TD de Complexité

Exercice 1. *En chemin !*

Le problème du voyageur de commerce est un problème d'optimisation qui, étant donné une liste de villes, et des distances entre toutes les paires de villes, détermine un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ.

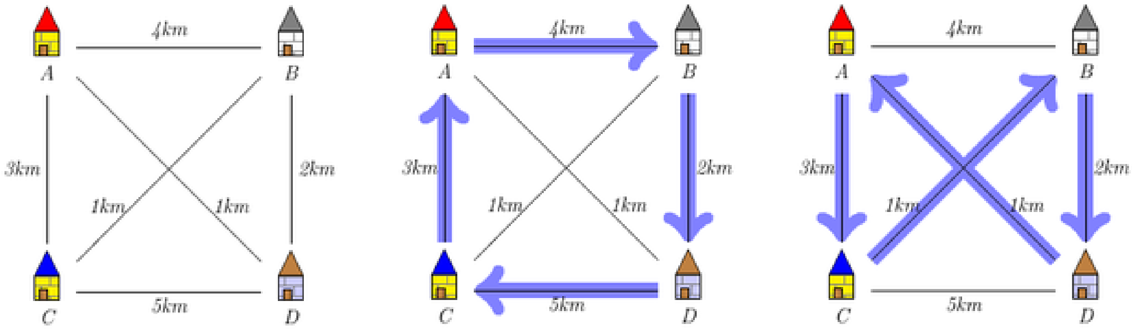


FIGURE 1 – Une instance du problème avec deux chemins : le premier de distance totale 14km, et le deuxième de 7km.

On s'intéresse à **TSP**, le problème de décision associé : étant donné un graphe complet pondéré (G, w) et un entier k, existe-t-il un cycle de poids au plus k passant par toutes les villes une et une seule fois ?

1. Combien existe-t-il de cycles différents passant par toutes les villes une et une seule fois ?
2. Montrer que $TSP \in \mathsf{NP}$.
3. Á l'aide d'une réduction depuis le problème HAMILTONIEN, montrer que TSP est NP-*dur*, c'est-à-dire que tout les problèmes de NP se réduisent à lui.
   Indication : on pourra poser $w((u, v)) = 0$ si $(u, v) \in G$, et 1 sinon.

Exercice 2. *Le retour de Horn*

On s'intéresse à la réduction polynomial de 2-SAT vers la recherche de chemin dans un graphe orienté. On rappelle que 2-SAT est la satisfiabilité d'une Forme Normale Conjonctive F comportant au maximum 2 littéraux par clause. On considère la transformation suivante d'une instance F de 2-SAT en un graphe *orienté* G appelé le *graphe d'implication* de F.

— Pour chaque variable propositionnelle x_i, G possède deux sommets étiquetés x_i et $\overline{x}_i$.
— Pour chaque clause $l_i \vee l_j$, on créé une arête du sommet $\bar{l}_i$ vers l_j ("si l_i est faux alors l_j doit être vrai") et une arête du sommet $\bar{l}_j$ vers l_i ("si l_j est faux alors l_i doit être vrai").

1. Dessinez le graphe d'implication correspondant à la formule suivante :
   $F = (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee x_3) \wedge (\overline{x_3} \vee x_1)$.

2. Donnez l'ordre de grandeur de la complexité de la construction du graphe d'implications en fonction du nombre n de variables et du nombre m de clauses de l'instance 2-SAT à transformer.

3. Soit F une instance de 2-SAT et G le graphe d'implications correspondant. Montrez que, s'il existe dans G un circuit passant par deux sommets x_i et $\overline{x_i}$, alors F est insatisfiable.
   Indication : Montrez (par double implication) que les littéraux reliés par une chaîne fermée d'implications (un circuit du graphe G) ne peuvent qu'avoir la même valeur de vérité (Vrai ou Faux), dans une interprétation satisfiant F.

4. Donner un algorithme qui permet de décider deux sommets donnés font partie d'un même circuit.

5. En déduire que 2-SAT appartient à P.

Exercice 3. *Cahier de coloriage*

On s'intéresse au problème K-COULEUR suivant : étant donné un graphe $G = (V, E)$, existe-t-il une fonction $c : V \to \{1, 2, \ldots, k\}$ telle que pour toute arête $(u, v) \in E$, on a $c(u) \neq c(v)$.

