

Codage SAT pour l'OPP

Richard Ostrowski

05 avril 2023

Motivations

Les problèmes de packing (et variantes) ont de nombreuses applications pratiques :

- chargement de camions
- l'agencement de circuits intégrés à grande échelle (LSI)
- découpes pour minimiser les pertes (cutting)

Solveurs SAT de plus en plus performants :

- nombreux domaines d'applications : théorie des graphes, cryptographie, vérification formelle, planning, bioinformatique. . .
- utilisés pour trouver un contre exemple pour les triplets pythagoricien (taille de la preuve 4To)
- la performance est lié à la façon dont sont codés certains types de contraintes

Introduction

- Problème SAT
- Encodages pour SAT
 - direct encoding
 - support encoding
 - log encoding
 - order encoding
- Modélisations SAT de OOP
 - order encoding
 - à base de graphes d'intervalles
 - nouvelle modélisation
- Résultats expérimentaux
- Perspectives

Définitions

Problème de satisfaisabilité booléenne

- Etant donné une formule logique propositionnelle Σ , déterminer s'il existe une affectation des variables propositionnelles rendant Σ vraie
- Problème central en théorie de la complexité : premier problème à avoir été prouvé NP-Complet
- Les formules sont représentées en forme normale conjonctive :
 - Σ est une conjonction de clauses ($C_1 \wedge C_2 \wedge \dots \wedge C_n$)
 - une clause est une disjonction de littéraux ($l_1 \vee l_2 \vee l_3$)
 - un littéral l_i est une variable propositionnelle positive l_i ou négative $\neg l_i$

Du bon choix de la modélisation : contraintes de cardinalités

Pour $k \in \mathbb{N}$, n variables propositionnelles $\{x_1, x_2, \dots, x_n\}$ et une relation $\# \in \{<, \leq, >, \geq, =\}$, une contrainte de cardinalité est de la forme : $x_1 \vee x_2 \vee \dots \vee x_n \# k$

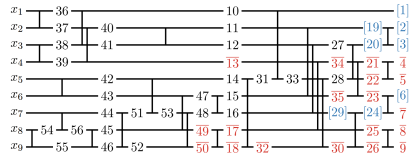
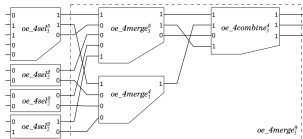
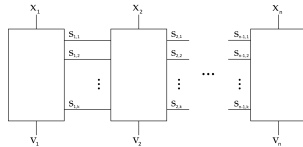


Figure 1: Séquentiel, Sorting , Odd/Even

Section 1

Encodages pour SAT

Direct, Supports et log encodings

Soit deux variables X et Y d'un CSP, avec $D_x = D_y = \{1, 2, 3, 4\}$ et une relation de type pair ou impair.

direct encoding

- chaque valeur du domaine d'une variable est représentée par une variable propositionnelle ($x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$)
- contrainte au moins 1 : les deux clauses $x_1 \vee x_2 \vee x_3 \vee x_4$ et $y_1 \vee y_2 \vee y_3 \vee y_4$
- contrainte au plus 1 : les clauses binaires $\{\neg x_1 \vee \neg x_2, \neg x_1 \vee \neg x_3, \neg x_1 \vee \neg x_4, \neg x_2 \vee \neg x_3, \neg x_2 \vee \neg x_4, \neg x_3 \vee \neg x_4, \neg y_1 \vee \neg y_2, \neg y_1 \vee \neg y_3, \neg y_1 \vee \neg y_4, \neg y_2 \vee \neg y_3, \neg y_2 \vee \neg y_4, \neg y_3 \vee \neg y_4\}$
- les clauses binaires de la relation : $\{\neg x_1 \vee \neg y_2, \neg x_1 \vee \neg y_4, \neg x_2 \vee \neg y_1, \neg x_2 \vee \neg y_3, \neg x_3 \vee \neg y_2, \neg x_3 \vee \neg y_4, \neg x_4 \vee \neg y_1, \neg x_4 \vee \neg y_3\}$

Direct, Supports et log encodings

Soit deux variables X et Y d'un CSP, avec $D_x = D_y = \{1, 2, 3, 4\}$ et une relation de type pair ou impair.

