

UNE NOUVELLE METHODE DE DECOMPOSITION
DES GRANDS SYSTEMES OU LA PARTITION
PRECEDE L'AFFECTATION

par A. Thuairé et J.P. Malengé

I.M.A.N. et LA 190

Université de Nice

06000-NICE FRANCE

I - INTRODUCTION

La modélisation des systèmes physiques, économiques, sociologiques conduit parfois à de grands systèmes de relations ($N > 100$) où le nombre de variables intervenant dans chaque relation est faible ($M \sim 10$). L'expérience montre que ces grands systèmes peuvent souvent être décomposés en une séquence ordonnée de sous-systèmes où seules les relations appartenant à un même sous-système doivent être résolues simultanément. Le but d'une méthode de décomposition est d'identifier les différents sous-systèmes et de déterminer leur ordre de résolution. Il est évident que la résolution du système décomposé est plus simple que celle du système initial.

La décomposition d'un grand système est liée à la structure de celui-ci et ne dépend pas de la forme des relations. On représente cette structure sous forme matricielle de la façon suivante :

- A chaque relation et à chaque variable sont respectivement associées une ligne et une colonne d'une matrice booléenne appelée matrice d'occurrence ou matrice structurale.
- Un élément a_{ij} de cette matrice est tel que $a_{ij} = 1$ si, et seulement si, la variable x_j intervient dans la relation f_i .

Soit par exemple le système de huit équations :

$$f_1(x_1, x_5) = 0$$

$$f_5(x_2, x_5, x_7) = 0$$

$$f_2(x_2, x_7, x_8) = 0$$

$$f_6(x_1, x_5, x_8) = 0$$

$$f_3(x_1, x_3, x_4, x_6) = 0$$

$$f_7(x_5, x_8) = 0$$

$$f_4(x_1, x_2, x_3, x_4, x_6) = 0$$

$$f_8(x_2, x_3, x_6, x_7, x_8) = 0$$

La matrice d'occurrence associée à ce système est la matrice booléenne A de la fig.1

	1	2	3	4	5	6	7	8
1	1				1			
2		1					1	1
3	1		1	1		1		
4	1	1	1	1		1		
5		1			1		1	
6	1				1			1
7					1		1	
8		1	1			1	1	1

fig.1. Matrice A

Il faut noter que la matrice d'occurrence d'un système donné est toujours définie à une permutation près des lignes et des colonnes. En permutant les lignes et les colonnes de la matrice précédente on peut par exemple obtenir la matrice B de la fig. 2

	5	8	1	7	2	3	4	6	
7	1	1							}
6	1	1	1					}	
1	1		1						
8		1		1	1				}
5		1		1	1			}	
8		1		1	1	1	1		
3			1		1	1	1		}
4			1		1	1	1		

fig.2. Matrice B

Il apparaît clairement sur cette seconde matrice que le système donné peut être décomposé en trois sous systèmes S_1 , S_2 et S_3 . On commencera donc, pour résoudre ce système, par résoudre les équations 7, 6 et 1 pour obtenir la valeur des variables 5, 8 et 1. On peut alors résoudre les équations 2 et 5 puis les équations 8, 3 et 4 et déterminer ainsi la valeur de toutes les variables du système.

Le but d'une méthode de décomposition est donc de trouver, parmi toutes les combinaisons possibles des lignes et des colonnes d'une matrice d'occurrence donnée au départ, une combinaison (il en existe plusieurs) qui représente cette matrice sous une forme triangulaire par blocs.

De nombreux algorithmes de décomposition ont déjà été publiés. La plupart d'entre eux font intervenir trois phases distinctes : affectation, partition et hiérarchisation, sur lesquelles nous reviendrons en détail au § II. La programmation de ces trois phases met en évidence les points suivants :

- La capacité en mémoire est proportionnelle à N^2
- Les temps d'exécution obtenus sont proportionnels au mieux à N^2

Ces algorithmes ne sont de ce fait utilisables que sur de gros ordinateurs.

Le but de cet article est de présenter une nouvelle méthode de décomposition diminuant notablement la place mémoire et le temps d'exécution et applicable sur de petits ordinateurs. On démontre comment, à partir d'une propriété remarquable des systèmes partitionnables, on peut ramener un problème de décomposition à un problème

de minimisation. On donne également les caractéristiques d'un algorithme qui, programmé en FORTRAN et en BASIC, a permis d'obtenir :

- un encombrement mémoire proportionnel à $N \times M$
- des temps d'exécution proportionnels à N

II - BIBLIOGRAPHIE

Les principaux algorithmes de décomposition connus sont basés sur le même principe : ramener le problème de la décomposition à un problème de recherche des composantes fortement connexes d'un graphe dirigé construit à partir du système donné. On y distingue pratiquement trois phases : affectation, partition et hiérarchisation.

