

LASZLO N. KISS

JEAN M. MARTEL

Hierarchisation d'entités à partir de comparaisons binaires assistée par un système interactif

RAIRO. Recherche opérationnelle, tome 25, n° 2 (1991),
p. 129-159

http://www.numdam.org/item?id=RO_1991__25_2_129_0

© AFCET, 1991, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>



HIÉRARCHISATION D'ENTITÉS A PARTIR DE COMPARAISONS BINAIRES ASSISTÉE PAR UN SYSTÈME INTERACTIF (*)

par Laszlo N. KISS ⁽¹⁾ et Jean M. MARTEL ⁽²⁾

Résumé. — *Dans la pratique de la gestion, on a de plus en plus tendance à vouloir prendre en considération des aspects socio-économique, technique, politiques et des objectifs multiples souvent conflictuels. Le manager est ainsi confronté à des problèmes passablement complexes qu'il cherche à structurer en établissant une certaine hiérarchisation dans tous ces éléments.*

C'est précisément pour l'assister dans cette phase de structuration que nous avons développé quelques résultats issus de l'analyse combinatoire et mis au point des algorithmes itératifs et interactifs, conviviaux avec présentation visuelle.

Mots clés : Macroentités; décomposition hiérarchique; système relationnel d'ordre; graphe partiellement et totalement saturé; relations binaires; système interactif.

Abstract. — *There is a growing tendency in the real world to take account of socio-economical, technical, political aspects and often conflictual objectives when making decisions. The manager can thus be confronted with complex problems that he would try to structure through the development of a hierarchy among all these elements.*

The focus of this paper is to help the manager establish such a structure among the conflicting objectives. Results emanating from combinatory analysis are presented together with iterative and interactive convivial algorithms and visual support.

Keywords : Macroentities; hierarchical decomposition; order relational system; completely and partially saturated graph; binary relations; interactive system.

INTRODUCTION

Dans l'analyse des problèmes de gestion (autant dans la gestion publique que privée), on a de plus en plus tendance à vouloir prendre simultanément en considération les aspects économique, humain, technique, politique, etc. On se trouve dès lors confronté à une multiplicité d'objectifs, de contraintes, de revendications, etc. L'hypothèse simplificatrice selon laquelle l'unique

(*) Reçu octobre 1989.

(¹) Centre National de Management, Budapest, Hongrie et Faculté des Sciences de l'Administration de l'Université Laval, Québec, Canada G1K 7P4.

(²) Faculté des Sciences de l'Administration de l'Université Laval, Québec, Canada G1K 7P4.

objectif d'un dirigeant est la maximisation du profit ne semble pas très réaliste. Bien que le profit à long terme soit toujours important pour les entreprises, ces dernières poursuivent le plus souvent tout un ensemble d'objectifs pas toujours très bien explicités, généralement conflictuels et même parfois ambigus.

Les situations sont très nombreuses où un manager doit établir une hiérarchisation (un ordre de priorité) parmi les éléments d'un ensemble parfois assez grand d'entités. Par exemple, en analyse multicritère, que se soit avec la méthode de la fonction d'utilité multiattribut [Keeney and Raiffa, 1982], avec une méthode basée sur une relation de surclassement [Roy, 1968, 1972, 1978], avec une méthode lexicographique [Fishburn, 1974], etc., on est amené à hiérarchiser des objectifs et (ou) à établir une relation d'ordre entre des critères, des attributs. D'ailleurs, il existe présentement un courant de recherche fort important où l'on part de relations d'ordre d'importance entre des sous-ensembles de critères, soit pour expliciter la valeur des coefficients de pondération de ces critères ou soit pour, tout au moins, établir des intervalles de variation pour les valeurs de ces coefficients [Vansnick, 1986; Sólymosi et Dombi, 1986; Pasche, 1987; Roy *et al.*, 1986].

Face à un ensemble de critères construit pour éclairer une décision, il est exceptionnellement rare qu'ils revêtent tous la même importance aux yeux du décideur, ou bien que l'importance de chacun soit totalement différente. Fréquemment, à partir de jugements révélés par un décideur sur des paires de critères, on cherche à construire une hiérarchisation de ces derniers selon leur ordre d'importance. D'ailleurs, certaines méthodes d'analyse multicritère exigent uniquement une telle relation d'ordre concernant l'importance des critères [Roubens, 1980; Paelinck, 1978; Hinloopen *et al.*, 1983].

Dès que l'on aborde une décision en cherchant à prendre en compte ses multiples facettes, on est la plupart du temps conduit à des questions du type de celles qui sont discutées dans ce texte. Notre objectif premier n'est cependant pas de chercher à résoudre un problème de décision multicritère, comme cela est le cas avec l'analyse hiérarchique de Saaty [1980], mais de contribuer éventuellement à sa structuration.

Les situations où se posent des questions de hiérarchisation ne se retrouvent pas uniquement en analyse multicritère mais également dans bien d'autres contextes dans la gestion des organisations. On peut penser par exemple :

- au responsable d'un groupe travaillant sur la qualité de vie au travail qui, après avoir recensé une liste souvent interminable d'irritants et d'en avoir regroupé certains, doit établir un ordre de priorité entre tous ces éléments afin

de proposer à la direction de son organisation une séquence d'interventions en vue d'améliorer les conditions de travail;

- à des exercices de *brain storming* où l'on emploie entre autres la technique du groupe nominal pour susciter des idées que l'on doit par la suite mettre en ordre;

- à la planification du développement, de l'implantation de projets nécessitant un large ensemble d'activités et de tâches parmi lesquelles il faut établir un ordre de priorité, certaines devant être réalisées concurremment, d'autres séquentiellement.

Alors qu'il peut être relativement facile pour quelqu'un de porter un jugement sur deux entités à la fois, la tâche qui consiste à hiérarchiser tout un ensemble d'entités devient rapidement difficile lorsque leur nombre est le moins grand. Aussi avons-nous développé un algorithme interactif, convivial, avec présentation graphique pour assister les managers, les politiciens et autres dans cette tâche.

Dans un premier temps, cet outil permet de regrouper les entités jugées similaires en macroentités. Dans un deuxième temps, à partir de jugements révélés par le manager sur des paires de macroentités, l'algorithme établit un ou plusieurs préordres.

La prochaine section de ce document présente les notations utilisées, ainsi que la conception algorithmique qui comporte deux phases :

- l'agrégation partielle des entités en macroentités sur la base des « équivalences » révélées;

- la décomposition hiérarchique sur la base des « priorités » révélées.

Dans les autres sections, on retrouve la décomposition hiérarchique; les divers résultats reliés d'une part à la variété des décompositions hiérarchiques respectant un système relationnel d'ordre donné et, d'autre part, la variété des systèmes relationnels d'ordre conduisant à une décomposition hiérarchique particulière; et, enfin, la conclusion résumant brièvement les résultats de cette recherche ainsi que ses limites.

NOTATIONS ET CONCEPTION

On considère un ensemble fini $X = \{X_1, X_2, \dots, X_m\}$ de m entités et une relation binaire (une relation d'ordre : un ordre de priorité, d'importance, de préférence) $R = \{P, I\} \equiv \{>, \sim\}$ définie sur les couples $(X_i; X_j) \in X \cdot X$, telle que

• $X_i P X_j$ une relation d'importance entre X_i et X_j où **P** signifie « X_i est plus important que X_j », ou « X_i est prioritaire à X_j », ou « X_i est préférée à X_j », ... selon le contexte; ($P \equiv \succ$)

• $X_i I X_j$ une relation d'indifférence entre X_i et X_j où **I** signifie « X_i est de même importance que X_j », ou « X_i et X_j sont équivalents », selon le contexte; ($I \equiv \sim$).

Si l'on désigne par $\mathcal{R} \subset X \times X$ l'ensemble des couples vérifiant l'une ou l'autre de ces relations binaires, alors on désire que le graphe $G = (X, \mathcal{R})$ soit transitif, c'est-à-dire que l'on ne tolère pas l'intransitivité. De plus, il est naturel que la relation **I** (relation d'équivalence) soit *symétrique*, c'est-à-dire que si $(X_i, X_j) \in I \Rightarrow (X_j, X_i) \in I$; $\forall X_i \in X$; $\forall X_j \in X$; $\forall i$; $\forall j$; alors que la relation **P** est *antisymétrique*, c'est-à-dire que si $(X_i, X_j) \in P \Rightarrow (X_j, X_i) \notin P$; $\forall X_i \in X$; $\forall X_j \in X$; $i \neq j$.

Il est important d'observer que même si l'on parvient à se prononcer sur le sens de la relation **R** pour toutes les paires $(X_i, X_j) \in X \times X$, il n'est pas évident que l'on puisse déduire le préordre (total) résultant de ce système relationnel d'ordre. De plus, pour un ensemble de m entités il faudrait que l'on compare $m(m-1)/2$ paires, ce qui est assez laborieux dès que m est le moins grand; il n'est d'ailleurs pas certain qu'un manager soit en mesure et (ou) veuille faire toutes ces comparaisons. On ne supposera donc pas que le manager se soit prononcé sur toutes les paires d'entités possibles mais qu'il l'ait fait seulement sur celles pour lesquelles il est suffisamment certain du sens de **R**. Il s'ensuit que le graphe G ne sera pas nécessairement complet; on travaille en quelque sorte avec une « information incomplète » ⁽¹⁾ (voir fig. 1).

