Algorithmique Avancée – 2020

Denis Trystram

October 31, 2020

Consignes

Lire attentivement cette section avant de commencer votre épreuve.

- Essayez de vous placer dans des conditions au plus proches des conditions classiques d'un examen :

  Pas de distractions, concentrez vous au maximum sur l'épreuve ! Sauf circonstances exceptionnelles...

  L'épreuve est prévue pour une durée de 3 heures, respectez ce temps et ne passez pas plus des 3 heures, même si vous n'avez pas terminé...

- L'idéal serait d'envoyer vos copies (rédigées à la main/scannées ou avec un éditeur – word, latex, etc.) en format pdf.

  En cas de saisie via un éditeur de texte, indiquez le temps passé à la rédaction proprement dite.

- En cas de problème pour déposer les fichiers sur Teide, contactez moi par email sans faute en expliquant vos problèmes.

- J'ai basé ma relation avec chacun(e) d'entre vous sur la confiance, je suppose donc que vous travaillerez seul(e) et sans recours à Internet (on perd facilement du temps à trouver les "bonnes" sources...), par contre, vous pouvez consulter mes notes de cours.

1 Analyse du problème de Coloriage

1.1 Présentation du problème

On considère un graphe $G = (V, E)$ d'ordre n (nombre de sommets).
Un k-coloriage de G est une application qui associe à chaque sommet v_i de G une *couleur* (c'est-à-dire un indice de $\{1, ..., k\}$) tel que **tous les sommets adjacents sont assignés à une couleur différente**.

Ce problème peut également être vu comme une partition de l'ensemble des sommets V en k sous-ensembles disjoints V_1, ..., V_k correspondant aux sommets de même couleur.
Formellement, on s'intéresse à la résolution du problème de décision suivant :

k-COLOR
Instance. Un graphe $G = (V, E)$ (representé par exemple par sa matrice d'adjacence) et un entier k.
Question. Existe-t-il un k-coloriage de G ?

3-COLOR est la variante du problème où $k = 3$.
L'objectif de la section suivante est de montrer que ce problème est NP-complet.

1.2 Complexité de 3-COLOR

Question 1. (1 pt)
Montrer que 3-COLOR est dans $\mathcal{NP}$.
On détaillera l'algorithme en pseudo-code.

La réduction se fait à partir de la variante *NAE-3SAT* de 3SAT (Not All Equal-3SAT) définie comme les formules logiques qui possèdent une instantiation où toutes les clauses ont au moins un littéral VRAI et au moins un littéral FAUX.

Question 2. (Bonus : +2 pts) Cette question est facultative, on pourra admettre le résultat.
Montrer que le problème NAE-3SAT est NP-complet.

On construit maintenant une réduction de k-COLOR à partir de NAE-3SAT. On considère une instance (positive) ϕ de NAE-3SAT à n variables x_i et m clauses et on construit un graphe G_ϕ d'ordre $2n+3m+1$ de la façon suivante (cette réduction est proche de celle vue en cours pour VC).

- On relie les sommets x_i à $\bar{x}_i$ $\forall i$, $1 \leq i \leq n$.
- On relie toutes les variables x_i à un sommet y.

- Pour chaque clause $C_j = l_{j,1} \vee l_{j,2} \vee l_{j,3}$, on ajoute au graphe un *triangle* (graphe complet de 3 sommets A_j, B_j et C_j) et on les relie respectivement aux trois littéraux l_j d'une même clause[1].

Question 3. (1 pt) Construire le graphe correspondant à la formule suivante (4 variables booléennes et 3 clauses) :

$\phi = (x_1 \vee x_2 \vee \bar{x_3}) \wedge (x_3 \vee \bar{x_1} \vee \bar{x_4}) \wedge (\bar{x_2} \vee x_4 \vee \bar{x_3})$.

Question 4. (2 pts) Montrer que si $\phi \in$ NAE-3SAT, on obtient un 3-coloriage du graphe G_ϕ à partir d'une instantiation valide de ϕ de la façon suivante :

- y est colorié en vert
- x_i (resp. $\bar{x_i}$) est colorié en rouge si la clause est vraie, en bleu sinon
- finalement, on complète chaque triangle de clause avec les autres couleurs (parmi vert, bleu et rouge)

On suppose maintenant que G_ϕ est 3-colorié et que y est en vert.

Question 5. (2 pts) Montrer que si G_ϕ est un 3-coloriage, alors ϕ est positive au sens de NAE-3SAT[2].
En déduire que 3-COLOR est $\mathcal{NP}$-complet.

1.3 Indice chromatique

Le nombre (ou indice) chromatique d'un graphe G, noté $\mathcal{X}(G)$, est le **plus petit entier k pour lequel G admet un k-coloriage**.

Question 6. (1 pt) Déterminer l'indice chromatique du graphe de la figure 1. On justifiera la réponse.

1.4 Résolution exacte

Δ désigne le degré maximal du graphe G.
L'objectif ici est d'établir une borne supérieure de l'indice chromatique.

Question 7. (2 pts) Prouver que les graphes de degré maximum Δ sont $(\Delta + 1)$-coloriables en utilisant un argument de borne inférieure et en construisant un algorithme de coloriage polynomial.

On cherche maintenant à résoudre ce problème de façon exacte.

[1]On rappelle ici que les littéraux sont des variables ou leur complémentaire
[2]Montrer que chaque triangle de clause est relié à un littéral rouge et un littéral bleu

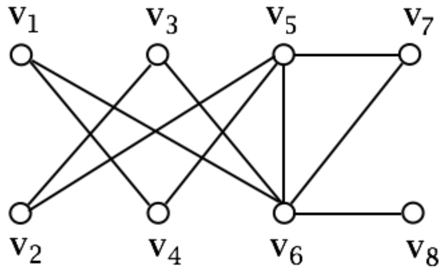

Figure 1: Exemple

Question 8. (2 pts) Proposer une politique de séparation en considérant pour chaque sommet une assignation de couleur pour ce sommet.
On pourra expliciter la politique en s'appuyant sur l'exemple de la figure précédente.

Question 9. (1 pt) Quelle évaluation de borne des sous-arbres suggérez-vous pour couper des branches ?

1.5 Approximation de 3-COLOR

On considère l'algorithme glouton (*First-Fit*) suivant :

- colorier les sommets dans l'ordre $v_1, v_2, \ldots, v_n$
- on utilise la plus petite couleur possible pour colorier v_i (en d'autres termes, v_i a la plus petite couleur non encore utilisée par les voisins de v_i déjà coloriés)

Question 10. (2 pts) Construisez une instance où First-Fit utilise un nombre arbitrairement grand de couleurs par rapport à l'indice chromatique.

Question 11. (1 pt) Que se passe-t-il si les sommets de G étaient triés par degrés décroissants ?
On justifiera la réponse.

Question 12. (2 pts)
Construire un algorithme polynomial pour le 2-coloriage de graphes bipartis (c'est-à-dire les graphes ne possédant pas de cycles impairs).

Autre exercice (indépendent) page suivante...

2 Une variante de 2Partition (3 pts)

Prouver que le problème suivant est NP-complet.

PP (Partition parfaite)
Instance: On considère $2n$ entiers s_i.
Question: Existe-t-il une *partition parfaite* en deux ensembles de même cardinaux A_1 et A_2, tels que $\Sigma_{i \in A_1} s_i = \Sigma_{i \in A_2} s_i$ où $|A_1| = |A_1| = n$?

Décrire une solution en programmation dynamique pour résoudre ce problème.