sim x_{i1}) \wedge (\sim x_{i0} \vee \sim x_{i2}) \wedge \ldots \wedge (\sim x_{ik-2} \vee \sim x_{ik-1})$, pentru pentru $0 < i <= n$.

Astfel ne asiguram ca doar o singura variabila corespunzatoare unui nod va fi true.

3) Pentru cea de-a 3a constrangere, distingem 2 cazuri referitoare la existenta muchiilor dintre 2 noduri, fiecare cu cate 3 subcazuri:

daca amandoua nodurile sunt true si muchia exista => true;
daca amandoua nodurile sunt true si muchia nu exista => false
daca unul din noduri e true si muchia exista => true
daca unul din noduri e true si muchia nu exista => true
daca nici un nod nu e true si muchia exista => true
daca nici un nod nu e true si muchia nu exista => true

Putem observa faptul ca, in cazul in care muchia exista, rezultatul va fi mereu true. Astfel, acest caz devine irelevant. Ceea ce ne intereseaza este situatia in care ambele noduri sunt alese si muchia nu exista. In aceasta situatie, deducem formula urmatoare:

$$(\sim x_{i0} \lor \sim x_{i1} \lor \ldots \lor \sim x_{i(k-1)} \lor \sim x_{j0} \lor \sim x_{j1} \lor \ldots \lor \sim x_{j(k-1)})$$

pentru $0 < i <= n$, $0 < j <= n$ si $i != j$

4) Cum la pasul 1 am impus ca la fiecare alegere sa fie ales cel putin un nod, acum va trebui sa ne asiguram ca este ales doar unul singur. Astfel, deducem urmatoarea formula:

$$(x_{1j} \lor x_{2j} \lor \ldots \lor x_{nj}) \wedge (\sim x_{1j} \lor \sim x_{2j}) \wedge (\sim x_{1j} \lor x_{3j}) \wedge \ldots (\sim x_{n-1j} \lor \sim x_{nj}), \text{ pt } 0 <= j <= k - 1$$

Formula finala consta in concatenarea tuturor constrangerilor de mai sus.

Complexitate:

1) $O(n^2)$ pentru parcurgerea matricei de adiacenta.
2) $O(n^k)$ pentru prima constrangere
3) $O(n^k)$ pentru a doua constrangere
4) $O(n^p)$ pentru a 3a constrangere, unde $p = (m^2 - 2m – 1) / 2$ si reprezinta numarul de muchii dintr-un graf complet alcatuit din acelasi numar de nodurica si graful nostru, dar care nu se afla si in graful dat ca input.
5) $O(n^k)$ pentru a 4a constrangere
In concluzie, complexitatea va fi una polinomiala.