

SAT : des victoires contre les problèmes difficiles

 **Laurent Simon**

 LaBRi/Bordeaux INP, Bordeaux



Pourquoi s'intéresser à SAT ?

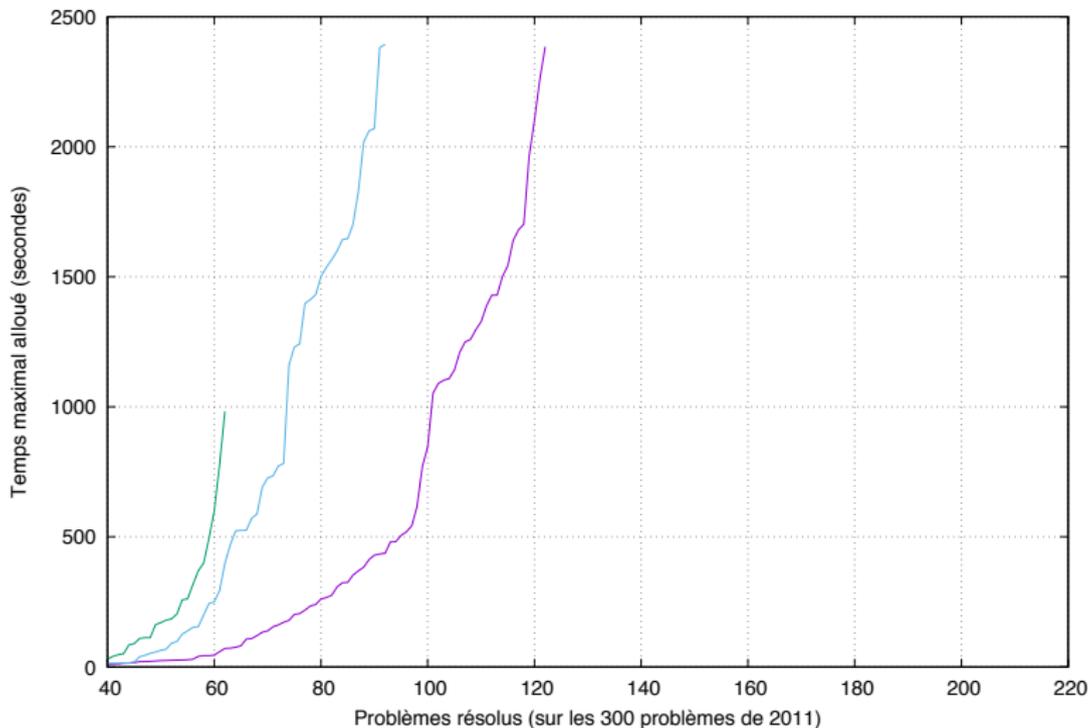
SAT, un problème théorique à **un million de dollars**

- Le principal problème ouvert en informatique théorique.
- Le plus simple des problèmes difficiles
- Il faut l'affronter dans beaucoup de problèmes du monde réel

S. Aaronson, MIT : « **If $P = NP$** , then the world would be a profoundly different place than we usually assume it to be. There would be no special value in 'creative leaps,' no fundamental gap between solving a problem and recognizing the solution once it's found. **Everyone who could appreciate a symphony would be Mozart ; everyone who could follow a step-by-step argument would be Gauss.** » ¹

