REVUE FRANÇAISE D'AUTOMATIQUE INFORMATIQUE RECHERCHE OPÉRATIONNELLE. MATHÉMATIQUE

IVAN LAVALLEE

Un algorithme de détermination de couvertures de cardinal minimal

Revue française d'automatique informatique recherche opérationnelle. Mathématique, tome 7, n° R2 (1973), p. 17-28

<http://www.numdam.org/item?id=M2AN_1973__7_2_17_0>

R.A.I.R.O.
(7e année, R-2, 1973, p. 17-28)

UN ALGORITHME DE DETERMINATION DE COUVERTURES DE CARDINAL MINIMAL (P 4)

par Ivan LAVALLEE (1)

Résumé. — *On donne, dans cet article, un algorithme de détermination d'une couverture de cardinalité minimale d'un ensemble à partir d'une sous famille de parties de celui-ci.*

Pour ce faire, on pose le problème en termes de réduction d'une fonction booléenne explicitée sous forme d'un produit booléen de sommes booléennes. Cet algorithme ne fait appel qu'à la formulation boolèenne du problème, ce qui en fait la relative simplicité, et en facilite l'exploitation en machine.

Pour résoudre ce problème, on utilise seulement :

— Une borne minimale pour la cardinalité de la couverture (cette borne est donnée par la condition C_1).

— Un ordre lexicographique induit par la liste des lettres ordonnées suivant leur nombre d'occurrences décroissantes dans l'ensemble des sommes du produit booléen considéré.

— Une énumération implicite partielle dans l'ensemble des parties de l'ensemble à couvrir.

— Une limite d'énumération au-delà de laquelle l'énumération en cours devient inutile (condition C_2).

— Un pas calculable pour l'incrémentation de la borne minimale du cardinal de couverture.

Le problème des recouvrements est l'un des thèmes majeurs de la recherche opérationnelle; il a donné lieu à de nombreuses études (par exemple : Balinsky, Malgrange, Roy...).

Or il apparaît dans de nombreux problèmes, qu'on recherche des recouvrements minimums, c'est-à-dire des recouvrements minimaux qui sont de plus, de cardinalité minimale.

Il en est ainsi dans de nombreux problèmes de théorie des graphes; problèmes de la recherche des ensembles absorbants minimaux, des ensembles stables maximaux, des ensembles transversaux d'un hypergraphe, etc... (voir à ce sujet Herz [P 2], Berge [L 1], Maghout [L 12], Pichat [L 15]).

(1) Ingénieur informaticien, chercheur au C.N.R.S. (groupe de recherche Structures de l'Information, Université (I.I.E.), Paris, VI).

Notre propos sera ici de donner un algorithme permettant de déterminer de telles couvertures de cardinalité minimale en n'ayant recours qu'à la formulation booléenne du problème.

RAPPELS

Soit $E = \{ e_i \mid i \in M \}$, $M = \{ 1, 2, ..., m \}$. On dit qu'une famille

$$B = \{ b_j \mid b_j \subset E \quad , \quad b_j \neq \varnothing \quad , \quad j \in J \quad , \quad J \subset N^* \}$$

est un recouvrement de l'ensemble E si elle possède en sus la propriété suivante

$$\bigcup_j b_j = E$$

Un recouvrement de E est donc une famille de parties non vides de E telle que pour tout élément e_i de E, il existe au moins un $j \in J$ tel que :

$$e_i \in b_j$$

On dit alors qu'il y a recouvrement de E par les b_j (on utilisera indifféremment le terme de couverture et celui de recouvrement).

Exemple :

$$E = \{ 1, 2, 3, 4 \}$$
$$B = \{ a, b, c, d \}$$

avec : $a = \{ 1, 2, 4 \}$; $b = \{ 3, 4 \}$; $c = \{ 2, 4 \}$; $d = \{ 1, 2 \}$.

Une couverture de E sera par exemple :

$$\{ a, b, c \} \quad \text{ou encore} \quad \{ b, d \}.$$

En effet :

1 est couvert par a

2 est couvert par a et c

3 est couvert par b

4 est couvert par a et b et c.

On remarquera que tout sur-ensemble d'une couverture est une couverture; par exemple ici, $\{ a, b, d \}$ est couverture de E.

