Ré-formé en 2006

Agrégation externe de mathématiques, session 2005
Épreuve de modélisation, option calcul scientifique : méthodes numériques et symboliques

(560) RÉSOLUTION DE SYSTÈMES LINÉAIRES EN ENTIERS NATURELS

Résumé : On étudie les solutions à coefficients dans **N** d'un système d'équations linéaires.

Thème applicatif, mots clefs : équations linéaires, géométrie affine euclidienne.

► *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

1. Introduction

La résolution de systèmes linéaires est un problème connu quand on résout dans un corps ou dans un anneau euclidien. Ici, on s'intéresse à la résolution de systèmes d'équations linéaires dans une structure beaucoup moins riche, celle de monoïde, **N** en l'occurrence. L'étude proposée se borne au seul cas d'équations linéaires homogènes.

2. Notations

Un système linéaire homogène en entiers naturels à p équations et q inconnues s'écrit sous la forme

$$\begin{cases} a_{11}x_1 & +\cdots+ & a_{1j}x_j & +\cdots+ & a_{1q}x_q & = & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{i1}x_1 & +\cdots+ & a_{ij}x_j & +\cdots+ & a_{iq}x_q & = & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{p1}x_1 & +\cdots+ & a_{pj}x_j & +\cdots+ & a_{pq}x_q & = & 0 \end{cases}$$

où, pour $1 \leq i \leq p$ et $1 \leq j \leq q$, $a_{ij} \in \mathbf{Z}$ et où l'inconnue $x = (x_1, \ldots, x_q) \in \mathbf{N}^q$: les solutions sont donc des q-uplets d'entiers naturels. On note A la matrice du système et a l'application linéaire de $\mathbf{R}^q$ dans $\mathbf{R}^p$ associée à A dans la base canonique $(e_1, \ldots, e_q)$ de $\mathbf{R}^q$. Pour $1 \leq j \leq q$, $a(e_j) = (a_{1j}, \ldots, a_{pj})$ est appelé *le $j^{ème}$ vecteur défaut du système.*

3. Ensembles de solutions

L'ensemble $\mathscr{S}$ des solutions d'un tel système est stable par combinaisons linéaires à coefficients entiers naturels, ce qui lui confère une structure de monoïde additif.

Une partie $\mathscr{G}$ de $\mathscr{S}$ est appelée une *famille génératrice* de $\mathscr{S}$ si toute solution de $\mathscr{S}$ s'écrit comme combinaison linéaire à coefficients entiers naturels d'éléments de $\mathscr{G}$.

On munit $\mathbf{N}^q$ de l'ordre partiel strict $\gg$, qui étend l'ordre naturel $>$ sur les entiers :
$(x_1,\ldots,x_q) \gg (x'_1,\ldots,x'_q)$ si $\forall i \in [\![1,q]\!]\ \ x_i \geq x'_i$ et $\exists j \in [\![1,q]\!]\ \ x_j > x'_j$.
On dit alors que $x=(x_1,\ldots,x_q)$ *domine strictement* $x'=(x'_1,\ldots,x'_q)$.

Une solution $x=(x_1,\ldots,x_q)$ est dite *minimale* si ce n'est pas la solution triviale $(0,\ldots,0)$ et qu'elle ne domine strictement aucune solution autre que $(0,\ldots,0)$.

On appelle *famille génératrice minimale* toute famille génératrice de $\mathscr{S}$ constituée exclusivement de solutions minimales du système.

Par exemple :

(S_1) $\quad -x_1 + x_2 + 2x_3 - 3x_4 = 0$ possède 6 solutions minimales ;

(S_2) $\begin{cases} -x_1 + x_2 + 2x_3 - 3x_4 = 0 \\ -x_1 + 3x_2 - 2x_3 - x_4 = 0 \end{cases}$ possède 2 solutions minimales ;

(S_3) $\begin{cases} 4x_1 + 6x_2 + 2x_3 - 2x_4 = 0 \\ -3x_1 + x_2 + 5x_3 + 3x_4 = 0 \end{cases}$ ne possède pas de solution minimale ;

(S_4) $\begin{cases} -7x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 0 \\ -5x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 0 \\ -3x_5 + x_6 + x_7 + x_8 = 0 \end{cases}$

possède 11942 solutions minimales.

