

Optimisation discrète, séance 5 : exercices corrigés

THÉORIE de la COMPLEXITÉ

Question 1

On note k -COL le problème consistant à déterminer si les noeuds d'un graphe (G, A) peuvent être coloriés avec k couleurs sans que deux noeuds adjacents aient la même couleur (rappel : on dit que l'indice chromatique est k).

• Comment peut-on représenter un graphe à n noeuds ? Quelle est la taille des données dans ce codage.

Corr. On le représente par la matrice d'adjacence \mathbf{M} qui est booléenne, ce qui fait au plus n^2 bits pour un graphe à n noeuds. Le coloriage est défini par la donnée d'un vecteur \mathbf{V} définissant la couleur de chaque noeud (un nombre de 1 à k), la couleur étant représentée en unaire (c'est à dire avec des bâtons), ce qui fait kn bits. Soit $\phi(\mathbf{M})$ la fonction qui vaut 1 ou 0 selon que le graphe est ou non coloriable avec k couleurs.

• Montrer que le problème du k coloriage d'un graphe (en abbréviation k -COL) est un problème NP.
Corr. La donnée d'un coloriage nous sert ici de *témoin*. Le coloriage est défini par la donnée d'un vecteur \mathbf{V} définissant la couleur de chaque noeud (un nombre de 1 à k), la couleur étant représentée en unaire (c'est à dire avec des bâtons), ce qui fait kn bits. La longueur des données $|\mathbf{M}| + |\mathbf{V}| = n^2 + kn$ d'un graphe colorié est inférieure (pour n grand) à $2|\mathbf{M}|$. Soit $\psi(\mathbf{M}, \mathbf{V})$ la fonction qui vaut 1 ou 0 selon que le graphe est coloriable ou pas. Pour vérifier si le graphe est coloriable la fonction ψ doit balayer les (au plus n^2) arêtes et pour chaque arête vérifier si les couleurs des extrémités sont différentes. Toutes ces opérations sont polynômiales par rapport à la longueur $|\mathbf{M}| + |\mathbf{V}|$ des données.

• Montrer directement, sans faire appel au théorème de Cook, que le problème k -COL est polynômialement représentable dans SAT, i.e. étant donné un graphe (G, A) on peut définir un ensemble de variables propositionnelles et un ensemble de clauses composées de ces variables tels que (G, A) est k coloriable si et seulement si les clauses sont satisfiables (et avec un codage des clauses ne prenant pas beaucoup plus de mémoires que celui du graphe).

Corr. On définit une variable booléenne P_i^l pour chaque noeud i et chaque couleur l . On introduit les clauses

$$C_i : P_i^1 \vee \dots \vee P_i^k$$

pour chaque noeud, et

$$A_{i,j,l} : \neg P_i^l \vee \neg P_j^l$$

pour chaque arête $(i, j) \in A$ et chaque couleur l .

Supposons cet ensemble de clause satisfiable, on attribue à un noeud i la couleur l du plus petit indice l tel que P_i^l est vraie (il existe puisque C_i est vraie). Deux noeuds adjacents n'ont pas la même couleur

puisque $A_{i,j,l}$ est vraie.

Réciproquement si le graphe k coloriable et si on donne la valeur vraie à P_i^l si le noeud i a la couleur l on rend vraies toutes les clauses. En outre le codage des clauses exige de l'ordre de kn^2 bits. Il reste donc d'un encombrement polynômial par rapport à celui du graphe.

Question 2

Soit un ensemble de n clauses à p éléments pris dans un ensemble de m variables propositionnelles P_j

$$L_i^1 \vee L_i^2 \vee \dots \vee L_i^p, \quad i = 1, \dots, n$$

où $L_i^k = P_j$ ou $L_i^k = \neg P_j$. On note p -SAT le problème consistant à décider si ces clauses sont satisfiables ou non, i.e. si il existe un ensemble de valeurs de vérité $P_j = V$ ou $P_j = F$ qui rende toutes les clauses vraies.

- Comment peut-on représenter les données de ce problème ? Quelle est la taille des données dans ce codage.

Corr. On peut coder les clauses à l'aide d'un tableau de dimension (n, p) de nombres compris entre 1 et $2m$ ($2m$ et non pas m pour noter la négation), soit au plus $2mnp$ bits.

- Montrer que 2-SAT est décidable avec un algorithme polynômial.

Corr. Si l'ensemble initial de clauses est satisfiable, toutes les clauses introduites en sont des conséquences, elles sont donc satisfiables, on ne peut donc avoir P_k et $\neg P_k$ dans la liste E_j .

Réciproquement, si on atteint $j = m$, démontrons par récurrence que E_j est satisfiable. Si on atteint $j = m$ l'ensemble E_m est vide ou contient une seule clause P_m ou $\neg P_m$, qui est satisfiable. Supposons qu'il existe des valeurs des variables P_k , $k \geq j + 1$ telles que la liste E_{j+1} soit satisfiable, montrons que l'on peut déterminer une valeur de P_j telle que E_j est satisfiable. Les clauses de E_j qui sont de la forme $P_k \vee P_l$ pour $k, l > j$ sont dans E_{j+1} et sont satisfaites. Il reste donc à montrer que l'on peut déterminer une valeur de P_j telle que toutes les clauses de la forme $P_j \vee P_k$ ou $\neg P_j \vee P_k$ soient satisfaites. Il suffit d'examiner les clause telles que P_k soit faux. Or on ne peut avoir simultanément une clause $P_j \vee P_k$ et une clause $\neg P_j \vee P_l$ avec P_k et P_l faux car la clause $P_j \vee P_l$ est par construction dans E_{j+1} et elle est donc satisfaite. Donc, ou bien toutes ces clauses sont de la forme $P_j \vee P_k$ et il suffit de prendre pour P_j la valeur vrai, ou bien elles sont toutes de la forme $\neg P_j \vee P_k$ et il suffit d'attribuer à P_j la valeur faux.

