Axel BERTRAND
Pierre-Elliott THIBOUD

Rapport de projet

Coloration de graphe

Description du projet

Pour ce projet, nous devions implémenter une structure de graphe simple ainsi que trois algorithmes (Greedy, WelshPowell et Dsatur) afin de réaliser la coloration de plusieurs graphes (soient : Crown 10, Queen 5x5, Queen 7x7, Queen 9x9, Queen 11x11, Queen 13x13, Queen 15x15).
Nous avons donc choisi d'implémenter une structure de graphe reposant sur des listes, qui sont plus simples et efficaces à parcourir.
Nous avons ainsi implémenté les classes :

- Noeud : décrivant la structure d'un Noeud, avec une liste d'adjacence, une couleur (représentée par un entier) et un numéro (pour faciliter la lisibilité des résultats)
- Graphe : contenant une liste de sommets, un nombre chromatique (généré par les algorithmes) et des méthodes pour lire les fichiers contenant les graphes
- Une classe statique contenant les trois algorithmes (sous forme de méthodes statiques) et les différentes méthodes de benchmark des résultats

Afin d'éprouver les algorithmes, nous avons réalisé un benchmark de chaque algorithme sur tous les graphes disponibles triés des trois manières possibles (décroissant, croissant et aléatoire) sur 5 000 essais.

Les tableaux contenant les valeurs (min, moy, max pour le temps et le nombre chromatique) de tous nos tests sont disponibles en annexe.

Description des algorithmes

Pour chacun des algorithmes, on crée une liste qui contiendra les sommets colorés. À chaque fois que l'on colore un sommet, il est ajouté à cette liste et supprimé de la liste de sommets initiale.

Greedy

Lors de la coloration d'un sommet, on récupère la liste des couleurs interdites, c'est-à-dire les couleurs données aux sommets adjacents, et on compte le nombre de couleurs distinctes. On obtient ainsi la plus petite couleur pouvant être donnée au sommet à colorer et on répète ceci pour tous les sommets dans leur ordre d'arrivée.

Complexité de notre implémentation : O(n * k), avec k le nombre de sommets adjacents

Welsh Powell

Avant d'entrer dans la deuxième boucle (celle parcourant les sommets du graphe), on conserve l'indice du sommet y, puis si y n'a pas de sommets adjacents avec la couleur k, on retire ce sommet de la liste L ce qui a pour effet de déplacer les sommets restant dans la liste, donc le sommet suivant y se retrouve au même indice que l'était y. En revanche, s'il y a un sommet adjacent avec la couleur k alors on prend l'indice suivant.

Complexité de notre implémentation : O(m * n)

