Graphes orientés eulériens

On rappelle qu'un graphe orienté G est la donnée d'un quadruplet $(V, E, \partial_0, \partial_1)$ d'un ensemble fini V dit de *sommets*, d'un ensemble fini E dit d'*arcs*, et de deux fonctions $\partial_0, \partial_1 : E \to V$. Si e est un arc, $\partial_0 e$ est la *source* de e, et $\partial_1 e$ est son *but* ; on dit aussi que e *va de* $\partial_0 e$ à $\partial_1 e$.

Un *chemin* π dans G est une suite alternée $v_1 e_1 v_2 e_2 \dots v_{n-1} e_{n-1} v_n e_n v_{n+1}$ de sommets v_1, …, v_{n+1} et d'arcs e_1, …, e_n tels que $\partial_0 e_i = v_i$ et $\partial_1 e_i = v_{i+1}$ pour tout i, $1 \leq i \leq n$. La *longueur* d'un tel chemin est son nombre d'arcs, soit n. On dit que π *va de* v_1 à v_{n+1}, et l'on étend les notations ∂_0 et ∂_1 en posant $\partial_0 \pi = v_1$, $\partial_1 \pi = v_{n+1}$.

Un chemin π est un *circuit* si et seulement si $\partial_0 \pi = \partial_1 \pi$.

Pour chaque chemin π dans G, et chaque arc e dans G, soit $|\pi|_e$ le nombre de fois où π passe par e. Formellement, si $\pi = v_1 e_1 v_2 e_2 \dots v_{n-1} e_{n-1} v_n e_n v_{n+1}$, alors $|\pi|_e$ est le nombre d'indices i, $1 \leq i \leq n$, tels que $e_i = e$.

Pour chaque chemin π dans G, et chaque sommet v dans G, notons $in_v(\pi)$ le nombre de fois où π arrive au sommet v et $ex_v(\pi)$ le nombre de fois où π part de v. Formellement, si $\pi = v_1 e_1 v_2 e_2 \dots v_{n-1} e_{n-1} v_n e_n v_{n+1}$, alors $in_v(\pi)$ est le nombre d'indices i, $1 \leq i \leq n$, tels que $v = v_{i+1}$, et $ex_v(\pi)$ est le nombre d'indices i, $1 \leq i \leq n$, tels que $v = v_i$.

1. Montrer que si π est un circuit, alors pour tout sommet v de G, $in_v(\pi) = ex_v(\pi)$.
2. On rappelle que le *degré entrant* in_v d'un sommet v dans G est le nombre d'arcs e tels que $\partial_1 e = v$; le *degré sortant* ex_v de v est le nombre d'arcs e tels que $\partial_0 e = v$.

   Un circuit π de G est dit *eulérien* si et seulement si π passe par tout arc de G exactement une fois, autrement dit si $|\pi|_e = 1$ pour tout e.

   Montrer que si G a un circuit eulérien, alors G vérifie la *loi de Kirchoff* : $in_v = out_v$ pour tout sommet v de G.
3. On rappelle qu'un graphe orienté G est *fortement connexe* si et seulement si, pour tous sommets v, v' de G, il existe un chemin de v à v'.

   Un *sommet isolé* de G est un sommet de degré entrant et de degré sortant nuls.

   Montrer que si G a un circuit eulérien, et si G n'a pas de sommet isolé, alors G est fortement connexe.
4. On considère un chemin π tel que $|\pi|_e \leq 1$ pour tout arc e de G, et maximal avec cette propriété. Montrer que si G vérifie la loi de Kirchoff, alors π est un circuit.
5. Si de plus $|\pi|_e = 0$ pour un arc e, et si G est fortement connexe, montrer qu'il existe un sommet v par lequel passe π, et un circuit π' de v à v passant par e, et qui ne passe que par des arcs par lesquels π ne passe pas.

   En déduire que π est eulérien. On aura donc montré qu'un graphe fortement connexe vérifiant la loi de Kirchoff a un circuit eulérien.
6. Décrire un algorithme efficace prenant en entrée en un graphe fortement connexe G, et retournant NON s'il n'a pas de circuit eulérien, et retournant un circuit eulérien de G sinon.
7. Comment peut-on tester si G est fortement connexe en temps polynomial ?

Graphes non orientés eulériens

On rappelle qu'un graphe (non orienté) G est la donnée d'un couple (V, E) d'un ensemble fini V dit de *sommets*, et d'un ensemble fini E de paires $\{v_0, v_1\}$ de sommets ($v_0 \neq v_1$). Les éléments de E sont appelés les *arêtes*.

