

Projet : Coloration des Graphes

Analyse, Visualisation et Comparaison d'Algorithmes

Nom et Prénom

December 12, 2025

Contents

1	Introduction	2
2	Coloration des sommets	2
2.1	Principe	2
2.2	Exemples	2
2.2.1	Graphe cycle	2
2.2.2	Graphe aléatoire (Erdős–Rényi)	3
2.2.3	Graphe biparti	3
3	Coloration des arêtes	4
3.1	Principe	4
3.2	Exemples	4
3.2.1	Graphe cycle	4
3.2.2	Graphe aléatoire (Erdős–Rényi)	5
3.2.3	Graphe biparti	5
4	Algorithmes de coloration	5
4.1	Greedy	5
4.2	Welsh–Powell	6
4.3	DSATUR	6
5	Comparaison des algorithmes	6
6	Conclusion	6

1 Introduction

La coloration des graphes est un problème classique en informatique. Il s'agit d'attribuer des couleurs aux sommets ou aux arêtes d'un graphe tout en respectant certaines contraintes. La coloration est utilisée dans plusieurs domaines :

- planification (emplois du temps),
- allocation de ressources,
- réseaux et communication,
- compilation (coloration de registres).

Le but est souvent de minimiser le nombre total de couleurs utilisées.

2 Coloration des sommets

2.1 Principe

On cherche à colorier les sommets d'un graphe de sorte que deux sommets adjacents ne possèdent pas la même couleur.

2.2 Exemples

2.2.1 Graphe cycle

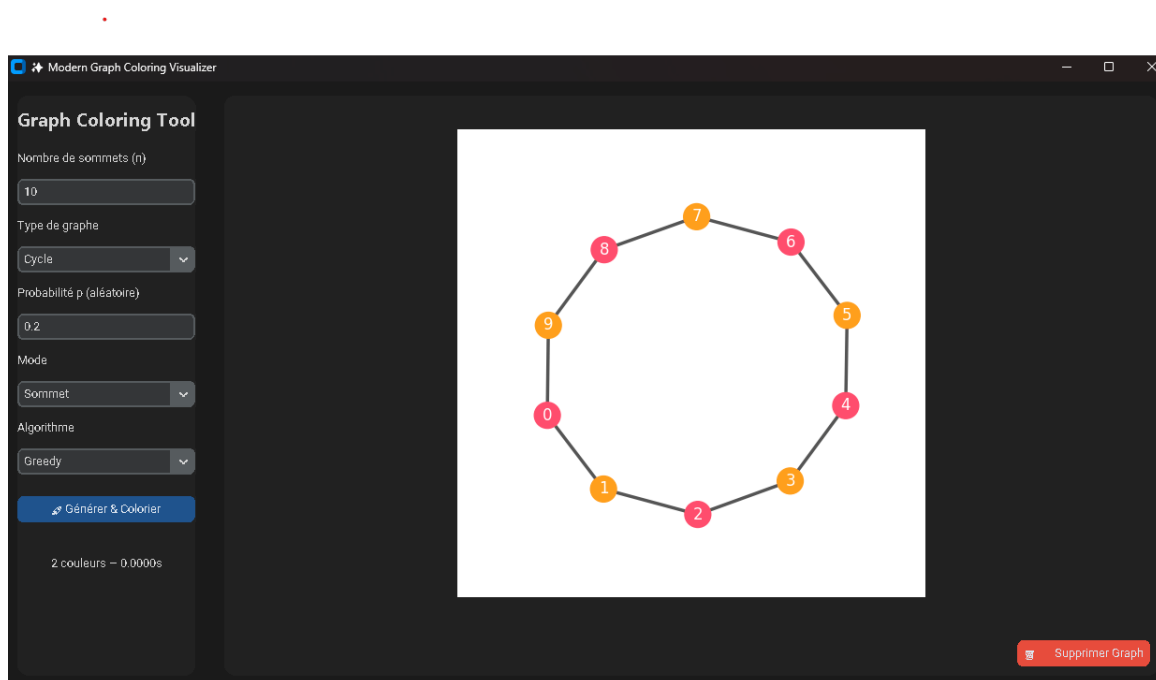


Figure 1: Coloration des sommets d'un graphe cycle.

Commentaire : Les cycles pairs utilisent 2 couleurs. Les cycles impairs nécessitent 3 couleurs.