II - 1. L'affectation

Cette première phase a pour but d'associer à chaque relation du système donné une variable intervenant dans cette relation. Deux variables différentes ne peuvent pas être associées à la même relation et vice-versa. On détermine ainsi un ensemble de couples variable-relation permettant de résoudre le système donné. Il est évident que la solution n'est en général pas unique. On démontre cependant que la décomposition finale ne dépend pas de la solution choisie. C'est dire en particulier que cette affectation pourra être modifiée au moment de la résolution proprement dite.

Parmi les algorithmes les plus efficaces pour ce type particulier de problème d'affectation on peut citer l'algorithme des chaînes alternées (K1) et l'algorithme de Steward (S1). Pour parvenir à une affectation totale ces deux algorithmes procèdent par itérations à partir d'une affectation partielle de base.

Un système affecté peut être représenté par une combinaison des lignes et des colonnes de la matrice d'occurrence où tous les éléments de la diagonale principale sont égaux à un. La fig.2 montrait une matrice affectée et partitionnée, la fig.3 montre la même matrice uniquement affectée.

	1	7	6	4	2	8	5	3
1	1							1
2		1			1	1		
3	1		1	1				1
4	1		1	1	1			1
5		1			1		1	
6	1					1	1	
7						1	1	
8		1	1	1	1			1

fig.3. Matrice d'occurrence affectée

II - 2. Le graphe dirigé associé - le partitionnement

Une affectation étant trouvée, on peut définir sur l'ensemble des relations du système la relation binaire suivante :

$$f_i \mathcal{R} f_j \text{ si la variable associée à } f_i \text{ apparaît dans } f_j$$

On associe ainsi au système donné un graphe dirigé $(\mathcal{F}, \mathcal{R})$, \mathcal{F} désignant l'ensemble des relations. On démontre alors que l'ensemble des sous-systèmes irréductibles du système donné coïncide avec l'ensemble des composantes fortement connexes du graphe $(\mathcal{F}, \mathcal{R})$.

Partitionner revient donc à rechercher l'ensemble des composantes fortement connexes d'un graphe. Parmi les algorithmes les plus efficaces on peut citer l'algorithme des puissances de matrices (H1), l'algorithme de Steward (S2) et l'algorithme de Billingsley (B1).

II - 3. La hiérarchisation

La hiérarchisation a pour but de déterminer l'ordre de résolution des différents sous-systèmes irréductibles de la partition. C'est une opération simple à réaliser :

- on construit un graphe $\mathcal{G}' = (\beta, \mathcal{R}')$ où β désigne l'ensemble des sous-systèmes irréductibles et \mathcal{R}' la relation binaire sur β :

$$B1 \mathcal{R}' B2 \iff \left\{ \begin{array}{l} \text{La variable associée à une des relations du sous-} \\ \text{système } B1 \text{ intervient dans une des relations du} \\ \text{sous-système } B2. \end{array} \right.$$

- on ordonne les sommets de ce graphe sans circuit sous forme d'arbre.

La matrice d'occurrence de la fig.4 représente le système donné au § I sous une forme partitionnée mais non hiérarchisée. La matrice de la fig.5 représente le même système sous une forme partitionnée et hiérarchisée.

	3	4	6	7	2	5	8	1
2				1	1		1	
5				1	1	1		
6						1	1	1
7						1	1	
1						1		1
8	1		1	1	1		1	
3	1	1	1					1
4	1	1	1		1			1

fig.4. Partition mais non hiérarchisation

	5	8	1	7	2	3	4	6
6	1	1	1					
7	1	1						
1	1		1					
2		1		1	1			
5	1			1	1			
8		1		1	1	1		1
3			1			1	1	1
4			1			1	1	1

fig.5. Partition et hiérarchisation

Pour simplifier l'exposé on appellera par la suite matrice partitionnée une combinaison des lignes et des colonnes de la matrice d'occurrence représentant le système sous une forme partitionnée et hiérarchisée.

II - 4. Critique des différentes méthodes

Les méthodes précédentes ont comme principal inconvénient de nécessiter beaucoup de place en mémoire : elles travaillent sur des matrices (matrice d'occurrence puis matrice associée au graphe (FR)) mémorisées sous forme de tableaux $N \times N$. De plus ces méthodes sont relativement lentes et leur temps d'exécution est proportionnel à N^2 (A1), ce qui est normal : les algorithmes d'affectation et de partition effectuent des recherches dans des matrices $N \times N$.

III - PRINCIPE DE NOTRE METHODE

Il nous a semblé que le défaut majeur des méthodes existantes était précisément de rechercher une affectation avant de partitionner, et cela pour trois raisons :

- 1- on a généralement intérêt à couper un problème en ses parties avant de le résoudre (D1)
- 2- parmi les $(N!)^2$ combinaisons possibles des lignes et des colonnes de la matrice d'occurrence, les matrices partitionnées sont beaucoup plus nombreuses que les matrices affectées.

3- affecter un sous-système irréductible est une opération simple : il s'agit par définition d'une composante fortement connexe et les algorithmes d'affectation convergent alors très vite.

