MATHÉMATIQUES ET SCIENCES HUMAINES

ISRAËL-CÉSAR LERMAN
VALÉRIE ROUAT

Segmentation de la sériation pour la résolution de #SAT

Mathématiques et sciences humaines, tome 147 (1999), p. 113-134
<http://www.numdam.org/item?id=MSH_1999__147__113_0>

Math. Inf. Sci. hum., (37e année, n° 147, *1999*, pp. 113-134)

SEGMENTATION DE LA SÉRIATION POUR LA RÉSOLUTION DE #SAT

Israël-César LERMAN[1], Valérie ROUAT[2]

RÉSUMÉ – *Le problème général traité est celui de l'évaluation approchée du nombre de solutions d'une formule booléenne F sous forme normale conjonctive. En appliquant le principe «diviser pour résoudre», la méthode présentée permet de réduire de façon considérable la complexité algorithmique du problème. Elle est basée sur la segmentation d'une sériation établie sur la table d'incidence associée à F. Nous montrons, dans des cas aléatoires difficiles de génération d'une formule F, l'intérêt de la sériation et de sa meilleure coupure en deux parties connexes et de tailles comparables. De plus, nous définissons la notion d'indépendance en probabilité pour F. On propose ici et on valide théoriquement et par une vaste expérimentation la méthode.*

MOTS-CLÉS – Satisfiabilité, théorie de la complexité, dénombrement de solutions, problèmes #P-complets, classification, sériation.

SUMMARY – Cutting seriation for approximate #SAT resolution

We propose here a general method for approximating the number of solutions of a boolean formula in conjunctive normal form F. By applying the principle «divise to resolve», this method reduces considerably the computational complexity. It is based on cutting a seriation established on an incidence data table associated with F. Moreover, the independence probability concept is finely exploited. Theoretical justification and intensive experimentation validate the proposed method.

KEYWORDS – Satisfiability, complexity theory, counting, #P-complete problems, classification, seriation.

1. INTRODUCTION

Relativement à un ensemble $V = \{x_1, \ldots, x_N\}$ de variables booléennes, une clause est une disjonction de littéraux de la forme $y_1 \vee \ldots \vee y_q \vee \ldots \vee y_r$ $(r < N)$, où y_q est pour un indice i, de la forme x_i ou $\neg x_i$. Une affectation des variables booléennes satisfait la clause, si l'une au moins des variables $y_q, 1 \leq q \leq r$, est à vrai. Une

[1] IRISA/Université de Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex, e-mail : lerman@irisa.fr

[2] CRIL, rue de l'Université, S.P. 16, 62307 Lens Cedex, e-mail : rouat@cril.univ-artois.fr

instance SAT est une conjonction de clauses construites sur V. Le problème SAT de la SATisfiabilité de l'instance est celui de l'existence d'une solution, c'est-à-dire d'une affectation des variables de V qui mette à vrai l'instance. Le problème #SAT est celui de l'évaluation du nombre de solutions d'une instance SAT.

Le problème SAT est à l'origine de la définition des problèmes NP-complets [3]. Ils constituent la sous-classe la plus difficile des problèmes NP (Non déterministes Polynomiaux). Ils regroupent une très large classe de problèmes de décision. Démontrer qu'il n'existe pas d'algorithme polynomial pour résoudre le problème posé ($P \neq NP$) est au centre de la théorie de la complexité en informatique et correspond d'ailleurs à une conjecture qui dure depuis plus d'un quart de siècle.

Le problème #SAT, dénombrement de toutes les solutions d'une instance SAT est à l'évidence et a priori plus difficile que le problème précédent, il se situe en effet dans la classe des problèmes #P-complets [25]. La classe ainsi définie a une importance toute particulière en théorie de la complexité puisqu'elle « contient » toute la hiérarchie polynomiale [24]. On comprend dans ces conditions pourquoi la théorie de la complexité accorde une part de plus en plus importante aux classes de comptage [17, 6].

Si le problème SAT a jusqu'à présent motivé une masse considérable de travaux, le problème #SAT est aujourd'hui reconnu comme aussi important et fondamental. D'une part, ce problème apparaît dans de nombreux problèmes d'intelligence artificielle, que ce soit le calcul de la probabilité de pannes de circuits ou encore la formalisation du raisonnement approximatif à base de probabilités. D'autre part, il tient une place importante dans la théorie de la complexité puisqu'il est un représentant de la classe des problèmes #P-complets.

Plusieurs algorithmes de résolution exacte de #SAT ont été développés [4, 11, 14] mais ils se heurtent à l'explosion combinatoire due à la complexité du problème pour des tailles, même réduites, des instances SAT.

Cependant, une valeur approchée ou estimation du nombre exact de solutions aura un intérêt crucial. [19] montre que même l'approximation est algorithmiquement difficile. Différents axes de recherche ont déjà été étudiés : interruption d'une méthode exacte [4], calcul approché utilisant une résolution exacte [21, 22] et « randomisation » (évaluation du résultat global à partir de résultats partiels obtenus par tirage aléatoire dans l'ensemble des solutions possibles [5, 2]).

La méthode que nous présentons approxime le nombre exact de solutions en utilisant le principe général « diviser pour résoudre ». Il s'agit de diviser le problème en deux sous-problèmes de tailles respectives comparables et de pouvoir reconstituer – en temps polynomial – une solution approchée du problème global à partir des solutions exactes aux deux sous-problèmes.

Associant à une instance SAT, une matrice d'existence Clauses × Variables, la méthode proposée consiste, dans ses grandes lignes :

(*i*) à appliquer une technique spécifique de sériation ;

(*ii*) à couper la sériation « au mieux » en deux parties connexes de tailles comparables, à partir d'un critère statistique de coût polynomial et adapté à l'évaluation du degré d'indépendance ;

(*iii*) à reconstituer à partir d'une formule dûment justifiée par le calcul des probabilités et les conditions expérimentales, une valeur approximative du nombre total de solutions.

Dans nos premières communications [21, 13], nous avions le souci de pouvoir reconnaître la meilleure qualité des résultats qu'on pouvait atteindre. De sorte que connaissant le nombre exact de solutions, on déterminait la coupure qui fournissait la meilleure approximation. En revanche, ici, la coupure est déterminée – comme cela se doit dans les cas réels – automatiquement, à partir d'un critère statistique de coût polynomial. À des fins de validation de la méthode proposée, nous continuons certes à comparer le nombre exact de solutions à celui approché.

L'ensemble de nos expériences porte sur le cas, classiquement considéré dans la littérature de 3SAT aléatoire, c'est-à-dire sur la réalisation d'instances SAT dans une hypothèse d'indépendance probabiliste totale et de distribution uniforme sur l'espace des clauses, où exactement trois variables par clause se trouvent instanciées. C'est en effet dans ce contexte où aucune structure statistique n'est cachée, que la difficulté algorithmique apparaît.

2. #SAT ; REPRÉSENTATION, NOTION D'INDÉPENDANCE ET SOLUTION APPROCHÉE PAR COUPURE

2.1. Représentation

Nous allons rappeler la représentation classique du problème #SAT ainsi que celle ensembliste [10] qui, précisément, permet une approche par l'analyse combinatoire des données et la statistique.

Définitions préliminaires

Clause : $V = \{x_1, \ldots, x_i, \ldots, x_N\}$ étant un ensemble de variables booléennes, une clause sur V d'ordre r est une disjonction de littéraux de la forme

$$C^r = y_1 \vee y_2 \vee \ldots \vee y_q \vee \ldots \vee y_r \quad (r < N) \tag{1}$$

où $\{1, 2, \ldots, q, \ldots, r\}$ représente un sous-ensemble de r éléments de $\{1, 2, \ldots, i, \ldots, N\}$ et où y_q, pour q représentant un indice i, est de la forme x_i ou $\neg x_i$.

Ainsi par exemple, en supposant N supérieur à 4, une clause C^3 d'ordre 3 peut être :

$$C^3 = x_1 \vee \neg x_3 \vee x_4. \tag{2}$$

Dans ce cas $y_1 = x_1$, $y_2 = \neg x_3$ et $y_3 = x_4$.

Une affectation des variables booléennes satisfait la clause si et seulement si l'une au moins des variables $y_q, 1 \leq q \leq r$, est à vrai. Dans l'exemple précédent c'est le cas si et seulement si x_1 est à vrai ou (non exclusif) x_3 est à faux ou (non exclusif) x_4 est à vrai.

Cylindre ponctuel associé à une clause :

À l'ensemble V des N variables booléennes associons le cube logique $\{0, 1\}^N$. Il représente l'ensemble des valeurs du vecteur $(x_1, \ldots, x_i, \ldots, x_N)$ des variables booléennes. Le cylindre ponctuel associé à une clause représente tout simplement

l'ensemble des sommets du cube logique qui falsifient la clause. Cet ensemble a une structure géométrique particulière, ce qui justifie son appellation. Plus précisément, notons la clause C^r (cf. (1)) sous la forme:

$$C^r = y_{i_1} \vee y_{i_2} \vee \ldots \vee y_{i_q} \vee \ldots \vee y_{i_r} \tag{3}$$

ce qui indique les variables x_i instanciées dans C^r, soit sous forme positive soit (exclusivement) sous forme négative.

