UNIVERSITÉ de Picardie Jules Verne

Problèmes « difficiles » et problème SAT

Dans la pratique, il y a de nombreux problèmes quotidiens qui peuvent être résolus rapidement (en un temps polynomial) par un algorithme, comme le tri d'un ensemble des entiers, la recherche du plus court chemin dans un réseau. Il est naturel de se demander si tous les problèmes peuvent être résolus rapidement. On ne connaît pas la réponse, car il existe certains problèmes comme la planification, la cryptographie, etc., qu'on ne sait pas résoudre à ce jour. Ces problèmes sont dites « difficiles ».

La question : **existe-il un algorithme rapide pour résoudre un problème « difficile »**, a été l'un des problèmes de recherche les plus étudiés et les plus troublants en informatique théorique, depuis qu'il a été posé en 1971 par Cook (« NP?=P »).

Parmi les problèmes « difficiles », il y a un fameux « problème SAT » qui est particulièrement intéressant, car tous les autres problèmes « difficiles » peuvent être encodés en SAT, et si le problème SAT est résolu rapidement, les autres problèmes « difficiles » peuvent être aussi résolus rapidement. Par conséquence, l'étude du problème SAT peut aider à avoir une meilleure compréhension aux problèmes « difficiles ».

Le problème SAT est représenté par une formule logique sous forme CNF. Une formule CNF est une conjonction de clauses, et une clause est une disjonction de variables booléennes (positifs ou négatifs). Une formule CNF est satisfaisable s'il est possible d'associer une valeur logique à chacune de ses variables booléennes de telle manière que cette formule soit logiquement vraie.

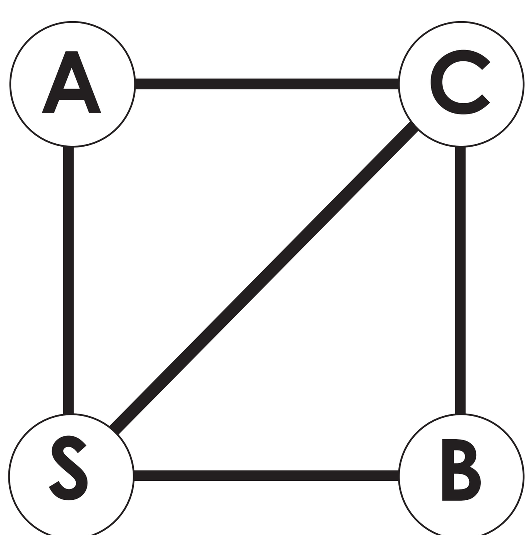
Exemple : planification des examens encodée en SAT

La planification générale est un problème « difficile » qui peut être encodé en SAT.

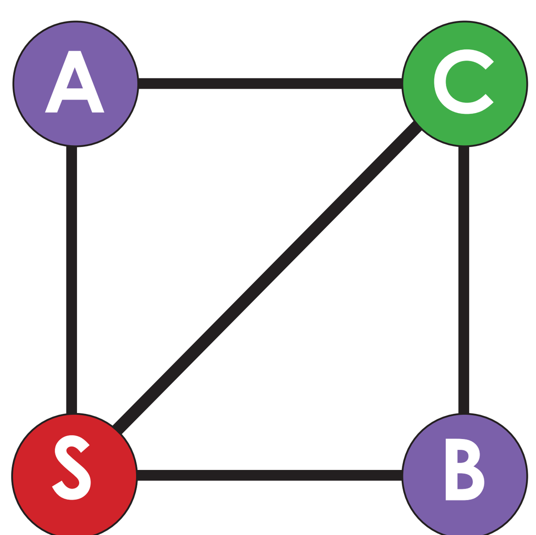
Étant donné 4 examens à planifier dans 3 créneaux :

- A = Anglais,
- B = Base de données,
- C = Langage C,
- S = Système d'exploitation.

On cherche des créneaux horaires où les contraintes sont : deux examens reliés dans graphe (1) ne peuvent se dérouler en même temps.



(1)



(2)

Poser des variables booléennes pour représenter si un examen est placé dans un créneau, e.g. pour A :

- a_1=1, A est placé dans créneau 1 ; sinon a_1=0,
- a_2=1, A est placé dans créneau 2 ; sinon a_2=0,
- a_3=1, A est placé dans créneau 3 ; sinon a_3=0.

Établir des clauses pour représenter qu'un examen ne peut que être placé dans un seul créneau en respectant des contraintes, e.g. pour A :

➔ $(a_1 \vee a_2 \vee a_3) \wedge (\neg a_1 \vee \neg a_2) \wedge (\neg a_1 \vee \neg a_3) \wedge (\neg a_2 \vee \neg a_3)$
$\wedge (\neg a_1 \vee \neg c_1) \wedge (\neg a_2 \vee \neg c_2) \wedge (\neg a_3 \vee \neg c_3)$
$\wedge (\neg a_1 \vee \neg s_1) \wedge (\neg a_2 \vee \neg s_2) \wedge (\neg a_3 \vee \neg s_3)$

Obtenir ainsi une formule CNF. Si cette formule est satisfaisable, e.g. avec a_3=1, b_3=1, c_2=1, s_1=1, et les autres variables booléennes étant égales à 0, la planification de ces examens est résolue, représentée par graphe (2) :

- créneau 1 = rouge,
- créneau 2 = vert,
- créneau 3 = violet.

Réalisé par l'unité de recherche Modélisation, Information et Système (MIS, EA 4290) - UPJV - Octobre 2012