

M. VAYSSADE

J. P. A. BARTHES

Une approche informatique d'un problème de partitionnement complexe

RAIRO. Recherche opérationnelle, tome 15, n° 2 (1981),
p. 153-164

http://www.numdam.org/item?id=RO_1981__15_2_153_0

© AFCET, 1981, tous droits réservés.

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

NUMDAM

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

UNE APPROCHE INFORMATIQUE D'UN PROBLÈME DE PARTITIONNEMENT COMPLEXE (*)

par M. VAYSSADE ⁽¹⁾ et J. P. A. BARTHES ⁽¹⁾

Résumé. — *Cet article concerne la solution pratique des problèmes complexes d'affectation en optimisation combinatoire. Par suite du trop grand temps de réponse des méthodes classiques pour des problèmes hautement combinatoires on s'est attaché à montrer la faisabilité d'une approche informatique sous forme d'un logiciel interactif utilisant un ensemble d'heuristiques et donnant une solution approchée rapide. Une approche expérimentale, consistant à écrire un système opérationnel pour un cas d'application précis, nous a permis de tirer un certain nombre de conclusions.*

Mots clés : Recherche opérationnelle, problèmes de partitionnement complexes, approche heuristique, système interactif, combinatoire.

Abstract. — *This paper deals with actual solutions to complex matching problems. Experience shows that many classical mathematical models lead to great inefficiency and high response time. Therefore we offer a different approach based upon a central data structure combined with simple but fast search algorithms and heuristic pruning. An experimental system has been written for a specific application, which allows us to offer some guidance for actually solving problems in the same class.*

Keywords: Operations research, complex matching problems, heuristics approach, interactive system, combinatorial optimization.

1. PRÉSENTATION GÉNÉRALE

Une des branches importantes de la recherche opérationnelle est celle qui s'intéresse à la résolution des problèmes d'affectation. Or les dimensions des problèmes pratiques étant de plus en plus importantes, on est amené à rechercher les méthodes les mieux adaptées aux problèmes de grande taille. La variété des applications pratiques d'une part, la facilité toujours plus grande pour disposer dans chaque cas d'un miniordinateur spécialisé d'autre part, nous ont conduit à tenter de résoudre des problèmes d'affectation complexes à l'aide de ce type de matériel.

(*) Reçu août 1979.

⁽¹⁾ Département de Mathématiques Appliquées et d'Informatique, Université de Technologie de Compiègne, B. P. n° 233, 60206 Compiègne.

2. RAPPEL DES MÉTHODES CLASSIQUES

Les problèmes d'affectation peuvent être considérés, au point de vue mathématique, comme un cas particulier des problèmes linéaires en nombres entiers. La formulation mathématique en est simple et compacte :

$$\text{Min } CX \quad \text{avec } AX > B.$$

Malheureusement cette mise en forme élégante qui a fait ses preuves dans le cas de variables continues, ne débouche, pour les variables entières, sur des algorithmes efficaces, que pour des problèmes de taille modeste. Le grand nombre d'algorithmes proposés dans la littérature pour résoudre ce type de problèmes, en montre bien la difficulté :

D'une part, une grande variété de méthodes mathématiques ont été développées :

- la méthode des coupes (cutting plane) [6];
- les méthodes d'énumération implicite [11];
- réduction de la fonction coût [9];
- réduction des contraintes [2] suivie d'une énumération implicite [7]; théorie des groupes [24, 10];
- transformations algébriques et énumération [3];
- application de la notion de dualité [21].

D'autre part, un certain nombre de méthodes heuristiques cherchant des solutions approchées :

- solution approchée par recherche de bornes sur les variables [20];
- recherche heuristique d'une solution admissible [18] et aussi [4, 26].

Toutefois ces méthodes utilisaient toutes une formulation classique, et, en particulier dans le cas de problèmes complexes, cette formulation aboutit généralement à un nombre de variables et d'équations qui rend ces méthodes impraticables sur un miniordinateur.

3. UNE APPROCHE INFORMATIQUE

Il est clair sur les approches précédentes qu'une partie des difficultés rencontrées lors de la résolution de tels problèmes sous forme de « PLE » est due à la multiplication du nombre de variables nécessaires à une modélisation mathématique complète et cohérente, alors que le problème au départ peut être défini en termes simples. Autrement dit, un grand nombre de difficultés sont introduites lorsque l'on ramène le problème pratique au modèle théorique bien que celui-ci soit simple et bien connu. Ces difficultés n'apparaissent réellement qu'au moment où l'on essaie de résoudre le problème à l'aide d'algorithmes classiques.