LE PROBLEME DES COUVERTURES

En règle générale, on ne cherche pas n'importe quelle couverture, le problème central du recouvrement est la recherche de couvertures ayant en sus d'autres propriétés, c'est-à-dire le plus souvent :

1) des couvertures *minimales* (couverture dont aucun sous-ensemble strict n'est une couverture) [L 15] ;

2) des couvertures rendant minimal le chevauchement de leurs parties, en particulier les partitions ;

3) des couvertures de coût minimal (quand chaque partie est affectée d'un coût) [L 16] ;

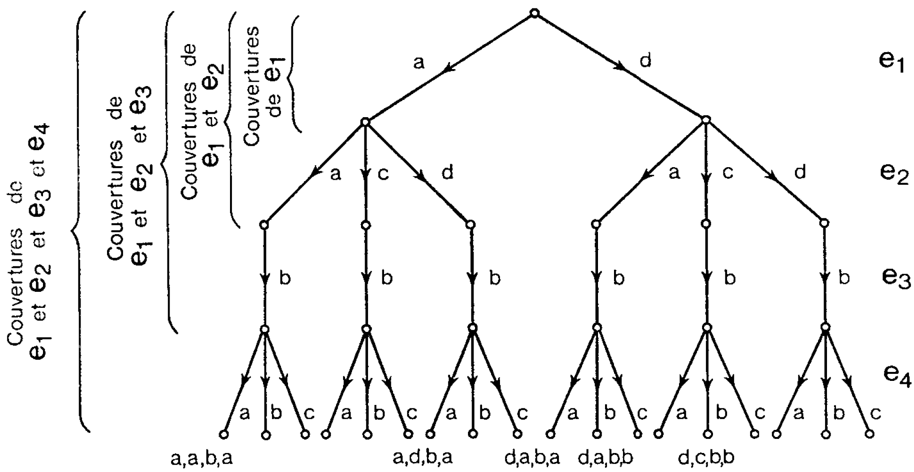
4) des couvertures de cardinalité minimale (ce sont des couvertures minimales particulières).

C'est à ce dernier aspect du problème que nous nous attacherons dans le présent article.

On remarquera d'ailleurs que les cas 2 et 3 se réduisent aux cas 1 et 4 lorsque la fonction coût d'une couverture est une fonction croissante des coûts de ses éléments.

FORMULATION BOOLEENNE DU PROBLEME (L 6)

Toute couverture contient l'une au moins des parties couvrant chaque élément de *E*.



Par exemple les couvertures :

de 1 contiennent au moins a ou d,

de 2 contiennent au moins a ou c ou d, donc les couvertures de 1 et 2 contiennent a ou (a, c) ou (a, d) ou (c, d) ou d, soit au moins a ou d,

de 3 contiennent au moins b, donc les couvertures de 1, 2 et 3 contiennent au moins (a, b) ou (b, d),

de 4 contiennent au moins a ou b ou c, donc les couvertures de 1, 2, 3, et 4 contiennent (a, b) ou (a, b, c) ou (a, b, d) ou (b, d) ou (b, c, d), soit au moins (a, b) ou (b, d).

Cette démarche peut se représenter par une arborescence à valeurs dans B.

Chaque chemin de l'arborescence, de sa racine à une de ses feuilles correspond à une couverture (l'ensemble des valeurs du chemin). Inversement, à toute couverture minimale correspond au moins un chemin dont l'ensemble des valeurs est la couverture minimale. Ainsi, à la couverture a, b, d correspondent au moins les deux chemins (d, a, b, a) et (d, a, b, b).

Pour mettre en évidence les couvertures minimales, nous avons utilisé dans cet exemple les propriétés d'associativité, de commutativité, d'idempotence, de distributivité mutuelle et vérifié la règle d'absorption, toutes propriétés des opérations d'intersection et d'union ensembliste qui s'appliquent conséquemment aux sous-ensembles que sont les couvertures.