2.2.2 Graphe aléatoire (Erdős–Rényi)

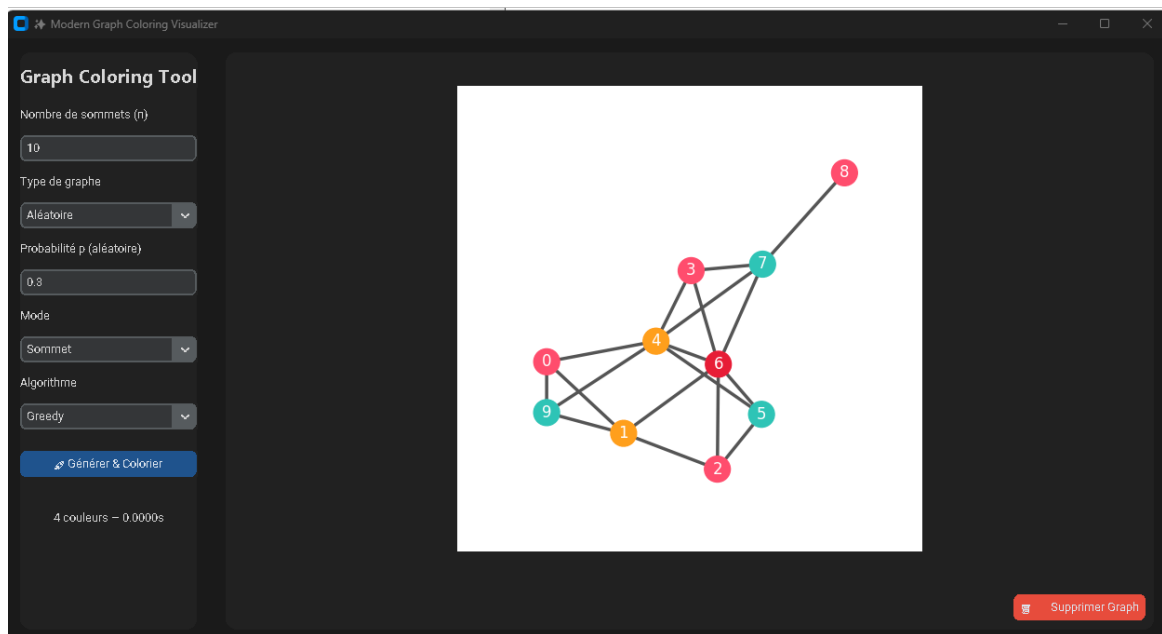


Figure 2: Coloration d'un graphe aléatoire avec probabilité d'arêtes $p = 0.3$.

Commentaire : Le nombre de couleurs dépend de la densité du graphe.

2.2.3 Graphe biparti

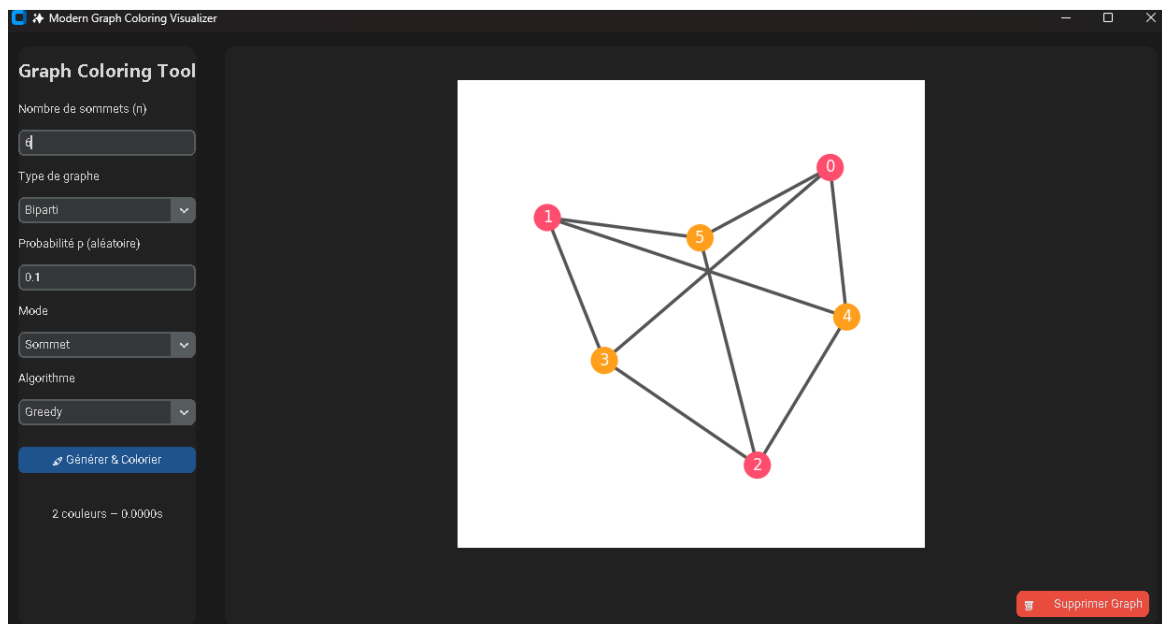


Figure 3: Coloration d'un graphe biparti complet.