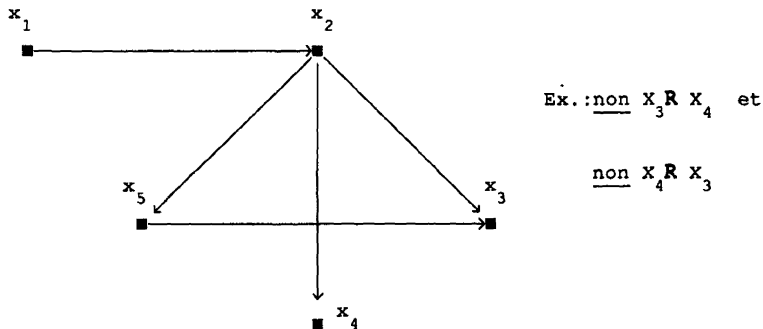


Figure 1

⁽¹⁾ Dans les relations de surclassement par exemple, *non* $X_i R X_j$ et *non* $X_j R X_i$ est interprété comme de l'incomparabilité entre X_i et X_j ; le *non* **R** pourrait également être interprété comme une hésitation entre la préférence stricte et l'indifférence.

Les questions que l'on peut être amené à se poser dans une telle situation sont :

- Si l'on est en présence d'un système relationnel d'ordre (S.R.O.) de type **R**, peut-on obtenir un préordre (partiel ou total) des éléments de **X**? Si oui, ce préordre (cette décomposition hiérarchique) est-il unique? Sinon, peut-on trouver les autres?

- Étant donné une décomposition hiérarchique particulière (une politique, une séquence d'interventions, etc.), peut-on identifier le ou les S.R.O. conduisant précisément à cette décomposition?

C'est à ces deux catégories de questions que l'on essaiera d'apporter des réponses dans la suite de ce texte. Les réponses aux premières questions nous renseignent sur le degré de liberté dont dispose le manager dans ses choix, tout en respectant un ensemble de contraintes « d'aspirations » exprimées, alors que les réponses aux questions de la deuxième catégorie peuvent s'avérer forts utiles pour justifier une « prise de position » de la part du manager.

Pour parvenir à des réponses, il nous semble qu'un *traitement en deux phases* peut être approprié, c'est-à-dire que l'on traitera de façon séparée les deux relations, **I** et **P**. Les relations **I** conduisent à une agrégation partielle (une partition) des entités de **X** en macroentités $A_{j[h_j]}$, $j = 1, 2, \dots, n$; $n \leq m$; chaque macroentité A_j est formée de h_j entités X_i jugées équivalentes. Les relations **P** seront appliquées aux macroentités et conduiront à une décomposition (une partition) de ces n macroentités en v niveaux hiérarchiques $D_{k[\eta_k]}$; chaque niveau comporte η_k macroentités, $k = 1, 2, \dots, v-1, v$. On doit avoir évidemment $\sum_{j=1}^n h_j = \sum_{k=1}^v \eta_k = m$ (voir fig. 2 et 3).

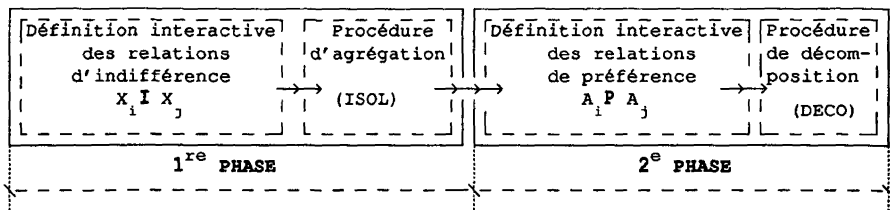


Figure 2. — Synopsis du traitement en deux phases.

DÉCOMPOSITION HIÉRARCHIQUE

Comme nous venons de le mentionner, nous traiterons les relations **R** en deux phases. On applique l'algorithme ISOL pour regrouper en

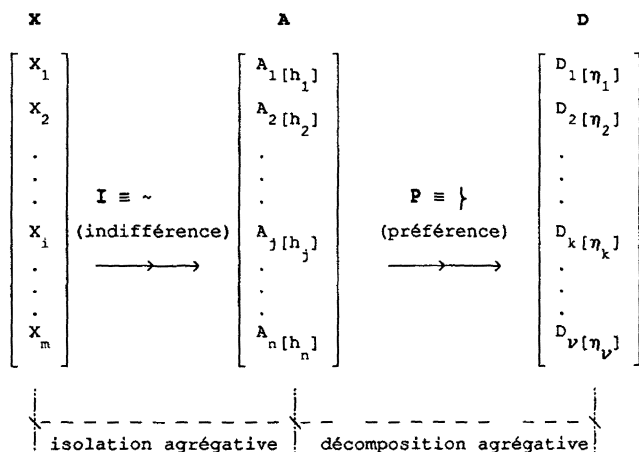


Figure 3. — Partition successive de l'ensemble X, puis de l'ensemble A.

I; P = opérateurs de relations binaires. $X_i = i$ -ième entité; $i = 1, 2, \dots, m$. $A_j[h_j] = j$ -ième sommet agrégé comportant un taux de cohésion γ_j constitué de h_j entités issues de l'ensemble X; $\sum_{j=1}^n h_j = m$; $2/h_j \leq \gamma_j \leq 1$; $n \leq m$; $j = 1, 2, \dots, n$. $D_k[\eta_k] = k$ -ième niveau constitué de η_k sommets issus de l'ensemble A; $\sum_{k=1}^v \eta_k = m$; $v \leq n \leq m$; $D_k[\eta_k] \cap D_r[\eta_r] = \emptyset$; $k = 1, 2, \dots, v$; $k \neq r$.

macroentités ⁽²⁾ les entités de X que l'utilisateur (le manager) juge similaires, c'est-à-dire les entités pour lesquelles la relation I est vérifiée. Il peut s'agir de critères que l'utilisateur juge de même importance ou de tâches qu'il considère devoir être exécutées simultanément, etc. On doit être conscient que selon le contexte et (ou) la procédure employée, l'ensemble X composé de m entités peut déjà être le résultat d'une rationalisation car, souvent, on part d'une très longue liste d'idées un peu pêle-mêle (comme avec la technique du groupe nominal par exemple) que l'on réorganise pour regrouper les idées qui se recoupent. Cette première phase, nous permet :

- de réduire la dimension de l'ensemble X en un nouvel ensemble A composé de n ($< m$) macroentités et
- de traiter le problème des éventuels circuits que le graphe G pourrait contenir.

Par la suite, l'utilisateur est demandé de révéler des relations d'ordre de type P sur des paires de macroentités. On s'assure, à l'aide d'un algorithme ⁽³⁾, que

⁽²⁾ On doit toutefois veiller à ce que le *taux de cohésion* entre les entités constituant chaque macroentité soit suffisamment élevé afin qu'elle forme réellement une *clique*.

⁽³⁾ Le système interactif contient un algorithme qui contrôle l'intransitivité (voir annexe 2, i).

ces derniers jugements sont cohérents, c'est-à-dire qu'ils ne sont pas intransitifs. Dans cette deuxième phase, on applique l'algorithme DECO au système relationnel d'ordre que l'utilisateur vient de définir pour obtenir une décomposition hiérarchique (un préordre total), dite directe, de celui-ci.

Si le nombre m d'entités est relativement petit (ex. : une demi-douzaine), la mise en œuvre de ces deux étapes de traitement est exécutable de façon manuelle, transparente et claire et, dans ce cas, on n'a pas réellement besoin d'une assistance algorithmique. Par contre, même avec beaucoup d'expérience et de capacité mentale, si m augmente, la clarté disparaît et les réponses aux questions évoquées précédemment deviennent rapidement difficiles à trouver. Il est presque impossible de constituer sans contradiction des relations binaires sur des ensembles d'entités de taille réelle lorsque le décideur ne dispose pas d'une assistance algorithmique interactive en arrière-plan.

Dans la pratique, nous avons eu plusieurs fois l'occasion d'être confronté à des situations où des gestionnaires pleinement engagés cherchaient à satisfaire une suite de relations d'indifférence et de préférence qui étaient contradictoires. Par conséquent, l'idée que le décideur puisse disposer d'une « assistance algorithmique » interactive nous semble tout à fait fondée. Pour assurer de façon continue et systématique le contrôle de la « cohérence » du décideur lorsqu'il révèle ses relations d'ordre, sachant que ces dernières exercent une influence décisive sur l'isomorphie entre la réalité et un modèle d'aide à la décision, un traitement en deux phases est également justifié.

L'algorithme de traitement de chacune de ces deux phases, dont les structures sont présentées en pseudo-codes aux annexes 1 et 2, se base sur les deux principes suivants :

1^{re} phase : algorithme ISOL (isolation agrégative)

Principe

Fusion systématique des lignes de la matrice $\mathcal{A}_{[m \times m]}$ associée au graphe $G=(X, I)$ à l'aide des opérateurs logiques $\langle OU \rangle$ ⁽⁴⁾.

La figure 4 illustre cet algorithme à partir d'un ensemble X composé de 21 entités.

2^e phase : algorithme DECO (décomposition agrégative)

Principe

Recherche et séparation séquentielle des sommets de $G=(A, P)$ dont les demi-degrés extérieurs $d_G^+(A)$ sont zéro et dont les demi-degrés intérieurs $d_G^-(A)$ ne sont pas nécessairement non-zéro.

⁽⁴⁾ Une opération logique $\langle OU \rangle$ dans l'algèbre booléenne produit 1 si et seulement si au moins l'une des deux entrées est à l'état 1.

