



HAL
open science

Réalisations optimales et heuristiques de questionnaires applications: plans d'inspection traduction des tables de décisions

Michel Lemaitre

► **To cite this version:**

Michel Lemaitre. Réalisations optimales et heuristiques de questionnaires applications: plans d'inspection traduction des tables de décisions. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 1975. Français. NNT: . tel-00176581

HAL Id: tel-00176581

<https://theses.hal.science/tel-00176581>

Submitted on 4 Oct 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

PRÉSENTÉE

DEVANT L'UNIVERSITÉ PAUL SABATIER DE TOULOUSE (SCIENCES)

POUR OBTENIR

LE GRADE DE DOCTEUR INGÉNIEUR

Spécialité : Informatique

PAR

Michel LEMAITRE

Ingénieur ENSEEIHT

RÉALISATIONS OPTIMALES ET HEURISTIQUES DE QUESTIONNAIRES APPLICATIONS : PLANS D'INSPECTION TRADUCTION DES TABLES DE DÉCISIONS

Soutenue le 21 Avril 1975 devant la Commission d'Examen :

MM. J. LAGASSE	Président
P. AZEMA	} Examineurs
R. BEAUFILS	
J. LEFEVRE	
C.-F. PICARD	
J.-P. RICHARD	

RÉSUMÉ

Un "questionnaire réalisable" est un graphe arborescent modélisant un processus séquentiel d'identification ou de décision. Les sommets terminaux sont associés aux éventualités ou objets à identifier, les sommets non terminaux à des questions, c'est-à-dire des partitions sur l'ensemble des éventualités. C.F. PICARD (1965, 1972) a étudié les propriétés des questionnaires en relation avec la théorie de l'information. Ici, l'ensemble des questions possibles est restreint à un ensemble de questions "formulables" donné a priori, et on cherche à construire un questionnaire minimisant le coût moyen d'identification.

Nous présentons un nouvel algorithme de réalisation optimale de questionnaire, basé sur une recherche séquentielle "en profondeur d'abord", implémentée en PL/I. L'apport le plus original concerne l'étude de critères heuristiques pour guider la construction de bonnes solutions sous-optimales. Un nouveau critère est présenté, meilleur que celui de la théorie de l'information habituellement préconisé. Les résultats acquis sont utilisés dans deux applications : l'établissement de plans d'inspection pour le diagnostic de panne ; la traduction des tables de décisions à entrées limitées en organigrammes.

ABSTRACT

A "feasible questionnaire" is a tree representing a sequential process of identification or decision-making. Terminal vertices correspond to the different objects or eventualities to be identified, and non-terminal vertices to questions, which are partitions of the set of eventualities. C.F. PICARD (1965, 1972) has studied the properties of questionnaires, from the main point of view of information theory. Here, the set of available questions is restricted to a set of "feasible" questions given a priori, and the aim is to build a questionnaire which minimizes the expected cost of identification.

We give a new algorithm for the optimal solution, based on a sequential search "in depth first", implemented in PL/I. The most original contribution is the study of heuristic criteria which would lead to the construction of good sub-optimal solutions. A new criterion is presented, which is better than that generally recommended given by the information theory. The results obtained are used for two applications : the setting up of an inspection schedule for failure diagnosis ; the conversion of limited-entry decision tables to flowcharts.

PRÉAMBULE

Le développement des moyens informatiques a permis et orienté les recherches et les applications vers des problèmes nouveaux, caractérisés par leur aspect complexe et combinatoire : reconnaissance des formes, commande et test des grands systèmes, recherche opérationnelle, voire "intelligence artificielle". La réalisation de questionnaires en est un exemple typique, à la fois par son intérêt pratique et son intérêt théorique.

Etant donné un ensemble d'éventualités E décrivant les états possibles d'un système déterministe, ainsi qu'un ensemble Q de questions -c'est-à-dire des partitions sur E - permettant de séparer ou d'isoler ces éventualités ; il s'agit de construire une procédure d'identification à l'aide des questions, ayant pour but de retrouver l'éventualité qui correspond à l'état du système. Aux éventualités sont affectés des poids ou des probabilités, aux questions des coûts. Toute procédure d'identification possède donc a priori une espérance mathématique de coût, qu'on cherche à minimiser.

On montrera que l'approche des questionnaires réalisables est la plus intéressante pour résoudre ce type de problème d'identification.

Un questionnaire réalisable est un graphe arborescent modélisant un processus séquentiel d'identification ou de décision, dont les sommets non terminaux sont associés aux questions, et les sommets terminaux aux éventualités séparées.

Les questionnaires réalisables s'opposent aux questionnaires libres étudiés principalement par C.F. PICARD (1965, 1972), dans lesquels toutes les questions (c'est-à-dire partitions sur E) imaginables peuvent être posées. Ici au contraire, l'ensemble des questions possibles est restreint à un ensemble Q de questions dites "formulables", donné a priori.

Outre, les problèmes de codage et le test des circuits logiques qui ont constitué, semble-t-il, la première application de ce modèle, le diagnostic de panne en milieu industriel, le diagnostic médical, la trans-

REMERCIEMENTS

Ce travail de recherche a été effectué au LABORATOIRE d'AUTOMATIQUE et d'Analyse des Systèmes du C.N.R.S., grâce à une allocation de recherche versée par le C.N.R.S. depuis Septembre 1972.

Mes remerciements vont à tous ceux qui, à des titres divers, ont participé à la réalisation de ce mémoire, et en particulier : Messieurs P. AZEMA, M. COURVOISIER, G. GIRALT, J.P. RICHARD ; les membres du collectif de recherche "Automatisation Intégrée" ; Madame M.T. IPPOLITO et Monsieur E. LAPEYRE-MESTRE du Service Documentation ; Monsieur B. MEUNIER, opérateur ; Madame J. PENAVAYRE à qui l'on doit la dactylographie ; Messieurs R. ZITTEL, D. DAURAT, R. LORTAL de l'Atelier d'Imprimerie.

Je remercie également :

- Monsieur le Professeur J. LAGASSE, Directeur du LABORATOIRE d'AUTOMATIQUE et d'Analyse des Systèmes,
 - Monsieur P. AZEMA, Maître de Recherche au C.N.R.S.,
 - Monsieur R. BEAUFILS, Professeur à l'Université Paul Sabatier de Toulouse,
 - Monsieur J. LEFEVRE, Ingénieur à la RADIODÉCHARGE-COMPELEC à Caen, Responsable du Département Qualité,
 - Monsieur C.F. PICARD, Directeur de Recherche au C.N.R.S.,
 - Monsieur J.P. RICHARD, Maître-Assistant à l'Université Paul Sabatier de Toulouse,
- qui me font l'honneur de participer au Jury de soutenance de thèse.

formation des tables de décisions en organigrammes, la classification en reconnaissance des formes, et les problèmes combinatoires tels les tris, peuvent se poser en termes de questionnaires réalisables ; c'est-à-dire, d'une manière générale, tous les problèmes d'organisation en vue de la prise de décisions.

Le modèle que nous avons étudié fait intervenir les aspects poids des éventualités, et surtout coûts des questions qui sont souvent négligés.

Le but de notre travail est d'établir et d'étudier des algorithmes de construction de questionnaires réalisables, fournissant des solutions optimales, puis heuristiques.

Le mémoire est divisé en trois parties :

Dans une première partie, on définit les questionnaires réalisables par un algorithme de construction récursif, et on en déduit les propriétés classiques des questionnaires. Après avoir donné les différentes expressions du coût d'un questionnaire : coût moyen, coût maximum, et des bornes inférieures, on étudie des outils de représentation de l'univers des solutions, et des problèmes équivalents à la construction de questionnaires. Le problème de la réalisation optimale est enfin situé, grâce à de récents travaux sur la complexité des algorithmes, dans l'ensemble de la Recherche Opérationnelle, soulignant l'importance des méthodes sous-optimales pour la résolution pratique.

La seconde partie est consacrée aux constructions directes de questionnaires réalisables, optimales et heuristiques. Après revue des différentes méthodes et améliorations possibles, on construit, à partir de la définition, un algorithme optimal nouveau, implémenté en langage PL/1. Une étude statistique de l'algorithme est faite, grâce à un générateur pseudo-aléatoire de problèmes d'identification.

Pour guider la construction de bonnes solutions sous-optimales, un nouveau critère heuristique est construit et comparé théoriquement et pratiquement avec celui préconisé habituellement.

Les résultats acquis sont utilisés en troisième partie, pour deux applications :

.3.

- le diagnostic d'incidents répertoriés : une implantation en site industriel est présentée à partir d'un cas concret,
- la traduction des tables de décisions en organigrammes.

BIBLIOGRAPHIE

OUVRAGES ET THESES

BAGLIN G. et KLEE J. (1970)

Les tables de décision, initiation pratique
Entreprise moderne d'édition, Paris, 1970.

DUBAIL F. (1967)

"Algorithmes de questionnaires réalisables optimaux au sens de différents critères"

Thèse de 3ème Cycle, Lyon, 1967.

GAREY M.R. (1970)

"Optimal binary decision trees for diagnostic identification problems"
Doctoral thesis, University of Wisconsin, Madison, 1970.

KAUFMANN A. (1968)

Introduction à la combinatoire en vue des applications
DUNOD, Paris, 1968.

KNUTH D.E. (1973)

The art of computer programming

Vol. 3 : Sorting and Searching - Addison-Wesley, New-York, 1973.

NILSSON N.J. (1971)

Problem-solving methods in artificial intelligence

Mc Graw-Hill, U.S.A., 1971.

PICARD C.F. (1965)

Théorie des questionnaires

Gauthier-Villars, Paris, 1965.

PICARD C.F. (1972)

Graphes et questionnaires : Tome 1 : Graphes - Tome 2 : Questionnaires.
Gauthier-Villars, Paris, 1972.

POAGE J.F. (1963)

"The derivation of optimum tests for logic circuits"
Ph.D. Thesis Princeton University, 1963.

POLLACK S.L., HICKS H.T., and HARRISON W.J. (1971)

Decision tables : theory and practice
Wiley, New-York, 1971.

ROY B. (1970)

Algèbre moderne et théorie des graphes
DUNOD, Paris, 1970.

TERRENOIRE M. (1970)

"Un modèle mathématique de processus d'interrogation : les pseudo-
questionnaires"
Thèse, Grenoble, 1970.

ARTICLES, NOTES ET RAPPORTS

Questionnaires et problèmes d'identifications

BEGON F. et TREMOLIERES R. (1971)

"Elaboration d'un modèle logique de diagnostic"
I.R.I.A., Note Scientifique n° 5, Octobre 1971.

CHENAIS D. et TERRENOIRE M. (1971)

"Pseudo-questionnaires"
R.I.R.O. n° R-2, 1971

DOBBIE J.M. (1968)

"A survey of search theory"
Operations Research, 16, pp.525-537, 1968.

GAREY M.R. (1972 a)

"Simple binary identification problems"

I.E.E.E. Trans. on Comp., vol. C-21, pp.588-590, June 1972.

GAREY M.R. (1972 b)

"Optimal binary identification procedures"

SIAM J. Appl. Math., vol 23, pp.173-186, September 1972.

GAREY M.R. and GRAHAM R.L. (1974)

"Performance bounds on the splitting algorithm for binary testing"

Acta Informatica 3, pp.347-355, 1974.

HENRICHON E.G. and KING-SUN FU (1969)

"A nonparametric partitioning procedure for pattern classification"

I.E.E.E. Trans. on Comp., vol. C-18, n°7, July 1969.

HU T.C., TUCKER A.C. (1971)

"Optimal computer search trees and variable-length alphabetical codes"

SIAM J. Appl. Math., vol. 21, n°4, pp.514-532, December 1971.

HYAFIL L. and RIVEST R.L. (1973)

"Graph partitioning and constructing optimal decision trees are polynomial complete problems"

I.R.I.A. Rapport de recherche n°33, Octobre 1973.

KISLITSYN S.S. (1966)

"On discrete search problems"

Cybernetics, 2, n°4, pp.52-58, 1966.

KNUTH D.E. (1971)

"Optimum binary search trees"

Acta Informatica 1, pp.14-25, 270, 1971.

PARKHOMENKO P.P. (1969)

"Optimum questionnaires with questions of unequal value"

Doklady Akademii Nauk SSSR, vol. 184, pp.51-54, January 1969 (en russe)

traduction dans Soviet Physics Doklady, vol. 14, pp.12-14, July 1969.

SLAGLE J.R. and R.C.T. LEE (1971)

"Application of game tree searching techniques to sequential pattern recognition"

Comm. of ACM, 14, 2, February 1971.

Test des circuits logiques et diagnostic des systèmes

CHANG H.Y. (1965)

"An algorithm for selecting an optimum set of diagnostic tests"

I.E.E.E. Trans. on Electr. Comp., vol. EC-14, pp.706-711, October 1965.

CHANG H.Y. (1968)

"A distinguishability criterion for selecting efficient diagnostic tests"

1968 Spring Joint Comput. Conf. AFIPS Conf. Proc., vol. 32, Washington, D.C. : Thomson, 1968, pp.529-534 .

CHU W.W. (1967)

"A mathematical model for diagnosing system failures"

I.E.E.E. Trans. Electr. Comp., vol. EC-16, pp.327-331, June 1967.

KAUTZ W.H. (1968)

"Fault testing and diagnosis in combinational digital circuits"

I.E.E.E. Trans. on Comp., vol. C-17, pp.352-366, April 1968.

L.A.A.S. (1973, 1974)

"Etude visant au contrôle et à la conduite de chaînes de fabrication de dispositifs à semiconducteurs"

Publication n°1022, Avril 1973 - n°1174, Juin 1974.

S.E.M.A. (1970)

"Identification des processus de fabrication relevant d'une technologie de pointe: Détermination optimale de plans d'inspections" :

Notes de travail 1,2,3,4,6 de Janvier à Août 1970, S.E.M.A. (Metra International, 16/20 Rue Barbien, 92128 MONTRouGE).

TABOURIER Y. (1971)

"Sur les plans d'inspection"

Note de travail 141, S.E.M.A., Paris, 1971.

Traduction des tables de décisions

GANAPATHY S. and RAJARAMAN V. (1973)

"Information theory applied to the conversion of decision tables to computer programs"

Comm. ACM 16,9, pp.532-539 (September 1973).

KING P.J.H. (1966)

"Conversion of decision tables to computer programs by rule mask techniques"

Comm. ACM, 9,11, pp.796-801, November 1966.

KING P.J.H. and JOHNSON R.G. (1974)

"Comments on the algorithms of Verhelst for the conversion of limited entry decision tables to flowcharts"

Comm. ACM, 17,1, pp.43-45, January 1974.

MUTHUKISHNAN C.R. and RAJARAMAN V. (1970)

"On the conversion of decision tables to computers programs"

Comm. ACM, 13,6, pp.347-351, June 1970.

MYERS H.J. (1972)

"Compiling optimized code from decision tables"

I.B.M. J. Res. Develop., pp.489-503, September 1972.

POLLACK, Solomon L. (1965)

"Conversion of limited entry decision tables to computer programs"

Comm. ACM, 8,11, pp.677-682, November 1965.

REINWALD L.T. and SOLAND R.M. (1966)

"Conversion of limited entry decision tables to optimal computer programs I: minimum average processing time"

J.ACM, vol. 13, n°3, pp.339-358, July 1966.

SHWAYDER K. (1971)

"Conversion of limited entry decision tables to computer programs - A proposed modification to Pollack's algorithm"

Comm. ACM, 14,2, pp.69-73, February 1971.

SHWAYDER K. (1974)

"Extending the information theory approach to converting limited-entry decision tables to computer programs"

Comm. ACM, 17,9, pp.532-980, September 1974.

VERHELST M. (1972)

"The conversion of limited-entry decision tables to optimal and near-optimal flowcharts : two new algorithms"

Comm. ACM, 15,11, pp.974-980, November 1972.

Divers

COOK S.A. (1971)

"The complexity of theorem proving procedures"

Proceedings of Third Annual ACM Symposium on Theory of Computing,
Shaker Heights, Ohio, pp.151-158, May 1971.

DIJKSTRA E.W. (1972)

"The humble programmer"

Comm. ACM, 15,10, pp.859-866, October 1972.

HUFFMAN D.A. (1952)

"A method for the construction of minimum redundancy codes"

Proc. IRE, Vol. 40,pp. 1098-1101, September 1952.

KARP R.M. (1972)

"Reducibility among combinational problems"

Complexity of Computer Computations, pp.85-103, éd. Miller et Thatcher,
Plenum Press, 1972.

1ÈRE PARTIE

PROBLÈMES D'IDENTIFICATIONS ET

QUESTIONNAIRES RÉALISABLES

CHAPITRE 1.

-:-

HYPOTHÈSES ET BUT - HISTORIQUE - FORMALISATION

I.1. HYPOTHESES ET DEFINITIONS

Un problème d'identification a pour données le quadruplet (E, Q, e, q) ainsi défini :

- * les éventualités forment l'ensemble fini $E = \{E_1, E_2, \dots, E_p\}$, appelé aussi, selon les applications, ensemble de causes ou d'objets à identifier.
- * un poids $e_i > 0$ est affecté à l'éventualité E_i . On pose $e = (e_1, e_2, \dots, e_p)$. La notion de poids est aisément remplacée par celle de probabilité, en normant le vecteur e de façon que $\sum_{1 \leq i \leq p} e_i = 1$.
- * les questions, ou tests, forment l'ensemble fini $Q = \{Q_1, Q_2, \dots, Q_n\}$. Une question Q_i détermine une partition de E en b_i classes : $\{\mathcal{E}_i^1, \mathcal{E}_i^2, \dots, \mathcal{E}_i^{b_i}\}$. Le nombre entier b_i , supérieur ou égal à 2, est appelé base de la question.
On peut aussi considérer une question comme une application surjective, de E dans l'ensemble $\{1, 2, \dots, b_i\}$ des b_i premiers nombres entiers, définie par : $Q_i(E_j) = s \Leftrightarrow E_j \in \mathcal{E}_i^s$. On dit que s est l'issue de la question Q_i pour l'éventualité E_j .
- * un coût $q_i > 0$ est affecté à la question Q_i . On pose $q = (q_1, q_2, \dots, q_n)$.

Ces données peuvent être résumées sous forme d'une table d'identification dont un exemple est donné figure I.1. La table est à double entrée : les p colonnes sont étiquetées par les noms des p éventualités, avec leur poids respectif ; les n lignes sont étiquetées par les noms de n questions, avec leur coût. Enfin, l'élément de la table situé ligne i , colonne j est l'entier $Q_j(E_i)$.

On dit qu'une question Q_i partage ou sépare un sous-ensemble F de E si et seulement si il existe $E_k, E_l \in F \subseteq E$, tels que $Q_i(E_k) \neq Q_i(E_l)$.

Exemple : Q_1 partage $F = \{E_2, E_3, E_4\}$, tandis que Q_3 ne partage pas F (Figure I.1.).

On appelle question élémentaire, une question de base 2 qui sépare E en isolant une seule éventualité des autres. Une question non élémentaire est dite globale. Exemple : la question Q_3 de la Figure I.1. est élémentaire car elle isole l'éventualité E_1 . Les autres questions sont globales.

	E_1 (0,4)	E_2 (0,3)	E_3 (0,2)	E_4 (0,1)
Q_1 (20)	1	1	2	2
Q_2 (10)	1	2	1	2
Q_3 (5)	2	1	1	1

Figure I.1.

1.2. SEMANTIQUE ET BUT : LA PROCEDURE D'IDENTIFICATION

Les données précédentes servent de modèle à une situation dans laquelle nous voulons identifier une éventualité qui vient de se réaliser, sachant que :

- c'est l'un des éléments de E ,
- l'identification ne peut se faire qu'en utilisant les questions de Q et leurs issues.

Exemple : Un sac contient 10 jetons :

- 4 jetons ronds et blancs
- 3 jetons triangulaires noirs
- 2 jetons carrés blancs
- 1 jeton carré noir.

Après le tirage aveugle d'un objet, on doit déterminer, à l'aide de questions, la nature du jeton tiré. Les seules questions autorisées sont :

- le jeton est-il carré ? (coût 20 francs)

- le jeton est-il noir (coût 10 francs)
- le jeton est-il rond (coût 5 francs)

On doit payer le prix indiqué pour chaque question posée.

La figure I.2. résume la situation. On remarque qu'elle est identique à la table de la figure I.1. Une question non formulable serait, par exemple, celle qui effectue la partition $\{\circ \blacksquare\}, \{\blacktriangle \square\}$ -

	 (4)	 (3)	 (2)	 (1)
carré ? (20)	NON	NON	OUI	OUI
noir ? (10)	NON	OUI	NON	OUI
rond ? (5)	OUI	NON	NON	NON

Figure I.2.

- Le but à atteindre est de mécaniser l'identification des éventualités, en produisant, pour chaque table donnée, une procédure d'identification déterministe.
- Lorsque l'identification d'une éventualité est faite, les réponses obtenues aux questions utilisées caractérisent un élément et un seul de E, sans ambiguïté. Autrement dit, la procédure à construire doit permettre de séparer les éléments de E les uns des autres, à l'aide des questions de Q.

Ceci n'est possible que si toutes les colonnes de la table sont différentes.

- Toute procédure étant déterministe (c'est-à-dire qu'une même éventualité est toujours identifiée de la même façon), on pourra attribuer à chacune un coût : coût fixe, coût maximum, ou espérance de coût, suivant les types de procédures.

1.3. DIFFERENTS TYPES DE PROCÉDURES D'IDENTIFICATION

Deux types de procédures peuvent être développées à partir des hypothèses précédentes :

Procédures du type "combinatoire"

On cherche un sous-ensemble de tests de Q , tel que l'application simultanée, ou dans un ordre quelconque, de tous ces tests, permette de distinguer toutes les éventualités les unes des autres.

Dans l'exemple figure I.1., l'ensemble (Q_1, Q_2) suffit à séparer E entièrement. Montrons sur le même exemple, qu'il s'agit en fait d'un problème classique de recouvrement (minimal).

La table I.3. est construite ainsi : si un test permet de distinguer un couple d'événements, l'élément correspondant (à l'intersection de la ligne du test et de la colonne du couple) vaut '1', sinon '0'.

	$E_1 E_2$	$E_1 E_3$	$E_1 E_4$	$E_2 E_3$	$E_2 E_4$	$E_3 E_4$	
Q_1	0	1	1	1	1	0	←
Q_2	1	0	1	1	0	1	←
Q_3	1	1	1	0	0	0	

Figure I.3.

Résoudre le problème d'identification par une procédure de type "combinatoire", c'est sélectionner un sous-ensemble de lignes, qui, dans leur ensemble, couvrent (par des '1') toutes les C_4^2 colonnes de la table.

J.F. POAGE (1963) a remarqué que ce problème est identique à celui de trouver une couverture (minimale) d'une fonction booléenne à partir d'une table d'implicants premiers. Il existe des méthodes classiques pour le résoudre.

L'idée de l'identification "combinatoire" est reprise principalement par H.Y. CHANG (1965) et W.H. KAUTZ (1968), dans le domaine du test des circuits digitaux. F. BEGON et R. TREMOLIERES (1971) formalisent le modèle à l'aide de la logique mathématique, en vue de l'appliquer au diagnostic médical. Tous proposent des algorithmes pour construire des procédures d'identification combinatoire, optimale ou "assez bonne". Dans la dernière référence apparaissent les notions de coût des questions et poids des éventualités.

Procédures de type "séquentiel" ou "questionnaires réalisables"

Jusque là les questions étaient posées indépendamment les unes des autres ; au contraire, on tient compte maintenant, au cours de l'identification, du résultat des questions précédentes pour poursuivre.

La procédure est décrite par une arborescence (exemple figure I.4.) dont les sommets non terminaux représentent les questions à poser. -en particulier la racine indique la première- et les sommets terminaux correspondent aux différentes éventualités séparées.

Une fois la procédure construite pour un problème d'identification donné, on utilise un seul chemin de l'arborescence à chaque identification nouvelle, depuis la racine jusqu'à l'éventualité recherchée.

Le chemin est déterminé par les issues successives des questions, figurées par les arcs de sortie des questions.

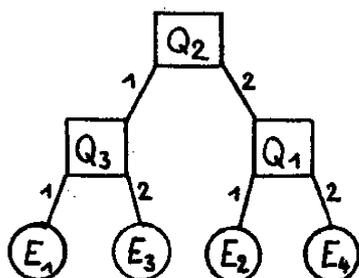


Figure I.4.

Il s'agit bien d'une procédure déterministe, puisque pour une même éventualité, on posera toujours les mêmes questions dans le même ordre. Cette remarque est importante car elle permettra d'affecter un coût à chaque procédure choisie. On pourra dès lors parler d'optimalité ou sous-optimalité d'une procédure, et nous verrons que l'approche "questionnaire" est plus intéressante que la première, car de coût moindre.

L'idée d'arborescence de questions est apparue indépendamment dans de nombreux domaines. Les principaux travaux dans lesquels la même formulation est étudiée sont ceux de H.Y. CHANG (1968) et W.H. KAUFZ (1968) en vue du test des circuits digitaux, dans lesquels les notions de poids des éventualités et de coût des questions n'apparaissent pas. Des algorithmes de construction de procédures sous-optimales sont proposés dans ces articles. Les questionnaires (C.F. PICARD (1965)) sont des procédures d'identification séquentielle dans lesquelles toutes les questions (c'est-

à-dire les partitions) imaginables peuvent être utilisées. Seules, les éventualités à séparer sont pondérées, les questions ont des coûts égaux. On rejoint les problèmes sur le codage étudiés par D.A. HUFFMAN (1952). P.P. PARKHOMENKO (1969) fait apparaître le coût des questions. L'idée de restreindre l'ensemble des questions à un ensemble de "questions formulables", de façon à englober dans le modèle des problèmes plus concrets, était déjà contenue implicitement dans les derniers chapitres de l'ouvrage de C.F. PICARD (1965). Elle est développée par F. DUBAIL (1967) et dans la nouvelle édition sur la théorie des questionnaires de C.F. PICARD (1972), Tome 2, sous le nom de "questionnaire réalisable" repris dans ce mémoire. Ces auteurs s'intéressent aux aspects théoriques du problème liés à la théorie de l'information et la combinatoire. Y. TABOURIER (1971) et la S.E.M.A. (1970) utilisent le modèle pour le diagnostic de pannes dans les usines de fabrication, et proposent des algorithmes pour construire des questionnaires optimaux ou sous-optimaux appelés "plans d'inspection".

M.R. GAREY (1972 a,b) étudie spécialement les propriétés des questionnaires réalisables optimaux, (questions de base 2) pour en faciliter la construction, et propose un algorithme basé sur la programmation dynamique.

Nous verrons en troisième partie que le problème de la traduction des tables de décision se pose dans des termes presque identiques. Citons, par exemple, L.T. REINWALD et R.M. SOLAND (1966) (Construction de procédures optimales), S. GANAPATHY et V. RAJARAMAN (1973) (Construction sous-optimale).

Les processus de décision arborescents peuvent être utilisés en reconnaissance des formes. Exemple : E.G. HENRICHON et KING-SUN FU (1969), J.R. SLAGLE et R.T.C. LEE (1971). De même en informatique pour les algorithmes de tris et de recherche : voir C.F. PICARD (1972, chap. 14), D.E. KNUTH (1973).

Dans la suite du mémoire, nous utiliserons toujours le mot "questionnaire", l'adjectif "réalisable" étant sous-entendu sauf mention explicite. Nous ne nous intéresserons plus qu'à ce type de procédure d'identification.

La figure I.5. schématise les deux étapes de l'identification :
- l'étape de construction du questionnaire, à partir de la table

d'identification, analogue à l'étape compilation d'un programme source. (L'analogie deviendra réalité pour la traduction des tables de décision).

- l'étape utilisation du questionnaire, analogue à l'étape exécution du programme objet, dans laquelle les "données" seraient les issues successives des questions.

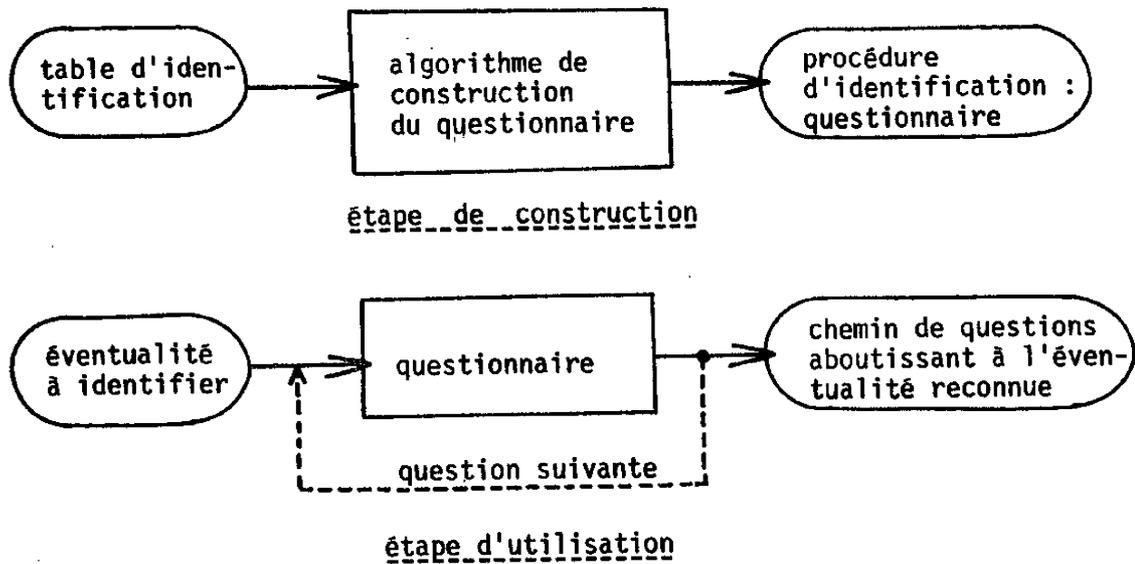


Figure I.5.

I.4. DEFINITION ET CONSTRUCTION DES QUESTIONNAIRES REALISABLES

La définition plutôt intuitive du paragraphe précédent doit être remplacée par une définition plus formelle, qu'on présente sous forme d'un algorithme de construction.

A partir de cette définition, on dégagera les propriétés principales des questionnaires (réalisables).

La terminologie employée pour les graphes est en accord avec celle exposée par C.F. PICARD (1972), Tome 1 : Graphes.

Définition : Soit (E, Q, e, q) un quadruplet décrivant un problème d'identification. On appelle questionnaire réalisable sur F sous-ensemble de E , un graphe $\mathcal{Q} = (X, \Gamma)$ construit par l'algorithme QUESTIONNAIRE (F) ci-après.

Un sommet du questionnaire (un élément de X) est de la forme (A,i) , avec A sous-ensemble de F , et $i \in \mathbb{N}$ ou $i = 0$. Un sommet (A,i) représente le fait de poser la question Q_i sur l'ensemble A d'éventualités. Si $i = 0$, on ne pose pas de question.

Dans l'algorithme QUESTIONNAIRE (F), la fonction CHOIX (F) fournit l'indice d'un question qui partage F ; si aucune question ne partage F , CHOIX (F) vaut "0".

```

1 algorithme : QUESTIONNAIRE ( $F$ ) récuratif ;
2    $i :=$  CHOIX ( $F$ ) ;
3   produire le sommet  $(F,i)$ , appelé "racine du questionnaire sur  $F$ " ;
4   si  $i \neq 0$  alors faire ;
5     pour  $s := 1$  à  $b_i$ , pas 1, faire ;
6        $F_i^s := F \cap E_i^s$  ;
7       si  $F_i^s \neq \emptyset$  alors faire ;
8         QUESTIONNAIRE ( $F_i^s$ ) ;
9       produire un arc orienté de  $(F,i)$  vers la
        racine du questionnaire sur  $F_i^s$  ;
10      fin ;
11    fin ;
12  fin ;
13 fin QUESTIONNAIRE ;

```

Commentaires

- a) lorsque F vaut E lui-même, on obtient un questionnaire pour le problème (E,Q,e,q) .
- b) plusieurs questionnaires sur E , différents, peuvent être construits à partir d'une même table. L'algorithme de construction, grâce à la fonction CHOIX, contient potentiellement toutes ces possibilités. La fonction CHOIX peut-être considérée comme non-déterministe, au sens où on ne peut en prévoir le résultat avant l'exécution ; et de même pour l'algorithme (voir la discussion du chapitre IV).
- c) un questionnaire sur F formé d'un seul sommet est dit dégénéré. Cela se produit lorsque F n'est séparable par aucune question de Q : le sommet est donc $(F,0)$. Dans ce cas :

- soit $\text{Card}(F) = 1$, et F correspond à une éventualité isolée ;
 - soit $\text{Card}(F) > 1$, et F est constitué d'éventualités inséparables, ayant des colonnes identiques dans la table d'identification.
- d) l'ensemble des F_i^S non vides produits par la ligne 6 de l'algorithme constitue une partition de F . Cela découle directement du fait que $\{e_i^1, e_i^2, \dots, e_i^{b_i}\}$ est une partition de E .