Associons maintenant à C^r sa négation, l'anti-clause $\neg C^r$, elle se met sous la forme de la conjonction:

$$\neg C^r = \neg y_{i_1} \wedge \neg y_{i_2} \wedge \ldots \wedge \neg y_{i_q} \wedge \ldots \wedge \neg y_{i_r} \tag{4}$$

où on a $\neg y_{i_q} = x_{i_q}$ (resp. $\neg x_{i_q}$) selon que $y_{i_q} = \neg x_{i_q}$ (resp. x_{i_q}).

L'ensemble des sommets du cube logique satisfaisant (4) se présente sous la forme d'un vecteur dont seules les composantes $i_1, i_2, \ldots, i_q, \ldots, i_r$ se trouvent instanciées, les autres composantes étant indéterminées. Plus précisément, en désignant par α_i la i-ème composante d'un tel vecteur, on a:

$$\begin{aligned} \alpha_{i_q} &= 1 \text{ (resp. } 0) \text{ si } y_{i_q} = \neg x_{i_q} \text{ (resp. } x_{i_q}), \ 1 \leq q \leq r, \\ \alpha_i &= \varepsilon \text{ si } i \notin \{i_1, i_2, \ldots, i_q, \ldots, i_r\} \end{aligned}$$

où ε est un booléen indéterminé.

Une telle structure correspond bien à un cylindre ponctuel puisque certaines composantes ne sont pas spécifiées et que la base du cylindre est un point dans le sous espace des composantes $(i_1, i_2, \ldots, i_q, \ldots, i_r)$.

Instance SAT:

Une instance SAT est une conjonction de clauses qu'on peut mettre sous la forme

$$F = C_1^{r_1} \wedge C_2^{r_2} \wedge \ldots \wedge C_i^{r_i} \wedge \ldots \wedge C_p^{r_p} \tag{5}$$

qui représente une forme normale conjonctive. $C_i^{r_i}, 1 \leq i \leq P$, a la même forme que (3) ci-dessus. En associant à chaque clause $C_i^{r_i}$ son cylindre ponctuel que nous noterons $E_i^{r_i}, 1 \leq i \leq P$, la négation de la formule $\neg F$ sera représentée par la réunion des cylindres ponctuels:

$$G = \bigcup_{1 \leq i \leq P} E_i^{r_i}. \tag{6}$$

Le problème SAT et le problème #SAT:

Le problème SAT de la SATisfiabilité est celui de la reconnaissance de l'existence d'au moins une solution ; c'est-à-dire, de l'existence d'une affectation des variables de V, qui mette l'instance F à vrai. Toute l'importance de ce problème a été exprimée dans l'introduction. Dans la pratique, la plupart des algorithmes de reconnaissance exhibent une solution effective au problème posé lorsqu'elle existe.

Étant donnée notre représentation du problème où à une clause on associe son cylindre ponctuel (cf. ci-dessus), la question posée est de savoir si la réunion G des cylindres ponctuels (cf. (6)) recouvre la totalité du cube logique $\{0,1\}^N$, ou bien s'il reste un vide (un trou). Dans ce dernier cas, l'instance est satisfiable.

Imaginons que toutes les clauses soient d'un même ordre r et appelons rSAT le problème SAT appliqué à de telles clauses. En d'autres termes et relativement à (5) :

$$r_1 = r_2 = \ldots = r_i = \ldots = r_p = r \tag{7}$$

Dans ces conditions, on sait qu'il existe un algorithme linéaire en temps pour 2SAT [1] mais que 3SAT est un représentant irréductible de la classe des problèmes NP-complets [3].

Considérons alors le problème 3SAT. Deux conditions expérimentales s'avèrent nécessaires pour que le cas de résolution soit difficile par rapport au nombre N de variables.

La première est qu'il n'existe aucune structure statistique cachée derrière l'ensemble des clauses ; ce qui se traduit par l'absence de toute relation statistique particulière sur l'ensemble des cylindres ponctuels dans le cube $\{0,1\}^N$. Par conséquent, les clauses devront être engendrées selon un modèle probabiliste d'indépendance mutuelle. D'autre part, la génération aléatoire d'une même clause se fera conformément à un modèle d'indépendance uniforme, relativement à l'ensemble des N variables booléennes. Plus précisément, en s'exprimant en termes de cylindres ponctuels, il y a $\binom{N}{3}2^3$ cylindres ponctuels dont la base est un sommet qui se situe dans un sous espace de dimension 3 du cube ; et, il s'agira d'en choisir un uniformément au hasard pour instancier une clause de 3SAT. Dans la pratique, on commencera par choisir les 3 variables à instancier (sous forme soit positive soit négative) ; puis, on choisira une instanciation parmi les 2^3 possibles. Ainsi, la probabilité d'une clause donnée est $1/\binom{N}{3}2^3$.

La deuxième condition expérimentale pour la réalisation aléatoire – conformément au modèle que nous venons de définir – d'une instance SAT difficile pour la reconnaissance de la satisfiabilité, consiste en le respect d'un rapport entre le nombre de clauses et celui des variables. Ce dernier doit être autour de 4.25 [16].

Le problème #SAT, dénombrement de toutes les solutions d'une instance SAT est à l'évidence et a priori plus difficile que le problème précédent, il se situe en effet dans la classe des problèmes #P-complets [25]. Nous avons déjà exprimé toute l'importance de cette classe de problèmes. Rappelons également que même pour des tailles réduites d'instances SAT, les algorithmes de résolution exacte qui ont été proposés [4, 10, 11, 14] se heurtent à l'explosion combinatoire de la complexité du problème.

Relativement à notre représentation des clauses en termes de cylindres ponctuels, le nombre de solutions d'une instance SAT représentée par F (cf. (5)) est exactement le complémentaire à 2^N du cardinal de G (cf. (6)). Il s'agit dans ces conditions d'évaluer la cardinalité de la réunion des cylindres ponctuels respectivement associés aux clauses.

Ici encore nous allons considérer des instances SAT dont les clauses ont le même ordre r (cf. (7)). Le problème #rSAT est celui de l'évaluation du nombre de solutions d'une instance rSAT. #2SAT rentre déjà dans la classe des problèmes #P-complets. Cependant, à des fins de comparaison avec les résultats obtenus dans la littérature, nous allons nous intéresser au problème #3SAT. Comme pour le problème SAT, le cas difficile est fourni par le modèle aléatoire de génération d'une instance défini ci-dessus. Ici, le pic de difficulté correspond à un rapport de 1.2 entre le nombre de clauses et celui des variables [20].

À défaut du nombre exact de solutions d'une instance SAT, une valeur suffisamment approchée ou estimation aura un intérêt crucial; à condition bien sûr d'une réduction notable de la complexité permettant de l'obtenir. Différentes approches ont déjà été étudiées: interruption d'une méthode exacte [4], inférence à partir de résultats partiels obtenus par sondage aléatoire dans l'ensemble des solutions possibles [5, 2].

La méthode que nous présentons donne une approximation du nombre exact de solutions en utilisant le principe général « diviser pour résoudre ». Il s'agit de diviser le problème en deux sous-problèmes de tailles respectives comparables et de pouvoir reconstituer – en temps polynomial – une solution approchée du problème global à partir des solutions exactes aux deux sous-problèmes.

Associant à une instance SAT, une matrice d'existence Clauses × Variables, la méthode proposée consiste, dans ses grandes lignes:

(i) à appliquer une technique spécifique de sériation;

(ii) à couper la sériation « au mieux » en deux parties connexes de tailles comparables, à partir d'un critère statistique de coût polynomial et adapté à l'évaluation du degré d'indépendance;

(iii) à reconstituer à partir d'une formule dûment justifiée par le calcul des probabilités et les conditions expérimentales, une valeur approximative du nombre total de solutions.

Ainsi, une notion fondamentale apparaît; c'est celle d'indépendance en probabilité entre deux ensembles de clauses disjoints (aucune clause de l'un des ensemble ne se retrouve dans l'autre). Nous l'appréhendons plus directement à partir des deux ensembles de cylindres ponctuels respectivement associés aux deux ensembles de clauses.

