Devoir maison n°5

À rendre le mercredi 18/01

⋆⋆⋆

Programmation linéaire et relaxation continue

La programmation linéaire (*Linear Programming* ou `LP`) est un problème d'optimisation qui peut se présenter sous différents formats selon les contraintes imposées. La forme la plus générale qu'on peut lui donner est la suivante : `Programmation Linéaire (PL)` :

* **Instance** : une matrice $A \in \mathcal{M}_n(\mathbb{R})$ et deux vecteurs $B, C \in \mathcal{M}_{n,1}(\mathbb{R})$.
* **Solution** : un vecteur $X \in \mathcal{M}_{n,1}(\mathbb{R})$ vérifiant $AX \leqslant B$.
* **Optimisation** : Maximiser $C^T X$.

Il a été montré en 1979 que `PL` peut être résolu en temps polynomial, mais nous ne nous intéresserons pas aux algorithmes permettant de résoudre ce problème, dont la complexité temporelle a été améliorée jusqu'à environ $\mathcal{O}(n^{2,1} \log M)$ où M est la plus grande valeur absolue d'un coefficient d'une des matrices.

Ce problème est important car un grand nombre de problèmes d'optimisation peuvent être formulés en termes de programmation linéaire, avec plus ou moins de contraintes.

Question 1 Certaines formulations du problème imposent $X \geqslant 0$ comme condition supplémentaire de la solution, considèrent A comme une matrice non carrée, ou considèrent une minimisation de $C^T X$ plutôt qu'une maximisation. Justifier brièvement que ces formulations sont équivalentes à la précédente.

1 Programmation entière

On considère deux variantes de décision du problème :

- `Programmation Entière (PE)` :
  * **Instance** : une matrice $A \in \mathcal{M}_n(\mathbb{R})$ et un vecteur $B \in \mathcal{M}_{n,1}(\mathbb{R})$.
  * **Question** : existe-t-il un vecteur $X \in \mathcal{M}_{n,1}(\mathbb{Z})$ vérifiant $AX \leqslant B$?
- `Équations Zéro-Un (EZU)` :
  * **Instance** : une matrice $A \in \mathcal{M}_n(\mathbb{R})$ et un vecteur $B \in \mathcal{M}_{n,1}(\mathbb{R})$.
  * **Question** : existe-t-il un vecteur $X \in \mathcal{M}_{n,1}(\{0,1\})$ vérifiant $AX \leqslant B$?

Le premier problème consiste à déterminer l'existence d'un vecteur à coefficients entiers (même si ceux des entrées ne le sont pas), le deuxième à coefficients dans $\{0,1\}$. Cette modification qui peut sembler anodine transforme un problème dans P en un problème NP-complet.

Question 2 Montrer que `EZU` $\leqslant_m^p$ `PE`.

On cherche à prouver la NP-complétude par une réduction depuis `3-SAT`. On considère une formule en 3-FNC $\varphi = \bigwedge_{i=1}^{m} C_i$ où C_i est une clause disjonctive à trois littéraux, sur l'ensemble de variables $\mathcal{V} = \{v_1, \ldots, v_n\}$. On suppose que chaque C_i ne contient qu'au plus une occurrence de chaque variable (avec une négation ou non).

Question 3 Soit $\mu \in \{0,1\}^{\mathcal{V}}$ une valuation et $C = \ell_1 \vee \ell_2 \vee \ell_3$ une clause de φ. Montrer que $\mu(C) = 1$ si et seulement si $\mu(\ell_1) + \mu(\ell_2) + \mu(\ell_3) \geqslant 1$.

Question 4 En déduire `3-SAT` $\leqslant_m^p$ `EZU`.
Indication : on cherchera un vecteur X de la forme $X = (\mu(v_1)\mu(v_2)\cdots\mu(v_n))^T$.

Question 5 En déduire que `EZU` et `PE` sont NP-complets.

Pour toute la suite, on utilisera les noms `PE` et `EZU` pour désigner les variantes d'optimisation des deux problèmes précédents (c'est-à-dire dont la formulation est similaire à `PL` en imposant $X \in \mathcal{M}_{n,1}(\mathbb{Z})$ ou $X \in \mathcal{M}_{n,1}(\{0,1\})$).

2 Couverture des arêtes par les sommets

On considère ici la variante d'optimisation du problème de couverture par les sommets déjà étudié dans les chapitres précédents : `Couverture par Sommets` :

* **Instance** : un graphe non orienté $G = (S, A)$.
* **Solution** : une couverture des arêtes par les sommets C, c'est-à-dire un ensemble $C \subseteq S$ tel que pour tout $a \in A$, $a \cap C \neq \emptyset$.
* **Optimisation** : minimiser $|C|$.

On suppose pour simplifier que $S = [\![1, n]\!]$ pour tout graphe $G = (S, A)$ considéré.

On considère une formulation de `EZU` comme :

* **Instance** : un graphe non orienté $G = (S, A)$.
* **Solution** : un vecteur $X \in \mathcal{M}_{n,1}(\{0, 1\})$ vérifiant $x_i + x_j \geqslant 1$ pour tout $\{i, j\} \in A$.
* **Optimisation** : minimiser $\sum\limits_{i=1}^{n} x_i$.

Question 6 Justifier que trouver la solution au problème précédent est équivalent à trouver une solution de `Couverture par Sommets`.