Propriétés des questionnaires

- P1 : un questionnaire sur F est
- .soit réduit à un sommet : $(F,0)$.
 - .soit le sous-graphe obtenu par suppression de la racine (F,i) est formé de questionnaires dont les successeurs de (F,i) sont les racines.

Cette propriété, illustrée par la figure I.6., découle directement de la forme récursive de la définition : si $i = 0$ (ligne 4), l'algorithme se termine en n'ayant produit qu'un seul sommet ; sinon, la boucle (lignes 5 à 11) produit pour chaque issue de la première question, un questionnaire nouveau possédant au moins un sommet. \square

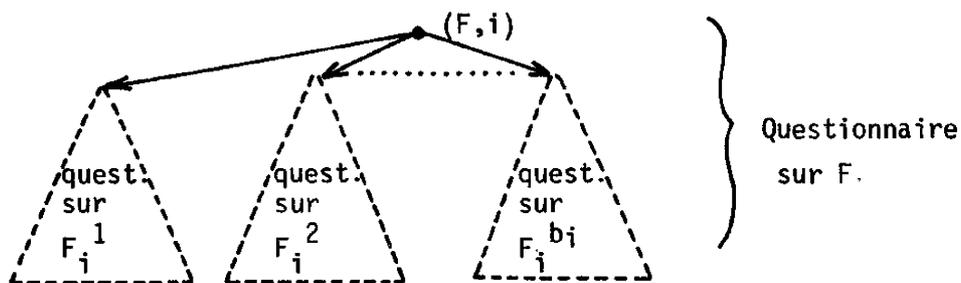


Figure I.6.

- P2 : un questionnaire est une arborescence finie.

Pour montrer que c'est une arborescence, montrons, dans le cas non dégénéré, qu'il existe un sommet et un seul (la racine), relié à tout autre par un chemin unique dont il est l'origine :

- par construction, il y a un chemin depuis la racine du questionnaire jusqu'à tout sommet,

- tout sommet, sauf la racine, possède un antécédent et un seul : en effet, tout sommet est créé par la ligne 3 de l'algorithme, et le seul arc incident intérieurement à ce sommet est créé par l'algorithme au niveau de récursivité précédent, ligne 9 ; le chemin est donc unique depuis la racine jusqu'à tout sommet,
- la racine est le seul sommet possédant cette propriété, puisque, en particulier, aucun arc du questionnaire n'aboutit à la racine.

Montrons, en deux étapes, que le questionnaire est un graphe fini :

- si le questionnaire sur F est infini, alors l'un au moins des questionnaires sur les F_i^S est infini (le nombre d'issues de Q_i est fini). En poursuivant ainsi par induction, on exhibe un chemin infini depuis la racine.
- or, si le sommet (A,i) est l'antécédent du sommet (B,j) , on a $B \subset A$, l'inclusion étant stricte. Le long d'un chemin, les sous-ensembles de E sont inclus les uns dans les autres. Aucun chemin ne saurait donc être de longueur supérieure à $\text{Card}(E)-1$, et ceci contredit l'hypothèse du graphe infini \square

En démontrant que le graphe produit par l'algorithme est fini, on démontre du même coup que cet algorithme se termine effectivement.

P3 : l'ensemble des sommets d'un questionnaire non dégénéré peut être partagé en deux classes :

- la classe des sommets terminaux de l'arborescence, du type $(A,0)$.
- la classe des sommets non terminaux du type (A,i) avec $i \neq 0$, ayant au moins deux arcs incidents extérieurement.

L'accès à la boucle (ligne 5 à 11) qui produit des arcs incidents extérieurement, est commandé par le test sur i ligne 4. Par conséquent, $i = 0$ équivaut à (A,i) sommet terminal, et $i \neq 0$: (A,i) sommet non terminal.

Supposons $i \neq 0$. Par définition de CHOIX, Q_i partage A en au moins deux sous-ensembles ; dans la boucle (5-11), la condition $A \cap \mathcal{C}_i^S \neq \emptyset$ (ligne 7) sera vraie au moins deux fois : le sommet (A,i) a au moins deux arcs incidents extérieurement. \square

Les sommets non terminaux apparaissent donc comme les supports des questions, tandis que les sommets terminaux supportent les éventualités simples ou inséparables.

Si toutes les éventualités sont séparables, les sommets terminaux sont $(E_1,0), (E_2,0)\dots(E_p,0)$.

Dans la représentation courante des questionnaires, un sommet non terminal sera représenté par un carré avec le nom de la question, un sommet terminal par le(s) nom(s) de(s) éventualité(s) séparée(s) entouré(s) d'un cercle. (exemple figure I.7., reprenant les données de la figure I.1.).

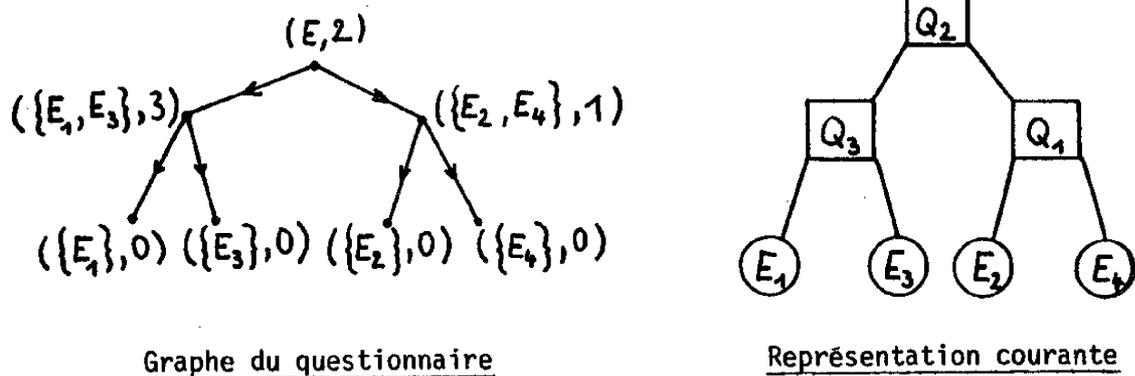


Figure I.7.

P4 : sur tout chemin d'un questionnaire, les indices des questions employées sont distincts.

Soient (A,i) et (B,j) deux sommets quelconques d'un même chemin, (A,i) précédant (B,j) sur ce chemin.

Il y a une issue et une seule, s , de la question Q_i sur A , conduisant au sommet (B,j) , et donc $B \subseteq A \cap \mathcal{E}_i^s$. Par conséquent,

$(\forall E_k, E_l \in B) (Q_i(E_k) = Q_i(E_l)) : Q_i$ ne partage pas B , et donc $i \neq j$. \square

De cette propriété, on déduit qu'aucun chemin du graphe ne peut être de longueur supérieure à $\text{Card}(Q)$.

Remarquons que la même question peut être employée sur deux chemins distincts du questionnaire.

P5 : dans tout questionnaire, il existe un chemin du sommet (A,i) au sommet (B,j) si et seulement si $A \supset B$.

Autrement dit : la relation d'ordre partiel induite sur les sommets du questionnaire est équivalente à la relation d'inclusion pour les sous-ensembles de E utilisés.

On a déjà montré que si (B,j) est un descendant de (A,i) , alors $B \subset A$.

Réciproquement, supposons $B \subset A$:

- soit (A,i) et (B,j) sont sur le même chemin, auquel cas (A,i) ne peut être qu'un ascendant de (B,j) ,
- soit (A,i) et (B,j) ne sont pas sur le même chemin.

Considérons le chemin intersection des deux chemins menant de la racine à (A,i) et (B,j) , et nommons (C,k) le dernier sommet de ce chemin (ce peut être la racine du questionnaire). Voir figure I.8.

A et B sont alors deux sous-ensembles de C correspondant à des issues distinctes de la question Q_k sur C ; donc $A \cap B = \emptyset$.

Conclusion : la seule possibilité, lorsque $B \subset A$, est que (B,j) soit un descendant de (A,i) . \square

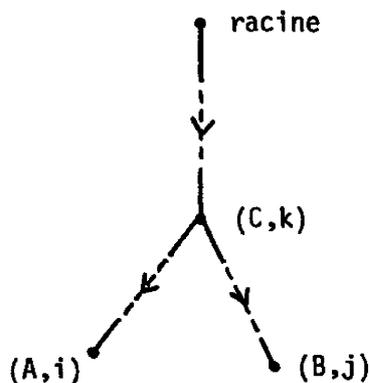


Figure I.8.

Le nombre d'arcs incidents extérieurement à un sommet (A,i) sera appelé base réelle de la question Q_i appliquée sur A . Il se peut qu'une base réelle de Q_i soit inférieure à b_i .

Énonçons la propriété classique des arborescences :

P6 : soit $a = (a_1, a_2, \dots, a_m)$ l'ensemble des bases réelles des m questions utilisées pour un questionnaire séparant E complètement. Alors $\sum_{1 \leq i \leq m} a_i - m + 1 = \text{Card}(E)$.

En effet, il y a $\sum_{1 \leq i \leq m} a_i$ arcs incidents extérieurement, et $m - 1 + \text{Card}(E)$ arcs incidents intérieurement dans l'arborescence. \square

1.5. RESTRICTIONS ET EXTENSIONS DU MODELE

De nombreux modèles rencontrés en informatique et en algorithmique se mettent sous la forme de problèmes d'identification : diagnostic, traduction des tables de décision, codage, reconnaissance des formes, recherches en tables, tris,...

Parmi les modèles proposés pour les étudier, nous avons déjà cité les questionnaires avec questions libres de PICARD.

Nous allons présenter rapidement deux problèmes typiques d'identification, cas particuliers du modèle général exposé : l'identification élémentaire et l'identification lexicographique ; puis une extension possible : les pseudo-questionnaires.

Identification élémentaire

Un objet caché se trouve dans une boîte parmi n , qu'il s'agit de déterminer. La probabilité pour l'objet d'être dans la boîte i est e_i , le coût d'ouverture de la boîte est q_i .

C'est un problème d'identification de n objets avec n questions élémentaires, dont nous reparlerons. Il existe un algorithme pour construire un questionnaire optimal en $k \cdot n \cdot \log n$ étapes (M.R. GAREY (1972 a) et chapitre V.2.).

Identification lexicographique (recherche en dictionnaire)

Un mot inconnu X doit être identifié ou classé parmi n 'clés'. Les questions autorisées sont du type : "X est-il inférieur, égal ou supérieur à K_i ?". Le résultat de l'identification sera :

$X = K_i$ (probabilité p_i)
ou bien $K_{i-1} < X < K_i$ (probabilité p'_i).

On se donne comme hypothèse les $2n+1$ probabilités p_1, p_2, \dots, p_n et p'_0, p'_1, \dots, p'_n , avec $\sum p_i + \sum p'_i = 1$.

Ce problème, qui a une grande importance en informatique (recherche en tables, codes) a été étudié pour le cas $p_i = 0$ par T.C. HU & A.C. TUCKER (1971), qui proposent un algorithme pour construire un questionnaire optimal en $K \cdot n \cdot \log n$ étapes, et par D.E. KNUTH (1971) dans le cas général ($K \cdot n^2$ étapes).

D'autres références et une présentation plus complète sont établies dans D.E. KNUTH (1973, chap. 6).

Ce problème peut être posé comme un problème d'identification, avec $2n+1$ éventualités : $E'_0, E_1, E'_1, E_2, E'_2, \dots, E_n, E'_n$, avec : poids de $E'_i = p'_i$, poids de $E_i = p_i$; la question Q_i effectue la partition :

$$\{E'_0, E_1, E'_1, \dots, E_{i-1}, E'_{i-1}\}, \{E_i\}, \{E'_{i+1}, E_{i+1}, E'_{i+1}, \dots, E_n, E'_n\}$$

Pseudo-questionnaires

Notre modèle suppose qu'une même éventualité provoque toujours la même issue d'une même question. Il peut être intéressant pourtant d'utiliser un type de question, dont l'issue, pour une même éventualité, serait plus ou moins sûre.

Il se peut même, pour certaines applications, que ce soit le seul type de question disponible, par exemple pour certains diagnostics médicaux.

Pour être plus précis, au lieu de se donner des partitions sur E , on se donne pour chaque question une matrice de probabilités conditionnelles (matrice stochastique) :

$$M = (e_i^j) \text{ avec } e_i^j = \Pr \{ \text{issue } j / E_i \} .$$

Dans le modèle précédent, ces e_i^j valaient 0 ou 1.

De la même façon, on établit un graphe arborescent.

A chaque sommet est associé, non plus un sous-ensemble de E , mais un vecteur donnant la probabilité de chaque éventualité, étant donné le chemin parcouru. Ce vecteur est calculé à partir des matrices M et de la distribution initiale des probabilités sur E .

Dans le modèle précédent, le processus s'arrête (sommet terminal) lorsque la probabilité d'une éventualité, étant donné le chemin

parcouru, est 1 (on suppose toutes les éventualités séparables). Ici, on pourra ne jamais aboutir à un tel résultat. En plus du choix des questions, vient donc se greffer le problème de l'arrêt de processus.

Cette généralisation des questionnaires a été étudiée par M. TERRENOIRE (1970) et D. CHENAIS & M. TERRENOIRE (1971).

On trouvera d'autres types de problèmes voisins d'identifications et de recherches dans les bibliographies de S.S. KISLITSYN (1966) et J.M. DOBBIE (1968).

CHAPITRE II.

-:-

COÛTS D'UN QUESTIONNAIRE - BORNES INFÉRIEURES

-

II.1. COÛTS D'UN QUESTIONNAIREIntroduction

Soit un problème d'identification (E, Q, e, q) , et F un sous-ensemble de E pouvant être E lui-même, sur lequel on construit un questionnaire particulier \mathcal{Q} , dont nous allons évaluer le coût.

On supposera, dans ce chapitre, que toutes les éventualités de E sont séparables, c'est-à-dire que les sommets terminaux de notre questionnaire sont du type $(\{E_i\}, 0)$. Si tel n'est pas le cas, on s'y ramène simplement en regroupant les sous-ensembles de E inséparables en nouvelles éventualités dont les poids sont la somme des poids des éventualités regroupées, et en renommant E . A toute éventualité de F correspond donc un unique chemin maximal dans le questionnaire.

Lorsque nous utilisons pratiquement le questionnaire, nous empruntons un chemin maximal et un seul pour chaque nouvelle identification. Nous définirons naturellement le coût d'une identification par la somme des coûts des questions utilisées ;

Soit \mathcal{Z}_i l'ensemble des indices des questions utilisées pour identifier E_i dans le questionnaire, c'est-à-dire les indices des questions sur le chemin maximal aboutissant à $(\{E_i\}, 0)$:

$\mathcal{Z}_i = \{k / (A, k)\}$ est un ascendant de $(\{E_i\}, 0)$. Le coût d'identification de E_i est alors $\sum_{k \in \mathcal{Z}_i} q_k$.

Pour évaluer le coût d'un questionnaire, on s'intéresse :

- soit au coût maximum d'une identification, auquel cas les poids ou probabilités des éventualités n'interviennent pas,
- soit au coût pondéré d'une identification ; c'est ce dernier coût qui sera le plus intéressant dans la pratique, car il fait intervenir le poids des éventualités.

Dans les deux cas, on parle de coût du questionnaire sur F , et on le note respectivement $K(F)$ et $C(F)$, en faisant référence implicitement au questionnaire donné \mathcal{Q} .

Coût maximum d'identification

c'est donc

$$K(F) = \text{Max}_{E_i \in F} \sum_{k \in \mathcal{Z}_i} q_k \quad (1)$$

Montrons que (1) peut se mettre sous la forme récursive :

$$K(F) = q_r + \text{Max}_{1 \leq s \leq b_r} K(F_r^s) \quad (2)$$

avec la convention : $\text{Card}(A) = 0$ ou $1 \Rightarrow K(A) = 0$; où (F, r) est la racine du questionnaire, et $K(F_r^s)$ le coût du questionnaire sur F_r^s obtenu après suppression de la racine dans \mathcal{Q} (Voir définition § I.4. et figure I.6.).

- tous les \mathcal{Z}_i dans (1) comportent l'indice r de la racine, donc on peut isoler dans (1) q_r , apparaissant sous tous les \sum :

$$K(F) = q_r + \text{Max}_{E_i \in F} \left(\sum_{k \in \mathcal{Z}_i \setminus r} q_k \right) ;$$

en faisant apparaître la partition induite par Q_k sur F , on obtient :

$$K(F) = q_r + \text{Max}_{1 \leq s \leq b_r} \left\{ \text{Max}_{E_i \in F_r^s} \left(\sum_{k \in \mathcal{Z}_i \setminus r} q_k \right) \right\} ;$$

l'expression entre accolades n'est autre que l'expression (1) de $K(F_r^s)$, en accord avec la convention énoncée. \square

Lorsque $q_j = 1 \forall j$, $K(F)$ est la hauteur du questionnaire.

Coût pondéré d'identification

Par définition, ce sera la somme pondérée des coûts des chemins maximaux du questionnaire :

$$C(F) = \sum_{E_i \in F} e_i \left(\sum_{k \in \mathcal{Z}_i} q_k \right) \quad (3)$$

Soit \mathcal{E}_j l'ensemble des indices des éventualités qui nous amènent à utiliser Q_j pour être identifiées :

$\mathcal{E}_j = \{k / \exists (A,j) \text{ tel que } (\{E_k\}, 0) \text{ est descendant de } (A,j)\}$. Le poids de la question Q_j , ou bien la probabilité de poser cette question est défini par $\sum_{k \in \mathcal{E}_j} e_k$; ($\mathcal{E}_j = \emptyset$ si Q_j n'est pas utilisée par le questionnaire). D'où une autre expression du coût d'un questionnaire, somme des coûts pondérés des questions posées :

$$C(F) = \sum_{Q_j \in Q} q_j \left(\sum_{k \in \mathcal{E}_j} e_k \right) \quad (4)$$

Les expressions (3) et (4) sont égales :

A notre questionnaire sur F, nous pouvons faire correspondre de façon unique la matrice binaire B suivante :

$$B = (b_{ij}), \quad j \text{ de } 1 \text{ à Card}(Q) \text{ (lignes)}$$

$$i \text{ de } 1 \text{ à Card}(E) \text{ (colonnes)}$$

et $b_{ij} = (j \in \mathcal{Z}_i)$. Autrement dit, $b_{ij}=1$ si et seulement si on utilise Q_j pour identifier E_i , et $b_{ij}=0$ sinon. ($b_{ik}=0$ si $E_i \notin F$, et $b_{kj}=0$ si Q_j n'est pas utilisée, $\forall k$).

Alors : $\sum_{k \in \mathcal{Z}_i} q_k = \sum_{Q_k \in Q} b_{ik} q_k$, et (3) devient :

$$\sum_{E_i \in F} e_i \left(\sum_{Q_k \in Q} b_{ik} q_k \right) = \sum_{\substack{E_i \in F \\ Q_j \in Q}} e_i b_{ij} q_j ; \text{ soit encore:}$$

$$C(F) = q B e' \quad (5)$$

où e' est le transposé de $e = (e_1, e_2, \dots, e_p)$.

Mais $j \in \mathcal{Z}_i \Leftrightarrow ((\{E_i\}, 0) \text{ est descendant d'un } (A,j)) \Leftrightarrow i \in \mathcal{E}_j$

Donc $b_{ij} = (i \in \mathcal{E}_j)$. L'expression (4) devient à son tour :

$$\sum_{Q_j \in Q} q_j \left(\sum_{E_k \in E} b_{kj} e_k \right) = \sum_{\substack{Q_j \in Q \\ E_i \in E}} q_j b_{ij} e_i = q B e' \quad \square$$

Nous avons trouvé, pour montrer que (3) et (4) sont égales, une nouvelle expression (5) du coût d'un questionnaire. Remarquons que les ensembles \mathcal{Z}_i et \mathcal{E}_j et la matrice B sont liés à la donnée d'un questionnaire particulier sur F.

Le coût moyen d'identification peut se mettre aussi sous forme récursive :

$$C(F) = P(F) q_r + \sum_{1 \leq s \leq b_r} C(F_r^s) \quad (6)$$

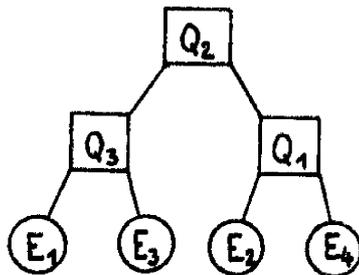
avec $P(F) = \sum_{E_i \in F} e_i$, et la même convention que pour (2) :

Card (A) = 0 ou 1 $\Rightarrow C(A) = 0$.

La preuve que (3) et (6) sont égales est analogue à celle utilisée au § 2 pour montrer (1)=(2).

Les expressions récursives (2) et (6) permettent d'inclure facilement le calcul du coût qu'un questionnaire dans l'algorithme - définition I.4.

Exemples de calcul du coût pour le questionnaire et la table correspondante, ci-dessous :



	E_1 (0,4)	E_2 (0,3)	E_3 (0,2)	E_4 (0,1)
Q_1 (20)	1	1	2	2
Q_2 (10)	1	2	1	2
Q_3 (5)	2	1	1	1

(1) $K(E) = \text{Max}(10+5, 10+5, 10+20, 10+20) = 30$

(2) $K(E) = 10 + \text{Max}(5, 20) = 30$

(3) $C(E) = 0,4 \cdot (10+5) + 0,3 \cdot (10+20) + 0,2 \cdot (10+5) + 0,1 \cdot (10+20) = 21$

(4) $C(E) = 10 \cdot (0,4+0,3+0,2+0,1) + 5 \cdot (0,4+0,2) + 20 \cdot (0,3+0,1) = 21$

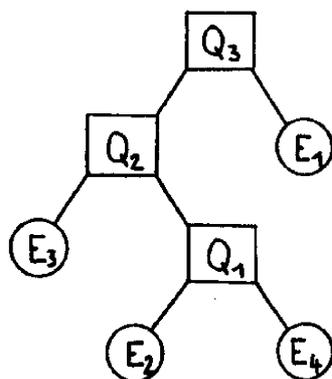
(5) $C(E) = (20 \ 10 \ 5) \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0,4 \\ 0,3 \\ 0,2 \\ 0,1 \end{pmatrix} = 21$

(6) $C(E) = (0,4+0,3+0,2+0,1) \cdot 10 + 3 \cdot 8 = 21$

La table marquée

La matrice B définie ci-dessus caractérise en quelque sorte le questionnaire choisi, et son coût. Elle est de dimension Card (Q) x Card (E), comme la table d'identification définie en I.1.

Dans cette table, nous allons "marquer" la case, colonne i , ligne j , si et seulement si $b_{ij} = 1$, c'est-à-dire si dans le questionnaire considéré, Q_j est utilisée pour identifier E_i . On donne un exemple, figure II.1., d'une questionnaire et de sa table marquée associée.



	E_1 0,4	E_2 0,3	E_3 0,2	E_4 0,1
Q_1 20	1	1	2	2
Q_2 10	1	2	1	2
Q_3 5	2	1	1	1

$$K(E) = 35 ; C(E) = 19$$

Figure II.1.

Appelons 'valeur' d'une case marquée le produit $e_i q_j$. Le coût $C(F)$ est la somme des valeurs des cases marquées. $K(F)$ est le maximum de la somme, par colonne, des valeurs des cases marquées.

La correspondance questionnaire-table marquée est-elle bi-univoque ? Non :

- on ne peut faire correspondre une table marquée à une arborescence si aucune ligne de la table n'est complètement marquée pour $E_i \in F$: la question de la racine est utilisée pour toutes les éventualités,
- à une table marquée peut correspondre deux ou plusieurs questionnaires : exemple figure II.2. ; mais deux questionnaires ayant même table marquée ont même coût.

Comparaison des identifications 'combinatoire' et 'séquentielle'

Nous sommes maintenant en mesure de comparer, par leur coûts, les procédures combinatoires et séquentielles (questionnaires) définies en I.3. Par coût d'une procédure combinatoire sur F , noté $V(F)$, on entend la somme des coûts des questions sélectionnées.

- soit $V^*(F)$ le minimum de $V(F)$ pour toutes les procédures combinatoires possibles sur F . Ordonnons les questions sélectionnées pour

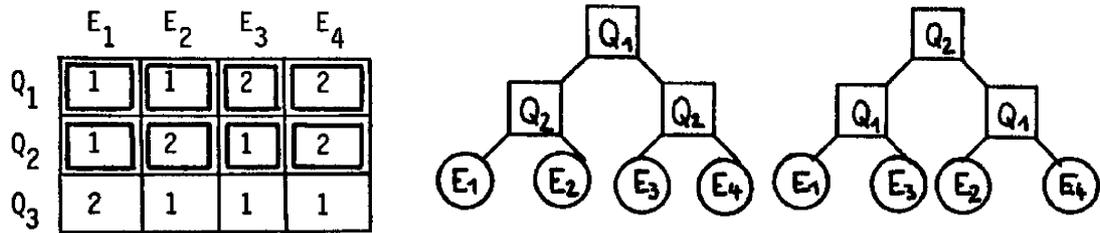


Figure II.2.

$V^*(F)$, solution optimale, de façon à les utiliser en arborescence : on forme un questionnaire dans lequel des questions sélectionnées peuvent ne pas être utilisées, car inutiles ; mais aucun chemin n'a un coût supérieur à $V^*(F)$. Donc, pour ce questionnaire, nous avons : $K(F) \leq V^*(F)$. Appelons $K^*(F)$ le minimum de $K(F)$ pour tous les questionnaires réalisables sur F : $K^*(F) \leq K(F)$; par conséquent $K^*(F) \leq V^*(F)$.

- Pour tout questionnaire :

$$C(F) = \sum_{E_i \in F} e_i \left(\sum_{k \in Z_i} q_k \right) \leq \sum_{E_i \in F} (e_i \cdot \text{Max}_i \sum_{k \in Z_i} q_k) = P(F) \cdot K(F).$$

Considérons un questionnaire pour lequel $K(F) = K^*(F)$; pour celui-ci on a également $C(F) \leq P(F) \cdot K(F)$, et $C(F) \leq P(F) \cdot K^*(F)$. Soit $C^*(F)$ le minimum de $C(F)$ pour tous les questionnaires sur F : on obtient alors

$$C^*(F) \leq P(F) \cdot K^*(F)$$

En résumé : $C^*(F) / P(F) \leq K^*(F) \leq V^*(F)$ (7)

Ces inégalités justifient ce qui a été dit en I.3. : les questionnaires sont une approche plus intéressante, puisqu'ils peuvent fournir un coût moyen et un coût maximum meilleur que celui d'une procédure d'identification combinatoire. Exemple : pour la table de la figure II.3., nous avons :

$$V^*(E) = 35 ; K^*(E) = 25 ; C^*(E)/P(E) = 22,5$$

	E_1	E_2	E_3	E_4
	(1)	(1)	(1)	(1)
Q_1 (20)	1	1	2	2
Q_2 (10)	1	2	2	2
Q_3 (5)	1	1	1	2

Figure II.3.

Diverses propriétés des coûts

- Soit un questionnaire \mathcal{Q} donné sur E, pour un problème (E,Q,e,q), où e et q peuvent varier. Dans ce paragraphe, on notera les coûts de ce questionnaire \mathcal{Q} par K(q) et C(e,q).

De (1), on déduit immédiatement :

$$K(\lambda q) = \lambda K(q) \quad (8) \quad \text{et} \quad K(q+p) \leq K(q) + K(p) \quad (9)$$

$$\text{De (5) :} \quad C(\lambda e, \mu q) = \lambda \mu C(e, q) \quad (10)$$

$$\left. \begin{aligned} C(e+d, q) &= C(e, q) + C(d, q) \\ C(e, q+p) &= C(e, q) + C(e, p) \end{aligned} \right\} \quad (11)$$

$$\text{De (3) :} \quad \frac{\partial C}{\partial e_i} = \sum_{k \in \mathcal{Z}_i} q_k \quad (12) \quad \text{et} \quad \frac{\partial C}{\partial q_i} = \sum_{k \in \mathcal{E}_i} e_k \quad (13)$$

- K et C sont des fonctions continues des q et e ; leurs minima sur tous les questionnaires pour le problème sont également des fonctions continues de q et e.

Notons $K^*(q) = \min_{\mathcal{Q}} K(q)$ et $C^*(e, q) = \min_{\mathcal{Q}} C(e, q)$; que devient $C^*(e, q)$ lorsque e et q subissent les variations Δe et Δq ? On s'intéresse à la quantité $\Delta C^* = C^*(e + \Delta e, q + \Delta q) - C^*(e, q)$. On fera l'hypothèse : $\sum_{E_i \in E} e_i = \sum_{E_i \in E} (e_i + \Delta e_i) = P(E)$, ce qui est logique puisqu'on compare ainsi deux coûts pour une même pondération totale (Si tel n'est pas le cas, on s'y ramène aisément). M.R. GAREY (1972 b) a démontré la formule suivante :

$$|\Delta C^*| \leq \left(\sum_{Q_j \in Q} \min(q_j, q_j + \Delta q_j) \cdot \sum_{E_i \in E} |\Delta e_i| + P(E) \cdot \sum_{Q_j \in Q} |\Delta q_j| \right) \quad (14)$$

Pour la démonstration, il suffit de considérer le cas $C^*(e, q) \leq C^*(e + \Delta e, q + \Delta q)$, à cause de la symétrie de (14). A $C^*(e, q)$ correspond un certain questionnaire optimal, de matrice associée B ; soit C' le coût du même questionnaire, mais utilisé pour e + Δe , q + Δq : $C' = (q + \Delta q) B (e + \Delta e)'$. Nous avons : $C^*(e, q) \leq C^*(e + \Delta e, q + \Delta q) \leq C'$; donc : $|\Delta C^*| \leq |C^*(e, q) - C'|$; $C^*(e, q)$ et C' sont des coûts pour un même questionnaire ; il suffit donc de montrer que C(e,q) ne subit pas de variation, ΔC , en valeur absolue, supérieure à l'expression (14), pour un même questionnaire :

Dans la table marquée, la valeur de la case (i,j) marquée, subit la variation :

$$\begin{aligned}\Delta C_{ij} &= (e_i + \Delta e_i) (q_j + \Delta q_j) - e_i q_j \\ &= \Delta e_i (q_j + \Delta q_j) + \Delta q_j e_i = \Delta e_i \cdot q_j + \Delta q_j \cdot (e_i + \Delta e_i) \\ \text{d'où: } |\Delta C_{ij}| &\leq |\Delta e_i| \cdot (q_j + \Delta q_j) + |\Delta q_j| \cdot e_i \\ &\leq |\Delta e_i| \cdot q_j + |\Delta q_j| \cdot (e_i + \Delta e_i)\end{aligned}$$

En sommant pour i de 1 à $\text{Card}(E)$, en utilisant l'hypothèse $\sum e_i = \sum (e_i + \Delta e_i) = P(E)$, puis pour j de 1 à $\text{Card}(Q)$, on trouve :

$$\sum_{i,j} |\Delta C_{ij}| \leq \left(\sum_{Q_j \in Q} \min(q_j, q_j + \Delta q_j) \cdot \sum_{E_i \in E} |\Delta e_i| + P(E) \cdot \sum_{Q_j \in Q} |\Delta q_j| \right)$$

Mais $|\Delta C| \leq \sum |\Delta C_{ij}|$, car on ne change pas le sens de l'inégalité en ajoutant les $|\Delta C_{ij}|$ pour les cases non marquées. \square

- On trouve aisément une formule du même type pour $K(e, q)$, le coût maximum :

$$\text{par (9) : } K(q + \Delta q) - K(q) \leq K(\Delta q) = \max_{E_i \in E} \sum_{Q_j \in Q} b_{ij} \Delta q_j$$

$$\text{or } \max_{E_i \in E} \sum_{Q_j \in Q} b_{ij} \Delta q_j \leq \sum_{Q_j \in Q} |\Delta q_j|$$

puis par un raisonnement analogue au précédent, on prouve :

$$\boxed{|\Delta K^*| \leq \sum_{Q_j \in Q} |\Delta q_j|} \quad (15)$$

11.2. RECHERCHE DE BORNES INFÉRIEURES DU CÔUT PONDÉRE

La connaissance d'une borne inférieure ou du coût d'un questionnaire ainsi qu'une solution obtenue par exemple par un procédé heuristique, de coût C , fournit un intervalle dans lequel se situe le coût optimal : $m \leq C^* \leq C$; d'autre part, la connaissance d'une borne inférieure permettra d'orienter la construction d'un questionnaire optimal (voir § V.1.). Il existe plusieurs méthodes d'obtention de bornes inférieures, et souvent, les meilleures bornes (c'est-à-dire les plus proches de C^*) seront aussi les plus difficiles à obtenir en général.