2.2. Indépendance probabiliste entre deux ensembles de clauses

Selon [23], deux clauses C et C' sont "logiquement indépendantes" si et seulement s'il n'y a pas une affectation des variables qui puisse les contredire simultanément. Désignons par $E(C)$ et $E(C')$ les cylindres ponctuels respectivement associés à C et à C'. Cette notion d'"indépendance logique" correspond alors exactement à la notion de disjonction entre les deux ensembles $E(C)$ et $E(C')$; ce qui est un cas d'exclusion probabiliste entre les deux événements représentés par les deux cylindres ponctuels [10]. Par conséquent, en termes de probabilités, c'est un cas de dépendance négative totale.

C'est la notion d'indépendance en probabilité que nous considérons et cherchons à exploiter. L'expérience aléatoire sous-jacente consiste en le choix, selon un modèle

uniforme, d'une affectation des variables. Ainsi, chaque sommet du cube logique $\{0,1\}^N$ a la probabilité $1/2^N$ d'apparaître. Les événements de base qui nous intéressent sont les cylindres ponctuels. Un événement général se trouve défini par une réunion finie de tels événements de base. Il en résulte une algèbre des événements puisque l'intersection de deux cylindres ponctuels est un cylindre ponctuel [10]. Ainsi, l'indépendance en probabilité entre deux ensembles de clauses $\mathcal{C}$ et $\mathcal{C}'$ est définie par l'indépendance en probabilité entre les deux événements $\mathcal{E}$ et $\mathcal{E}'$ respectivement associés. Précisément, $\mathcal{E}$ (resp. $\mathcal{E}'$) est formé de la réunion des cylindres ponctuels respectivement associés aux clauses de $\mathcal{C}$ (resp. $\mathcal{C}'$).

Soient C et C' deux clauses et soient $W(C)$ et $W(C')$ les deux ensembles de variables respectivement instanciées dans C et dans C', on a dans ces conditions la propriété suivante :

LEMME. Les deux clauses C et C' sont indépendantes en probabilité si et seulement si les deux ensembles de variables $W(C)$ et $W(C')$ sont disjoints.

DÉMONSTRATION. — Désignons par

$$r = card[W(C)], r' = card[W(C')] \text{ et } s = card[W(C) \cap W(C')]. \tag{8}$$

Désignons par $E(C)$ et $E(C')$ les deux cylindres ponctuels respectivement associés aux clauses C et C'. On a :

$$\mathbb{P}[E(C)] = 2^{-r} \quad \text{et} \quad \mathbb{P}[E(C')] = 2^{-r'}. \tag{9}$$

$E(C) \cap E(C')$ est un cylindre ponctuel dont le volume (nombre de sommets qu'il contient) est nul dès lors qu'il existe au moins une variable appartenant à $W(C)$ et à $W(C')$, instanciée de façon opposée entre C et C'. On a dans ce cas

$$\mathbb{P}[E(C) \cap E(C')] = 0 \tag{10}$$

Dans le cas contraire, $E(C) \cap E(C')$ est un cylindre ponctuel dont le nombre de variables instanciées est $r + r' - s$ et donc, dont le volume est $2^{N-r-r'+s}$. Dans ces conditions, on a :

$$\mathbb{P}[E(C) \cap E(C')] = 2^{-r-r'+s}. \tag{11}$$

Ainsi, le seul cas où la relation suivante d'indépendance

$$\mathbb{P}[E(C) \cap E(C')] = \mathbb{P}[E(C)] \times \mathbb{P}[E(C')] \tag{12}$$

se trouve vérifiée est celui où s = 0. ■

THÉORÈME 1. Deux ensembles de clauses $\mathcal{C}$ et $\mathcal{C}'$ sont indépendants si quelles que soient les clauses C et C', appartenant respectivement à $\mathcal{C}$ et à $\mathcal{C}'$ ($C \in \mathcal{C}$ et $C' \in \mathcal{C}'$), C et C' sont indépendantes.

DÉMONSTRATION. — Nous allons établir par récurrence cette condition suffisante de l'indépendance entre deux ensembles de clauses $\mathcal{C}$ et $\mathcal{C}'$. Désignons par $\{E_1, \ldots, E_c\}$ l'ensemble des cylindres ponctuels associés aux clauses de $\mathcal{C}$ et par $\{E_{c+1}, \ldots, E_l\}$ l'ensemble de ceux associés aux clauses de $\mathcal{C}'$.

La récurrence va porter sur le nombre total l, en supposant $c \geq 1$ et $l > c$.

La propriété est vraie par définition pour $l = 2$.

Supposons qu'elle soit vraie pour $l \leq M$ et démontrons la pour $l = M + 1$.

Soient dans ces conditions $\{E_1, \ldots, E_c\}$ et $\{E_{c+1}, \ldots, E_M, E_{M+1}\}$ les deux ensembles de cylindres ponctuels respectivement associés aux deux ensembles de clauses $\mathcal{C}$ et $\mathcal{C}'$. Soient

$$A = \bigcup_{1 \leq i \leq c} E_i \quad \text{et} \quad B = \bigcup_{c+1 \leq i \leq M} Ei, \tag{13}$$

notons d'autre part $E = E_{M+1}$. Sachant que E_i et $E_{i'}$ sont indépendants pour $1 \leq i \leq c$ et $c+1 \leq i' \leq M+1$, il s'agit d'établir – compte tenu de l'hypothèse de récurrence – que

$$\mathbb{P}[A \cap (B \cup E)] = \mathbb{P}[A] \times \mathbb{P}[B \cup E]. \tag{14}$$

Or le premier membre se met sous la forme

$$\mathbb{P}[(A \cap B) \cup (A \cap E)] = \mathbb{P}[A \cap B] + \mathbb{P}[A \cap E] - \mathbb{P}[A \cap B \cap E] \tag{15}$$

Compte tenu de l'hypothèse de récurrence

$$\mathbb{P}[A \cap B] = \mathbb{P}[A] \times \mathbb{P}[B], \tag{16}$$

et

$$\mathbb{P}[A \cap E] = \mathbb{P}[A] \times \mathbb{P}[E]. \tag{17}$$

Maintenant $B \cap E$ peut se mettre sous la forme

$$\bigcup_{c+1 \leq i' \leq M} E_{i'} \cap E_{M+1}. \tag{18}$$

Considérons un couple d'indices quelconque (i, i') tel que $1 \leq i \leq c$ et $c+1 \leq i' \leq M$.

Si $E_{i'} \cap E_{M+1} = \emptyset$, E_i et $E_{i'} \cap E_{M+1}$ sont indépendants puisque

$$\mathbb{P}[E_i \cap (E_{i'} \cap E_{M+1})] = \mathbb{P}[E_i] \times \mathbb{P}[E_{i'} \cap E_{M+1}] = 0. \tag{19}$$

Sinon, les indépendances entre E_i et $E_{i'}$ d'une part et entre E_i et E_{M+1} d'autre part, conduisent à l'indépendance entre E_i et le cylindre ponctuel $E_{i'} \cap E_{M+1}$ puisque, d'après le lemme, l'ensemble des variables instanciées dans E_i est disjoint de chacun des deux ensembles de variables instanciées dans $E_{i'}$ et dans E_{M+1}.

On a donc

$$\mathbb{P}[A \cap B \cap E] = \mathbb{P}[A \cap B'] \tag{20}$$

où B' se met sous la forme :

$$B' = \bigcup_{1 \leq i' \leq M} (E_{i'} \cap E) \tag{21}$$

de sorte que l'hypothèse de récurrence s'applique pour A et B' ; et on a

$$\mathbb{P}[A \cap B'] = \mathbb{P}[A] \times \mathbb{P}[B'] \tag{22}$$

D'autre part, vu que B' se met sous la forme $B \cap E$ et compte tenu des relations (16), (17) et (22), le second membre de (15) se développe comme suit :

$$\mathbb{P}[A] \times \mathbb{P}[B] + \mathbb{P}[A] \times \mathbb{P}[E] - \mathbb{P}[A] \times \mathbb{P}[B \cap E] = \mathbb{P}[A] \times \mathbb{P}[B \cup E]. \quad (23)$$

■

Il est important de noter que la condition définie par la théorème 1 est suffisante mais non nécessaire. Nous allons en effet considérer un contre-exemple où E_1, E_2 et E_3 étant trois cylindres ponctuels, on a

$$\mathbb{P}[(E_1 \cup E_2) \cap E_3] = \mathbb{P}[E_1 \cup E_2] \times \mathbb{P}[E_3] \quad (24)$$

sans que l'on ait les indépendances mutuelles entre E_1 et E_3 d'une part et, E_2 et E_3 d'autre part.