Commentaire : Un graphe biparti est toujours 2-coloriable.

3 Coloration des arêtes

3.1 Principe

Deux arêtes incidentes ne peuvent pas partager la même couleur. Une méthode courante consiste à utiliser le *graphe des lignes*, où chaque arête devient un sommet.

3.2 Exemples

3.2.1 Graphe cycle

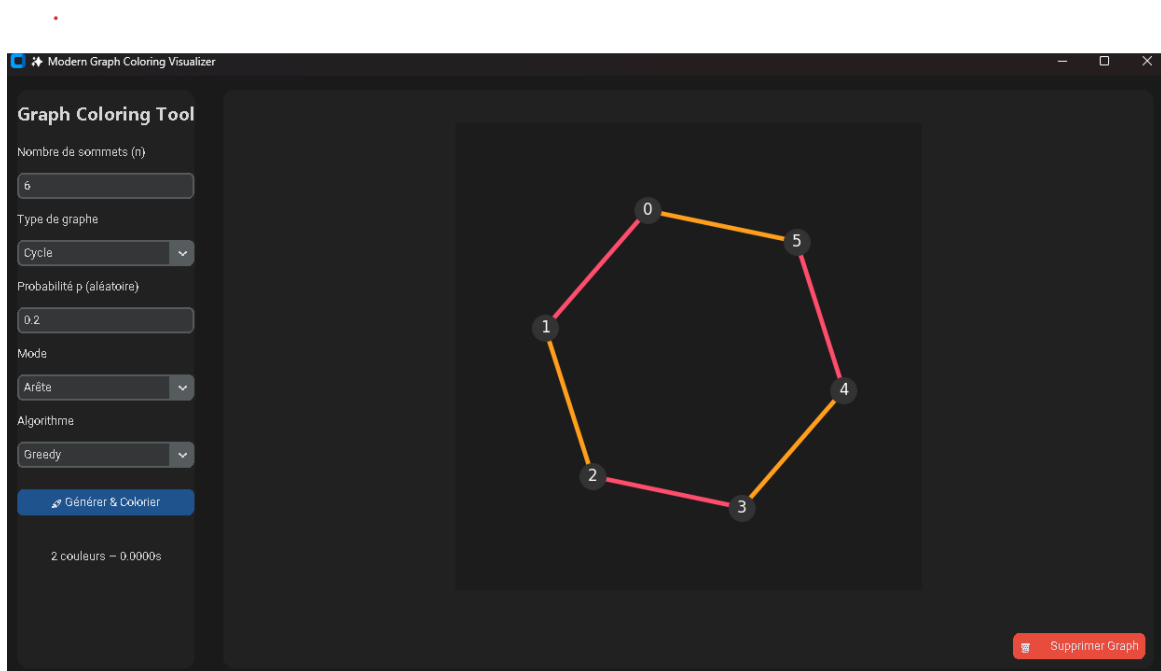


Figure 4: Coloration des arêtes d'un cycle.

