

3-Coloration de graphes

K. Chai, M. Lacote, D. Lesnoff, I. Martayan, C. Zeng

Ecole d'été de mathématique et d'informatique

30 août 2019

Mathinfo



- 1 Introduction
 - Présentation du problème
- 2 Formalisation
 - Formalisation en logique du premier ordre
- 3 Modèle pour SAT-Solver
 - Variables et clauses
- 4 Fonctionnement du programme
 - Structure du projet
 - Type de données et algorithmes
 - Performances
- 5 Résultats
 - Résultats du programme
 - Conclusion et autres applications

Définition

Un *graphe* est la donnée d'un couple (V, E) composé de :

- V d'un ensemble de *sommets* ou *noeuds* ;
- $E \subseteq \{\{x, y\} | (x, y) \in V^2 \wedge x \neq y\}$, d'un ensemble d'arêtes, qui sont des paires de sommets.

Un graphe *orienté* possède des arêtes ayant une orientation.

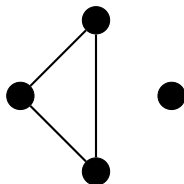


Figure 1 – Un graphe avec $|V| = 4$ et $|E| = 3$

Problème de k -coloration

Etant donné un graphe non orienté $G = (V, E)$, et un entier $k \geq 1$, existe-t-il une *coloration* $c : V \rightarrow \{1, 2, \dots, k\}$ telle que :

$$\forall \{u, v\} \in E, c(u) \neq c(v) ?$$

On se place dans le cas $k = 3$.

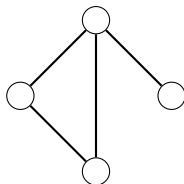


Figure 2 – Exemple de graphe à colorer

3-coloration : exemples

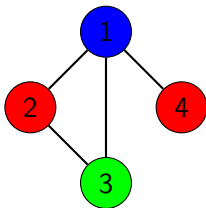


Figure 3 – 3-coloriage avec $|V| = 4$ et $|E| = 4$

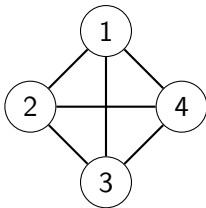


Figure 4 – 3-coloriage impossible avec $|V| = 4$ et $|E| = 6$



Notations

Un graphe *non orienté* $G = (V, E)$:

- $n := |V|$ le nombre de noeuds ;
- $p := |E|$ le nombre d'arêtes ;

Signature

Nous considérons la signature $\Sigma = \{e^{r2}, c^{r2}, n^{f0}, p^{f0}\}$ ayant la sémantique suivante :

- $e(x, y)$: x est voisin de y ;
- $c(x, j)$: x a la couleur j .

Contraintes

- 1 Tout sommet du graphe est coloré ;

Expressions en logique du premier ordre

- 1 $\forall x \in \llbracket 1, n \rrbracket, \exists j \in \llbracket 1, k \rrbracket, c(x, j) ;$

Contraintes

- 1 Tout sommet du graphe est coloré ;
- 2 Tout sommet possède au plus une couleur ;

Expressions en logique du premier ordre

- 1 $\forall x \in \llbracket 1, n \rrbracket, \exists j \in \llbracket 1, k \rrbracket, c(x, j) ;$
- 2 $\forall x \in \llbracket 1, n \rrbracket, \exists i \in \llbracket 1, k \rrbracket, (c(x, i) \implies (\bigwedge_{j \neq i} \neg c(x, j))) ;$

Contraintes

- 1 Tout sommet du graphe est coloré ;
- 2 Tout sommet possède au plus une couleur ;
- 3 Deux sommets voisins ont des couleurs différentes.

Expressions en logique du premier ordre

- 1 $\forall x \in \llbracket 1, n \rrbracket, \exists j \in \llbracket 1, k \rrbracket, c(x, j) ;$
- 2 $\forall x \in \llbracket 1, n \rrbracket, \exists i \in \llbracket 1, k \rrbracket, (c(x, i) \implies (\bigwedge_{j \neq i} \neg c(x, j))) ;$
- 3 $\forall x, y \in \llbracket n \rrbracket, \forall i \in \llbracket 1, k \rrbracket, \neg (e(x, y) \wedge (c(x, i) \wedge c(y, i))) .$

Fichier d'entrée

- n et p sur la première ligne
- p lignes décrivant les paires de sommets reliées par une arête

Variables

- $3n$ variables $x_{(i-1)n+(j-1)}$ où $i \in \llbracket 1, n \rrbracket$, $j \in \llbracket 1, 3 \rrbracket$