Le développement du premier membre de (24) donne

$$\mathbb{P}[(E_1 \cap E_3) \cup (E_2 \cap E_3)] = \mathbb{P}[E_1 \cap E_3] + \mathbb{P}[E_2 \cap E_3] - \mathbb{P}[E_1 \cap E_2 \cap E_3]. \quad (25)$$

Le développement du second membre de (24) donne

$$(\mathbb{P}[E_1]+\mathbb{P}[E_2]-\mathbb{P}[E_1 \cap E_2]) \times \mathbb{P}[E_3] = \mathbb{P}[E_1]\mathbb{P}[E_3]+\mathbb{P}[E_2]\mathbb{P}[E_3]-\mathbb{P}[E_1 \cap E_2]\mathbb{P}[E_3]. \quad (26)$$

On considère maintenant pour N = 8

$$E_1 = (1, 1, 1, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon)$$

$$E_2 = (0, \varepsilon, \varepsilon, \varepsilon, 0, 1, \varepsilon, \varepsilon)$$

$$E_3 = (\varepsilon, \varepsilon, 1, 1, 1, \varepsilon, \varepsilon, \varepsilon)$$

On constate que ni E_1 ni E_2 ne sont indépendants de E_3. Alors que le premier membre de (24) donne à travers (25),

$$2^{-5} + 0 - 0 = 2^{-5}$$

et que, le second membre de (24) donne à travers (26),

$$2^{-3} \times 2^{-3} + 2^{-3} \times 2^{-3} - 0 = 2^{-5}$$

ce qui établit le contre-exemple.

Maintenant, relativement à un ensemble $\mathcal{C}$ de clauses, désignons par $W(\mathcal{C})$ l'ensemble des variables dont chacune se trouve instanciée au moins une fois dans l'une des clauses de $\mathcal{C}$.

THÉORÈME 2. $\mathcal{C}$ et $\mathcal{C}'$ étant deux ensembles de clauses, une condition nécessaire et suffisante pour que chacune des clauses de $\mathcal{C}$ soit indépendante de chacune des clauses de $\mathcal{C}'$ est que les ensembles de variables $W(\mathcal{C})$ et $W(\mathcal{C}')$ soient disjoints.

DÉMONSTRATION. — Désignons par $C_1, C_2, \ldots, C_c$ les clauses de $\mathcal{C}$ et par C_{c+1}, $C_{c+2}, \ldots, C_P$ les clauses de $\mathcal{C}'$. On notera alors $W_i = W(C_i), 1 \leq i \leq P$. On a dans ces conditions :

$$W(\mathcal{C}) = \bigcup_{1 \leq i \leq c} W_i \quad \text{et} \quad W(\mathcal{C}') = \bigcup_{c+1 \leq i \leq P} W_i \tag{27}$$

La condition est trivialement suffisante. Nous allons montrer qu'elle est nécessaire, c'est-à-dire que si pour tout (i, i') appartenant à $\{1, 2, \ldots, c\} \times \{c+1, c+2, ..., P\}$, C_i et $C_{i'}$ sont indépendants, alors $W(\mathcal{C})$ et $W(\mathcal{C}')$ sont disjoints.

D'après le lemme 1, l'indépendance entre C_i et $C_{i'}$ est équivalente à la relation

$$W_i \cap W_{i'} = \emptyset. \tag{28}$$

Par conséquent, l'indépendance mutuelle entre chaque C_i de $\mathcal{C}$ et chaque $C_{i'}$ de $\mathcal{C}'$, conduit à la relation

$$\bigcup_{\substack{1 \leq i \leq c \\ c+1 \leq i' \leq P}} W_i \cap W_{i'} = \emptyset. \tag{29}$$

Or, compte tenu de (27), le premier membre de (29) peut se mettre sous la forme

$$W(\mathcal{C}) \cap W(\mathcal{C}') \tag{30}$$

■

Le théorème fournit de façon limite la justification formelle de la nécessité d'une structure sériationnelle pour la décomposition en 2 de l'ensemble des clauses. D'autre part et de façon liée, il justifie l'adoption d'un critère de coupure mesurant l'interdépendance entre deux ensembles de clauses à partir des comparaisons deux à deux.

2.3. Solution approchée par coupure ; critères de coupure

Comme nous l'avons déjà exprimé, le principe général de notre méthode est celui bien connu de « diviser pour résoudre ». Dans notre cas, c'est le critère d'indépendance en probabilité qui doit guider la division.

Soit $\mathcal{C} = \{Ci | 1 \leq i \leq P\}$ un ensemble de clauses. Imaginons que $\mathcal{C}$ puisse se décomposer en deux sous-ensembles

$$\mathcal{A}_c = \{C_i | 1 \leq i \leq c\} \text{ et } \mathcal{B}_c = \{C_i | c+1 \leq i \leq P\} \tag{31}$$

tels que $\mathcal{A}_c$ et $\mathcal{B}_c$ soient deux ensembles de clauses indépendants. On peut désigner par

$$\{E_i | 1 \leq i \leq c\} \text{ et } \{E_i | c+1 \leq i \leq P\} \tag{32}$$

les deux ensembles de cylindres ponctuels respectivement associés à $\mathcal{A}_c$ et $\mathcal{B}_c$. On notera enfin

$$A_c = \bigcup_{1 \leq i \leq c} E_i \quad \text{et} \quad B_c = \bigcup_{c+1 \leq i \leq P} E_i. \tag{33}$$

Compte tenu de la formule

$$\mathbb{P}[A_c \cup B_c] = \mathbb{P}[A_c] + \mathbb{P}[B_c] - \mathbb{P}[A_c \cap B_c], \tag{34}$$

la relation d'indépendance entre A_c et B_c conduit à l'équation :

$$\mathbb{P}[A_c \cup B_c] = \mathbb{P}[A_c] + \mathbb{P}[B_c] - \mathbb{P}[A_c] \times \mathbb{P}[B_c]. \tag{35}$$

Soit maintenant la formule telle que (5) résultant de la conjonction des clauses de $\mathcal{C}$. Notons $\text{NBS}(F)$ le nombre exact de solutions de l'instance F. Si $\mathbb{P}[F]$ est la probabilité d'insatisfiabilité de F, on a

$$\text{NBS}(F) = (1 - \mathbb{P}[F]) \times 2^N, \tag{36}$$

ou, de façon réciproque,

$$\mathbb{P}[F] = 1 - \text{NBS}(F) \times 2^{-N}. \tag{37}$$

En remplaçant dans (35), on obtient

$$\text{NBS}(F) = \text{NBS}(\mathcal{A}_c \cup \mathcal{B}_c) = \text{NBS}(\mathcal{A}_c) \times \text{NBS}(\mathcal{B}_c) \times 2^{-N}. \tag{38}$$

Ainsi, l'indépendance en probabilité entre $\mathcal{A}_c$ et $\mathcal{B}_c$ permet de réduire l'évaluation de $\text{NBS}(\mathcal{A}_c \cup \mathcal{B}_c)$ à celle de $\text{NBS}(\mathcal{A}_c)$ d'une part et de $\text{NBS}(\mathcal{B}_c)$ d'autre part. Ces deux dernières évaluations peuvent d'ailleurs être effectuées en parallèle. Cette réduction du problème est d'autant plus intéressante que les nombres de clauses de $\mathcal{A}_c$ et de $\mathcal{B}_c$ sont proches. Ainsi, si $card(\mathcal{A}_c) = card(\mathcal{B}_c)$, il y a une division par 2 de la taille du problème.

Pour les instances SAT aléatoires, notamment 3SAT (cf. §2.1), il y a une probabilité extrêmement faible de pouvoir dégager deux ensembles de clauses de cardinalités comparables et strictement indépendants. La répartition aléatoire des variables instanciées dans les clauses est telle que la décomposition de l'ensemble $\mathcal{C}$ en deux sous-ensembles disjoints $\mathcal{A}_c$ et $\mathcal{B}_c$ obéissant conjointement aux deux conditions :

(*i*) $\mathcal{A}_c$ et $\mathcal{B}_c$ indépendants,

(*ii*) $\mathcal{A}_c$ et $\mathcal{B}_c$ de même taille,

ne peut se faire que de façon approchée. De telle sorte que le second membre de la formule (38) fournit une approximation de $\text{NBS}(F)$, qu'il s'agit de rendre la plus précise possible.

C'est au paragraphe suivant que nous indiquerons le principe de l'algorithme de résolution. Sa mise en œuvre nécessite de façon fondamentale la définition d'un critère statistique mesurant le degré de dépendance entre deux ensembles disjoints de clauses $\mathcal{A}_c$ et $\mathcal{B}_c$.

Reprenons les ensembles A_c et B_c [cf. (33)] respectivement associés à $\mathcal{A}_c$ et $\mathcal{B}_c$. Une mesure du degré de dépendance est définie par l'éloignement à 1 de la densité de probabilité :

$$\frac{\mathbb{P}[A_c \cap B_c]}{\mathbb{P}[A_c] \times \mathbb{P}[B_c]} \tag{39}$$

Le calcul des probabilités intervenant dans l'expression (39) est exponentiel par rapport aux tailles respectives des arguments. Ainsi, par exemple, $A_c \cap B_c$ représente l'union de $c \times (P - c)$ cylindres ponctuels. Pour éviter la complexité de ce calcul, on tient compte du théorème 1 ci-dessus pour proposer la valeur approchée fournie par l'expression suivante :

$$\mathrm{d}^o\mathrm{dep}(A_c, B_c) = \frac{\sum_{a \in A_c} \sum_{b \in B_c} \mathbb{P}[a \cap b]}{[\sum_{a \in A_c} \mathbb{P}[a]] \times [\sum_{b \in B_c} \mathbb{P}[b]]} \tag{40}$$

On peut remarquer que $\mathrm{d}^o\mathrm{dep}(A_c, B_c)$ vaut également 1 en cas d'indépendance entre A_c et B_c.