Il est donc intéressant d'étudier une alternative consistant à abandonner un modèle mathématique simple débouchant sur des algorithmes compliqués au profit d'une représentation plus complexe du problème réel associée à des algorithmes plus simples.

Cette représentation se traduit par une description informatique et non comme un modèle mathématique [1].

L'effort d'analyse est alors reporté des algorithmes sur la structure des données, la partie algorithmique pouvant être consolidée par l'intervention d'un ou plusieurs décideurs dont l'expérience dans le domaine peut être utilisée pour accélérer le processus d'affectation.

L'approche classique a donc été abandonnée au profit d'un petit système fonctionnant à l'aide d'un algorithme d'affectation rustique et rapide entouré de divers modules dont le rôle est de présenter les résultats aux décideurs de la façon la plus claire possible, et de leur laisser prendre toutes les décisions ayant un caractère politique marqué.

Malheureusement, il n'existe pas, à l'heure actuelle, de théorie formalisée permettant la démonstration qu'une telle approche peut aboutir à une solution réalisable.

Il est vraisemblable que la mise au point d'une telle théorie, demande plusieurs années de recherche, aussi a-t-on décidé, dans une première étape, de faire une étude expérimentale.

Pour cela, on a essayé de choisir un problème qui soit représentatif d'une certaine complexité, et on a tenté de mettre au point un système capable de résoudre ce problème sur un miniordinateur non dédié.

4. PRÉSENTATION DU PROBLÈME CHOISI EN EXEMPLE

Le problème d'affectation d'étudiants à des groupes d'enseignement

L'application retenue concerne l'affectation d'étudiants, choisissant leurs cours « à la carte », à des groupes d'enseignement.

La complexité vient du nombre d'étudiants (plusieurs centaines), de la structure de l'enseignement (découpage en cours, TP, TD avec une périodicité et des quota variés), de la période prise en compte (5 années d'université), des variations possibles des données (addition ou suppression de groupes, changement d'emploi du temps, modification des quota, prise en compte des retardataires), de la contrainte sur le temps de réponse admis avant publication d'un résultat définitif (2 à 3 jours).

L'enseignement est décomposé en unités de valeurs (UV), une UV peut posséder une structure complexe et se diviser en activités diverses de durées et de

périodicités variables, la structure la plus fréquemment rencontrée étant : cours, travaux dirigés (TD) et travaux pratiques (TP) (*fig. 1*). Pour des raisons