Un *chemin* π dans G est une suite alternée $v_1 e_1 v_2 e_2 \dots v_{n-1} e_{n-1} v_n e_n v_{n+1}$ de sommets v_1, …, v_{n+1} et d'arêtes e_1, …, e_n tels que $e_i = \{v_i, v_{i+1}\}$ pour tout i, $1 \leq i \leq n$. La *longueur* d'un tel chemin est son nombre d'arêtes, soit n. On dit que π *va de* v_1 à v_{n+1}.

Un chemin π est un *cycle* s'il va d'un sommet v au même sommet v.

Pour chaque chemin π dans G, et chaque arête e dans G, soit $|\pi|_e$ le nombre de fois où π passe par e. Formellement, si $\pi = v_1 e_1 v_2 e_2 \dots v_{n-1} e_{n-1} v_n e_n v_{n+1}$, alors $|\pi|_e$ est le nombre d'indices i, $1 \leq i \leq n$, tels que $e_i = e$.

Pour chaque chemin π dans G, et chaque sommet v dans G, notons $in_v(\pi)$ le nombre de fois où π arrive au sommet v et $ex_v(\pi)$ le nombre de fois où π part de v. Formellement, si $\pi = v_1 e_1 v_2 e_2 \dots v_{n-1} e_{n-1} v_n e_n v_{n+1}$, alors $in_v(\pi)$ est le nombre d'indices i, $1 \leq i \leq n$, tels que $v = v_{i+1}$, et $ex_v(\pi)$ est le nombre d'indices i, $1 \leq i \leq n$, tels que $v = v_i$.

1. Montrer que si π est un cycle, alors pour tout sommet v de G, $in_v(\pi) = ex_v(\pi)$.
2. On rappelle que le *degré* deg_v d'un sommet v dans G est le nombre d'arêtes e telles que $v \in e$.

   Un cycle π de G est dit *eulérien* si et seulement si π passe par tout arête de G exactement une fois, autrement dit si $|\pi|_e = 1$ pour tout e.

   Montrer que si G a un cycle eulérien, alors G vérifie la *loi de Kirchoff* : deg_v est pair pour tout sommet v de G.
3. On rappelle qu'un graphe G est *connexe* si et seulement si, pour tous sommets v, v' de G, il existe un chemin de v à v'.

   Un *sommet isolé* de G est un sommet de degré nul.

   Montrer que si G a un cycle eulérien, et si G n'a pas de sommet isolé, alors G est connexe.
4. On considère un chemin π tel que $|\pi|_e \leq 1$ pour tout arête e de G, et maximal avec cette propriété. Montrer que si G vérifie la loi de Kirchoff, alors π est un cycle.
5. Si de plus $|\pi|_e = 0$ pour une arête e, et si G est connexe, montrer qu'il existe un sommet v par lequel passe π, et un cycle π' de v à v passant par e, et qui ne passe que par des arêtes par lesquels π ne passe pas.

   En déduire que π est eulérien. On aura donc montré qu'un graphe connexe vérifiant la loi de Kirchoff a un cycle eulérien.
6. Décrire un algorithme efficace prenant en entrée en un graphe connexe G, et retournant NON s'il n'a pas de cycle eulérien, et retournant un cycle eulérien de G sinon.
7. Comment peut-on tester si G est connexe en temps polynomial ?

Décision de systèmes d'inéquations simples

On considère des inéquations de la forme (1) $x \leq y + c$, où x et y sont des inconnues, ou *variables*, et c est une constante dans $\mathbb{Z}$. Un *système* S d'inéquations est un ensemble fini d'inéquations, vu comme une conjonction : une *solution* de S est une collection de valeurs réelles, une pour chaque inconnue, qui rend vrai chacune des inéquations de S.

1. À chaque système d'inéquations S on associe un graphe orienté étiqueté $G(S)$ comme suit. Les sommets de $G(S)$ sont les inconnues apparaissant dans S, et il y a un arc de x vers y étiqueté par l'entier c si et seulement s'il existe une inéquation $x \leq y + c$ dans S.

   Le *poids* d'un chemin est la somme des étiquettes des arcs de ce chemin. On rappelle qu'un *circuit* est un chemin d'un sommet vers lui-même. Un chemin est *simple* si et seulement si aucun sommet, sauf celui de départ, n'apparaît deux fois dans le circuit.

   Montrer que si S a une solution, $G(S)$ n'a pas de circuit de poids strictement négatif.