Un autre coefficient mesurant la dépendance entre les deux ensembles A_c et B_c résulte de la généralisation que nous avons effectuée [9] du coefficient de Karl Pearson [18]. Ce coefficient d'association permet de mesurer l'intensité du lien entre deux attributs booléens définis sur un ensemble $\mathcal{O}$ d'objets. Il a une nature analogue à celle d'un coefficient de corrélation et est compris entre -1 et +1 ; la valeur 0 correspondant à l'indépendance. Nous allons rappeler rapidement une manière constructive permettant de l'obtenir [10].

Un attribut booléen a sur $\mathcal{O}$ est représenté par le sous-ensemble $\mathcal{O}_a$ des objets où il est à vrai. $\mathcal{O}_a$ peut être codé par sa fonction indicatrice Φ_a où $\Phi_a(x)$ est égal à 1 ou 0, selon que l'attribut a est à vrai sur x ou non. Relativement à une paire $\{a, b\}$ d'attributs booléens sur $\mathcal{O}$, on introduit l'indice brut de proximité

$$n(a \wedge b) = card(\mathcal{O}_a \cap \mathcal{O}_b) = \sum_{x \in \mathcal{O}} \Phi_a(x)\Phi_b(x). \tag{41}$$

Au couple $(\mathcal{O}_a, \mathcal{O}_b)$, on fait correspondre un couple $(\mathcal{O}_{a^*}, \mathcal{O}_{b^*})$ de parties aléatoires indépendantes telles que $\mathcal{O}_{a^*}$ (resp. $\mathcal{O}_{b^*}$) est un élément pris uniformément au hasard dans l'ensemble $\mathcal{P}(n_a, \mathcal{O})$ (resp. $\mathcal{P}(n_b, \mathcal{O})$) des parties de $\mathcal{O}$ de mêmes cardinalité n_a (resp. n_b) que $\mathcal{O}_a$ (resp. $\mathcal{O}_b$). $n(a^* \wedge b^*)$ est alors une variable hypergéométrique dont l'espérance mathématique $\mathcal{E}[n(a^* \wedge b^*)]$ et la variance $var[n(a^* \wedge b^*)]$ permettent de centrer et de réduire $n(a \wedge b)$ pour obtenir comme coefficient :

$$Q(a, b) = \frac{n(a \wedge b) - \mathcal{E}[n(a^* \wedge b^*)]}{\sqrt{var[n(a^* \wedge b^*)]}}. \tag{42}$$

En remplaçant les cardinaux par les proportions relativement à la taille n de $\mathcal{O}$, on obtient l'expression suivante :

$$Q(a, b) = \sqrt{n-1} \times \frac{p(a \wedge b) - p(a)p(b)}{\sqrt{p(a)p(\bar{a})p(b)p(\bar{b})}} \tag{43}$$

où $\bar{a}$ (resp. $\bar{b}$) représente l'attribut négation de a, $\neg a$ (resp. négation de b, $\neg b$).

Le coefficient de Karl Pearson est défini par

$$R(a, b) = \frac{p(a \wedge b) - p(a)p(b)}{\sqrt{p(a)p(\bar{a})p(b)p(\bar{b})}}. \tag{44}$$

Considérons à présent sur l'ensemble $\mathcal{O}$ des objets, deux ensembles disjoints A et B, formés chacun d'attributs booléens. On suppose qu'il n'y a aucun lien logique particulier entre deux attributs quelconques pris dans $A \cup B$. Le problème est la généralisation du coefficient $R(a,b)$ (cf. (44)) à la comparaison entre A et B.

Maintenant, relativement à un ensemble A d'attributs booléens sur $\mathcal{O}$, mutuellement sans lien logique, on introduit sur $\mathcal{O}$, la fonction d'appartenance

$$(\forall x \in \mathcal{O}), \Phi_A(x) = \frac{1}{c(A)} \sum_{a \in A} \Phi_a(x) \tag{45}$$

où $c(A)$ représente le cardinal de A. Ainsi, $\Phi_A(x)$ est la proportion d'attributs de A qui sont à vrai chez x. En désignant par $\bar{A}$ l'ensemble des attributs respectivement opposés aux attributs de A:

$$\bar{A} = \{\bar{a} = \neg a | a \in A\}, \tag{46}$$

on a la relation logique fondamentale:

$$(\forall x \in \mathcal{O}), \Phi_A(x) + \Phi_{\bar{A}}(x) = 1 \tag{47}$$

Dans ces conditions, l'indice brut d'association entre les deux ensembles d'attributs A et B ci-dessus représentés, se met sous la forme:

$$n(A \wedge B) = \sum_{x \in \mathcal{O}} \Phi_A(x) \Phi_B(x) \tag{48}$$

(voir (41)).

L'analyse d'une démarche étendant celle de la comparaison entre deux attributs booléens conduit à un indice brut aléatoire $n(A^* \wedge B^*)$ dont l'espérance mathématique et la variance sont respectivement

$$\mathcal{E}[n(A^* \wedge B^*)] = n\mu(\Phi_A)\mu(\Phi_B),$$

$$var[n(A^* \wedge B^*)] = \frac{n^2}{n-1} var(\Phi_A) var(\Phi_B), \tag{49}$$

où

$$\mu(\Phi_A) = \frac{1}{n} \sum_{x \in \mathcal{O}} \Phi_A(x) = \frac{1}{c(A)} \sum_{a \in A} p_a,$$

$$\mu(\Phi_B) = \frac{1}{n} \sum_{x \in \mathcal{O}} \Phi_B(x) = \frac{1}{c(B)} \sum_{a \in B} p_b,$$

$$var(\Phi_A) = \frac{1}{c(A)^2} \sum_{(a,a') \in A \times A} (p_{a \wedge a'} - p_a p_{a'}),$$

et

$$var(\Phi_B) = \frac{1}{c(B)^2} \sum_{(b,b') \in B \times B} (p_{b \wedge b'} - p_b p_{b'}) \tag{50}$$

où des expressions telles que p_a et $p_{a\wedge a'}$ indiquent des proportions de la forme n_a/n et $n(a \wedge a')/n$.

L'extension du coefficient $R(a,b)$ (cf. (44)) devient dans ces conditions:

$$R(A,B) = \frac{p(A \wedge B) - \mu(\Phi_A)\mu(\Phi_B)}{\sqrt{var(\Phi_A)var(\Phi_B)}}, \tag{51}$$

où nous avons noté $p(A \wedge B)$ pour $n(A \wedge B)/n$.

$R(A,B)$ va donner lieu à une nouvelle mesure du degré d'indépendance mutuelle entre deux ensembles disjoints de clauses $\mathcal{A}_c$ et $\mathcal{B}_c$. À cette fin et par rapport au précédent contexte, on assimilera $\mathcal{O}$ au cube logique $\{0,1\}^N$, de sorte qu'un cylindre ponctuel sera la représentation d'un attribut booléen. Ainsi, il suffira de remplacer dans l'expression (51) A et B par A_c et B_c (cf. (33)) pour obtenir une mesure $R(A_c, B_c)$ du lien entre les deux ensembles de clauses. A_c et B_c seront d'autant plus indépendants que la valeur absolue $|R(A_c, B_c)|$ est voisine de zéro. On remarquera que par rapport à la première mesure adoptée (40), le coefficient (51) fait intervenir dans son dénominateur des proportions telles que $p_{a\wedge a'}$ et $p_{b\wedge b'}$, pour $(a,a') \in A \times A$ et $(b,b') \in B \times B$, qui seront calculées conformément aux équations (10) et (11) ci-dessus.

Le degré de dépendance (40) et le coefficient d'association (51) déterminent des critères cruciaux pour partitionner l'ensemble des clauses en deux classes de tailles comparables et aussi indépendantes que possible. Cette décomposition nécessite l'organisation de l'ensemble des clauses selon une structure statistique. Celle qui s'est avérée la plus adaptée est une sériation. Nous allons chercher au paragraphe suivant à en expliquer la raison.

3. DIVISER POUR RÉSOUDRE; COUPER UNE SÉRIATION ASSOCIÉE À SAT

3.1. Structure recherchée du tableau d'incidence associé à SAT

Soit une instance SAT dont on désigne — comme précédemment — par $\{x_1, \ldots, x_j, \ldots, x_N\}$ l'ensemble des variables et par $\{C_1, \ldots, C_i, \ldots, C_P\}$ l'ensemble des clauses.