Les couvertures minimales d'un ensemble peuvent donc être obtenues par développement d'un produit booléen de sommes booléennes

$$\prod_{\text{éléments de } E} \sum (\text{éléments de } B \text{ couvrant } e_i,\ e_i \in E)$$

Soit encore en reprenant les définitions du début :

$$E = \{ e_i \mid i \in M \} \quad , \quad M = \{ 1, 2, \ldots, m \}$$

et

$$B = \{ b_j \mid b_j \subset E \quad , \quad b_j \neq \varnothing \quad , \quad j \in J \quad , \quad J \subset N^* \}$$

On considère pour i donné :

$$C_i = \bigcup_k b_k : \{ b_k \mid e_i \in b_k \quad \text{et} \quad b_k \in B \}$$

Les couvertures minimales seront alors des sous-ensembles de P tels que tout e_i soit élément de la couverture minimale S_l et que cette couverture n'en contienne strictement aucune autre.

Si l'on considère

$$P = \bigcap_i C_i$$

alors S couverture minimale de E

$$\forall\, e_i,\ e_i \in S_l \qquad \text{et} \qquad S_l \subset P$$

et

$$\nexists\, S_k \subset S_l : \forall\, i,\ e_i \in S_k$$
$$P = \bigcup_l S_l$$

Ce qui exprimé en termes booléens peut se mettre sous forme d'un produit booléen de sommes booléennes à réduire.

$$\prod_{\forall i \in M\, M=\{i \mid e_i \in E\}} \sum (\text{éléments de } B \text{ couvrant } e_i).$$

Le résultat se présente sous forme d'une somme de produits dont chacun représente une couverture minimale (c'est ce qui correspond à $P = \bigcup_l S_l$).

Pour notre problème, nous nous contenterons d'écrire le produit de sommes, sans chercher à le développer.

Dans l'exemple précédent, la forme booléenne serait :

$$(a + d)(a + c + d)(b)(a + b + c)$$

En développant ce produit, on obtient les couvertures minimales :

Soit : *ab*, *bd*.

Pichat [L 15] a résolu le problème de la détermination des couvertures minimales dans le cadre de la théorie des treillis; la liaison se fera facilement avec ce qui précède. On pourra aussi généraliser l'algorithme ci-dessous dans le cadre de la théorie des treillis comme nous l'avons fait pour l'algorithme de Pichat [L 10].

De plus, on fera aisément la liaison avec la théorie des graphes et des hypergraphes en considérant que les b_j (ici a, b, c, etc...) forment les arêtes d'un hypergraphe [L 1], [L 10], [L 15].

Nous proposons donc ici un algorithme permettant de déterminer directement au moins une couverture de cardinal minimal.

NOTATIONS

Par rapport à ce qui précède, on entendra par « lettre », dans l'énoncé de l'algorithme, tout b_j élément de B.

De plus, nous considérons deux tableaux T et P définis comme suit :

T tableau des lettres ordonné suivant le nombre d'occurrences de chaque lettre dans le produit de sommes booléennes considéré.

P tableau des nombres d'occurrences (que nous appellerons désormais : Poids des lettres) des lettres, tel que l'ordre sur T corresponde à l'ordre des poids sur P. Ainsi la $k$ième lettre de T a pour poids le $k$ième élément du tableau P.

ALGORITHME ORDONNE DIRECT DE DETERMINATION D'UNE COUVERTURE DE CARDINAL MINIMAL

I. Constituer un tableau T des lettres, ordonnées suivant leur nombre d'occurrences (ce nombre sera, ici, appelé « poids » d'une lettre) décroissant et un tableau P des poids des lettres ordonné comme T.

En notant $p(i)$ le poids de la $i$ème lettre de T :

$$p(i) \in P$$

et

$$j \geqslant i \Leftrightarrow p(j) \geqslant p(i).$$

II. A) On construit la séquence des q premiers termes de $T(t_1, t_2, \ldots, t_q)$, avec q tel que :

$$\sum_{i=1}^{i=q-1} p(i) < F \leqslant \sum_{i=1}^{i=q} p(i) \qquad \text{(condition } C_1\text{)}$$

F étant ici le nombre de facteurs du produit initial.

B) On construit, suivant l'ordre lexicographique sur T les séquences successives de q termes. Soit S_k une telle séquence.

Pour construire S_k on pratique de la façon suivante.

On introduit $t_i \in T$ (t_i première lettre possible dans l'ordre sur T) dans la séquence en formation.