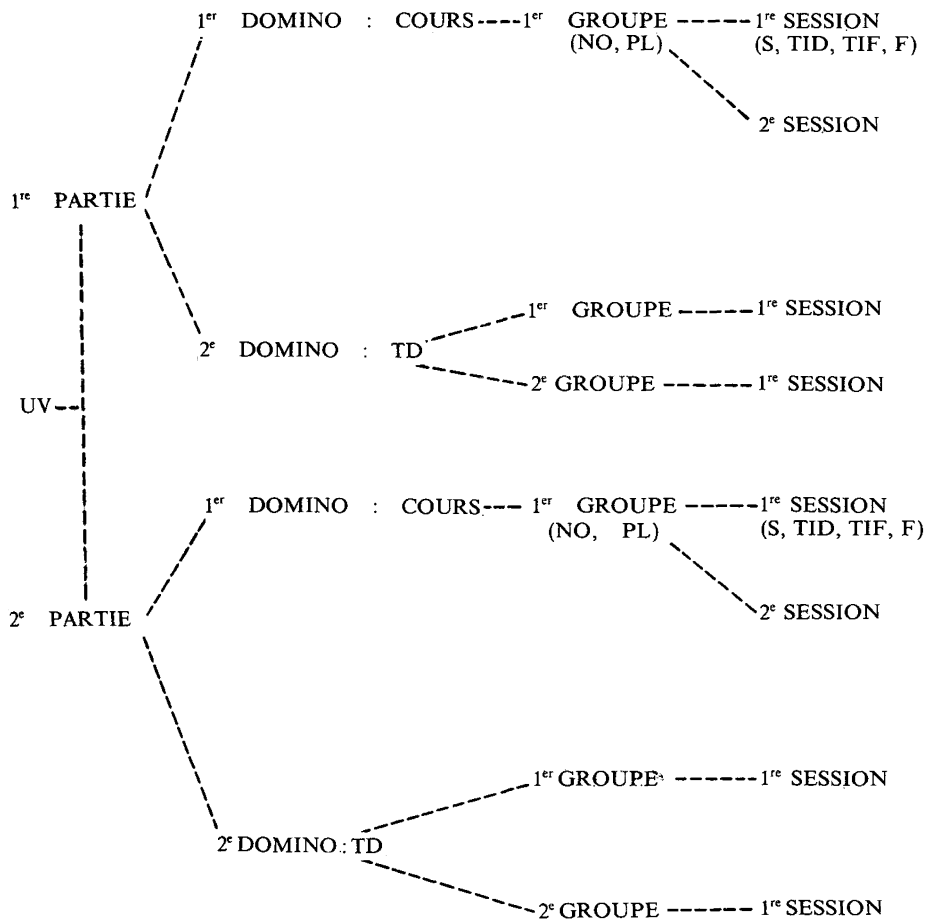


Figure 1. — Exemple de la structure d'une UV. NO, numéro global du groupe (sert de pointeur sur la table PLB); PL, nombre de places maximales; S, numéro de la salle (sert de pointeur sur SAL : table contenant le nom des salles); TID, numéro du TIK de début de session; TIF, numéro de TIK de fin de session; F, fréquence de la session (1 = toutes les semaines, 2 = toutes les 2 semaines...).

pratiques, dans chacune de ces activités, les étudiants inscrits à une UV sont séparés en groupes d'enseignement. Le nombre de groupes ouverts pour chaque UV est calculé au plus juste d'après le nombre d'étudiants inscrits ou prévus dans l'UV. L'étudiant a le choix de ses unités de valeur.

Pour utiliser au mieux les locaux disponibles, chaque groupe a des horaires le plus souvent différents des autres. Donc, selon qu'un étudiant est inscrit dans le premier groupe ou le deuxième, son emploi du temps sera différent. Il convient d'autre part de l'inscrire dans chacune des UV qu'il a demandée à des groupes dont les horaires sont compatibles entre eux. Cette contrainte est à respecter pour tous les étudiants.

Aspect combinatoire

Un ensemble d'affectations à des groupes pour un étudiant (exemple : 1° UV; TD : Groupe 1, TP Groupe 2; 2° UV, TD Groupe 3; TP Groupe 1, etc.) est appelé un ÉTAT de cet étudiant si toutes ces affectations sont compatibles entre elles, i. e. si l'étudiant n'a pas deux choses à faire en même temps (les étudiants sont supposés ne pas avoir le don d'ubiquité ce qui n'était pas toujours le cas jusqu'à présent, en particulier pour certains redoublants).

Un étudiant peut posséder de nombreux états (i. e. de nombreuses combinaisons compatibles) : le problème consiste donc à choisir pour chaque étudiant un état de telle sorte que les groupes de TD et TP soient remplis. C'est ici qu'apparaissent les problèmes combinatoires.

Le nombre de places offertes dans chaque groupe est limité. On ne peut donc pas choisir pour les étudiants un ensemble d'états tels que 50 d'entre eux seraient inscrits dans un groupe de 25 places. Il faut donc choisir une combinaison (i. e. un chemin dans le graphe) telle que les groupes soient remplis de façon homogène.

Or supposons que 30 étudiants n'aient qu'un seul état et que pour cet état tous soient inscrits dans un même groupe de 25 places : il y a impossibilité : 5 d'entre eux sont rejetés bien qu'ils aient une affectation possible. Le problème, simple à appréhender pour les étudiants à un état, est le même pour les étudiants à n états.

Il n'est pas démontré qu'un chemin résolvant le problème global de l'affectation existe. Quand bien même il en existerait un, il ne serait pas pour autant facile à trouver : en effet supposons n étudiants possédant chacun KI états, le nombre de combinaisons est $C = K_1 \times K_2 \times K_3 \dots \times K_I \dots \times K_N$.