Le tableau d'incidence associé à une instance SAT est défini par:

$$a_{ij} = \begin{cases} 0 \text{ si } x_j \text{ et } \neg x_j \text{ n'apparaissent pas dans la } i^{\text{e}} \text{ clause } C_i \\ 1 \text{ si } x_j \text{ ou } \neg x_j \text{ apparait dans la } i^{\text{e}} \text{ clause } C_i \end{cases} \text{ pour } \begin{matrix} 1 \leq i \leq P, \\ 1 \leq j \leq N. \end{matrix}$$

Compte tenu du théorème 2 ci-dessus, la structure idéale qu'il s'agit de mettre en évidence par permutation des lignes et des colonnes du tableau d'incidence est celle de bloc sériation [8, 7, 15], mais avec exactement deux blocs. De plus, il importe pour que cette mise en forme soit efficace, que les deux blocs aient approximativement le même nombre de lignes (voir condition (*ii*) du §2.3). Il est clair que cette forme pure est inaccessible dans les cas réels et surtout – comme nous l'avons fait remarquer (cf. §2.3) – pour SAT aléatoire. Il s'agit néanmoins de se rapprocher au mieux de cette forme pour pratiquer avec le plus de précision possible, la formule d'approximation (35).

On aurait certes pu songer à des méthodes relevant de l'optimisation combinatoire, cependant nos premières expériences ont consisté à pratiquer un couple de classification hiérarchiques duales, sur respectivement, l'ensemble des lignes et l'ensemble des colonnes du tableau d'incidence associé à une instance SAT aléatoire (programme CHAVL [12]). On cherche alors à reconnaître à partir de la première classification hiérarchique une partition en deux classes de tailles aussi comparables que possible. On considère également une partition en deux classes de l'ensemble des colonnes, guidée par leur classification hiérarchique. Si la structure de bloc sériation en deux classes domine statistiquement, la permutation des lignes et des colonnes du tableau d'incidence conformément à leurs classifications hiérarchiques respectives, aboutissant chacune en une partition en deux classes, devrait la faire apparaître. En fait, et c'est lié à la nature de la donnée qui est une instance 3SAT aléatoire, la forme apparue correspondait à une sériation continue.

Par conséquent, une approche qui s'avère directe, simple et naturelle consiste d'abord à opérer une sériation globale et ensuite, à chercher à la segmenter en deux parties de tailles comparables, les plus indépendantes possibles, dans les termes des ensembles de clauses $\mathcal{A}_c$ et $\mathcal{B}_c$ qu'elles définissent (cf. §2.3).

On aurait certes pu envisager un algorithme de classification sous contraintes, où d'une part, le nombre de classes est deux et où d'autre part, les tailles des classes sont égales. Pratiquer alors un couple de sériations sur respectivement les deux classes et les juxtaposer, aurait très vraisemblablement conduit à un résultat équivalent à une sériation globale. D'autre part, de la sorte, on ne contrôle plus le degré d'indépendance des deux classes extraites et, il est plus judicieux de pouvoir ajuster les tailles – restant comparables – des deux classes, afin de diminuer au mieux leur degré de dépendance.

On comprend dans ces conditions la pertinence et la souplesse de notre approche qui consiste à déterminer une sériation globale, schématisée par la figure 1, et à la couper au mieux.

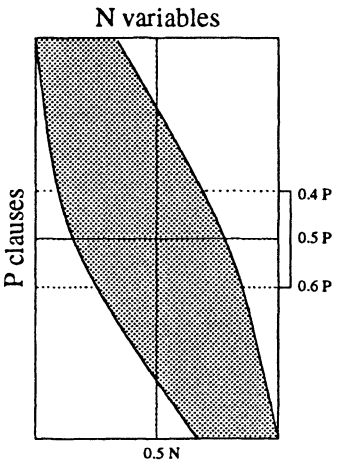

FIG. 1 – *Matrice d'existence : la partie en blanc ne contient que des 0*

Précisément, considérons une instance SAT après application d'un algorithme de sériation, ordonnant respectivement les ensembles de clauses et de variables.

On désigne par $\{C_1, \ldots, C_i, \ldots, C_P\}$ et $\{x_1, \ldots, x_j, \ldots, x_N\}$ ces deux ensembles ainsi ordonnés. Il s'agit alors de déterminer l'indice de coupure optimal c, afin d'obtenir les deux sous-ensembles $\mathcal{A}_c = \{C_1, \ldots, C_c\}$ et $\mathcal{B}_c = \{C_{c+1}, \ldots, C_P\}$ les plus indépendants possibles.

L'objectif principal étant de diminuer la complexité de la résolution, nous optons pour restreindre le choix de c à l'intervalle $[0.4P, 0.6P]$ afin d'équilibrer les deux sous-ensembles de clauses. L'amplitude de choix est alors suffisante et la complexité nettement diminuée.

Nous avons défini deux critères statistiques de mesure du degré d'indépendance entre deux ensembles de clauses $\mathcal{A}_c$ et $\mathcal{B}_c$, que nous avons respectivement notés $\mathrm{d}^o\mathrm{dep}(A_c, B_c)$ (cf. (40)) et $R(A_c, B_c)$ (cf. (51)). Le calcul de ces indices est polynomial par rapport à c et à $(P - c)$. Très précisément, le calcul de $\mathrm{d}^o\mathrm{dep}(A_c, B_c)$ est linéaire par rapport à $c \times (P - c)$; alors que celui de $R(A_c, B_c)$ est linéaire par rapport à $max[c(P - c), c^2/2, (P - c)^2/2]$.

Pour le premier critère, c est déterminé par

$$\mathrm{argmin}\{|\log(\mathrm{d}^o\mathrm{dep}(A_c, B_c))| / \ 0.4P \leq c \leq 0.6P\} \tag{52}$$

et pour le second critère, c est déterminé par

$$\mathrm{argmin}\{|R(A_c, B_c)| / \ 0.4P \leq c \leq 0.6P\} \tag{53}$$

3.2. Un algorithme de sériation

Le problème de la sériation a tout d'abord été vu comme le problème d'ordonnancement chronologique d'objets à partir de leurs caractéristiques : un archéologue veut pouvoir dater une tombe en fonction des objets qu'elle contient et réciproquement. Ce problème a été étudié dans une grande variété de contextes et selon différents points de vue [8, 7, 15]. Ici, nous considérons le problème théorique et nous privilégions l'aspect visuel du tableau d'incidence [8] en proposant une technique nouvelle.

Considérons une instance SAT et le tableau d'incidence associé tels que définis au §3.1. Afin d'approcher l'indépendance, nous cherchons à ordonner les ensembles de variables et de clauses pour obtenir un aspect visuel proche de la figure 1.

La sériation du tableau d'incidence se fait par permutation des lignes et des colonnes. L'objectif étant de faire apparaître deux zones diagonales disjointes chargées de 1, on considère qu'une ligne (respectivement une colonne) est bien placée si sa contribution à la dispersion horizontale (respectivement verticale) est minimale.

DÉFINITION 1. La dispersion horizontale d'une instance SAT est donnée par :

$$\sum_{i=1}^{P} \frac{1}{\sum_{j=1}^{N} |a_{ij}|} \sum_{j=1}^{N} (j \times |a_{ij}| - b_i)^2$$

où b_i est la position (abscisse) de la i^{e} clause C_i sur la diagonale du tableau d'incidence associé à l'instance.

DÉFINITION 2. La dispersion verticale d'une instance SAT est donnée par:

$$\sum_{j=1}^{N} \frac{1}{\sum_{i=1}^{P} |a_{ij}|} \sum_{i=1}^{P} (i \times |a_{ij}| - b_j)^2$$

où b_j est la position (ordonnée) de la j^{e} variable x_j sur la diagonale du tableau d'incidence associé à l'instance.

Chaque permutation tend à diminuer la part de la dispersion lui revenant: la permutation des lignes a pour but de diminuer la dispersion horizontale, la permutation des colonnes la dispersion verticale. Ainsi, toute permutation peut aussi introduire une dispersion supplémentaire: on ne peut donc pas parler de « convergence » de la méthode. La méthode détermine un optimum après un certain nombre d'itérations. Cet optimum correspond à une expression visuelle de l'instance très proche de celle recherchée (FIG. 1).

4. RÉSULTATS EXPÉRIMENTAUX

Pour valider les indices de coupure que nous proposons, nous avons procédé à de nombreuses expérimentations sur différentes classes d'instances 3SAT aléatoires pour lesquelles nous pouvons calculer le nombre exact de solutions en un temps raisonnable. Le modèle de génération aléatoire d'une instance 3SAT ne comprend aucune structure statistique cachée: il y a équiprobabilité et indépendance. Nous nous plaçons donc dans le cas le plus défavorable pour la sériation.