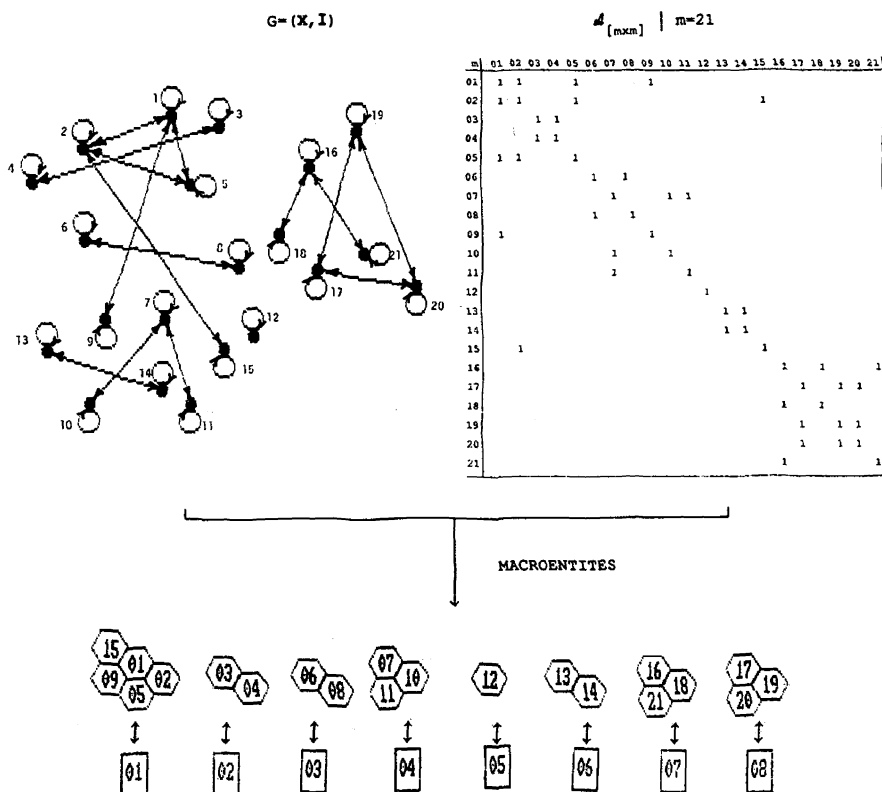
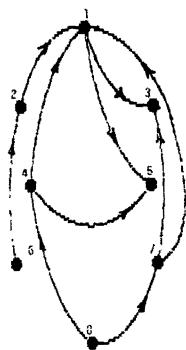


Figure 4. — Isolation aggregative (exemple).

La figure 5 illustre cet algorithme à partir de l'ensemble des macroentités obtenues à la figure 4. Afin d'expliciter d'avantage l'exemple présenté (fig. 4 et 5), voici quelques explications supplémentaires. Bien entendu, la topologie des graphes des figures 4 et 5 ainsi que les relations introduites sont tout à fait arbitraires.

Nous pouvons supposer que l'ensemble $X = \{X_1, X_2, \dots, X_{21}\}$ représente une liste de 21 tâches différentes et que le décideur identifie une relation d'indifférence I entre les couples suivants :

$$\begin{aligned} &X_{01} I X_{02}; X_{01} I X_{05}; X_{01} I X_{09}; X_{02} I X_{05}; X_{02} I X_{15}; \\ &X_{03} I X_{04}; X_{06} I X_{08}; X_{07} I X_{10}; X_{07} I X_{11}; X_{13} I X_{14}; \\ &X_{16} I X_{18}; X_{16} I X_{21}; X_{17} I X_{19}; X_{17} I X_{20}; X_{19} I X_{20}. \end{aligned}$$

$G=(A, P)$ $X_{[n \times n]} \mid n=8$ 

n	01	02	03	04	05	06	07	08
01			1		1			
02		1						
03								
04		1			1			
05								
06			1					
07		1		1				
08				1			1	

HIÉRARCHISATION

niveaux 4 3 2 1

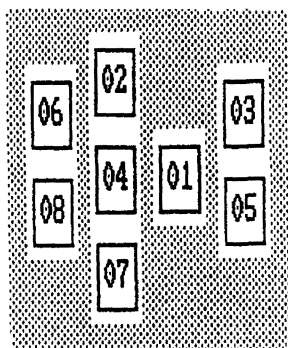
 D_0

Figure 5. - Décomposition agrégative (exemple).

Le traitement d'isolation agrégative ISOL conduit à une agrégation partielle produisant l'ensemble

$$A = \{A_1[5]; A_2[2]; A_3[2]; A_4[3]; A_5[1]; A_6[2]; A_7[3]; A_8[3]\},$$

dont les éléments représentent 8 macroentités, « conglomérats » des entités provenant de l'ensemble X , où

$$A_1[5] = X_{01} \cup X_{02} \cup X_{05} \cup X_{09} \cup X_{15}$$

$$A_2[2] = X_{03} \cup X_{04}$$

$$\begin{aligned}
A_{3[2]} &= X_{06} \cup X_{08} \\
A_{4[3]} &= X_{07} \cup X_{10} \cup X_{11} \\
A_{5[1]} &= X_{12} \\
A_{6[2]} &= X_{13} \cup X_{14} \\
A_{7[3]} &= X_{16} \cup X_{18} \cup X_{21} \\
A_{8[3]} &= X_{17} \cup X_{19} \cup X_{20}
\end{aligned}$$

(voir la figure 4).

La dimension initiale de 21 entités est réduite à 8, facilitant ainsi la lucidité, l'expressivité structurée du problème. Le décideur établit ensuite des relations de préférence **P** entre les huit macroentités lesquelles sont sous le contrôle immédiat et permanent d'un mécanisme algorithmique afin de prévenir la production éventuelle de circuits dans le graphe. Supposons que le décideur avait introduit les relations suivantes :

$$\begin{aligned}
&A_{1[5]} \mathbf{P} A_{3[2]}; A_{1[5]} \mathbf{P} A_{5[1]}; A_{2[2]} \mathbf{P} A_{1[5]}; A_{4[3]} \mathbf{P} A_{1[5]}; A_{4[3]} \mathbf{P} A_{5[1]}; \\
&A_{6[2]} \mathbf{P} A_{2[2]}; A_{7[3]} \mathbf{P} A_{1[5]}; A_{7[3]} \mathbf{P} A_{3[2]}; A_{8[3]} \mathbf{P} A_{4[3]}; A_{8[3]} \mathbf{P} A_{7[3]};
\end{aligned}$$

alors le traitement de décomposition agrégative DECO nous fournit entre autre l'ensemble

$$D_0 = \{D_{1[2]}; D_{2[1]}; D_{3[3]}; D_{4[2]}\};$$

où

$$\begin{aligned}
D_{1[2]} &= A_{3[2]} \cup A_{5[1]} = (X_{06} \cup X_{08}) \cup X_{12}; \\
D_{2[1]} &= A_{1[5]} = X_{01} \cup X_{02} \cup X_{05} \cup X_{09} \cup X_{15}; \\
D_{3[3]} &= A_{2[2]} \cup A_{4[3]} \cup A_{7[3]} = \\
&= (X_{03} \cup X_{04}) \cup (X_{07} \cup X_{10} \cup X_{11}) \cup (X_{16} \cup X_{18} \cup X_{21}); \\
D_{4[2]} &= A_{6[2]} \cup A_{8[3]} = (X_{13} \cup X_{14}) \cup (X_{17} \cup X_{19} \cup X_{20}).
\end{aligned}$$

(voir la figure 5).

Notons que le traitement de décomposition agrégative fournit non seulement le regroupement sur le même niveau hiérarchique des macroentités pouvant être exécutés concurremment, mais également une structure hiérarchique séquentielle respectant les exigences de préséance du décideur. On crée ainsi des *relations d'indifférences implicites* dont le pragmatisme, du point de vu du décideur est indiscutablement important puisque cela peut stimuler des

perceptions nouvelles, de nouveaux axes de jugement du problème initialement posé.

VARIÉTÉ DES DÉCOMPOSITIONS HIÉRARCHIQUES ET DES SYSTÈMES RELATIONNELS D'ORDRE

Il est assez facile de voir qu'il n'existe que rarement une correspondance biunivoque entre les systèmes relationnels d'ordre et les décompositions hiérarchiques. En effet, il existe en général plusieurs décompositions hiérarchiques découlant d'un même S.R.O. et plusieurs S.R.O. peuvent conduire à la même décomposition hiérarchique. Le fait que la correspondance ne soit pas biunivoque n'est pas sans intérêt. En effet, elle conduit à des questions d'analyse combinatoire qui ne sont pas suffisamment explorées jusqu'à présent dans la littérature et elle indique la marge de manœuvre dont disposent le manager et le politicien dans le choix de leur stratégie d'intervention.

Dans ce texte, notre analyse combinatoire se limite à la deuxième phase, c'est-à-dire à celle des relations du type $P, \Gamma : A \Leftrightarrow D$. Toutefois nous sommes conscients de l'existence des aspects combinatoires des relations $I, \Delta : X \Leftrightarrow A$ ainsi que des relations $R, \Lambda = \Gamma \circ \Delta : X \Leftrightarrow D$, dont l'exploration exigera encore des efforts de recherche dans l'avenir.

Les situations de comparaisons binaires dans la pratique de la gestion semblent être dominées par les relations de priorité (voire de préférence), d'où « l'urgence » relative de se concentrer sur l'exploration de la relation $\Gamma : A \Leftrightarrow D$.

Dans un premier temps, on vérifie la multiplicité des décompositions hiérarchiques.