On élimine du produit tous les facteurs couverts par t_l, t_l étant la $l^{\text{ième}}$ lettre de S_k. Si l'introduction de t_l dans la séquence ne provoque l'élimination d'aucun facteur du produit, on supprime t_i de la séquence en formation, et on s'interdit de construire toutes les séquences ayant les $l - 1^{\text{ième}}$ premières lettres identiques, et contenant t_i.

On tente ensuite d'introduire t_{i+1} (on continue jusqu'à ce que S_k soit de cardinal q).

En fin de construction d'une séquence S_k, il faut envisager plusieurs cas.

1) Le nombre de facteurs du produit résiduel ([1]) est nul. On exhibe la séquence S_k qui est couverture de cardinal minimal.

(1) Nous appelons produit résiduel, que nous noterons Π_r, l'ensemble des facteurs de Π non couverts par S_k.

REMARQUE

Si on désire obtenir toutes ou plusieurs couvertures de cardinal minimal, on continuera la construction de séquences S_k en reprenant directement le début de B, sans tenir compte du paragraphe II. B. 2). Mais on tiendra compte de notre remarque 2 qui évitera de construire certaines séquences inutiles.

L'algorithme est terminé.

2) Le nombre de facteurs du produit résiduel n'est pas nul. On considère alors les facteurs formant Π_r. On crée un tableau P' issu de P en ne considérant dans $T - S_k$ que les lettres suivantes de la dernière lettre introduite dans S_k.

Soit alors q_r le nombre tel que :

$$\sum_{i=1}^{i=q_r-1} p'(i) < |\Pi_r| \leqslant \sum_{i=1}^{i=q_r} p'(i) \qquad p'(i) \in P'$$

III. Notons $q_{r\ \min}$ le minimum des q_r calculés, quand il n'est pas nul.

On prend alors comme nouveau minorant du cardinal minimal, le nombre :

$$q' = q + q_{r\ \min}$$

et on recommence au paragraphe II.B avec :

$$q \leftarrow q'$$

PREUVE

Validité du paragraphe III.2

Le plus petit cardinal possible pour le produit Π est :

$$q' = q + q_{r\ \min}$$

En effet, soient :

Π le produit initial;

Π_c le produit « couvert » par une séquence S_k afférente à $q_{r\ \min}$;

Π_r le produit résiduel.

on a :

$$\Pi = \Pi_c \, . \, \Pi_r$$

La séquence S_k couvrant Π_c est de cardinal q.

Le cardinal minimum possible pour couvrir Π_r est $q_{r\ \min}$.

Ces deux séquences sont disjointes (sinon cela voudrait dire que la séquence couvrant Π_c couvrirait d'autres facteurs de Π_r dont Π_c et Π_r auraient été mal construits).

Par construction de q et $q_{r\ \min}$, il n'existe pas de séquence de cardinal p telle que :

$$q < p < q' \quad \text{avec} \quad q' = q + q_{r\ \min}$$

couvrant Π.

REMARQUE 1

Si on a trouvé une couverture de cardinal minimal, elle est minimale.

REMARQUE 2

Il peut arriver en cours de construction d'une séquence S_k que l'on introduise la $i$ème lettre du tableau T et que l'on ait :

$$F - |\ \Pi_{S_k - \{t_i\}}\ | - p(i) > \sum_{j=1}^{j=q_{r\min}} p'(i) \qquad (1)$$

(condition C_2)

On peut alors arrêter la construction de S_k car on aura toujours :

$$q_r \geqslant q_{r\ \min}$$

et on passe alors au paragraphe III de l'algorithme.

REMARQUE 3

Il s'avère qu'appliquée aux graphes, la condition C_1 permet de retrouver le théorème de Berge [L 1], p. 266, qui permet de démontrer le théorème de Turan.

EXEMPLE ET SCHEMA D'ETUDE EN MACHINE

Etant donné un produit sous sa forme littérale, le problème est de l'implanter le plus facilement possible en machine.

Nous proposons donc :

On crée une matrice binaire $A = [a_{ij}]$ de la façon qui suit :

Les vecteurs colonnes représentent les « lettres », les vecteurs lignes, représentent les facteurs.

$a_{ij} = 1$ la $j$ème lettre figure dans le $i$ème facteur.

