RAIRO. RECHERCHE OPÉRATIONNELLE

J. M. COUVREUR

S. HADDAD

J. F. PEYRE

Résolution paramétrée de familles de systèmes linéaires

RAIRO. Recherche opérationnelle, tome 26, nº 2 (1992), p. 183-206

<http://www.numdam.org/item?id=RO_1992__26_2_183_0>

Recherche opérationnelle/Operations Research
(vol. 26, n° 2, 1992, p. 183 à 206)

RÉSOLUTION PARAMÉTRÉE DE FAMILLES DE SYSTÈMES LINÉAIRES (*)

par J. M. COUVREUR [1], S. HADDAD [1] et J. F. PEYRE [1]

Résumé. — *La validation de modèles formels du parallélisme tels que les systèmes d'addition de vecteurs ou les réseaux de Petri nécessite la connaissance d'invariants positifs. Ces invariants se calculent par des techniques classiques de programmation linéaire telles que l'algorithme du Simplexe ou l'algorithme de Farkas. Or celles-ci sont inapplicables dans le cas de modèles de haut niveau tels que les réseaux colorés. Aussi présentons-nous ici une nouvelle méthode applicable à ces modèles de haut niveau. Cette méthode repose sur la résolution de manière paramétrée de différentes familles de systèmes d'équations linéaires de structures identiques, mais dont le nombre d'inconnues et d'équations diffèrent. La première famille d'équations que nous résolvons est :* $A.X_1=A.X_2=\ldots A.X_n$ *où A est une matrice et les inconnues* X_i *sont des vecteurs à coefficients positifs; n est le paramètre de cette famille. A l'aide de ce premier résultat nous obtenons la résolution de deux autres familles d'équations conduisant à l'obtention d'une famille génératrice d'invariants positifs dans deux types de réseaux colorés. La résolution de la troisième famille nécessite des techniques de calcul formel.*

Mots clés : Programmation linéaire; algorithme de Farkas; calcul paramétré; validation formelle.

Abstract. — *Validation of formal models of paralellism, such as vector-addition systems and Petri Nets, requires the calculation of nonnegative invariants. In general, these invariants are calculated by means of classical techniques of linear programming, such as the Simplexe algorithm or the Farkas' algorithm. Unfortunately, these techniques are not applicable in the case of High level models as Colored Nets. In this paper, we develop a new useful method, for this kind of models. This technique is based on the resolution — in a parametrized way — of some families of linear equations systems, whose structures are identical, but that have a different number of equations and unknowns. The first equations family that we solve is of the type* $A.X_1=\ldots=A.X_n$ *where A is an integer matrix and the unknowns* $\{X_1, \ldots, X_n\}$ *are a set of nonnegative vectors; n is the parameter of this family. Once solutions for the system have been obtained, we solve two other families of equations, whose solutions provide us a generative family of nonnegative invariants.*

Keywords : Linear programming; Farkas algorithm; parametrized computation; formal validation.

(*) Reçu juin 1991.
[1] Université P.-et-M.-Curie et C.N.R.S. M.A.S.I., 4, place Jussieu, 75252 Paris Cedex 05.

1. INTRODUCTION

Dans des modèles formels de systèmes parallèles tels que les systèmes d'addition de vecteurs ou les réseaux de Petri [Bra 83], l'état du système ainsi que les actions modifiant cet état sont représentés par des vecteurs. Aussi l'équation du changement d'état d'un état V vers un état V' par une séquence d'actions s'exprime :

$$V' = V + S^t . A$$

où S est le vecteur d'occurrences des actions dans la séquence et A est la matrice dont les lignes sont les vecteurs associés aux actions du système.

Pour un vecteur X vérifiant $A . X = 0$, on obtient en reportant dans l'équation du changement d'état $V' . X = V . X + S^t . A . X = V . X$, ainsi $V . X$ est un invariant du système. Pour la validation de tels systèmes on est donc intéressé aux solutions de $A . X = 0$ et plus précisément au calcul d'une famille génératrice de solutions. Il existe deux types de solutions qui sont intéressantes, les solutions à valeur dans $\mathbb{Q}$, ensemble des rationnels, dont une famille génératrice est calculée par une élimination de Gauss et les solutions à valeur dans $\mathbb{Q}^+$, ensemble des rationnels positifs, dont une famille génératrice est calculée par l'algorithme de Farkas [Mem 83].

Les modèles formels précédents ne permettent pas de modéliser et de valider des algorithmes ou des protocoles où par exemple le nombre de processus ou de sites est laissé variable. De nouveaux modèles ont donc été introduits tels que les réseaux de haut niveau qui permettent de modéliser des familles de systèmes à structure identique mais de taille différente [Gen 81], [Jen 86]. Pour chacun de ces systèmes l'équation de changement d'état s'écrit $V' = V + S^t . A_n$ où les A_n sont des matrices à structure identique mais dont la taille dépend de n. *Aussi on est intéressé à résoudre la famille d'équations* $\{ A_n . X = 0 \}_{n \in \mathbb{N}}$ *en une seule fois, c'est-à-dire, en laissant n variable — résolution paramétrée — ce qui est impossible avec les méthodes existantes — Farkas, Simplexe — qui de plus présentent une complexité prohibitive dès que n devient grand.*

Si pour les solutions à valeur dans $\mathbb{Q}$, de nombreux résultats sont maintenant connus [Had 86, Vau 84, Sil 85, Cou 90], il n'existait jusqu'à présent aucun résultat sur les solutions à valeur dans $\mathbb{Q}^+$. En effet les résultats obtenus pour $\mathbb{Q}$ s'appuient sur la décomposition des espaces vectoriels et des applications linéaires, ce qui est impossible dans $\mathbb{Q}^+$ qui n'est pas un anneau.

Nous présentons ici la résolution paramétrée de trois types de famille d'équations. Le premier type de famille est l'équation $A . X_1 = A . X_2 = \ldots = A . X_n$ où les X_i sont les vecteurs inconnus. L'idée générale de la résolution est que le système pour un n assez grand dépendant de A, fournit la forme générale d'une famille génératrice pour n quelconque. Cette famille de systèmes d'équations ne correspond pas à un modèle particulier mais elle sert de base à la résolution paramétrée de deux autres types de familles qui sont issues de modèles connus dans les extensions des réseaux de Petri — réseaux à prédicats unaires [Vau 84] et réseaux réguliers unaires [Had 87] —. Dans sa thèse d'état Gérard Memmi [Mem 83] avait fourni quelques caractérisations sur la forme et la taille d'une famille génératrice pour les réseaux à prédicats unaires mais qui ne permettait pas d'obtenir la résolution paramétrée du système.

Nous rappelons d'abord les résultats sur le calcul d'une famille génératrice de systèmes d'équations à solutions rationnelles positives et en particulier l'algorithme de Farkas. Puis nous présentons l'algorithme de résolution du premier type de famille avec sa preuve. Ensuite nous appliquons ce résultat aux deux autres types de familles. Enfin nous concluons, en évoquant les autres types de familles d'équations susceptibles d'être résolus de manière paramétrée.

2. RÉSOLUTION DE $A.X=0$

2.1. Terminologie

Avant de rappeler la méthode la plus couramment utilisée pour résoudre un seul système d'équation, l'algorithme de Farkas [Far 02], il nous semble utile de faire un petit rappel de vocabulaire. En effet, deux points de vue différents peuvent être adoptés :

• le point de vue algébrique, adopté dans la théorie des réseaux de Petri, où l'on parlera de solutions positives de support minimal d'un système d'équations,

• le point de vue géométrique, adopté en programmation linéaire, où l'on parlera de direction extrémales — ou rayons extrémaux — d'un cône polyédrique.

TABLEAU

Correspondance des termes employés en algèbre et en géométrie convexe.

Algèbre	Géométrie Convexe
solution de $AX=0$, $X\geqq 0$	rayon infini associé à la matrice A
ensemble de solutions de $AX=0$, $X\geqq 0$	cône polyédrique $A.u=0$, $u\geqq 0$
solution de support minimal de $AX=0$, $X\geqq 0$	rayon extrémal
famille génératrice des solutions de $AX=0$, $X\geqq 0$	ensemble des rayons extrémaux du cône polyédrique
inclusion de support	dominance

Hormis ces différences de vocabulaire, les algorithmes développés dans chacun de ces domaines sont identiques et peuvent se diviser en deux grandes classes :

- les algorithmes proches du Simplexe basés sur des opérations de pivotement – on va de solutions en solutions –, sensibles à la dégénérescence des solutions et donc mal adaptés dans le cas de matrices creuses;
- les algorithmes duaux de la méthode de Fourier [Fou 27, Fou 90, Aba 64] dont le plus célèbre est l'algorithme de Farkas – en fait utilisé de façon implicite en page 11 de [Far 02] comme moyen de démonstration –.