Pour chaque instance SAT, le nombre approché de solutions s'obtient par le processus suivant:

1. `sériation de l'instance` par la méthode présentée
2. `détermination de la meilleure coupure` par calcul du degré de dépendance (40) (resp. du coefficient d'association (51)) et application de (52) (resp. (53))
3. `calcul du nombre exact de solutions` pour les deux sous-instances obtenues par un algorithme du type Davis & Putnam
4. `calcul du nombre approché de solutions` de l'instance de départ par la formule (38)

Les deux critères – degré de dépendance et coefficient d'association – donnent, sur les multiples expérimentations effectuées, des résultats sensiblement identiques et comme, de plus, le calcul du coefficient d'association est légèrement plus couteux, le degré de dépendance est le critère que nous avons retenu.

Les figures 3, 5, 8, et 10 présentent les différents résultats obtenus avec la coupure déterminée par l'indice retenu, alors que les figures 2, 4, 7, et 9 présentent les meilleurs résultats connaissant le nombre exact de solutions.

Chaque point correspond à une instance aléatoire de la classe d'instances considérée: nous mettons en correspondance le nombre exact de solutions calculé directement par l'algorithme de type Davis & Putnam et le nombre approché de solutions de cette instance calculé par notre méthode.

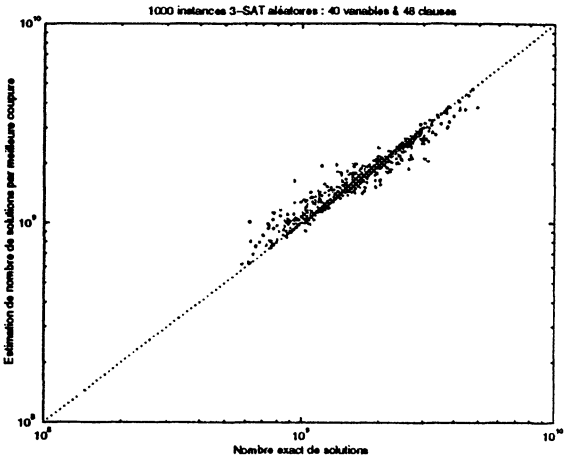

FIG. 2 – *40 variables et 48 clauses*

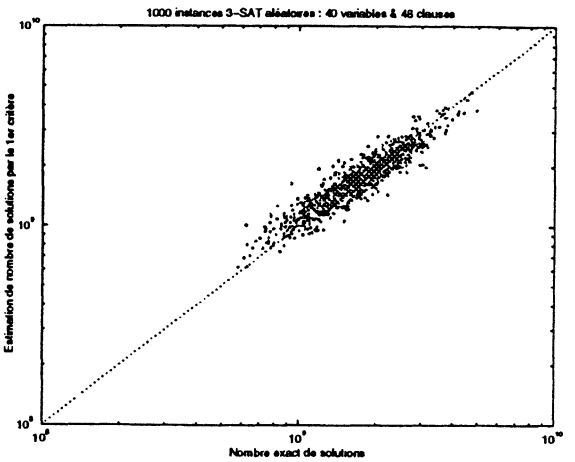

FIG. 3 – *40 variables et 48 clauses*

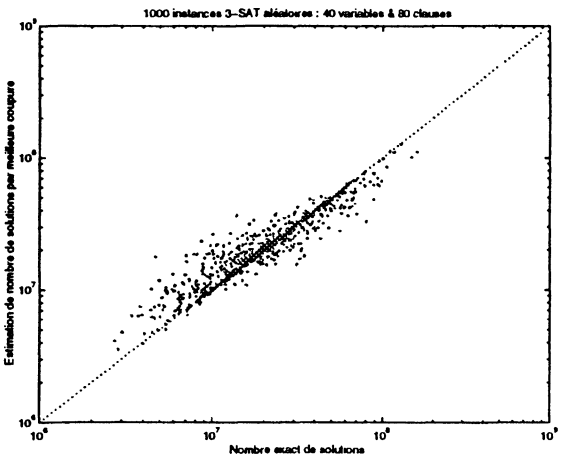

FIG. 4 – *40 variables et 80 clauses*

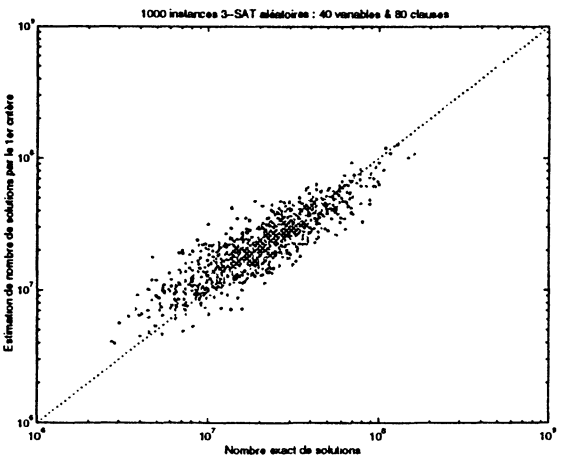

FIG. 5 – *40 variables et 80 clauses*

FIG. 6 – *Corrélation entre le nombre exact et le nombre estimé de solutions pour 1000 instances de la classe d'instances 3SAT aléatoires de 40 variables pour un rapport clauses/variables de 1.2, de 2. La première colonne de figures présente l'estimation donnée par la meilleure coupure connaissant le nombre exact de solutions; la seconde colonne l'estimation donné par le 1er critère*

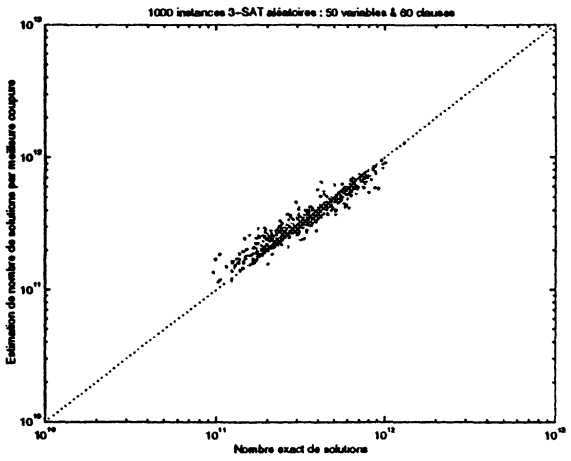

FIG. 7 – *50 variables et 60 clauses*

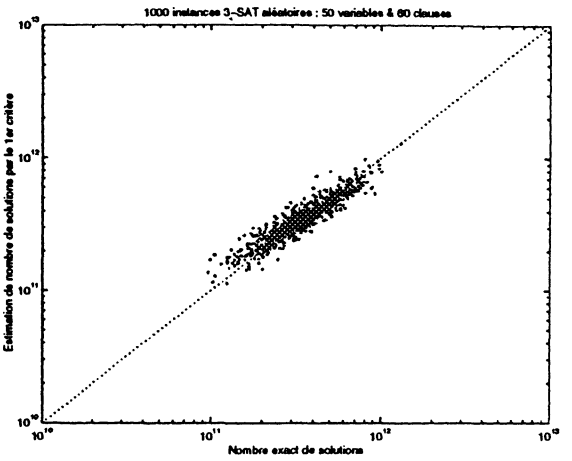

FIG. 8 – *50 variables et 60 clauses*

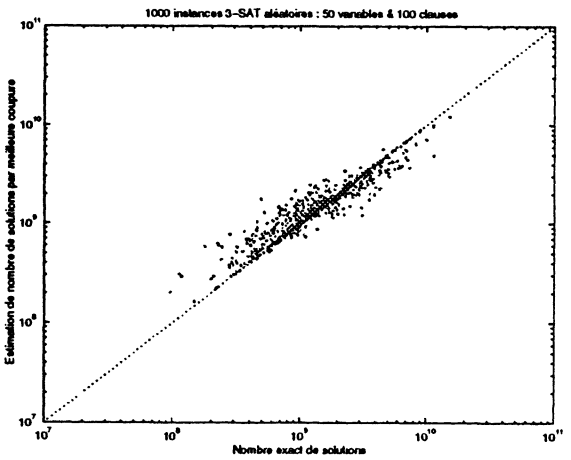

FIG. 9 – *50 variables et 100 clauses*

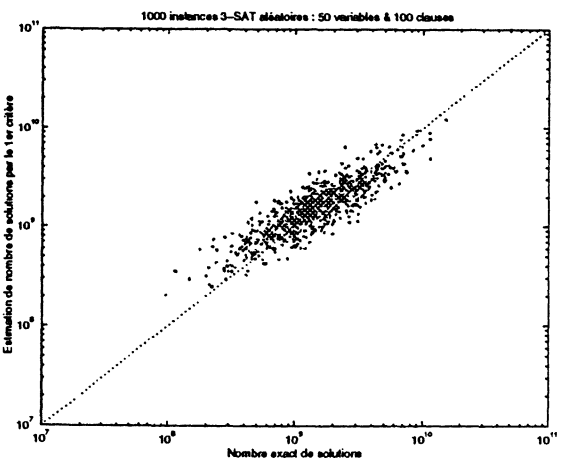

FIG. 10 – *50 variables et 100 clauses*

FIG. 11 – *Corrélation entre le nombre exact et le nombre estimé de solutions pour 1000 instances de la classe d'instances 3SAT aléatoires de 50 variables pour un rapport clauses/variables de 1.2, de 2. La première colonne de figures présente l'estimation donnée par la meilleure coupure connaissant le nombre exact de solutions ; la seconde colonne l'estimation donné par le 1er critère*

Pour estimer la qualité de l'approximation, nous introduisons la notion de déviation absolue :

DÉFINITION 3. Pour une instance SAT donnée, on appelle *déviation absolue* le rapport suivant :

$$\frac{|NbAp - NbEx|}{\max(NbAp, NbEx)}$$

où $NbAp$ est le nombre approché du nombre exact de solutions $NbEx$ de l'instance SAT.