Nombre de clauses

- $27n + 3p$ clauses

Programme OCaml

- Lecture du fichier d'entrée
- Génération de la forme normale conjonctive

Programme OCaml

- Lecture du fichier d'entrée
- Génération de la forme normale conjonctive

Programme Python

- Appel du programme OCaml
- Appel du SAT solveur
- Affichage du graphe à partir de la solution

```
type formule =  
  | Var of int  
  | Non of formule  
  | Et of formule * formule  
  | Ou of formule * formule;;
```

- 1 Traduction récursive en forme normale conjonctive
- 2 Réécriture avec des fonctions récursives terminales
- 3 Généralisation pour le problème de k -coloration

Performances du programme

Amélioration du modèle

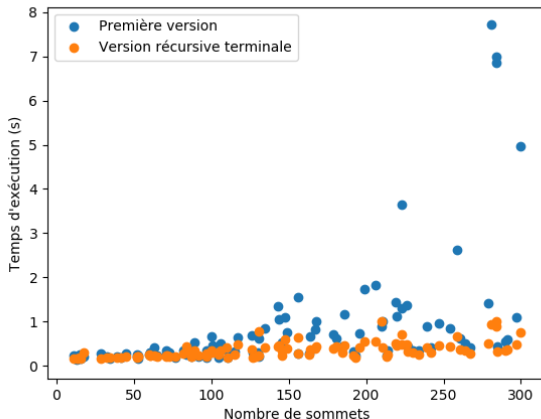


Figure 5 – Temps d'exécution en fonction de n



Performances du programme

Amélioration du modèle

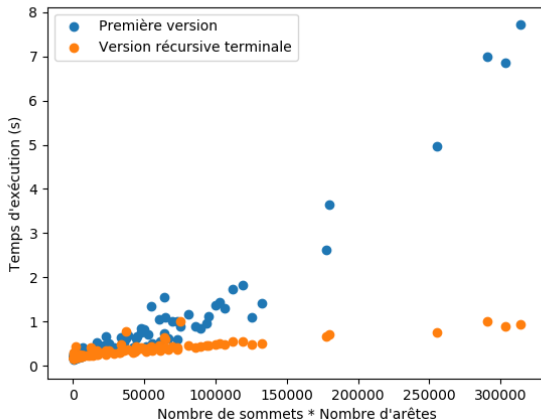


Figure 6 – Temps d'exécution en fonction de np



Performances du programme

Limites du modèle

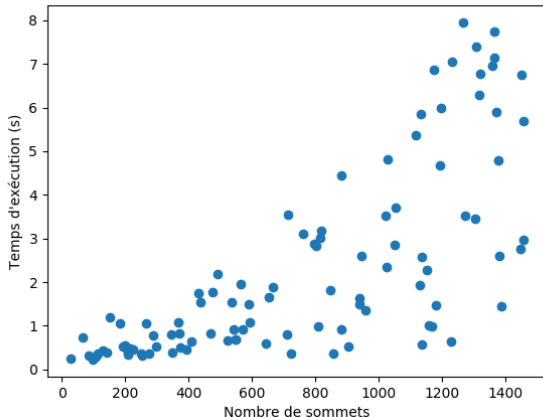


Figure 7 – Temps d'exécution en fonction de n



Performances du programme

Limites du modèle

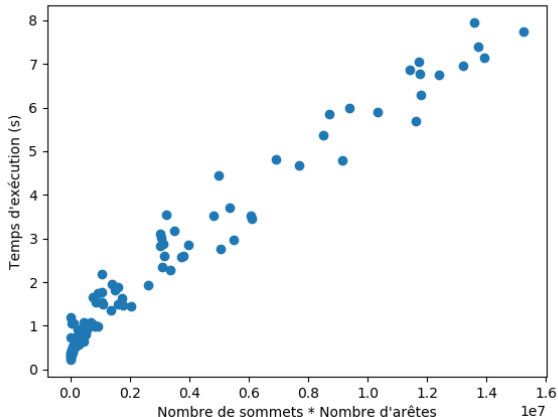


Figure 8 – Temps d'exécution en fonction de np



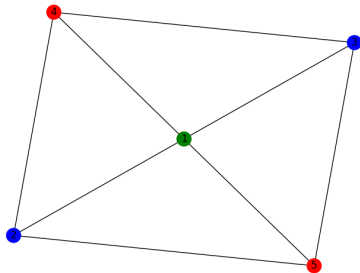


Figure 9 – Une solution pour $n = 5$ et $p = 8$

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

- 9 couleurs
- 81 sommets
- lignes, colonnes et blocs reliés

Figure 10 – Une grille de sudoku



P. Lafourcade.

Cours de logique.

Ecole d'été de mathématique et d'informatique - Mathinfoly, 2019.



D. Babic.

Satisfiability Suggested Format.

Domagoj-Babic-Website, 1993.



Wikipedia

Boolean satisfiability problem.

Wikipedia page, 2019.