Nous choisissons désormais de nous placer dans le contexte algébrique et de prendre comme algorithme de référence l'algorithme de Farkas. Le lecteur pourra se référer à [Chv 83] pour une étude de l'algorithme du Simplexe et à [Far 02], [Aba 64] pour une étude de l'algorithme de Farkas.

2.2. Algorithme de Farkas

Le principe de l'algorithme consiste à obtenir la famille génératrice pour le système réduit aux i premières équations à partir de celui réduit au i-l premières équations en conservant les vecteurs qui vérifient l'équation (i) et en combinant deux à deux les vecteurs pour lequel le résultat de l'équation (i) est positif avec ceux pour lequel le résultat est négatif.

L'inconvénient bien connu de cette méthode est que le nombre d'inconnues peut augmenter considérablement. On essaie donc dans la mesure du possible d'éliminer les solutions non nécessaires et l'on parle alors de méthode minimale [Aba 64].

DÉFINITION 2.1 : Soit A une matrice où T est l'ensemble des indices de lignes et P l'ensemble des indices de colonnes. Une famille génératrice des solutions dans $Q+$ de l'équation $A.X=0$ est une famille $\{V_1, \ldots, V_m\}$ de vecteurs non nuls de $(\mathbb{Q}^+)^P$ telle que :

– $\forall V_i, A.V_i=0$

– $\forall V \in (\mathbb{Q}^+)^P$ *avec* $A.V=0, \exists \lambda_1, \ldots, \lambda_m \in \mathbb{Q}^+$ *tels que*

$$V=\lambda_1 . V_1 + \ldots + \lambda_m . V_m$$

DÉFINITION 2.2 : Soit V un vecteur indicé par un ensemble E. On note $V=\sum_{e \in E} V_e . e$. Le support de V, noté $\mathrm{Supp}(V)$, est le sous-ensemble de E défini par $\mathrm{Supp}(V)=\{e \in E / V_e \neq 0\}$.

Exemple :

$$E=\{a, b, c, d\}; \text{si } V=2a+3c \text{ alors } \mathrm{Supp}(V)=\{a, c\}$$

Il existe une caractérisation simple d'une famille génératrice, caractérisation qui est le point de départ de l'algorithme de Farkas – on rejette les solutions non nécessaires –.

Caractérisation 2.1 (dite de génération) [Aba 64, Mem 83]

Soit une famille $F=\{V_1, \ldots, V_m\}$ de vecteurs non nuls de $(\mathbb{Q}^+)^P$ solutions de l'équation $A.X=0$. Alors F est une famille génératrice si et seulement si :

$$\forall V \in (\mathbb{Q}^+)^P \textit{ avec } A.V=0 \textit{ et } V \neq 0, \exists V_i \textit{ avec } \mathrm{Supp}(V_i) \subseteq \mathrm{Supp}(V)$$

On dit alors que V_i est de support plus petit ou égal à V.

Remarque : Le symbole $\subseteq$ dénote l'inclusion large – inclus ou égal –.

On peut chercher à minimaliser la taille d'une famille génératrice. La caractérisation qui suit répond à ce problème.

DÉFINITION 2.3 : Une famille génératrice $F=\{V_1, \ldots, V_m\}$ des solutions dans $\mathbb{Q}^+$ de l'équation $A.X=0$ est une plus petite famille génératrice si :

$$\forall i, F \setminus \{V_i\} \textit{ n'est pas une famille génératrice.}$$

Caractérisation 2.2 (dite de minimalité) [Mem 83]

Une famille génératrice F des solutions dans $\mathbb{Q}^+$ de l'équation $A.X=0$ est une plus petite famille génératrice si et seulement si :

$$\forall V \neq V' \in F \textit{ on n'a pas } \mathrm{Supp}(V') \subseteq \mathrm{Supp}(V)$$

Remarque : Soit $V \neq V'$ deux vecteurs de E solutions de $A.X=0$. Si $\mathrm{Supp}(V')=\mathrm{Supp}(V)$ alors V et V' sont proportionnels. Des deux caractérisations précédentes, on obtient l'algorithme de Farkas qui calcule une plus petite famille génératrice en trouvant une famille de vecteurs associés aux supports minimaux.

Algorithme de Farkas
$S := \{ e_k / k \in P \}$ # S est la base canonique de $\mathbb{Q}^P$ #
Pour i **parcourant** T **faire**
séparation des vecteurs de S en fonction de leur signe par rapport à l'équation (i)
  $S^0 := \{ s \in S / A_i . s = 0 \}$ # A_i est le i-ième vecteur de A #
  $S^+ := \{ s \in S / A_i . s > 0 \}$
  $S^- := \{ s \in S / A_i . s < 0 \}$
  $S := S^0$
    # combinaison des vecteurs de S^+ et de S^- #
  Pour s^+ **parcourant** S^+ **faire**
    Pour s^- **parcourant** S^- **faire**
      $S := S \cup \{ (A_i . s^+) . s^- - (A_i . s^-) . s^+ \}$
      Fin pour
  Fin pour
      # suppression des vecteurs de support non minimal #
  Pour s **parcourant** S **faire**
      Pour $s' \neq s$ **parcourant** S **faire**
        Si Supp (s) $\supset$ Supp (s') Alors
          $S := S / \{ s \}$
        Fin si
      Fin pour
  Fin pour
Fin pour # S est une plus petite famille génératrice #
Fin

Exemple : Soit la matrice $A = \begin{bmatrix} -2 & 0 & 1 & 2 & -1 \\ 1 & 2 & 1 & -2 & -2 \end{bmatrix}$
et soit e_1, e_2, e_3, e_4, e_5 la base canonique de $\mathbb{Q}^+$. S est initialement l'ensemble $\{ e_1, e_2, e_3, e_4, e_5 \}$.

On annule l'équation ($i = 1$)

Les cinq vecteurs de $S\, \mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, $\mathbf{e}_4$, $\mathbf{e}_5$ ont respectivement comme projection sur l'équation $i = 1$, c'est-à-dire par rapport à la première ligne de la matrice A les valeurs : $\mathbf{-2}$, $\mathbf{0}$, $\mathbf{2}$, $\mathbf{-1}$; On sépare alors ces vecteurs de S en trois ensembles :

$$S^0 = \{ e_2 \}, S^+ = \{ e_3, e_4 \}, S^- = \{ e_1, e_5 \}$$

On opère les combinaisons linéaires des vecteurs de S^+ et de S^- afin d'annuler l'équation ($i = 1$) et l'on obtient l'ensemble 3.

$$S = \{ e_2, (e_1 + 2\, e_3), (e_1 + e_4), (e_3 + e_5), (e_4 + 2\, e_5) \}.$$

Tous ces vecteurs étant minimaux il n'y a aucune suppression à effectuer.

On annule l'équation ($i = 2$)

Les cinq vecteurs de S $\mathbf{e}_2$, $(\mathbf{e}_1 + \mathbf{2}\, \mathbf{e}_3)$, $(\mathbf{e}_1 + \mathbf{e}_4)$, $(\mathbf{e}_3 + \mathbf{e}_5)$, $(\mathbf{e}_4 + \mathbf{2}\, \mathbf{e}_5)$ ont respectivement comme projection sur l'équation $i = 2$, c'est-à-dire par rapport à la deuxième ligne de la matrice A les valeurs : $\mathbf{2}$, $\mathbf{3}$, $\mathbf{-1}$, $\mathbf{-1}$, $\mathbf{-6}$;

Ce qui donne les trois ensembles :

$$S^0=\{\ \}, S^+=\{e_2, (e_1+2e_3)\}, \qquad S^-=\{(e_1+e_4), (e_3+e_5), (e_4+2e_5)\}$$

On opère les combinaisons linéaires des vecteurs de S^+ et de S^- afin d'annuler l'équation $(i=2)$ et l'on obtient l'ensemble S :

$$S=\{(e_2+2e_1+2e_4), (e_2+2e_3+2e_5), (3e_2+e_4+2e_5), (4e_1+2e_3+3e_4), (e_1+5e_3+3e_5), (2e_1+4e_3+e_4+2e_5)\}$$

On supprime alors le vecteur $(2e_1+4e_3+e_4+2e_5)$ dont le support contient celui du vecteur $(4e_1+2e_3+3e_4)$.
On obtient alors comme famille génératrice de $A.X=0$ la famille :

$$S=\{(\mathbf{e}_2+\mathbf{2}\,\mathbf{e}_1+\mathbf{2}\,\mathbf{e}_4), (\mathbf{e}_2+\mathbf{2}\,\mathbf{e}_3+\mathbf{2}\,\mathbf{e}_5), (\mathbf{3}\,\mathbf{e}_2+\mathbf{e}_4+\mathbf{2}\,\mathbf{e}_5), (\mathbf{4}\,\mathbf{e}_1+\mathbf{2}\,\mathbf{e}_3+\mathbf{3}\,\mathbf{e}_4), (\mathbf{e}_1+\mathbf{5}\,\mathbf{e}_3+\mathbf{3}\,\mathbf{e}_5)\}.$$

Pour plus de détails le lecteur pourra se reporter à [Mem 83] ou à [Col 89] dans lequel de nombreuses optimisations pour cet algorithme sont proposées.

