TD 4 : Problèmes d'optimisation

Théorie de la complexité S5.

2022

Exercice 1 — *PO et NPO*

Pour chacun des problèmes suivants, écrivez le sous forme d'un problème d'optimisation et déterminez s'il appartient à la classe PO ou NPO.

1. (CHROMA)

▶ **Correction**

Soit un graphe $G = (V, E)$, trouver une coloration des nœuds de sorte que deux nœuds voisins n'aient pas la même couleur et minimisant le nombre de couleurs.

— Instances $\mathcal{I} = \{\text{Graphe (simple) } G = (V, E)\}$

— Solution $\mathcal{S}(G) = \{\text{coloration } C \text{ de } V\} = \{c_v \in \mathbb{N} | c_v \leq |V| \text{ et } v \in V, \text{ tel que pour tout}(u, v) \in E, c_u \neq c_v\}$. Chaque couleur est représentée par un entier unique inférieur à $|V|$.

— Mesure $\mathcal{M}(G, C) =$ nombre d'éléments distincts dans C

— Objectif = min

(CHROMA) est dans NPO ?

— On représente les graphes avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial.

— On représente la coloration avec un nombre binaire de taille au plus $(|V|+1)^2$, contenant $|V|$ nombre unaires inférieurs à $|V|$ séparés par des 0, le i^e nombre correspond à la couleur du i^e nœud. On a bien $|V| \leq |G|$, donc la taille des solutions est bornée polynomialement par rapport à celles des instances.

— Si on a un nombre binaire de taille au plus $(|V| + 1)^2$, on peut vérifier qu'il est bien au format $|V|$ nombres unaires inférieurs à $|V|$ séparés par des 0 et on peut vérifier, pour chaque arête (u, v) que les entiers associés à u et v sont différents.

— Connaissant G et C, on peut calculer $\mathcal{M}(G, C)$ en $O(|V|^3)$ (on parcours les couples de nœuds, et on supprime de C les couleurs identiques pour calculer le nombre de couleur distinctes.) *Il existe sans aucun doute un algorithme plus efficace que ça*

Donc (CHROMA) est dans NPO. On peut noter que si on n'avait pas borné les couleurs avec $|V|$ alors on aurait un nombre infini de solutions réalisables et donc (CHROMA) ne serait pas dans NPO.

Il est peut probable que (CHROMA) soit dans PO puisque $(\text{CHROMA})_D$ est NP-Complet.

2. (SIZE)

▶ **Correction**

C'est un piège, (SIZE) n'est pas un problème d'optimisation, c'est un problème de construction (on veut calculer la taille d'un graphe).

3. (INS)

▶ **Correction**

Soit un graphe $G = (V, E)$, trouver un stable de taille maximum de G (un stable est un graphe sans arête).

— Instances $\mathcal{I} = \{\text{Graphe (simple) } G = (V, E)\}$

— Solution $\mathcal{S}(G) = \{\text{stable } S \text{ de } G\} = \{S \subset V \text{ tel que pour tout } (u, v) \in S, (u, v) \notin E\}$.

— Mesure $\mathcal{M}(G, S) = |S|$

— Objectif $=$ max

(INS) est dans NPO ?

— On représente les graphes avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial.

— On représente les stables avec un nombre binaires de taille $|V|$, le i^e bit est un 1 si le i^e nœud est placée dans S et 0 sinon. On a bien $|V| \leq |G|$, donc la taille des solutions est bornée polynomialement par rapport à celles des instances.

— Si on a un nombre binaire de taille $|V|$ (i.e. un sous ensemble $S \subset V$), on peut vérifier que c'est un stable en $O(|V|^2)$, on vérifie tous les couples de nœuds un par un.

— Connaissant G et S, on peut calculer $\mathcal{M}(G, S)$ en $O(|V|)$ (on compte les 1 dans le nombre binaire)

Donc (INS) est dans NPO.
Il est peut probable que (INS) soit dans PO puisque $(\text{INS})_D$ est NP-Complet.

4. (WINS)

▶ **Correction**

Soit un graphe $G = (V, E)$ et des poids $\omega : V \to \mathbb{N}$, trouver un stable de poids maximum de G.

— Instances $\mathcal{I} = \{\text{Graphe (simple) } G = (V, E), \text{ poids } \omega : V \to \mathbb{N}\}$

— Solution $\mathcal{S}(G, \omega) = \{\text{stable } S \text{ de } G\} = \{S \subset V \text{ tel que pour tout } (u, v) \in S, (u, v) \notin E\}$.

— Mesure $\mathcal{M}(G, \omega, S) = \sum\limits_{v \in S} \omega(v)$

— Objectif $=$ max

(WINS) est dans NPO ?

— On représente les graphes avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial. On représente les poids par une liste de $|V|$ entiers écrits en binaire qu'on peut aussi reconnaître en temps polynomial.