Considérons l'ensemble $A = \{A_{1[h_1]}, A_{2[h_2]}, \dots, A_{j[h_j]}, \dots, A_{n[h_n]}\}$ qui est le résultat de l'isolation agrégative de l'ensemble X (voir l'algorithme ISOL) où

$$X = \{X_1, X_2, \dots, X_i, \dots, X_m\}; \quad n \leq m,$$

puis l'ensemble $D = \{D_{1[\eta_1]}, D_{2[\eta_2]}, \dots, D_{k[\eta_k]}, \dots, D_{v[\eta_v]}\}$, $v \leq n \leq m$ qui est le résultat de la décomposition agrégative de l'ensemble A (voir l'algorithme DECO) avec, naturellement

$$\bigcup_{k=1}^v \left(\bigcup_{k_1=1}^{\eta_k} D_{k[k_1]} \right) = \bigcup_{j=1}^n \left(\bigcup_{s=1}^{h_j} A_{j[s]} \right) = \bigcup_{i=1}^m X_i.$$

Soit

$$d_j(\omega_{k,k_1}) = \min d_j \{ (\omega_{k,k_1} - \omega_{k+k_2,k_3}) \mid \omega_{k+k_2,k_3} \in \mathcal{H}_{k,k_1}; k=1, 2, \dots, v-1; \\ k_1=1, 2, \dots, \eta_k; k_2=1, 2, \dots, v-k; k_3=1, 2, \dots, \eta_{k+k_2} \} \\ d_j(\omega_{k,k_1}) \in \mathbb{N}, \quad \forall j=1, 2, \dots, n,$$

le multiplicateur de distance entre la j -ième macroentité agrégée en tant que le k_1 -ième élément du k -ième niveau ω_{k,k_1} et toutes les autres macroentités agrégées, chacune en tant que le k_3 -ième élément du $(k+k_2)$ -ième niveau ω_{k+k_2,k_3} se trouvant sur l'un des $v-k$ niveaux restants à condition que $\omega_{k+k_2,k_3} \in \mathcal{H}_{k,k_1}$ où \mathcal{H}_{k,k_1} représente l'ensemble des macroentités avec lesquelles ω_{k,k_1} entretient une relation d'ordre ⁽⁵⁾.

Algorithme pour la recherche itérative de ces multiplicateurs (INDI)

Principe

Surveiller le changement de position du niveau d'une macroentité, lequel peut causer — par les superpositions de transitivités — une réaction en chaîne des changements de position du niveau d'une série de macroentités reliées en cascade et (ou) en grappe, parmi lesquelles au moins une précède directement cette macroentité.

Alors, le dénombrement λ du nombre de décompositions hiérarchiques satisfaisant le même système relationnel d'ordre peut être donné par l'expression suivante :

$$\lambda = \prod_{j=1}^n d_j; \quad \lambda \geq 1 \quad (1)$$

La décomposition est *unique* si $\lambda=1$ et *multiple* si $\lambda>1$. L'unicité ou la multiplicité dépend de la « qualité » du S.R.O. construit par le manager. L'unicité indique que l'on est en présence d'une relation Γ biunivoque; λ s'approche de plus en plus de 1 lorsque le S.R.O. est de plus en plus « saturé » en relations d'ordre. *La multiplicité signifie*, dans un contexte de gestion, *que l'on peut satisfaire par plusieurs stratégies équivalentes* (plusieurs décompositions) les mêmes exigences de base, *la même philosophie de gestion*.

Dans la pratique de la gestion, sous l'aspect d'une décomposition donnée, il est exceptionnellement rare que l'on établisse un système relationnel d'ordre

⁽⁵⁾ Autrement dit, \mathcal{H}_{k,k_1} est l'une des lignes de la matrice \mathcal{X} contenant des relations d'ordre, concernant des macroentités de l'ensemble A .

qui minimise la dimension de ce dernier. Bien au contraire, il est plutôt fréquent que ce système relationnel d'ordre soit surchargé par la superposition d'exigences redondantes ou superflues.

Si $\lambda > 1$, c'est-à-dire lorsque la relation $\Gamma : A \Leftrightarrow D$ n'est pas biunivoque, ce qui est le plus souvent le cas, on peut être intéressé à déterminer les λ décompositions hiérarchiques résultant de ce S.R.O. ou à s'interroger sur les divers S.R.O. conduisant à une même décomposition hiérarchique. A partir de l'algorithme DECO, on peut établir les deux préordres extrêmes ⁽⁶⁾ ainsi que toutes les décompositions intermédiaires si λ n'est pas trop grand, ou quelques-unes de ces décompositions (à l'aide de simulations) lorsque λ est grand.

Les prémisses et corollaires qui suivent permettent de déterminer les divers S.R.O. conduisant à une même décomposition hiérarchique D_0 et apportent des éclaircissements sur la complexité combinatoire de $\Gamma : A \Leftrightarrow D$.

Prémisse 1

Le nombre minimum C_{\min} de relations d'ordre résultant de la même décomposition D_0 est déterminé par la formule suivante :

$$C_{\min} = \sum_{\substack{i=2 \\ (\text{incr.} - 1)}}^{i=v} \eta_i; \quad v \geq 2 \quad (2)$$

où

v = le nombre de niveaux de la décomposition D_0

η_i = le nombre de macroentités appartenant au i -ième niveau, $i = v, v-1, \dots, 2$.

Remarque. — $C_{\min} = n-1$ si et seulement si $v = n \geq 2$, c'est-à-dire si les niveaux sont constitués par une seule macroentité.

Prémisse 2

Le nombre maximum C_{\max} de relations d'ordre résultant de la même décomposition D_0 est donné par la formule ci-dessous :

$$C_{\max} = \frac{n(n-1)}{2} - \sum_{i=1}^{i=v} \frac{\eta_i(\eta_i-1)}{2} \quad (3)$$

⁽⁶⁾ Le premier préordre extrême correspond à la décomposition directe, alors que le deuxième représente la décomposition inverse à partir du graphe $G' = (A, P^{-1})$.

où

n = le nombre de macroentités constituant l'ensemble A

v = le nombre de niveaux de la décomposition D_0

η_i = le nombre de macroentités appartenant au i -ième niveau, $i = 1, 2, \dots, v$.

Remarque. — $C_{\max} = n(n-1)/2$ si et seulement si $v = n \geq 2$, c'est-à-dire si les niveaux sont constitués par une seule macroentité.

L'application de C_{\max} résulte d'une saturation totale et absolue en relations d'ordre dans S.R.O.

Prémisse 3

Le nombre quasimaximal $C_{q_{\max}}$ de relations d'ordre résultant de la même décomposition D_0 est égal à la somme sériale des produits, deux par deux, du nombre d'éléments sur les niveaux successifs, c'est-à-dire :

$$C_{q_{\max}} = \sum_{\substack{i=2 \\ (\text{incr.} - 1)}}^{i=v} \eta_i \eta_{i-1} \quad (4)$$

où

v = le nombre de niveaux de la décomposition D_0

η_i = le nombre de macroentités appartenant au i -ième niveau, $i = v, v-1, \dots, 2$.

Remarque. — $C_{q_{\max}} = C_{\min} = n-1$, si et seulement si $v = n \geq 2$ et évidemment $C_{\min} \leq C_{q_{\max}} \leq C_{\max}$.

L'application de $C_{q_{\max}}$ résulte d'une saturation relative en relations d'ordre par rapport à deux niveaux successifs.

Le dénombrement de l'ensemble des systèmes relationnels d'ordre résultant des mêmes niveaux hiérarchiques

En vertu du caractère combinatoire du problème, il est facile de se rendre compte que le décideur peut aboutir aux mêmes niveaux hiérarchiques à partir de nombreux états initiaux différents, autrement dit, à partir de plusieurs systèmes relationnels d'ordre.

COROLLAIRE 1 : *Le nombre κ_0 de systèmes relationnels d'ordre où chacun des systèmes est construit par un nombre minimal C_{\min} de relations résultant toujours de la même décomposition D_0 (en v niveaux où ces derniers sont*

constitués respectivement par $\eta_1, \eta_2, \dots, \eta_v$ macroentités) peut être obtenu par l'expression suivante :

$$\kappa_0 = f_0(\eta_i | i = v, v-1, \dots, 2) = \prod_{\substack{i=2 \\ (\text{incr.} - 1)}}^{i=v} (\eta_{i-1})^{\eta_i} \quad (5)$$

Nous ne donnons pas la démonstration de ce corollaire puisqu'elle n'offre aucune difficulté particulière; κ_0 étant les $(v-1)$ -uplets formés chacun de la η_i -puissance de η_{i-1} macroentités, avec $i = v, v-1, \dots, 2$.

COROLLAIRE 2 — Saturation relative en relations : *Le nombre κ_q de systèmes relationnels d'ordre où la dimension de chacun des systèmes varie entre le nombre minimal C_{\min} et le nombre quasimaximal $C_{q\max}$ résultant toujours de la même décomposition \mathbf{D}_0 (en v niveaux où ces derniers sont constitué respectivement par $\eta_1, \eta_2, \dots, \eta_v$ macroentités) peut être obtenu de la façon suivante :*

$$\kappa_q = f_q(\eta_i | i = v, v-1, \dots, 2) = \prod_{\substack{i=2 \\ (\text{incr.} - 1)}}^{i=v} (2^{\eta_{i-1}} - 1)^{\eta_i} \quad (6)$$

Remarque. — Il n'est pas nécessairement vrai que $\prod_{\substack{i=2 \\ (\text{incr.} - 1)}}^{i=v} (2^{\eta_{i-1}} - 1)^{\eta_i}$ soit

égal à

$$\prod_{\substack{i=1 \\ (\text{incr.} + 1)}}^{i=v-1} (2^{\eta_{i+1}} - 1)^{\eta_i}.$$

Démonstration. — Une macroentité se trouvant sur le i -ième niveau de la décomposition \mathbf{D}_0 doit précéder au moins une macroentité se trouvant sur le $(i-1)$ -ième niveau de \mathbf{D}_0 .