On démontre le résultat suivant :

Proposition 1. *Tout système linéaire homogène en entiers naturels admet une unique famille génératrice minimale (éventuellement vide si l'unique solution du système est la solution nulle). De plus, cette famille génératrice est de cardinal fini.*

4. Généralités sur les algorithmes de résolution

On décrit ici quelques algorithmes de détermination de la famille génératrice minimale $\mathscr{M}$ de l'ensemble des solutions d'un système linéaire homogène en entiers naturels.

Le principe de base de ces algorithmes consiste à obtenir les solutions minimales par construction de séquences de q-uplets d'entiers naturels dans lesquelles une composante est incrémentée à chaque étape. Par exemple, une solution minimale égale à $(0,1,2,1)$ peut être obtenue par la séquence $(0,0,1,0),(0,0,1,1),(0,0,2,1),(0,1,2,1)$ (parmi plusieurs autres possibles). La solution triviale nulle étant exclue de la recherche, les calculs de toutes les séquences sont initialisées par un vecteur de la base canonique. De plus, les calculs de toutes les séquences possibles doivent être menés en parallèle, de façon à obtenir toutes les solutions minimales en temps fini et à stopper le développement d'une séquence dès que le dernier q-uplet calculé domine strictement une solution minimale déjà obtenue.

Ce faisant, on est amené à engendrer tous les q-uplets strictement dominés par une solution minimale. Cependant, il est possible de limiter de manière sensible l'exploration de cet espace de recherche en autorisant l'incrémentation des composantes des q-uplets uniquement lorsqu'une certaine contrainte géométrique (C) est satisfaite. Remarquons toutefois que le même q-uplet peut participer à plusieurs séquences différentes.

L'algorithme générique $\mathscr{A}$ (dépendant de la contrainte (C)) donné en figure 1 formalise plus précisément les idées précédentes.

```
L ← {e_1,...,e_q}
M ← ∅
tant que L ≠ ∅ faire
  L' ← ∅
  pour chaque x ∈ L faire
    s'il n'existe pas v ∈ M ∪ [L \ {x}] tel que v = x ou x ≫ v
      alors
        si a(x) = 0
          alors M ← M ∪ {x}
          sinon L' ← L' ∪ {x + e_j tel que 1 ≤ j ≤ q et (x, e_j) vérifie la contrainte (C)}
  fin pour
  L ← L'
fin tant que
renvoyer M
```

FIG. 1. Algorithme $\mathscr{A}$

La recherche d'une contrainte (C) judicieuse est capitale. Aussi va-t-on procéder progressivement du cas particulier d'une seule équation au cas d'un système général.

5. Cas particulier d'un système réduit à une seule équation

Commençons par examiner le cas d'une équation unique

$$a_{11}x_1 + \cdots + a_{1j}x_j + \cdots + a_{1q}x_q = 0.$$

L'entier relatif $a_{11}x_1 + \cdots + a_{1j}x_j + \cdots + a_{1q}x_q$ représente le *défaut* du vecteur $x = (x_1, \ldots, x_q)$: si ce défaut est nul, on détient une solution ; s'il est négatif, on essaie de l'augmenter ; s'il est positif, on essaie de le diminuer. La contrainte géométrique mentionnée dans l'algorithme $\mathscr{A}$ est alors la suivante :

(C_1) Etant donnés un vecteur x de $\mathbf{N}^q$ et un vecteur e_j de la base canonique, le couple (x, e_j) vérifie la contrainte (C_1) si $a(x) \times a(e_j) < 0$.

6. Algorithme de résolution d'un système équation par équation

Disposant d'un algorithme pour une seule équation, on peut en déduire un algorithme général pour un système quelconque par substitutions successives.

