Exercice 1 (Hahn-Banach). Démontrer le théorème de Hahn-Banach utilisé en cours.

Exercice 2 (**Points extrêmes et solution des programmes linéaires**) Soit $\mathcal{P}$ le polyèdre dans $\mathbb{R}^n$ défini par les points x satisfaisant $a_1^t x \leq b_1, \ldots, a_m^t x \leq b_m$. De manière plus concise, on a donc

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}. \tag{1}$$

On suppose de plus que $\mathcal{P}$ est borné et dans ce cas on dit que $\mathcal{P}$ est un polytope. On appelle *face* de ce polytope tout sous-ensemble F de $\mathcal{P}$ pour lequel il existe un vecteur c tel que l'ensemble des solutions du programme linéaire

$$\min_{x \in \mathcal{P}} c^t x \tag{2}$$

est exactement F. On appelle point extrémal de $\mathcal{P}$ tout point x de $\mathcal{P}$ tel que si il existe x_1 et x_2 dans $\mathcal{P}$ et t dans $(0, 1)$ tels que $x = tx_1 + (1 - t)x_2$ alors $x_1 = x_2$. Les points extrêmes sont parfois très vulagirement traités de *coin* du polytope $\mathcal{P}$.

Montrer que pour tout programme linéaire sur $\mathcal{P}$ il existe au moins un point extrême qui est solution.

Exercice 3 (**Application de la programmation linéaire aux flots dans les graphes**) On considère un graphe constitué de n noeuds et d'arêtes. On suppose de plus que les arêtes sont orientées, c'est-à-dire que l'on donne un sens de parcours sur chaque arête. Le sens de parcours est repésenté par une flèche lorsqu'on dessine le graphe. On numérote les noeuds de 1 à n et les arêtes de 1 à m.

a. Dessinez un graphe orienté de votre choix avec 5 noeuds et autant d'arêtes orientées que souhaité. Attendez 5mn. Laissez mijoter a à feux doux. Le plus jeune de la classe choisit un voisin (à gauche, à droite, devant ou derrière) de manière équiprobable. Si son graphe n'est pas le même que celui de son voisin, tirez à pile ou face pour savoir lequel des deux conserve sont graphe, l'autre recopiant le graphe de son voisin. Passez alors au second plus jeune et recommencez. Lorsqu'on a atteint le plus âgé, on repart avec le plus jeune. Quand ce processus a atteint un graphe commun à toute la classe, compter le nombre d'itérations dont vous avez eu besoin pour en arriver là et passez à la question suivante. (NB. Il est aussi autorisé de se mettre d'accord dès le début du TD sur un même graphe).

La matrice d'incidence I d'un graphe orienté est la matrice dont les lignes sont indicées par les noeuds et les colonnes sont indicées par les arêtes et telle que pour chaque colonne, j on pose $I_{ij} = 0$ si l'arête j n'est pas connectée au noeud i, $I_{ij} = -1$ si l'arête quitte le noeud i et $I_{ij} = 1$ si l'arête arrive au noeud i.

b. Donner la matrice d'incidence du graphe commun.

le problème que l'on considère maintenant est celui d'acheminer des marchandises de certains noeuds dans ce graphe à d'autres noeuds dans le graphe avec un coût minimum. Ce problème intervient en particulier dans les problème d'acheminement de pétrole mais aussi dans la gestion de la distribution de l'information dans les réseaux informatiques et c'est pourquoi de tels graphes orientés sont aussi appelés réseaux orientés. On suppose que chaque arête a un coût de transport c_j. Les variables à déterminer sont les quantités x_j de produit qui sont transportées sur l'arête j. On suppose que à chaque noeud i, on a une loi de conservation qui dit que la quantité totale quittant ce noeud est égale à la quantité totale y arrivant au noeud plus une quantité b_i, qui modélise le fait que certains noeud sont des sources (lorsque $b_i > 0$), des destinations (lorsque $b_i < 0$) ou simplement des points de passage (lorsque $b_i = 0$). On suppose enfin que la somme des b_i est égale à zero de manière à ce qu'il y ait autant de quantité introduite dans le réseau que de quantité absorbée.

c. Modélisez ce problème comme un problème de programmation linéaire en utilisant la matrice d'incidence I.

NB. Ce problème est appelé *le problème de flot* et est très important du point de vue culturel dans la commnunauté des gens qui font de la recherche opérationnelle, étant apparu historiquement très tôt et ayant des relations étroites avec certains problèmes d'optimisation combinatoire que nous verrons dans la suite du cours.

Exercice 4 (**Problème linéaire booléen**). On considère le problème suivant appelé *Problème linéaire booléen.* Il s'agit de résoudre un problème de programmation linéaire avec comme contrainte supplémentaire le fait que les variables sont booléennes, c'est à dire qu'elle sont soit 0 soit 1. On obtient donc le problème

$$\min_{x \in \mathbb{R}^n} c^t \text{ t. q. } Ax \leq b \text{ et } x \in \{0,1\}^n. \tag{3}$$

Ce problème modélise une quantité inépuisable de problèmes qu'on appelle problèmes d'optimisation combinatoire.

a. Ce problème est-il convexe ?

On appelle *relaxation linéaire* du problème linéaire booléen, le programme linéaire

$$\min_{x \in \mathbb{R}^n} c^t \text{ t. q. } Ax \leq b \text{ et } 0 \leq x_i \leq 1, \;\; i = 1, \ldots, n. \tag{4}$$

b. Montrer que la valeur optimale de la relaxation linéaire est inférieure à la valeur optimale du problème linéaire booléen.

c. Il arrive parfois, comme on va le montrer dans la suite du cours qu'il existe une solution x^* à la relaxation linéaire telle que $x^* \in \{0,1\}^n$. Le point x^* est-il alors solution du problème linéaire booléen de départ ?