2. Supposons que $G(S)$ n'a pas de circuit de poids strictement négatif. Supposons de plus que $G(S)$ a une *racine* x_r, c'est-à-dire un sommet x_r depuis lequel tous les sommets sont accessibles. Montrer que pour tout sommet x dans $G(S)$, il existe un chemin de x_r vers x de poids minimum. Soit n_x ce poids minimum: montrer que la collection des valeurs $-n_x$ pour chaque sommet x est une solution de S.

3. En déduire que S a une solution ssi $G(S)$ n'a pas de circuit de poids strictement négatif. Lorsque $G(S)$ n'a pas de racine, on se ramènera au cas où $G(S)$ a une racine.

4. Soient $x_1, \ldots, x_n$ les variables de S. On construit la matrice $M(S) = (m_{ij})_{1 \leq i,j \leq n}$ en posant $m_{ij} =$ le plus petit poids d'un arc allant de x_i à x_j, ou $+\infty$ s'il n'y en a pas.

   $\mathbb{Z} \cup \{+\infty\}$ a une structure de semi-anneau où l'addition est $\min$ et la multiplication est $+$. (Un semi-anneau, informellement, est un anneau où les opposés n'existent pas nécessairement; autrement dit, c'est un monoïde pour la loi additive de même que pour la loi multiplicative, et les deux lois distribuent.) Cette structure induit une addition et une multiplication correspondante des matrices. Montrer que $I + M(S) + M^2(S) + \ldots + M^k(S)$, où $+$ est l'addition des matrices en question (le min entrée par entrée), et I est la matrice identité ($I_{ii} = 0$, $I_{ij} = +\infty$ si $i \neq j$), est une matrice dont l'entrée (i, j) est le poids du plus petit chemin de longueur $\leq k$ de x_i à x_j, ou $+\infty$ s'il n'existe pas de tel chemin.

5. En déduire un algorithme en temps polynomial testant si S a une solution ou non, et retournant une telle solution si elle existe.

6. Montrer que si S a une solution, et x_0 est une variable quelconque, alors S a une solution dans laquelle x_0 vaut 0. Pourquoi peut-on donc décider en temps polynomial des systèmes d'inéquations de la forme : (1) (cf. introduction), (2) $x \leq c$, (3) $x \geq c$?

7. En observant la forme des solutions retournée par l'algorithme, montrer qu'un système S a une solution ssi S a une solution *entière*, c'est-à-dire une où toutes les variables prennent des valeurs dans $\mathbb{Z}$.

Problèmes de sac à dos

1. On souhaite remplir un sac à dos avec des objets de sorte que leur valeur totale soit la plus grande possible. Les objets considérés sont classés en catégories 1, …, n. La masse d'un objet de catégorie i est $M_i > 0$, et sa valeur est $V_i > 0$. On a un nombre non borné d'objets de catégorie i pour tout i, mais le sac à dos ne peut supporter qu'une masse totale M sans casser.

   En notant x_i le nombre d'objets de catégorie i que l'on peut mettre dans le sac à dos, montrer que le problème du remplissage optimal du sac à dos décrit ci-dessus s'exprime en cherchant à maximiser

   $$\sum_{i=1}^{n} V_i x_i$$

   lorsque x_1, …, x_n sont dans $\mathbb{N}$, sujet à la contrainte

   $$\sum_{i=1}^{n} M_i x_i \leq M$$

2. Soit $V(m, M)$ la valeur maximale de $\sum_{i=1}^{m} V_i x_i$ sous la contrainte $\sum_{i=1}^{n} M_i x_i \leq M$, pour tout $m \leq n$. Trouver une relation de récurrence définissant $V(m, M)$.
3. En déduire un algorithme efficace calculant $V(n, M)$. Cet algorithme fonctionne-t-il en temps polynomial ? Discuter.
4. On suppose maintenant que l'on n'a plus qu'un seul objet de chaque catégorie (autrement dit chaque x_i varie dans $\{0, 1\}$), et que notre seule contrainte est que la valeur du sac à dos $\sum_{i=1}^{n} V_i x_i$ soit une valeur fixée V.

   Supposons que pour tout $m \leq n$, $V_m > \sum_{i=1}^{m-1} V_i$. Proposer un algorithme en temps polynomial répondant si l'objectif $\sum_{i=1}^{n} V_i x_i = V$ ($x_i \in \{0, 1\}$) est réalisable, et si oui retourne un vecteur $(x_1, \ldots, x_n)$ le réalisant. Ce vecteur est-il unique ?
5. Si la condition $V_m > \sum_{i=1}^{m-1} V_i$ n'est pas vérifiée, proposer un algorithme résolvant le même problème. En déduire que le problème est dans NP.