support encoding

- chaque valeur du domaine d'une variable est représentée par une variable propositionnelle ($x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$)
- contrainte au moins 1 : les deux clauses $x_1 \vee x_2 \vee x_3 \vee x_4$ et $y_1 \vee y_2 \vee y_3 \vee y_4$
- contrainte au plus 1 : les clauses binaires $\{\neg x_1 \vee \neg x_2, \neg x_1 \vee \neg x_3, \neg x_1 \vee \neg x_4, \neg x_2 \vee \neg x_3, \neg x_2 \vee \neg x_4, \neg x_3 \vee \neg x_4, \neg y_1 \vee \neg y_2, \neg y_1 \vee \neg y_3, \neg y_1 \vee \neg y_4, \neg y_2 \vee \neg y_3, \neg y_2 \vee \neg y_4, \neg y_3 \vee \neg y_4\}$
- les clauses supports de la relation :
 $\{\neg x_1 \vee y_1 \vee y_3, \neg x_2 \vee y_2 \vee y_4, \neg x_3 \vee y_1 \vee y_3, \neg x_4 \vee y_2 \vee y_4, \neg y_1 \vee x_1 \vee x_3, \neg y_2 \vee x_2 \vee x_4, \neg y_3 \vee x_1 \vee x_3, \neg y_4 \vee x_2 \vee x_4\}$

Direct, Supports et log encodings

Soit deux variables X et Y d'un CSP, avec $D_x = D_y = \{1, 2, 3, 4\}$ et une relation de type pair ou impair.

log encoding

- les valeurs des domaines sont représentées par un nombre logarithmique de variables propositionnelles (x_1, x_2, y_1, y_2)
 - le couple $(x_1 = 0, x_2 = 0)$ représente la première valeur du domaine de X ($(x_1 = 1, x_2 = 1)$ la dernière)
- les clauses de la relation sont codées soit en direct ou support encoding

Order encoding

une comparaison $x \leq a$ est encodée par une variable propositionnelle $p(x \leq a)$

- Avantages :
 - La propagation unitaire (SAT) correspond à la propagation des bornes (CSP)
 - problèmes traitables en CSP le restent aussi pour SAT ¹
- Inconvénient :
 - instances très grandes quand le domaine des variables devient très grand
 - Une contrainte du type $z = x + y$ demande d^2 clauses

¹Petke, J., Jeavons, P.: The Order Encoding: From Tractable CSP to Tractable SAT 2011

Order encoding

Soit deux variables X et Y d'un CSP, avec
 $D_x = D_y = \{1, 2, 3, 4, 5, 6\}$ et une relation de type pair ou impair.

Order encoding

- les valeurs des domaines sont représentés par les variables propositionnelles ($x_1, x_2, x_3, x_4, x_5, y_1, y_2, y_3, y_4, y_5$)

| valeurs | x_1, y_1 | x_2, y_2 | x_3, y_3 | x_4, y_4 | x_5, y_5 |
|---------|------------|------------|------------|------------|------------|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 0 | 1 | 1 | 1 | 1 |
| 5 | 0 | 0 | 1 | 1 | 1 |
| 2 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 |

- $\forall i \forall j, i, j \in \{1, 2, 3, 4, 5\}, i < j : \neg x_i \vee x_j$
- les 2 contraintes binaires pour la relation : $x_3 \vee \neg y_3$ et $\neg x_3 \vee y_3$

Section 2

Modélisations SAT de OOP

Orthogonal packing problem

Soit un conteneur de taille (W, H) et un ensemble de pièces rectangulaires P_i de taille (w_i, h_i) ($W, H, w_i, h_i \in \mathbb{N}$) tel que $W \times H \geq \sum_i^n w_i \times h_i$

- existe-t-il un placement des pièces P_i pouvant tenir dans le conteneur ?
- variante : cutting packing problem



Figure 2: non cutting vs cutting

Graphe d'intervalles

- Un graphe d'intervalles est le graphe d'intersection d'un ensemble d'intervalles de la droite réelle

Etant donnés des intervalles $I_1, I_2, \dots, I_n \in \mathbb{R}$, le graphe d'intervalles correspondant est $G = (V, E)$ où :

- $V = \{I_1, I_2, \dots, I_n\}$ et
- $\{I_i, I_j\} \in E \Leftrightarrow I_i \cap I_j \neq \emptyset \forall i, j \in \mathbb{N}$

Un graphe G est d'intervalles ssi il est **cordal** et son graphe **complémentaire** est un graphe de **comparabilité**

- pas facile à modéliser en SAT (pour les cycles)
- peut être décidé en temps en $O(|V| + |E|)$

G est d'intervalles ssi les cliques maximales M_1, M_2, \dots, M_k de G peuvent être ordonnées telles que pour tout $v \in M_i \cap M_k$, alors $v \in M_j$ pour tout entier j , $i \leq j \leq k$.