3. RÉSOLUTION DE $A.X_1=A.X_2=\ldots=A.X_n$

3.1. Notations

– $\mathbb{N}$ désigne l'ensemble des entiers, $\mathbb{Z}$ l'ensemble des entiers relatifs

– Soit Y un ensemble, $\mathscr{P}(Y)$ désigne l'ensemble des parties de Y

– A désigne une matrice d'éléments de $\mathbb{Z}$ où T est l'ensemble des indices des lignes et P l'ensemble des indices des colonnes.

Dans ce qui suit, les solutions recherchées se trouveront donc dans l'espace $((\mathbb{Q}^+)^P)^n$ où n est un entier supérieur ou égal à 2. Aussi introduisons-nous les notations suivantes :

– $E=(\mathbb{Q}^+)^P$, d'où $E^n=((\mathbb{Q}^+)^P)^n$,

– E^n s'identifie à la somme directe de n copies de E, chaque copie étant notée E_i,

– d'où $E^n=\oplus E_i$, pour i de 1 à n,

– L'image d'un vecteur V de E par la bijection canonique de E dans E_i est notée $V(i)$

– Un vecteur V de E^n s'écrit donc de manière unique $\sum V_i(i)$ avec $V_i \in E$.

Ce vecteur sera noté $V=\langle V_1, V_2, \ldots, V_n\rangle$ avec $V_i=\langle f_{i,1}, f_{i,2}, \ldots, f_{i,p}\rangle$.

Lorsque cela sera nécessaire, on emploiera la notation somme directe : $V = \oplus V_i(i)$

— Le support d'un vecteur V de E^n est un sous-ensemble de P^n que l'on note — par symétrie avec la notation de E^n — :

$$\mathrm{Supp}(V) = (\mathrm{Supp}(V_1), \ldots, \mathrm{Supp}(V_n)).$$

Soient V, $W \in E^n$, on rappelle que $\mathrm{Supp}(V) \subset \mathrm{Supp}(W)$ si et seulement si :

$$\forall i, \quad \mathrm{Supp}(V_i) \subset \mathrm{Supp}(W_i).$$

Exemple : Soit $|P| = 3$, $n = 2$.

$$E = \mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+, \qquad E^2 = \langle \mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+ \rangle \times \langle \mathbb{Q}^+ \times \mathbb{Q}^+ \times \mathbb{Q}^+ \rangle.$$

Soit e_1, e_2, e_3 la base canonique de E.

Soient $V = \langle 4, 0, 2 \rangle$ et $W = \langle 0, 1, 0 \rangle$ vecteurs de E;

Soit $X = \langle V, W \rangle$ vecteur de E^2. Alors

$$X = \langle \langle 4, 0, 2 \rangle, \langle 0, 1, 0 \rangle \rangle \text{ et } X(1) = V, X(2) = W$$
$$\mathrm{Supp}(X) = (\{e_1, e_3\}, \{e_2\})$$

3.2. Résultats préalables

Nous allons essayer de caractériser à l'aide des propositions suivantes les solutions de support minimal de l'équation $A.X_1 = A.X_2 = \ldots = A.X_n$, équation que nous noterons (1).

PROPOSITION 3.1 : *Soit $V = \langle V_1, V_2, \ldots, V_n \rangle$ une solution non nulle de support minimal de l'équation* (1). *Soit $b_0 = A.V_1$ vecteur de $\mathbb{Q}^T$. Alors pour tout i, V_i est une solution de support minimal de l'équation $A.X = b_0$. On dira que b_0 est la direction de V.*

Preuve : Supposons qu'un des V_i soit de support non minimal dans l'ensemble des solutions de $A.X = b_0$. Alors il existe V'_{i_0} tel que $A.V'_{i_0} = b_0$ et $\mathrm{Supp}(V'_{i_0}) \subset \mathrm{Supp}(V_i)$. Soit le vecteur $W = \langle W_1, W_2, \ldots, W_n \rangle$ avec $W_i = V_i$ pour $i \neq i_0$ et $W_{i_0} = V'_{i_0}$. Alors par construction W est solution de (1) et $\mathrm{Supp}(W) \subset \mathrm{Supp}(V)$ ce qui contredit la minimalité de v. □

PROPOSITION 3.2 : *Soit $V = \langle V_1, V_2, \ldots, V_n \rangle$ une solution non nulle de support minimal de l'équation* (1) *de direction $b = 0 - \forall i, A.V_i = 0 -$. Alors il existe un seul j tel que $V_j \neq 0$ et V_j solution de support minimal de $A.X = 0$.*

Preuve : Immédiate du fait de la minimalité de V. □

PROPOSITION 3.3 : *Soit* $V = \langle V_1, V_2, \ldots, V_n \rangle$ *une solution de support minimal de l'équation* (1). *Alors pour tout vecteur* $W = \langle W_1, W_2, \ldots, W_n \rangle$ *solution de* (1) *non nulle et non proportionnel à* V, *il existe* i *tel que pour tout* j, $\operatorname{Supp}(W_j) \nsubseteq \operatorname{Supp}(V_i)$.

Preuve : Supposons que $\forall i, \exists j \operatorname{tq} \operatorname{Supp}(W_j) \subset \operatorname{Supp}(V_i)$ et soit alors $Y_i = W_j$. Alors le vecteur $Y = \langle Y_1, Y_2, \ldots, Y_n \rangle$ vérifie (1), $- \forall i, A.Y_i = A.W_j -$, et par construction Y est de support plus petit que V, ce qui contredit la minimalité de V. □

Les propositions 3.1 et 3.2 nous conduisent à généraliser l'algorithme de Farkas à la recherche des solutions de support minimal de l'équation $A.X = b$, la proposition 3.3 nous conduira à étendre la notion de support à une famille de vecteurs.

3.2.1. *Généralisation de l'algorithme de Farkas*

Nous venons de voir que, étant donné un vecteur b non nul de $\mathbb{Q}^T$, nous sommes intéressés par la recherche d'une famille de solutions du système $A.X = b$ vérifiant la caractérisation 2.1 dite de génération. Nous qualifions une telle famille de « *pseudo-génératrice* ».

DÉFINITION 3.1 : Soit une famille $\{V_1, \ldots, V_m\}$ de vecteurs non nuls de E, solutions de $A.X = b$. $F = \{V_1, \ldots, V_m\}$ est une famille pseudo-génératrice de l'équation $A.X = b$ si et seulement si :

$$\forall V \in E \text{ avec } A.V = b \text{ et } V \neq 0, \exists V_i \in F \text{ avec } \operatorname{Supp}(V_i) \subseteq \operatorname{Supp}(V)$$

Comme la caractérisation 2.1 l'établit, si b est égal à 0, une famille pseudo-génératrice est génératrice.

Algorithme de recherche d'une famille pseudo-génératrice

(1) Résoudre par Farkas $(A - b).\begin{pmatrix} X \\ \lambda \end{pmatrix} = 0$. Soit F la famille trouvée.
(2) Éliminer de F les vecteurs dont la composante en λ est nulle.
(3) Normaliser la famille en divisant les vecteurs par la coordonnée λ, puis éliminer la coordonnée $\lambda (= 1)$. On obtient alors une famille de vecteurs de E.

