

# Complexité et Calculabilité 2016, Coloration de graphe

Paul Dorbec, Anca Muscholl, Mathias Petreolle, Thomas Place

Le problème de coloration d'un graphe est un problème classique de théorie des graphes, connu comme étant NP-complet dans le cas général. Nous souhaitons dans ce devoir utiliser des solveurs puissants pour résoudre des formules SAT afin de traiter des problèmes de coloration de graphe.

## 1 Définition du problème

Pour un graphe donné et un entier  $k$ , on souhaite savoir s'il est possible d'attribuer une couleur parmi  $k$  à chacun des sommets du graphe de telle sorte que les extrémités de chaque arête aient des couleurs différentes. Si la définition de la coloration d'un graphe n'est pas claire pour vous, nous vous invitons à consulter la page correspondante de Wikipedia<sup>1</sup>.

## 2 Format des graphes

Pour vous épargner le travail consistant à écrire un parser de graphe et à créer une structure de donnée graphe, vous permettant ainsi de vous focaliser sur l'efficacité de votre réduction, nous vous fournissons des fichiers C comportant chacun les fonctions suivantes :

```
int orderG();  
int sizeG();
```

vous donnent respectivement le nombre de sommets et d'arêtes du graphe,

```
int are_adjacent(int u, int v);
```

---

1. [https://fr.wikipedia.org/wiki/Coloration\\_de\\_graphe](https://fr.wikipedia.org/wiki/Coloration_de_graphe)

prend en paramètre deux numéros de sommet (entre 0 et `nb_sommet()-1`) et renvoie 1 si  $u$  et  $v$  sont adjacents, 0 sinon.

Étant donné un graphe et un nombre de couleurs  $k$ , vous devez générer une formule SAT qui représente l’assertion “le graphe fourni est coloriable en  $k$  couleurs” pour la donner à un solveur, qui décidera pour vous si le graphe est ou n’est pas coloriable. En outre, si le graphe est coloriable et que le solveur retourne une affectation des variables qui satisfait la formule, vous devrez transposer cette affectation en une coloration du graphe.

Des exemples de fichiers C traduisant des graphes seront proposés sur la page suivante : <http://www.labri.fr/perso/dorbec/CoCa/>. Ils vous permettront de tester votre codage.

### 3 Solveur SAT

Le solveur que vous allez utiliser est **Glucose**, un solveur basé sur **Minisat** développé par G. Audemard et L. Simon, dont voici la page web<sup>2</sup>. Le format d’entrée de ce programme est un format DIMACS, utilisé pour les compétitions. Vous pouvez en trouver une description ici<sup>3</sup>.

Le principe est simple, le contenu d’un fichier doit ressembler à cela :

```
c
c start with comments
c
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

où les lignes au début commençant par ‘c’ sont des commentaires, puis une ligne `p cnf nbvar nbclauses` qui annonce le nombre de variables et de clauses, enfin une ligne pour chaque clause (terminant par 0), indiquant les numéros des variables contenues dans les clauses précédées du signe - si la négation est utilisée.

---

2. <http://www.labri.fr/perso/lSimon/glucose/>

3. <http://www.satcompetition.org/2009/format-benchmarks2009.html>

## 4 Évaluation

Vous réaliserez le travail par binômes. Vous remettrez par mail votre code à vos responsables au plus tard le 25 novembre, ainsi qu'une description de la réduction que vous utilisez et des optimisations que vous avez mises en place. Les évaluations auront lieu la semaine suivante, du 28 novembre. Votre code sera testé sur les machines du CREMI et devra donc pouvoir s'y exécuter.

Pour tester le code, nous vous fournirons de nouveaux programmes de génération de graphes de plus en plus gros pour tester les limites de votre codage.