On considère maintenant une formulation de `PL` comme :

* **Instance** : un graphe non orienté $G = (S, A)$.
* **Solution** : un vecteur $X \in \mathcal{M}_{n,1}(\mathbb{R})$ vérifiant $0 \leqslant x_i \leqslant 1$ pour tout $i \in S$ et $x_i + x_j \geqslant 1$ pour tout $\{i, j\} \in A$.
* **Optimisation** : minimiser $\sum\limits_{i=1}^{n} x_i$.

Cette généralisation du problème est appelée **relaxation continue** (on passe d'un problème discret « difficile » à un problème continu « facile »).

On note $OPT_{\texttt{EZU}}(G)$ et $OPT_{\texttt{PL}}(G)$ la valeur minimale $\sum\limits_{i=1}^{n} x_i$ obtenue dans les deux problèmes précédents.

Question 7 Montrer que pour un graphe G, $OPT_{\texttt{PL}}(G) \leqslant OPT_{\texttt{EZU}}(G)$.

Si $X = (x_1 x_2 \cdots x_n)^T$ est une solution au problème `PL` précédent, on note $Y = (y_1 y_2 \cdots y_n)^T$ défini par $y_i = \begin{cases} 0 & \text{si } x_i < \frac{1}{2} \\ 1 & \text{sinon} \end{cases}$, c'est-à-dire la solution entière obtenue en arrondissant la solution continue.

Question 8 Montrer que $\sum\limits_{i=1}^{n} y_i \leqslant 2 OPT_{\texttt{EZU}}(G)$.

Question 9 En déduire une 2-approximation de `Couverture par Sommets`.

3 Couverture d'ensemble

On considère le problème `Couverture d'Ensemble` :

* **Instance** : un entier N et une collections d'ensembles $S_1, \ldots, S_n \in \mathcal{P}([\![1, N]\!])$.
* **Solution** : un sous-ensemble $I \subseteq [\![1, n]\!]$ tel que $\bigcup\limits_{i \in I} S_i = [\![1, N]\!]$.
* **Optimisation** : minimiser $|I|$.

Pour $x \in [\![1, N]\!]$, on note $J(x) = \{j \in [\![1, n]\!] \mid x \in S_j\}$.

Question 10 Donner une formulation équivalente de ce problème comme un problème `EZU`.

La relaxation continue de ce problème peut se formuler comme

* **Instance** : un entier N et une collections d'ensembles $S_1, \ldots, S_n \in \mathcal{P}(\llbracket 1, N \rrbracket)$.
* **Solution** : un vecteur $X \in \mathcal{M}_{n,1}(\mathbb{R})$ vérifiant $0 \leqslant x_i \leqslant 1$ pour tout $i \in \llbracket 1, n \rrbracket$ et $\sum\limits_{j \in J(x)} x_j \geqslant 1$ pour tout $x \in \llbracket 1, N \rrbracket$.
* **Optimisation** : minimiser $\sum\limits_{i=1}^{n} x_i$.

Si $X = (x_1 x_2 \cdots x_n)^T$ est une solution au problème `PL` précédent, on définit l'algorithme `Couv_Alea` suivant :

> **Début algorithme**
> $I \leftarrow \emptyset$.
> **Pour** $i = 1$ à n **Faire**
> Ajouter i à I avec probabilité x_i.
> **Renvoyer** I.

Question 11 Montrer que si I est l'ensemble renvoyé par l'algorithme précédent, alors $\mathbb{E}(|I|)$ est inférieur ou égal à la taille d'une solution optimale à `Couverture d'Ensemble`.

Question 12 Montrer que pour $x \in \llbracket 1, N \rrbracket$, $\mathbb{P}\left(x \in \bigcup\limits_{i \in I} S_i\right) \geqslant 1 - \dfrac{1}{\mathrm{e}}$.

Indication : on pourra utiliser, après l'avoir montrée, l'inégalité $1 - t \leqslant e^{-t}$ *pour* $t \geqslant 0$.

Pour $k \in \mathbb{N}^*$, on définit maintenant l'algorithme `Couv_Alea2` :

> **Début algorithme**
> $I \leftarrow \emptyset$.
> **Pour** $i = 1$ à $\lceil 2 \ln N \rceil$ **Faire**
> $I \leftarrow I \cup$ `Couv_Alea()`.
> **Renvoyer** I.

Question 13 Montrer que si I est l'ensemble renvoyé par l'algorithme précédent, alors pour $x \in \llbracket 1, N \rrbracket$, $\mathbb{P}\left(x \in \bigcup\limits_{i \in I} S_i\right) \geqslant 1 - \dfrac{1}{N^2}$.

Question 14 En déduire que l'algorithme précédent renvoie une couverture de l'ensemble $\llbracket 1, N \rrbracket$ avec une probabilité $\geqslant 1 - \frac{1}{N}$.

Question 15 On note I^* une solution optimale à `Couverture d'Ensemble`. Montrer que :

$$\mathbb{P}(|I| \leqslant (4 \ln N + 2)|I|^*) \geqslant \frac{1}{2}$$

Indication : on pourra utiliser l'inégalité de Markov.

Remarque

> La méthode de relaxation continue est parfois utilisée comme heuristique d'évaluation dans un algorithme *Branch and Bound*. C'est cette méthode qui a été appliquée en considérant le problème `Sac à dos fractionnaire`.

⋆ ⋆ ⋆