Méthode des "cases essentielles"

On cherche s'il n'y a pas, dans la table d'identification, des cases qui seront obligatoirement marquées pour tout questionnaire, nommées "cases essentielles".

Si, pour un couple d'éventualités E_i et E_j , il n'existe qu'une seule question Q_k pouvant les séparer, alors les cases (i,k) et (j,k) sont "essentiels". La somme des valeurs des cases essentielles fournit une borne inférieure du coût pondéré. Cette évaluation peut être encore améliorée, sachant qu'au moins une ligne de la table sera entièrement marquée.

	E_1 (0,4)	E_2 (0,3)	E_3 (0,2)	E_4 (0,1)	
Q_1 (20)	1	1	2	2	→ 8 ou 20
Q_2 (10)	1	2	1	2	→ 3 ou 10
Q_3 (5)	2	1	1	1	→ 0 ou 5

Figure II.4.

Exemple : sur la figure II.4., les cases essentielles sont cerclées. Elles correspondent, dans la figure I.3., aux colonnes ne possédant qu'un seul '1'. En faisant la somme de la valeur de ces cases, on obtient $m = 11$. Mais :

si Q_1 est la racine, le coût est au minimum $20 + 3 + 0 = 23$
 si Q_2 " " " " " " " " " $8 + 10 + 0 = 18$
 si Q_3 " " " " " " " " " $8 + 3 + 5 = 16$

une meilleure borne inférieure est donc $m = 16$.

Méthode du "coût de preuve minimal"

Cette méthode est exposée dans S.E.M.A.(1970) et TABOURIER (1971). L'expression (3) peut se mettre sous la forme $C = \sum_{E_i \in E} e_i c_i$, où c_i apparaît comme le coût de séparer l'éventualité E_i des autres; ce coût a un minimum r_i , pour l'ensemble des questionnaires, et correspond à la séquence de questions de coût minimal, permettant d'isoler E_i : c'est

Le "coût de preuve minimal" de E_i . Une borne inférieure du coût d'un questionnaire est alors $\sum_{E_i \in E} c_i r_i$.

La détermination de r_i est un problème de recouvrement analogue à celui de l'identification combinatoire présenté § I.3. : une fois dressée la table de couverture (exemple, figure I.3.), on doit couvrir au moindre coût les colonnes étiquetées par E_i .

Un algorithme efficace est présenté dans S.E.M.A. (1970, note n°4). Exemple : problème figure II.4. : on trouve $r_1=5$ (Q_3), $r_2=25$ (Q_1 et Q_3), $r_3=15$ (Q_2 et Q_3), $r_4=30$ (Q_1 et Q_2)

d'où $m = 5.0,4 + 25.0,3 + 15.0,2 + 30.0,1 = 15,5$.

Remarquons qu'on retrouve dans cette borne au moins la somme des valeurs des cases essentielles, puisqu'on utilise au moins les questions obligatoires correspondant à ces cases (questions soulignées dans l'exemple précédent).

On pourra également faire une étape supplémentaire en supposant la première question déjà posée et en reportant l'évaluation sur les sous-familles séparées ; on prend le minimum obtenu pour chaque question prise comme racine ; toutefois, l'amélioration obtenue ne compense pas en général l'effort de calcul requis.

Lorsque les questions sont peu nombreuses, nous aurons beaucoup de cases essentielles, et la première méthode s'avère avantageuse car rapide. Au contraire, s'il y a beaucoup de questions disponibles, et peu de cases essentielles, on aura recours à la méthode du "coût de preuve minimal" pour obtenir une borne intéressante.

Méthode du "questionnaire libre"

Supposons que l'on connaisse à priori l'ensemble des coûts des questions effectivement utilisés pour un questionnaire optimal, soit $c=(c_1, c_2, \dots, c_m)$, c étant une famille extraite de $q=(q_1, q_2, \dots, q_n)$, ainsi que l'ensemble des bases réelles de ces questions : $a=(a_1, a_2, \dots, a_m)$, vérifiant l'égalité I(P6).

Un questionnaire libre, c'est-à-dire autorisant toute partition pour les questions, utilisant les deux ensembles c et a , s'il est optimal, fournit une borne inférieure du coût d'un questionnaire réalisable pour le problème initial.

Or, ce questionnaire libre de coût minimal se construit aisément par une extension de l'algorithme de HUFFMANN exposée dans le livre de C.F. PICARD (1972) ; rappelons-le ici :

Algorithme de HUFFMANN étendu, pour la construction d'un questionnaire libre optimal sur les ensembles (e,c,a) :

- 1) construire un questionnaire de HUFFMANN sur (e,a) , c'est-à-dire :
 - a) renommer les e_i et a_i de façon que :
$$e_1 \geq e_2 \geq \dots \geq e_p ; a_1 \geq a_2 \geq \dots \geq a_m$$
 - b) sélectionner les a_m poids les plus faibles, en faire la somme σ , supprimer de la liste les poids sélectionnés, y ajouter σ enfin supprimer a_m de la liste des bases .
 - c) si la liste des bases est non vide, aller en a), sinon :
 - d) construire l'arbre correspondant aux regroupements de poids effectués, depuis les noeuds terminaux jusqu'à la racine : c'est le questionnaire de HUFFMANN cherché. A chaque sommet non terminal, affecter un poids, somme des poids des sommets-fils.
- 2) affecter aux sommets non terminaux, ordonnés en poids non croissants, les coûts c_i en ordre non décroissant.

Le coût pondéré de ce questionnaire libre optimal (somme des coûts pondérés des sommets intérieurs) est une borne inférieure du coût d'un questionnaire réalisable sur (E,Q,e,q) .

Malheureusement, on ne connaît pas a priori les ensembles a et c tirés de Q , bases réelles et coûts employés dans un questionnaire réalisable optimal. Tout au plus peut-on savoir que l'ensemble a vérifie I-(P6) et que les questions correspondant à des cases essentielles seront utilisées dans le questionnaire réalisable optimal.

Des difficultés surgissent du fait qu'une même question de Q peut être utilisée plusieurs fois. En particulier, si toutes les bases valent 2, on ne peut pas simplement choisir pour c les $n-1$ coûts les plus faibles. De même, une question peut être employée dans le questionnaire avec une base réelle plus faible qu'au départ. Notons qu'une question élémentaire n'est utilisée qu'une seule fois, et plus généralement, la questions Q_i est utilisée au plus $\min \text{Card}(\mathcal{G}_i^S)$ fois. Dans le cas des

des questions toutes dichotomiques ces remarques permettent de sélectionner une famille c' avec $c'_i \leq c_i \quad \forall i$.

Exemple (figure II.4.) : $n=4$; toutes bases 2 donc $a=(2,2,2)$; $Q_3(5)$ est élémentaire, et les autres questions sont nécessaires : $c'=(5,10,20)$. Le questionnaire de HUFFMANN construit sur e a des sommets non terminaux de poids : 1 0,6 0,3 , qu'on affecte respectivement des coûts 5 10 20 ; d'où la borne inférieure $m=17$.

Utilisation de la théorie de l'information

Nous voulons dans cette méthode purement algébrique éviter de construire le questionnaire de HUFFMANN, support de l'évaluation précédente. Nous nous restreindrons à un ensemble de questions dichotomiques (base 2) et supposons $\sum_{E_i \in E} e_i = 1$. On notera C^* le coût minimal d'un questionnaire réalisable sur E pour le problème (E, Q, e, q) ; on appellera longueur d'un questionnaire son coût lorsque toutes les questions sont de coût 1. Soit L^* la longueur minimale pour le même problème. Enfin, on note $I(e) = \sum_{E_i \in E} e_i \log_2 \frac{1}{e_i}$ l'information au sens de SHANNON, contenue dans la distribution e .

On sait que $I(e)$ est une borne inférieure de la longueur d'un questionnaire construit librement sur E , c'est-à-dire sans contrainte sur les partitions (voir C.F. PICARD 1972). En particulier : $I(e) \leq L^*$ (16)

Lorsque toutes les questions sont de coût x , le coût optimal sera $x L^*$. Comparons $x L^*$ et C^* grâce à l'inégalité (14) : seuls les coûts sont modifiés et on obtient :

$$|C^* - x L^*| \leq \sum_{Q_j \in Q} |q_j - x| \tag{17}$$

posons $\sum_{Q_j \in Q} |q_j - x| = U(x)$; en utilisant (16) et (17) on obtient une borne inférieure pour C^* :

$$x I(e) - U(x) \leq C^* \tag{18}$$

mais qui dépend de x . Cherchons le maximum de $x I(e) - U(x)$ pour obtenir la meilleure borne inférieure :

- Ordonnons les q_i pour que $q_1 \leq q_2 \leq \dots \leq q_n$. La fonction $U(x)$ est représentée par une ligne polygonale convexe avec un sommet pour chaque $x=q_i$ (voir la figure II.5. qui correspond à l'exemple figure II.4.). Pour $x < q_1$, la pente vaut $-n$, pour $x > q_n$ elle vaut $+n$,

- entre q_i et q_{i+1} (sauf si confondus) la pente est égale à $-n+2i$.
- La fonction $x I(e) - U(x)$ possède bien un maximum : pour séparer E il faut au moins $\lceil \log_2 \text{Card}(E) \rceil$ questions dans Q . D'autre part, on sait que $I(e) \leq \log_2 \text{Card}(E)$, donc $n \geq I(e)$. Pour $x \rightarrow +\infty$ $x I(e) - U(x) \rightarrow -\infty$; de même pour $x \rightarrow -\infty$;
 - La fonction $x I(e) - U(x)$ étant polygonale concave, le maximum est atteint en un point $x=q_k$ (ou sur un intervalle q_k, q_{k+1} dans des cas particuliers, mais cela ne modifie pas le raisonnement) que l'on va déterminer :

$$\forall x : x I(e) - U(x) \leq q_k I(e) - U(q_k) .$$

donc si $x \rightarrow q_k$ à droite : $I(e) \leq \frac{U(x) - U(q_k)}{x - q_k} = -n + 2k$

et si $x \rightarrow q_k$ à gauche : $I(e) \geq \frac{U(x) - U(q_k)}{x - q_k} = -n + 2(k-1)$

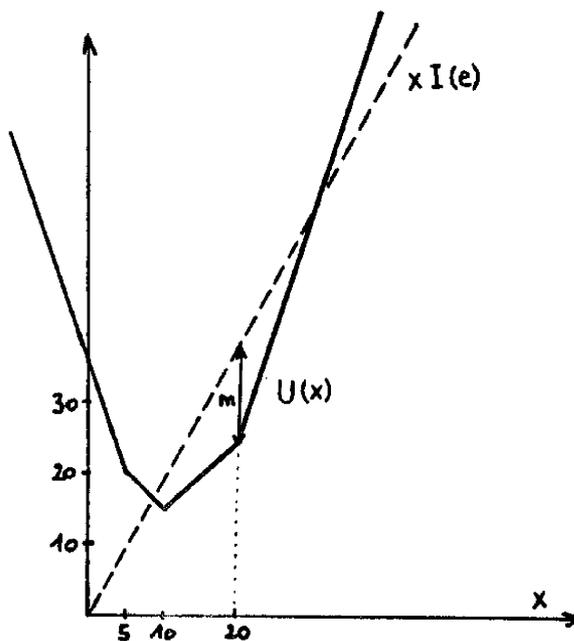


Figure II.5.

En conclusion, une borne inférieure est $q_k I(e) - U(q_k)$ pour k vérifiant $-n + 2(k-1) \leq I(e) \leq -n + 2k$.

Cette borne est généralement moins bonne que les précédentes, mais elle est beaucoup plus facile à calculer car elle ne fait intervenir que des opérations algébriques.

Exemple (figure II.4.) : $I(e) = 1,846$; $n=3$; $k=3$;
 $q_k I(e) - U(q_k) = 11,92$; comme le coût est un multiple de 0,5 on peut
 affirmer $m=12$.

Une méthode intermédiaire est obtenue en remplaçant $I(e)$
 par la longueur du questionnaire de HUFFMANN vu au paragraphe précédent.
 Pour le même exemple : $L_H(e) = 1,9$; $m=13$.

Pour les problèmes à questions de bases supérieures à 2,
 on n'a rien trouvé d'intéressant qui puisse dispenser de construire un
 support, à part de prendre les logarithmes dans la base maximum. On rejoint
 les problèmes d'"information hétérogène" et "acquisition" discutés dans
 l'ouvrage de C.F. PICARD.

Cette méthode ne donne de bons résultats que si les coûts
 sont relativement homogènes ; sinon $U(q_k)$ devient trop grand.

Remarquons qu'une borne inférieure du coût moyen en est
 une aussi pour le coût maximum K (II.(7)). On vérifiera qu'une borne infé-
 rieure pour K peut être obtenue :

- en prenant le maximum des coûts de preuves minimaux pour chaque
 éventualité,
- en remplaçant $I(e)$ par $\lceil \log_2 \text{Card}(E) \rceil$, dans la dernière méthode :
 c'est une borne inférieure du nombre maximum de questions nécessaires
 pour une identification dichotomique.

La table suivante donne la valeur des bornes inférieures
 du coût pondéré obtenues par ces différentes méthodes, pour notre exemple,
 pour un exemple de M.R. GAREY (1972 b), et enfin pour un problème réel
 d'identification d'incident (plan d'inspection) rencontré en fabrication
 industrielle.

	Exemple (fig.II.4.)	M.R.GAREY (fig.II.6)	Problème réel (fig.II.7.)
Cases essentielles	11	27,5	50,475
" amélioré	16	31	54,475
Coût de preuve mini.	15,5	27,5	50,775
Quest. libre	17	37	26,825
Th.de l'information	12	24,5	négatif
Coût optimal	19	42	75,425

Dans l'exemple de M.R. GAREY (figure II.6.), toutes les lignes comportent des cases essentielles.

Les questions sont dichotomiques et $p=n-1$.; pour le questionnaire libre, on peut donc prendre $c=c'$.

Pour le problème réel (figure II.7.), le travail peut commencer après suppression des questions 11 (qu'on peut remplacer par 3 et 4) et 14 (à cause de 3 et 13) qui sont redondantes (voir V.2.).

		E ₁ 0,2	E ₂ 0,2	E ₃ 0,1	E ₄ 0,1	E ₅ 0,2	E ₆ 0,1	E ₇ 0,1
Q ₁	25	2	1	(2)	(1)	2	1	1
Q ₂	15	1	(2)	1	(1)	1	2	2
Q ₃	5	(2)	1	(1)	1	1	1	1
Q ₄	20	1	1	(1)	1	(2)	1	1
Q ₅	20	1	(1)	1	1	1	(2)	1
Q ₆	15	1	(1)	1	1	1	1	(2)

Figure II.6.

RTC - PAS

NOMBRE DE CAUSES = 13
NOMBRE DE TESTS = 14

MATRICE DES RELATIONS TESTS-CAUSES										COUTS		
2	(1)	2	2	2	2	2	2	2	2	2	1	5
(1)	2	2	2	2	2	2	2	2	2	2	2	5
2	2	2	2	2	2	2	2	(2)	(1)	1	2	5
2	2	2	2	2	2	2	2	2	2	1	2	5
2	2	(1)	2	(2)	2	2	2	2	2	2	2	5
2	2	2	(1)	2	(2)	2	2	2	2	2	2	5
2	2	2	2	(1)	2	2	2	2	2	2	2	5
2	2	2	2	2	(2)	2	2	(1)	2	2	2	5
2	2	2	2	2	2	2	2	(2)	2	2	(1)	5
2	2	2	2	2	2	2	2	2	2	1	2	10
2	2	2	2	2	2	2	2	2	2	2	2	10
2	2	2	2	2	(2)	(1)	2	2	2	2	2	12C
1	1	2	2	2	(2)	2	2	(1)	1	2	1	26C
2	2	1	1	1	1	1	1	1	2	3	2	246C

POIDS
50 150 10 95 10 10 190 95 190 50
25 25 100

Figure II.7.

CHAPITRE III.

-:-

CONSTRUCTIONS DE QUESTIONNAIRES : OUTILS DE REPRÉSENTATION, ET PROBLÈMES ÉQUIVALENTS

-

On aborde dans ce chapitre la construction des questionnaires, éventuellement optimaux, en recherchant de nouvelles représentations du problème. On vise deux buts :

- a) dégager des outils de représentation de l'univers des solutions, en vue de la résolution pratique décrite en seconde partie de ce mémoire,
- b) vérifier, par des tentatives de "réductions", si le problème n'est pas déjà connu sous une autre forme, qui en rendrait plus aisée la résolution.

III.1. REPRÉSENTATIONS DIRECTES

Arbre des solutions

Soit un problème (E, Q, e, q) dont toutes les éventualités sont séparables. Choisissons dans un questionnaire sur E , des sommets sur des chemins distincts, à partir desquels nous supprimons les descendants véritables de ces sommets.

Le sous-graphe obtenu est appelé questionnaire inachevé sur E . Il représente une étape dans la construction d'un questionnaire d'amont en aval, lorsque toutes les éventualités ne sont pas encore séparées.

La façon la plus naturelle de représenter l'univers des solutions possibles, pour la construction d'un questionnaire, consiste à le développer sous forme d'arbre. Les sommets non terminaux de cet arbre représentent des questionnaires inachevés ; la racine correspond à l'état original du problème, lorsqu'aucun choix n'a encore été fait ; un questionnaire complet est figuré par un sommet terminal ; le chemin aboutissant à ce sommet représente les différentes étapes dans la construction du

questionnaire, c'est-à-dire les choix successifs des questions. Un début d'arbre est donné figure III.1., pour la table I.1.

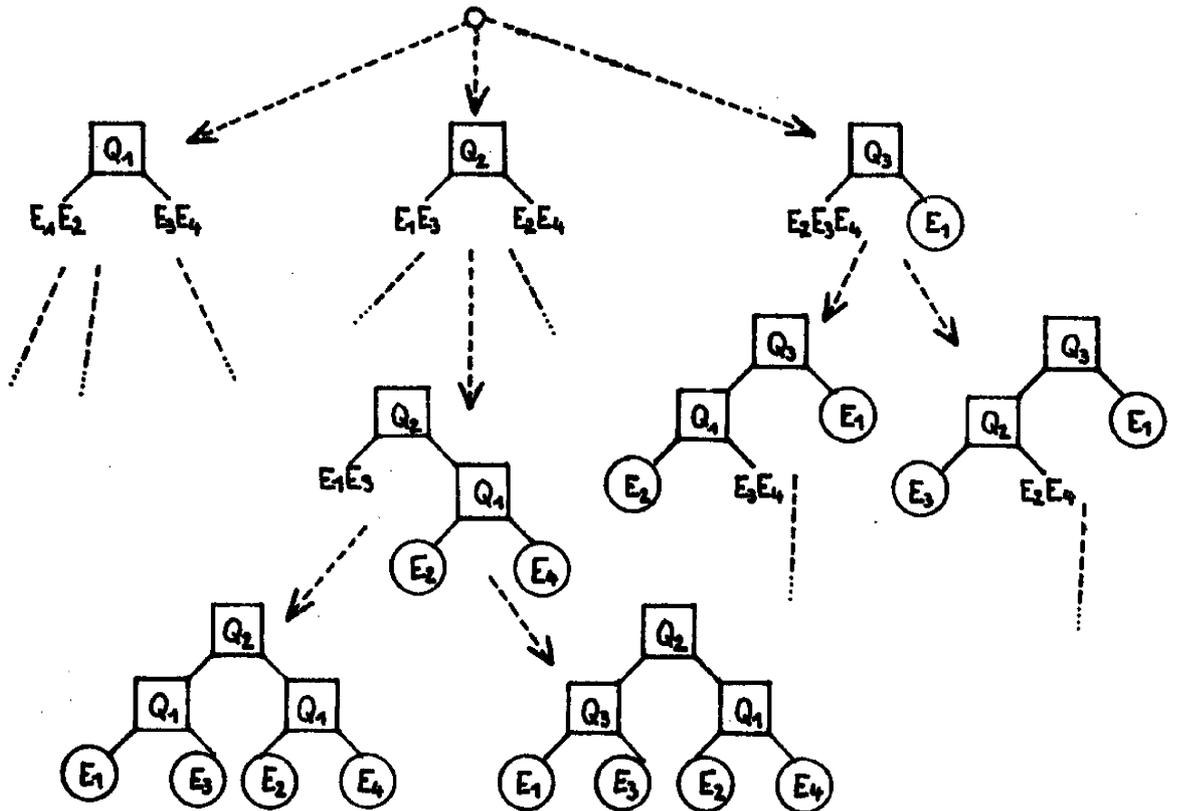


Figure III.1.

Graphe ET/OU des solutions

Dans la construction de l'arbre des solutions, on rencontre deux types d'expansion des sommets pendants, correspondant en fait aux deux sortes de choix distincts à opérer pendant la construction d'un questionnaire :

- a) le choix de l'issue à développer : l'algorithme - définition impose l'examen des issues dans l'ordre indiciel, mais il est évident que l'ordre est sans importance, pourvu que toutes les issues soient explorées. Le choix est du type ET.
- b) le choix de la question à employer pour séparer un sous-ensemble d'éventualités, du type OU (exclusif).

Construisons le graphe orienté suivant, possédant deux types de sommets :

- a) des sommets ET figurant les questions posées, ayant pour successeurs immédiats des sommets associés aux sous-ensembles d'éventualités séparés,
- b) des sommets OU représentant des sous-ensembles d'éventualités à séparer, et dont les successeurs immédiats supportent les différentes questions qu'il est possible de poser.

Un chemin dans ce graphe est formé d'alternances de sommets ET de sommets OU. Le graphe entier correspond à l'ensemble de toutes les exécutions possibles de notre algorithme - définition.

On extrait un questionnaire de ce graphe entièrement développé, en sélectionnant de proche en proche, depuis la racine jusqu'aux feuilles, soit un seul arc incident extérieurement pour les sommets OU, soit tous les arcs pour un sommet ET. Cette extraction est directement liée à une exécution particulière de notre algorithme - définition.

Une économie supplémentaire est réalisée en n'écrivant qu'une seule fois dans le graphe les sous-ensembles distincts d'éventualités. Un sommet OU peut avoir plusieurs arcs incidents intérieurement. Un sommet ET n'a toujours qu'un seul arc incident intérieurement, et une solution-questionnaire extraite a toujours une forme d'arbre.

La figure III.2. représente le graphe ET/OU des solutions pour l'exemple I.1. Les arcs sont implicitement orientés de haut en bas.

Les sommets OU, de forme ovale, contiennent les sous-ensembles d'éventualités à séparer sous forme d'une liste d'indices. Les sommets ET sont carrés et supportent l'indice de la question utilisée.

Le petit arc de cercle reliant les arcs incidents extérieurement aux sommets ET rappelle que ces arcs doivent être sélectionnés ensemble si le sommet l'est. On n'a pas besoin de représenter les sommets OU contenant une seule éventualité.

Les chiffres placés près des sommets permettent le calcul d'un questionnaire de coût pondéré optimal. Explicitons ce calcul en anticipant sur sa justification, apportée au chapitre V par la formule V-(1) du coût pondéré optimal ; on procède d'aval en amont :

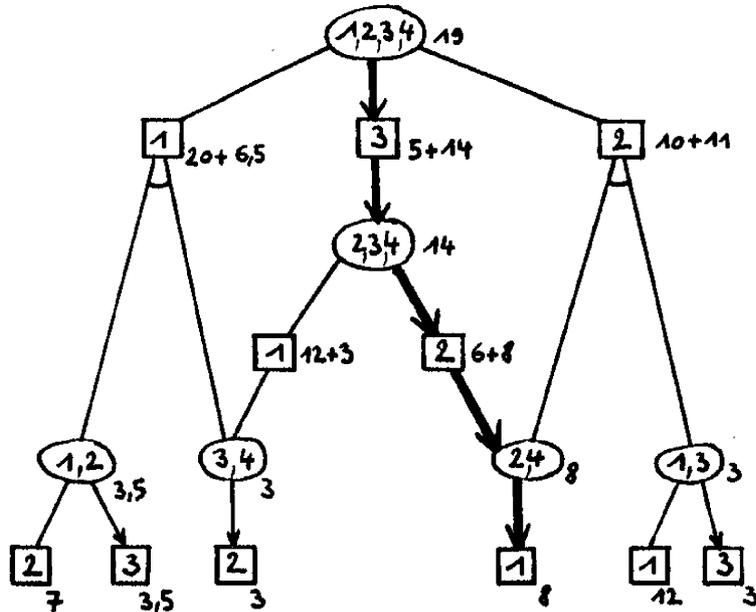


Figure III.2.

- à un sommet ET terminal est associé le coût de la question, pondéré par le poids des éventualités séparées,
- lorsque tous les successeurs d'un sommet OU ont été évalués, l'arc aboutissant au sommet de valuation minimum est marqué (ici par une flèche) et on reporte cette valuation sur le sommet OU père,
- pour un sommet ET, non terminal, dont tous les successeurs ont été évalués, on fait la somme du coût pondéré de la question et des valuations attachées à tous ses successeurs immédiats,
- on propage cette valuation jusqu'à la racine du graphe.

Soit F un sous-ensemble d'éventualités, associé à un sommet OU. En suivant les arcs marqués d'une flèche, depuis ce sommet jusqu'aux sommets terminaux, on met en évidence un questionnaire de coût minimal sur F . En particulier, en partant de la racine, on isole un questionnaire optimal sur E . (en traits renforcés, figure III.2.).

Le même algorithme peut servir à trouver un questionnaire à coût maximal optimal, si on remplace dans le calcul précédent "coût pondéré d'une question" par "coût de la question" et "somme" par "maximum".

Cette représentation de l'espace des solutions sous forme de graphe ET/OU semble la plus économique, car elle ne contient aucune redondance, contrairement à la représentation sous forme d'arbre (paragra-

phe précédent), et à la représentation sous forme de chemin dans un graphe que nous verrons tout à l'heure.

Nous avons vu au paragraphe I.4., que l'algorithme - définition des questionnaires réalisables pouvait être considéré comme non-déterministe, du fait de la fonction CHOIX non précisée. La liaison entre algorithme non-déterministe et graphe ET/OU nous a été suggérée par l'ouvrage de N.J. NILSSON (1971). Bien qu'écrit dans un contexte différent (intelligence artificielle), on y trouvera des informations sur l'utilisation et la résolution des graphes ET/OU.

III.2. PROBLEMES EQUIVALENTS

Chemin (minimal) dans un graphe de partitions

Soient $(A,i), (B,j), \dots (G,l)$ l'ensemble des sommets terminaux d'un questionnaire inachevé. $\{A, B, \dots, G\}$ est une partition de E , car $\forall E_i \in E$, par construction E_i appartient à un et un seul des sous-ensembles précédents.

Construisons le graphe valué suivant :

- 1) l'ensemble de ses sommets est l'ensemble des partitions qu'il est possible d'effectuer sur E avec des questionnaires inachevés, plus les sommets $\alpha = \{ \{E\} \}$ et $\omega = \{ \{E_1\}, \{E_2\}, \dots, \{E_p\} \}$.
- 2) il y a un arc de λ à μ si on peut passer de la partition λ à la partition μ en développant un seul sommet d'un questionnaire inachevé effectuant λ .
- 3) soit (A,i) le sommet développé pour passer de la partition λ à la partition μ . L'arc (λ, μ) est valué par la quantité $v = q_i \sum_{E_k \in A} e_k$

Ce graphe possède les propriétés suivantes :

- un seul sommet n'a pas d'antécédent : α ; un seul sommet n'a pas de successeur : ω ;
- en associant à chaque sommet le nombre de blocs de la partition correspondante, on définit sur le graphe une fonction ordinale. (par exemple, KAUFMANN A. 1968, chap. 3);
- ce graphe est sans circuit, et tout chemin de α à ω peut être associé à la construction, question par question, d'un questionnaire

Recouvrement disjoint d'hypercube

L'ensemble $\{Q_1(E_k), Q_2(E_k), \dots, Q_n(E_k)\}$ des issues des questions obtenues pour une éventualité E_k (kième colonne de la table d'identification), constitue les coordonnées d'un point dans un hypercube à $\text{Card}(Q)=n$ dimensions : $H = \prod_{Q_i \in Q} \{1, 2, \dots, b_i\}$.

Supposons construit un questionnaire sur E , entièrement séparable. Au chemin identifiant E_k , associons dans l'hypercube l'ensemble F_k des points de coordonnées (x_1, x_2, \dots, x_n) qui vérifient $x_j = Q_j(E_k)$ pour toute question Q_j utilisée sur ce chemin. ($F_k = \{(x_1, x_2, \dots, x_n) \in H / j \in \mathcal{I}_k \Rightarrow x_j = Q_j(E_k)\}$).

D'après cette définition, F_k est une place de l'hypercube, à laquelle appartient le point de coordonnées $x_j = Q_j(E_k)$, caractéristique de E_k .

Supposons tout d'abord les questions dichotomiques, nous verrons ensuite comment généraliser. Montrons la propriété :

P7 : à tout questionnaire dichotomique séparant E complètement, correspond un recouvrement de l'hypercube $H = \{1, 2\}^n$ par autant de faces disjointes que d'éventualités:

Soit $x = (x_1, x_2, \dots, x_n)$ un point de H . L'ensemble des x_j considéré comme issues des questions Q_j , permet de suivre dans le questionnaire réalisé, un chemin et un seul aboutissant à une éventualité E_k . Donc $x \in F_k$; supposons d'autre part que $x \in F_\ell$, $\ell \neq k$; alors le chemin suivi devrait aboutir à $E_\ell \neq E_k$: c'est impossible. Tout point de H appartient donc à une face F_k et une seule \square

Lorsqu'il se trouve dans Q des questions de base supérieure à 2, ces questions peuvent être utilisées dans le questionnaire avec une base réelle inférieure à la base d'origine : une telle question aura alors une issue ou plusieurs non utilisées.

Soit m le nombre de sommets du questionnaire. Le nombre de sommets terminaux, plus le nombre d'issues non utilisées est $\sum_{\substack{(A,i) \in Q \\ i \neq 0}} b_i - m + 1$. En tenant compte des faces d'hypercube qui correspondent à des issues non utilisées, on peut énoncer la propriété plus générale :

P'7 : à tout questionnaire \mathcal{Q} séparant E complètement, correspond un recouvrement de l'hypercube $H = \prod_{Q_i \in \mathcal{Q}} \{1, 2, \dots, b_i\}$ par $\sum_{\substack{(A,i) \in \mathcal{Q} \\ i \neq 0}} (b_i - 1) + 1$ faces disjointes.

	E_1	E_2	E_3	E_4
Q_1	1	1	2	2
Q_2	1	2	1	2
Q_3	1	2	2	2

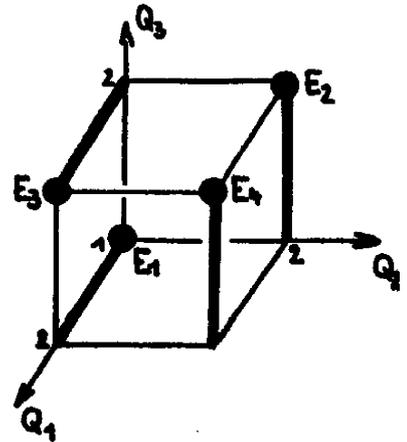
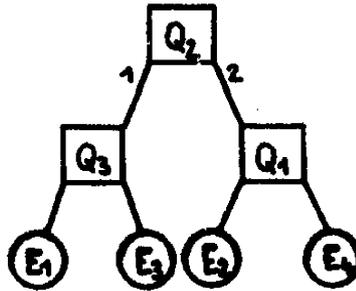


Figure III.4.

La figure III.4. représente un questionnaire réalisé pour la table indiquée, et le recouvrement, traits forts, d'hypercube associé.

La réciproque des propriétés P7 et P'7 est fautive : on voit par exemple figure III.5. un recouvrement d'hypercube par des faces disjointes ne pouvant correspondre à aucun questionnaire.

Remarquons que la recherche d'une procédure d'identification combinatoire (I.3.) n'est autre que la recherche d'un sous-espace de l'hypercube H tel que les projections des points - éventualités sur ce sous espace soient distinctes.