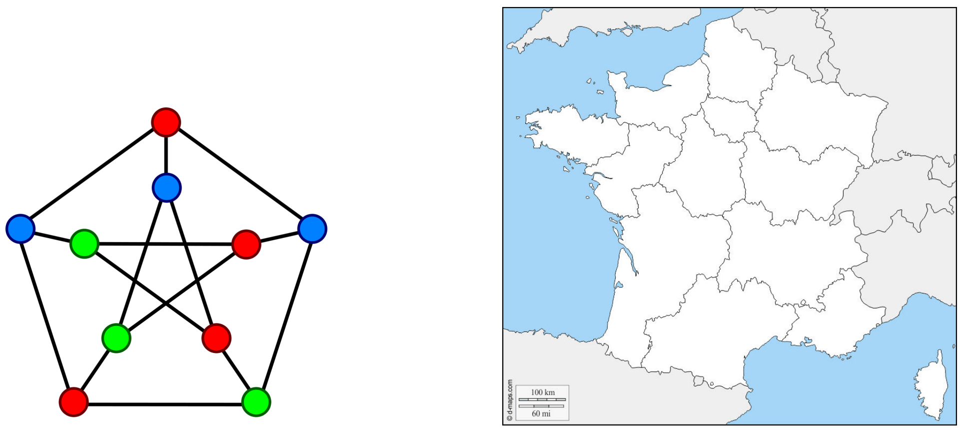
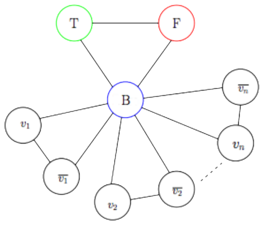
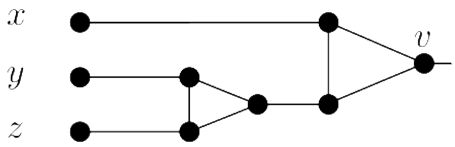
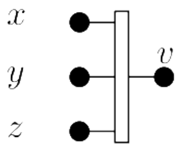


FIGURE 2 – Une instance résolue de 3-coloriage à gauche et une instance à résoudre à droite.

1. Trouver un 4-coloriage de la carte des régions françaises.
2. Montrer que 2-COULEUR $\in$ P.
3. Montrer qu'un graphe G est 2-coloriable si et seulement s'il n'existe pas de cycle de longueur impaire dans G.
4. On suppose maintenant que le problème 2-SAT suivant est dans P : étant donnée une formule F en forme normale conjonctive dans laquelle chaque clause a exactement 2 littéraux, F est-elle satisfiable ?
   Re-montrer, à l'aide d'une réduction à 2-SAT, que 2-COULEUR est dans P.
5. Montrer que 3-Couleur $\in$ NP.
6. Montrer que 3-Couleur est NP-dur.
7. Montrer que 4-Couleur est NP-Complet.







Exercice 4. *De la décision à la résolution.*

✎ Montrer que si P = NP, il existe un algorithme polynomial qui prend en entrée une formule φ sous forme normale conjonctive et retourne une affectation valide de φ s'il en existe une et 0 sinon.

Exercice 5. *Éspérons que ça marche !*

Dans cet exercice, nous nous intéressons à des classes de problèmes résolus par des algorithmes probabilistes :

— ZPP (zero-error probabilistic polynomial time) : un problème de décision P appartient à ZPP si et seulement s'il existe un algorithme probabiliste A qui résout P et dont l'espérance du temps de calcul est polynomiale.

— RP (randomized polynomial time) : un problème de décision P appartient à RPsi et seulement s'il existe un algorithme probabiliste polynomial A tel que, pour toute instance I du problème : si $I \notin P$, alors $A(I) = \text{NON}$; si $I \in P$, alors $\mathbb{P}(A(I) = \text{OUI}) \geq 1/2$.

— coRP : $L \in \text{coRP} \Leftrightarrow \overline{L} \in \text{RP}$

1. Montrer qu'on ne change pas la classe RP si, dans sa définition, on remplace la probabilité de succès de $1/2$ par un nombre $\alpha \geq 1/2$.

2. En déduire que $(\text{RP} \cap \text{coRP}) \subseteq \text{ZPP}$.

3. Montrer que $\text{ZPP} \subseteq (\text{RP} \cap \text{coRP})$.
(Indication : on pourra utiliser l'inégalité de Markov : si $X \geq 0$, alors $\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}(\mathbb{X})}{a}$)