$a_{ij} = 0$ sinon.

Les données se présentent alors en machine sous forme d'une matrice binaire [L 10].

(1) $|\Pi_{S_k - \{t_i\}}|$ est le nombre de facteurs couverts par la séquence $S_k - \{t_i\}$ avec $|S_k - \{t\}| = q - 1$.

Donnons un exemple simple :

Soit le produit

$$(a + b + d)(d + b + e + f)(a + c + e)(a + f)(c + d)$$

On construit la matrice binaire représentative de ce produit

	a	b	c	d	e	f
1	1	1	0	1	0	0
2	0	1	0	1	1	1
3	1	0	1	0	1	0
4	1	0	0	0	0	1
5	0	0	1	1	0	0
Nombres d'occurrences	3	2	2	3	2	2

L'ordre que nous prendrons sur les lettres est donc ici :

$$a \quad d \quad b \quad c \quad e \quad f$$

REMARQUE : Signalons toutefois que Laurière [L 9] propose un autre ordre sur les lettres en utilisant la notion de « degré généralisé », il s'avère que cet ordre est particulièrement efficace dans le cas de recouvrements exacts et notamment dans le problème d'habillage des horaires d'une ligne d'autobus [P 3] et dans d'autres problèmes de même nature [L 9].

La première séquence possible est donc de cardinal 2 puisque :

$$p(a) + p(d) = 6 > F \qquad (F = 5)$$

$$\begin{array}{lcccccc} & a & d & b & c & e & f \\ P_1 = & 3 & 3 & 2 & 2 & 2 & 2 \end{array}$$

Nous introduisons donc a comme première lettre dans la séquence, couverture plausible, et nous supprimons du produit initial tous les facteurs couverts par a.

La matrice binaire représentative du produit résiduel devient alors :

$$\begin{array}{c|cccccc} & a & d & b & c & e & f \\ \hline 2 & 0 & 1 & 0 & 1 & 1 & 1 \\ 5 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}$$

C'est-à-dire, sous forme littérale :

$$(b + d + e + f)(c + d)$$

avec :

$$\begin{array}{lcccccc} & a & d & b & c & e & f \\ P_2 = & 0 & 2 & 1 & 1 & 1 & 1 \end{array}$$

En appliquant notre algorithme, nous introduisons la lettre « d » dans la séquence en construction, elle couvre tous les facteurs restants. Donc *ad* est une couverture de cardinal minimal pour le produit considéré.

Nous savons désormais que toutes les couvertures de cardinalité minimale sont de cardinal 2. Et, en poursuivant la construction des séquences, il apparaîtrait dans le cas présent que seule la couverture *ad* répond à la question.

ESSAIS EN MACHINE

Nos essais ont été réalisés sur l'ordinateur 360-65 du Centre IBM Sablons, avec pour mode d'accès le système R.J.P.

Nous avons généré de façon aléatoire les matrices binaires représentant nos données, car nous ne nous intéressions a priori qu'aux dimensions de la matrice binaire, ce qui fait que nous n'avons pas pris la peine de tenter une première simplification du problème par la règle d'absorption.

En outre, notre programme a été conçu pour utiliser le moins de place possible en mémoire, et il serait possible d'améliorer les temps obtenus en utilisant plus de place.

Le programme a été réalisé en langage FORTRAN IV-G.

Occupation mémoire

Compilation (FORTRAN IV-G)	150 K
Edition des liens	132 K
Exécution	46 K

Temps d'exécution

Nous avons testé, dans un premier temps, les temps d'exécution sur des matrices carrées. La nature même de l'algorithme fait que l'influence du nombre de facteurs est minime, et que le temps est surtout fonction du nombre de variables.

En effet, le temps passé est essentiellement fonction de l'énumération, laquelle varie en fonction du nombre des combinaisons des n lettres q à q.

Dimensions de la matrice	Temps en 10^{-4} heures
10	1
20	9
25	14
30	18
35	36
36	86
37	108
38	138
39	160
40	117
41	127
42	78
43	84
45	21
50	259

Ces temps ont été obtenus avec un remplissage de 50 % de la matrice binaire, elle-même générée aléatoirement.