Modélisation Soh et al.²

Basée sur l'order encoding : chaque pièce possède un positionnement horizontal et vertical

On modélise le fait que pour deux pièces i et j :

- soit i est à droite de j soit l'inverse
- soit i est au dessus de j soit l'inverse
- clauses de type : $lr_{i,j} \vee lr_{j,i} \vee ud_{i,j} \vee ud_{j,i}$

En supposant l'une des variables à vraie :

- pour chaque position de la pièce à gauche on détermine la position minimale de la pièce de droite (largeur ou hauteur)
- exemple : si $lr_{i,j} = 1$ alors j est située à une distance d'au moins la largeur de i

²T. Soh, K. Inoue, N. Tamura, M. Banbara, and H. Nabeshima. A SAT-based Method for Solving the Two-dimensional Strip Packing Problem, 2008

Modélisation Pinto et al.³

Basée sur la caractérisation de S.P. Fekete et al. de l'OOP:

- pour chaque dimension i le graphe G_i représente le chevauchement des pièces dans la dimension i
- chaque G_i doit être d'intervalles
- dans chaque graphe G_i , tout stable doit être i -réalisable
- une arête entre deux pièces ne peut apparaître dans tous les graphes

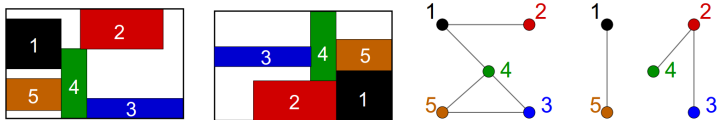


Fig. 1. Two packings corresponding to the same interval graphs in a two-dimensional space

³S. Grandcolas, C. Pinto, A SAT encoding for Multi-dimensional Packing Problems

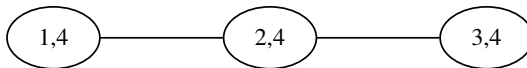
Modélisation Pinto et al.

L'espace de recherche se fera sur une représentation des graphes d'intervalles par une suite de cliques ordonnées (de taille n):

- chaque pièce sera représentée par une clique de début et de fin
 - toute clique intermédiaire possèdera cette pièce
- les stables i -réalisables seront calculés et modélisés pour chaque G_i par une clause

Inconvénients :

- Le nombre de stables i -réalisables peut être combinatoire
- Certaines configurations peuvent ne pas représenter la réalité



Nouvelle modélisation basée sur les graphes d'intervalles

- l'espace de recherche : suite de cliques ordonnées (de taille W ou H)
- Chaque pièce a une position (order encoding) et conserve sa largeur ou hauteur
- Une contrainte sous la forme d'une clause indiquera si deux pièces se trouvent dans une même clique

Avantages de cette modélisation :

- pas besoin de calculer les stables i -réalisables
- élimination des configurations ne représentant pas la réalité
- gain de 20% en variables et 45% en clauses par rapport à Soh

Désavantage :

- la même que pour l'order encoding : plus W et H sont grand plus il faudra de variables propositionnelles
 - amélioration possible : compact order encoding

Section 3

Résultats expérimentaux

Problème $1 \times n, 2 \times n - 1, \dots, n \times 1$ ($n=13$ et $n=14$)

| problème | tps PS | #V | #C | tps Soh | #V | #C | tps R | #V | #C | tps satpack |
|------------|--------|-----|------|---------|-----|------|-------|------|-------|-------------|
| 13_(13x35) | 0 | 754 | 6111 | 0.46 | 598 | 3345 | 0 | 3224 | 13139 | 0.27 |
| 13_(19x24) | 0 | 689 | 5265 | 7.25 | 533 | 2860 | 0.56 | 3224 | 13022 | 6.26 |
| 13_(17x27) | 0 | 702 | 5434 | 1.67 | 546 | 2958 | 0.58 | 3224 | 13039 | 2.07 |
| 13_(20x23) | 0 | 689 | 5265 | 15.15 | 533 | 2858 | 0.69 | 3224 | 13019 | 4.05 |
| 13_(14x33) | 0 | 741 | 5941 | 0.48 | 585 | 3247 | 0.01 | 3224 | 13115 | 3.04 |
| 13_(21x22) | 0 | 689 | 5265 | 18.47 | 533 | 2857 | 0.58 | 3224 | 13020 | 5.6 |
| 13_(16x29) | 0 | 715 | 5603 | 0.29 | 559 | 3054 | 0.45 | 3224 | 13058 | 0.97 |