Or, lors de l'utilisation opérationnelle à la rentrée d'automne 1976 on a montré que plus de la moitié des 400 étudiants traités possédaient 60 états ou plus, certains peuvent aller jusqu'à 500 états.

L'exploration systématique si rapide soit-elle est donc impossible.

L'utilisation de méthodes de branchement pour l'exploration de l'arbre a été envisagée mais la difficulté de trouver une fonction d'évaluation efficace a entraîné le choix de méthodes plus simples.

Approche informatique

Dans ce cas, après avoir choisi une structure de données, la démarche la plus simple est la suivante :

On commence par choisir le premier état du premier étudiant :

Puis, pour l'étudiant K , le premier état qui satisfait les contraintes de places, etc. en général, il y a P étudiants qui ne peuvent être affectés. On essaie de réduire P par des méthodes heuristiques.

La première conséquence néfaste de cette méthode est de remplir certains groupes de façon préférentielle alors que d'autres restent presque vides. Il faut donc faire en sorte que les groupes se remplissent de façon homogène au fur et à mesure que l'on progresse dans l'affectation.

On voit donc que cette approche ne peut être efficace sans l'aide de règles heuristiques.

La principale heuristique utilisée est fondée sur une remarque simple : tant qu'il reste de la place dans tous les groupes, on est sûr de pouvoir affecter l'étudiant suivant. Il faut donc essayer, lorsqu'on affecte un étudiant, de choisir un état, si possible, tel que aucune nouvelle contrainte de place ne devienne saturée par ce choix.

D'autre part, la solution optimale n'est pas nécessairement $Z = n$. Elle peut être $Z = n - R$, R = le nombre d'étudiants rejetés. Si R est trop important, il peut apparaître comme nécessaire de modifier la grille horaire et de refaire le calcul pour obtenir une nouvelle valeur $R1$ et ce jusqu'à ce que $R1$ soit jugé satisfaisant, d'où la préférence pour un algorithme rustique et rapide.

5. DESCRIPTION DE LA MÉTHODE : STRUCTURE DE DONNÉES, HEURISTIQUES

On a donc décidé de construire un prototype réalisant une approche informatique du problème.

On a tout d'abord écrit un premier prototype batch, fonctionnant sur une machine puissante pour voir quel était le comportement d'un tel système. Puis, ce premier prototype ayant donné satisfaction, on a réalisé un prototype interactif sur un miniordinateur (un PDP 11/45).

Organisation générale des données

La structure d'information manipulée par le système interactif est composée de données de deux types :

- des données statiques correspondant à la description initiale du système [25];