Notons $a^{(1)}(x) = 0, \ldots, a^{(p)}(x) = 0$ les p équations du système $a(x) = 0$ et notons $s_1, \ldots, s_l$ les solutions minimales de sa première équation $a^{(1)}(x) = 0$ (que l'on peut obtenir par l'algorithme du § 5). Toute solution minimale du système complet est une solution (peut-être pas minimale) de cette première équation ; elle est donc de la forme $\sum_{k=1}^{k=l} y_k s_k$, où les y_k sont des entiers naturels. La résolution du système d'origine se ramène donc à celle du nouveau système à p équations et l inconnues : $\sum_{k=1}^{k=l} y_k a(s_k) = 0$. Mais la première équation de ce nouveau système est triviale ($0 = 0$), si bien qu'il s'agit en fait d'un système à $p - 1$ équations et l inconnues.

Il suffit alors de réitérer le processus.

7. Algorithme global de résolution

L'idée consiste à généraliser directement l'algorithme présenté dans le § 5. Le principe est exactement le même, si ce n'est que la contrainte (C_1) unidirectionnelle est remplacée par la contrainte (C_p) suivante, pour tenir compte du fait que l'espace de recherche est cette fois de dimension p :

(C_p) Etant donnés un vecteur x de $\mathbf{N}^q$ et un vecteur e_j de la base canonique, le couple (x, e_j) vérifie la contrainte (C_p) si $\langle a(x), a(e_j) \rangle < 0$, où $\langle , \rangle$ désigne le produit scalaire usuel dans $\mathbf{R}^p$.

8. Propriétés des algorithmes

Les trois algorithmes décrits précédemment satisfont les propriétés suivantes :

- la propriété de *correction*, à savoir qu'ils ne calculent que des solutions minimales ;
- la propriété de *complétude*, à savoir qu'ils calculent toutes les solutions minimales ;
- la propriété de *terminaison*, à savoir qu'ils s'arrêtent au bout d'un nombre fini d'étapes.

Signalons cependant que la preuve de terminaison de l'algorithme du § 7 est délicate.

Il faut également être bien conscient que le problème que l'on cherche à résoudre est difficile en général. On pourrait se demander s'il est possible d'obtenir des bornes sur le nombre de solutions minimales d'un système, mais l'exemple du système (S_4) (§ 3) n'est guère encourageant. De même, on pourrait essayer de borner la taille des solutions. De telles bornes ont été obtenues, mais elles sont toutes exponentielles et, par expérience, l'algorithme $\mathscr{A}$, dont on ne sait pas à ce jour estimer la complexité, est en général beaucoup plus efficace que l'algorithme naïf consistant à explorer de manière exhaustive un domaine calculé *a* priori.

Suggestions pour le développement

▶ *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins*

fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.

- D'un point de vue purement algorithmique, on pourra :
  - faire tourner l'algorithme $\mathscr{A}$ sur les systèmes (S_1) et (S_2) donnés au § 3 : une exécution graphique de l'algorithme par additions successives des vecteurs $a(e_j)$ sera appréciée ;
  - programmer l'algorithme $\mathscr{A}$ dans sa version la plus générale (§ 7) et programmer l'algorithme de résolution équation par équation (§ 6).
- En ce qui concerne les propriétés des algorithmes, on pourra :
  - proposer une interprétation géométrique des contraintes (C_1), puis (C_p) et justifier que ces contraintes n'empêchent pas d'assurer aux algorithmes respectifs les propriétés de complétude et de correction ;
  - dans le cas d'une seule équation seulement (§ 5), prouver la terminaison de l'algorithme $\mathscr{A}$ et proposer une borne sur la taille des solutions ;
  - caractériser géométriquement à l'aide des vecteurs $a(e_j)$ les systèmes qui n'admettent pas de solution minimale ;
  - commenter l'algorithme de résolution équation par équation (§ 6), en expliquant notamment pourquoi il est complet et à terminaison finie, et en décrire les inconvénients prévisibles ;
  - proposer une extension de l'algorithme pour résoudre un système linéaire *non homogène* en entiers naturels (le second membre du système étant constitué d'un p-uplet $(b_1,\ldots,b_p)$ d'entiers relatifs). On commencera par préciser l'allure de l'ensemble des solutions ;
  - proposer d'améliorer l'algorithme $\mathscr{A}$ de façon à éviter d'engendrer le même q-uplet de plusieurs façons différentes.
- On pourra également prouver tout ou partie de la proposition énoncée au § 3.