| problème | tps PS | #V | #C | tps Soh | #V | #C | tps R | #V | #C | tps satpack |
|------------|--------|-----|------|---------|-----|------|-------|------|--------|-------------|
| 14_(14x40) | 0 | 910 | 8072 | 1.06 | 728 | 4390 | 0.01 | 3934 | 16436 | 2.38 |
| 14_(16x35) | 0 | 868 | 7483 | 2.06 | 686 | 4061 | 1.15 | 3934 | 16326 | 1.48 |
| 14_(20x28) | 0 | 826 | 6895 | 18.2 | 644 | 3726 | 2.56 | 3934 | 16187 | 43 |
| 14_(17x33) | 0 | 854 | 7287 | 3.63 | 672 | 3950 | 1.8 | 3934 | 16279 | 9.71 |
| 14_(21x27) | 0.19 | 826 | 6895 | 45.45 | 644 | 3722 | 2.95 | 3934 | 16.191 | 99 |
| 14_(15x38) | 0 | 896 | 7875 | 0.37 | 714 | 4277 | 0.44 | 3934 | 16399 | 7 |

Problème $1 \times n, 2 \times n - 1, \dots, n \times 1$ ($n=15$ et $n=16$)

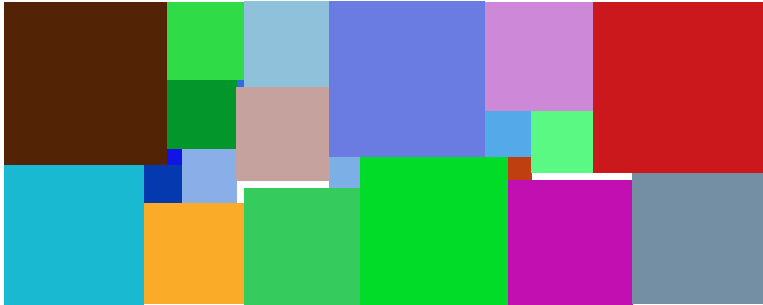
| problème | tps PS | #V | #C | tps Soh | #V | #C | tps R | #V | #C | tps satpack |
|------------|--------|------|-------|---------|-----|------|-------|------|-------|-------------|
| 15_(17x40) | 0.05 | 1035 | 9720 | 13.56 | 825 | 5247 | 4.5 | 4740 | 20188 | 65 |
| 15_(20x34) | 0.05 | 990 | 9045 | 50.72 | 780 | 4870 | 5.8 | 4740 | 19952 | 7.86 |
| 15_(22x31) | 0.11 | 975 | 8820 | 99 | 765 | 4741 | 6.8 | 4740 | 19868 | 35 |
| 15_(18x38) | 0.18 | 1020 | 9495 | 21.9 | 810 | 5121 | 5.46 | 4740 | 20107 | 207 |
| 15_(19x36) | 0.22 | 1005 | 9270 | 44.47 | 795 | 4995 | 7.5 | 4740 | 20031 | 6.51 |
| 15_(16x43) | 0.17 | 1065 | 10170 | 4.73 | 855 | 5494 | 2.74 | 4740 | 20301 | 1.3 |
| 15_(15x46) | 0.08 | 1095 | 10621 | 7.93 | 885 | 5742 | 4.46 | 4740 | 20419 | 15.4 |
| 15_(23x30) | 1.49 | 975 | 8820 | 307 | 765 | 4737 | 9.4 | 4740 | 19865 | 890 |
| 15_(21x33) | 2.2 | 990 | 9045 | 130 | 780 | 4865 | 8.7 | 4740 | 19929 | 76 |
| 15_(24x29) | 0.28 | 975 | 8820 | 0.47 | 765 | 4734 | 0.44 | 4740 | 19858 | 35 |

| problème | tps PS | #V | #C | tps Soh | #V | #C | tps R |
|------------|--------|------|-------|---------|------|------|-------|
| 16_(16x51) | 0.04 | 1280 | 13369 | 9.27 | 1040 | 7193 | 5.95 |
| 16_(17x48) | 0.08 | 1248 | 12856 | 30.43 | 1008 | 6912 | 9.9 |
| 16_(24x34) | 0.13 | 1136 | 11064 | 410 | 896 | 5920 | 20 |
| 16_(19x43) | 0.16 | 1200 | 12088 | 77.2 | 960 | 6490 | 12.46 |
| 16_(21x39) | 0.46 | 1168 | 11576 | 308 | 928 | 6206 | 24.57 |
| 16_(20x41) | 0.61 | 1184 | 11832 | 222 | 944 | 6348 | 15.76 |
| 16_(25x33) | 4.3 | 1136 | 11064 | to | 896 | 5916 | 39.87 |
| 16_(18x46) | 1.67 | 1232 | 12600 | 54.5 | 992 | 6769 | 14.24 |
| 16_(23x36) | 0.1 | 1152 | 11320 | 7.36 | 912 | 6060 | 11.64 |