Codage instantané

Revenons à des questions exclusivement dichotomiques. Soit $r(E_k)$ le rang de E_k dans le questionnaire, égal au nombre de questions sur ce chemin. La face F_k possède $r(E_k)$ coordonnées spécifiées et comporte donc $2^{n-r(E_k)}$ points. De P7 on tire alors :

$$\sum_{E_k \in E} 2^{n-r(E_k)} = 2^n, \text{ soit } \sum_{E_k \in E} 2^{-r(E_k)} = 1 \tag{1}$$

On retrouve, comme dans l'ouvrage de C.F. PICARD (1972, tome 2) sur les questionnaires libres, la limite supérieure de l'inégalité de KRAFT pour les longueurs des mots d'un code instantané. Un question-

naire est effectivement équivalent à un codage instantané de $\text{Card}(E)$ mots dans l'alphabet constitué par les issues des questions utilisées.

Pour des bases supérieures à 2, on peut généraliser (1) en:

$$\sum_{E_k \in E} \prod_{i \in Z_k} \frac{1}{b_i} \leq 1 \quad (2)$$

l'égalité ayant lieu : si on remplace les b_i par les bases réelles employées, ou si on étend la sommation aux chemins aboutissant aux issues non utilisées.

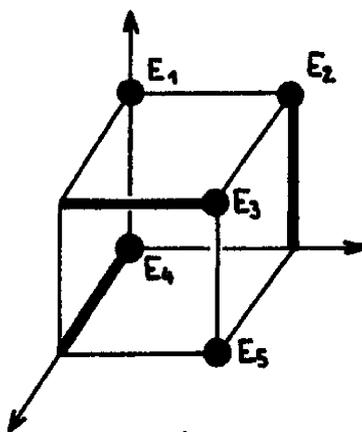


Figure III.5.

Programme linéaire en variable bivalente

Dans le cas des problèmes à questions exclusivement dichotomiques, la contrainte exprimée par la propriété P7 peut se mettre sous forme d'égalité matricielle $MX=U$:

- soient $F_1, F_2, \dots, F_i, \dots, F_k$ l'ensemble des faces de l'hypercube H qui contiennent un point-éventualité et un seul,
- soient h_1, h_2, \dots, h_d , avec $d=2^n$ la liste des points de l'hypercube,
- on définit la matrice M par $m_{ij}=1$ si $h_j \in F_i$
 $=0$ sinon
- X est un vecteur (x_1, x_2, \dots, x_k) caractérisant un questionnaire réalisable, défini par : $x_i = 1$ si F_i fait partie de recouvrement correspondant au questionnaire.

La contrainte " h_j est recouvert une fois et une seule par une face" se traduit par $\sum_{i=1}^k m_{ij} x_i = 1$, et pour tous les points : $MX=U$ où U est un vecteur de longueur d, d'éléments unités.

- à toute face F_j , associons les coûts k_j et c_j , éléments du coût maximal et du coût pondéré, calculés grâce aux valeurs dans la colonne correspondante dans la table. Le problème de trouver un questionnaire de coût minimum s'énonce :

$$\left\{ \begin{array}{l} \text{minimiser } \max_{1 \leq j \leq \ell} k_j x_j \text{ ou minimiser } \sum_{1 \leq j \leq \ell} c_j x_j \\ \text{sous la contrainte } MX = U. \end{array} \right.$$

La résolution effective du problème sous cette forme paraît impraticable, les dimensions de la matrice M croissant très vite avec la taille de la table d'identification.

De plus, on doit vérifier que la solution X obtenue, reportée en cases marquées, correspond bien à un questionnaire.

Séparation des éventualités 2 à 2

Nous supposerons toutes les éventualités séparables.

Énonçons la propriété :

P8 : Dans tout questionnaire, la séparation de deux éventualités se fait avec une question et une seule.

La démonstration s'appuie sur une propriété des arborescences :

- si $(E_k, 0)$ et $(E_\ell, 0)$ sont parmi les sommets terminaux du questionnaire, au moins une question les sépare.
- supposons que Q_u et Q_v , employées dans le questionnaire, séparent E_k et E_ℓ : il existe un sommet (A, u) ayant deux arcs distincts incidents extérieurement, menant à E_k et E_ℓ ; et de même pour un autre sommet (B, v) .

(A, u) et (B, v) sont ascendants de $(E_k, 0)$, et il existe un chemin de l'un à l'autre. D'après P5 (I.4.), on a donc (par exemple) $B \subset A$.

Nous sommes dans la situation de la figure III.6. : il y a deux chemins distincts de (A, u) à $(E_\ell, 0)$ ce qui est impossible dans une arborescence. \square

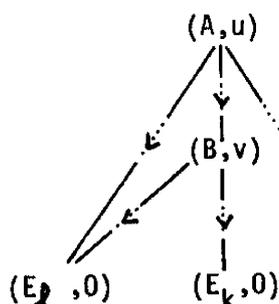


Figure III.6.

Soit \mathcal{G} l'ensemble des paires (non ordonnées) d'éventualités. La propriété précédente montre qu'à tout questionnaire, on peut associer une application $\varphi : \mathcal{G} \rightarrow Q$ telle que $\varphi(E_i, E_j) = Q_k$ équivaut à : E_i et E_j sont séparées par Q_k dans le questionnaire.

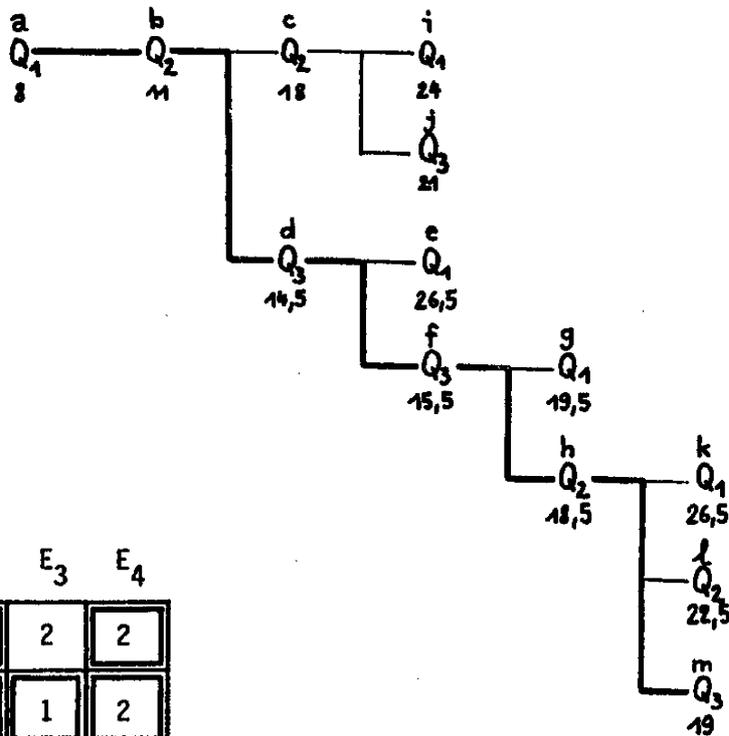
Inversement, on peut construire des applications φ vérifiant $\varphi(E_i, E_j) = Q_k$ si E_i et E_j peuvent être séparées par Q_k ; malheureusement elles ne correspondent pas toujours à un questionnaire : exemple, figure III.8. L'idée est de construire pas à pas une fonction φ réalisant un questionnaire, de coût intéressant ou optimal.

Nous sommes aidés par la connaissance des cases essentielles (II.2.), notion qui fait intervenir des paires d'éventualités.

La construction figure III.7. correspond à l'exemple, figure I.1.

Dans la partie supérieure, on a indiqué les possibilités de séparations, pour chaque paire d'éventualité. L'arborescence représente la construction d'une solution optimale par une méthode SEP (voir le chapitre V). Les questions choisies sont écrites sous les colonnes correspondant aux paires séparées.

	$E_2 E_4$	$E_3 E_4$	$E_1 E_2$	$E_1 E_3$	$E_2 E_3$	$E_1 E_4$
Q_1	1			1	1	1
Q_2		1	1		1	1
Q_3			1	1		1



	E_1	E_2	E_3	E_4
Q_1	1	1	2	2
Q_2	1	2	1	2
Q_3	2	1	1	1

Figure III.7.

Les colonnes ont été préalablement ordonnées, de façon que les premiers choix s'opèrent sur les cases essentielles, puis sur les doubles possibilités, ...etc... ceci afin d'accélérer la convergence de l'algorithme. Les lettres minuscules indiquent l'ordre de production des choix.

L'affectation sélectionnée est représentée en traits forts. Elle est indiquée, sous forme de cases marquées, en bas et à gauche de la figure. On vérifie qu'il s'agit bien d'un questionnaire.

$E_i E_j$	$E_2 E_4$	$E_3 E_4$	$E_1 E_2$	$E_1 E_3$	$E_2 E_3$	$E_1 E_4$
φ	Q_1	Q_2	Q_3	Q_3	Q_1	Q_2

	E_1	E_2	E_3	E_4
Q_1	1	1	2	2
Q_2	1	2	1	2
Q_3	2	1	1	1

Figure III.8.

CONCLUSION

Les différents essais de réductions proposés semblent inopérants -sauf peut-être pour la séparation des éventualités 2 à 2- pour la réalisation pratique des questionnaires. Le chapitre suivant apporte des lumières théoriques sur ce semi-échec. Mais, nous avons pu cependant dégager d'autres outils de représentations.

A cet égard, les graphes ET/OU constituent le meilleur outil de représentation, car non redondant et lié directement à l'algorithme -définition I.4. On l'aura toujours à l'esprit en seconde partie, lorsque nous étudierons la construction directe des questionnaires.

CHAPITRE IV.

-:-

LE PROBLÈME DU QUESTIONNAIRE OPTIMAL EST POLYNOMIAL COMPLET

-

Le but de ce court chapitre est de résumer les travaux de S.A. COOK (1971), R.M. KARP (1972), L. HYAFIL et R.L. RIVEST (1973) sur la complexité des problèmes combinatoires, mesurée en temps de calcul nécessaire pour la résolution. En particulier, le problème de former un questionnaire de coût optimal est fortement suspecté de n'être pas résoluble en temps polynomial en fonction de la taille des données. Nous en tirerons les conséquences pour la résolution pratique.

Il est généralement admis que seuls les problèmes solubles en un temps borné par $p(l)$, où p est une fonction polynôme et l la taille des données, sont traitables en temps raisonnable et satisfaisant sur ordinateur. Soit P l'ensemble de ces problèmes, parmi lesquels on trouve : chemin minimal dans un graphe, coupe minimale, résolution d'équations linéaires,...

Il existe beaucoup d'autres problèmes, de recherche opérationnelle notamment, qu'on n'est pas encore parvenu à résoudre en temps polynomial, mais seulement par des procédures de recherches en arbre avec retour arrière possible sur les choix déjà faits ("backtracking") : "Branch and Bound", procédures de Séparations et d'Evaluations Progressives (PSEP) ou Séquentielles (PSES), etc. Soit NP l'ensemble des problèmes ainsi résolus, avec des arbres de recherche bornés polynomialement en profondeur.

On trouvera une définition plus formelle de NP dans les travaux cités. R.M. KARP montre que NP est aussi l'ensemble des problèmes solubles en temps polynomial par des algorithmes non déterministes. Ce terme sous entend également une définition formelle issue de la théorie des automates. Les algorithmes non déterministes n'ont pas d'existence réelle. On peut en avoir une idée en imaginant un algorithme qui, confronté

à un choix entre plusieurs alternatives, possède la faculté de se dédoubler en autant de copies de lui-même qu'il y a de choix possibles, pouvant ensuite travailler parallèlement ; ou bien, un algorithme qui, placé devant un tel choix, peut de façon magique "deviner" la bonne solution.

Par exemple, notre algorithme définition du paragraphe I.4., doté d'une fonction CHOIX magique donnant la question qui assure toujours l'optimalité du questionnaire, nous permettrait une construction de ce questionnaire en un temps borné par un polynôme fonction de la taille du problème. (La taille étant ici $\text{Card}(E) \times \text{Card}(Q)$ par exemple). Le problème du questionnaire optimal appartient donc à l'ensemble NP.

Cet ensemble paraît très vaste. Cependant, il existe des problèmes, issus de la théorie des langages, dont l'appartenance à NP fait encore doute.

Plutôt que d'essayer de montrer, pour chaque problème de NP pris séparément, qu'il est ou n'est pas dans P, on essaie de montrer globalement soit $P = NP$, soit $P \neq NP$. Dans une telle démarche, il faut pouvoir comparer les problèmes entre-eux ; on dit qu'un problème L est réductible à un problème M si on peut convertir les données de L en données pour M, ceci en temps polynomial. S.A. COOK (1971) a démontré l'important théorème : tout problème de NP se réduit au problème de satisfiabilité. La satisfiabilité est le problème de déterminer si une fonction booléenne, écrite sous forme conjonctive normale, possède un point vrai.

La définition suivante est due à R.M. KARP (1972) :
L est polynomial complet si a) $L \in NP$; b) la satisfiabilité se réduit à L.

Ces problèmes sont en quelque sorte les plus compliqués dans NP. Voici le résultat obtenu par KARP : soit les problèmes polynomiaux complets sont tous solubles en temps polynomial, soit aucun d'entre-eux ne l'est. Dans le premier cas, il suffit de le montrer pour un seul d'entre-eux, et $P = NP$.

Les problèmes suivants sont polynomiaux complets : programmation linéaire en variables bivalentes, couverture minimale d'ensembles, circuit hamiltonien dans un graphe, nombre chromatique, ordonnancement de tâches...

L. HYAFIL et R.L. RIVEST (1973) ont montré, par une suite de réductions successives, que le problème du questionnaire de coût optimal est polynomial complet. Nous avons vu pourtant au paragraphe I.5. deux cas particuliers de ce problème (identifications élémentaire et lexicographique) à résolution polynomiale.

Remarquons que la réduction, paragraphe III.1., du problème du questionnaire optimal à un problème de P (chemin minimal) ne peut se faire en temps polynomial.

Ces études théoriques ont pour l'ingénieur l'intérêt d'indiquer des limites de faisabilité, des points de repère pour la résolution pratique de ces problèmes : bien qu'on n'ait pas encore prouvé à l'heure actuelle que $P \neq NP$, cette étude montre qu'on peut douter de pouvoir résoudre les problèmes polynomiaux complets, et celui du questionnaire optimal avec eux, en un temps croissant moins vite que l'exponentielle en fonction de la taille des données.

La résolution optimale du problème du questionnaire paraissant intrinsèquement difficile, il semble primordial, lorsqu'on est confronté à la pratique, de développer des méthodes heuristiques permettant de construire en temps raisonnable, des questionnaires sous-optimaux aussi proches que possible de la solution optimale ; en somme, un algorithme heuristique doit comporter une fonction CHOIX réelle, aussi proche que possible de la fonction magique de l'algorithme non déterministe.

Ceci est l'objet du chapitre VI de la seconde partie.

2ÈME PARTIE

CONSTRUCTIONS DIRECTES

DE QUESTIONNAIRES RÉALISABLES

INTRODUCTION

Après avoir vu en quels termes se pose le problème de la construction des questionnaires, et les diverses représentations qu'on peut lui donner, nous abordons la construction pratique.

La voie directe nous paraît la meilleure : les formulations en termes de plus court chemin, de recouvrement... ne semblent pas pouvoir amener une résolution plus simple.

Par méthode directe, nous entendons la construction d'un questionnaire, question par question, soit de la racine vers les feuilles (amont-aval), soit inversement comme dans la construction de HUFFMANN pour les questionnaires libres (II.2.), d'aval en amont, depuis les éventualités séparées jusqu'à la première question.

Nous présentons d'abord, chapitre V, une recherche d'algorithmes pour la résolution optimale, puis pour une construction heuristique chapitre VI.

Grâce aux algorithmes optimaux, les solutions obtenues peuvent être comparées à celles produites par des procédures heuristiques dont on veut apprécier la valeur. L'étude des algorithmes heuristiques reflète le souci d'applicabilité des méthodes de résolution. On obtiendra en fait des solutions souvent optimales, toujours en temps raisonnable.

CHAPITRE V.

-:-

CONSTRUCTIONS OPTIMALES

-

Construire un questionnaire réalisable optimal est un problème de nature fortement combinatoire, et les recherches présentées au chapitre IV nous incitent à penser qu'il n'existe probablement pas de méthodes autres qu'énumératives pour le résoudre.

On s'intéressera principalement au critère du coût pondéré ; la plupart des algorithmes s'appliquent cependant au critère du coût maximum.

Le premier paragraphe décrit les différentes méthodes possibles de constructions optimales. Puis, on présente au second paragraphe des améliorations possibles dues à la structure même du problème, basées principalement sur l'échange des questions.

Enfin, on établit pas à pas un nouvel algorithme de réalisation optimale de questionnaires. Deux méthodes avaient été jusqu'ici proposées : l'une, par F. DUBAIL (1967) et reprise par M.R. GAREY (1972 b) avec des améliorations, d'aval en amont ; l'autre, est une procédure de séparations et d'évaluations progressives, écrite par la S.E.M.A. (1970), opérant d'amont en aval. Nous avons programmé un algorithme basé sur une recherche "en profondeur d'abord", d'amont en aval.

V.1. DIFFERENTES METHODES POSSIBLESExpression du coût optimal

Rappelons l'expression II.(6) du coût pondéré d'un questionnaire \mathcal{Q} construit sur $F \subseteq E$, pour les données (E, Q, e, q) :

$$C(F) = P(F) \cdot q_r + \sum_{1 \leq s \leq b_r} C(F_r^s)$$

avec : $P(F) = \sum_{E_i \in E} e_i$; F_r^s est le s ème bloc de la partition qu'effectue Q_r sur F . Par convention $C(F_r^s) = 0$ si $\text{Card}(F_r^s) = 0$ ou 1. Q_r est la question racine du questionnaire.

Nous notons $C^*(F)$ le coût minimum d'un questionnaire sur F .
 Démontrons l'expression récursive du coût pondéré optimal :

$$C^*(F) = \min_{Q_i \in Q} \left\{ P(F) \cdot q_i + \sum_{1 \leq s \leq b_i} C^*(F_i^s) \right\} \quad (1)$$

- Supposons optimal le questionnaire Q construit : $C(F) = C^*(F)$. Nous avons aussi $C(F_s^r) = C^*(F_s^r) \forall s$, sinon $C^*(F)$ pourrait être minimisé, et ne serait pas optimal. Donc :

$$C^*(F) = P(F) \cdot q_r + \sum_{1 \leq s \leq b_r} C^*(F_s^r)$$

Mais Q_r est la racine du questionnaire optimal. r est donc l'indice assurant le minimum de la quantité :

$$P(F) \cdot q_i + \sum_{1 \leq s \leq b_i} C^*(F_i^s)$$

ce qui prouve (1).

Cette formule généralise l'expression de la longueur d'un questionnaire réalisable ($q_i=1, \forall i$) mis en évidence par F. DUBAIL (1967) dans la première étude faite sur le sujet. La forme récursive de (1) évoque la programmation dynamique, mais l'expression est à la base de toutes les méthodes que nous allons voir, même celles ne relevant pas explicitement de la programmation dynamique.

Le "principe d'optimalité", selon lequel toute solution partielle d'une solution optimale est elle-même optimale, est ici théorème.

On montre de la même manière pour le coût maximum :

$$K^*(F) = \min_{Q_i \in Q} \left\{ q_i + \max_{1 \leq s \leq b_i} K^*(F_i^s) \right\} \quad (2)$$

Constructions aval-amont

Dans le cas des questionnaires libres, il est remarquable de constater que seules -à notre connaissance- des procédures opérant d'aval en amont (algorithme d'HUFFMAN) fournissent des solutions optimales, sans énumération.

Une construction par l'algorithme de HUFFMANN qui respecterait la sémantique serait optimale. Mais dans le cas général, on ne trouve pas toujours de question correspondant au regroupement d'éventualités demandé par l'algorithme.

La méthode générale de construction aval-amont relève explicitement de la programmation dynamique : on construit de proche en proche des solutions partielles optimales, en commençant par des questionnaires de deux éventualités, puis trois... et ainsi de suite, jusqu'au questionnaire de $\text{Card}(F)$ éventualités, en s'aidant à chaque étape des résultats obtenus aux ordres précédents. C'est la méthode utilisée au paragraphe III.2., mais facilitée par la connaissance du graphe ET/OU.

Cette voie, déjà indiquée par F. DUBAIL (1967), a été choisie aussi par M.R. GAREY (1972 b).

Il y a $2^{|F|}$ sous-ensembles de F , ce qui correspond, en appliquant brutalement la méthode, à autant de sous problèmes à résoudre, chacun nécessitant $|F|.|Q|$ étapes. On ne peut donc pas pratiquement utiliser directement cette procédure, tant pour la place mémoire que pour le temps d'exécution nécessaires, avec $|F|$ dépassant 15 ou 20. La méthode nécessite des améliorations (V.2.).

Dans les constructions aval-amont, on va engendrer des sous-ensembles à séparer qui sont peut-être "non-formulables", c'est-à-dire n'appartenant pas au graphe ET/OU des solutions. C'est là l'inconvénient majeur de la méthode. Voyez, par exemple, figure III.2. : il n'y a qu'un seul sous-ensemble à trois éventualités formulable .

Les solutions partielles obtenues pour ces sous-ensembles non formulables ne seront d'aucune utilité par la suite.

La construction amont-aval, se situant strictement sur le support du graphe ET/OU, évite ce défaut.

Constructions amont-aval

Les solutions sont développées en arbre ; chaque sommet de cet arbre des solutions \mathcal{A} (voir III.1.), s'il est non terminal, représente un questionnaire inachevé. Au cours de l'exploration de \mathcal{A} , les questionnaires énumérés sont construits de proche en proche, en commençant par la racine, puis le second niveau de question... enfin les éventualités séparées.

On veut éviter d'explorer explicitement \mathcal{A} en entier. Pour cela, on s'aide d'une fonction évaluée en tout sommet s (questionnaire inachevé) de \mathcal{A} examiné. Un coût peut déjà être attribué à s , celui des questions déjà posées ; d'autre part, un coût minimum peut être évalué pour compléter la solution (cf. les techniques du paragraphe II.2.). La somme de ces deux coûts fournit une borne inférieure des coûts de toutes les solutions complètes pouvant être développées à partir de s . Soit L_s cette borne inférieure. A ce niveau deux méthodes générales d'énumération (B. ROY, 1970, tome 2 et A. KAUFMANN, 1968) peuvent être ici appliquées :

- une procédure de séparations et d'évaluations séquentielles (PSES), parfois nommée énumération implicite.

On explore en profondeur d'abord, c'est-à-dire en s'enfonçant toujours dans l'arbre des solutions, directement vers un questionnaire achevé quand c'est possible, sinon en remontant au dernier sommet laissé pendant. L'avantage est de conserver en mémoire uniquement le chemin aboutissant au sommet actuellement examiné, sous forme de pile. Si on connaît déjà une solution complète, soit par une méthode heuristique, soit obtenue par l'énumération, de coût C , on ne développera pas les sommets tels que $L_s \geq C$. Les solutions issues d'un tel sommet éliminé sont implicitement explorées. A chaque fois qu'on obtient une solution complète, on la conserve en mémoire à la place de celle retenue antérieurement, si son coût est moindre. L'algorithme s'arrête après exploration complète (implicite) de l'arbre \mathcal{A} .

- la procédure de séparations et d'évaluations progressives (PSEP), ou "Branch and Bound".

La fonction L_s qui précédemment était une aide, devient ici le "moteur" de l'algorithme. Appelons "sommet pendant" de \mathcal{A} , un sommet examiné (c'est-à-dire évalué) mais non encore développé. A chaque étape, on sélectionne le sommet pendant possédant la plus petite valeur L_s . Si nous pouvons ainsi sélectionner un sommet terminal (questionnaire achevé), nous avons une solution optimale. Sinon, ce sommet est développé, donnant naissance à d'autres sommets pendants, et on reprend la sélection comme plus haut, parmi tous les sommets pendants. Le développement de \mathcal{A} se fait en somme, en largeur.

Cette méthode nécessite beaucoup plus de place mémoire que la précédente, car il faut mémoriser tous les sommets pendants. Elle a été décrite et programmée par la S.E.M.A. (1970).

La connaissance d'une bonne solution complète de coût C n'est utile que pour libérer la place occupée par les sommets pendants tels que $L_s \geq C$.

Bien que la méthode PSEP, contrairement à sa concurrente, exige du temps de calcul pour les rangements en mémoire, les comparaisons et sélections, on peut penser qu'elle a des chances d'être plus rapide. En fait, cela dépend de la complexité du problème, du temps d'évaluation de L_s , de la place mémoire dont on dispose, et du temps d'accès à la mémoire.

Ces deux procédures laissent encore deux degrés de liberté, liés davantage à la nature du problème :

- 1) le choix de la fonction L_s
- 2) le choix dans la façon de construire l'arbre des solutions : en effet, un sommet pendant peut être développé de différentes manières, pourvu que l'on n'omette aucune solution pouvant découler de ce sommet : ici, on peut jouer sur l'ordre d'application des questions, ou du développement des issues.

L'efficacité d'un algorithme dépendra pour beaucoup de ces deux choix.

La méthode PSES a l'avantage de fournir rapidement des solutions, de plus en plus proches de l'optimum ; mais on ne peut pas donner une bonne évaluation de cette proximité. Au contraire, avec la PSEP, nous avons, à chaque étape, un intervalle dans lequel se trouve C^* : supposons obtenue une solution complète (par une méthode heuristique, ou par développement d'un sommet ayant donné naissance à un sommet terminal) de coût C ; si L_s est la fonction associée au dernier sommet sélectionné, alors : $L_s \leq C^* \leq C$. On peut arrêter l'algorithme dès que l'intervalle est jugé assez étroit. De plus, à tout moment, il est possible de "plonger" vers une solution complète, à l'aide d'un algorithme heuristique (VI).

V.2. AMÉLIORATIONS DUES AU PROBLÈME LUI-MÊME

Certaines améliorations peuvent être apportées aux méthodes générales d'énumération. Citons par exemple la procédure "alpha-bêta", dont on trouvera une présentation dans le livre de N.J. NILSSON (1971). On

s'intéresse plutôt aux améliorations dues aux propriétés des questionnaires permettant de réduire l'énumération. Certaines seront employées dans le programme que nous avons écrit.

Elimination des redondances

Soient $F = \{F_1, F_2, \dots, F_p\}$ et $G = \{G_1, G_2, \dots, G_q\}$ deux partitions sur un même ensemble. On dit que F est plus fine que G (noté $F \ll G$) si et seulement si $\forall_{i,j}, F_i \cap G_j \neq \emptyset \Rightarrow F_i \subseteq G_j$. C'est une relation d'ordre partiel sur les partitions d'un ensemble, permettant de construire une structure de treillis. Un chemin dans le graphe des partitions (III.2.) est constitué de partitions de plus en plus fines.

- redondances du premier ordre : une question Q_i est dite plus fine que Q_j si la partition induite par Q_i est plus fine que celle induite par Q_j . Autrement dit, Q_i sépare davantage que Q_j . Si, de plus, $q_i \leq q_j$, Q_j peut être éliminée de la table d'identification, car elle est redondante : le remplacement de Q_j par Q_i , dans toute construction, permettrait de réduire le coût. Dans la figure V.1., Q_2 est éliminée par Q_1 .

	E_1	E_2	E_3	E_4	E_5
Q_1 (5)	1	1	2	2	3
Q_2 (6)	1	1	2	2	2

Figure V.1.

- redondances du second ordre : on dit que $H = \{H_1, H_2, \dots, H_n\}$ est la partition produit de F et G (noté $H = F.G.$) si et seulement si : $\forall k, \exists (i,j) / H_k = F_i \cap G_j$. On a $H = F.G. \Rightarrow H \ll F$ et $H \ll G$.

L'utilisation de deux questions Q_i et Q_j permet d'induire sur l'ensemble séparé, la partition produit des deux partitions effectuées par Q_i et Q_j . Soit $Q_i \cdot Q_j$ la question fictive obtenue ainsi par produit des partitions, à laquelle on attribue le coût $q_i + q_j$ (ou $\max(q_i, q_j)$ si on s'intéresse au coût maximal). Toute question Q_k telle que $Q_i \cdot Q_j \ll Q_k$, et $q_i + q_j \leq q_k$ (ou $q_k \geq \max(q_i, q_j)$ pour le coût maximal) peut être éliminée de la table d'identification : le remplacement de Q_k par la succession $Q_i - Q_j$ dans toute construction ne peut que réduire le coût. Figure V.2., Q_3 peut être éliminée, en fonction de Q_1 et Q_2 .

	E_1	E_2	E_3	E_4
Q_1 (5)	2	1	1	2
Q_2 (5)	2	2	1	2
Q_3 (10)	2	1	2	2
$Q_1 \cdot Q_2$ (10)	2	3	1	2

Figure V.2.

- ces deux types de redondance sont aisément généralisés au produit de 3 ou plusieurs partitions de questions.

Ce type d'amélioration est intéressant dans la construction amont-aval. On a intérêt à éliminer au moins les redondances de 1er ordre, non seulement dans la table d'identification du problème original, mais aussi pour les sous-problèmes engendrés.

Eventualités compatibles

L'amélioration suivante est due à M.R. GAREY (1972 b) ; cet auteur l'utilise dans un algorithme de construction aval-amont, dans le cas de questions toutes dichotomiques. Elle vise à éliminer a priori certains sous-problèmes qui ne peuvent faire partie de la solution optimale. Donnons une idée de la méthode, qui ne concerne que le critère coût pondéré.

- deux eventualités sont compatibles si :
 - a) elles ne sont séparables l'une de l'autre que par l'emploi d'une question élémentaire,
 - b) l'éventualité de plus fort poids est identifiée avec la question élémentaire de plus faible coût (les questions élémentaires manquantes sont supposées avoir un coût infini).
- cette notion permet d'énoncer le théorème :

"si E_i et E_j sont compatibles, avec $\frac{q_i}{e_i} \leq \frac{q_j}{e_j}$ alors il existe une solution optimale dans laquelle E_j appartient à tout sommet non terminal qui contient E_i ". (E_i est isolée avant ou en même temps que E_j).

- la construction de "partitions ordonnées" permet, grâce au théorème, d'engendrer lexicographiquement un nombre plus restreint de sous-problèmes à traiter.

Ainsi, l'exemple donné par cet auteur (figure II.6.) nécessite l'examen de seulement 43 sous-problèmes au lieu de $2^7 - 7 - 1 = 120$.

En fait, l'amélioration obtenue dépend essentiellement de la structure du problème, et est d'autant plus grande qu'il y a d'éventualités compatibles. Elle n'empêche pas l'examen de sous-ensembles "non-formulables".

Echange de questions successives

L'idée d'échanger les coûts des questions est à la base de la construction du questionnaire de HUFFMANN avec coûts libres rappelée au § III. (P.P. PARKHOMENKO, 1969 et C.F. PICARD, 1972, T2). Mais dans le cas des questionnaires réalisables, on se heurte à la sémantique des questions : on échange aussi les partitions.

Ceci ne concerne que l'optimisation du coût pondéré. On se limite aux questions dichotomiques.

Soit un questionnaire sur F, duquel on isole le sous-graphe formé de la première question (soit U) et de la suivante (soit V), sur l'une des issues de U (figure V.3.). La succession des deux questions induit sur F la partition $\{\alpha, \beta, \gamma\}$.

Les deux questions sont dites échangeables, si la succession U-V peut être changée en V-U sans modifier la partition induite.

L'idée est d'optimiser localement le questionnaire en observant si l'échange de questions successives peut réduire le coût du questionnaire.

Ayant choisi la racine U, on rejettera, comme successeurs, les questions V, échangeables avec U, si la configuration U-V coûte plus que V-U : elle conduirait à une solution obligatoirement non-optimale.

Quand deux questions sont-elles échangeables ?

- si V partage α , la succession V-U ne produit pas la même partition, U et V ne sont pas échangeables ;
- sinon, les questions sont échangeables. Soient u et v les coûts respectifs de U et V.

coût local de la succession U-V : $u.P(F) + v.P(\beta U \gamma)$

coût local de la succession V-U : $v.P(F) + u.P(\alpha U \beta)$

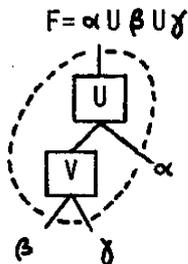


Figure V.3.

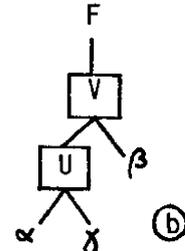
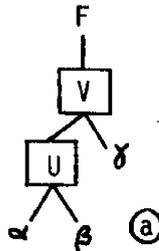


Figure V.4.