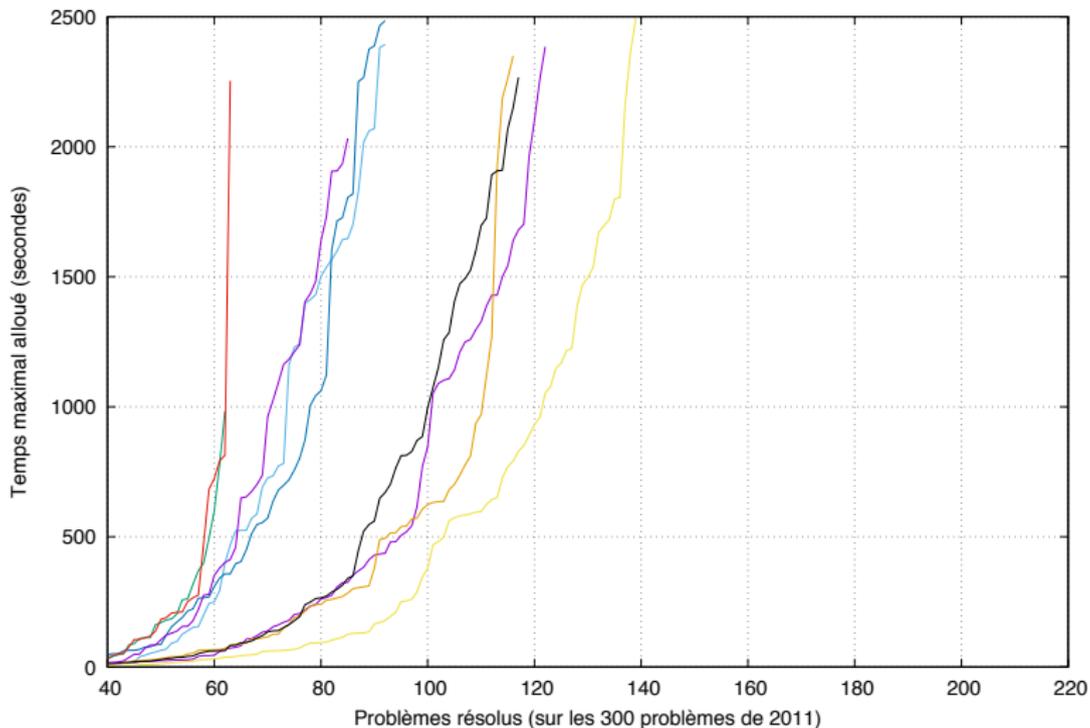
1. D'après [Vardi, 2015]