— On représente les stables avec un nombre binaires de taille $|V|$, le i^e bit est un 1 si le i^e nœud est placée dans S et 0 sinon. On a bien $|V| \leq |G|$, donc la taille des solutions est bornée polynomialement par rapport à celles des instances.

— Si on a un nombre binaire de taille $|V|$ (i.e. un sous ensemble $S \subset V$), on peut vérifier que c'est un stable en $O(|V|^2)$, on vérifie tous les couples de nœuds un par un.

— Connaissant G et S, on peut calculer $\mathcal{M}(G, S)$ en $O(|V| \cdot \max(\log(\omega(v))))$.

Donc (WINS) est dans NPO.
Il est peut probable que (WINS) soit dans PO puisque $(\text{WINS})_D$ est NP-Complet.

5. (MATCHING)

▶ **Correction**

Soit un graphe $G = (V, E)$, trouver un couplage maximum de G. (Couplage : un ensemble d'arêtes qui n'ont pas d'extrémité en commun)

— Instances $\mathcal{I} = \{$Graphe (simple) $G = (V, E)\}$

— Solution $\mathcal{S}(G) = \{$couplage de $G\} = \{F \subset E$, tel que pour tout $f_1 f_2 \in F$, f_1 et f_2 n'ont pas d'extrémité en commun$\}$

— Mesure $\mathcal{M}(G, F) = |F|$

— Objectif = max

(MATCHING) est dans NPO ?

— On représente les graphes avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial.

— On représente les couplages avec un nombre binaires de taille $|E|$, le i^{e} bit est un 1 si la i^{e} arête est placée dans F et 0 sinon. On a bien $|E| \leq |G|$, donc la taille des solutions est bornée polynomialement par rapport à celles des instances.

— Si on a un nombre binaire de taille $|E|$ (i.e. un sous ensemble $F \subset E$), on peut vérifier que c'est un couplage en $O(|E|^2)$, on vérifie tous les couples d'arêtes un par un.

— Connaissant G et F, on peut calculer $\mathcal{M}(G, F)$ en $O(|E|)$ (on compte les 1 dans le nombre binaire)

Donc (MATCHING) est dans NPO.
Et il existe un algorithme polynomial pour résoudre le problème de construction $(\text{MACTHING})_C$ avec l'algorithme d'Edmonds `https://fr.wikipedia.org/wiki/Algorithme_d\%27Edmonds_pour_les_couplages` donc (MATCHING) est dans PO.

6. (VERTEX COVER)

▶ **Correction**

Soit un graphe $G = (V, E)$, trouver un sous-ensemble de nœuds C de taille minimum et couvrant toutes les arêtes (chaque arête de E possède une extrémité dans C).

— Instances $\mathcal{I} = \{$Graphe (simple) $G = (V, E)\}$

— Solution $\mathcal{S}(G) = \{$couverture C des arêtes de $G\} = \{C \subset V$ tel que pour tout $(u, v) \in E, u \in C$ ou $v \in C\}$.

— Mesure $\mathcal{M}(G, C) = |C|$

— Objectif = min

(VERTEX COVER) est dans NPO ?

— On représente les graphes avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial.

— On représente les couvertures avec un nombre binaires de taille $|V|$, le i^{e} bit est un 1 si le i^{e} nœud est placée dans C et 0 sinon. On a bien $|V| \leq |G|$, donc la taille des solutions est bornée polynomialement par rapport à celles des instances.

— Si on a un nombre binaire de taille $|V|$ (i.e. un sous ensemble $C \subset V$), on peut vérifier que c'est une couverture des arêtes en $O(|V||E|)$, on vérifie pour chaque arête si une de ses extrémités est dans C.

— Connaissant G et C, on peut calculer $\mathcal{M}(G, C)$ en $O(|V|)$ (on compte les 1 dans le nombre binaire)

Donc (VERTEX COVER) est dans NPO.
Il est peut probable que (VERTEX COVER) soit dans PO puisque $(\text{VERTEX COVER})_D$ est NP-Complet.

7. (SHP)

▶ **Correction**

Soit un graphe orienté $G = (V, A)$, des poids $\omega : A \to \mathbb{N}$ et des nœuds s et t de V, trouver un chemin de poids minimum de s à t.

— Instances $\mathcal{I} = \{$Graphe (simple) $G = (V, A)$, poids $\omega : A \to \mathbb{N}$, nœuds s et $t\}$

— Solution $\mathcal{S}(G, \omega, s, t) = \{$chemin élémentaire P de s à $t\}$.

— Mesure $\mathcal{M}(G, \omega, s, t, P) = \sum\limits_{a \in P} \omega(a)$

— Objectif = min

(SHP) est dans NPO ?