En posant $k = \frac{P(\gamma)}{v}$ (cas (a) figure V.4.) ou $k = \frac{P(\beta)}{v}$ (cas (b)), on déduit facilement qu'il faut rejeter la succession U-V si $k > \frac{P(\alpha)}{u}$

Lorsque $k = \frac{P(\alpha)}{u}$, une seule des deux configurations nécessite un développement.

Remarquons que si U et V sont échangeables, ils le sont seulement pour une issue de U ; en effet, V ne partage pas α .

Pratiquement, dans le cas de questions échangeables, seule la bonne succession est à retenir.

Énonçons la règle :

- une question élémentaire est échangeable avec n'importe quelle autre question. Dans ce cas, on retrouve localement les résultats de M.R. GAREY,
- plus généralement, deux questions sont échangeables si leur partition-produit a trois blocs.

Pratiquement, pour chaque sous-problème, on effectue C_n^2 comparaisons, en notant pour la suite les successions 'interdites'.

L'amélioration en temps de calcul dépend encore de la nature du problème : la réduction du nombre de sous-problèmes à examiner ne compense pas toujours le temps utilisé à calculer les successions interdites. Le tableau V.5. donne les résultats obtenus avec notre algorithme, pour l'exemple de M.R. GAREY (figure II.6.), et pour un exemple industriel réel (figure II.7.), avec et sans échanges. Dans le second exemple, l'échange des questions permet de réduire le temps de calcul d'un facteur 3.

	Q	E	sans échange		avec échange	
			n	t	n	t
Figure II.6.	6	7	37	0,05	18	0,09
Figure II.7.	14	13	12 146	82	1172	27

n = nombre de sous problèmes rencontrés.

t = temps de calcul, en secondes, sur IBM 370/168

Tableau V.5.

Cas particulier des questions élémentaires

Lorsque p éventualités sont à séparer par des questions toutes élémentaires, le questionnaire prend la forme d'une succession de p-1 questions (figure V.6.).

D'après l'étude sur l'échange des questions, une condition nécessaire d'optimalité est : $\frac{q_i}{q_{i+1}} \geq \frac{q_{i+1}}{q_{i+2}}$ pour deux questions Q_i et Q_{i+1} successives.

- si $|Q|=|E|-1$, le résultat précédent permet de construire directement le questionnaire optimal,
- sinon, $|Q|=|E|$, et une question de Q ne sera pas utilisée.

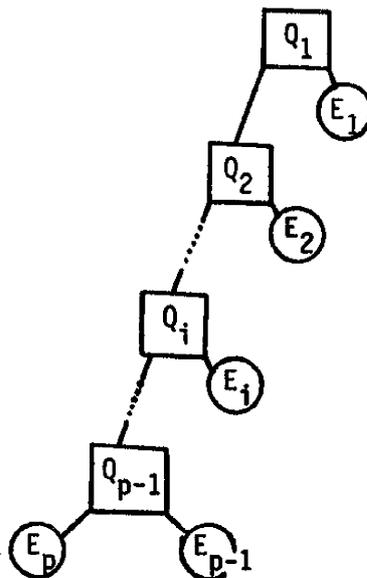


Figure V.6.

	E_1 (0,5)	E_2 (0,25)	E_3 (0,25)	$\frac{e_i}{q_i}$
Q_1 (10)	1	2	2	0,0500
Q_2 (6)	2	1	2	0,0415
Q_3 (6)	2	2	1	0,0415

(a)

	E_1 (0,7)	E_2 (0,2)	E_3 (0,1)	$\frac{e_i}{q_i}$
Q_1 (10)	1	2	2	0,0700
Q_2 (9)	2	1	2	0,0222
Q_3 (5)	2	2	1	0,0200

(b)

Figure V.7.

V.3. ETUDE D'UN ALGORITHME DE CONSTRUCTION DE QUESTIONNAIRES, DE COUT PONDERE OPTIMAL

Présentation générale

L'algorithme proposé est construit à partir de l'algorithme-définition (I.4.). Nous conservons la forme récursive, ce qui a deux avantages :

- 1) l'étroite parenté avec l'algorithme-définition, et l'expression récursive du coût optimal (1), constitue une preuve du programme
- 2) la compréhension en est facilitée, les risques d'erreurs de programmation sont diminués.

Cette forme récursive dissimule en réalité une exploration de l'arbre des solutions selon une méthode de type PSES, à l'aide d'une pile gérée implicitement par l'algorithme.

Nous avons choisi PL/1 comme langage de programmation, car il permet l'emploi de fonctions récursives, les traitements logiques

de masques, et les chaînages par pointeurs très utiles pour mémoriser une structure d'arbre. De plus, le programme est rendu plus lisible et plus souple grâce à la structure de blocs.

Dans les différentes versions présentées ici, la procédure centrale "QUESTOPT" comprend de 100 à 150 instructions PL/1 ; le programme entier comporte environ 400 instructions PL/1, correspondant à 25 K octets. La mémoire utilisée au total varie de 75 à 130 K octets selon les problèmes traités. L'ordinateur utilisé est un IBM 370/168 du réseau CIRCE.

Organigramme de principe

La figure V.8. montre en parallèle deux organigrammes. Le premier : QUEST (F, \mathcal{Q} , C) construit sur F un questionnaire \mathcal{Q} de coût C. Il est identique à l'algorithme-définition I.4., mises à part les modifications suivantes :

- F est supposé entièrement séparable, ce qui permet de supprimer lignes 4 et 12 de la définition, le test correspondant aux éventualités non séparables ;
- on ne construit pas les sommets de type (E,0) du questionnaire : ils sont sous-entendus et occuperaient inutilement de la place mémoire. Pour ce faire, on modifie la ligne 7 en : "si Card(F_i^S) \neq 0 ou 1 alors faire :".

Le second organigramme : QUESTOPT (F, \mathcal{Q}^* , C^*), construit sur F un questionnaire optimal \mathcal{Q}^* de coût C^* . C'est la traduction algorithmique de l'expression récursive du coût optimal, rappelée en-dessous. On remarque en particulier la boucle extérieure de minimisation, avec ses variables auxiliaires.

L'intérieur de la boucle est identique à QUEST en vis-à-vis, où les questionnaires construits sur les sous-ensembles F_i^S séparés sont optimaux.

Toutes les variables sont locales à leur niveau de récursivité, et les arguments sont transmis par valeur.

La fonction QUESTOPT calcule toutes les solutions possibles, à une restriction importante près : à tout niveau, lorsqu'on cherche un s+ième questionnaire optimal, on a déjà trouvé des questionnaires optimaux pour les s premières issues.

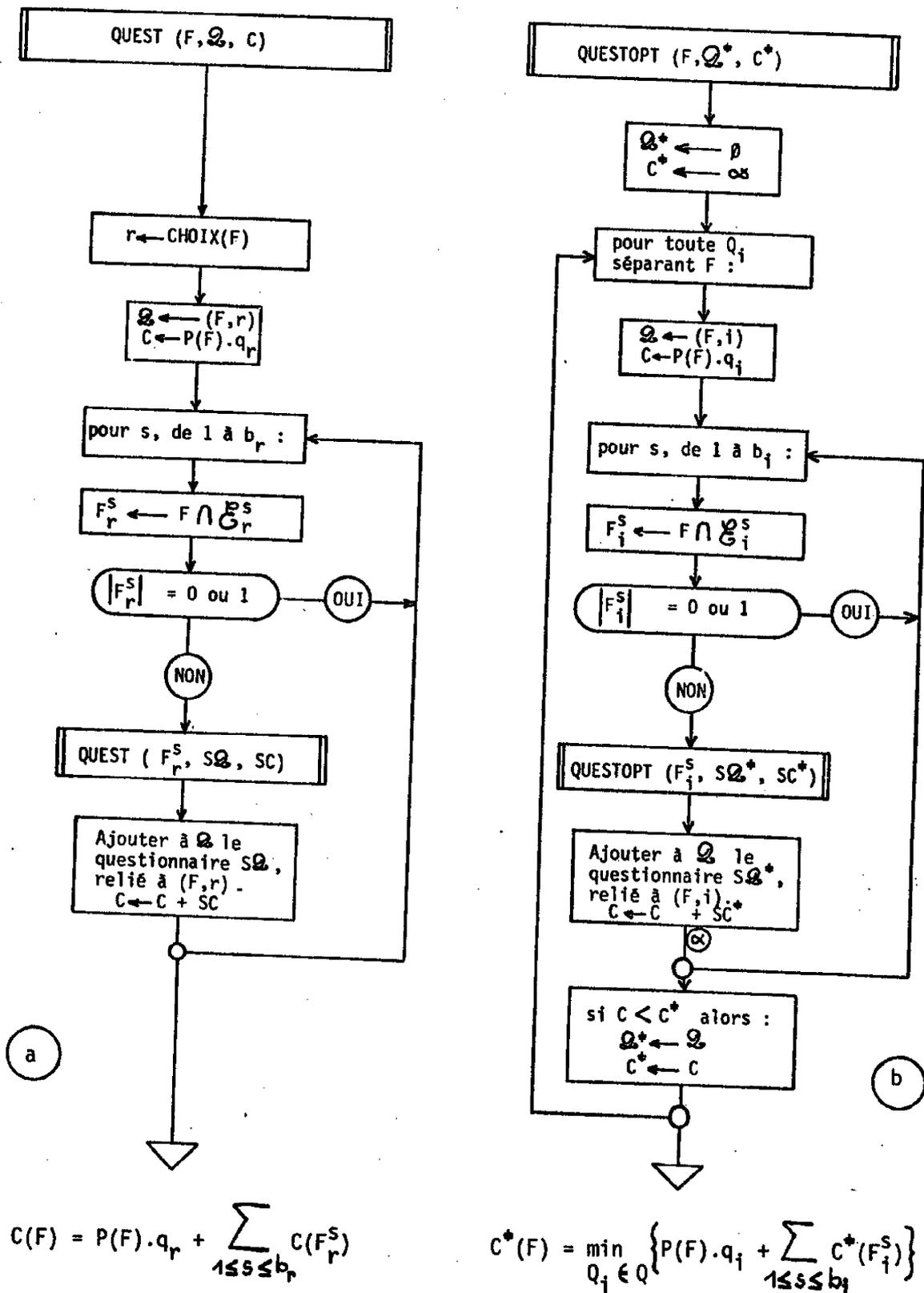


Figure V.8.

Comment tenir compte du coût déjà dépensé en cours d'énumération

Il n'est pas nécessaire d'examiner toutes les issues d'une question, lorsque ce qu'on a déjà construit dépasse en coût une solution meilleure retenue. En \otimes , nous pourrions donc mettre un test : "Si $C \geq C^*$, passer à la question suivante". Nous pouvons encore aller plus loin dans ce type de raccourci et utiliser aussi la dépense déjà faite aux niveaux antérieurs de récursivité. Pour cela, nous utilisons C^* à la fois comme paramètre d'entrée et de sortie de la procédure QUESTOPT : on recherche un questionnaire optimal de coût inférieur à C^* (paramètre d'entrée). S'il en existe, alors, comme précédemment, le résultat se trouve dans \mathcal{Q}^* et C^* (paramètre de sortie) : sinon, \mathcal{Q}^* est retourné vide et C^* reste inchangé. Ces améliorations introduisent l'exploration implicite de l'arbre des solutions.

Le paramètre d'entrée C^* transmis au premier niveau, sera le coût calculé par une procédure heuristique.

Comment stocker les solutions partielles

On ne transmet pas, bien sûr, les paramètres \mathcal{Q}^* sous forme d'arbre explicitement. Il suffit de transmettre l'indice de la question racine du questionnaire optimal, soit r^* et de conserver dans une table annexe tous les couples (F, r^*) engendrés au cours de la construction. Une fois la procédure terminée, la connaissance de la racine suffit, grâce à la table, pour reconstruire le questionnaire optimal.

Un même sous problème peut être rencontré plusieurs fois au cours de l'énumération. La consultation de la table permettra de gagner du temps si une solution optimale y figure déjà. Cet ensemble de solutions partielles déjà calculées doit être organisé de façon efficace pour accélérer son accès. On a donc choisi une structure d'arbre binaire, pour laquelle un élément peut être retrouvé en un temps moyen proportionnel au logarithme du nombre total d'éléments (D.E. KNUTH, 1973). Cette structure d'information a aussi l'avantage de pouvoir être utilisée concurremment en construction et en consultation. L'élément, représentant un sommet éventuel du questionnaire optimal, comprend 5 champs :

M	e	γ	lien droit	lien gauche
---	---	----------	------------	-------------

- M est le masque d'un ensemble F (ième bit de M à '1' si $E_i \in F$)
- Si $e \neq 0$: e est l'indice d'une question racine d'un questionnaire optimal sur M, de coût γ .
- Si $e = 0$: il n'existe pas de questionnaire sur F de coût inférieur à γ .

La clé de comparaison et d'accès est M, considéré comme un nombre écrit en binaire.

Redondances

Enfin, nous ajoutons un "filtre" qui élimine, pour chaque sous problème examiné, les questions qui ne séparent pas et celles qui sont redondantes au 1er ordre. Pour faciliter cette opération, les questions sont ordonnées au départ en coûts non décroissants.

La figure V.9. montre l'organigramme général du programme écrit.

Variantes

Une autre version intègre l'échange des questions (V.2.) : on intercale la recherche des successions "interdites" après l'élimination des questions redondantes. Les indices des questions ne pouvant succéder sous peine de non-optimalité, sont passés comme paramètres à la procédure de niveau suivant. Le nombre d'appels de la fonction récursive est très sensiblement réduit. Dans l'exemple de M.R. GAREY (tableau V.5.) ce nombre passe de 37 à 18 (43 sous problèmes sont examinés par une méthode aval-amont, d'après cet auteur). Pour l'autre exemple, de 12 146 à 1172. Malheureusement, cette réduction se paye par un temps de calcul plus long à chaque étape, si bien que cette version ne devient intéressante que dans le cas où la proportion de questions élémentaires est proche de 1. Il semble que l'échange des questions et la retenue de solutions partielles n'aient pas d'effets complémentaires : l'échange évite de repasser souvent par des solutions déjà construites, mais consomme beaucoup de temps de calcul.

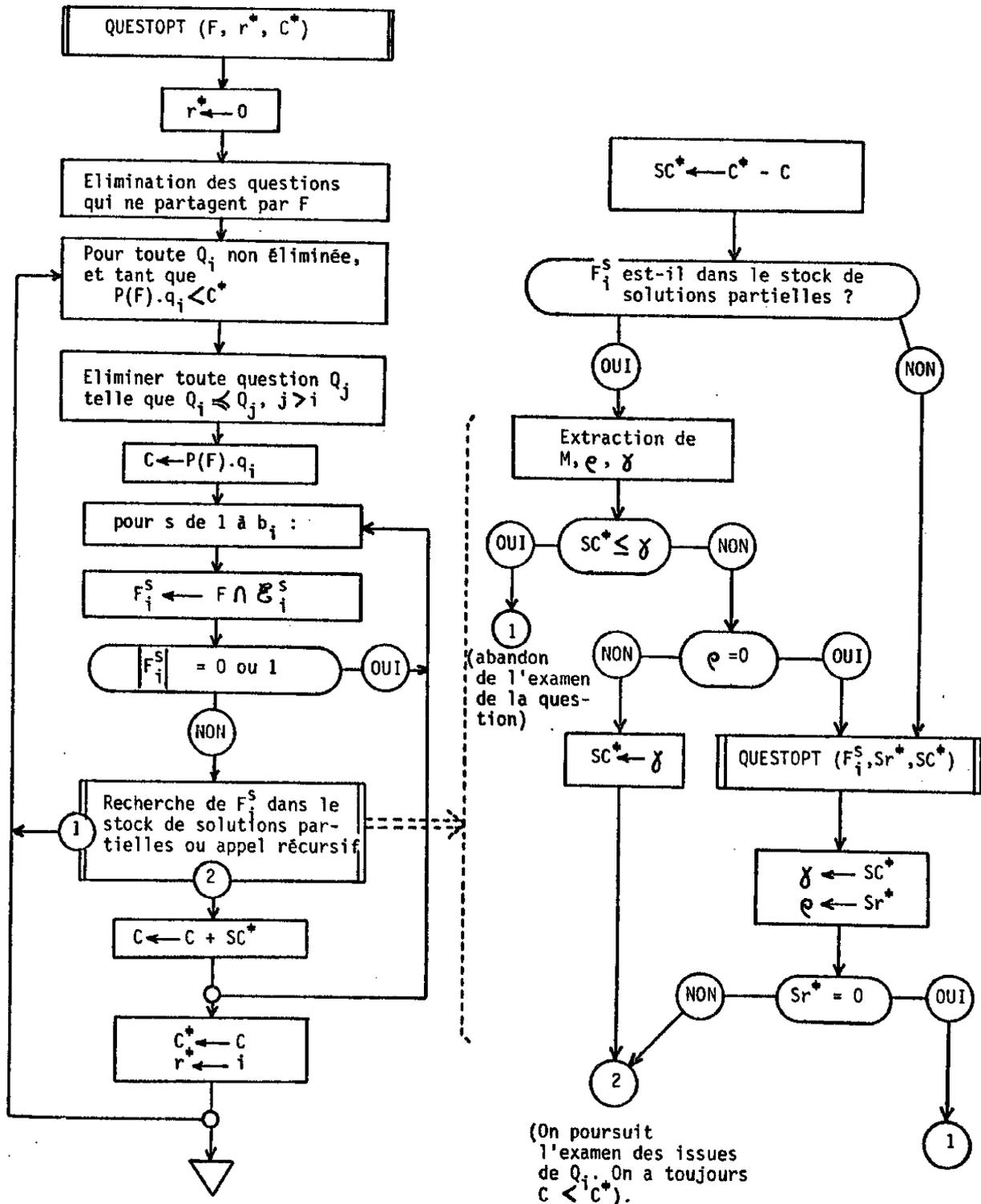


Figure V.9.

Performances de l'algorithme et des variantes

Pour étudier le comportement statistique des différentes versions, nous avons construit un générateur de problèmes d'identification dont les paramètres sont :

- le nombre d'éventualités $|E|$
- le nombre de questions $|Q|$
- la proportion de questions élémentaires : QE
- la proportion de questions de base 3 : Q3 (les autres sont de base 2).

A l'aide d'un générateur de nombres pseudo-aléatoires, on produit une série de tables d'identification, avec leurs vecteurs coûts et poids, en accord avec les valeurs données des paramètres.

Les poids et coûts engendrés sont des nombres entiers, dont on peut faire varier la distribution et l'échelle (nombre de chiffres).

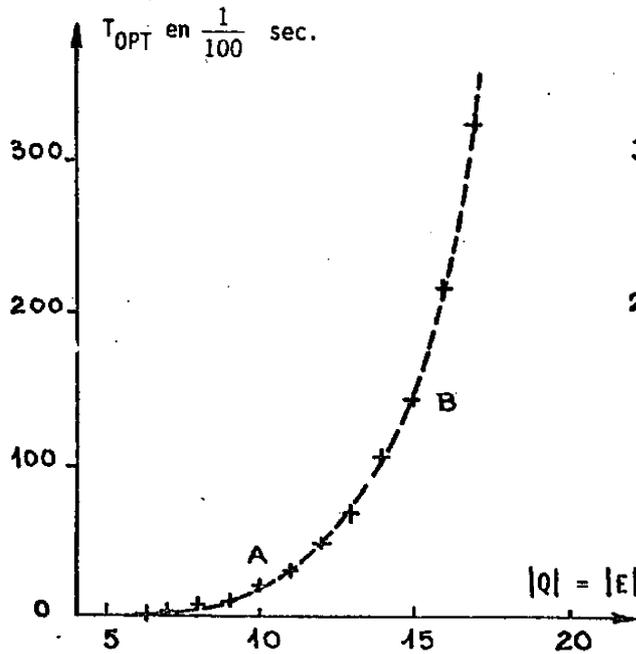
Les courbes 1, 2 et 3 donnent les temps moyens de calcul T_{OPT} en 1/100 s sur la machine IBM 370/168, dans les conditions suivantes :

	taille du problème $ Q = E $	nature des questions	
		QE	Q3
Courbe 1	6 → 17	0	0
Courbe 2	10	0 → 1	0
Courbe 3	15	0	0 → 1

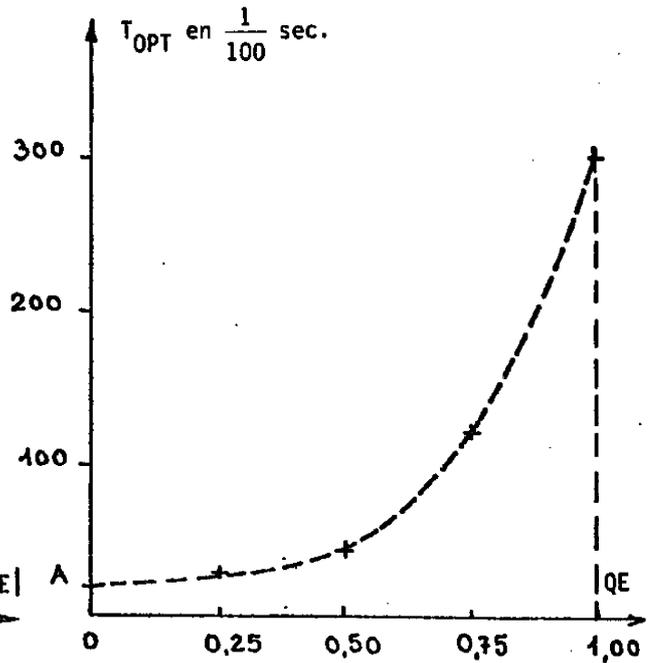
Les distributions des coûts et des poids sont uniformes entre 1 et 1000 ; chaque point est une moyenne sur 100 problèmes différents, de mêmes caractéristiques. Pour les courbes 1 et 2, toutes les questions sont de base 2 (Q3 = 0). Le chiffre 100 a été choisi empiriquement, en tenant compte du fait que les résultats obtenus pour 50 et 100 problèmes engendrés sont très proches.

L'allure de ces courbes traduit la nature fortement combinatoire du problème.

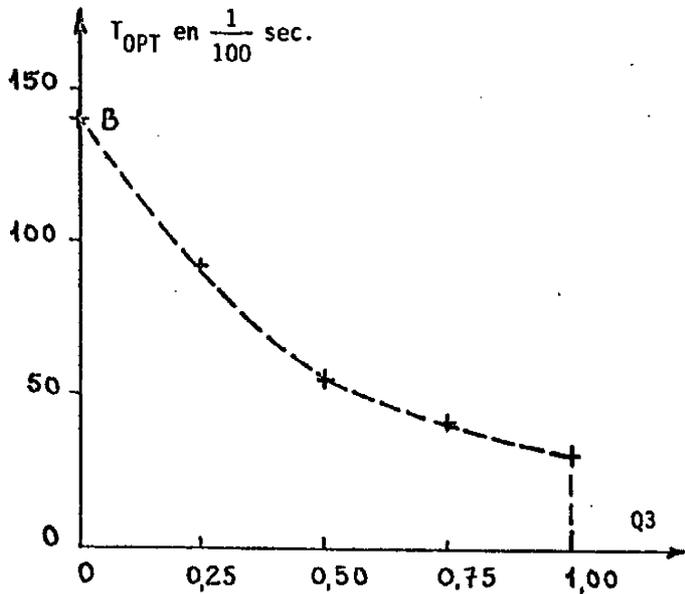
Les temps obtenus sont équivalents pour une distribution gaussienne des poids et/ou coûts. Lorsque le nombre de chiffres signifi-



Courbe n° 1. Temps de calcul des solutions optimales en fonction de la taille du problème.



Courbe n° 2. Temps de calcul des solutions optimales en fonction de la proportion des questions élémentaires ($|Q| = |E| = 10$).



Courbe n° 3. Temps de calcul des solutions optimales en fonction de la proportion des questions de base 3 ($|Q| = |E| = 15$).

catifspasse de 1 à 4, les temps de calcul sont diminués d'environ 10 % (pour les problèmes du point A), ce qui correspond au fait que, lorsque la précision est plus grande, les coûts des solutions sont moins proches les uns des autres.

Les écarts-type sur les temps de calcul sont de l'ordre de $\frac{T_{OPT}}{2}$; à titre indicatif, nous avons, pour deux points de la première courbe :

	$ Q = E = 10$	$ Q = E = 15$
temps de calcul	23 (11)	141 (97)
nombre d'appels de la fonction récursive	102 (46)	432 (249)
nombre de solutions partielles retenues	68 (22)	247 (105)

Les écarts-type sont indiqués entre parenthèses sous les moyennes. On sait qu'il n'est pas aisé de mesurer exactement le temps CPU pur passé sur une partie de programme pour une machine travaillant en multiprogrammation : d'après une étude de consistance des temps fournis par le sous-programme chronométré, on peut admettre 10 % d'erreur.

La figure V.10. donne un exemple de répartition des temps de calcul pour un lot de 100 problèmes de taille 15 x 15. Nous présentons figure V.11. un exemple de table d'identification de taille 32 x 32 fournie par le générateur, et son traitement page suivante, par un algorithme heuristique (chap. VI) et par un algorithme de construction optimale.

Les mesures sur l'effet de la rectangularité de la table montrent que les temps de calcul dépendent beaucoup de $|E|$ lorsque $|E| < |Q|$; Si $|E| > |Q|$, l'ordre de grandeur est le même que pour une table carrée.

Rappelons qu'on doit fournir, à l'entrée de la procédure QUESTOPT un paramètre : le coût d'une solution possible ou coût initial C_i . Si $C_i > C^*$, on obtient en fin de calcul un questionnaire optimal, et sinon

```

NT= 15 NE= 15 NCSC= 1.000000E+03
NCSP= 1.000000E+03 IX= 899 SEUILT2= 1.000000E+00
SEUILTE= 0.000000E+00 SEUILT2= 1.000000E+00 NCSC= 1.000000E+03
STATISTIQUES SUR LE TEMPS DE CALCUL SOL OPT ETY= 9.746798E+01
VMAX= 5.700000E+02 VMIN= 4.000000E+01 MCY= 1.414299E+02
KINF= 0 KSUP= 2 TCTAL= 1.414300E+04
REPARTITION DE -1.509738E+02 A 4.338337E+02 EN 21 CLASSES DE LARGEUR 2.784799E+01
1 -1.2313E+02 0
2 -9.5279E+01 0
3 -6.7439E+01 0
4 -3.9582E+01 0
5 -1.1724E+01 0
6 1.9114E+01 0
7 4.2992E+01 4
8 7.1810E+01 19
9 9.9658E+01 22
10 1.2751E+02 17
11 1.5335E+02 8
12 1.8209E+02 5
13 2.1105E+02 6
14 2.5890E+02 6
15 2.6673E+02 3
16 2.7459E+02 2
17 3.2244E+02 2
18 3.5029E+02 1
19 3.7814E+02 2
20 4.0599E+02 1
21 4.3383E+02 0

```

Figure V.10. Répartition des temps de calcul pour un lot de 100 problèmes engendrés
 $|E| = |Q| = 15$; $QE = 0$; $Q3 = 0$;
(toutes questions de base 2).

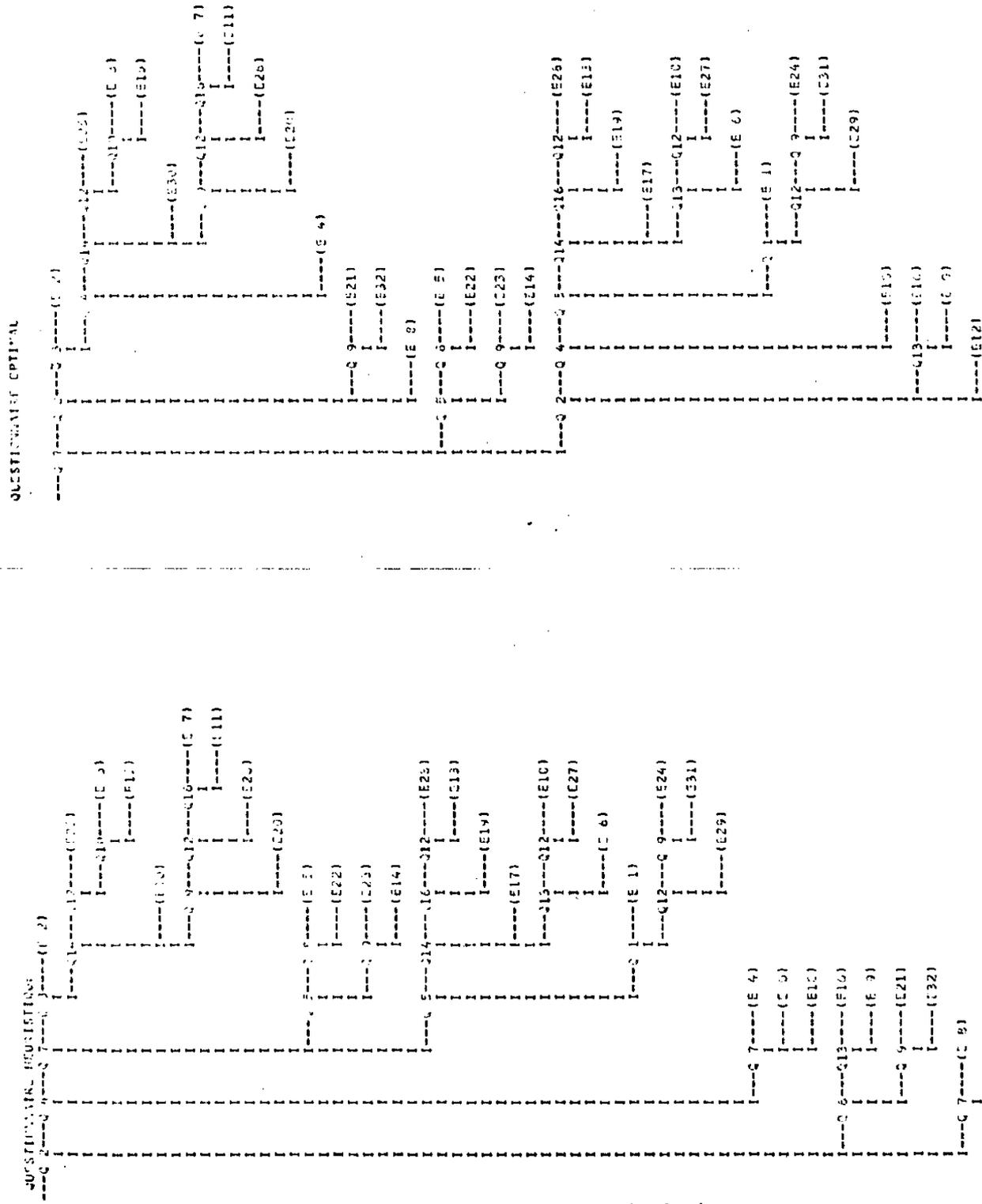


Figure V.11.(suite)

COUT DU QUESTIONNAIRE OPTIMAL = 1001 6476
 NOMBRE D'ALLOCATIONS OPTIMALES FAITES = 4220
 NOMBRE D'APPELS DE LA FONCTION PROGRESSIVE QUESTIONNAIRE = 10074
 TEMPS DE CALCUL DE LA SOLUTION OPTIMALE EN 1/100 S = 10361

COUT DU QUESTIONNAIRE QUESTIONS = 10063411
 TEMPS DE CALCUL DE LA SOLUTION OPTIMALE EN 1/100 S = 62

pas de solution. Dans les résultats précédents, nous avons pris $C_i = C_H$, coût d'une solution heuristique. Or, ce choix n'abaisse pas sensiblement le temps calcul moyen par rapport au choix $C_i \gg C^*$. Cela tient au fait que la recherche "en profondeur" fournit rapidement des solutions complètes. On a tout de même intérêt à choisir $C_i = C_H$ à cause de certains cas de nette amélioration, d'autant qu'une solution heuristique se calcule rapidement.

Pour un sous-ensemble donné, les questions sont examinées dans l'ordre non décroissant des coûts. En guidant les choix successifs des questions par un critère heuristique (voir chapitre VI), on diminue sensiblement le nombre d'appels récursifs, mais on augmente malheureusement le temps de calcul. (Les variations sont respectivement - 10 % et + 60 % au point B courbe n° 1).

Un même type de résultat est obtenu si on examine les issues des questions, en commençant par celles associées aux sous-ensembles F_i^S de plus fort poids.

Ces deux idées ne sont pas à abandonner pour autant : une programmation plus adaptée ou un algorithme différent pourraient peut être les mettre en valeur.