Jusqu'à trente six variables, la courbe s'apparente à une fonction exponentielle, mais des facteurs perturbateurs interviennent ensuite.

En effet, à partir d'une certaine « dimension du problème » la remarque 2 permet de restreindre beaucoup l'énumération des séquences inutiles. La programmation de cette condition requiert tout de même un certain temps d'exécution et, tant qu'elle n'intervient pas de façon utile (c'est-à-dire tant que l'écart n'est pas nettement supérieur à q initial) elle fait perdre du temps. Mais dès qu'elle se révèle efficace, elle en fait gagner énormément. De plus, il arrive aussi que la séquence couverture soit « bien située » dans l'énumération et qu'elle soit trouvée plus vite que la moyenne (cela semble être le cas ici pour 45 lettres).

BIBLIOGRAPHIE

LIVRES-THÈSES

[L 1] BERGE C., *Graphes et Hypergraphes*, Dunod, 1971.

[L 2] BOUCHON B., Thèse de 3e Cycle, *Réalisations de questionnaires et propositions logiques*, Paris VI, juin 1972.

[L 3] BOURBAKI N., *Ensembles ordonnés*, fascicule XX, Hermann, 1968.

[L 4] BOURBAKI N., *Structures*, fascicule XXII, Hermann, 1966.

[L 5] BIRKOFF G., *Lattice theory A.M.S.*, Colloquim publications, volume XXV, 1948.

[L 6] C.N.A.M., *Initiation à la théorie des graphes*, Conférences télévisées, Éditions Scientifiques, Riber, p. 89-117, 1971-1972.

[L 7] FAURE R., DENIS-PAPIN M. et KAUFFMANN, *Cours de calcul booléen appliqué*, 1re édition 1963, 2e édition 1971, Albin Michel, p. 230-235.

[L 8] KAUFMANN, *Initiation à la combinatorique en vue de ses applications*, Dunod, 1968.

[L 9] LAURIERE J. L., *Sur la coloration de certains hypergraphes*, Thèse de 3e Cycle, Paris VI, 23 juin 1971.

[L 10] LAVALLEE I. et LIGNAC (C. de), *Contribution à l'algorithmique non numérique dans les structures ordonnées*, Mémoire d'ingénieur (I.I.E.), juin 1971.

[L 11] LEMAIRE B., *Problèmes de tournées avec contraintes multiples*, Thèse d'Ingénieur-Docteur, Paris VI, 9 décembre 1971.

[L 12] MAGHOUT K., *Applications de l'algèbre de Boole à la théorie des graphes*, Cahiers du Centre d'Études et de Recherche Opérationnelle, Bruxelles, vol. 11, n° 1-2, 1963.

[L 13] MALGRANGE Y., *Recherche des sous-matrices premières d'une matrice à coefficients binaires*. Pages 230-242, 2e Congrès AFCALTI, octobre 1961, Gauthier-Villars, 1962.

[L 14] MALGRANGE Y. et DENIS-PAPIN M., *Exercices de calcul booléen avec leurs solutions*, Eyrolles, 1966.

[L 15] PICHAT E., *Contribution de l'algorithmique non numérique dans les ensembles ordonnés*, Thèse d'État, Grenoble, octobre 1970.

[L 16] ROY B., *An algorithm for a general constrained set covering problem* into Graph Theory and Computing, Read editor, Academic Press inc. New-York, 1972.

[L 17] ROY B., *Algèbre moderne et théorie des graphes*, Dunod, 1969-1970.

REVUES. ARTICLES. PUBLICATIONS.

[P 1] BALINSKY M. L., *Integer Programming : Methods uses computation*, Management Science, vol. 12, n° 3, 1965.

[P 2] HERZ, *Note sur le problème des tables rondes*, Recherche de cliques dans les très grands graphes. Étude n° 772 — 0079 — 0 IBM développement scientifique, 1969.

[P 3] HEURGON, *Un problème de recouvrement : l'« Habillage des horaires d'une ligne d'autobus*, R.A.I.R.O. (6e année, n° V-1, 1972, p. 13-29).

[P 4] LAVALLEE I., *Un algorithme de détermination d'une couverture de cardinal minimal*. Étude n° 141, service 161 — IBM développement scientifique 1972.