

Author: Mohamed MIMOUNI.

20 Rue kadissia Oujda 60000 Maroc.

Email1: mimouni.mohamed@gmail.com

Email2: m_mohamed64@yahoo.fr

Comments : 2 pages.

Subj – Class : La théorie de la complexité.

Abstract: Je donne l'algorithme polynomial pour résoudre CLIQUE.

Cet algorithme, permet de trouver la clique maximum dans un graphe simple non orienté, ce qui implique que $NP=P$.

Note:

La prouve qu'il s'agit d'un algorithme polynomial et universel sera au fur et à la mesure de la présentation de cet algorithme.

La clé

Soit **G** un graphe simple non orienté, **S** un sommet donner et **C** une clique dans **G**.

S pourra être ajouté à **C** ?

Ce problème de décision est un problème polynomial déterministe, parce que la réponse "oui" si tous les sommets de **C** sont relier a **S**, se qui est en temps polynomial, alors que la réponse "non" si au moins un sommet de **C** qui n'est pas relier à **S**.

L'algorithme

Soit **G** un graphe simple non orienter qui contient **n** sommets.

Chaque sommet pourra être au moins dans une clique. On note chaque clique par des nombres successifs.

Alors voici les étapes:

1^{er} partie

1. Le premier sommet S_1 sera dans le clique1.
2. On prend le deuxième S_2 , s'il est relié à S_1 il sera dans clique1, si non il sera dans clique2.
3. On prend le troisième S_3 , si il est relié à tous les sommets de clique1 il sera dans clique1, si il est relié à tous les sommets de clique2 il sera dans clique2, si non il sera dans clique3.
4. On fait se procédure pour tous les sommets: si un sommet est relié avec tous les sommets d'un même clique il sera ajouter à ce clique, si non il ne sera pas ajouter (il est en temps polynomial).
5. Faire les étapes de 1 à 4, une autre fois: pour assurer que chaque sommet est bien placé, dans toutes les cliques trouvées.

2^{ème} partie

La première partie, distribue les arrêts d'un graphe, dans des cliques séparer les un au autres. D'apprêt cette note, on obtient plusieurs cliques, mais la première partie ne détecte pas toutes les cliques, alors une deuxième procédure doit être. Cette fois on cherche un sous graphe de G , qui contient les arrêts qui ne sont pas dans les cliques trouvé dans la première partie.

Avec cette remarque, on fait l'algorithme une autre fois pour le sous graphe. Et on arrête quand on trouve un seul groupe-cliques.

La première partie, localise des cliques pseudo maximum¹, alors pour trouver les autres cliques, on cherche les arrêts qui ne se trouvent dans aucunes cliques.

La prouve

L'algorithme, a la fonction de trouver plusieurs cliques dans un graphe; la procédure se fait dans un temps polynomial, parce que a chaque tour on a un sommet et plusieurs cliques; l'algorithme test si on peut ajouter un sommet dans des cliques ou pas, ce qui est polynomiale.

Cependant il peut que pendant la première partie, on ne trouve pas la clique maximum. Des cliques non détecter par cet algorithme, et la solution est de trouver les arrêts qui ne sont pas traité et cela est polynomial.

Le cas le plus critique, est le cas au le graphe, ne peut pas être modifié. Dans cette cas on modifier l'ordre des traitements des sommets, d'une manière que S_1 et S_2 seront conjoint et que l'arrêt (S_1, S_2), n'est pas dans aucune clique trouver dans la première partie.

¹ Une clique dont on ne peut pas ajouter aucun sommet, mais qui n'est pas la clique maximum.