Suggestions pour des recherches ultérieures

— On sait trouver d'une manière non énumérative la solution optimale d'un problème ne comportant que des questions élémentaires (V.1). La courbe n°2 suggère d'utiliser concurremment les deux algorithmes : on détecterait, à l'intérieur de notre procédure, les sous-problèmes n'ayant plus que des questions élémentaires, et ces sous-problèmes seraient traités directement par le second algorithme. Ceci serait utile spécialement pour les problèmes industriels d'inspection (chap. VII) où QE est proche de 1 en général.

— Une procédure de construction amont-aval permet aussi d'utiliser les "éventualités compatibles" (V.1).

— L'évaluation de coûts minimaux de séparation pour des sous-ensembles non encore séparés (II.2.), en réduisant l'énumération, est susceptible de diminuer également le temps de calcul. Ceci est utilisé dans la PSEP programmée par la S.E.M.A. (1970).

— L'algorithme est aisément adaptable en critère du coût maximum : on remplace " $P(F).q_i$ " par " q_i ", et " $C \leftarrow C + SC^*$ " par " $C \leftarrow \text{Max}(C, SC^*)$ " (figure V.9.).

— La version itérative de ce programme récursif abaissera certainement les temps d'exécutions; mais sans pour autant modifier l'allure des courbes 1.2.3.

CHAPITRE VI.

-:-

CONSTRUCTIONS HEURISTIQUES

-

Les études théoriques présentées au chapitre III et l'allure exponentielle de la courbe n° 1 au chapitre V (temps de calcul de solutions optimales en fonction de la taille du problème), soulignent la nécessité de constructions heuristiques pour la réalisation pratique des questionnaires.

Nous appelons heuristique une méthode permettant l'obtention rapide de "bonnes" solutions sous optimales, c'est-à-dire proches de l'optimum. Toute la difficulté consiste à apprécier cette proximité, et à justifier les guides heuristiques utilisés.

C'est ce que nous tenterons de faire, après avoir examiné ce qui existe déjà dans ce domaine, et proposé un nouveau critère heuristique pour la construction de questionnaires.

On se borne, dans ce chapitre, au coût pondéré.

VI.1. PRINCIPES DE LA CONSTRUCTION HEURISTIQUE

Le critère-guide heuristique préconisé habituellement

Le principe général est d'éviter l'énumération et pour cela de ne pas revenir sur des choix partiels déjà effectués.

Pour établir des plans d'inspection, la S.E.M.A. (1970) utilise une construction amont-aval : on développe le questionnaire depuis sa racine jusqu'aux éventualités séparées, en choisissant, pour séparer chaque sous-ensemble d'éventualités, la question maximisant un certain critère. L'algorithme de la définition (I.4. et organigramme figure V.8.a.), dans lequel la fonction CHOIX (F) fournit l'indice de la question qui maximise un critère heuristique, élabore une construction de ce type. Le critère choisi par la S.E.M.A. est le suivant :

$$CR (Q_j) = \frac{1}{q_j} I (p_1, p_2, \dots, p_a)$$

où .les p_j sont les probabilités (renormées) des a classes de la partition effectuée par Q_i sur F

$$p_j > 0, \sum_{1 \leq j \leq a} p_j = 1$$

. a est la base réelle de Q_i sur F

$$.I(p_1, p_2, \dots, p_a) = \sum_{1 \leq j \leq a} p_j \log \frac{1}{p_j}$$

La fonction $I(p_1, p_2, \dots, p_a)$ est appelée "fonction d'information de SHANNON". Le maximum de I est atteint lorsque les p_j sont égaux et vaut $\log N$. La base du logarithme doit être la même pour toutes les questions.

L'idée d'un critère de la forme I/q est employée également par S.GANAPATHY et V.RAJARAMAN (1973), dans le contexte de la traduction des tables de décisions.

Les propriétés de la fonction d'information de SHANNON, liées à la longueur de cheminement dans un questionnaire libre, ont été mises en évidence par C.F. PICARD (1965, puis 1972, tome 2).

Dans les deux emplois cités du critère I/q , il n'y a pas de justification mathématique à proprement parler. Voici cependant une interprétation possible :

Une interprétation du critère heuristique I/q

- soit un questionnaire inachevé (figure VI.1.), effectuant sur E la partition A . Soit à séparer le bloc F de A .

Si pour cela nous utilisons la question Q_i , cela induit sur F une partition F^i , et A est transformée en une partition plus fine notée A_F^i .

Appelons $C(A)$ le coût déjà dépensé pour effectuer A , somme des coûts pondérés des questions utilisées. On a :

$$C(A_F^i) = C(A) + q_i \cdot P(F) \quad (1)$$

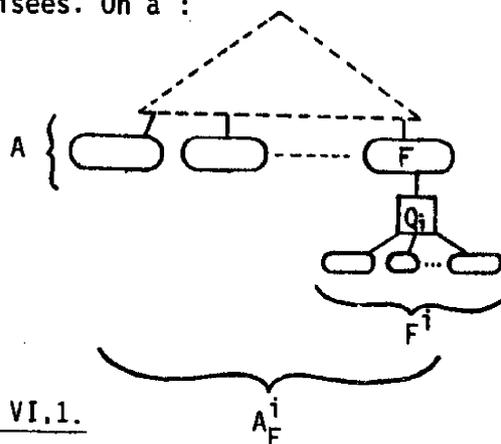


Figure VI.1.

- Soient (p_1, p_2, \dots, p_N) les poids des blocs de la partition A ; notons abusivement $I(A) = I(p_1, p_2, \dots, p_N)$. Une propriété fondamentale de la fonction I est : (additivité et pondération)

$$I(p_1, p_2, \dots, p_N \cdot t, p_N(1-t)) = I(p_1, p_2, \dots, p_N) + p_N \cdot I(t, 1-t).$$

Cette propriété se généralise facilement au cas où on partage p_N en plusieurs morceaux : soit $\alpha + \beta + \dots + \lambda = 1$

on a aussi : $I(p_1, p_2, \dots, p_{N-1}, p_N \cdot \alpha, p_N \cdot \beta, \dots, p_N \cdot \lambda)$

$$= I(p_1, p_2, \dots, p_N) + p_N \cdot I(\alpha, \beta, \dots, \lambda).$$

Ce qui entraîne, dans notre construction :

$$I(A_F^i) = I(A) + I(F^i) \cdot P(F) \tag{2}$$

C.F. PICARD appelle $I(F^i)$: information traitée par Q_i

$I(F^i) \cdot P(F)$: information transmise par Q_i

et $I(e_1, e_2, \dots, e_{|E|})$ = information transmise par le questionnaire. Lorsque, après des questions successives telles Q_i , le questionnaire est complété, $I(A_F^i)$ vaut $I(\{E\}) = I(e_1, e_2, \dots, e_{|E|})$. (On retrouve un résultat de C.F. PICARD : la somme des informations transmises par les questions est égale à l'information transmise par le questionnaire). Remarquons que $C(A_F^i)$ dépend du choix des questions, mais $I(A_F^i)$ uniquement de la partition A_F^i .

- Les expressions (1) et (2) ont une forme analogue.

Formons le rapport :

$$\frac{C(A_F^i)}{I(A_F^i)} = \frac{C(A) + q_i \cdot P(F)}{I(A) + I(F^i) \cdot P(F)} \tag{3}$$

qui caractérise le questionnaire au fur et à mesure de sa construction. Alors que $C(A_F^i)$ converge vers $C(\{E\})$, coût du questionnaire, non connu à l'avance, $I(A_F^i)$ converge vers $I(\{E\})$ connu a priori. Or, au niveau du choix de Q_i , la forme de (3) montre que le plus petit accroissement du rapport C/I est obtenu pour la question assurant le plus petit $q_i/I(F^i)$, justifiant ainsi ce critère heuristique.

Autres fonctions possédant la propriété d'additivité

A vrai dire, il est possible de construire des fonctions de forme analogue à (2), de valeur fixe pour tout questionnaire achevé.

Exemples :

. $S(A)$ = somme des poids des éventualités isolées dans la partition A

$$S(A_F^i) = S(A) + S(F^i)$$

$$\text{et } S(\{E\}) = \sum_{E_j \in E^i} e_j$$

le critère à maximiser serait : $S(F^i) / q_i$

. $N(A)$ = nombre d'éventualités isolées dans A :

$$N(A_F^i) = N(A) + N(F^i) ; N(\{E\}) = |E| ; \text{critère : } N(F^i)/q_i$$

. des combinaisons linéaires des fonctions précédentes, par exemple :

$$\lambda I(A_F^i) + \mu S(A_F^i) = \lambda I(A) + \mu S(A) + \lambda I(F^i) \cdot P(F) + \mu S(F^i).$$

D'où le critère local à maximiser :

$$\frac{1}{q_i} \left(\lambda I(F^i) + \mu \frac{S(F^i)}{P(F)} \right)$$

Remarquons que les critères issus de S et N ne visent pas à partager "moitié-moitié", comme I; dans la pratique, elles ne sont pas assez discriminantes. Les résultats expérimentaux obtenus pour ces critères y compris les combinaisons linéaires, sont moins bons que ceux obtenus avec le critère de l'information de SHANNON.

Est-il possible de construire un questionnaire heuristique d'aval en amont, comme dans l'algorithme de HUFFMANN ?

On se heurtera toujours à des problèmes de sémantique si on s'interdit les retours en arrière sur les choix déjà faits. Il semble donc que la construction aval-amont soit impraticable pour les heuristiques.

VI.2. CONSTRUCTION D'UN CRITERE HEURISTIQUE

Nous voulons construire un critère-guide à maximiser, de la forme $CR(Q_i) = \frac{1}{q_i} J(p_1, p_2, \dots, p_a)$; avec J fonction symétrique des p_j , probabilités (renormées) des blocs de la partition effectuée par Q_i sur le sous-ensemble à séparer.

La fonction J que nous allons élaborer ne possède pas la propriété d'additivité (2). Sa justification sera établie d'une autre façon, en deux étapes.

Le principe conducteur est de rechercher les propriétés locales d'une situation globale optimale, et d'essayer de les traduire par le choix de la fonction J.

Choix de la fonction $J(p_1, p_2)$ pour les questions de base 2

Avant d'effectuer un classement entre les questions candidates par le moyen du critère, nous les comparons d'abord 2 à 2.

En se restreignant à deux questions Q_1 et Q_2 , pour séparer E, deux cas sont possibles :

- 1) le produit des partitions a 4 blocs, auquel cas les questionnaires possibles sont tous de coût $q_1 + q_2$ et l'ordre des questions est indifférent.
- 2) le produit des partitions a 3 blocs, et nous pouvons échanger les questions (V.2.).

Nous sommes donc amenés à examiner les problèmes du type suivant :

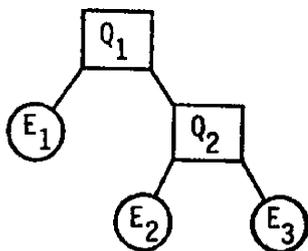
$$\begin{array}{lll}
 E = \{E_1, E_2, E_3\} & e = \{\alpha, \beta, \gamma\} & \alpha + \beta + \gamma = 1 \\
 Q = \{Q_1, Q_2\} & q = \{u, v\} & Q_1 = \{\{E_1\}, \{E_2, E_3\}\} \\
 & & Q_2 = \{\{E_2\}, \{E_1, E_3\}\}
 \end{array}$$

pouvant être résumés par la table d'identification :

		E_1 (α)	E_2 (β)	E_3 (γ)
Q_1	(u)	1	2	2
Q_2	(v)	2	1	2

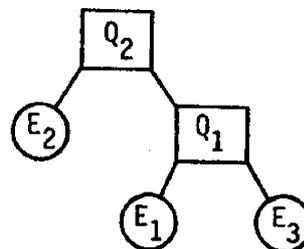
Ce problème se pose réellement à la fin des constructions des questionnaires, lorsque les sous-ensembles à séparer n'ont plus que 3 éventualités, ce qui arrive assez souvent.

Deux questionnaires sont possibles :



Questionnaire U

coût : $u + v (\beta + \gamma)$



Questionnaire V

coût : $v + u (\alpha + \gamma)$

Posons : $t = \frac{u}{v}$; si $\frac{\alpha}{\beta} \geq t$ l'arbre U est optimal
 si $\frac{\alpha}{\beta} \leq t$ l'arbre V est optimal.

Dans des axes de coordonnées (α, β) (figure a), un problème de ce type est représenté par un point $M(\alpha, \beta)$ du triangle OAB et la droite OT de pente t. Cette droite partage le triangle OAB en deux triangles :

$\left\{ \begin{array}{l} M \in OAT \iff \text{l'arbre U est optimal} \\ M \in OBT \iff \text{l'arbre V est optimal} \end{array} \right.$ (figure b)

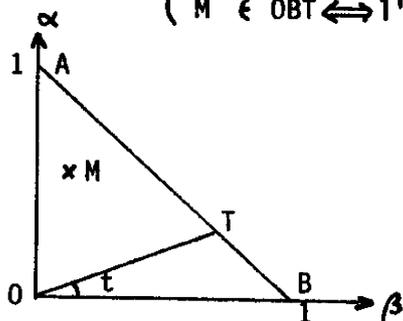


Figure a

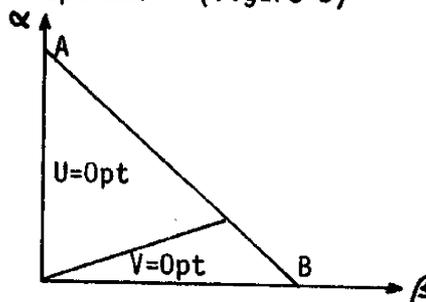


Figure b

Supposons choisi un critère heuristique. Nous pouvons aussi partager le triangle OAB en deux régions R_1 et R_2 telles que :

$\left\{ \begin{array}{l} M \in R_1 \iff \text{U est le questionnaire heuristique} \\ M \in R_2 \iff \text{V est le questionnaire heuristique.} \end{array} \right.$

Dans le cas du critère fourni par la théorie de l'information au sens de SHANNON, et en supposant $t \leq 1$ (sans nuire à la généralité de l'exposé), on montre que ces deux régions sont partagées par une courbe

OB symétrique par rapport à l'axe d ($\beta = 0,5$) et tangente en O à OT, en B à BS (figure c) et entièrement située sous OSB. Les régions R_1 et R_2 sont indiquées sur la figure d. La démonstration est reportée en fin de paragraphe *.

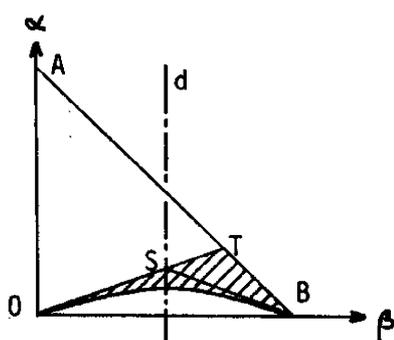


Figure c

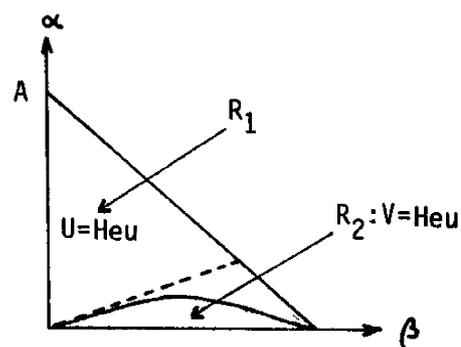


Figure d

On a $R_2 \subset OTB$. En comparant les figures b et d, on s'aperçoit que pour les problèmes tels que $M \in OTB \cap R_1$ (région hachurée figure c) les questionnaires heuristiques ne sont pas optimaux.

Pour tous les critères tels que la fonction J est symétrique, on montre que la courbe séparatrice des régions R_1 et R_2 est symétrique par rapport à l'axe d. Un meilleur critère heuristique pour ce type de problème simplifié est obtenu lorsque la région R_2 se confond avec le triangle OBS (figure c) ; on peut vérifier que ce critère est défini par :

$$J(p_1, p_2) = \min (p_1, p_2)$$

La région pour laquelle le plan heuristique n'est pas optimal est réduite au triangle BTS.

On déduit également de cette étude, qu'avec une fonction J symétrique, on peut toujours fabriquer des problèmes pour lesquels le questionnaire heuristique n'est pas optimal.

En utilisant cette fonction J dans le cas général des problèmes avec questions de base 2, les résultats "expérimentaux" sont les suivants :

NSO = nombre de solutions heuristiques qui sont optimales sur les 100 problèmes engendrés (voir V.3. pour le générateur)

R = moyenne du rapport $(C_H - C^*) / C^*$; C_H : coût heuristique.
 C^* : coût optimal.

	Q = E = 10		Q = E = 15	
	NSO	R en %	NSO	R en %
$J(p_1, p_2) = \sum_{i=1}^2 p_i \log \frac{1}{p_i}$	36	1,6	10	1,7
$J(p_1, p_2) = \min(p_1, p_2)$	66	0,6	43	0,6

Le nouveau critère est meilleur que celui issu de la notion d'information au sens de SHANNON, bien que relativement proches.

*Démonstration de la figure c.

— On pose $H(x) = x \log \frac{1}{x} + (1-x) \log \frac{1}{1-x}$. La frontière entre R_1 et R_2 est composée de points (α, β) tels que $H(\alpha) = t H(\beta)$, avec $t \leq 1$; soit L le lieu de ces points. La fonction $H(x)$ est symétrique par rapport à la droite $x = \frac{1}{2}$ (figure VI.2.), c'est-à-dire que $H(x) = H(1-x)$. On en déduit que si $(\alpha, \beta) \in L$, alors $(1-\alpha, \beta) \in L$, de même $(\alpha, 1-\beta)$ et $(1-\alpha, 1-\beta)$. Il nous suffit donc d'étudier ce lieu pour $0 < \alpha \leq \frac{1}{2}$ et $0 < \beta \leq \frac{1}{2}$.

Sur l'intervalle $]0, \frac{1}{2}]$ $H(x)$ est inversible. Pour β fixé dans $]0, \frac{1}{2}]$, $t H(\beta) < \log 2$ et α , tel que $H(\alpha) = t H(\beta)$ est unique, dans $]0, \frac{1}{2}]$.

— Montrons que $\forall (\alpha, \beta) \in L$, on a $\alpha \leq t\beta$:
 $H(x)$ est concave ; $t H(x) \leq H(tx)$; donc $H(\alpha) \leq H(t\beta)$ et comme H est croissante monotone sur $]0, \frac{1}{2}]$, alors $\alpha \leq t\beta$; la courbe séparatrice des régions R_1 et R_2 est bien située toute entière dans le triangle OBS. A l'évidence, 0 fait partie du lieu L, par continuité.

— Montrons que L est tangent en 0 à OT.

Soit $f(x)$ la fonction implicitement définie par

$H(f(x)) = t H(x)$, $0 < t \leq 1$, $f(x)$ et $x \in]0, \frac{1}{2}]$. On a $t H(x) \in]0, \log 2]$, et H est inversible sur cet intervalle : $f(x) = H^{-1}(t H(x))$. Donc $f(x)$ est continue, dérivable ; $f'(x) = t \cdot g(x)$, avec $g(x) = H'(x)/H'[t H(x)]$. On cherche $\lim_{x \rightarrow 0} g(x)$. Les $\mathcal{E}_1(x)$ qu'on va construire tendent vers 0 pour $x \rightarrow 0^+$.

$$H'(x) = \log\left(\frac{1-x}{x}\right) = -\log x + \mathcal{E}_1(x) ; t H(x) \rightarrow 0 \text{ pour } x \rightarrow 0$$

$$\text{donc : } H'[t H(x)] = -\log[t H(x)] + \mathcal{E}_2(x) ; g(x) = \frac{-\log x + \mathcal{E}_1(x)}{-\log t - \log H(x) + \mathcal{E}_2(x)}$$

$$g(x) \text{ a la même limite que } \frac{\log(x)}{\log H(x)} ; H(x) = -x [-\log x - 1 + \mathcal{E}_3(x)] ;$$

$$\text{on pose } X = -\log x \rightarrow +\infty \text{ pour } x \rightarrow 0^+ ; H \log H(x) = -X + \log[X-1 + \mathcal{E}_3(x)] .$$

$$\frac{\log H(x)}{\log x} = 1 - \frac{\log[X-1 + \mathcal{E}_3(x)]}{X} \rightarrow 1 \text{ pour } X \rightarrow \infty ; \text{ donc } f'(x) \rightarrow t \text{ pour } x \rightarrow 0^+ \square$$

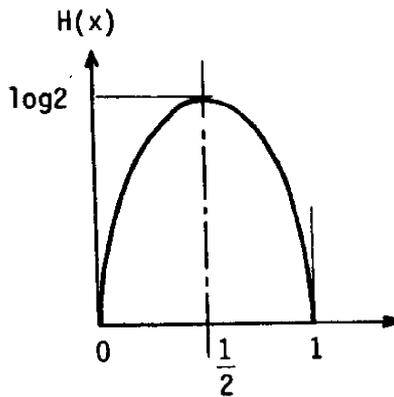


Figure VI.2.

Généralisation pour des bases supérieures à 2

Si on généralise le nouveau critère par $J(p_1, p_2, \dots, p_a) = \min_{1 \leq i \leq a} p_i$, on obtient de mauvais résultats lorsqu'on emploie des questions de bases différentes. En effet, on a alors $\max J = \frac{1}{a}$; or, à coût égal, lorsque la base des questions augmente, la profondeur de l'arbre diminue et le coût de l'arbre aussi. Un critère heuristique doit donc favoriser les questions de bases supérieures.

Il nous faut trouver un moyen de mesurer l'efficacité d'une question de base a par rapport à une question de base b.

La situation globale optimale dont nous voulons extraire les propriétés locales sera fournie par la séparation optimale de N éventualités équiprobables par des questions de base a (ou inférieure à a), de coût 1. PICARD (1972) a montré que l'arbre optimal de séparation a un coût $L_a(N) = \log_a N + \mathcal{E}(a, N)$, avec $\mathcal{E}(a, N)$ positif et d'ordre inférieur à $\log_a N$ pour N croissant.

Donc $\frac{L_a(N)}{L_b(N)}$ tend vers $\frac{\log_a N}{\log_b N} = \frac{\ln b}{\ln a}$ pour $N \rightarrow \infty$

Voici quelques valeurs pour $a = 2, b = 3 ; \frac{\ln 3}{\ln 2} = 1,58$

N	3	4	6.....9	10.....100
$\frac{L_2(N)}{L_3(N)}$	1,66	1,33	1,45....1,61	1,55....1,57

Ces arbres optimaux ont la propriété locale d'avoir des questions "équilibrées", c'est-à-dire que les poids attachés aux issues d'une question tendent vers l'égalité. On conservera cette propriété pour la généralisation :

$$\hat{J}_b = \max J(p_1, \dots, p_b) = J\left(\frac{1}{b}, \frac{1}{b}, \dots, \frac{1}{b}\right)$$

— Entre un arbre optimal A de questions de bases a, coûts C_a et un arbre optimal B de questions de bases b, coûts C_b

le rapport des coûts vaut : $\frac{C_a L_a(N)}{C_b L_b(N)}$ et tend vers $\frac{C_a}{C_b} \cdot \frac{\ln b}{\ln a}$

La proposition "A est meilleur que B" se traduit au niveau global par :

$$\frac{C_a}{C_b} < \frac{\ln a}{\ln b} \quad (\text{pour } N \text{ assez grand}) \quad (5)$$

Au niveau local, nous voulons traduire cette proposition par:

$$\frac{\hat{J}_a}{C_a} > \frac{\hat{J}_b}{C_b} ; \text{ soit } \frac{C_a}{C_b} < \frac{\hat{J}_a}{\hat{J}_b} \quad (6)$$

pour que (5) et (6) soient compatibles, il est donc souhaitable que :

$$\frac{\hat{J}_a}{\hat{J}_b} = \frac{\ln a}{\ln b} \quad (7)$$

— On remarquera que pour le critère $J = \sum p_i \log_2 \frac{1}{p_i}$ de l'information, on a $\hat{J}_b = \log_2 b$ et la condition (7) se trouve bien réalisée.

— Il nous faut trouver une fonction J vérifiant (7) et égale à $k \cdot \min(p_1, p_2)$ pour $b = 2$; deux voies sont possibles :

$$J(p_1, p_2, \dots, p_b) = b \log b \min_{1 \leq i \leq b} p_i$$

$$\text{ou } J(p_1, p_2, \dots, p_b) = \frac{b \log b}{b-1} (1 - \max_{1 \leq i \leq b} p_i)$$

Les meilleurs résultats "expérimentaux" sont obtenus avec la seconde fonction.

— La courbe 4 compare les performances des deux critères

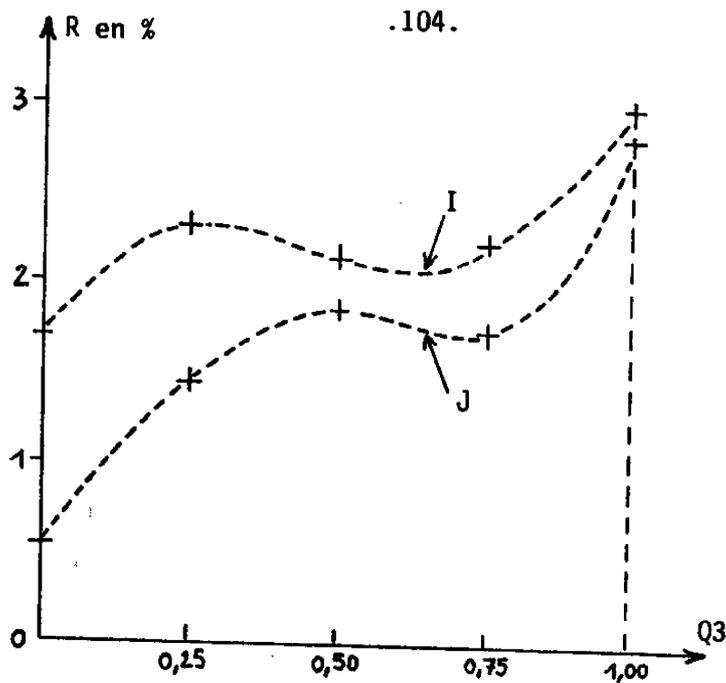
$$I = \sum_{1 \leq i \leq b} p_i \log \frac{1}{p_i}$$

$$J = (1 - \max_{1 \leq i \leq b} p_i) \frac{b \log b}{b-1}$$

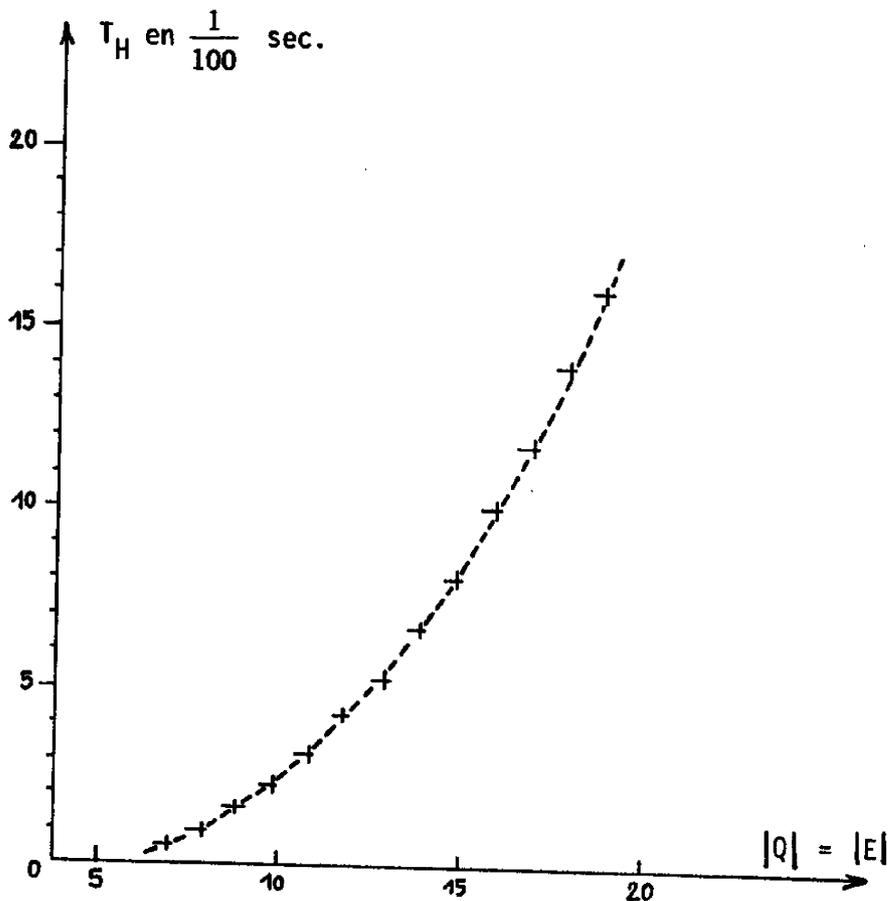
et donne, pour $|Q| = |E| = 15$, en fonction de la proportion Q3 de questions de base 3, la valeur de la moyenne du rapport $R = \frac{C_H - C^*}{C^*}$ (sur 100 problèmes engendrés).

La courbe 5 donne les temps de calcul des solutions heuristiques, en 1/100 sec. pour des problèmes avec questions de base 2.

— Une résolution heuristique nécessite au plus le calcul de $|Q|$ critères en chacun des $m \leq |E| - 1$ noeuds intérieurs de l'arbre, et pour calculer chaque critère, on parcourt les $|E|$ éventualités. Le temps de calcul est donc de la forme $T_H \leq k \cdot |Q| \cdot |E|^2$; expérimentalement, pour $|Q| = |E| = 14$ on a $\frac{T_H}{T_{OPT}} = 8,5 \%$.



Courbe n° 4. Proximité relative des solutions heuristiques aux solutions optimales en fonction de la proportion de tests de base 3, pour les critères I et J - $|Q| = |E| = 15$.



Courbe n° 5. Temps de calcul des solutions heuristiques en fonction de la taille du problème.

La figure VI.3. représente un questionnaire heuristique obtenu par le critère J, pour un problème de plan d'inspection pris dans le rapport S.E.M.A. (1970). Notons que, si le coût obtenu par le nouveau critère J est meilleur que celui obtenu par le critère I de la théorie de l'information, la recherche d'une solution optimale a été abandonnée également après 5 minutes de calcul.

Pour comparer deux questionnaires, l'un heuristique, l'autre optimal, on se reportera à la figure V.11. (problème de taille 32 x 32).

VI.3. COMPARAISON THEORIQUE DES COÛTS OPTIMAUX ET HEURISTIQUES

Expérimentalement, la différence relative $\frac{C_H - C^*}{C^*}$ est toujours de l'ordre de quelques pour cent pour les problèmes engendrés pseudo-aléatoirement (voir la courbe 4). Dans quelles limites théoriques le rapport $\frac{C_H}{C^*}$ peut-il croître en fonction de la taille du problème ? Ceci paraît difficile à résoudre. Cependant, M.R. GAREY et R.L. GRAHAM (1974) ont démontré que dans le cas des questions de coût 1, et avec une heuristique qui partage aussi également que possible les poids attachés aux issues (c'est le cas des fonctions I et J), $\frac{C_H}{C^*}$ ne peut être borné par une constante. Plus précisément, il existe des problèmes pour lesquels :

$$\frac{C_H}{C^*} > \frac{1}{10} \cdot \left[\frac{\log_2 |E|}{\log_2 \log_2 |E|} \right]$$

L'ordre de croissance du maximum de ce rapport, en fonction de la taille du problème, n'est pas encore connu.

Le problème semble encore plus difficile dans le cas général des coûts distincts. Pour la table d'identification simplifiée discutée en VI.2. avec $|E|=3$, nous pouvons montrer cependant :

$$\frac{C_H}{C^*} < 2, \text{ pour notre critère } \frac{1}{q_i} J(p_1, p_2), J(p_1, p_2) = \min(p_1, p_2) :$$

— Dans la région où le questionnaire heuristique n'est pas optimal (triangle STB, figure c), on a $R = \frac{C_H}{C^*} = \frac{u + v(1-\alpha)}{v + u(1-\beta)} = \frac{t + 1 - \alpha}{1 + t(1-\beta)}$.

Pour t et β fixés, R croît lorsque α décroît.

