# Coloration de graphe

## Cadre

Un graphe est un objet mathématique décrit par deux ensembles, formellement noté  :

* (vertex en anglais) : Un ensemble de sommets ;
* (edge en anglais) : Un ensemble d’arcs inclus dans .

Graphiquement, un graphe se représente avec des ronds (symbolisant les sommets) et des flèches reliant des sommets et symbolisant les arcs.

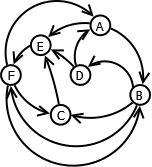


Figure 1- Exemple de graphe

L’objectif du problème de coloration de graphe et de trouver une affectation des sommets du graphe de sortes que deux sommets reliées par un arc soit affectées dans des groupes différents et que le **nombre de groupes** soit le plus petit possible. Les groupes peuvent être identifier à une couleur ; d’où le nom de problème de coloration de graphe. La Figure 2 donne un exemple de coloration du graphe de la Figure 1. Dans la pratique, le sens de l’arc n’importe pas et on pourra considérer les arcs non-orientés.

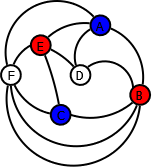


Figure 2-Exemple de coloration. On utilise ici trois couleurs.

Ce problème théorique présente de nombreux applications (optimisation lors de la compilation de programme, transport de matière première, …). Ce problème et l’approche de Welsh-Powell ont été au programme de spécialité - mathématiques des séries ES. Vous trouverez des informations sur internet à son propos[[1]](#footnote-1).

## Travail à réaliser

1. Se documenter :
   1. Vous trouverez des informations sur internet (le problème posé étant célèbre et la méthode étant classique de nombreux pages web/tuto présentent l’algorithme) ;
   2. Pensez à regarder aussi dans ouvrage de références (livre de références, manuels scolaire, …) ;
   3. Garder les références (**Attention** : du fait de la profusion des sources vous risquez de ne plus savoir d’où viennent vos connaissances lors de l’écriture du rapport) ;
   4. Discuter et critiquer vos sources avec professeur et vos camarades.
2. Implanter une structure de donnée permettant de gérer un (des) graphe(s). Vous devez notamment savoir :
   1. Lire un fichier contenant un graphe dans un format simple ;
   2. Ecrire un graphe dans un format simple dans un fichier[[2]](#footnote-2) ;
   3. Modifier un graphe ;
   4. Proposer une coloration pour un graphe ;
   5. Lire et écrire une coloration ;
   6. Evaluer une coloration.
3. Développer au moins deux algorithmes
   1. Greedy search ;
      1. Welsh-Powell ;
      2. D-SATUR ;
      3. Autres ;
   2. Hill-climbing ;
   3. Tabu search ;
   4. Evolutionary algorithms ;
   5. Hybrid algorithms ;
   6. …
4. Tester votre approche sur au moins trois ou quatre graphes différents dont certains possèdent plus de **cent** sommets ;
5. Ecrire un rapport sur le projet :
   1. Les choix discutés et retenus pour le projet ;
   2. Présentation des bibliothèques développées ;
   3. Une présentation des algorithmes développés (analyseur syntaxique, méthodes de coloration choisies) ;
   4. Présenter vos résultats

## Ressources

<https://sites.google.com/site/graphcoloring/vertex-coloring>

## Amélioration : Liste non exhaustive

* Implanter plus de deux approches ;
* Mettre au point une compétition dans la promotion (raconter la compétition dans le rapport)
  + Standardiser le mécanisme d’évaluation des algorithmes ;
  + Mesurer la qualité de la solution ;
  + Mesurer le temps de calcul ;
  + Centraliser les résultats.

### Barème

Le tableau suivant présente une grille d’évaluation. Les cases grisées indiquent des sommes partielles des coefficients de la partie. Ce barème a pour but de vous aider à **amorcer le travail** et identifiant des points clés du projet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Projet | Rendu conforme | | Makefile | 0,4 |
| Archive | 0,2 |
| Compilation | 0,4 |
| **1** | |
| Rapport | Document | Langue | 0,5 |
| Traitement de texte | 0,5 |
| **1** | |
| Introduction | Problème | 0,5 |
| Equipe | 0,5 |
| Méthode de travail | 0,5 |
| **1,5** | |
| Solution | Technologie | 0,5 |
| Présentation des différent algorithmes | 1,5 |
| Expériences (protocole) | 1,5 |
| Expériences (résultats) | 1,5 |
| Difficultés surmontées | 0,5 |
| **6,5** | |
| Conclusion | Discussion | 0,5 |
| Prolongation | 0,5 |
| **1** | |
| **10** | | |
| Développement | Développement  Général | Bibliothèque séparée | 0,5 |
| Gestion des variables | 0,5 |
| Outils de débugage | 0,5 |
| Ergonomie du code | 0,5 |
| **2** | |
| Application | Types (graphe, coloration, autres) | 1 |
| Entrées | 0,5 |
| Sauve | 0,5 |
| Constructeurs des types | 1 |
| Destructeurs des types | 1 |
| Opérateurs élémentaires | 1 |
| Algorithme 1 | 1,5 |
| Autre Algorithme 2 | 1,5 |
| **8** | |
| **10** | | |
| **20** | | | |

### Mise en garde

Une fois les outils de base implantés, le coût de développement d’un algorithme supplémentaire sera faible. Dans ce cas, vous pourrez être tenté d’améliorer vos approches en proposant de nombreux algorithmes. Cependant vous devez vous restreindre car vous n’aura pas le temps de faire des tests pertinents pour toutes vos algorithmes.

1. Attention ! Cette méthode fournit une « approximation » du nombre chromatique. C’est-à-dire qu’il peut exister des colorations meilleures (au sens qu’elles utilisent moins de couleurs) que celle trouvée par l’algorithme de Welsh et Powell. [↑](#footnote-ref-1)
2. Il serait judicieux que la lecteur et l’écriture soit des opérations symétriques l’une de l’autre. [↑](#footnote-ref-2)