Performances après 2001



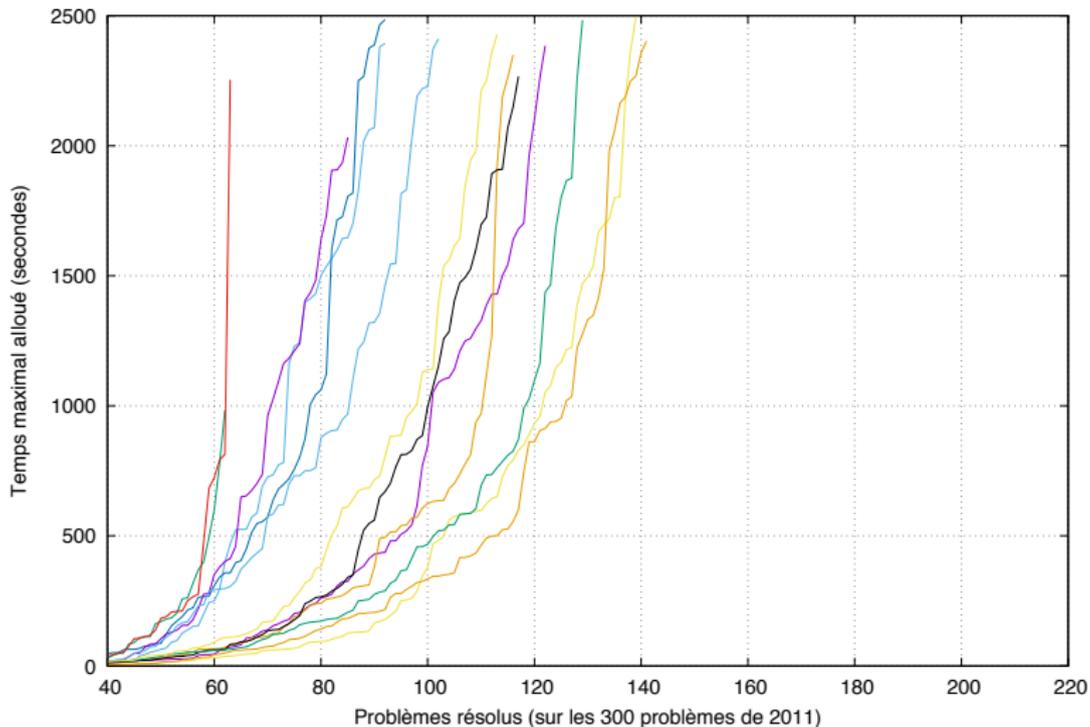
2002

Performances après 2001



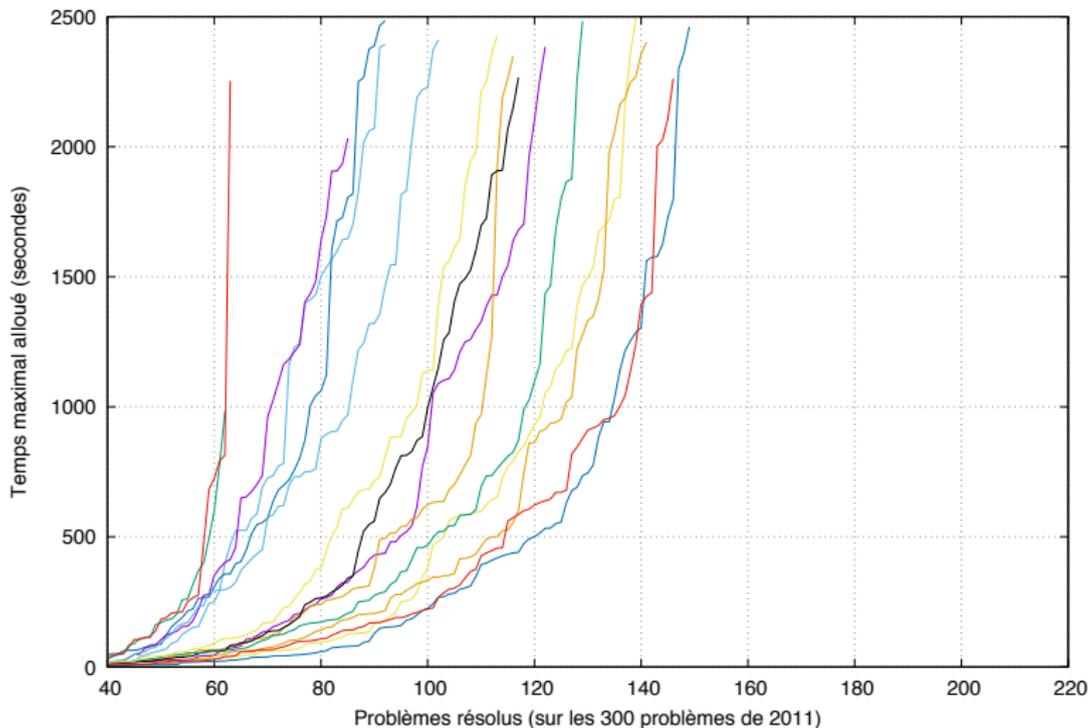
2003