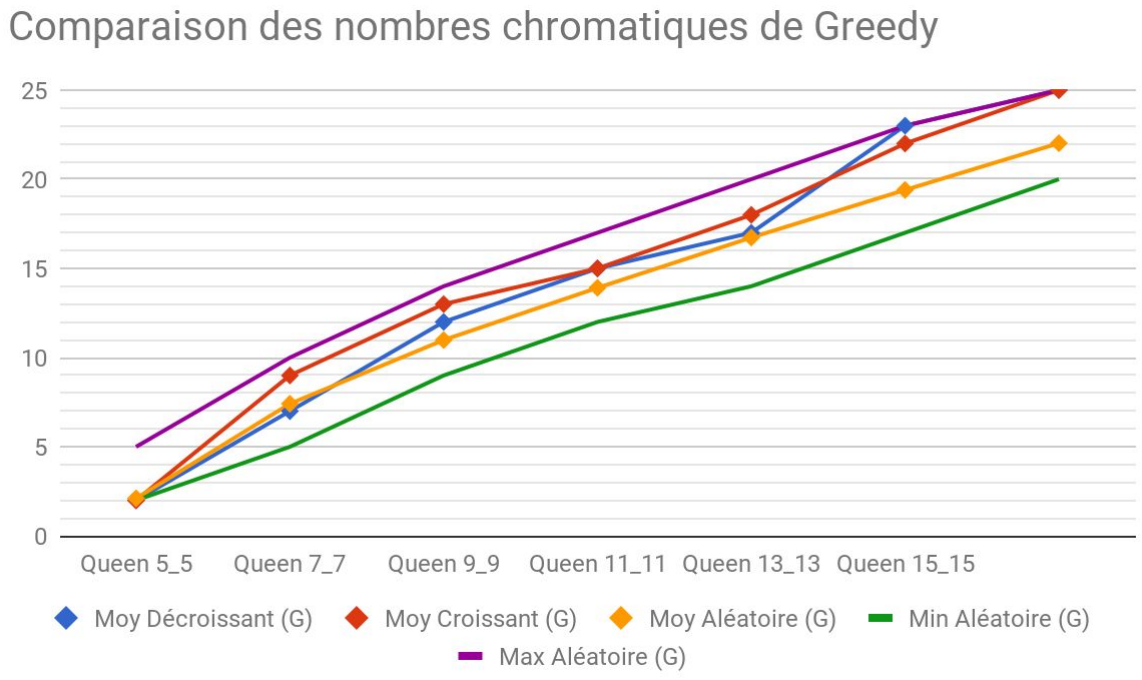
Dsatur

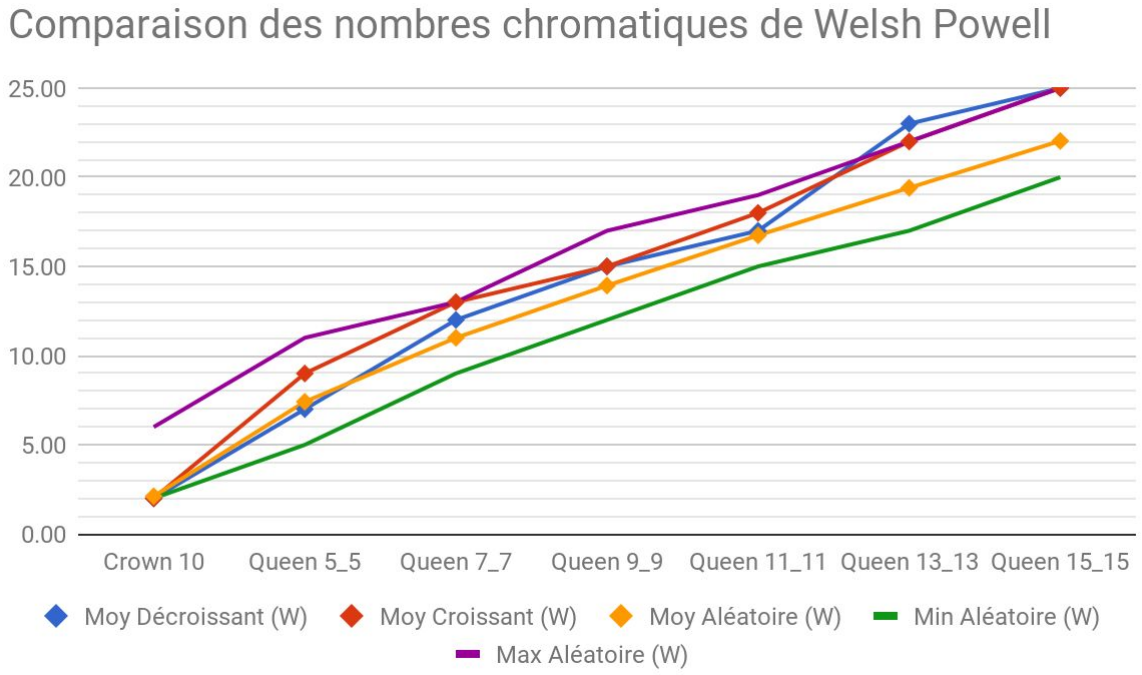
Nous nous sommes basés sur l'algorithme greedy pour effectuer l'heuristique Dsatur. Au lieu de prendre le premier sommet de la liste L comme dans Greedy, on choisi le sommet avec la plus grande valeur de saturation, en sachant que cette valeur correspond au nombre de couleur différentes sur les sommets adjacents.