α décroît jusqu'à ce que le point $M(\alpha, \beta)$ soit sur le segment SB :

$$\alpha = t(1-\beta), \text{ et alors } R = \frac{t+1-t(1-\beta)}{1+t(1-\beta)}. \text{ Lorsque } \beta \rightarrow 1, \text{ alors}$$

$R \rightarrow 1+t$ en croissant.

Conclusion : $\frac{C_H}{C^*} < 1 + t \leq 2 \quad \square$

VI.4. METHODES HEURISTIQUES SEMI-ENUMERATIVES

On peut obtenir des solutions sous-optimales encore meilleures, en utilisant des méthodes mixtes issues des études sur les procédures optimales (chapitre V), et sur les algorithmes heuristiques ici présentés, mais au prix d'un temps de calcul plus long. Remarquons en effet que, dans l'algorithme de construction optimale, toutes les questions sont "essayées" pour chaque sous-problème rencontré, alors que dans les constructions heuristiques, une seule question est directement sélectionnée grâce au critère. L'idée naturelle est de choisir un moyen terme entre ces deux solutions extrêmes, en limitant le nombre de questions essayées pour chaque sous-problème à 2 ou 3, ces questions étant sélectionnées par un critère heuristique.

C'est ce que nous avons expérimenté, en reprenant l'algorithme optimal du chapitre V, auquel très peu de modifications sont nécessaires afin d'obtenir cette méthode heuristique semi-énumérative.

Les résultats obtenus sont consignés dans le tableau suivant qui regroupe également les valeurs correspondantes des algorithmes optimaux et heuristiques, afin de faciliter la comparaison :

		n	NSO	R en %	T en $\frac{1}{100}$ sec.
algorithmes heuristiques (critère J)	pur	1	43	0,6	8
	semi- énumératif	2	88	0,04	21
		3	97	0,005	38
algorithme optimal		toutes questions	100	0	141

Les essais portent sur un lot de 100 problèmes engendrés pseudo-aléatoirement, de taille $|Q| = |E| = 15$.

n = nombre de questions "essayées" à chaque sous-problème.

NSO = nombre de solutions, parmi les 100 obtenues, qui sont optimales.

R = moyenne du rapport $\frac{C_H - C^*}{C^*}$

T = temps de calcul moyen d'une solution sur IBM 370/168.

De ces résultats, il faut retenir qu'en un quart du temps de calcul des solutions optimales, on peut obtenir par la méthode semi-énumérative, des solutions heuristiques qui sont 97 fois sur 100 optimales.

CONCLUSION

Nous avons dans ce chapitre étudié différents critères-guides heuristiques pour la construction directe de questionnaires, sans retours en arrière. La plupart de ces critères fournissent des solutions très proches de l'optimum. Nous avons construit cependant un critère heuristique qui semble statistiquement plus robuste que celui issu de la théorie de l'information préconisé habituellement. Les propriétés informationnelles de la fonction J construite seraient à étudier.

Un compromis très intéressant a été mis en évidence, entre les algorithmes heuristiques purs sans "backtracking", et les procédures optimales où toutes les questions doivent être essayées (implicitement au moins) à chaque étape. Ces méthodes heuristiques semi-énumératives devraient pouvoir être encore perfectionnées ; par exemple, en modulant le nombre de questions essayées, en fonction :

- du niveau dans l'arbre
- de la proximité relative des critères obtenus pour les meilleures questions
- du temps attribué au calcul, ou déjà écoulé.

3ÈME PARTIE

APPLICATIONS

INTRODUCTION

L'approche des questionnaires réalisables est susceptible de s'appliquer à de nombreux problèmes pratiques d'identification : diagnostic d'incidents, tests logiques de circuits, diagnostic médical, traduction des tables de décision, reconnaissance des formes, tris combinatoires,...

Ces problèmes ont en commun :

- une expression sous forme logique (par opposition à stochastique), conséquence d'un comportement déterministe sous-jacent ;
- une redondance de l'information, disponible sous forme "parallèle" (la table d'identification) dans laquelle on cherche à opérer une sélection séquentielle.

CHAPITRE VII.

-:-

DIAGNOSTIC D'INCIDENTS RÉPERTORIÉS

-

Ce problème a déjà été modélisé sous la forme questionnaire réalisable par la S.E.M.A. (1970) sous le nom "établissement de plans d'inspection". Nous appuyant sur l'étude théorique exposée précédemment, nous avons repris cette idée dans le cadre de l'opération "Automatisation Intégrée", visant au "contrôle et à la conduite de chaînes de fabrication de dispositifs à semiconducteurs" (L.A.A.S. 1973,1974) poursuivie au L.A.A.S. en collaboration avec les industriels intéressés.

Il s'agit d'automatiser, ou d'aider le diagnostic de pannes de systèmes industriels.

VII.1. LE MODELE

Le document de base pour le diagnostic est le "catalogue d'incidents" : on y trouve la liste de tous les incidents qui peuvent se produire sur tout ou partie du système.

En regard de la description de chaque incident, on peut lire une liste E de causes possibles (éventualités) pour expliquer l'incident, et une liste Q de tests (questions) praticables, avec leurs issues possibles. Les causes sont affectées de probabilités et les tests affectés de coûts exprimés en monnaie ou en temps.

Lorsqu'un incident se produit, le modèle suppose qu'il est dû à une cause et une seule de l'ensemble E qui lui correspond dans le catalogue. Au cours de la réalisation matérielle d'un test, il se produit un événement qu'on associe à une issue et une seule. Enfin, les relations causes-tests sont données par une matrice analogue à une table d'identification, décrivant les partitions effectuées par les tests sur l'ensemble des causes possibles.

Si on introduit la panne particulière correspondant au fonctionnement correct du système, on transforme le problème du diagnostic en problème de maintenance.

VII.2. DISCUSSION DU MODELE

Les causes sont considérées comme exclusives. Mais, on peut toujours faire figurer dans la liste E, des causes correspondant à des pannes multiples, si on juge leur probabilité non négligeable. Exemple : soit p_1 et p_2 deux pannes distinctes éventuelles, auxquelles on associe les deux causes E_1 et E_2 . E_1 signifie "panne p_1 seulement". La panne multiple $p_1 \wedge p_2$ sera associée à une troisième cause E_3 , distincte des autres.

Nous cherchons à utiliser au mieux l'information contenue dans la table d'identification, de façon à faire un diagnostic au moindre coût. Aussi, alors que certaines causes seront "caractérisées" par un test spécifique, d'autres seront seulement isolées par "élimination". Dans la réalité, il se peut qu'une cause ne figurant pas dans E soit la véritable origine de l'incident. Le questionnaire obtenu étant déterministe, on isolera toujours une cause de E.

Faire un diagnostic de façon que toute cause isolée soit "caractérisée" suppose un autre modèle beaucoup plus simple, par exemple celui de W.W. CHU (1967), dans lequel tout test Q_i est élémentaire et caractéristique de la cause E_i . Il suffit alors d'ordonner l'application de ces tests dans l'ordre des rapports $\frac{e_i}{q_i}$ non croissants pour obtenir un diagnostic optimal. On doit bien distinguer les deux modèles.

En principe un test doit être considéré comme une "prise d'information" sur le système, et ne doit pas le modifier. Cependant, on peut admettre des tests modifiant le système (en particulier pour des tests élémentaires et destructifs) à condition que la modification n'entraîne pas de changement dans la partition des autres tests. On pourra toutefois contourner la difficulté en fixant des contraintes dans l'ordre d'application des tests.

D'une façon générale, on pourrait définir la "faisabilité" d'un test comme fonction logique de l'application d'autres tests.

Plus précisément :

A chaque étape, définissons

$x_i = 1$ si le test n° i a été effectué

$x_i = 0$ sinon

puis

$y_m = 1$ si le test n° m-est applicable

$y_m = 0$ sinon.

On aurait

$$y_m = f_m(x_i, x_j, \dots, x_1)$$

f_m : fonction de variables logiques.

Notre modèle suppose qu'une même cause provoque toujours la même issue d'un même test. Nous avons écarté tous les tests ne possédant pas cette propriété.

Il peut être intéressant d'utiliser pourtant un genre de test dont l'issue, pour une même cause, serait plus ou moins sûre. Il peut arriver, pour certains incidents, que ce soit le seul type de test disponible. On est ramené alors au modèle des pseudo-questionnaires présenté en I.5.

VII.3. ETABLISSEMENT DU CATALOGUE D'INCIDENTS

Nous avons recherché une méthode systématique de recueil des données, permettant d'établir le catalogue d'incident. Pour cela, nous nous sommes appuyés sur la structure particulière de chaîne du système à diagnostiquer.

Tout d'abord, un incident est caractérisé par une sortie ou une combinaison de sorties hors normes pour des tests effectués de façon systématique sur la chaîne (mesures, points de contrôle...). La figure VII.1., tirée du rapport L.A.A.S. (1973, partie I), représente en haut, la structure de chaîne (début). Les opérations successives sont numérotées en séquence, les liaisons indiquées correspondent aux variables entrant en jeu, et/ou modifiées en cours de traitement.

Pour dresser un catalogue d'incidents, on fera la liste chronologique des tests systématiques. Chaque test ou groupe de tests ainsi effectué à un endroit précis de la chaîne détermine un ou plusieurs incidents (partie intermédiaire de la figure).

Pour chaque incident mis en évidence, on dresse une liste de causes pouvant l'expliquer, aussi exhaustive que possible E, avec leurs

Causes possibles d'incidents

Mesures systématiques

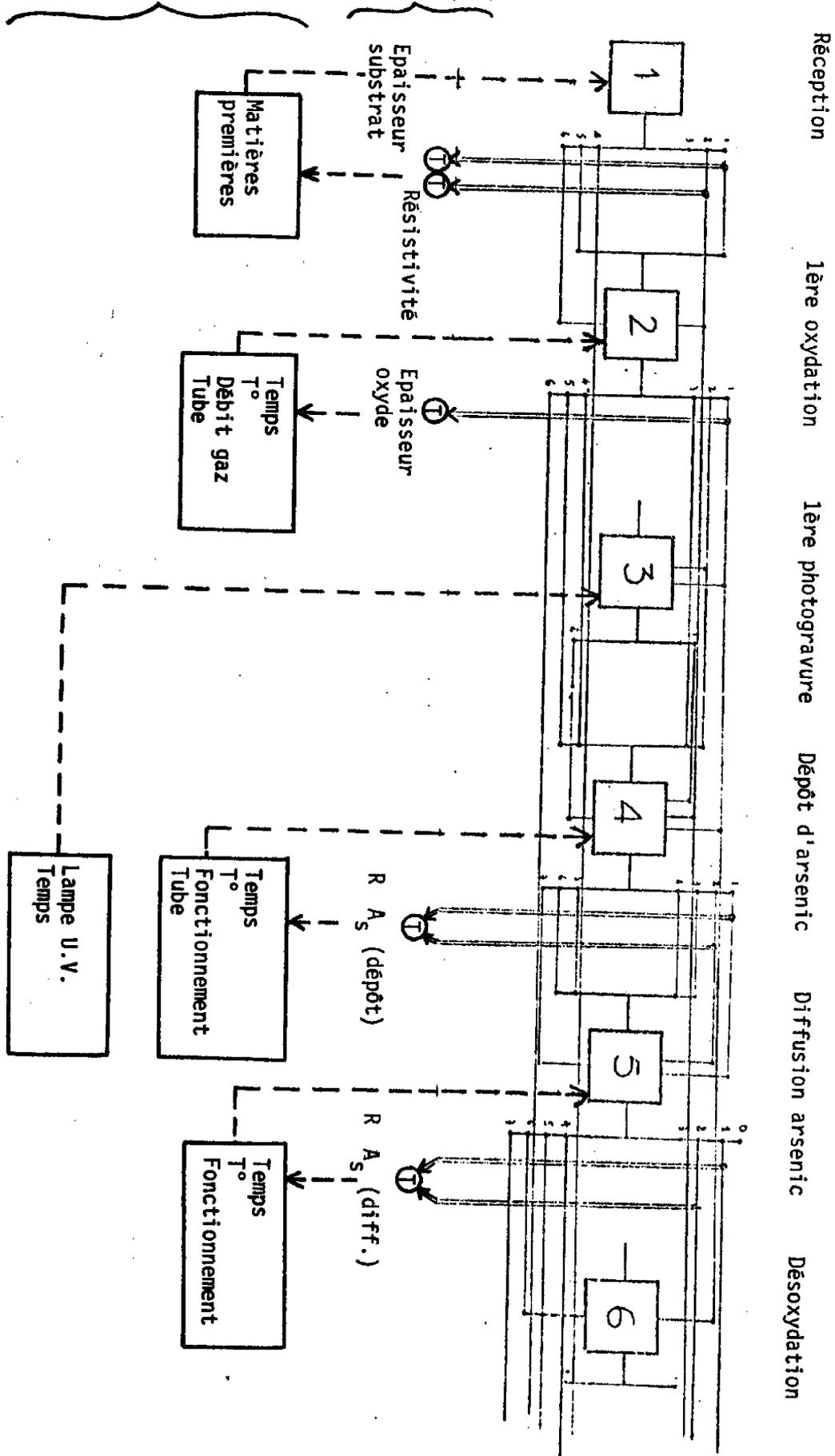


Figure VII.1.

probabilités (e) (partie inférieure de la figure) ; puis une liste de tous les tests complémentaires imaginables Q pouvant séparer E, avec leur coûts. Il est utile de remarquer que les tests de Q ne doivent pas en principe comporter de tests systématiques, mais uniquement complémentaires.

La cause est à rechercher évidemment en aval du test systématique ayant servi à caractériser l'incident. L'analyse structurelle de la chaîne sera utile pour effectuer une recherche aussi complète que possible des causes.

Cette façon de procéder utilise au mieux la structure série du processus. On doit tendre vers un découpage de la chaîne en tronçons dont les bornes seraient les points de contrôle, comme c'est le cas figure VII.1.

VII.4. IMPLANTATION EN SITE INDUSTRIEL

Nous avons écrit un programme de diagnostic en FORTRAN, pour être implanté en site industriel. Ce programme, décrit dans L.A.A.S. (1973, partie II), possède les caractéristiques suivantes :

- L'algorithme lui-même est entouré de sous-programmes utilitaires, notamment :
 - .stockage du catalogue sur disque
 - .routines d'entrées-sorties à partir d'un télétype
 - .comptabilité, éditions diverses.
- On peut compléter et remettre à jour le catalogue d'incidents ; les probabilités peuvent être modifiées.
- Il est possible de réajuster périodiquement les probabilités, sur un modèle de type bayésien (S.E.M.A., 1970, note n° 1), en fonction d'incidents survenus lors de la dernière période, avec pondération variable du passé.
- Il est possible de fixer des contraintes sur l'ordre d'exécution des tests, en fournissant comme donnée supplémentaire au programme une matrice de priorités MPRIO :
$$\text{MPRIO}(I,J) = 1 \iff \text{le test } n^{\circ}J \text{ ne peut être effectué qu'après le test } I.$$
 Si il existe I et I' tels que $\text{MPRIO}(I,J) = \text{MPRIO}(I',J) = 1$, alors J ne peut être effectué que si I et I' sont faits.

- Des tests de base quelconque peuvent être utilisés (en fait jusqu'à 5 mais ceci est modifiable).
- Le programme est implanté sur des calculateurs de taille moyenne ; il est utilisé à chaque fois qu'un incident est signalé, et doit fournir une solution rapidement. Pour ces deux raisons, nous avons bien sûr opté pour une procédure heuristique.
- Lorsqu'on utilise un plan d'inspection, résultat du programme, un seul chemin est suivi depuis le premier test appliqué jusqu'à la cause isolée. Beaucoup de sommets ne sont pas utilisés. On pourrait donc s'épargner du temps de calcul en soumettant les résultats test par test, et en les appliquant au fur et à mesure, au lieu de fournir l'arbre tout entier. Mais, les temps de calcul des solutions heuristiques nous permettent de proposer ce plan en entier, recalculé à chaque incident. De plus, une exploitation "pas à pas" serait ressentie comme trop directive. Le plan proposé doit rester, dans ce type d'application, une indication. Le responsable de la recherche de la panne doit conserver toute liberté d'action.

Exemples d'incidents

Le premier document est un formulaire en deux feuilles permettant la description complète d'un incident, avant mise sous cartes perforées. C'est une page du catalogue d'incidents.

Ici, le formulaire est rempli selon des données recueillies au L.A.A.S.

Le second document montre le traitement d'un incident réel figurant sur les catalogues R.T.C. (seuls figurent la matrice test-cause, les poids et coûts). Le plan optimal est identique au plan heuristique (on a utilisé le programme décrit en 2ème Partie).

1	2	3	4	5	6	7	8	9	10	10	
convergence	temps d'attente	différentiel	gauge	nettoyage	the coil	regulation	the roof	contrainte	fluorimétrie	the air	the air
0	0	5	2	3	1	5	5	5	5	10	
libelles des caisses											

1	2	3	4	5	6	7	8	9	10	10
1	1	1	1	1	1	1	1	1	1	2
1	2	1	1	1	1	1	1	1	1	1
1	1	2	1	1	1	1	1	1	1	1
1	1	1	2	1	1	2	2	2	2	1
1	1	1	1	2	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	1
1	1	1	1	1	1	2	1	1	1	1

identification incident
 RD trop grand (LARS)

1	Mez Rd Sc 2
2	" temps 1
3	" diff-gaz 1
4	vent nettoys 5
5	feinweis 10
6	snip tube 20
7	ventilation 5
8	meurme 7
9	ventilation 1
10	vent. etape 1

no libelles des tests

pois du passé 100 (5)

remplir dans l'ordre 1 2 3 4 5
 contraintes sur l'ordre des tests

DIAGNOSTIC - description d'incident -

(2)

RTC - PAS

NOMBRE DE CAUSES = 13
NOMBRE DE TESTS = 14

MATRICE DES RELATIONS TESTS-CAUSES

2	1	2	2	2	2	2	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	1	1	2	
2	2	2	2	2	2	2	2	2	2	2	1	2	
2	2	1	2	2	2	2	2	2	2	2	2	2	
2	2	2	1	2	2	2	2	2	2	2	2	2	
2	2	2	2	1	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	1	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	
1	1	2	2	2	2	2	2	1	1	2	1		
2	2	1	1	1	1	1	1	2	5	3	2		

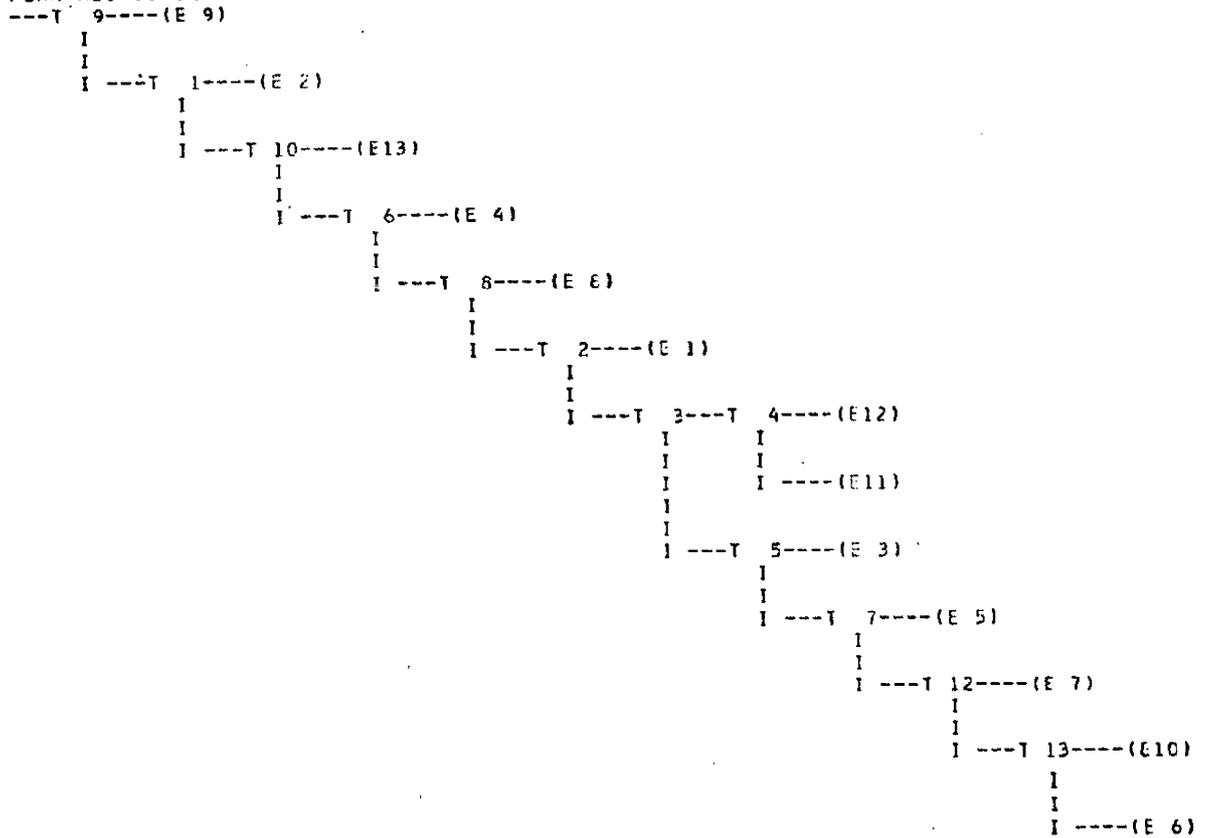
COUTS

1	5
2	5
3	5
4	5
5	5
6	5
7	5
8	5
9	5
10	5
11	10
12	120
13	360
14	2400

POIDS

50	150	10	95	10	10	190	95	190	50
25	25	100							

PLAN HEURISTIQUE



COUT DU PLAN HEURISTIQUE = 75425
TEMPS DE CALCUL DE LA SOLUTION HEURISTIQUE EN 1/100 S = 8

PLAN OPTIMAL

IDENTIQUE

COUT DU PLAN OPTIMAL = 75425
NOMBRE D'ALLOCAIONS MEMOIRE FAITES = 1171
NOMBRE D'APPELS DE LA FONCTION RECURSIVE PLANOPT = 1172
TEMPS DE CALCUL DE LA SOLUTION OPTIMALE EN 1/100 S = 2746

CONCLUSION

L'approche questionnaire réalisable, appliquée au diagnostic d'incidents, possède deux avantages principaux :

- le premier est lié bien entendu à l'amélioration en coût du diagnostic, obtenue par rapport à une résolution par l'expérience seule, ou même à la main avec le catalogue établi.
- le second avantage, d'ordre qualitatif, n'est cependant pas négligeable, et a été constaté "sur le tas" : l'effort de systématisation requis pour dresser le catalogue d'inspection permettra de rassembler les connaissances des divers spécialistes en un document disponible pour tous, transmissible et autorisant un suivi. Le plan d'inspection peut souligner les points faibles du diagnostic, et, sans toutefois la remplacer, soutenir utilement l'expérience du responsable.

Enfin, on pourra, dans un stade ultérieur, exploiter le catalogue en vue d'une meilleure connaissance de la structure de la chaîne, et des liaisons logiques entre variables.

CHAPITRE VIII.

-:-

TRADUCTION DES TABLES DE DÉCISIONS À ENTRÉES LIMITÉES EN ORGANIGRAMMES

-

INTRODUCTION

La traduction des tables de décisions à entrées limitées (abréviation : TDEL) en organigramme, peut se poser en termes de réalisation de questionnaires.

Dans ce chapitre, nous montrons comment utiliser les résultats établis en seconde partie -algorithmes, critères- pour la traduction des TDEL.

La présentation des TDEL est faite à partir des tables d'identification (abréviation : TI). Les problèmes spécifiques et améliorations possibles, par rapport à la construction de questionnaires, sont établis à la lumière des travaux déjà effectués dans ce domaine, revus en fin de chapitre.

VIII.1. LES TABLES DE DÉCISIONS À ENTRÉES LIMITÉES

TDEL complètement développée

On appelle ainsi une TI comportant :

- n questions nommées ici conditions, affectées ou non de coûts, à deux issues possibles : oui ou non
- 2^n éventualités nommées règles, affectées de poids ou fréquences
- la correspondance habituelle conditions-règles sous forme tabulaire. Aux 2^n règles développées correspondent les 2^n combinaisons de conditions
- on se donne en plus un ensemble d'actions $A = \{A_1, \dots, A_p\}$, partition de l'ensemble des 2^n règles : $\forall E_i \in E, E_i$ appartient à un A_j et un seul.

L'analogie avec les fonctions booléennes de n variables à valeurs dans A est totale.

Traduction en organigramme

Le problème de la traduction en organigramme se formule ainsi : établir un processus déterministe qui, pour chaque combinaison possible de entrées conditions, permet d'identifier, non pas la règle correspondante (problème trivial) mais l'action qui contient cette règle.

Comme dans le problème d'identification général décrit au chapitre I, deux voies sont ouvertes :

- une voie "combinatoire" ou interprétation
- une voie "séquentielle" par l'approche questionnaire, que nous allons étudier ; le résultat est un organigramme sous forme d'arbre de conditions, les sommets terminaux figurant les actions

La seule différence avec la construction de questionnaire est la suivante : la séparation des ensembles d'éventualités peut s'arrêter dès que le sous-ensemble F de règles à séparer appartient à l'un des A_i . Les solutions possibles sont des questionnaires inachevés. On cherchera à minimiser le coût moyen de l'arbre, ce qui correspond à minimiser un temps moyen d'exécution de programme.

Exemple

	R_1 (0,15)	R_2 (0,2)	R_3 (0,05)	R_4 (0,1)	R_5 (0,2)	R_6 (0,2)	R_7 (0,05)	R_8 (0,05)
C_1 (10)	0	0	0	0	N	N	N	N
C_2 (8)	0	0	N	N	0	0	N	N
C_3 (5)	0	N	0	N	0	N	0	N
	A_1			A_2		A_3	A_4	

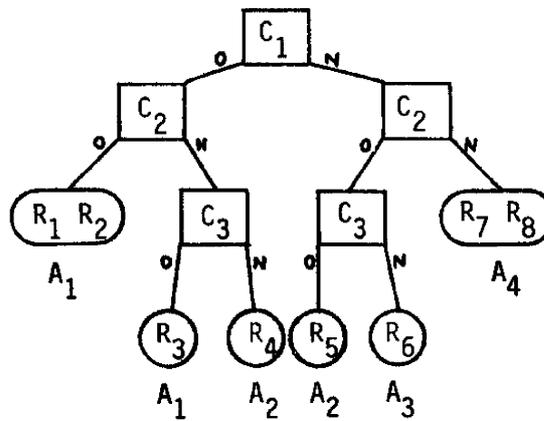


Figure VIII.1.

La figure VIII.1. représente une table complètement développée et une traduction possible, de coût :

$$10 + 8 + 5 \cdot (0,05 + 0,1 + 0,2 + 0,2) = 20,75.$$

Comme pour les problèmes d'identification, (III.2.), on peut représenter une TDEL, et sa traduction, sur un hypercube : exemple figure VIII.2. On remarque que tous les points de l'hypercube sont "attribués", ce qui n'est pas le cas en général pour les problèmes d'identification, exemple figure III.4.

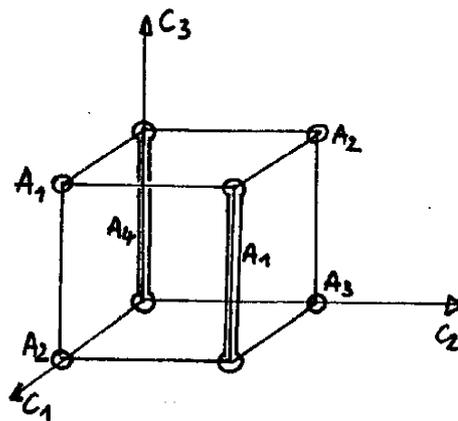


Figure VIII.2.

Simplifications : les TDEL

En fait, on représente rarement une table de décision sous forme complètement développée. Des simplifications d'écriture sont obtenues grâce aux conventions suivantes :

- 1) L'utilisation d'une colonne "AUTRES" permet de regrouper des règles non explicitées menant à une seule action.
- 2) Il n'est pas nécessaire de distinguer logiquement un ensemble de règles appartenant à une même action. Si dans cet ensemble, figure des règles "adjacentes", c'est-à-dire ne différant que par l'issue d'une seule condition, ces règles peuvent être regroupées en utilisant un '-' (dash) pour cette condition. Avec k '-' on regroupe 2^k règles.

La table précédente peut être ainsi simplifiée :

	R_{12}	R_3	R_6	R_{78}	AUTRES (R_{45})
C_1	0	0	N	N	
C_2	0	N	0	N	
C_3	-	0	N	-	
	A_1		A_3	A_4	A_2

La représentation condensée n'est généralement pas unique. On aurait pu ici regrouper R_2 et R_3 plutôt que R_1 et R_2 .

- 3) Lorsque des conditions sont liées, certaines règles peuvent devenir impossibles sémantiquement, comme le montre l'exemple de la figure VIII.3. : la règle R_1 correspond à une impossibilité. Comme elle ne se produit jamais, on pourrait la regrouper avec R_2 pour faire "0 -" ou/et avec R_3 : "- 0". Certains auteurs parlent alors d'"entrées impliquées" et notent R_2 : "0 N^* ", R_3 : " N^* 0" ou "0 (N)" et "(N) 0". Cela revient à supprimer la règle impossible dans la TDEL. Nous verrons les difficultés introduites par ces impossibilités dans l'optimisation de la traduction.

	R_1	R_2	R_3	R_4
$C_1 : x < 0$	0	0	N	N
$C_2 : x > 1$	0	N	0	N

Figure VIII.3.Consistance

Une TDEL est généralement écrite sous forme condensée, en tous cas dès que le nombre des conditions dépasse 5 ou 6. Il faut donc vérifier si les simplifications implicitement faites en utilisant les '-' sont correctes, c'est-à-dire si les ensembles A_i forment bien un recouvrement disjoint de l'ensemble des règles complètement développées.

C'est un problème de consistance, au départ syntaxique, mais pouvant comporter des aspects sémantiques : le recouvrement de deux ensembles A_i et A_j peut être dû à l'existence d'une règle impossible, comme nous l'avons vu si des conditions sont liées.

Dans la pratique des tables de décision, ce problème de consistance est important et difficile. Il va de pair généralement avec la simplification des tables (maximisation du nombre de "-") analogue au problème de simplification de fonctions booléennes. Ceci sort du cadre de notre travail et nous supposons par la suite que nous avons à traiter des TDEL consistantes, développables sans ambiguïtés, avec des règles impossibles connues implicitement ou explicitement, et des conditions exprimées sous forme abstraite.

Distinction conceptuelle entre TI et TDEL

Bien que formellement proches, les TI et les TDEL sont des modèles de deux réalités différentes, qu'il importe de distinguer :

Pour une TI, on se donne d'abord l'ensemble E des éventualités, correspondant aux états possibles du "système" identifié, non nécessairement en nombre 2^n . Puis, on détermine les issues des questions formulables pour chaque éventualité, issues bien définies puisque le système est supposé déterministe.

Au contraire, dans une TDEL, ce sont les conditions et leurs combinaisons qui déterminent l'état du système. On ajoute en plus une correspondance états \rightarrow actions.

VIII.2. AMELIORATIONS POSSIBLES DUES A LA STRUCTURE DU PROBLEME

Nous présentons dans ce paragraphe, deux notions introduites par S. GANAPATHY et V. RAJARAMAN (1973), qui permettent d'optimiser la traduction des tables en organigrammes. Ces notions s'appliquent parfaitement dans le cadre strict de la définition donnée ci-dessus des TDEL. Mais dans le cas, fréquent en pratique, où des règles ne sont attribuées à aucun A_i (cas des règles impossibles), nous montrons par un contreexemple que l'application de ces notions peut amener des erreurs. On complètera les définitions pour que les résultats soient utilisables.

Condition totalement pertinente

"Si les règles d'une action ne peuvent en aucune façon se combiner pour former un "-" pour une condition particulière, alors la condition est dite pertinente pour cette action. Si une condition est pertinente pour toutes les actions, elle est dite totalement pertinente".

Propriété : les auteurs cités affirment qu'on peut tester en premier les conditions totalement pertinentes sans nuire à l'optimalité de l'arbre, puisque "une condition totalement pertinente doit être testée sur tous les chemins de l'arbre".

Démonstration complète :

- Soit C_i , pertinente pour A_k , et R une règle de A_k . Notons R^i la règle adjacente à R obtenue en changeant dans R l'issue de C_i . Si R^i appartient à une action A_l , alors $A_l \neq A_k$, sinon C_i ne serait pas pertinente ; R et R^i devront être séparées, la seule façon de le faire étant d'employer C_i .

C_i se trouve donc sur le chemin menant à R.