Le principe de notre méthode est très simple : chercher d'abord la partition, affecter ensuite chacun des sous-systèmes. En fait d'après la remarque 3 le seul problème est de partitionner, l'affectation d'un sous-système étant très peu coûteuse. Le paragraphe suivant montre comment on peut, grâce à une propriété remarquable des matrices partitionnées, ramener ce problème à un problème de minimisation d'une quantité numérique $\mu(A)$ associée à chaque combinaison des lignes et des colonnes de la matrice d'occurrence.

IV - UNE PROPRIÉTÉ REMARQUABLE DES SYSTÈMES PARTITIONNABLES

Sur la matrice partitionnée de la fig.6 (qui est la même que celle de la fig.5) on constate que la ligne brisée qui délimite la zone de zéros de la partie supérieure droite n'est pas monotone décroissante.

	5	8	1	7	2	3	4	6
6	1	1	1					
7	1	1						
1	1		1					
2		1		1	1			
5	1			1	1			
8		1		1	1	1		1
3			1			1	1	1
4			1		1	1	1	1

fig.6.

Mais, par des permutations n'intervenant que sur les lignes et les colonnes relatives à un même sous-système, on peut rendre cette séparatrice monotone décroissante (fig.7)

	5	8	1	7	2	3	6	4
7	1	1						
6	1	1	1					
1	1		1					
2		1		1	1			
5	1			1	1			
8		1		1	1	1	1	
3			1			1	1	1
4			1		1	1	1	1

fig.7.

Soit μ le nombre de zéros du coin supérieur droit. Le but de ce paragraphe est de montrer qu'il suffit de maximiser μ , sur l'ensemble des matrices d'occurrence ayant la séparatrice monotone décroissante, pour obtenir une matrice partitionnée. Après avoir précisé les notations, nous énoncerons plus rigoureusement cette propriété et la démontrerons.

IV - 1. Notations

Pour faciliter leur identification les relations et les variables seront numérotées de 1 à N et désignées par ce seul numéro.

- F : ensemble des N relations du système donné
- E : ensemble des N variables intervenant dans ces relations
- A : matrice d'occurrence du système. Les relations et les variables sont placées respectivement en ligne et en colonne dans l'ordre 1,2,...,N
- S_A : ensemble des matrices X pouvant être obtenues par permutation des lignes et des colonnes de la matrice A
- L_x : vecteur permutation ligne d'un élément X de S_A
- C_x : vecteur permutation colonne d'un élément X de S_A
- \mathcal{D}_A : sous-ensemble de S_A constitué par les matrices triangulaires inférieures par blocs représentant le système partitionné et hiérarchisé
- $\overline{\mathcal{D}_A}$: complémentaire de \mathcal{D}_A par rapport à S_A
- \mathcal{E}_A : ensemble des matrices échelon (cf définition ci-dessous)
- x_{ij} : terme général d'une matrice X
- $\text{Inf } (i/x_{ij} \neq 0)$: valeur minimale de l'indice i pour laquelle le terme x_{ij} n'est pas nul
- $\text{Sup } (j/x_{ij} \neq 0)$: valeur maximale de l'indice j pour laquelle le terme x_{ij} n'est pas nul

IV - 2. Définition d'une matrice échelon ligne

On appelle matrice échelon ligne une matrice appartenant à S_A et vérifiant la propriété

$$\text{Sup } (j/x_{ij} \neq 0) \leq \text{Sup } (j/x_{i+1,j} \neq 0) \quad \forall i \in \{1, 2, \dots, N\}$$

En d'autres termes, si l'on considère chaque ligne de la matrice d'occurrence comme la représentation binaire d'un nombre (poids faible à gauche), une matrice échelon ligne est une matrice dont les lignes sont classées par ordre croissant de leur bit de poids le plus fort. Si on considère la matrice d'occurrence A de la fig.1 la matrice ci-dessous (fig.7) est une matrice échelon ligne de S_A :

	1	2	3	4	5	6	7	8
1	1				1			
3	1		1	1		1		
4	1	1	1	1		1		
5		1			1		1	
8			1				1	1
6	1				1			1
7					1			1
8		1	1			1	1	1

fig.7. Matrice échelon ligne

On notera \mathcal{L}_A le sous-ensemble de S_A constitué par les matrices échelon ligne et \mathcal{C}_L la transformation qui associe à un élément de S_A un élément de \mathcal{L}_A

Iv - 3. Définition d'une matrice échelon colonne

On appelle matrice échelon colonne une matrice appartenant à S_A et vérifiant la propriété :

$$\text{Inf } (i/x_{ij} \neq 0) \leq \text{Inf } (i/x_{i,j+1} \neq 0) \quad \forall j \in \{ 1, 2, \dots, N \}$$

En d'autres termes, si l'on considère chaque colonne de la matrice d'occurrence comme la représentation binaire d'un nombre (poids faible en bas), une matrice échelon colonne est une matrice dont les colonnes sont classées par ordre décroissant de leur bit de poids le plus fort. Reprenons la matrice d'occurrence A de la fig.1, la matrice de la fig.8 est une matrice échelon colonne appartenant à S_A :

	1	5	8	7	8	3	4	6
1	1	1						
2			1	1	1			
3	1					1	1	1
4	1	1				1	1	1
5		1	1	1				
6	1	1			1			
7		1			1			
8			1	1	1	1	1	1

fig.8. Matrice échelon colonne

On notera \mathcal{E}_A le sous-ensemble de S_A constitué des matrices échelon colonne et φ_c la transformation qui permet d'associer à un élément de S_A un élément de \mathcal{E}_A .