● Le nombre maximal de relations possibles entre chacune des macroentités du i -ième niveau et toutes les macroentités du $(i-1)$ -ième niveau est évidemment égal à

$$\binom{\eta_{i-1}}{1} + \binom{\eta_{i-1}}{2} + \dots + \binom{\eta_{i-1}}{\eta_{i-2}} + \binom{\eta_{i-1}}{\eta_{i-1}} = \sum_{k=1}^{\eta_{i-1}} \binom{\eta_{i-1}}{k} = 2^{\eta_{i-1}} - 1$$

● Le nombre maximal de relations possibles entre deux niveaux successifs est alors à

$$(2^{\eta_{i-1}} - 1)^{\eta_i}$$

● Par conséquent, le nombre maximal de relations possibles donnant la même hiérarchisation des macroentités, voire la décomposition D_0 en v niveaux, est nécessairement égal à (6).

Q.E.D.

COROLLAIRE 3 — Saturation absolue en relations : *Le nombre κ de systèmes relationnels d'ordre où la dimension de chacun des systèmes varie entre le nombre minimal C_{\min} et le nombre maximal C_{\max} résultant toujours de la même décomposition D_0 (en v niveaux avec $\eta_1, \eta_2, \dots, \eta_v$ macroentités par niveaux respectivement) est donné par l'expression suivante :*

$$\kappa = f(\eta_i | i = v, v-1, \dots, 2) = \prod_{\substack{i=v \\ (\text{incr.} - 1)}}^{i=2} \left(\sum_{k=1}^{k=i-1} (2^{\eta_i-k} - 1) \right)^{\eta_i} \quad (7)$$

Remarque. — Il n'est pas nécessairement vrai que

$$\prod_{\substack{i=2 \\ (\text{incr.} - 1)}}^{i=2} \left(\sum_{k=1}^{k=i-1} (2^{\eta_i-k} - 1) \right)^{\eta_i}$$

soit égal à

$$\prod_{\substack{i=1 \\ (\text{incr.} + 1)}}^{i=v-1} \left(\sum_{k=i+1}^{k=v} (2^{\eta_k} - 1) \right)^{\eta_i}.$$

Démonstration. — En admettant qu'une macroentité se trouvant sur le i -ième niveau de la décomposition D_0 doit précéder non seulement au moins une macroentité appartenant au $(i-1)$ -ième niveau, mais au moins une macroentité appartenant au $(i-2)$ -ième niveau, ainsi de suite jusqu'au 1^{er} niveau, alors l'interprétation de κ comme le $(v-1)$ -uplets formés chacun de $\sum_{k=i-1}^{k=v} (2^{\eta_k} - 1)$; $i = v, v-1, \dots, 3, 2$ est évidente.

Q.E.D.

On peut interpréter κ_0 , κ_p et κ comme « les degrés de liberté » du décideur en même temps qu'ils sont aussi un indicateur de l'inutilité de saturer le graphe d'exigences superflues. Le nombre de relations sans influence — bien qu'elles soient logico-mathématiquement toujours correctes — s'accroît très rapidement en fonction du nombre de macroentités impliquées dans le problème étudié.

D'ailleurs, le calcul numérique de (5), (6) et (7) nécessite quelquefois une prouesse informatique. Par exemple, on a été contraint de construire une

arithmétique virtuelle sur un IBM-AT afin de pouvoir effectuer ces calculs, quelque soit la dimension de A ⁽⁷⁾.

On peut synthétiser les divers ensembles de S.R.O. découlant de ces corollaires de la façon suivante :

$$\bullet S_{[\kappa]} = \{ S_2, S_2, \dots, S_i, \dots, S_\kappa \}$$

où

$S_{[\kappa]}$ = l'ensemble des systèmes relationnels d'ordre de cardinalité κ admettant une saturation absolue en contraintes;

S_i = un système relationnel d'ordre dont la dimension peut varier de C_{\min} à C_{\max} et qui résulte toujours en la même décomposition D_0 ,

$$S_i \neq \emptyset; \forall i; i = 1, 2, \dots, \kappa \text{ (voir la prémisses 2 et le corollaire 3);}$$

$$\bullet \hat{S}_{[\kappa_q]} = \{ \hat{S}_1, \hat{S}_2, \dots, \hat{S}_i, \dots, \hat{S}_{\kappa_q} \}$$

où

$\hat{S}_{[\kappa_q]}$ = l'ensemble des systèmes relationnels d'ordre de cardinalités κ_q admettant une saturation relative en contraintes;

\hat{S}_i = un système relationnel d'ordre dont la dimension peut varier de C_{\min} à $C_{q_{\max}}$ et qui résulte toujours en la même décomposition D_0 ,

$$\hat{S}_i \neq \emptyset; \forall i; i = 1, 2, \dots, \kappa_q \text{ (voir la prémisses 3 et le corollaire 2).}$$

$$\bullet \tilde{S}_{[\kappa_0]} = \{ \tilde{S}_1, \tilde{S}_2, \dots, \tilde{S}_i, \dots, \tilde{S}_{\kappa_0} \}$$

où

$\tilde{S}_{[\kappa_0]}$ = l'ensemble des systèmes relationnels d'ordre minimal de cardinalité κ_0 .

\tilde{S}_i = un système relationnel d'ordre minimale dont la dimension est C_{\min} et qui résulte toujours en la même décomposition D_0 ,

$$\tilde{S}_i \neq \emptyset; \forall i; i = 1, 2, \dots, \kappa_0 \text{ (voir la prémisses 1 et le corollaire 1).}$$

$$\text{Bien entendu } \tilde{S}_{[\kappa_0]} \subseteq \hat{S}_{[\kappa_q]} \subseteq S_{[\kappa]} \text{ et } \kappa_0 \leq \kappa_q \leq \kappa.$$

Pour donner un aperçu visuel, nous présentons, à partir de l'exemple introduit aux figures 4 et 5, une série de figures à l'aide desquelles nous pouvons peut-être illustrer mieux toute la richesse :

- des S.R.O. conduisant à la même décomposition hiérarchique et
- des décompositions hiérarchiques découlant d'un même S.R.O.

Considérons la décomposition D_0 de la figure 5. En utilisant (2) et (5), on obtient $C_{\min} = 6$ et $\kappa_0 = 18$, c'est-à-dire qu'il y a 18 graphes de base, dont chacun contient exactement 6 arcs orientés. En appliquant l'algorithme

⁽⁷⁾ Sans vouloir détourner l'attention du lecteur, signalons que κ prend une valeur d'ordre 10^{100} si le nombre d'entités se situe autour de 70.