Fin

PROPOSITION 3.2 : *La famille obtenue par l'algorithme précédent est une famille pseudo-génératrice.*

Preuve :

- Chaque élément de la famille vérifie $A.X = b$ par construction.
- Toute solution V de $A.X = b$ fournit une solution (V, λ) de $A.X = \lambda.b$; cette solution est engendrée par la famille trouvée au point (1). Puisque la

composante en λ de (V, λ) est différente de 0, la combinaison linéaire qui permet de l'obtenir comprend au moins une solution V' de la famille obtenue au point (2). Alors la solution déduite de V' au point (3) a un support contenu dans le support de V. $\square$

Remarque : La famille pseudo-génératrice construite par cet algorithme vérifie la caractérisation de minimalité.

DÉFINITION 3.3 : Si $b \neq 0$, $\mathrm{Sol}(A, b)$ est une famille pseudo-génératrice de $A.X=b$.

$\mathrm{Sol}(A, 0)$ est une famille génératrice de $A.X=0$.

Comme nous l'avons déjà vu, tout vecteur b de $\mathbb{Q}^T$ induit pour tout $n \geqq 2$ un ensemble de solutions de l'équation (1); aussi posons-nous la définition suivante.

DÉFINITION 3.4 : Si $b \neq 0$,

$$\mathrm{Sol}^n(A, b)=\left\{ V=\bigoplus_{i=1}^{n} V_i(i) \in E_n / \forall i,\ V_i \in \mathrm{Sol}(A, b) \right\}.$$

si $b=0$, $\mathrm{Sol}^n(A, 0)=\{\langle 0, \ldots, 0, V_i, 0, \ldots, 0\rangle \in E^n / V_i \in \mathrm{Sol}(A, 0)\}$.

Remarquons que par définition, pour $V=\oplus V_i(i) \in \mathrm{Sol}^n(A, b)$, on a $A.V_i=A.V_j=b$, c'est-à-dire que V est une solution de l'équation (1).

3.2.2. *Extension de la notion de support*

La notion de support s'étend de manière naturelle aux familles de vecteurs :

DÉFINITION 3.5 :

$$\begin{aligned}&\mathrm{Supp}: \mathscr{P}(\mathrm{E}) \to \mathscr{P}(\mathscr{P}(P)).\\ &\text{telle que } \mathrm{Supp}(F)=\{\mathrm{Supp}(V) / V \in F\}\end{aligned}$$

On peut alors étendre la comparaison de support aux familles de vecteurs de la manière suivante : une famille de vecteurs F est de plus petit support qu'une famille F si pour tout V' de F' on peut trouver un V dans F de support plus petit ou égal.

DÉFINITION 3.6 : Soient F, $F \in \mathscr{P}(E)$, $F < F'$ si et seulement si :

$$\forall X' \in \mathrm{Supp}(F'), \exists X \in \mathrm{Supp}(F) \text{ tel que } X \subseteq X'$$

Cette dernière relation est évidemment réflexive et transitive.

Exemple : Soient F et F' deux familles de vecteurs de $(\mathbb{Q}^+)^3$,

$F=\{\langle 1, 0, 1\rangle, \langle 0, 0, 2\rangle\}$, $F'=\{\langle 2, 0, 1\rangle, \langle 0, 3, 1\rangle\}$. Alors $F<F'$.

La proposition suivante garantira la terminaison de l'algorithme de résolution de l'équation (1) présenté en 3.3 et traduit le fait qu'il ne peut exister de suite infinie de familles incomparables par rapport à la relation « $<$ ».

PROPOSITION 3.7 : *$\forall \{F_i\}_{i\in\mathbb{N}}$ avec $F_i\in\mathscr{P}(E)$, $\exists\, i, j$ avec $i<j$ tels que $F_i<F_j$.*

Preuve : Soit la suite $\{\mathrm{Supp}(F_i)\}_{i\in\mathbb{N}}$, cette suite est à valeurs dans $\mathscr{P}(\mathscr{P}(P))$. Or $\mathscr{P}(\mathscr{P}(P))$ est fini donc $\exists\, i<j$ tels que $\mathrm{Supp}(F_i)=\mathrm{Supp}(F_j)$. Par définition il vient $F_i<F_j$. $\square$

3.3. Algorithme de recherche d'une famille génératrice

Soit X un vecteur inconnu de E^n; ce vecteur s'écrit de manière unique $X=\sum X_i(i)$ avec $X_i\in E$. L'équation $A.X_1=A.X_2=\ldots=A.X_n$ peut donc se réécrire $A_n.X=0$ avec :

$$A_n=\begin{bmatrix} -A & A & 0 & \ldots & 0 \\ -A & 0 & A & \ldots & 0 \\ \vdots & \ldots & 0 & \ddots & \vdots \\ -A & 0 & 0 & \ldots & 0 \end{bmatrix}$$

L'intérêt de cette réécriture est l'utilisation de la caractérisation de génération pour vérifier qu'une famille est bien génératrice et de la caractérisation de minimalité pour la minimalité d'une telle famille. On remarque d'autre part que pour un b quelconque, $\mathrm{Sol}^n(A, b)$ fournit une famille de solutions. L'algorithme consiste donc à construire par enrichissements successifs une famille $\{b_i\}_{i=0\ldots m}$ telle que $\bigcup_{i=0\ldots m}\mathrm{Sol}^n(A, b_i)$ soit une famille génératrice et ceci pour tout $n\geqq 2$. Nous définissons tout d'abord la notion d'ensemble b_- générateur.

DÉFINITION 3.7 : *Soit* $S=\{b_0=0, b_1, \ldots, b_m\}$ une famille de vecteurs de $\mathbb{Q}^T$ et N_0 un entier supérieur ou égal à 2. Alors S est b_- générateur de (1) au rang N_0, ssi $\bigcup_{i=0\ldots m}\mathrm{Sol}^{N_0}(A, b_i)$ est une famille génératrice de $A.X_1=A.X_2=\ldots=A.X_{N_0}$.

Remarque : Si S est un ensemble b_- générateur de (1) au rang N_0, alors S est b_- générateur de (1) pour tout rang n avec $2\leqq n\leqq N_0$.

Nous allons maintenant nous intéresser aux conditions nécessaires pour qu'un ensemble S qui est b_- générateur de (1) au rang N_0 ne le soit plus au rang N_0+1; ce qu'établit le lemme suivant.

LEMME 3.1 : *Soit* $S=\{0, b_1, \ldots, b_m\}$ *une famille de vecteurs de* $\mathbb{Q}^T$ *tel que* S *soit* b_- *générateur de* (1) *au rang* N_0 *et ne le soit plus au rang* N_0+1.

Alors $\exists\, b' \in \mathbb{Q}^T$ *tq* :

(i) $b' \notin S$.

(ii) $\exists\, V \in E$, $\langle \lambda_1, \ldots, \lambda_m \rangle \in (\mathbb{Q}^+)^m / A \,.\, V = b'$ *et* $\langle V, \lambda_1, \ldots, \lambda_m \rangle$ *est une solution non nulle de support minimal de l'équation* $A\,.\,X=\sum_1^m \lambda_j b_j$.

(iii) $\nexists\, b \in S/\mathrm{Sol}(A, b) < \mathrm{Sol}(A, b')$.

Preuve : Soit $F_{S, N_0+1}=\{\bigcup_{i=1 \ldots m} \mathrm{Sol}^{N_0+1}(A, b_i)\, b_i \in S\}$ la famille induite par S au rang N_0+1. S n'étant pas b_- générateur au rang N_0+1, il existe — prop. 2.1 — des vecteurs de E^{N_0+1} solutions de (1) incomparables en support par rapports aux vecteurs de F_{S, N_0+1}. Soit M cet ensemble. Soit maintenant W un vecteur de support minimal dans M et b' son vecteur associé — $b'=A\,.\,W_1$ —. Montrons que b' vérifie les trois propriétés du lemme.

W s'écrit : $W=\bigoplus_{i=1}^{N_0+1} W_i(i)$ avec $W_i \in E$ et $b'=A\,.\,W_i$.

(i) $b' \notin S$:

Si $b' \in S$, alors (Prop. 3.1, 3.2) $W \in \mathrm{Sol}^{N_0+1}(A, b')$: contradiction. Donc $b' \notin S$.