Problème des carrés $1 \times 1, 2 \times 2, \dots, n \times n$

| problème carrés | PS(s) | Soh(s) | R(s) | satpack(s) |
|-----------------|-------|--------|------|------------|
| 14(18) | 0.62 | 1.83 | 1.13 | 46.55 |
| 15(27) | 3.19 | 6.84 | 3.11 | 193 |
| 16(16) | 2.19 | 12.56 | 4.11 | 192 |
| 17(10) | 2 | 11.87 | 5.18 | 370 |
| 18(29) | 152 | 468 | 37 | to |
| 19(23) | 120 | 1163 | 88 | to |
| 20(25) | 498 | to | 86 | to |
| 21(35) | to | to | 557 | to |
| 22(30) | to | to | 2360 | to |

Problème des carrés : solution $n = 22(39 \times 98)$



Autres problèmes

| problème | PS(s) | #V | #C | Soh(s) | #V | #C | R(s) | #V | #C | satpack |
|----------|-------|------|-------|--------|------|-------|------|-------|--------|---------|
| HT1 | 0 | 951 | 8104 | 0.04 | 711 | 4204 | 0.04 | 5648 | 25919 | 13.5 |
| HT2 | 0 | 1049 | 9231 | 1.17 | 777 | 5075 | 0.3 | 6664 | 30883 | 304 |
| HT3 | 0 | 958 | 8216 | 0.03 | 718 | 4377 | 0.05 | 5648 | 26415 | 33 |
| HT4 | 8.4 | 2326 | 29600 | 0.02 | 1726 | 15240 | 0.04 | 19400 | 232479 | 512 |
| HT5 | 0 | 2318 | 29400 | 2.82 | 1718 | 15456 | 2.78 | 19400 | 233371 | 526 |
| HT6 | 0 | 2716 | 29350 | 0.19 | 1716 | 15426 | 0.3 | 19400 | 319141 | 322 |
| HT7 | 6.57 | 3565 | 59318 | 14.38 | 2809 | 30429 | 24.6 | 26684 | 621773 | to |
| HT8 | 0.26 | 3774 | 64322 | 122 | 2962 | 33545 | 100 | 29464 | 656907 | to |
| HT9 | 15.3 | 3597 | 60214 | 0.23 | 2841 | 30936 | 0.12 | 26684 | 505444 | to |

Autres problèmes de la littérature

| problème | PS(s) | Soh(s) | R(s) | satpack |
|---------------|-------|--------|------|---------|
| E00N23(20x20) | 0.97 | to | to | to |
| E00N23(21x20) | 0.15 | 0 | 0 | to |
| E00N23(20x21) | 0.97 | 0 | 0 | 33 |
| E00X23(20x20) | 21.16 | to | to | to |
| E00X23(21x20) | 0.65 | 0 | 0 | to |
| E00X23(20x21) | 21.16 | 0 | 0 | to |
| E02N20 | 5.18 | 0 | 0 | 0 |
| E03N16 | 0.86 | 0 | 0 | 19 |
| E05X15 | 1.29 | 0.1 | 0.08 | 3.35 |
| j1(40x15) | 0.05 | 19 | 2.76 | to |
| j1(41x15) | 3.26 | 0.18 | 0.33 | to |
| j1(40x16) | to | 0.02 | 0.16 | to |
| j2(40x15) | 1.55 | to | to | to |
| j2(41x15) | to | 2.1 | 5.31 | to |
| j2(40x16) | to | 0.06 | 1.3 | to |

Conclusion et perspectives

- Nouveau codage SAT pour les problèmes de packing
 - basé sur les graphes d'intervalles
 - utilisation de l'order encoding
- Perspectives
 - variantes de l'order encoding : sugar, azucar, abbacus
 - SAT incrémental pour le problème du strip packing
 - ajout de contraintes dans la modélisation pour les problèmes de cutting
 - minimiser la largeur des cliques