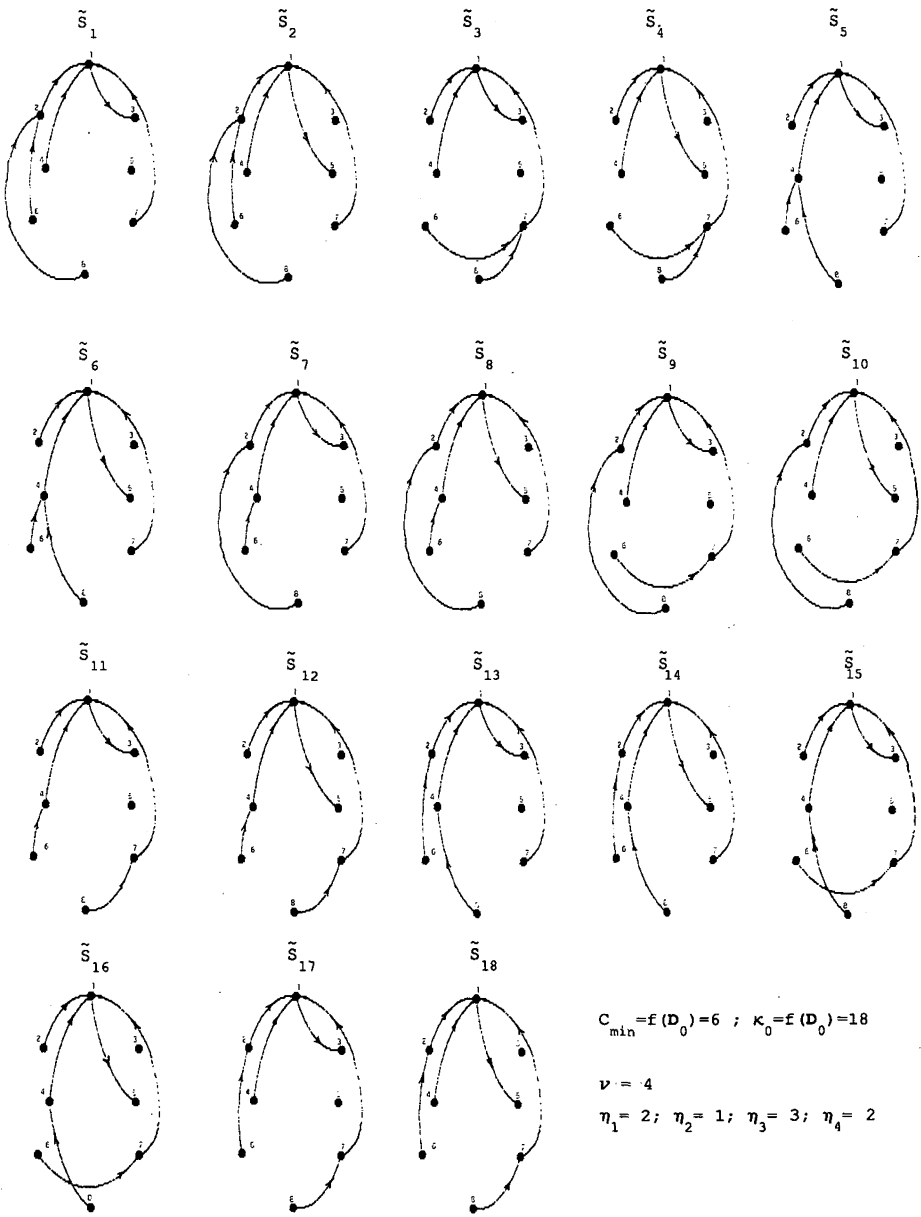


Figure 6

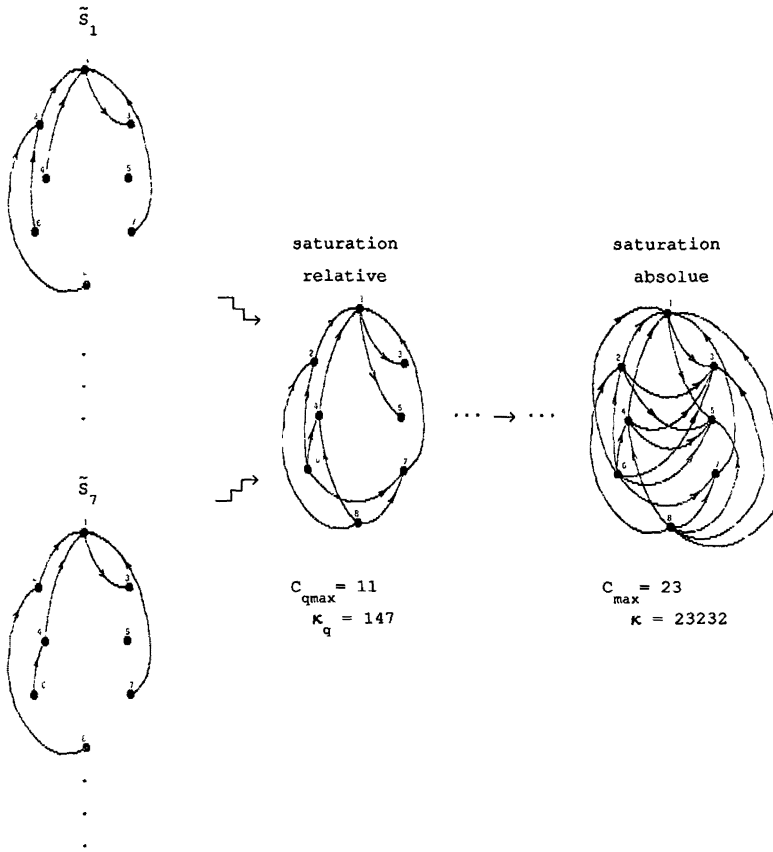


Figure 7

DECO sur ces 18 graphes de base, nous obtenons toujours la même décomposition \mathbf{D}_0 (fig. 6).

L'état de saturation relative peut être réalisé à l'aide de $C_{q_{max}} = 11$ arcs orientés [équation (4)], alors que la réalisation de l'état de saturation absolue nécessite $C_{max} = 23$ arcs orientés [équation (3)]. Si le nombre d'arcs peut varier entre $C_{min} = 6$ et $C_{q_{max}} = 11$, alors on peut définir $\kappa_q = 147$ graphes différents [équation (6)], tandis qu'à partir d'un S.R.O. contenant entre $C_{min} = 6$ et $C_{max} = 23$ arcs orientés, on peut définir $\kappa = 23\,232$ graphes différents [équation (7)]. Soulignons que chacun de ces graphes conduit toujours à la même décomposition directe \mathbf{D}_0 (fig. 7).

Comme nous l'avons indiqué précédemment, plusieurs décompositions hiérarchiques différentes peuvent être issues du même système relationnel d'or-

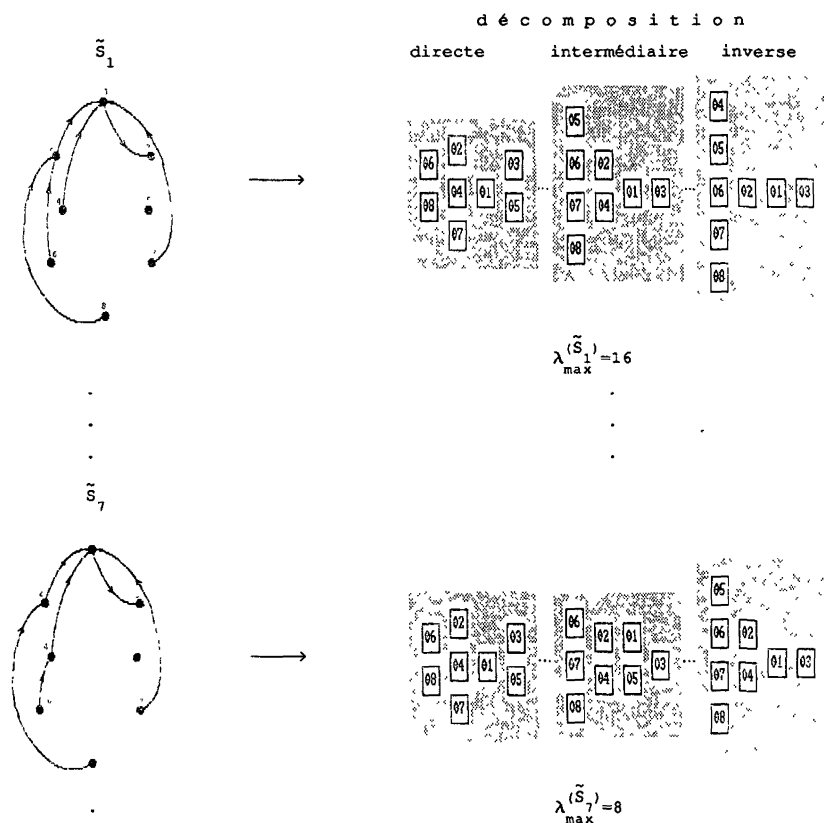
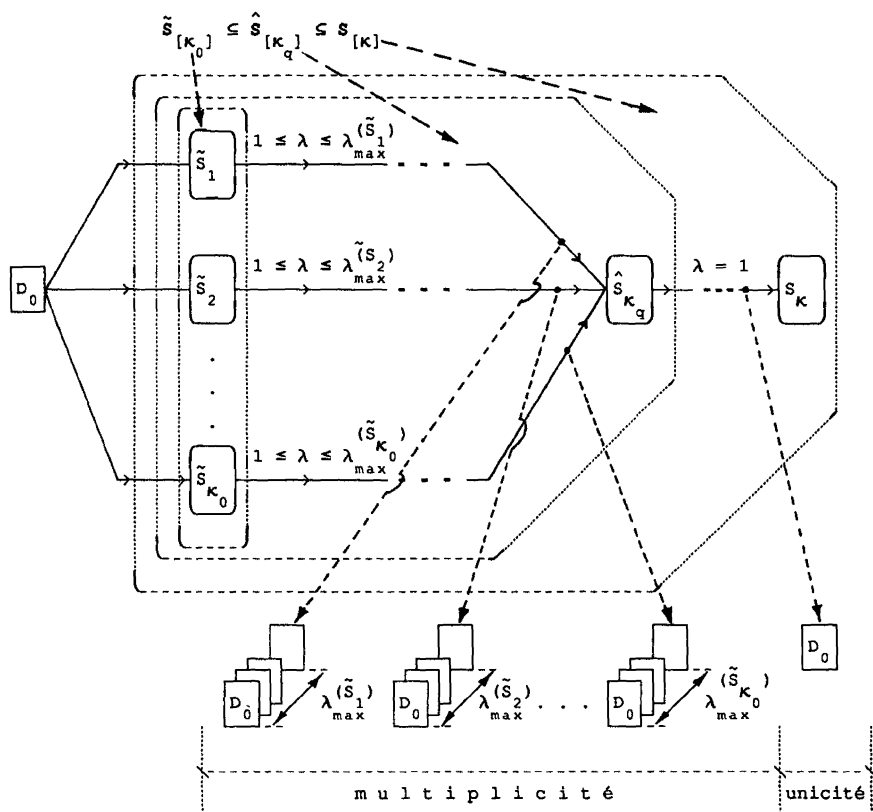


Figure 8

dre. Le coefficient de multiplicité λ atteint son maximum local si les deux préordres extrêmes (décomposition directe et décomposition inverse) ont été déterminé à partir de l'un des κ_0 graphes de base. Après avoir appliqué la procédure algorithmique INDI, dont le principe a été présenté dans les pages qui précèdent, à l'aide de la formule (1), on peut déterminer $\lambda_{\max}(\tilde{S}_i)$, $i = 1, 2, \dots, \kappa_0$ (voir fig. 8).

La figure 9 fait la synthèse des divers S.R.O. pouvant conduire à une même décomposition hiérarchique D_0 ainsi que des différentes décompositions hiérarchiques issues de ces divers S.R.O. On note en particulier que tous les S.R.O. dont la dimension varie entre le nombre quasimaximal $C_{q_{\max}}$ et maximal C_{\max} de relations d'ordre conduisent à une décomposition unique.

Figure 9. — Synthèse de $\Gamma : A \Leftrightarrow D$.

CONCLUSION

Dans la pratique de la gestion, on a de plus en plus tendance à considérer simultanément de nombreux aspects (économique, humain, technique politique, etc.) et de multiples objectifs, le plus souvent conflictuels. Le manager est ainsi confronté à des problèmes passablement complexes. Il peut chercher à les structurer en établissant une certaine hiérarchisation de toutes ces entités (objectifs, critères, opérations, etc.).

C'est précisément pour cette phase de structuration que notre travail veut apporter une contribution. En effet, nos résultats d'analyse combinatoire ainsi que les algorithmes que nous avons développés, accompagnés d'un support informatique visuel et convivial, fournissent au praticien une aide

constructive et interactive pour mettre un peu plus d'ordre dans ce que l'on est souvent justifié d'appeler un « fouillis ». L'accroissement très rapide du nombre de S.R.O. pouvant être définie à partir d'un ensemble d'entités et des décompositions hiérarchiques qui en sont issues justifie amplement le développement de tels outils.