Le nombre de clauses dans E_j étant au plus $4m^2$ toutes les opérations ci-dessus sont polynômiales vis à vis de la longueur des données.

- Montrer qu'une clause quelconque $L^1 \vee L^2 \vee \dots \vee L^p$, est équivalente à un ensemble de $p - 3$ clauses à 3 éléments (Indic. : introduire, pour $i = 3, p - 1$ une variable propositionnelle Q_i représentant $L^i \vee L^{i+1} \vee \dots \vee L^p$). En déduire que p-SAT est polynômialement équivalent à 3-SAT et que donc 3-SAT est NP-Complet.

Corr. La clause $L^1 \vee L^2 \vee \dots \vee L^p$ est satisfiable si et seulement si l'ensemble des $p - 3$ propositions

$$\begin{aligned} &L^1 \vee L^2 \vee Q_3 \\ Q_3 &\Rightarrow (L^3 \vee Q_4) \\ Q_4 &\Rightarrow (L^4 \vee Q_5) \\ &\dots \end{aligned}$$

$$Q^{p-1} \Rightarrow (L^{p-1} \vee L^p)$$

le sont, c'est à dire si l'ensemble des clauses

$$\begin{aligned} &L^1 \vee L^2 \vee Q_3 \\ &\neg Q_3 \vee L^3 \vee Q_4 \\ &\neg Q_4 \vee L^4 \vee Q_5 \\ &\dots \\ &\neg Q^{p-1} \vee L^{p-1} \vee L^p \end{aligned}$$

est satisfiable.

Pour l'équivalence : d'une part 3-SAT est un cas particulier de SAT, d'autre part SAT est représentable par un problème 3-SAT avec des données qui peuvent être représentées par un tableau de $(3, n(p-2))$ nombres compris entre 1 et $2m$. Toutes ces données ont un encombrement qui n'excède pas $6mnp$.

Question 3

- Décrire un algorithme polynômial pour 2-COL.

Corr. On vérifie facilement qu'un graphe est 2-coloriable si ou seulement si tous les cycles sont d'ordres pairs ou encore si les chemins qui relient deux noeuds sont tous pairs ou tous impairs. Pour colorier le graphe on peut, par exemple, construire un arbre engendrant le graphe ¹, puis colorier un noeud de l'arbre en rouge et tous les autres noeuds en rouge ou vert selon que leur distance au premier est paire ou impaire, et enfin vérifier que deux noeuds adjacents n'ont pas la même couleur. Toutes ces opérations sont polynômiales par rapport au nombre n^2 de bits décrivant le graphe.

Soit un ensemble de n clauses à 3 éléments pris dans un ensemble de m variables propositionnelles P_j dont on veut vérifier la satisfiabilité. On associe un graphe à ce problème de la manière suivante (les noeuds portent le même nom que les objets qu'ils représentent) :

- On définit un noeud pour chaque variable P_i et $\neg P_i$ et une arête entre les deux.
- On ajoute un noeud O relié à chacun des noeuds P_i et $\neg P_i$.
- On définit trois noeuds $L_i^k, k = 1, 2, 3$ pour chaque clause $L_i^1 \vee L_i^2 \vee L_i^3$ et trois arêtes entre eux.
- On relie par une arête L_i^k à $\neg P_j$ (resp. P_j) si L_i^k est P_j (resp. $\neg P_j$).

On "colorie" ce graphe avec les "couleurs" bleu, rouge, vert qui correspondront (mais pas nécessairement dans cet ordre) à : vrai, faux, indifférent.

- Montrer à l'aide de cette représentation que 3-SATsym est polynômialement équivalent à 3-COL. En déduire que 3-COL (ainsi que k-COL) est NP-complet.

Corr. Si les clauses vérifient la propriété Π , on associe au noeud O la couleur verte, au noeud P_j (resp. $\neg P_j$) la couleur bleue (resp. rouge) et au noeud L_i^k la couleur bleue ou rouge selon la valeur de vérité de la variable qu'il représente (choisie dans un ensemble de valeurs de ces variables qui satisfont les clauses). Chaque "triangle" L_i^1, L_i^2, L_i^3 a alors au moins un sommet en bleu et un en rouge,

¹voir séance 1, vérifier que cet algorithme est polynômial. Si le graphe n'est pas connexe, l'arbre ne sera pas couvrant, on recommence avec un noeud qui n'est pas sur l'arbre

mais deux sommets ont la même couleur : on change la couleur de l'un des deux en vert. On obtient ainsi un 3-coloriage du graphe.

Dans l'autre sens, si on a un 3-coloriage du graphe, on peut, sans réduire la généralité, supposer que le noeud O est vert. On en déduit que les noeuds P_j et $\neg P_j$ sont (exclusivement) ou bleus ou rouges puisque $O, P_j, \neg P_j$ est un triangle. Supposons que L_i^k est relié à $\neg P_j$ par construction ; L_i^k a alors la couleur de P_j ou est vert. De même si L_i^k est relié à P_j par construction, L_i^k a alors la couleur de $\neg P_j$ ou est vert. Si on attribue aux noeuds P_j ou $\neg P_j$ la valeur Vraie s'ils sont bleus et Faux s'ils sont rouges on obtient un ensemble de valeurs de vérités compatibles avec l'interprétation de L_i^k comme étant P_j ou $\neg P_j$ (s'il est relié par un arc à l'un de ces noeuds) et qui donne aux clauses la propriété Π , puisque chaque triangle a un noeud bleu et un noeud rouge.