(ii) $\exists\, V \in E$, $\langle \lambda_1, \ldots, \lambda_m \rangle \in (\mathbb{Q}^+)^m / A\,.\,V=b'$ *et* $\langle V, \lambda_1, \ldots, \lambda_m \rangle$ *est une solution non nulle de support minimal de l'équation* $A\,.\,X=\sum_1^m \lambda_j b_j$.

Par hypothèse S étant b_- générateur au rang N_0 on a :

$$\bigoplus_{i=1}^{N_0} W_i(i)=\sum_{j=1}^{m}\left[\sum_{V_1, \ldots, V_{N_0} \in \mathrm{Sol}(A, b_j)} \lambda_{V_1, \ldots, V_{N_0}} \cdot \bigoplus_{i=1}^{N_0} V_i(i)\right] + \sum_{V \in \mathrm{Sol}(A, 0)} \bigoplus_{i=1}^{N_0} \mu_V^i\, V(i)$$

Supposons $\exists\, \alpha_V^i \neq 0$.

On a $\mathrm{Supp}(V(i)) \subseteq \mathrm{Supp}(W)$ et $V(i) \in \mathrm{Sol}^{N_0+1}(A, 0)$. W étant une solution de support minimal alors $W = \mu V(i)$. Donc $b' \in S$ ce qui contredit la propriété (i).

Donc $\forall i, v : \mu_V^i = 0$. Autrement dit :

$$\forall i \leqq N_0, \quad W_i = \sum_{j=1}^{m} \sum_{V_1, \ldots, V_{N_0} \in \mathrm{Sol}(A, B_j)} \lambda_V 1, \ldots, \mathrm{V}_{N_0} V_i.$$

Par définition on a pour tout $i\, A \,.\, W_{N_0+1} = A \,.\, W_i = b'$, et pour tout j et tout V_i dans $\mathrm{Sol}(A, b_j)$, $A \,.\, V_i = b_j$, ce qui donne :

$$A \,.\, W_1 = \sum_{j=1}^{m} \Big(\sum_{V_1, \ldots, V_{N_0} \in \mathrm{Sol}(A, b_j)} \lambda_{V_1, \ldots, V_{N_0}} \Big) \,.\, b_j$$

On pose :

$$\lambda_j = \sum_{V_1, \ldots, V_{N_0} \in \mathrm{Sol}(A, b_j)} \lambda_{V_1, \ldots,} V_{N_0} \quad \Rightarrow \quad A \,.\, W_1 = \sum_{j=1}^{m} \lambda_j \,.\, b_j$$

Donc $\langle W_{N_0+1}, \lambda_j, \ldots, \lambda_j \rangle$ est solution de $A \,.\, X = \sum \lambda_j \,.\, b_j$ et $A \,.\, W_{N_0+1} = b'$. A ce point, si $\langle W_{N_0+1}, \lambda_j, \ldots, \lambda_j \rangle$ est une solution de support minimal alors la propriété (iii) est prouvée.

Supposons que $\langle W_{N_0+1}, \lambda_1, \ldots, \lambda_m \rangle$ *ne soit pas une solution de support minimal.*

Il existe donc $\langle X', \lambda'_j, \ldots, \lambda'_m \rangle$ solution dont le support est plus petit.

Posons pour $V_{1, \ldots,} V_{N_0} \in \mathrm{Sol}(A, b_j)$

$$\lambda'_{V_1, \ldots, V_{N_0}} = \begin{cases} \dfrac{\lambda'_j}{\lambda_j} \,.\, \lambda_{V_1, \ldots, V_{N_0}} & \text{si} \quad \lambda_j \neq 0 \\ 0 & \text{sinon} \end{cases}$$

et posons :

$$\forall i \leqq N_0, \quad W'_i = \sum_{j=1}^{m} \sum_{V_1, \ldots, V_{N_0} \in \mathrm{Sol}(A, b_j)} \lambda'_{V_1, \ldots, V_{N_0}} \,.\, V_i, \qquad \text{et} \qquad W'_{N_0+1} = X'$$

et enfin :

$$W' = \bigoplus_{1}^{N_0+1} W'_i(i)$$

W' est donc solution de l'équation (1) au rang (N_0+1) – $\forall i \leqq N_0$, $A . W_i = A . W_{N_0+1} = A . X' = \sum \lambda'_j$ – et, par construction, $\mathrm{Supp}(W') \subset \mathrm{Supp}(W)$; deux cas se présentent :

- W' est de support strictement minimal par rapport à chacun des vecteurs de F_{S, N_0+1} et alors W' appartient à M. Donc du fait de la minimalité de ces deux vecteurs, W et W' sont colinéaires.
- W' n'est pas de support strictement minimal par rapport à chacun des vecteurs de F_{S, N_0+1} et donc il existe V dans F_{S, N_0+1} tel que $\mathrm{Supp}(V) \subset \mathrm{Supp}(W')$ et il vient par transitivité $\mathrm{Supp}(V) \subset \mathrm{Supp}(W)$, ce qui contredit le fait que W soit dans M.

(iii) $\nexists\, b \in S / \mathrm{Sol}(A, b) < \mathrm{Sol}(A, b')$.

Supposons que cette propriété soit contredite par un vecteur b_0. D'après la définition de la relation $<$ il vient :

$$\forall i, \exists j, V_j \in \mathrm{Sol}(A, b_0)\ tq\ \mathrm{Supp}(V_j) \subseteq \mathrm{Supp}(W_i).$$

Posons $V = \bigoplus_{j=1}^{N_0+1} V'_j(j)$. V appartient donc à F_{S, N_0+1}.

Par construction $\mathrm{Supp}(V) \subseteq \mathrm{Supp}(W)$ ce qui contredit le fait que W soit dans M. □

Nous sommes maintenant en mesure de proposer un algorithme qui calcule une famille génératrice de (1). Le principe est le suivant : on commence par calculer l'ensemble S des vecteurs b pour $n=2$ – résolution de $A . X_1 = A . X_2$ – puis l'on enrichit cette famille à l'aide du lemme 3.1.

Algorithme 1

1) $\mathrm{Inc}\, S := \varnothing$
2) $T :=$ une plus petite famille génératrice de $A . X = A . Y$ # calculée par Farkas #
3) $S := \{ b = A . X / \langle X, Y \rangle \in T \}$ # $S = \{0, b_1, \ldots, b_k\}$ #
4) **FAIRE**
   4.1) $K := \mathrm{Card}(S \setminus \{0\})$ – nombre de b_i non nul dans S –
   4.2) Calculer une plus petite famille génératrice :
$$F = \left\{ (X, \lambda_1, \ldots, \lambda_1) \in E \times (\mathbb{Q}^+)^K / [A - b_1 \ldots - b_K] . \begin{bmatrix} X \\ \lambda_1 \\ \vdots \\ \lambda_K \end{bmatrix} = 0 \right\}$$
   4.3) $\mathrm{Inc}\, S := \{ b' = A . X / \langle X, \lambda_1, \lambda_2, \ldots, \lambda_K \rangle \in F$ *et* il n'existe aucun $b \in S$ *tel que* $\mathrm{Sol}(A, b) < \mathrm{Sol}(A, b') \}$
   4.4) $S := S \cup \mathrm{Inc}\, S$

TANT QUE $\mathrm{Inc}\, S \neq \varnothing$.

5) Le résultat est S # $\bigcup \mathrm{Sol}^n(A, b)$ pour $b \in S$ est la famille génératrice #

Fin

Preuve de l'algorithme :

Terminaison : Supposons que cet algorithme ne se termine pas, et prenons alors un vecteur b_i dans $\operatorname{Inc} S$ à chaque tour de boucle. On obtient alors un ensemble infini de vecteurs $\{b_i\}$ indexés par ordre d'insertion dans cet ensemble. Soit la famille $\{\operatorname{Sol}(A, b_i)\}$, l'instruction 4.3 implique que cette famille ne vérifie pas la proposition 3.1 d'où contradiction. $\square$

Correction : On va prouver par récurrence sur n que $\{\bigcup \operatorname{Sol}^n(A, b), b \in S\}$ est une famille génératrice du système $A.X_1 = A.X_2 = \ldots = A.X_n$ et ceci à l'aide de la caractérisation de génération; c'est-à-dire que l'on montre que pour toute solution V de l'équation $A.X_1 = \ldots = A.X_n$ il existe un vecteur W appartenant à $\{\bigcup \operatorname{Sol}^n(A, b)\}$ tel que le support de W soit inclus dans celui de V. On note $S = \{0, b_1, \ldots, b_m\}$.