A notre connaissance, on trouve nulle part dans la littérature des indications sur le nombre de S.R.O. pouvant être définis à partir de n entités ou macroentités. Pourtant, il s'agit d'une question fort intéressante puisqu'elle nous renseigne sur l'ampleur des possibilités combinatoires sous-jacentes. On retrouve dans l'annexe 3, un corollaire fournissant quelques indications dans ce sens.

Les résultats obtenus dans cette recherche nous ont permis de mieux percevoir les dimensions des problèmes combinatoires sous-jacents et nous incitent à poursuivre l'effort dans ce domaine. Dans le présent travail, on se limite aux questions de hiérarchisation d'entités sur la base de comparaisons binaires. On ignore les aspects plus complets des problèmes de décision, comme la modélisation des préférences du décideur, l'estimation des valeurs de certains paramètres, le choix entre plusieurs actions potentielles, etc. On s'est plutôt enfoncé dans un logique mathématico-informatique, alors que le comportement humain, le système organisationnel, les systèmes fiscal et financier, le système d'information, l'environnement technologique, économique, politique, etc. affectent de façon très concrète toutes les décisions de gestion.

Malgré cela, nous croyons que cette recherche apporte une contribution originale et qu'elle s'inscrit bien dans le courant des travaux sur l'aide à la décision.

ANNEXE 1

1^{re} phase : algorithme ISOL (isolation agrégative)

Soit $\mathcal{A}_{[m \times n]}$ la matrice associée à m entités portant les relations binaires d'indifférence \mathbf{I} et soit $a_j \in \mathcal{A}_{[m \times n]}$; $i = 1, 2, \dots, m$; $j = 1, 2, \dots, m$ tels que

$$a_{ij} = \begin{cases} 1, & \text{si } \mathbf{X}_i \mathbf{I} \mathbf{X}_j \text{ est vérifié pour les } i \neq j \\ 0 & \text{ailleurs} \end{cases}$$

Début 0

Initialiser à m le nombre d'entités

Lire $A_{[m \times m]}$

Initialiser à zéro le vecteur-indicateur $V_i, i=1, 2, \dots, m$

Initialiser à 1 le compteur ϕ des phases

Initialiser à zéro le compteur S_ϕ des sommets agrégées

Fin 0

Début 1

Incrémenter i de 1 à m

Si $V_i = 1$ alors $i = i + 1$

Sinon

Initialiser à zéro le pointeur e

Incrémenter j de 1 à m

Si $a_{ij} \neq 1$ alors $j = j + 1$

Sinon

Si $V_j = 0$ alors $e = e + 1$

Sinon

Fin si

Fin si

Fin incrément(j)

Si $e = 0$ alors aller à la fin incrément(i)

Sinon

• $S_\phi = S_\phi + 1$

• Incrémenter u de 1 à m

• Mettre $b_{\phi, S_\phi, u} = a_{ij}$
 $\left\{ \begin{array}{l} \text{Remarque: } b_{\phi, S_\phi, u} \in \mathcal{B}_{[S_\phi, xm]}, \text{ où } \mathcal{B}_{[S_\phi, xm]} \\ \text{est un tableau de travail dont la} \\ \text{dimension verticale momentanée est} \\ \text{égale à } S_\phi \end{array} \right.$

Fin incrément(u)

Incrémenter j de 1 à m

Si $a_{ij} \neq 1$ alors aller à la fin incrément(j)

Sinon

Si $V_j = 1$ alors aller à la fin incrément(j)

Sinon

$V_j = 1$

Incrémenter k de 1 à m

Si $a_{ik} \neq 0$ et $a_{jk} \neq 0$ alors $b_{\phi, S_\phi, k} = 1$

Sinon


```

    Fin si
    Fin incrément(k)
  Fin si
  Fin si
  Fin incrément(j)
Fin si
Fin si
Fin incrément(i)
Contrôle du critère de la terminaison de la procédure d'agrégation

  Si  $\sum_{s=1}^{S_\phi} b_{\phi,s,i} = 1; \forall i; i=1,2,\dots,m$  alors aller à la Fin 1 , c.à d.
                                     FIN DE L'ISOLATION AGRÉGATIVE!!!
Sinon
   $\phi = \phi + 1$ 
  Initialiser à zéro le vecteur-indicateur  $V_i; i=1,2,\dots,S_{\phi-1}$ 
  Initialiser à zéro le compteur  $S_\phi$ 
  Incrémenter i de 1 à  $S_{\phi-1}$ 
  Si  $V_i = 1$  alors aller à la fin incrément(i)
Sinon
   $V_i = 1$ 
   $S_\phi = S_{\phi-1} + 1$ 
  Incrémenter u de 1 à m
  Mettre  $b_{\phi,S_\phi,u} = b_{\phi-1,i,u}$ 
  Fin incrément(u)
  Si  $i = S_{\phi-1}$  alors aller à la fin si
Sinon
  Incrémenter j de  $i+1$  à  $S_{\phi-1}$ 
  Si  $V_j = 1$  alors aller à la fin incrément(j)
Sinon
  Incrémenter k de 1 à m
  Si  $b_{\phi-1,i,k} \neq 1$  et  $b_{\phi-1,j,k} \neq 1$  alors aller à la fin
                                     incrément(k)
Sinon
   $V_j = 1$ 
  Incrémenter l de 1 à m
  Si  $b_{\phi-1,i,l} = 1$  et  $b_{\phi-1,j,l} = 1$  alors  $b_{\phi,i,l} = 1$ 
Sinon
  Si  $b_{\phi-1,i,l} = 1$  ou  $b_{\phi-1,j,l} = 1$  alors  $b_{\phi,i,l} = 1$ 

```

```

Sinon
Fin si
Fin si
Fin incrément(l)
Incrémenter ml de 1 à m
Mettre  $b_{\phi-1, i, m_l} = b_{\phi, i, m_l}$ 
Fin incrément(ml)
Aller à la fin incrément(j)
Fin si
Fin incrément(k)
Fin si
Fin incrément(j)
Fin si
Fin si
Fin incrément(i)
Fin si
Fin contrôle
Fin 1
Début 2
Incrémenter i de 1 à  $S_\phi$ 
Initialiser à zéro le compteur  $h_i$  du nombre d'entités agrégées dans
le ième «macrosommet»  $A_i$ 
Incrémenter j de 1 à m
Si  $b_{\phi, i, j} = 1$  alors  $h_i = h_i + 1$  et  $z_{i, h_i} = j$  { Remarque:
 $z_{i, h_i} \in A_i \subset A$ 
 $i = 1, 2, \dots, S_\phi ; 1 \leq h_i \leq m$ 
Sinon
Fin si
Fin incrément(j)
Fin incrément(i)
Afficher icôniquement les «macrosommets»  $A_i$  (c.à d. les entités
agrégées par isolation);  $i = 1, 2, \dots, S_\phi$ 
Mettre  $n = S_\phi$ 
Fin 2
Appeler l'algorithme DECO

```

ANNEXE 2

2^e phase : algorithme DECO (décomposition agrégative)

(i) Étant donné l'ensemble A composé des entités fusionnées de l'ensemble X (voir *fig. 3*) où $\forall A_{i[h_i]} \in A; i = 1, 2, \dots, n$ représentent des macrosommets, résultats du traitement par l'algorithme ISOL sous les contraintes des relations d'indifférence $X_i I X_j$, alors le contrôle permanent pour éviter la formation de circuits (c'est-à-dire l'intransitivité) lors de la *saisie* des relations d'ordre $A_{i[h_i]} P A_{j[h_j]}$ est réalisé à l'aide du segment de programme ci-dessous :

Début

Génération d'un couple $(A_{i[h_i]}, A_{j[h_j]}) ; i=1, 2, \dots, n ; j=1, 2, \dots, n$
et $i \neq j$

Si la relation $A_{i[h_i]} P A_{j[h_j]}$ est retenue par le décideur, alors

Si $i > j$ alors la dimension momentanée de la matrice carrée X
associée est $i \times i$, donc on a $X_{[i \times i]}$

Sinon, la dimension momentanée est $j \times j$, donc on a $X_{[j \times j]}$

Fin si

Contrôle de l'existence de circuits

- chercher les points de sortie du graphe (ligne nulle de la matrice associée X)
- chercher les points d'entrée du graphe (colonne nulle de la matrice associée X)
- supprimer les lignes et colonnes des éléments retenus

Si la procédure de contrôle est applicable jusqu'à ce qu'il ne reste aucun point, alors il n'y a pas de circuit, la relation $A_{i[h_i]} P A_{j[h_j]}$ est acceptée; retour au Début

Sinon (c.à d. s'il n'est plus possible de supprimer des éléments de la matrice associée X), il existe au moins un circuit, la relation $A_{i[h_i]} P A_{j[h_j]}$ est rejetée; retour au