STRUCTURE DE L'ENSEIGNEMENT

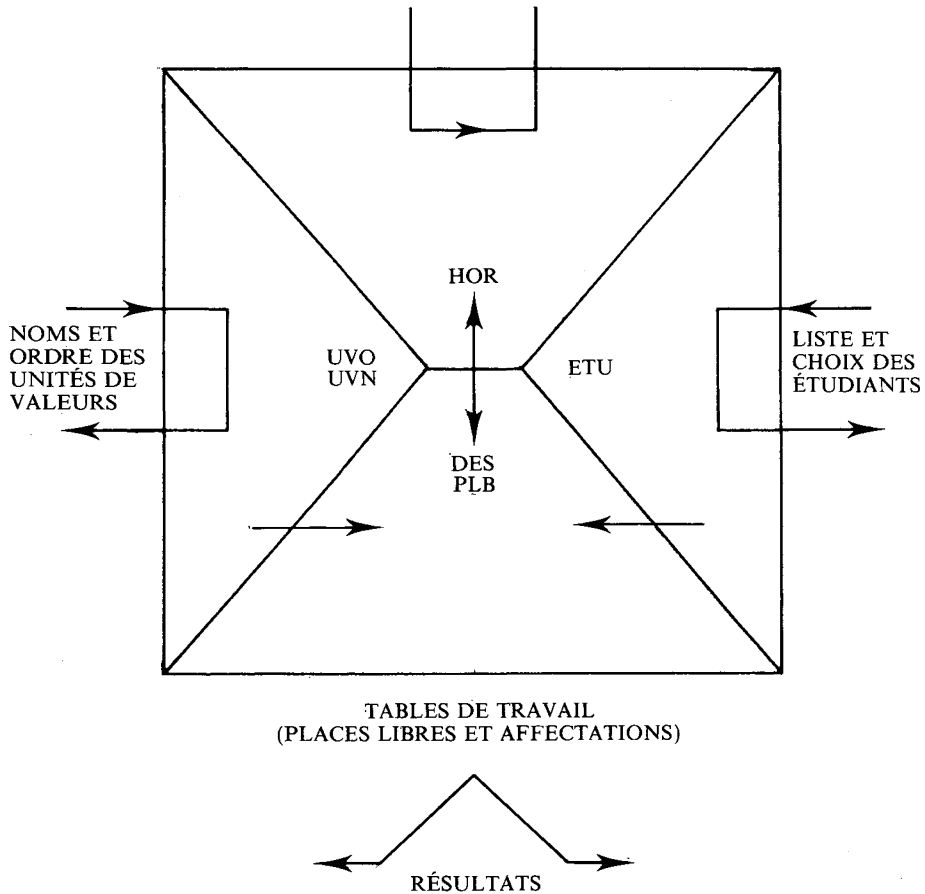


Figure 2. — Organisation générale des données. La zone des PLB correspond aux données dynamiques, les flèches à l'interaction avec l'utilisateur.

— des données dynamiques utilisées par les algorithmes d'affectation.

Les données statiques comprennent :

- la liste des unités de valeurs (noms et ordre);
- la structure de l'enseignement (pour chaque groupe d'enseignement; numéro de groupe, nombre maximal de places, salle, horaire, fréquence);
- la liste des étudiants (noms, choix des UV).

Les données dynamiques sont calculées et comprennent :

- la liste des places libres dans chaque groupe;

- la liste des affectations des étudiants.

Ces données servent à préparer les documents de sortie, tels que liste des inscrits à un groupe, fiches individuelles d'étudiants, etc.

L'ensemble des données principales est représenté sur la figure 2.

Pour les données statiques, des interpréteurs particuliers permettent à l'utilisateur de vérifier, modifier, et imprimer chaque zone de façon indépendante.

En pratique, les données sont conservées sous forme de tables résidant en mémoire ou stockées sur disque. Le détail du contenu de ces tables se trouve dans l'annexe technique.

Organisation générale des traitements

Les traitements sont effectués grâce à un dialogue utilisateur/système. L'utilisateur a une variété de commandes à sa disposition qui lui permettent à tout moment de vérifier ou modifier les données, faire une affectation, contrôler le résultat de l'affectation, obtenir des informations complémentaires concernant la cause de non-affectation, imprimer des listes variées concernant les données ou les résultats.

Les commandes sont traitées par une hiérarchie d'interpréteurs appelant les modules fonctionnels correspondants.

Les entrées peuvent se faire soit de manière individuelle et interactive, soit en masse (lecteur de cartes), le mode interactif pour saisir les données du type emploi du temps, les données étudiants et les choix des cours étant entrés à partir de paquets de cartes, chaque étudiant ayant constitué son paquet à partir de cartes perforées à l'avance.

Les contrôles sont faits en mode interactif sous forme de consultations ponctuelles.

Les sorties peuvent se faire sur l'imprimante, à tout moment.

6. L'UTILISATION ACTUELLE DU SYSTÈME

Le premier prototype du système a fonctionné en septembre 1976 avec 460 étudiants à répartir dans 150 groupes de 48 unités de valeur. Le pourcentage de cas non traités fut d'environ 12 %. Depuis, le système est régulièrement utilisé deux fois par an (il y a une rentrée par semestre) à l'Université de Technologie de Compiègne. En septembre 1980, il a aidé à répartir 799 étudiants dans 309 groupes d'enseignement de 97 unités de valeur. Seuls 43 cas ont dû être traités

manuellement, soit 5,4 %, c'est le pourcentage moyen de cas non affectés : l'utilisation d'heuristiques mieux adaptées a permis de diviser par deux en moyenne le pourcentage de cas non traités par rapport à l'essai du premier prototype.

Les utilisateurs habituels du système sont la Direction de l'Enseignement et de la Pédagogie, et les responsables de branches, qui, ensemble, possèdent l'autorité pour changer l'emploi du temps et ajouter ou supprimer des groupes d'enseignement. Ces utilisateurs n'ont pas de formation spéciale en informatique.