3.2.2 Graphe aléatoire (Erdős–Rényi)

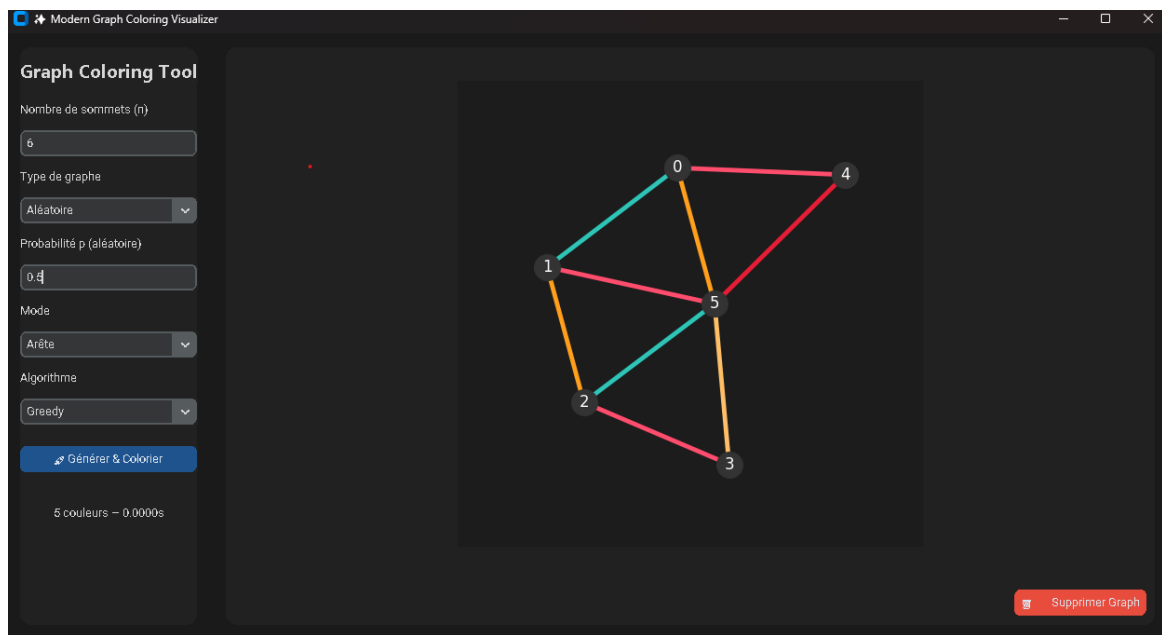


Figure 5: Coloration des arêtes d'un graphe aléatoire.

3.2.3 Graphe biparti

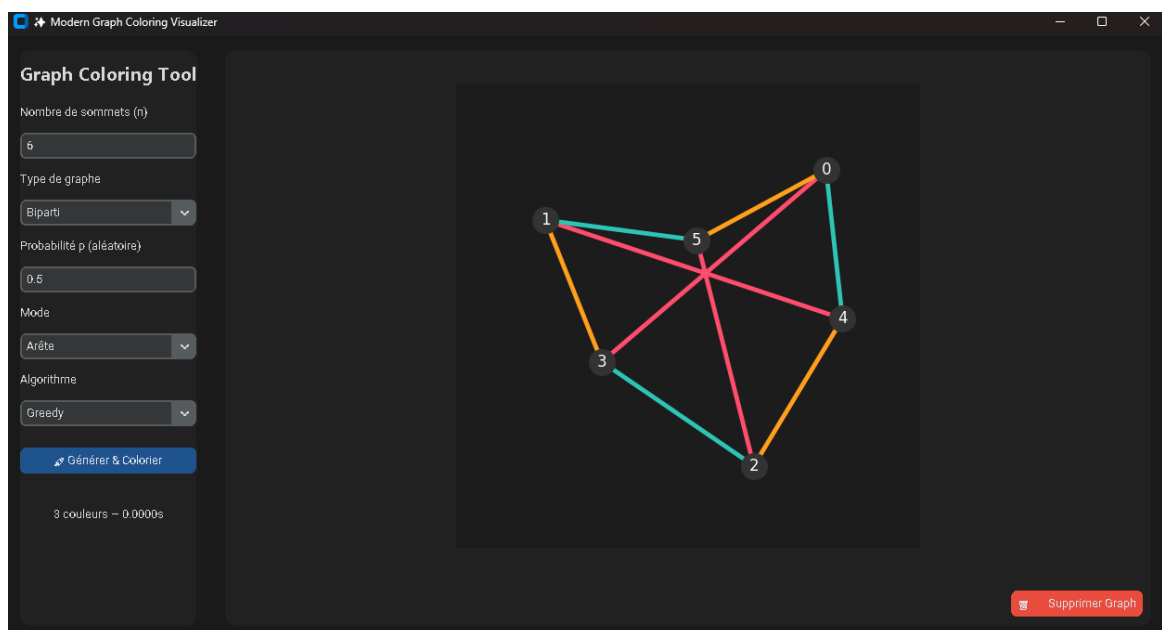


Figure 6: Coloration des arêtes d'un graphe biparti complet.

4 Algorithmes de coloration

4.1 Greedy

On parcourt les sommets dans un ordre fixe et on attribue la plus petite couleur disponible.

Avantages :

- très simple,
- temps d'exécution faible.

Inconvénients :

- résultat dépend fortement de l'ordre des sommets.

4.2 Welsh–Powell

Les sommets sont triés par degré décroissant avant d'appliquer Greedy.

Avantages :

- améliore généralement le résultat,
- efficace pour les graphes denses.

4.3 DSATUR

À chaque étape, on choisit le sommet ayant la plus grande saturation (nombre de couleurs différentes dans son voisinage).

Avantages :

- très bon résultat,
- souvent optimal.

Inconvénients : temps de calcul plus élevé.

5 Comparaison des algorithmes

Algorithme	Critère de sélection	Complexité	Nombre de couleurs
Greedy	Ordre fixe	Faible	Moyen à élevé
Welsh–Powell	Degré décroissant	Moyenne	Moyen
DSATUR	Saturation maximale	Plus élevée	Faible

Table 1: Comparaison des trois algorithmes de coloration.

6 Conclusion

Dans ce projet, nous avons étudié plusieurs méthodes de coloration et observé leur comportement sur différents types de graphes. DSATUR fournit généralement les meilleurs résultats, tandis que Greedy reste la méthode la plus rapide. La visualisation permet de mieux comprendre les étapes successives de coloration.