Pour $n=2$: Soit $W = W_1(1) + W_2(2)$ une solution non nulle de support minimal de l'équation $A.X_1 = A.X_2$. Par construction (instruction 3), $A.W_1 \in S$.

Cas 1 $A.W_1 \neq 0$: $\exists V_1, \ V_2 \in \operatorname{Sol}(A, A.W_1)\ tq \operatorname{Supp}(V_1) \subseteq \operatorname{Supp}(W_1)$ et $\operatorname{Supp}(V_2) \subseteq \operatorname{Supp}(W_2)$.

D'où pour $V = V_1(1) + V_2(2) \in \operatorname{Sol}^2(A, A.W_1)$, on a $\operatorname{Supp}(V) \subseteq \operatorname{Supp}(W)$.

Cas 2 $A.W_1 = 0$: Soit $W_1 \neq 0$, soit $W_2 \neq 0$. Supposons $W_1 \neq 0$ (l'autre cas est symétrique)

$$\exists V_1 \in \operatorname{Sol}(A, 0) \quad \text{tel que } \operatorname{Supp}(V_1) \subset \operatorname{Supp}(W_1)$$

et donc pour $V = V_1(1) \in \operatorname{Sol}^2(A, 0)$, on a $\operatorname{Supp}(V) \subset \operatorname{Supp}(W)$.

Supposons la propriété vérifiée pour n : Soit W une solution non nulle de support minimal de l'équation $A.X_1 = A.X_2 = \ldots = A.X_{n+1}$:

Cette solution s'écrit : $W = \bigoplus_{1}^{n+1} W_i(i)$ avec $W_i \in E$.

Posons $b'=A.W_i$. Du fait de la lemme 3.1 et par hypothèse de récurrence, le vecteur b' appartient à l'ensemble S. Sol(A, b') étant une famille pseudo-génératrice on obtient alors :

$$\forall i, \exists V_i \in \mathrm{Sol}(A, b') \text{ tq } \mathrm{Supp}(V_i) \subset \mathrm{Supp}(W_i)$$

Posons $V = \bigoplus_{1}^{n+1} V_i(i)$, alors par construction V appartient à $\{\bigcup \mathrm{Sol}^n(A, b)\}$ et $\mathrm{Supp}(V) \subseteq \mathrm{Supp}(W)$.

Nous avons donc montré que $\bigcup \mathrm{Sol}^n(A, b)$ pour $b \in S$ vérifiait la caractérisation 2.1 dite de génération et que donc cette famille est génératrice. □

Remarque : Un argument d'algèbre linéaire – communication interne de J. L. Lambert non encore publiée – permet de prouver que l'algorithme 1 ne peut en fait effectuer plus de (*nombre de colonne de A*) tours de boucle.

A partir de la famille S construite par l'algorithme il est possible de trouver une plus petite famille génératrice.

PROPOSITION 3.3 : *Soient :*

$S=\{0, b_1, \ldots, b_m\}$ la famille calculée par l'algorithme ci-dessus.

$\mathrm{Sol}_k(A, b_j)=\{V \in \mathrm{Sol}(A, b_j)/\forall V' \in \mathrm{Sol}(A, b_k)$, $\mathrm{Supp}(V')$ n'est pas inclus dans $\mathrm{Supp}(V)\}$.

$\mathrm{Sol}_S^n(A, b_j)=\{\sum V_i(i) \in \mathrm{Sol}^n(A, b_j)/\forall k \neq j, \exists V_i \in \mathrm{Sol}_k(A, b_j)\}$.

$\mathrm{Sol}_S^n(A, 0)=\mathrm{Sol}^n(A, 0)$.

Alors, $\bigcup \mathrm{Sol}_S^n(A, b)$ pour $b \in S$ est une plus petite famille génératrice

Preuve : Nous montrons d'abord que $\bigcup \mathrm{Sol}_S^n(A, b)$ reste une famille génératrice. Pour cela il suffit de montrer que les vecteurs que nous avons retirés de $\bigcup \mathrm{Sol}^n(A, b)$ n'étaient pas de support minimal.

Soit $\sum V_i(i) \in \mathrm{Sol}^n(A, b_j) \setminus \mathrm{Sol}_S^n(A, b_j)$, alors $\exists k \neq j$ tel que $\forall V_i$, $\exists V_i'$ avec $\mathrm{Supp}(V_i' \subseteq \mathrm{Supp}(V_i)$ et par conséquent :

$$\sum V_i'(i) \in \mathrm{Sol}^n(A, b_k) \text{ et } \mathrm{Supp}(\sum V_i'(i)) \subseteq \mathrm{Supp}(\sum V_i(i)).$$

Pour terminer la preuve utilisant la caractérisation de minimalité, il reste à montrer que $\bigcup \mathrm{Sol}_S^n(A, b)$ ne contient que des vecteurs de supports incomparables :

Cas 1 : $V = V_i(i) \in \mathrm{Sol}_S^n(A, 0)$

– son support ne peut contenir le support d'un autre vecteur de $\mathrm{Sol}_S^n(A, 0)$ puisque déjà $\mathrm{Sol}^n(A, 0)$ est une famille de vecteurs de supports incomparables puisque construite par l'algorithme de Farkas.

– son support ne peut contenir le support d'un vecteur de $\mathrm{Sol}_S^n(A, b_j)$ puisque V est de support nul sur les E_k pour $k \neq i$

Cas 2 : $V = \sum V_i(i) \in \mathrm{Sol}_S^n(A, b_j)$

– son support ne peut contenir le support de $V'_k(k) \in \mathrm{Sol}_S^n(A, 0)$, sinon V_k ne serait pas une solution de support minimal de $A.X = b_j$ (en effet pour un λ bien choisi $V_k - \lambda.V'_k$ serait une solution de $A.X = b_j$ de support plus petit).

– son support ne peut contenir le support d'un vecteur de $\mathrm{Sol}_S^n(A, b_j)$ puisque tous les vecteurs de $\mathrm{Sol}(A, b_j)$ sont de supports incomparables.

– son support ne peut contenir le support d'un vecteur de $\mathrm{Sol}_S^n(A, b_k)$ pour $k \neq j$, par la définition de $\mathrm{Sol}_S^n(A, b_j)$. □

Remarque : Hormis le fait d'obtenir une famille génératrice paramétrée – pour tout n –, l'algorithme 1 se trouve être beaucoup plus efficace qu'un calcul direct pour un n donné. En effet, si K est la taille de la plus grande des familles $\mathrm{Sol}(A, b_i)\, b_i \neq 0$, alors la taille de la famille génératrice est d'ordre K^n – d'ordre n si $K = 0$ –, tandis que la taille de la représentation de la famille obtenue par l'algorithme 1 est inférieure à $k.K$ où k est le nombre de b_i calculés par cet algorithme.

4. APPLICATION À D'AUTRES FAMILLES DE SYSTÈMES D'ÉQUATIONS

4.1. Systèmes issus des réseaux à prédicats unaires

La matrice d'incidence A_n d'un réseau à prédicat unaire de paramètre n peut s'exprimer ainsi [Vau 84] :

$$A_n = \begin{bmatrix} A+H & 0 & \dots & 0 \\ A+H & A & \dots & 0 \\ \vdots & \dots & \ddots & \dots \\ A+H & 0 & \dots & A \end{bmatrix}$$

où A et H sont des matrices $P \times T$ et où A_n est une matrice $n \times n$ de blocs $P \times T$. Ici aussi les solutions sont dans E^n et en reprenant les notations du paragraphe 3 le système peut se réécrire :

$$(A+H).X_1=0 \qquad \text{et} \qquad \forall\, 2 \leqq i \leqq n\, H.X_1+A.X_i=0$$

d'où en retranchant la première équation des autres :

$$(A+H).X_1=0 \qquad \text{et} \qquad \forall\, 2 \leqq i \leqq n, \quad A.X_i-A.X_1=0$$

ce système étant bien sûr équivalent à :

$$(2) \qquad (A+H).X_1=0 \qquad \text{et} \qquad A.X_1=\ldots=A.X_n.$$

Écrit de cette manière la résolution de ce système peut s'obtenir en appliquant l'algorithme de la section précédente et en reportant les solutions dans la première équation; ce que fait l'algorithme suivant :

Algorithme 2

1) Calculer la famille S associée à la matrice A par l'algorithme 1, on note $S=\{b_0, \ldots, b_m\}$ avec $b_0=0$ et $\mathrm{Sol}(A, b_j)=\{V_{j,1}, \ldots, V_{j,nj}\}$