La conception « ouverte » du système a permis d'ajouter facilement de nouvelles fonctions. En 1979, à la demande des utilisateurs, un ensemble de sous-programmes a été ajouté en amont du système, pour permettre la saisie interactive des données relatives aux inscriptions des étudiants dans les unités de valeur, à partir de plusieurs terminaux simultanément. Ce nouvel ensemble a été utilisé en février et septembre 1980. En particulier, une de ces nouvelles fonctions, permet de tester immédiatement la compatibilité de la liste des unités de valeur auxquelles s'inscrit un étudiant, avec l'emploi du temps. Cette fonction de test immédiat a été beaucoup appréciée des utilisateurs.

L'utilisation du système est entrée depuis 2 ans dans une phase de « routine », l'intervention des informaticiens étant désormais réduite au minimum.

7. CONCLUSION

On peut tirer de cette expérience un certain nombre de conclusions :

Il est important, lorsque l'on effectue le calcul d'une affectation, que toutes les données nécessaires soient en mémoire, ou si ce n'est pas possible, qu'elles puissent être accédées rapidement, d'où une structure de données centrale : c'est sur la structure des données que doit porter l'effort principal d'analyse.

De plus, cette structure de données doit être organisée de façon à permettre une approche incrémentale des calculs, en particulier dans les algorithmes d'affectation.

Des expériences comparées des deux prototypes réalisées, on peut conclure que la formule interactive est de loin préférable, grâce à l'intervention de l'utilisateur, qui améliore le système s'il connaît bien le problème traité.

Une intégration en amont et en aval doit pouvoir être possible, sinon facile (en effet, l'apparition d'un système de ce type dans une organisation, change quelques habitudes, et peut amener à changer certaines méthodes de travail).

La façon de présenter les résultats, sous une forme habituelle ou non aux utilisateurs, a une grande influence sur l'efficacité globale du système.

Donc, en résumé :

- soigner la structure de données;
- permettre une approche incrémentale;
- choisir un système interactif;
- avoir à l'esprit une intégration possible;
- surtout, soigner l'interaction homme-machine.

Les résultats obtenus en utilisation opérationnelle ont confirmé le bien-fondé des hypothèses de départ. En effet, l'efficacité et la facilité d'emploi que l'on peut donner à un système interactif permet de le confier directement à l'utilisateur qui peut traiter ainsi son problème sans avoir à passer par l'intermédiaire d'un informaticien. En effet, dans le cas de problèmes complexes, l'expérience acquise et la connaissance du domaine d'application influent directement sur l'efficacité du système, et, partant de là, sur la performance globale.

De plus la conception même d'un système interactif permet de modifier les données jusqu'au dernier moment et de vérifier facilement que ces modifications ne détériorent pas la solution obtenue.

D'autre part, l'approche consistant à construire une structure de données centrale étoffée par des fonctions d'accès et de visualisation présente une grande souplesse et facilite l'addition de programmes et de fonctions si le besoin s'en fait sentir. De plus, le système est apte à prendre en compte des contraintes plus informelles difficilement modélisables sous forme d'équations mathématiques.

Cette recherche a donc montré qu'une approche informatique de problèmes d'affectation très complexes permettrait de construire un système d'aide à la décision capable d'apporter une aide considérable au décideur sans pour cela s'interposer entre son problème et lui. Cette approche apparaît beaucoup plus efficace que l'application de modèles mathématiques coûteux implantés sous forme de programmes automatiques, ou, pour des raisons pratiques, l'utilisateur perd une partie de la responsabilité de ses choix.

BIBLIOGRAPHIE

1. J. P. BARTHES et D. WILDE, *Linear Branching Algorithms in Combinatorial Optimizations*, Stanford University, Engineering, Design division.
2. G. H. BRADLEY, P. L. HAMMER et L. WOLSEY, *Coefficient Reduction for Inequalities in 0-1 Variables*, Department of Combinatorics and Optimization, University of Waterloo, Canada, Research report corr. 73-6, mars 1973.
3. G. H. BRADLEY et P. R. WAHI, *An Algorithm for Integer Linear Programming: A Combined Algebraic and Enumeration Approach*, Administrative Sciences, Yale, Report n° 29, décembre 1969, révisé février 1971.