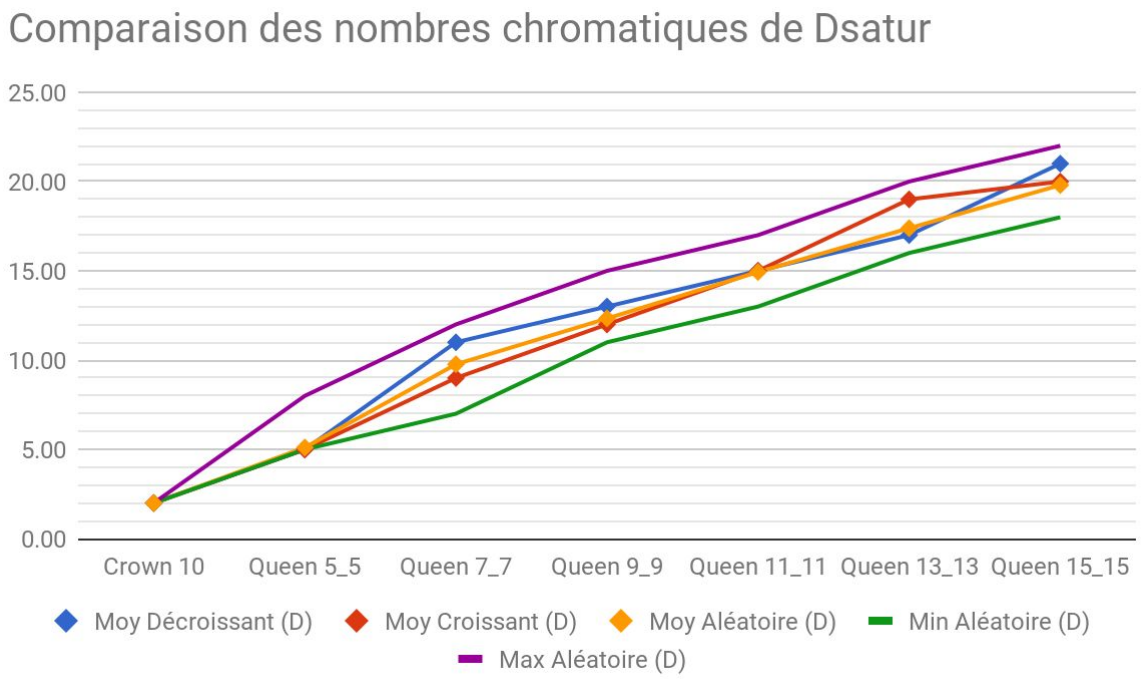
Complexité de notre implémentation : O(n² * k), avec k le nombre de sommets adjacents

Résultats des algorithmes

Les Min et Max des nombres chromatiques des tris croissant et décroissant ne sont pas affichés sur les graphiques car ils sont identiques au nombre chromatique moyen (pour tous les algorithmes).




Comparaison des nombres chromatiques de Welsh Powell
25.00
20.00
15.00
10.00
5.00
0.00
Crown 10
Queen 5_5
Queen 7_7
Queen 9_9
Queen 11_11
Queen 13_13
Queen 15_15
Moy Décroissant (W)
Moy Croissant (W)
Moy Aléatoire (W)
Min Aléatoire (W)
Max Aléatoire (W)


Comparaison des nombres chromatiques de Dsatur
25.00
20.00
15.00
10.00
5.00
0.00
Crown 10
Queen 5_5
Queen 7_7
Queen 9_9
Queen 11_11
Queen 13_13
Queen 15_15
Moy Décroissant (D)
Moy Croissant (D)
Moy Aléatoire (D)
Min Aléatoire (D)
Max Aléatoire (D)