2) Résoudre par l'algorithme de Farkas :

$$\sum_{j=0}^{m} \sum_{V_{jk} \in \mathrm{Sol}(A, b_j)} \mu_{j,k}.(A+H).V_{j,k}=0$$

3) On note $S^*=\{\langle \mu_{j,k} \rangle\}$ la famille trouvée

Fin

PROPOSITION 4.1 : Soit F la famille composée des vecteurs de E^n :

- $\forall\, i \geqq 2$, $\forall\, V \in \mathrm{Sol}(A, 0)$, le vecteur $W=V(i)$
- $\forall \langle \mu_{j,k} \rangle \in S^*$, $\forall\, V_i^{j,k} \in \mathrm{Sol}(A, b_j)$, le vecteur W

$$W=\sum_{V_{0,k} \in \mathrm{Sol}(A, 0)} m_{0,k}.V_{0,k}(1) + \sum_{j=1}^{m} \left(\sum_{V_{j,k} \in \mathrm{Sol}(A, b_j)} m_{j,k}.\left(V_{j,k}(1) \oplus \bigoplus_{i=2}^{n} V_i^{j,k}(i) \right)\right)$$

Alors F est une famille génératrice du système (i)

Preuve :

- Montrons d'abord que tout vecteur de F vérifie (2) :

Cas 1. *$W=V(i)$ pour $i \geqq 2$ et $V \in \mathrm{Sol}(A, 0)$*

$$W_i=V \text{ et } \forall\, k \neq i,\, W_k=0$$

Donc $(A+H).W_1=(A+H).0=0$ et $\forall\, k \neq i$, $A.W_k=0=A.W_i$

Cas 2.

$$W = \sum_{V_{0,k} \in \mathrm{Sol}(A,0)} \mu_{0,k} \cdot V_{0,k}(1) + \sum_{j=1}^{m} \sum_{V_{j,k} \in \mathrm{Sol}(A,b_j)} \mu_{j,k} \cdot (V_{j,k}(1) \oplus \bigoplus_{i=2}^{n} V_i^{j,k}(i))$$

$$W_1 = \sum_{V_{0,k} \in \mathrm{Sol}(A,0)} \mu_{0,k} \cdot V_{0,k} + \sum_{j=1}^{m} \sum_{V_{j,k} \in \mathrm{Sol}(A,b_j)} \mu_{j,k} \cdot V_{j,k}$$

Donc par définition des $\mu_{j,k}$, $(A+H) . W_1 = 0$.

De même, $\forall i \neq 1$, $W_i = \sum_{j=1}^{m} \sum_{V_{j,k} \in \mathrm{Sol}(A,b_j)} \mu_{j,k} \cdot V_i^{j,k}$, avec pour tout $(i, j, k)\, A . V_i^{j,k} = b_j$

D'où, $\forall i$, $A . W_i = \left[\sum_{j=1}^{m} \sum_{V_{j,k} \in \mathrm{Sol}(A,b_j)} \mu_{j,k} \cdot b_j\right]$ est un vecteur indépendant de i.

• Montrons maintenant que pour toute solution de (2) il existe un vecteur de F de support plus petit ou égal.

Soit W une solution non nulle du système (2), puisque W vérifie l'équation $A . X_1 = \ldots = A . X_n$, W s'écrit :

$$W = \sum_{j=1}^{m} \sum_{V_1, \ldots, V_n \in \mathrm{Sol}(A,B_j)} \lambda_{V_1, \ldots, V_n} \sum_{i=1}^{n} V_i(i) + \sum_{V \in \mathrm{Sol}(A,0)} \sum_{i=1}^{n} \lambda_V^i \, V(i)$$

Cas 1. $\exists \lambda_V^i \neq 0$ avec $i \neq 1$

On a alors : $\mathrm{Supp}(V(i)) \subseteq \mathrm{Supp}(W)$ et $V(i) \in F$.

Cas 2. $\forall i \neq 1$, $\forall \nu$, $\lambda_V^i = 0$.

Autrement dit : $\forall i \neq 1$, $W_i = \sum_{j=1}^{m} \sum_{V_1, \ldots, V_n \in \mathrm{Sol}(A,b_j)} \lambda_{V_1, \ldots, V_n} V_i$

et :

$$W_1 = \sum_{j=1}^{m} \left[\sum_{V_{j,k} \in \mathrm{Sol}(A,b_j)} \left(\sum_{V_2, \ldots, V_n \in \mathrm{Sol}(A,b_j)} \lambda_{V_{j,k}, V_2, \ldots, V_n}\right) . V_{j,k}\right] + \sum_{V_{0,k} \in \mathrm{Sol}(A,0)} \lambda_{0,k}^1 V_{0,k}$$

Posons : $\forall j \neq 0$, $\lambda_{j,k} = \left(\sum_{V_2, \ldots, V_n \in \mathrm{Sol}(A,B_j)} \lambda_{V_{j,k}, V_2, \ldots, V_n}\right)$ et $\lambda_{0,k} = \lambda_{V_{0,k}}^1$.

Puisque $W_1 \neq 0$ et $(A+H).W_1=0$, $\langle \lambda_{j,k} \rangle$ vérifie S^* et il existe $\langle \mu_{j,k} \rangle$ solution minimale de S^* de support plus petit ou égal.

Si $\mu_{j,k} \neq 0$ alors $\lambda_{j,k} \neq 0$ et pour $j \neq 0$, si $\lambda_{j,k} \neq 0$ alors $\exists \lambda_{V_{j,k}, V_2^{j,k}, \ldots, V_n^{j,k}} \neq 0$.

Soit :

$$V = \sum_{V_{0,k} \in \mathrm{Sol}(A,0)} \mu_{0,k} . V_{0,k}(1) + \sum_{j=1}^{m} \sum_{V_{j,k} \in \mathrm{Sol}(A,b_j)} \mu_{j,k} . (V_{j,k}(1) + \bigoplus_{i=2}^{n} V_i^{j,k}(i))$$

Alors $V \in F$ et par construction $\mathrm{Supp}(V) \subseteq \mathrm{Supp}(W)$. □

Remarque : Si il existe $\langle \mu_{j,k} \rangle$ solution minimale de S^* avec un $\mu_{j,k} \neq 0$ pour un $j \neq 0$, la taille de la famille génératrice calculée par l'algorithme 2 en fonction de n est de l'ordre de M^n pour un certain M. Sinon cette taille est de l'ordre de n.

4.2. Systèmes issus des réseaux réguliers unaires

La matrice d'incidence A_n d'un réseau régulier unaire de paramètre n peut s'exprimer ainsi [Had 87] :

$$A_n = \begin{bmatrix} A+H & H & \ldots & H \\ H & A+H & \ldots & H \\ \vdots & \ldots & \ddots & \ldots \\ H & H & \ldots & A+H \end{bmatrix}$$

où A et H sont des matrices $P \times T$ et où A_n est une matrice $n \times n$ de blocs $P \times T$. Ici aussi les solutions sont dans E^n et en reprenant les notations du paragraphe 3 le système peut se réécrire :

$$(x) \forall i, \quad \sum H.X_j + A.X_i = 0.$$

Soit le système (3) :

$$(3) \qquad \begin{cases} (A+n.H).\sum_{i=1}^{n} X_i = 0 \\ A.X_1 = A.X_2 = \ldots = A.X_n \end{cases}$$

La première ligne de ce système est obtenue en sommant toutes les équations de (x), la seconde en retranchant la première équation de (x) à toutes les autres.

Le système (3), équivalent au système (x), se résout d'une façon analogue au système (2) c'est-à-dire que l'on résout la seconde équation du système $A.X_1 = \ldots = A.X_n$ par l'algorithme 1, puis l'on reporte les solutions sur la première équation $(A+n.H).\sum X_j = 0$.

Cette résolution faisant appel à des notions de calcul formel qui sortent du cadre de cet article nous ne donnons que l'idée de l'algorithme; néanmoins le lecteur pourra se reporter à [CHP 90].