4. D. G. DANNENBRING, *Procedures for Estimating Optimal Solution Values for Large Combinatorial Problems*, Management Science, vol. 23, n° 12, août 1977, p. 1273-1283.
5. J. DELORME et E. HEURGON, *Problèmes de partitionnement : Exploration arborescente ou méthode de troncatures*, R.A.I.R.O., 9^e année, vol. 2, juin 1975, p. 53-65.
6. R. GARFINKEL et G. NEMHAUSER, *Integer Programming*, John Wiley and Sons, New York, 1972.
7. R. S. GARFINKEL et G. L. NEMHAUSER, *The Set-Partitioning Problem: set Covering with Equality Constraints*, Operational research, vol. 17, 1969, p. 848-856.
8. M. GONDRAAN et J. L. LAURIÈRE, *Un algorithme pour les problèmes de recouvrement*, R.A.I.R.O., 9^e année, vol. 2, juin 1975, p. 33-51.
9. M. GONDRAAN et J. L. LAURIÈRE, *Un algorithme pour le problème de partitionnement*, R.A.I.R.O., 8^e année, vol. 1, janvier 1974, p. 27-40.
10. G. A. GORRY et J. F. SHAPIRO, *An Adaptive Group. Theoretic Algorithm for Integer Programming Problems*, Management Science, vol. 17, n° 5, janvier 1971, p. 285-306.
11. H. GREEBERG, *Integer Programming*, Academic press, New York, 1971.
12. *Guide de l'Université de Compiègne*, 1976-1977, U.T.C., 60200 Compiègne.
13. R. W. HAISE, L. D. NELSON et T. RADO, *Computer Studies of a Certain Class of Linear Integer Problems*, Ohio State University Research Foundation.
14. P. L. HAMMER, E. L. JOHSON et U. N. PELED, *Regular 0-1 Programs*, Department of Combinatorics and Optimization, Research report corr 73-18, University of Waterloo, Canada, septembre 1973.
15. P. L. HAMMER, M. W. PADBERG et U. N. PELED, *Constraint Pairing in Integer Programming*, Department of Combinatorics and Optimization, Research report corr 73-7, avril 1973, University of Waterloo, Canada.
16. E. HEURGON, *Développement actuel des méthodes de construction automatique des tableaux de service*, R.A.I.R.O., recherche opérationnelle, vol. 10, n° 2, février 1976, p. 113-119.
17. F. S. HILLIER, *A Bound-and-Scan Algorithm for Pure Integer Linear Programming with General Variables*, Department of operations research, Stanford University, Technical report n° 11, 20 mai 1969.
18. F. S. HILLIER, *Efficient Heuristic Procedures for Integer Linear Programming with an Interior*, Department of operations research, Stanford University, Technical report n° 2, 28 février 1969.
19. A. KAUFMANN et A. HENRY-LABORE, *Méthodes et modèles de la recherche opérationnelle*, tomes 1, 2, 3, Dunod, Paris, 1970, 1972, 1974.
20. P. KROLAC, *The Bounded Variable Algorithm for String Integer Linear Programming Problems*, Thesis, Washington University, janvier 1968, Saint-Louis, Missouri.
21. J. P. LEGENDRE et M. MINOUX, *Une application de la notion de dualité en programmation en nombres entiers : sélection et affectation optimales d'une flotte d'avions*, R.A.I.R.O., vol. 11, n° 2, mai 1977, p. 201-222.
22. B. ROY, *Algèbre moderne et théorie des groupes*, Dunod, Paris, 1970.
23. D. S. RUBIN, *Integral Solutions of Integral Linear Systems*, University of Chicago.
24. H. THIRIEZ, *The set Covering Problem: a Group Theoretic Approach*, R.A.I.R.O., 5^e année, vol. 3, octobre 1973, p. 83-104.
25. M. VAYSSADE et J. P. A. BARTHES, *PAF 1.0*, U.T.C., mai, p. 76-23.
26. Stelios H. ZANAKIS, *Heuristic 0.1 Linear Programming; an Experimental Comparison of three Methods*, Management Science, vol. 24, n° 1, september 1977, p. 91-103.

ANNEXE TECHNIQUE

CONTENU DES PRINCIPALES TABLES DE LA STRUCTURE DE DONNÉES