IV - 4. Définition d'une matrice échelon

On appelle matrice échelon une matrice appartenant à l'ensemble $\mathcal{E}_A \cap \mathcal{L}_A$. On notera \mathcal{E}_A l'ensemble des matrices échelon. On peut remarquer que tout élément de \mathcal{E}_A possède une séparatrice de la zone de zéros du coin supérieur droit monotone décroissante. Si l'on considère à nouveau le système donné au § I, la matrice de la fig.9 est une matrice échelon de \mathcal{E}_A

		3	6	1	4	5	8	2	7
3	1	1	1	1					
1			1		1				
6			1		1	1			
7					1	1			
4	1	1	1	1				1	
2							1	1	1
5						1		1	1
8	1	1					1	1	1

fig.9. Matrice échelon

IV - 5. Enoncé et démonstration de la propriété

Nous allons démontrer que la matrice échelon qui a le plus grand nombre de zéros dans le coin supérieur droit est partitionnée (si elle est partitionnable).

On pose par définition, pour tous les éléments X appartenant à S_A

$$\mu(X) = \sum_{i=1}^N \sup (j/x_{ij} \neq 0)$$

On note X^* l'élément de S_A tel que :

$$\mu(X^*) = \inf_{X \in \mathcal{E}_A} \mu(X)$$

IV - 5. 1. Enoncé de la propriété P

Si A est la matrice d'occurrence d'un système partitionnable $X^* \in \mathcal{E}_A \cap \mathcal{D}_A$

IV - 5. 2. Démonstration de la propriété P

Supposons vraie la proposition suivante Q que l'on démontrera au § V :

$$\forall X \in \mathcal{E}_A \cap \overline{\mathcal{D}}_A \quad \exists Z \in \mathcal{E}_A \cap \mathcal{D}_A \quad \text{tel que } \mu(Z) < \mu(X)$$

La démonstration de la propriété P est alors très simple : il suffit de choisir

$X_0 \in \mathcal{E}_A \cap \overline{\mathcal{D}}_A$ tel que :

$$\mu(X_0) = \inf_{X \in \mathcal{E}_A \cap \overline{\mathcal{D}}_A} \mu(X)$$

Il nous reste donc, pour que la propriété P soit démontrée, à démontrer la proposition Q.

V - DEMONSTRATION DE LA PROPOSITION Q

Considérons un système F de N relations et de N variables partitionnable en k sous-systèmes irréductibles F_1, F_2, \dots, F_k . Le sous-système F_1 est le premier de la hiérarchie, F_k est le dernier.

On pose $\mathcal{F} = \bigcup_{i=1}^{k-1} F_i$ et on note \mathcal{U} et \mathcal{V} les sous-systèmes de variables correspondant respectivement à \mathcal{F} et F_k .

Le principe de la démonstration est le suivant :

- on se donne une matrice échelon quelconque X associée au système F
- on suppose que X n'est pas partitionnée mais que l'on connaît les numéros des équations et des variables qui forment les différents sous-systèmes.
- on construit (§ V.1) une matrice X'' partitionnée en deux sous-systèmes \mathcal{F} et F_k et telle que (§ V.2) $\mu(X'') < \mu(X)$
- on itère sur le sous-système réductible \mathcal{F} et on aboutit de proche en proche à une matrice Y complètement partitionnée (§ V.3) avec bien sûr, $\mu(Y) < \mu(X)$

- on montre pour terminer qu'une transformation ne comportant que des permutations à l'intérieur de chaque sous-système permet, à partir de Y , d'aboutir à une matrice échelon Z partitionnée et telle que $\mu(Z) \leq \mu(Y)$.

Il peut paraître choquant de supposer au début, que l'on connaît les numéros des variables et des équations de chaque sous-système, donc la partition. En fait il ne faut pas oublier que ce paragraphe a pour but de démontrer une propriété des systèmes partitionnables et non pas de démontrer la convergence d'un algorithme.

V - 1. Construction de la matrice X''

La matrice X'' est l'image de la matrice X par la composée $f_2 \circ f_1$ de deux transformations décrites ci-dessous.

