Fiche 06 :
Problème SAT, NP-complet
une correction

Licence 3 informatique
2022 / 2023

Exercice 1 : Problèmes

1.a. De très nombreux problèmes de décisions courants sont dans la classe de complexité P comme décider si un tableau est correctement trié, décider du plus court chemin dans un graphe, multiplier deux matrices, etc. etc.

1.b. Le plus connu est le premier problème de décision de la classe NP (démontré par Cook 1971) est le problème de décision SAT.

Exercice 2 : Coloration

Question 2.b. Pour chaque nœud, v_i avec $i \in \{1, \dots, n\}$ et chaque couleur $k \in \{1, \dots, K\}$ avec $K = 3$ dans cet exemple, définissons la variable booléenne $x_{i,k}$ telle que :

$$x_{i,k} = \begin{cases} 1 & \text{si } v_i \text{ est de couleur k} \\ 0 & \text{sinon} \end{cases}$$

Remarque : ce principe de codage d'un nombre entier par des variables booléennes s'appelle parfois en apprentissage automatique du "one hot encoding". Schématiquement, une variable avec K valeurs possibles peut de coder en un vecteur booléen de taille K : $v_i = k$ si et seulement si $[0, \dots, 0, 1(k^{\text{ième}}\text{coord.}), 0, \dots 0]$.

Ecrivons une expression booléenne pour exprimer la contrainte qu'un nœud ne possède qu'une unique couleur, c'est-à-dire que seulement une seule des variables associées à un nœud est vraie.

De manière non formelle, il faut qu'une variable au moins soit vraie, et que si un nœud est d'une couleur alors, il n'est pas d'une autre couleur.

De manière formelle, pour exprimer qu'une variable au moins est vraie :
pour tout $i \in \{1, \dots, n\}$, $C_i \equiv x_{i,1} \vee \dots \vee x_{i,K}$.

Et pour exprimer une seule couleur par nœud :
pour tout $i \in \{1, \dots, n\}$ et tout $k \in \{1, \dots, K\}$,
$C_{i,k} \equiv x_{i,k} \Rightarrow (\rceil x_{i,1} \wedge \dots \wedge \rceil x_{i,k-1} \wedge \rceil x_{i,k+1} \dots \wedge \rceil x_{i,K})$.
A noter que $C_{i,k}$ est équivalent à $\rceil x_{i,k} \vee (\rceil x_{i,1} \wedge \dots \wedge \rceil x_{i,k-1} \wedge \rceil x_{i,k+1} \dots \wedge \rceil x_{i,K})$

Ainsi, les contraintes sur la couleur des nœuds s'exprime par la conjonction des clauses :

$$\bigwedge_{i=1}^{n} C_i \wedge \bigwedge_{i=1}^{n} \bigwedge_{k=1}^{K} C_{i,k}$$

Nous pouvons écrire des clauses similaires pour exprimer que deux nœuds voisins n'ont pas la même couleur.

Pour chaque arête $e_j = (v_{j_1}, v_{j_2})$ avec $j \in \{1, \ldots m\}$, et chaque couleur $k \in \{1, \ldots, K\}$, $D_{j_1,j_2,k} \;\; \equiv x_{j_1,k} \Rightarrow \rceil x_{j_2,k}$. A noter que $D_{j_1,j_2,k} \;\; \equiv \rceil x_{j_1,k} \vee \rceil x_{j_2,k}$.

Ainsi, les contraintes sur la couleur des nœuds voisins s'exprime par la conjonction des clauses :

$$\bigwedge_{(j_1,j_2) \in E} \bigwedge_{k=1}^{K} D_{j_1,j_2,k}$$

Le problème de coloration se réduit donc à la décision d'une expression booléenne et donc à un problème SAT. Par conséquent, si nous avons un algorithme pour résoudre SAT, nous pouvons utiliser cet algorithme pour résoudre un problème de K-coloration de graphe.

Exercice 3 : Sudoku

Questions 3.a. Les 9 nombres du problème Sudoku peuvent être coder par une couleur. Chaque case du problème peut être encoder par un nœud du graphe. Dans un Sudoku, chaque nombre n'apparait q'une seule fois pour chaque "carré", ligne et colonne ce qui se traduit par des arêtes du graphe. Pour chaque couple de nœuds d'un carré, il faut définir une arête. De même pour les nœuds de chaque ligne, on définit une arrête et de même pour les colonnes.

Ainsi, le problème du Sudoku se réduit en un problème de coloration de graphe.