Supposons que b_j soit un vecteur obtenu par l'algorithme 1 sur la matrice A et que $\mathrm{Sol}(A, b_j)$ soit égal à l'ensemble $\{V_{j,k}\}$. Chaque vecteur de $\mathrm{Sol}^n(A, b_j)$ est une solution de $A.X_1 = \ldots = A.X_n$ et peut s'exprimer comme le vecteur formel :

$$X = \sum_{c \in C_{j,k}} V_{j,k}(c) \qquad \text{avec} \quad \{C_{j,k}\} \text{ une partition de } [1, n] \text{ si } b_j \neq 0$$

$$X = V_{0,k}(i) \qquad \text{pour} \quad i \in [1, n] \quad \text{si} \quad b_j = 0.$$

Aussi le report de ces vecteurs sur la première équation nous donnera pour chaque $b_j \neq 0$ le vecteur formel :

$$W_j = \sum k_{j,k}(A+nH).V_{j,k} \qquad \text{avec} \quad k_{j,k} = |C_{j,k}| \quad \text{et} \quad \sum k_{j,k} = n$$

et pour $b_j = 0$, nous obtenons pour chaque $V_{0,k}$ dans $\mathrm{Sol}(A, 0)$ le vecteur formel :

$$W_0^k = (A+nH).V_{0,k}$$

Il nous faut donc résoudre le système :

$$\sum_{V_{0,k} \in \mathrm{Sol}(A,0)} \mu_k^0.W_k^0 + \sum_{b_j \in S \setminus \{0\}} \mu_j.W_j = 0$$

où $\{\mu_k^0, \mu_j\}$ sont les inconnues ce qui donne développé :

$$\sum_{V_{0,k} \in \mathrm{Sol}(A,0)} \mu_k^0.(A+nH).V_{0,k} + \sum_{b_j \in S \setminus \{0\}} \mu_j.\left[\sum_{k=1}^{m_j} k_{j,k}(A+nH).V_{j,k}\right] = 0$$

avec pour chaque $b_j \neq 0$ $\sum_{k=1}^{m_j} k_{j,k} = n$.

Ce système se simplifie en utilisant le fait que pour tout b_j et chaque vecteur $V_{j,k}$ de $\mathrm{Sol}(A, b_j)$, $A.V_{j,k}=b_j$. On obtient alors le système :

$$\sum_{V_{0,k}\in \mathrm{Sol}(A,0)} \mu_k^0.(0+nB.V_{0,k})+\sum_{b_j\in S\setminus\{0\}} \mu_j.\left[n.b_j+\sum_{j=1}^{m_j} n.k_{j,k}.B.V_{j,k}\right]=0$$

avec pour chaque $b_j\neq 0 \sum_{k=1}^{m_j} k_{j,k}=n$.

Ce dernier système est équivalent à l'équation :

$$(4)\qquad \sum_{V_{0,k}\in \mathrm{Sol}(A,0)} \mu_k^0.(B.V_{0,k})+\sum_{b_j\in S\setminus\{0\}} \mu_j.\left[b_j+\sum_{j=1}^{m_j} k_{j,k}.B.V_{j,k}\right]=0$$

avec pour chaque $b_j\neq 0 \sum_{k=1}^{m_j} k_{j,k}=n$.

La résolution du système (3) consiste donc à construire l'ensemble S correspondant à la matrice A, puis à résoudre de manière formelle l'équation (4).

5. CONCLUSION

Nous avons présenté trois algorithmes de résolution paramétrée de familles de systèmes d'équations linéaires à solutions positives. Ces résultats sont à notre connaissance les premiers de ce genre.

Le premier algorithme qui est la base des deux autres consiste en la résolution de $A.X_1=A.X_2=\ldots=A.X_n$ où n est le paramètre de la famille, A est une matrice et les X_i sont les vecteurs inconnus. L'idée générale de la résolution est que le système pour un n assez grand dépendant de A, fournit la forme générale d'une famille génératrice pour un n quelconque. Cet algorithme a été implémenté et se révèle être plus efficace que la résolution brutale de $A.X_1=A.X_2=\ldots=A.X_i$ pour un i donné dès que cette valeur i est plus grande que 3 ou 4 pour des matrices de taille inférieure à 50×50.

Les deux autres familles sont issues de modèles connus dans les extensions des réseaux de Petri – réseaux à prédicats unaires [Vau 84] et réseaux réguliers unaires [Had 86]. Les solutions de ces systèmes fournissent des invariants du réseau dont ils sont issus et permettent aussi de déduire des propriétés intéressantes telles que la finitude de l'ensemble d'accessibilité. Le deuxième algorithme résout un problème ouvert par G. Memmi dans sa thèse d'état et a également été implémenté avec succès. Le troisième algorithme a nécessité pour son implémentation l'emploi du logiciel de calcul formel MAPLE ([1]).

Une perspective de ce travail est bien évidemment la résolution paramétrée de systèmes plus complexes. Dans cette direction, il existe deux possibilités : d'une part des systèmes similaires mais avec plusieurs paramètres et d'autre part des systèmes de structure plus complexe mais avec encore un seul paramètre. Citons par exemple le système $A.X_i = B.X_{(i+1) \bmod n}$ qui dans le cas où $B = A$ revient au premier système étudié ici.

BIBLIOGRAPHIE

[Aba 64] J. Abadie, Méthode de Fourier et méthode duale pour les systèmes d'inéquations linéaires, Communication invitée au « Mathematical Programming Symposium », Londres 64, Note E.D.F. N. HR 5.759/3, 5 juin 1964.

[Bra 83] G. W. Brams, Réseaux de Petri. Théorie et pratique, *Masson*, Paris, 1983.

[CHP 90] J. M. Couvreur, S. Haddad et J. F. Peyre, Résolution paramétrée d'une famille de systèmes d'équations linéaires à solutions positives, Rapport IBP 90.38, Université Pierre-et-Marie-Curie, Paris, septembre 1990.

[Chv 83] V. Chvatal, *Linear Programming*, W. H. Freeman and Company, New York, 1983.

[Col 89] J. M. Colom et M. Silva, Improving the Linearly Based Characterization of P/T Nets, *Proceding of the 10th International Conference on Application and Theory of Petri Nets.* Bad Honnef Godesberg, juin 1989.

[Cou 90] J. M. Couvreur, The General Computation of Flows for Coloured Nets, *Proceding of the 11th International Conference on Application and Theory of Petri Nets*, Paris, juin 1990, p. 204-223.

[Far 02] J. Farkas, Theorie der einfachen Ungleichungen, In: *Journal für die reine und andgewandte Mathematik*, 1902, 124, p. 1-27.

[Fou 27] J. Fourier, *Histoire de l'Académie Royale des Sciences, Analyse des travaux pendant l'année* 1824, Partie Mathématique, Paris, 1827, p. xlvij-lv.

[Fou 90] J. Fourier, *Œuvres*, 1890, *II*, p. 317-328, Paris.

[Gen 81] H. J. Genrich et K. Lautenbach, System Modelling with High-Level Petri Nets, *Theoret. Comput. Sci.*, 1981, 13, p. 103-136.

[Had 86] S. Haddad et C. Girault, *Algebraic structure of flows of a regular net*, Oxford England, juin 1986, Advances in Petri nets, G. Rozenberg ed., *L.N.C.S.*, n° 266, G. Rozenberg éd., Springer-Verlag, 1987, p. 73-88.

([1]) MAPLE, Department of Computer Science, University of Waterloo, Ontario Canada N2L 3G1.

[Had 87] S. HADDAD, Une catégorie régulière de réseau de Petri de haut niveau : définition, propriétés et réductions. Application à la validation de systèmes distribués, *Thèse*, Université Pierre-et-Marie-Curie, Paris, juin 1987.

[Mem 83] G. MEMMI, Méthodes d'analyse de réseaux de Petri, réseaux à files et applications aux systèmes temps réel, *Thèse d'État*, Université Pierre-et-Marie-Curie, Paris, juin, 1983.

[Jen 86] K. JENSEN, *Coloured Petri nets. Petri nets: Central models and their properties.* Advances in Petri nets, G. ROZENBERG éd., *L.N.C.S.*, n° 254, Springer-Verlag Part. I. Bad Honnef, september 1986, p. 248-299.

[Sil 85] M. SILVA, J. MARTINEZ, P. LADET et H. ALLA, Generalized inverses and the calculation of symbolic invariants for coloured Petri nets, *Techn. Sci. Inform.*, 1985, *4*, 1, p. 113-126.

[Vau 84] J. VAUTHERIN et G. MEMMI, *Computation of flows for unary predicates transitions nets*, Advances in Petri nets, G. ROZENBERG éd., *L.N.C.S.*, n° 188, Springer-Verlag, 1984, p. 455-467.