V - 1.1. Définition de la transformation f_1

$$\begin{array}{ccc} \mathcal{E}_A \cap \overline{\mathcal{D}}_A & \xrightarrow{f_1} & S_A \\ X & \xrightarrow{f_1} & f_1(X) = X' \end{array}$$

La matrice image X' est définie à l'aide des vecteurs $L_{X'}$ et $C_{X'}$, de la façon suivante :

- a) Le vecteur ligne $L_{X'}$, est obtenu en plaçant en tête les numéros des relations appartenant à \mathcal{F} , en respectant l'ordre qu'elles avaient les unes par rapport aux autres dans la matrice X . On place ensuite les relations appartenant à F_k .
- b) On ne modifie pas le vecteur colonne : $C_{X'} = C_X$

Reprenons à titre d'exemple la matrice échelon de la fig.9. Son image par l'application f_1 est la matrice de la fig.10.

		3	6	1	4	5	8	8	7	
$L_{X'}$	1			1		1				
	6		1		1	1				
	7			1	1					
	2					1	1	1		
	5				1		1	1		
	3	1	1	1	1					
	4	1	1	1	1			1		
	8	1	1				1	1	1	

fig.10. Matrice X'

V - 1.2. Définition de la transformation f_2

$$f_1 (\mathcal{E}_A \cap \bar{\mathcal{O}}_A) \xrightarrow{f_2} S_A$$

$$X' \xrightarrow{\quad} f_2(X') = X''$$

La matrice image X'' représente le système sous une forme partitionnée et hiérarchisée en deux sous-systèmes \mathcal{F} et F_k . Elle est définie à l'aide des vecteurs $L_{X''}$ et $C_{X''}$ de la façon suivante :

- a) le vecteur colonne $C_{X''}$ est obtenu en plaçant en tête les numéros des variables appartenant à \mathcal{V} , en respectant l'ordre qu'elles avaient les unes par rapport aux autres dans la matrice X' . On place ensuite les variables appartenant à V .
- b) on ne modifie pas le vecteur ligne : $L_{X''} = L_{X'}$

La matrice X'' de la figure 11 est l'image par f_2 de la matrice X' de la figure 10.

		1	5	8	2	7	3	6	4	
1	1	1								}
6	1	1	1							
7		1	1							
2			1	1	1					
5		1		1	1					
3	1					1	1	1		}
4	1			1		1	1	1		
8			1	1	1	1	1			

$\underbrace{\hspace{10em}}_{\mathcal{V}}$

$\underbrace{\hspace{10em}}_{V}$

fig.11 Matrice X''

V - 2. Comparaison de $\mu(X'')$ et de $\mu(X)$

Le vecteur colonne de X n'est pas modifié par l'application f_1 . On peut donc écrire que $\mu(X') = \mu(X)$. Par contre l'application f_2 fait subir deux types de variations au nombre μ :

- une diminution apportée par les relations appartenant à \mathcal{F}
- une augmentation apportée par les relations appartenant à F_k

Nous calculons ci-dessous ces deux variations.

V - 2.1. La diminution

On pose :

$$p = \text{Card } \mathcal{F} = \text{Card } \mathcal{V}$$

$q_{X'}(i)$: position de la variable numéro i dans $C_{X'}$,

$q_{X''}(i)$: position de la variable numéro i dans $C_{X''}$

$$T = \left\{ i \in \mathcal{V} / q_{X'}(i) \neq q_{X''}(i) \right\}$$

A l'issue de la première transformation, les p relations appartenant à \mathcal{F} occupent les p premières lignes de la matrice X' . Par suite les p premiers éléments x'_{ij} des colonnes associées aux variables de \mathcal{V} sont nuls. En d'autres termes :

$$(1) \quad \text{Inf } (i / x'_{i\ell} \neq 0) > p \quad \forall \ell \in \{1, 2, \dots, N\} / C_{X'}(\ell) \in T$$

Au cours de la deuxième transformation le déplacement de chaque élément de T produit, d'après (1), une diminution d'une unité du nombre μ pour chacune des p premières lignes. La diminution totale sera par conséquent de $p \times \text{Card } T$.

V - 2.2. L'augmentation

Il n'est pas possible de faire un calcul exact de l'augmentation totale, celle-ci variant d'un système à un autre. On peut néanmoins en déterminer un majorant. Pour cela on pose :

$$T' = \left\{ i \in \{1, 2, \dots, N\} / L_{X'}(i) \in F_k \text{ et } \text{Sup } (j / x'_{ij} \neq 0) < \text{Sup } (j / x''_{ij} \neq 0) \right\}$$

Pour une ligne donnée correspondant à un élément de T' , l'augmentation ne peut dépasser p . En effet, l'ordre qu'avaient les éléments de \mathcal{V} dans $C_{X'}$, les uns par rapport aux autres a été respecté au cours de la seconde transformation et le contraire contredirait alors l'hypothèse $\text{Card } \mathcal{F} = p$. On peut donc en déduire que, dans le cas le plus défavorable où tous les éléments de T' subissent une augmentation égale à p , l'augmentation totale est égale à $p \times \text{Card } T'$.