Début

Fin si

Sinon, retour au Début

Fin si

Fin

(ii) Une fois terminée la construction de l'ensemble des couples $\mathcal{R} \subset X \cdot X$ vérifiant la relation $\mathbf{P} \equiv >$, nous exécutons, selon la terminologie de la théorie de graphe, la décomposition d'un graphe sans circuit.

```

Debut 0
  Initialiser à n la nombre de sommets du graphe G
  Initialiser à zéro le compteur  $\nu$  des niveaux
  Mettre  $G_\nu = G$ 
  Initialiser à zéro le compteur  $\eta(\nu)$  du nombre de sommets à
    séparer du  $\nu^{\text{ème}}$  sous-graphe  $G_\nu$ 
Fin 0

Debut 1
  Incréments 1 de 1 à n
  Si  $d_{\nu}^*(\lambda_1) = 0$  alors  $\eta(\nu+1) = \eta(\nu) + 1$  et  $a_{\eta(\nu+1)} = \lambda_1$ 
  Sinon
  Fin si
  Fin incréments(1)
  Si  $\eta(\nu+1) > 0$  alors  $\nu = \nu + 1$  et  $\eta(\nu) = \eta(\nu-1)$ 
  Sinon
    • avertissement («le graphe possède des circuits»)
    • appeler le mécanisme algorithmique pour réviser les relations
      de préséance
    • retourner au Début 0
  Fin si
  Initialiser à zéro le compteur k du nombre des sommets du sous-
    graphe  $G_\nu$ 
  Incréments 1 de 1 à n
  Initialiser à zéro le compteur  $\tau$  de l'indice de congruité
  Incréments j de 1 à  $\eta(\nu)$ 
  Si  $d_{\eta(j)}^* \neq A$  alors  $\tau = \tau + 1$ 
  Sinon
  Fin si
  Fin incréments(j)
  Si  $\tau = \eta(\nu)$  alors  $k = k + 1$  et  $A_k = A_{k-1}$ 
  Sinon
  Fin si

```

```

Fin incrément (1)
Si k ≠ n-η(v) alors («erreur logique!!!»), aller à la Fin 1
Sinon
    n = k
    Si n = 0 alors tracer icôniquement le graphe décomposé; fin
        de la procédure de hiérarchisation; aller à la Fin 1
    Sinon , tracer le graphe partiellement décomposé
    Fin si
Fin si
Retourner au Début 1
Fin 1
HIÉRARCHISATION EST TERMINÉE

```

ANNEXE 3

COROLLAIRE : Le nombre $\Theta_{[n]}$ d'ensembles de systèmes relationnels d'ordre définissables à partir de n entités ($\Theta_{[n]} \in \mathbb{N}$) peut être déterminé par la formule suivante :

$$\Theta_{[n]} = \sum_{k=2}^n \sum_{i=1}^{Q_{[n]}^{(k)}} \left[\psi_{n,i} \sum_{t=1}^{\psi_{k,i}} \left\{ \prod_{j=k}^2 \left(\sum_{r=1}^{j-1} (2^{\eta_{j-r,t}} - 1) \right)^{\eta_{j,t}} \right\} \right]; \quad n \geq 2 \quad (8)$$

Notre raisonnement se fonde sur le principe de base de la partageabilité d'un entier n en 1, 2, ..., $n-1$, n parties :

$$Q_{[n]} = Q_{[n]}^{(1)} + Q_{[n]}^{(2)} + \dots + Q_{[n]}^{(k)} + \dots + Q_{[n]}^{(n-1)} + Q_{[n]}^{(n)} = \sum_{k=1}^n Q_{[n]}^{(k)}.$$

Explication

$Q_{[n]}$ = représente le dénombrement des partages possibles de n entités en 1, 2, ..., k , ..., n parties développées par un processus algorithmique récursif en appliquant le principe de treillis de Young; (notre algorithme fournit en même temps la liste des partages, comme support indispensable).

$Q_{[n]}^{(k)}$ = le dénombrement des partages possibles de n entités en k parties;

$\psi_{k,i}$ = le nombre des configurations possibles de chacun des partages en k parties (autrement dit, le nombre de structures topologiques possibles)

$$\psi_{k,i} = \frac{k!}{\varphi_{1,i}! \varphi_{2,i}! \dots \varphi_{u,i}! \dots \varphi_{l,i}!}; \quad \varphi_{1,i} + \varphi_{2,i} + \dots + \varphi_{l,i} = k; \quad l \leq k$$

$$k = 2, 3, \dots, n; \quad i = 1, 2, \dots, Q_{[n]}^{(k)}$$

l = le nombre d'éléments distincts dans le i -ième dispositif du partage de n en k parties ($1 \leq l \leq k$);

$\varphi_{u,i}$ = la fréquence du u -ième élément distinct dans le i -ième dispositif du partage de n en k parties ($u = 1, 2, \dots, l$; $\varphi_{u,i} \geq 1$);

$\psi_{n,i}$ = le nombre de rangements possibles de n entités dans des niveaux hiérarchiques de la i -ième structure topologique en k parties contenant respectivement $\eta_{1i}, \eta_{2i}, \dots, \eta_{ki}$ éléments;

$$\psi_{n,i} = \binom{n}{\eta_{1i}, \eta_{2i}, \dots, \eta_{wi}, \dots, \eta_{ki}}; \quad \eta_{1i} + \eta_{2i} + \dots + \eta_{ki} = n;$$

$$(\eta_{1i}, \eta_{2i}, \dots, \eta_{ki}) > 0; \quad i = 1, 2, \dots, Q_{[n]}^{(k)}$$

η_{wi} = le nombre des entités constituant le w -ième niveau de la i -ième structure topologique; $w = 1, 2, \dots, k$.

Pour illustrer l'accroissement rapide des $Q_{[n]}^{(k)}$ et de $Q_{[n]}$, nous présentons ci-après un tableau condensé, comme une sortie auxiliaire de l'algorithme développé :

$k \downarrow$	$n \rightarrow$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	...
1	$Q_{[n]}^{(k)}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...
2		1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	...
3			1	1	2	3	4	5	7	8	10	12	14	16	19	21	24	27	30	33	37	...
4				1	1	2	3	5	6	9	11	15	18	23	27	34	39	47	54	64	72	...
5					1	1	2	3	5	7	10	13	18	23	30	37	47	57	70	84	101	...
6						1	1	2	3	5	7	11	14	20	26	35	44	58	71	90	110	...
7							1	1	2	3	5	7	11	15	21	28	38	49	65	82	105	...
8								1	1	2	3	5	7	11	15	22	29	40	52	70	89	...
9									1	1	2	3	5	7	11	15	22	30	41	54	73	...
10										1	1	2	3	5	7	11	15	22	30	42	55	...
11											1	1	2	3	5	7	11	15	22	30	42	...
12												1	1	2	3	5	7	11	15	22	30	...
13													1	1	2	3	5	7	11	15	22	...
14														1	1	2	3	5	7	11	15	...
15															1	1	2	3	5	7	11	...
16																1	1	2	3	5	7	...
17																	1	1	2	3	5	...
18																		1	1	2	3	...
19																			1	1	2	...
20																				1	1	...
21																					1	...
...																						...
$Q_{[n]}$		2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	490	627	792	...

Il est évident que $\Theta_{[n]} \gg Q_{[n]}$ et ce dû aux opérations combinatoires impliquées dans (8).

BIBLIOGRAPHIE

1. C. BERGE, *Principes de combinatoire*, Dunod, Paris, 1968, p. 58.
2. D. M. CVETKOVIC, M. DOOB, J. GUTMAN et A. TORGASEV, *Recent results in theory of graph spectra*, North Holland P. C., Amsterdam, 1988, p. 21-41.
3. P. C. FISHBURN, Lexicographic Orders, Utilities and Decision Rules: A Survey, *Man. Science*, 1974, 20, 11, p. 1442-71.
4. H. FLESHNER, *Eulerian graph*, selected topics in graphs theory 2, L. W. BEINEKE et R. J. WILSON ed., Academic Press, London, 1983, p. 17-53.
5. E. HINLOOPEN, P. NIJKAMP et P. REITVELD, Quantitative discrete multiple criteria choice models in regional planning, *Regional Science and Urban Economics*, 1983, 13, p. 77-102.
6. R. KEENEY et H. RAIFFA, Decisions with multiple objectives: Preferences and Value Tradeoffs, *John Wiley & Sons*, 1982.
7. R. MACGRIMON, *An overview of multiple objective decision making*, Main enjoy under title: Multiple criteria decision making, J. L. COCHRAN et M. ZELENY éd., University of South Carolina Press, Columbia, South Carolina, 1973, p. 18-99.
8. J. PAELINCK, Qualiflex, a flexible multiple-criteria method, *Economic Letters*, 1978, 3, p. 193-197.
9. C. PASCHE, Une approche de l'analyse multicritère par les systèmes-experts, *Cahiers Centre Études Rech. Opér.*, 1987, 29, (1-2), p. 49-60.
10. A. RENYI, *Some remarks on the theory of trees*, Magyar Tudományos Akadémia Mat. Kut. Int. Közl., n° 4, Budapest, 1959, p. 73-85.
11. M. ROUBENS, Analyse et Agrégation des préférences : modélisation, ajustement et résumé de données relationnelles, *Revue Belge de Statistique d'Informatique et de Recherche Opérationnelle*, 1980, 20, 2, p. 36-67.
12. B. ROY, Classement et choix en présence de points de vue multiples (la méthode ELECTRE), *Revue Française d'Informatique et de Recherche Opérationnelle*, 1968, 2, 8, p. 57-75.
13. B. ROY et P. BERTIER, La méthode ELECTRE II – Une application au médiaplaning, dans *O.R.' 72*, M. Ross éd., North Holland P.C., p. 291-302.
14. B. ROY, ELECTRE-III : un algorithme de classement fondé sur une représentation floue des préférences en présence de critères multiples, *Cahiers Centre Études Rech. Opér.*, 1978, 20, 1.
15. B. ROY, M. PRESENT et D. SILHOL, A Programming Method For Determining which Paris Metro Stations Should be Renovated, *European J. Oper. Res.*, 1986, 24, p. 318-334.
16. T. L. SAATY, *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980.

17. T. SOLYMOSI et J. DOMBI, A method for determining the weights of criteria: The centralized weights, *European J. Oper. Res.*, 1986, 26, p. 35-38.
18. J. C. VANSNICK, On the problem of weights in MCDM (The noncompensatory approach), *European J. Oper. Res.*, 1986, 24, p. 288-294.
19. H. VOOGD, Multicriteria evaluation for urban and regional planning, Pion, London, 1983.