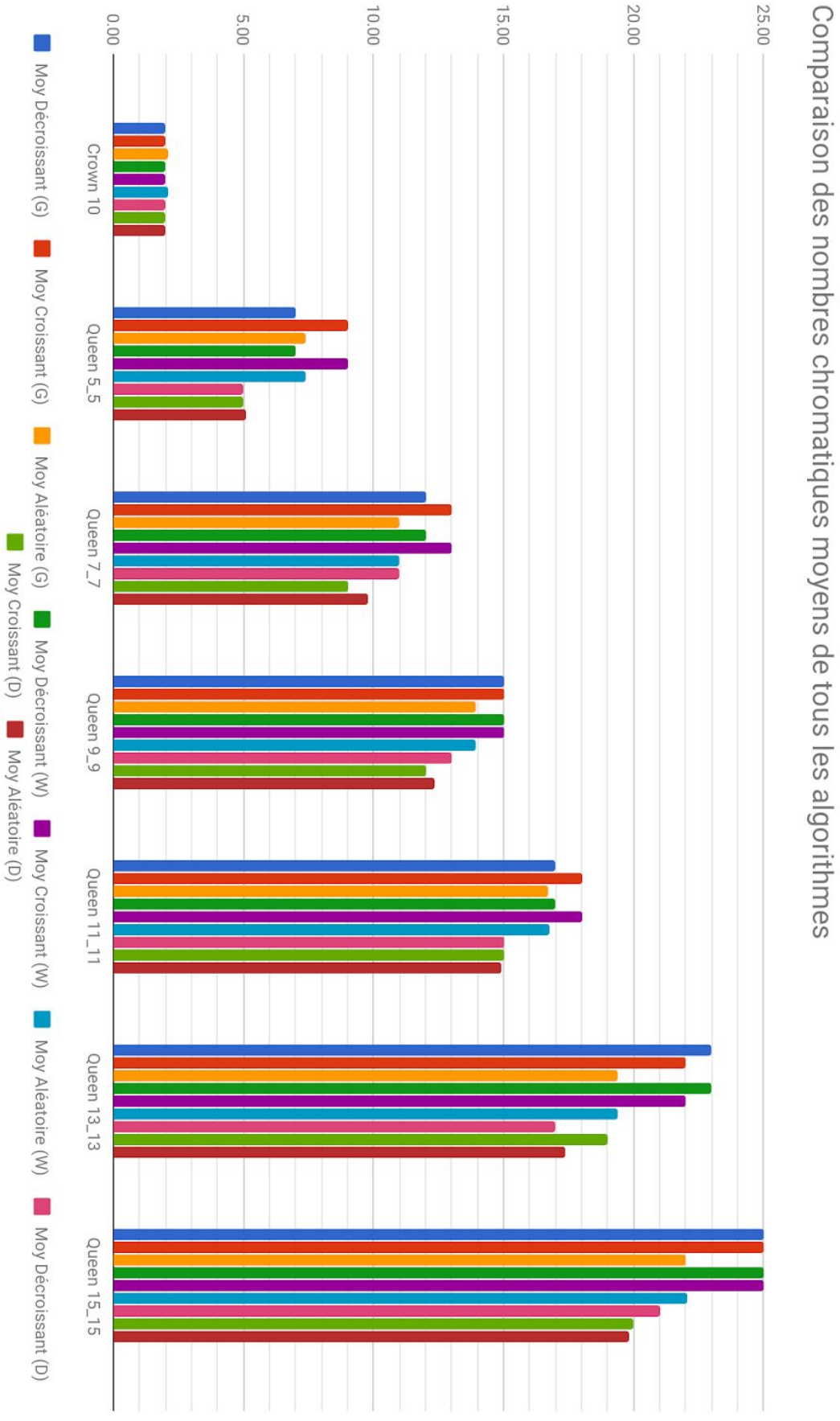
En analysant ces graphiques, on peut remarquer que les algorithmes Greedy et Welsh Powell sont en moyenne plus efficace lorsque les sommets sont triés aléatoirement, conclusion qui est beaucoup moins évidente pour Dsatur.
Pour ce dernier, le tri aléatoire produit un résultat presque toujours identique, voire moins bon, que le tri croissant.
Ce tri aléatoire permet également de trouver la meilleure coloration possible du graphe par l'algorithme (et non pas la meilleure coloration tout court). Il s'avère tout particulièrement efficace pour les graphes avec beaucoup de sommets pour les algorithmes Greedy et Welsh Powell, car ses résultats sont équivalents aux tris croissant et décroissant dans le pire des cas (courbe violette).

Ces graphiques nous permettent également de remarquer que les tris croissant et décroissant sont globalement équivalents, et ce, pour tous les algorithmes.