V - 2.3. Démonstration de la relation $\mu(X'') \leq \mu(X)$

D'après V-2.1. et V-2.2. on a :

$$\mu(X'') \leq \mu(X') - p \times \text{Card } T + p \times \text{Card } T'$$

Nous allons maintenant démontrer que, dans l'hypothèse où chaque élément de T' subit une augmentation de μ égale à p , on a, quel que soit le système, $\text{Card } T' < \text{Card } T$. Cette démonstration tient dans les deux remarques suivantes :

Remarque 1 : Tout élément m appartenant à T' vérifie après la seconde transformation l'inégalité ci-dessous :

$$\text{Sup } (j / x''_{mj} \neq 0) \leq p + \text{Card } T$$

En d'autres termes aucune variable appartenant à $\bigcup_V T'$ n'intervient dans l'ensemble des relations correspondant à T'

Remarque 2 : $\forall k \in T'$ et $\forall l \notin T' / l > p$ on a $k < l$

En d'autres termes les relations correspondant à T' sont placées, à l'issue de la première transformation, en tête du sous-système F_k . Cela provient du fait que l'ordre qu'avaient les éléments de F_k les uns par rapport aux autres dans L_x a été respecté dans $L_{x'}$.

On déduit de ces deux remarques et de l'inégalité (1) que, dans l'hypothèse où $\text{Card } T = \text{Card } T'$, F_k est partitionnable en deux sous-systèmes $F_k^{(1)}$ et $F_k^{(2)}$, $F_k^{(1)}$ désignant le sous-ensemble de relations correspondant à T' et $F_k^{(2)}$ son complémentaire par rapport à F_k . Ceci contredit bien entendu l'hypothèse de départ : " F_k sous-système irréductible" et on a donc dans tous les cas $\text{Card } T' \neq \text{Card } T$.

Il est évident par ailleurs que le cas $\text{Card } T' > \text{Card } T$ conduit, pour des raisons analogues, à $F_k^{(1)}$ sous-système indéterminé. On a donc dans tous les cas :

$$\mu(X'') < \mu(X).$$

V - 3. Construction de la matrice Y

Les transformations f_1 et f_2 que nous venons d'appliquer sur le système F peuvent être appliquées successivement sur les sous-systèmes : $\mathcal{F} = \bigcup_{i=1}^{k-1} F_i$,

$$\bigcup_{i=1}^{k-2} F_i, \quad \bigcup_{i=1}^{k-3} F_i, \quad \dots, \quad F_1 \cup F_2.$$

On obtient ainsi de proche en proche une matrice Y telle que :

$$Y \in \mathcal{O}_A \quad \text{et} \quad \mu(Y) < \mu(X)$$

V - 4. Construction de la matrice Z

Dans certains cas la matrice Y, qui appartient à \mathcal{D}_A et à \mathcal{E}_A , n'appartient pas à \mathcal{L}_A (fig.11). Nous allons démontrer que, dans tous les cas, on peut associer à la matrice Y une matrice Z appartenant à $\mathcal{E}_A \cap \mathcal{D}_A$ et telle que $\mu(Z) \leq \mu(Y)$. Cette démonstration achèvera la démonstration de la proposition Q et par conséquent (§ IV-5.) celle de la propriété P.

L'ensemble \mathcal{E}_A a été défini au (§ IV-4.) comme l'intersection des ensembles \mathcal{L}_A et \mathcal{E}_A . Rappelons (§ IV-2.) que la transformation φ_L , qui permet d'associer à un élément de S_A un élément unique de \mathcal{L}_A , est obtenue en classant par ordre croissant les lignes de la matrice suivant leur bit de poids le plus fort (poids faible à gauche) et que la transformation φ_C , qui permet d'associer à un élément de S_A un élément unique de \mathcal{E}_A , est obtenue en classant par ordre décroissant les colonnes de cette matrice suivant leur bit de poids le plus fort (poids faible en bas).

La matrice Y appartient à \mathcal{E}_A mais la matrice $\varphi_L(Y)$ n'appartient pas forcément à $\mathcal{E}_A \cap \mathcal{L}_A$ c'est à dire à \mathcal{E}_A . En effet, dans certains cas, un classement des lignes provoque un déclassement des colonnes. Nous allons démontrer qu'une transformation φ_C suivant cette transformation φ_L sur Y permet d'obtenir une matrice Z élément de \mathcal{E}_A (§ V-4.1.). Nous démontrerons ensuite (§ V-4.2.) que $\mu(\varphi_C \circ \varphi_L(Y))$ est inférieur ou égal à $\mu(Y)$

V - 4.1. La matrice $\varphi_C \circ \varphi_L(Y)$

On pose par définition :

$$U = \{k \in \{1, 2, \dots, N\} / \exists j \in \{1, 2, \dots, N\} \text{ tel que } \text{Sup}(j/y_{ij} \neq 0) = k\}$$

La démonstration repose sur les deux remarques suivantes :

Remarque 1 : D'après la définition de φ_L , les colonnes appartenant à U restent classées les unes par rapport aux autres dans la matrice $\varphi_L(Y)$

Remarque 2 : L'ordre des colonnes de U les unes par rapport aux autres sera encore conservé par une transformation φ_C suivant φ_L puisque, de par sa définition, φ_C ne fait qu'intercaler entre les lignes classées les lignes non classées.