Le tableau 1 montre que l'estimation optimale du nombre de solutions est de très bonne qualité. On remarquera une dispersion plus importante de la déviation pour les instances se situant pour un rapport clauses/variables de 2. Nos critères d'estimation de la meilleure coupure montre une dispersion identique à celle de la meilleure coupure mais une estimation de moins bonne qualité.

		40 variables		50 variables	
		48 clauses	80 clauses	60 clauses	100 clauses
C_M	moyenne	0.0444	0.1079	0.0599	0.1138
	médiane	0.0176	0.0406	0.0227	0.0358
	écart-type	0.0610	0.1363	0.0778	0.1465
C_1	moyenne	0.0932	0.2086	0.1098	0.2276
	médiane	0.0784	0.1846	0.0931	0.2048
	écart-type	0.0701	0.1438	0.0827	0.1529
C_2	moyenne	0.0960	0.2086	0.1104	0.2290
	médiane	0.0803	0.1862	0.0943	0.2054
	écart-type	0.0727	0.1432	0.0832	0.1534

TAB. 1 – *Distribution de la déviation absolue obtenue par les différents critères : C_M désigne la meilleure coupure possible connaissant le nombre exact de solutions, C_1 (resp. C_2) le premier (resp. second) critère présenté ; échantillon de taille 1000 pour chaque classe d'instances 3SAT aléatoires considérée.*

Plus le nombre de variables des instances est grand, plus le gain de temps est important (TAB. 2) : lorsque les instances se situent au pic de difficulté pour le problème #SAT, le gain est très important et augmente de façon exponentielle avec le nombre de variables. Lorsqu'on si place à droite du pic de difficulté, le gain, qui aurait pu être défavorable, reste important.

Les résultats obtenus ici sont naturellement moins précis que ceux de [13] mais meilleurs que ceux existants dans la littérature actuelle à notre connaissance.

5. CONCLUSION ET PERSPECTIVES

Nous avons présenté dans cet article une méthode de résolution approchée du problème de dénombrement des solutions d'une instance du problème SAT. Elle est basée sur le principe général de « diviser pour résoudre » et fait appel à des méthodes

P/N	N	gain moyen	temps exact	temps appr.
1.2	40	16	14s	1s
	50	37	3m20s	6s
	60	170	2h01m25s	58s
2	40	3	6s	2s
	50	7	1m45s	17s
	60	14	30m15s	2m45s

TAB. 2 – *Pour 1000 instances 3SAT aléatoires N variables, P clauses : gain moyen en temps ; temps moyen pour la résolution exacte ; temps moyen pour la résolution approchée ; les temps d'exécution sont des temps cpu obtenus sur une machine Sun Ultra 1*

de l'analyse combinatoire des données (la sériation) ainsi qu'à des considérations probabilistes pour segmenter le résultat de la sériation. La qualité des résultats obtenus est bonne. Notre recherche immédiate va porter sur la détermination d'un intervalle de confiance associé à chaque résultat afin de pouvoir considérer des instances SAT de taille plus importante et donc dont la résolution exacte n'est pas possible en temps raisonnable.

BIBLIOGRAPHIE

[1] Aspvall (B.), Plass (M.F.) et Tarjan (R.E.). – A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, vol. 8, n° 3, mars 1979, pp. 121–123.

[2] Bailleux (O.) et Chabrier (J.J.). – Counting by statistics on search trees: Application to constraint satisfaction problems. *Intelligent Data Analysis*, 1997.

[3] Cook (S.A.). – The complexity of theorem-proving procedures. *In : 3rd Annual ACM Symposium on the Theory of Computing*, éd. par ACM, pp. 151–158. – New-York, 1971.

[4] Dubois (O.). – Counting the number of solutions for instances of satisfiability. *Theoretical Computer Science*, no81, 1991, pp. 49–64.

[5] Karp (R.M.) et Luby (M.). – Monte-carlo algorithms for enumeration and reliability problems. *In : 24th IEEE Symposium of Foundations of Computer Science*, pp. 56–64. – 1983.

[6] Lassaigne (R.) et de Rougemont (M.). – *Logique et complexité.* – Hermes, 1996.

[7] Leredde (H.). – *La méthode des pôles d'attraction, la méthode des pôles d'agrégation ; deux nouvelles familles d'algorithmes en classification automatique et sériation.* – Thèse de 3^e cycle, Université de Paris VI, octobre 1979.

[8] Lerman (I.-C.). – Analyse du phénomène de la "sériation" à partir d'un tableau d'incidence. *Mathématiques et Sciences Humaines*, no38, 1972, pp. 39–57.

[9] Lerman (I.-C.). – Croisement de classifications floues. *Publication de l'institut statistique des universités de Paris*, vol. XXIV, n° 1-2, 1979, pp. 13–46.

[10] Lerman (I.-C.). – *Cartesian and Statistical Approaches of the Satisfiability Problem.* – rapport de recherche Inria n° 1685, Irisa, mai 1992.

[11] Lerman (I.-C.). – Statistical reduction of the satisfiability problem by means of a classification method. *Data Science and its Application*, 1995, pp. 219–234.

[12] Lerman (I.-C.), Peter (P.) et Leredde (H.). – Principes et calculs de la méthode implantée dans le programme CHAVL (classification hiérarchique par analyse de la vraisemblance des liens) 1ère partie. *La revue de Modulad*, no12, décembre 1993, pp. 33–70.

[13] Lerman (I.-C.) et Rouat (V.). – Une résolution approchée du problème #SAT par un algorithme de sériation. *In: Cinquièmes Rencontres de la Société Francophone de Classification*, pp. 29–34. – Lyon, septembre 1997.

[14] Lozinskii (E.L.). – Counting propositional models. *Information Processing Letters*, no41, avril 1992, pp. 327–332.

[15] Marcotorchino (F.). – Block seriation problems: a unified approach. *Applied Stochastic Models and Data Analysis*, vol. 3, n° 2, juin 1987, pp. 73–91.

[16] Mitchell (D.), Selman (B.) et Levesque (H.). – Hard and easy distributions of SAT problems. *In: Tenth National Conference on Artificial Intelligence (AAAI-92)*, pp. 459–465. – San Jose, 1992.

[17] Papadimitriou (C.H.). – *Computational complexity.* – Addison Wesley, 1994.

[18] Pearson (K.). – Notes on the history of correlation. *Biometrika*, no13, 1920, pp. 25–45.

[19] Roth (D.). – On the hardness of approximate reasoning. *Artificial Intelligence*, no82, 1996, pp. 273–302.

[20] Rouat (V.). – *Validité de l'approche classification dans la réduction statistique de la complexité de #SAT.* – Thèse de doctorat, Université de Rennes 1, janvier 1999.

[21] Rouat (V.) et Lerman (I.-C.). – Utilisation de la sériation pour une résolution approchée du problème #SAT. *In: JNPC'97, résolution pratique de problèmes NP-complets*, pp. 55–60. – Rennes, avril 1997.

[22] Rouat (V.) et Lerman (I.-C.). – Problématique de la coupure dans la résolution de #SAT par sériation. *In: JNPC'98, résolution pratique de problèmes NP-complets*, pp. 109–114. – Nantes, mai 1998.

[23] Simon (J.C.) et Dubois (O.). – Number of solutions of satisfiability instances – applications to knowledge bases. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 3, n° 1, 1989, pp. 53–65.

[24] Toda (S.). – On the computational power of PP and $\oplus$P. *In: 30th Annual Symposium on Foundations of Computer Science*, pp. 514–519. – 1989.

[25] Valiant (L.G.). – The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, vol. 8, n° 3, août 1979, pp. 410–421.