Comparaison des trois algorithmes

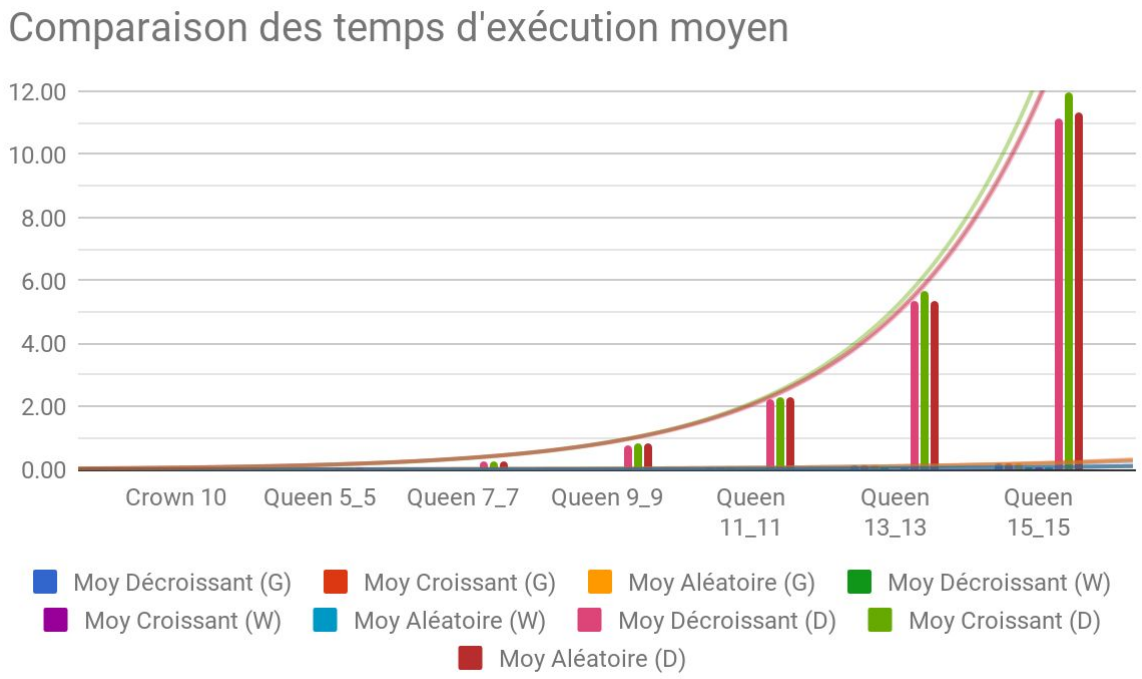
Cette comparaison se base sur le graphique ci-après, comparant les nombres chromatiques moyens obtenus par les trois algorithmes pour les trois types de tris. Les lettres G, W et D précisent l'algorithme utilisé pour obtenir ce résultat (respectivement Greedy, WelshPowell, Dsatur).

Nous pouvons constater grâce à ce graphique que l'algorithme Dsatur est le plus performant, peu importe le tri utilisé. En revanche, les algorithmes Greedy et Welsh Powell sont identiques peu importe le graphe ou le tri utilisé.
Les différences de résultats entre les algorithmes apparaissent très rapidement et s'accentuent avec l'augmentation du nombre de sommets.


Comparaison des nombres chromatiques moyens de tous les algorithmes
25.00
20.00
15.00
10.00
5.00
0.00
Crown 10
Queen 5_5
Queen 7_7
Queen 9_9
Queen 11_11
Queen 13_13
Queen 15_15
Moy Décroissant (G)
Moy Croissant (G)
Moy Aléatoire (G)
Moy Décroissant (W)
Moy Croissant (W)
Moy Aléatoire (W)
Moy Décroissant (D)
Moy Croissant (D)
Moy Aléatoire (D)

Temps d’exécution

En plus des comparaisons sur les nombres chromatiques obtenus, nous avons également comparé les temps d’exécution moyens des algorithmes sur les différents graphes (voir graphique ci-dessous).



Grâce à ce graphique, on remarque aisément que la complexité en n² de Dsatur fait croître son temps d’exécution de manière exponentielle. Alors que la différence en temps d’exécution pour Greedy et WelshPowell est presque imperceptible par rapport à l’évolution de la complexité des graphes colorés.

Conclusion

Si on recherche la coloration la plus optimale possible, sans aucune limite de temps ni de calcul, alors l’algorithme Dsatur avec un tri aléatoire des sommets permettra d’obtenir le meilleur résultat.
En revanche, dans le cas où l’on aurait besoin d’un résultat pas forcément optimal mais avec de fortes contraintes de temps et de calcul, il vaut mieux utiliser l’algorithme de Welsh Powell avec un tri aléatoire des sommets. C’est en effet le plus rapide des trois et l’algorithme pour lequel le tri aléatoire est le plus efficace et avec le moins d’inconvénients.