— On représente les graphes orientés avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial. On représente les poids par une liste de $|A|$ entiers écrits en binaire qu'on peut aussi reconnaître en temps polynomial. On représente s et t par deux entiers écrits en unaire de taille inférieur à $|V|$, qu'on peut reconnaître en temps polynomial. La valeur de ces entiers est l'indice de s et de t dans la liste des nœuds.

— On représente les chemins avec une suite de nombre unaires inférieurs à $|E|$. Le $^{\text{e}}$ entier encode le $^{\text{e}}$ arc du chemin. La valeur de cet entier est égal à l'indice de l'arc dans la liste des arcs. Cette suite contient au plus $|V| - 1$ entiers.

— Si on a une telle suite, on peut vérifier qu'elle encode bien un chemin en suivant ce chemin pour montrer qu'il lie bien s à t. Pour chaque arc, on vérifie que le nœud courant (démarrant à s) est bien l'origine de l'arc, on remplace ensuite le nœud courant par la destination de l'arc. Cela peut se faire en $O(|V| \cdot |E|)$ *(remarque, cette complexité suppose un accès en O(1) aux origines et destinations de l'arc, ce qui n'est peut être par le cas dans notre encodage)*

— Connaissant G, ω, s, t et P, on peut calculer $\mathcal{M}(G, \omega, s, t, P)$ en $O(|V| \cdot \max \log(\omega(v)))$.

Donc (SHP) est dans NPO.

On peut résoudre $(\text{SHP})_C$ en temps polynomial avec l'algorithme de Dijkstra. Donc (SHP) est dans PO.

Remarque : si on autorise des chemins non élémentaires, (SHP) n'est plus dans NPO puisqu'il y a un nombre infini de chemins non élémentaires dans un graphe avec un circuit. Et si on autorise des poids rationnels plutôt que entier, alors on ne respecte pas la définition d'un problème d'optimisation du cours qui impose que les mesures soient des entiers. Il n'empêche que, dans ces deux cas, Dijkstra fonctionne toujours.

8. (LOP)

▶ **Correction**

Soit un graphe orienté $G = (V, A)$, des poids $\omega : A \to \mathbb{N}$ et des nœuds s et t de V, trouver un chemin de poids maximum de s à t.

On peut remarquer que la suite de la réponse est quasiment identique à celle de (SHP).

— Instances $\mathcal{I} = \{$Graphe (simple) $G = (V, A)$, poids $\omega : A \to \mathbb{N}$, nœuds s et $t\}$

— Solution $\mathcal{S}(G, \omega, s, t) = \{$chemin élémentaire P de s à $t\}$.

— Mesure $\mathcal{M}(G, \omega, s, t, P) = \sum\limits_{a \in P} \omega(a)$

— Objectif = max

(LOP) est dans NPO ?

— On représente les graphes orientés avec une matrice d'adjacence, qu'on peut reconnaître en temps polynomial. On représente les poids par une liste de $|A|$ entiers écrits en binaire qu'on peut aussi reconnaître en temps polynomial. On représente s et t par deux entiers écrits en unaire de taille inférieur à $|V|$, qu'on peut reconnaître en temps polynomial. La valeur de ces entiers est l'indice de s et de t dans la liste des nœuds.

— On représente les chemins avec une suite de nombre unaires inférieurs à $|E|$. Le i^{e} entier encode le i^{e} arc du chemin. La valeur de cet entier est égal à l'indice de l'arc dans la liste des arcs. Cette suite contient au plus $|V| - 1$ entiers.

— Si on a une telle suite, on peut vérifier qu'elle encode bien un chemin en suivant ce chemin pour montrer qu'il lie bien s à t. Pour chaque arc, on vérifie que le nœud courant (démarrant à s) est bien l'origine de l'arc, on remplace ensuite le nœud courant par la destination de l'arc. Cela peut se faire en $O(|V| \cdot |E|)$ *(remarque, cette complexité suppose un accès en O(1) aux origines et destinations de l'arc, ce qui n'est peut être par le cas dans notre encodage)*

— Connaissant G, ω, s, t et P, on peut calculer $\mathcal{M}(G, \omega, s, t, P)$ en $O(|V| \cdot \max \log(\omega(v)))$.

Donc (LOP) est dans NPO.

Il est peut probable que (LOP) soit dans PO puisque $(\text{LOP})_D$ est NP-Complet.

Remarque : si on autorise des chemins non élémentaires, (LOP) n'est plus dans NPO puisqu'il y a un nombre infini de chemins non élémentaires dans un graphe avec un circuit. Et si on autorise des poids rationnels plutôt que entier, alors on ne respecte pas la définition d'un problème d'optimisation du cours qui impose que les mesures soient des entiers.

9. (TSP)
10. (MSPT)
11. (SET COVER)
12. (UST)
13. (DST)
14. (KNAPSACK)
15. (LP)
16. (ILP)
17. (FLOW)
18. (MIN SUDOKU)