On peut déduire de ces deux remarques que la transformation φ_C ne modifie pas le classement des lignes. Comme d'autre part les transformations φ_C et φ_L peuvent être faites indépendamment sur chaque bloc de la matrice Y, et ne modifient

donc pas la répartition des équations et des variables de chaque sous-système, la matrice $Z = \mathcal{Q}_C \circ \mathcal{Q}_L(Y)$ appartient à la fois à \mathcal{E}_A et à \mathcal{D}_A donc à $\mathcal{E}_A \cap \mathcal{D}_A$.

V - 4.2. Comparaison de $\mu(\mathcal{Q}_C \circ \mathcal{Q}_L(Y))$ et de $\mu(Y)$

La transformation \mathcal{Q}_L ne modifie pas l'ordre des colonnes d'une matrice. Par conséquent, quelque soit la matrice Y , $\mu(Y) = \mu(\mathcal{Q}_L(Y))$. Pour démontrer l'inégalité :

$$\mu(\mathcal{Q}_C \circ \mathcal{Q}_L(Y)) \leq \mu(Y)$$

il suffit donc de démontrer l'inégalité :

$$\mu(\mathcal{Q}_C \circ \mathcal{Q}_L(Y)) \leq \mu(\mathcal{Q}_L(Y))$$

Nous allons voir que cette seconde inégalité découle de la définition de \mathcal{Q}_C

Au cours de la transformation \mathcal{Q}_C , toute colonne que l'on déplace fait subir au nombre μ deux types de variations :

- une augmentation d'une unité de μ pour la ligne contenant le premier élément non nul de la colonne déplacée.
- Une diminution d'une unité pour toutes les lignes comprises entre la position, dans $\mathcal{Q}_L(Y)$, du premier élément non nul de la colonne déplacée et la position de cet élément après \mathcal{Q}_C . Il est évident que, pour chaque colonne déplacée, il y a au moins une ligne concernée puisque, dans le cas contraire, la colonne n'aurait pas été déplacée.

On déduit de ce qui précède que l'augmentation totale du nombre μ sur $\mathcal{Q}_C \circ \mathcal{Q}_L(Y)$ est toujours inférieure ou égale à la diminution totale et donc que l'inégalité : $\mu(\mathcal{Q}_C \circ \mathcal{Q}_L(Y)) \leq \mu(\mathcal{Q}_L(Y))$, est vérifiée quelque soit la matrice Y (obtenue après les transformations f_1 et f_2).

VI - MISE EN OEUVRE DE LA METHODE

L'utilisation de la propriété P pour le partitionnement d'un grand système consiste donc à trouver un algorithme permettant de construire une suite de matrices échelon dont la valeur de μ est décroissante. Si cet algorithme converge, la matrice finale sera partitionnée. En fait il n'est pas nécessaire d'atteindre le minimum de μ : il existe plusieurs matrices échelon partitionnées qui n'ont pas la

même valeur de μ (fig.12).

	1	8	5	2	7	3	4	6
6	1	1	1					
1	1		1					
7		1	1					
2		1		1	1			
5				1	1			
3	1					1	1	1
4	1			1		1	1	1
8		1	1	1	1	1		

$$\mu = 43$$

	1	5	8	2	7	3	6	4
1	1	1						
6	1	1	1					
7		1	1					
8				1	1			
5		1		1	1			
8			1	1	1	1	1	
3	1					1	1	1
4	1		1			1	1	1

$$\mu = 41$$

fig.12.

L'algorithme que nous avons écrit possède les trois caractéristiques suivantes :

- il fonctionne par dichotomie : il cherche à partitionner la matrice donnée en deux blocs puis il itère dans chacun des blocs
- il nécessite peu de place en mémoire (cf. VI-1.)
- sa convergence n'est pas prouvée théoriquement mais pratiquement

VI - 1. Mémorisation de la matrice d'occurrence

Contrairement à certaines méthodes existantes (puissances de matrices par exemple), la méthode décrite ici ne tirerait aucun profit d'une matrice d'occurrence mémorisée sous forme d'un tableau $N \times N$. C'est pourquoi, afin d'occuper une place mémoire aussi petite que possible, la matrice d'occurrence est mémorisée sous forme d'un tableau $N \times P$ construit de la façon suivante :

- chaque ligne représente une relation et contient les numéros des variables intervenant dans cette relation.
- le nombre P de colonnes est le nombre maximum de variables intervenant dans une relation du système donné.

VI - 2. Description sommaire de l'algorithme

L'algorithme sera décrit en détail dans un autre article. On se contente ici d'en donner les grandes lignes.

La partie principale de l'algorithme est celle qui, étant donnée une matrice échelon X_i , permet d'en obtenir une autre X_{i+1} telle que $\mu(X_{i+1}) \leq \mu(X_i)$, avec arrêt en cas d'égalité stricte. On utilise pour cela essentiellement cinq transformations $\varphi_L, \varphi_C, \varphi_S, \varphi_F$ et γ_S . Les transformations φ_L et φ_C ont été respectivement décrites au IV - 2. et IV - 3. Les transformations φ_S, φ_F et γ_S sont décrites ci-dessous.