Si C_i est totalement pertinente, alors le chemin aboutissant à toute règle R contient C_i , toujours à condition que R^i soit une règle appartenant à une action, c'est-à-dire non impossible \square

Le contreexemple figure VIII.4.a. montre l'erreur qu'on peut faire si on ne tient pas compte du fait que R et R^i doivent être possibles. La définition d'une condition totalement pertinente doit être ainsi précisée pour pouvoir utiliser la même propriété :

Définition : "Si les règles d'une action ne peuvent en aucune façon se combiner pour former un "-", y compris avec des règles impossibles, pour une condition particulière, alors la condition est dite pertinente pour cette action. Une condition pertinente pour toutes les actions est totalement pertinente".

Conditions redondantes

La redondance est employée ici dans un sens différent qu'en V.2. "Si une méthode existe pour combiner des règles de façon à obtenir uniquement des "-" pour une condition, cette condition est dite redondante".

Propriété : un arbre optimal n'emploie pas de condition redondante ; une telle condition peut donc être ôtée de la table.

Démonstration :

- Soit C_i une condition redondante. R désigne une règle non impossible. Deux cas sont à envisager :
 - 1) Si, $\forall R, R^i$ est non impossible également, alors R et R^i appartiennent à un même A_k , et utiliser C_i revient à partager le problème en deux problèmes identiques au premier au point de vue des actions. Aucun gain n'est retiré de l'opération. On peut ôter C_i de la table.
 - 2) Si $\exists R$ non impossible, telle que R^i est impossible, alors en utilisant C_i on ne partage pas le problème en deux problèmes identiques au premier. On ne peut donc ôter C_i de la table \square

Le contreexemple figure VIII.4.b. montre l'erreur qu'on peut faire si on utilise les règles impossibles pour déterminer les conditions redondantes. La définition d'une condition redondante doit être ainsi modifiée :

Définition : "Si on peut combiner entre elles des règles attribuées à une même action, sans utiliser les règles impossibles, de façon à obtenir uniquement des "-" pour une condition, cette condition est redondante".

	R ₁	R ₂	R ₃	R ₄	AUTRES
C ₁	0	0	N	N	
C ₂	0	N	0	N	
C ₃	0	N	N	N	
	A ₁	A ₂	A ₃	A ₄	impossibles

a

Sans combinaison avec les règles impossibles, les 3 conditions sont 'totalement pertinentes'. Or aucune ne doit être obligatoirement sur tous les chemins de la traduction.

	R ₁	R ₂	R ₃	R ₄
C ₁	0	0	N	N
C ₂	0	N	0	N
C ₃	-	-	-	-
	A ₁	A ₂	A ₃	A ₄

b

Maximiser le nombre de '-' pour C₃ en combinant avec les règles impossibles rend cette condition faussement 'redondante' alors qu'elle peut à l'évidence servir dans un arbre optimal.

Figure VIII.4.

VIII.3. UTILISATION DES RESULTATS ACQUIS EN SECONDE PARTIE POUR L'OPTIMISATION DE LA TRADUCTION

Hypothèses sur la TDEL à traduire

— P.J.H. KING et JOHNSON R.G. (1974) ont montré qu'il est nécessaire, pour optimiser une traduction, de connaître les fréquences attachées aux règles complètement développées. On s'expose sinon, comme le montre l'exemple figure VIII.5., à produire des arbres auxquels on ne pourrait attribuer un coût. Une solution consiste à admettre que les fréquences se partagent également entre règles regroupées.

— La table doit être complètement développée.

— Deux interprétations sont possibles pour les règles affectées de fréquences nulles :

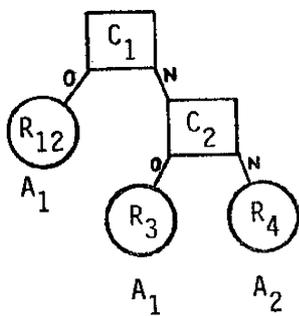
- 1) On les considère comme des règles habituelles au point de vue logique, mais de fréquence négligeable. Elles sont attribuées dans la table à une action, et donc aussi dans la traduction.

	R_{12} (0,4)	R_3 (0,2)	R_4 (0,4)
$C_1(10)$	0	N	N
$C_2(10)$	-	0	N
	A_1	A_2	

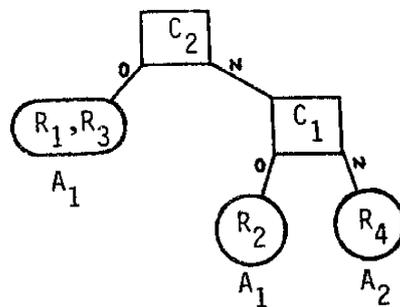
en développant :

	R_1 (α)	R_2 (β)	R_3 (0,2)	R_4 (0,4)
	0,4			
	0	0	N	N
	0	N	0	N
	A_1		A_2	

Deux traductions sont possibles :



coût : $10 + 10 \cdot (0,6) = 26$



Coût : $10 + 10 \cdot (\beta + 0,4) = 24 + 10 \cdot \beta$

β peut varier de 0 à 0,4 suivant la répartition des fréquences entre R_1 et R_2 . Si $\beta < 0,2$ la traduction de droite est meilleure : pour optimiser, on doit connaître la fréquence des règles développées.

Figure VIII.5.

- 2) Ce sont des règles logiquement impossibles auquel cas on ôte ces règles de la table. Toute traduction aura pour effet d'attribuer implicitement une règle impossible à une action, inconnue au départ.

Traduction optimale

Avec ces hypothèses, la TDEL se présente comme une TI, avec en plus l'application de l'ensemble des règles dans celui des actions. Il est clair que l'algorithme-définition I.4. peut être utilisé pour produire une traduction, pourvu qu'il soit modifié de façon qu'un ensemble F de règles constitue un sommet terminal (F,0) du questionnaire dès qu'il existe un A_i contenant F.

La même modification portée dans l'algorithme de construction optimale construira une traduction optimale de la TDEL.

Le test de production des sommets terminaux doit subir la modification suivante (figure V.8. et V.9.).



On montre sans difficulté que les techniques d'échange des questions et d'élimination des questions redondantes de 1ère et 2ème espèce (le sens est différent de celui du § VIII.2.) exposées en V.2., s'appliquent également ici.

Traduction heuristique

La traduction optimale n'est praticable sur ordinateur que lorsque le nombre de conditions ne dépasse pas 6 ou 7, en tous cas si le nombre de règles développées, non impossibles, reste de l'ordre de quelques dizaines.

Sinon, ou pour des calculs manuels, on devra recourir aux méthodes sous-optimales, comme dans le cas des questionnaires.

Avec les mêmes hypothèses et modifications que précédemment, l'algorithme heuristique développé au chapitre VI construit des traductions sous-optimales.

Cependant, les résultats risquent de ne pas être bons si le choix heuristique des conditions pour chaque séparation ne tient pas compte de l'application de E dans A, déterminante pour l'optimisation. A cette fin, nous proposons :

1) Soit d'utiliser un critère heuristique du type :

$$(I + S')/q \text{ ou } (J + S')/q \quad (\text{notations du chapitre VI})$$

avec S' = somme pondérée des fréquences des règles regroupées en actions distinctes (séparées) dans la partition.

La fonction S' a des propriétés analogues à celles de S étudiée au chapitre VI.

2) Soit d'utiliser l'algorithme de S. GANAPATHY et V. RAJARAMAN (1973) basé sur la théorie de l'information, et tenant compte des regroupements possibles en actions :

- avant de calculer la valeur du critère pour une condition, maximiser le nombre de "-" pour cette condition.
- poser p_0 = somme des fréquences des OUI } pour la condition
 p_N = somme des fréquences des NON } examinée
- calculer $I\left(\frac{p_0}{p_0+p_N}, \frac{p_N}{p_0+p_N}\right) = \left(\frac{p_0}{p_0+p_N}\right) \log\left(\frac{p_0+p_N}{p_0}\right) + \left(\frac{p_N}{p_0+p_N}\right) \log\left(\frac{p_0+p_N}{p_N}\right)$
 et $I' = (p_0 + p_N) I$
- choisir la condition qui maximise le critère I'/q .

3) Notre critère J développé au chapitre VI peut prendre place dans le schéma précédent :

- après avoir maximé le nombre de "-", calculer :

$$J\left(\frac{p_0}{p_0+p_N}, \frac{p_N}{p_0+p_N}\right) = \min\left(\frac{p_0}{p_0+p_N}, \frac{p_N}{p_0+p_N}\right)$$

$$\text{puis } J' = (p_0 + p_N) J = \min(p_0, p_N)$$

- choisir la condition qui maximise le critère J'/q . On obtient un critère plus simple à calculer, et qui, d'après les validations faites sur les questionnaires, doit donner des résultats au moins aussi bons que pour le critère issu de la théorie de l'information.

Cette dernière méthode est illustrée par la figure VIII.7. qui représente la traduction manuelle d'une table de décision tirée de l'article de M. VERHELST (1972) et reprise par K. SHWAYDER (1974). L'arborescence complète est dessinée figure VIII.6.

Comparons avec la traduction faite par ce dernier auteur, utilisant le critère de la théorie de l'information :

- 1) pour cette table, les règles correspondent directement à des actions. Il n'y a pas de regroupements supplémentaires possibles.

- 2) On doit supposer, au cours de la traduction, que les fréquences des règles regroupées se partagent moitié-moitié lorsqu'on les développe (cas des règles R_2 et R_3). Par un curieux hasard, le coût de l'arbre de K. SHWAYDER peut se calculer sans faire cette hypothèse.
- 3) On trouve un arbre de coût très légèrement inférieur (1658 au lieu de 1663).
- 4) Les calculs sont beaucoup plus simples : on n'a pas besoin de recourir à une table de logarithme, et les seules opérations utilisées sont : addition, division, minimum, maximum.

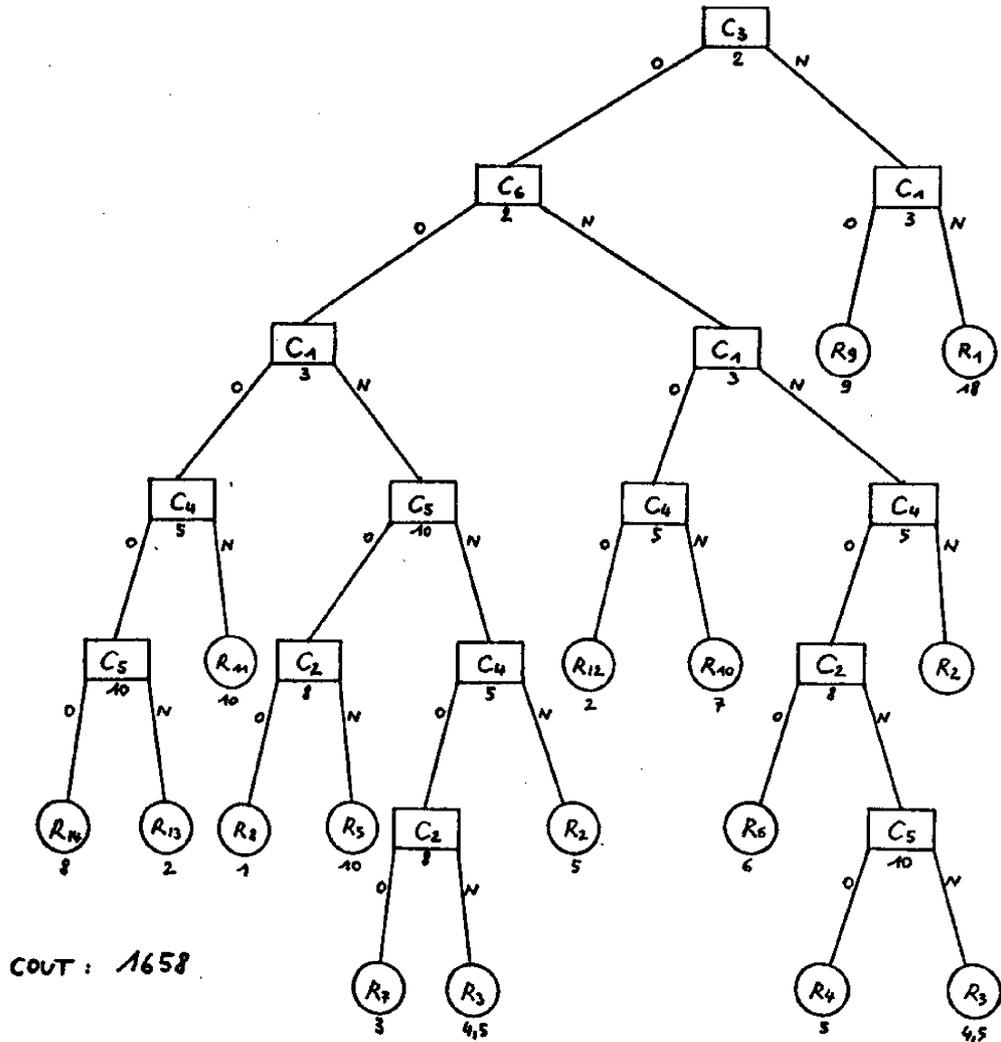


Figure VIII.6.

TABLE INITIALE
(VERHELST 1972
et SHWAYBER 1974)

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}
C_1 3	N	N	N	N	N	N	N	N	0	0	0	0	0	0
C_2 8	-	-	N	N	N	0	0	0	0	0	0	0	0	0
C_3 2	N	0	0	0	0	0	0	0	N	0	0	0	0	0
C_4 5	N	N	0	0	0	0	0	0	N	N	N	0	0	0
C_5 10	N	N	N	0	0	-	N	0	N	N	N	-	N	0
C_6 2	-	-	-	N	0	N	0	0	-	N	0	N	0	0

P_0	P_N	min	CR
38	62	38	12,7
48	24	24	3,0
73	27	27	13,5 *
46	54	46	9,2
24	68	24	2,4
34	20	20	10,0

$C_3 = 0$

	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}
C_1 3	N	N	N	N	N	N	0	0	0	0	0	0
C_2 8	-	N	N	N	0	0	0	0	0	0	0	0
C_4 5	N	0	0	0	0	0	0	N	N	0	0	0
C_5 10	N	N	0	0	-	N	0	N	N	-	N	0
C_6 2	-	-	N	0	0	0	0	N	0	N	0	0

P_0	P_N	min	CR
29	44	29	9,3
39	24	24	3,0
46	27	27	5,4
24	44	24	2,4
34	20	20	10,0 *

$C_6 = 0$

	R_2	R_3	R_5	R_7	R_{11}	R_{13}	R_{14}
C_1 3	N	N	N	N	0	0	0
C_2 8	-	N	N	0	0	0	0
C_4 5	N	0	0	0	0	N	0
C_5 10	N	N	0	0	N	N	0

P_0	P_N	min	CR
20	23,5	20	6,7 *
24	14,5	14,5	1,8
28,5	15	15	3,0
19	24,5	19	1,9

$C_6 = N$

	R_2	R_3	R_4	R_6	R_{10}	R_{12}
C_1 3	N	N	N	N	0	0
C_2 8	-	N	N	0	0	0
C_4 5	N	0	0	0	N	0
C_5 10	N	N	0	-	N	-

P_0	P_N	min	CR
9	20,5	9	3,0 *
3,5	15	3,5	1,2
17,5	12	12	2,4
16,5	5	5	0,5

$C_1 = N$

	R_2	R_3	R_5	R_7	R_8
C_2 8	-	N	N	0	0
C_4 5	N	0	0	0	0
C_5 10	N	N	0	N	0

P_0	P_N	min	CR
4	14,5	4	0,5
18,5	5	5	1,0
11	12,5	11	1,1 *

$C_1 = 0$

	R_{11}	R_{13}	R_{14}
C_4 5	N	0	0
C_5 10	N	N	0

P_0	P_N	min	CR
10	10	10	2,0 *
8	12	8	0,8

$C_5 = N$

	R_2	R_3	R_7
C_2 8	-	N	0
C_4 5	N	0	0

P_0	P_N	min	CR
4,5	3	3	0,4
5	7,5	5	1 *

$C_1 = N$

	R_2	R_3	R_4	R_6
C_2 8	-	N	N	0
C_4 5	N	0	0	0
C_5 10	N	N	0	-

P_0	P_N	min	CR
6	9,5	6	0,75 *
15,5	5	5	1 *
5	9,5	5	0,5

Figure VIII.7. Traduction manuelle d'une table de décision, en utilisant le critère "J/q".

$C_4 = 0$

	R_3	R_4	R_6
C_2 8	N	N	0
C_5 10	N	0	-

P_0	P_N	min	CR
6	9,5	6	0,75 *
5	4,5	4,5	0,45

VIII.4. BREVE REVUE DE LITTERATURE SUR L'OPTIMISATION DES TDEL

On trouvera une présentation générale des tables de décision dans les ouvrages de BAGLIN G. et KLEE J. (1970) pour l'initiation, et POLLACK S.L., HICKS H.T. and HARRISON W.J. (1971).

Des travaux de base pour la traduction par interprétation ("rule mask techniques") sont présentés dans les articles de KING P.J.H. (1966) et MUTHUKISHNAN C.R. and RAJARAMAN V. (1970).

Pour la traduction en organigramme, tous les travaux recensés visent à minimiser le coût moyen de l'arbre, ou la place mémoire utilisée dans la traduction. Pour le premier cas, les travaux les plus significatifs sont les suivants :

.POLLACK S.L. (1965) donne un algorithme de construction heuristique, basé sur le compte des "-" pour une condition. Les coûts sont supposés égaux.

.REINWALD L.T. and SOLAND R.M. (1966) proposent un algorithme optimal, avec coûts distincts, basé sur une méthode "Branch and Bound" faisant intervenir un coût minimal de séparation pour les règles non encore séparées.

.SHWAYDER K. (1971) propose une modification de l'algorithme de POLLACK, utilisant le critère de la théorie de l'information.

.L'algorithme heuristique de VERHELST M. (1972) introduit des coûts distincts dans celui de POLLACK ; mais son algorithme optimal est incorrect, comme le montre l'article de KING, P.J.H. and JOHNSON R.G. (1974).

.Enfin, S. GANAPATHY et V. RAJARAMAN (1973) et de même SHWAYDER K. (1974) reprennent le critère heuristique de la théorie de l'information, mais sous la forme I/q et avec les améliorations discutées en VIII.2.

Il faut signaler également l'article très complet de MYERS H.J. (1972) sur les problèmes d'implémentation des compilateurs de tables.

CONCLUSION

Le nouveau guide heuristique construit au chapitre VI nous procure un outil simple et efficace pour la traduction des tables de décision.

En allant dans le sens des préoccupations actuelles en matière de programmation structurée (exemple E.W. DIJKSTRA, 1972), on peut dire qu'il n'est pas souhaitable, ou possible sans erreur, de dresser manuellement des tables de décision ayant une taille supérieure à 6 conditions. Nous restons alors dans le domaine où la recherche d'une solution optimale est encore pratiquement possible sur ordinateur, soulignant ainsi l'intérêt de l'algorithme développé au chapitre V, pour la traduction optimale des tables de décision.

Les plus grandes tables se rencontreront, soit lorsqu'on voudra combiner une cascade de tables plus réduites, soit lorsqu'elles pourront être engendrées directement par un niveau supérieur.

CONCLUSION

Pour résoudre les problèmes d'identifications déterministes, nous avons défini deux approches possibles : les procédures de type combinatoire, et les procédures de type séquentiel ou questionnaires réalisables, et montré que ce dernier modèle est le plus intéressant du point de vue du coût de l'identification.

Un questionnaire réalisable a été défini formellement comme un objet mathématique produit par un algorithme, ce qui offre l'avantage d'une définition à la fois précise et directement opérationnelle pour les constructions optimales et heuristiques proposées. A partir de cette définition, on a donné de nouvelles démonstrations des propriétés classiques des questionnaires, ainsi que des expressions du coût pondéré et du coût maximum qui sont des généralisations des notions de longueur et de hauteur des questionnaires sans coûts. On a donné une nouvelle expression du coût pondéré, sous forme de produit matriciel, mettant en évidence ses propriétés de linéarité. Une étude a été consacrée aux recherches de bornes inférieures du coût pondéré : à côté des méthodes du "coût de preuve minimal" et du "questionnaire libre" déjà existantes, nous avons proposé une méthode par les "cases essentielles" et une autre utilisant la théorie de l'information.

Avant d'aborder la construction des questionnaires, nous avons dégagé des outils de représentation de l'univers des solutions, tels les graphes ET/OU. La réalisation (optimale) a été mise sous forme de la recherche d'un chemin (minimal) dans un graphe, de recouvrement d'hypercube, programme linéaire en variable bivalente, de séparation des éventualités deux à deux (affectation) ; ces "réductions" visaient à vérifier que le problème n'était pas plus facile à résoudre sous une autre forme. En fait, des travaux récents en informatique théorique ont montré que le problème de former un questionnaire réalisable optimal est "NP complet", c'est-à-dire équivalent en complexité au problème du voyageur de commerce, ou de la démonstration automatique de théorème ("satisfiabilité"). Ceci permet de

situer le problème dans l'ensemble de la Recherche Opérationnelle, et montrent l'importance des études sur les algorithmes approchés.

L'intérêt des algorithmes optimaux n'est pas seulement théorique : ils permettent aussi de comparer les solutions optimales aux solutions heuristiques dont on veut mesurer la valeur. Deux méthodes ont été jusqu'à présent proposées ; l'une, par F. DUBAIL (1967) et reprise par M.R. GAREY (1972 b), d'aval en amont, basée sur la programmation dynamique; l'autre est une procédure de séparations et d'évaluations progressives, écrite par la S.E.M.A. (1970), construisant les questionnaires en partant de la racine. Nous avons programmé, sous forme récursive, un algorithme de recherche "en profondeur d'abord", implémenté en PL/1, utilisant des techniques adaptées pour le stockage des solutions partielles et pour faciliter l'énumération implicite. Une amélioration intéressante, basée sur l'échange des questions successives, a été proposée. Un générateur pseudo-aléatoire de problèmes d'identification permet d'étudier statistiquement le comportement de l'algorithme, en temps de calcul et place mémoire utilisés, en fonction de la taille du problème et d'autres paramètres. Cette étude pourra servir de base de comparaison avec d'autres algorithmes.

L'apport le plus original concerne l'étude de critères heuristiques pour guider la construction de bonnes solutions sous-optimales. Le critère proposé habituellement est issu de la théorie de l'information : on choisit à chaque étape du développement de l'arbre la question maximisant le rapport information de SHANNON apportée sur coût de la question. Nous avons donné une interprétation de ce critère, basée sur le fait que coût pondéré et information possèdent des propriétés analogues d'additivité et de pondération.

Un nouveau critère a été construit, justifié théoriquement et pratiquement : pour une question de base 2, soient p_1 et p_2 les poids des deux blocs séparés par la question, et q son coût. On choisit, pour développer le questionnaire, la question maximisant le rapport $J(p_1, p_2)/q$, avec $J(p_1, p_2) = \min(p_1, p_2)$. Ce critère est généralisé pour les bases supérieures en : $J(p_1, p_2, \dots, p_b) = \frac{b \log b}{b - 1} (1 - \max_{1 \leq i \leq b} p_i)$

Les résultats expérimentaux obtenus avec ce critère très simple montrent qu'il est au moins aussi bon que le critère de la théorie de l'information. D'une façon générale, les coûts heuristiques sont toujours très proches des coûts optimaux.

On a construit d'autres algorithmes heuristiques semi-énumératifs en utilisant ces critères dans l'algorithme optimal présenté : en limitant le nombre de questions "essayées" pour chaque sous problème à 2 ou 3, en les sélectionnant par un critère heuristique, on limite le "back-tracking" et on obtient des solutions encore meilleures que par l'heuristique directe ; au prix, certes, d'un temps de calcul plus élevé. Ce compromis entre deux méthodes extrêmes donne des résultats très intéressants.

L'intérêt pratique des questionnaires réalisables a été souligné en dernière partie, par la présentation de deux applications à caractère d'actualité dans le domaine des Sciences pour l'Ingénieur : le diagnostic des systèmes déterministes et la traduction des tables de décision.

Le diagnostic d'incidents a été étudié du point de vue de l'établissement de "plans d'inspection". Une application industrielle a été réalisée. Il ressort deux avantages principaux de l'approche questionnaire : le premier est lié à l'optimisation possible du coût de diagnostic ; le second, non négligeable, vient de l'effort de formalisation et de systématisation requis.

Poser sous forme de problème d'identification la traduction des tables de décision, permet également une meilleure formalisation des tables et facilite la compréhension des problèmes posés par la traduction : simplifications, consistances. Les algorithmes proposés en 2^{de} partie peuvent être utilisés avec peu de modification.

Des arguments pratiques aussi bien que théoriques montrent que les méthodes heuristiques sont appelées à être de plus en plus utilisées ; nous pensons qu'il faut appuyer les recherches dans ce sens, essayer d'unifier les principes communs à ces méthodes, forger et étudier des outils nouveaux. L'informatique est peut-être un art, mais c'est avant tout une science expérimentale.

INDEX

- Actions, 125
Algorithmes de HUFFMANN, 39,68,69
Amont-aval, 65,67,69,93
Aval-amont, 65,67-69,96
- Base, 13
Base réelle, 24,25
Branch and bound, 59,70,138
- Cases essentielles, 37
Catalogue d'incidents, 115,117,123,124
Causes (voir éventualités), 13,115
Conditions, 125
 pertinentes, 130,131
 redondantes, 131
Consistance, 129
Coût d'une identification, 29,30
 d'une question, 13
 d'un questionnaire, 29,30
 maximum, 30,34,47,48,68,91
 moyen, 34
 optimal, 47,48,68
 pondéré, 29,30,95
Critère heuristique, 93...134-136
- Echange de questions, 74,75,82
Énumération implicite, 70,71,81
Éventualités, 13
 compatibles 73,90
 séparables 13,54
- FORTRAN, 119
Fréquences (voir poids), 125,132
 nulles, 132
- Générateur de problèmes d'identifications, 84
Graphe ET/OU, 46,69
- Hauteur d'un questionnaire, 30
- Information de SHANNON, 40,94,95,98,100,135,138
Issue d'une question, 13
- Longueur d'un questionnaire, 40
- Objets (voir éventualités), 13
- Partition, 13,72,94,95
Plan d'inspection, 18,93,105,115...
PL/1, 78,79
Poids d'une éventualité, 13
 d'une question, 31
Probabilité (voir poids), 13,119
Problème d'identification, 13
Procédure d'identification combinatoire, 16,33,34,52,126
 séquentielle, 17,33,34,126
Profondeur d'abord (exploration en), 70,90
Programmation dynamique, 68,69
PSEP, 55,59,70,71,91
PSES, 59,70,71,78
- Question, 13
 élémentaire, 14,73,76,82,90
 (non) formulable, 15,18
 globale, 14
Questionnaire de HUFFMANN, 39,40
 inachevé, 45,49,70,126
 libre, 17,18,38,39,68,94
 réalisable, 18,19

- Racine d'un questionnaire, 17,20
 Redondance d'une condition, 131
 d'une question, 43,72,82
 Règles (voir éventualités), 125
 adjacentes, 128
 développées, 125
 impossibles, 128-131, 133
 regroupées, 128

 Sommet pendant, 70,71
 Sous-ensemble d'éventualités (non) formulable, 69,74
 Sous-problème engendré, 74,81,90

 Table d'identification (TI), 13,88,125,129
 Table de décision à entrées limitées (TDEL), 125...
 Table marquée, 32,33
 Temps de calcul, 69,71,75,76,81,82,84-86,90,91,103,104,108,109
 Test (voir question), 13,115
 Traduction, 126

PRINCIPALES NOTATIONS

E	e	E_i	e_i	} . 13
Q	q	Q_i	q_i	
		b_i	\mathcal{E}_i^s	
\mathcal{Q}	F	F_i^s	19,20	
C	K	30		
	\mathcal{C}_i	20		
	\mathcal{E}_j	30,31		
B	b_{ij}	31		
A_i	125			
QE Q3	84			
'..'	128			

TABLE DES MATIÈRES

	Pages
PREAMBULE	1
BIBLIOGRAPHIE	5
PARTIE I : PROBLÈMES D'IDENTIFICATION ET QUESTIONNAIRES REALISABLES	 11
<u>CHAP. I. : HYPOTHESES ET BUT - HISTORIQUE - FORMALISATION</u>	 13
I.1. Hypothèses et définitions	13
I.2. Sémantique et but	14
I.3. Différents types de procédures d'identifications	15
- type "combinatoire"	16
- type "séquentiel" : questionnaires réalisables	17
I.4. Définition et construction des questionnaires	19
- définition et commentaires	19
- propriétés des questionnaires	21
I.5. Restrictions et extensions du modèle	25
- identifications élémentaires et lexicographiques	25
- pseudo-questionnaires	26
<u>CHAP. II. : COÛTS D'UN QUESTIONNAIRE - BORNES INFÉRIEURES</u>	 29
II.1. Coûts	29
- coût maximum	30
- coût pondéré	30
- la table marquée	32
- comparaison des procédures combinatoires et des questionnaires	 33
- diverses propriétés des coûts	35

11.2. Bornes inférieures du coût pondéré	36
- cases essentielles	37
- coût de preuve minimal	37
- questionnaire libre	38
- par la théorie de l'information	40
<u>CHAP. III. : CONSTRUCTIONS DE QUESTIONNAIRES : OUTILS DE</u>	
<u>REPRESENTATION ET PROBLEMES EQUIVALENTS</u>	45
III.1. Représentations directes	45
- arbre des solutions	45
- graphe ET/OU des solutions	46
III.2. Problèmes équivalents	49
- chemin dans un graphe de partitions	49
- recouvrement disjoint d'hypercube	51
- codage instantané	52
- programme linéaire en variable bivalente	53
- séparation des éventualités deux à deux	54
<u>CHAP. IV. : LE PROBLEME DU QUESTIONNAIRE OPTIMAL EST</u>	
<u>"POLYNOMIAL COMPLET"</u>	59
PARTIE II : CONSTRUCTIONS DIRECTES DE QUESTIONNAIRES REALISABLES	63
Introduction	65
<u>CHAP. V. : CONSTRUCTIONS OPTIMALES</u>	67
V.1. Différentes méthodes possibles	67
- expression du coût optimal	67
- constructions aval-amont	68
- constructions amont-aval :	69
PSES	70
PSEP	70
V.2. Améliorations dues au problème lui-même	71
- élimination des redondances	72

- éventualités compatibles	73
- échange de questions successives	74
- cas particulier des questions élémentaires	75
VI.3. Etude d'un algorithme construisant des questionnaires optimaux	78
- présentation générale, organigramme de principe	78
- coût déjà dépensé en cours d'énumération	81
- stockage des solutions partielles - redondances	81
- variantes	82
- performances de l'algorithme	84
- suggestions pour recherches ultérieures	90
<u>CHAP. VI. : CONSTRUCTIONS HEURISTIQUES</u>	93
VI.1. Principes de la construction heuristique	93
- le critère heuristique préconisé habituellement	93
- une interprétation du critère-guide : rapport information de SHANNON sur coût	94
- autres critères possédant la propriété d'additivité	95
VI.2. Construction d'un critère heuristique nouveau	96
- choix pour les questions de base 2	97
- généralisation aux bases supérieures	101
VI.3. Comparaison théorique des coûts optimaux et heuristiques	105
VI.4. Méthodes heuristiques semi-énumératives	108
Conclusion	109
PARTIE III : APPLICATIONS	111
Introduction	113
<u>CHAP. VII. : DIAGNOSTIC D'INCIDENTS REPERTOIRES</u>	115
VII.1. Modèle	115
VII.2. Discussion du modèle	116

VII.3. Etablissement du catalogue d'incidents	117
VII.4. Implantation en site industriel	119
Conclusion	124
<u>CHAP. VIII. : TRADUCTION DES TABLES DE DECISIONS A</u> <u>ENTREES LIMITEES EN ORGANIGRAMMES</u>	125
Introduction	125
VIII.1. Les tables de décisions à entrées limitées	125
- TDEL complètement développée	125
- traduction en organigramme	126
- simplifications - consistance	128
- distinction conceptuelle entre table d'identification et TDEL	129
VIII.2. Améliorations possibles dues à la structure du problème	130
- conditions totalement pertinente	130
- conditions redondantes	131
VIII.3. Utilisation des résultats établis en seconde partie	132
- hypothèses	132
- traduction optimale	133
- traduction heuristique	134
VIII.4. Brève revue de littérature sur l'optimisation des tables de décision	138
Conclusion	139
CONCLUSION	141
INDEX	145
TABLE DES MATIERES	147