VI - 2.1. Définition de φ_S

$$\begin{array}{ccc} S_A & \longrightarrow & S_A \\ X & \longmapsto & \varphi_S(X) \end{array}$$

La matrice image $\varphi_S(X)$ est telle que :

- $C_{\varphi_S(X)} = C_{\varphi_C(X)}$
- $L_{\varphi_S(X)} = L_{\varphi_L(X)}$

VI - 2.2. Définition de φ_F

$$\begin{array}{ccc} S_A & \longrightarrow & S_A \\ X & \longmapsto & \varphi_F(X) \end{array}$$

La matrice image $\varphi_F(X)$ est telle que

- $C_{\varphi_F(X)} = C_{\varphi_S \circ \varphi_C}$
- $L_{\varphi_F(X)} = L_{\varphi_L(X)}$

VI - 2.3. Définition de γ_S

$$\begin{array}{ccc} E_A & \longrightarrow & S_A \\ X & \longmapsto & \gamma_S(X) \end{array}$$

La matrice $\gamma_S(X)$ est définie par :

- $C\gamma_S(X)$: permutation colonne qui minimise la somme des écarts entre les deux derniers éléments non nuls de chaque ligne de X

- $L\gamma_S(X)$: permutation ligne qui minimise la somme des écarts entre les deux premiers éléments non nuls de chaque colonne de X .

La figure 13 montre un exemple d'application de γ_S .

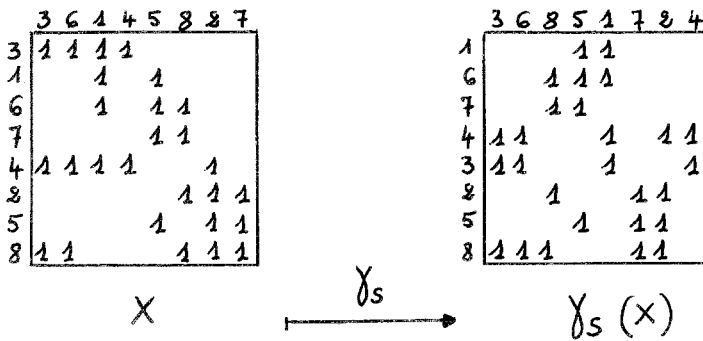


fig.13.

VI - 2.4. Construction de X_{i+1} à partir de X_i

Etant donné une matrice X_0 appartenant à S_A , on pose :

$$Y_{X_0} = \begin{cases} \varphi_c \circ \varphi_L(X_0) & \text{si } \mu(\varphi_c \circ \varphi_L(X_0)) < \mu(\varphi_L \circ \varphi_c(X_0)) \\ \varphi_L \circ \varphi_c(X_0) & \text{sinon} \end{cases}$$

X_{i+1} est alors définie par les formules

(1) X_0 donnée au départ

(2) $X_1 = Y_{\varphi_F}(X_0)$

(3) $X_{i+1} = Y_{\varphi_F}(\gamma_S(X_i))$

VI - 3. Les résultats

L'algorithme que nous venons de décrire très sommairement a été testé sur des systèmes partitionnables en deux sous-systèmes irréductibles, l'un des principes de la méthode étant de travailler par dichotomie. Les calculs ont été effectués sur un ordinateur T1600 Télémécanique de 16 k mots de 16 bits et ont porté sur plus de cent cinquante systèmes ($N=10$ à 50) partitionnables, générés aléatoirement. Dans tous les cas l'algorithme a partitionné rapidement le système donné. Les résultats détaillés seront publiés en même temps que l'algorithme.

VII - CONCLUSION

L'idée de base consistant à partitionner avant d'affecter nous a permis de trouver un algorithme qui s'est révélé efficace (rapidité et encombrement). Il est cependant probable que cet algorithme est loin d'être le plus performant et on peut penser que la poursuite des recherches dans cette voie sera fructueuse.

REFERENCES

- B1 Billingsley, D.S., Chem. Eng. Sci. 22 (4), 719 (1967) (Eng)
- H1 Himmelblau, D.M., Chem. Eng. Sci. 21 (5), 425-38 (1966) (Eng)
- H2 Himmelblau, D.M., and Bischoff, K.B., "Process Analysis and Simulation", Wiley, New York (1961)
- D1 Descartes, "Discours de la méthode" (1637)
- K1 Kaufmann, "Introduction à la combinatoire en vue de ses applications", DUNOD (1968)
- K2 Kevorkian, A.K., and Snoek J., "Decomposition in large scale systems", NATO Advanced Study Institute (1972)
- L1 Ledet, W.P., Ph. D. Thesis Univ. of Texas (1968)
- S1 Steward, D.V., SIAM Rev. 4,321 (1962)
- S2 Steward, D.V., J. SIAM Numer. Anal. Ser B, Vol 2, 2 (1965)