Performances après 2001



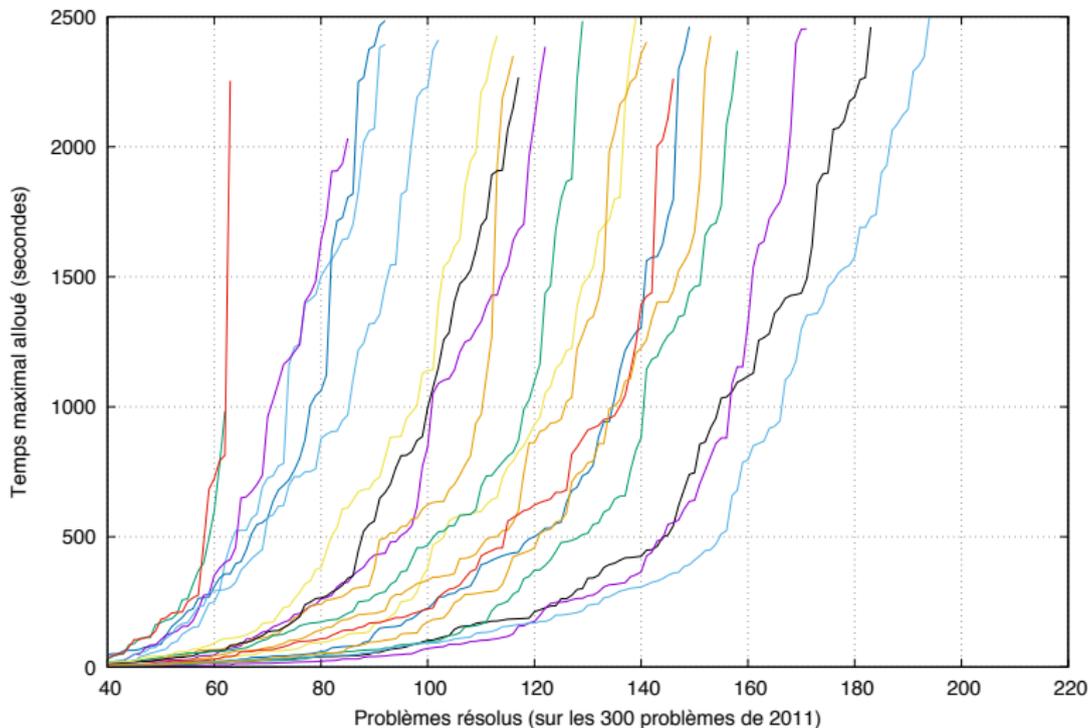
2005

Performances après 2001



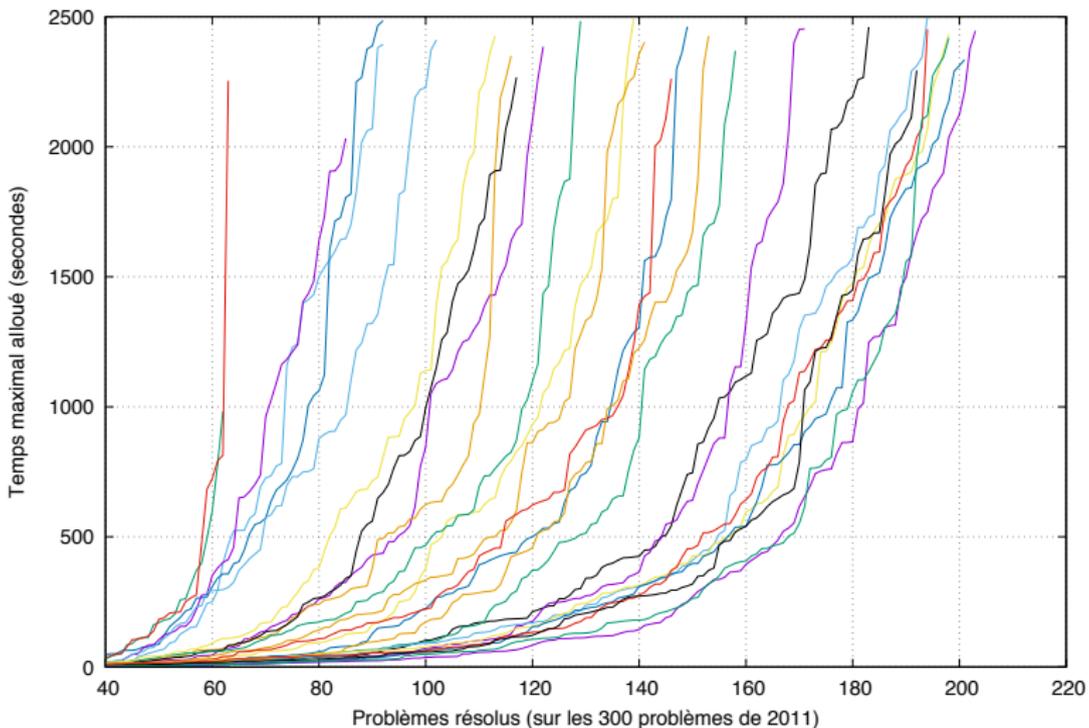
2007

Performances après 2001



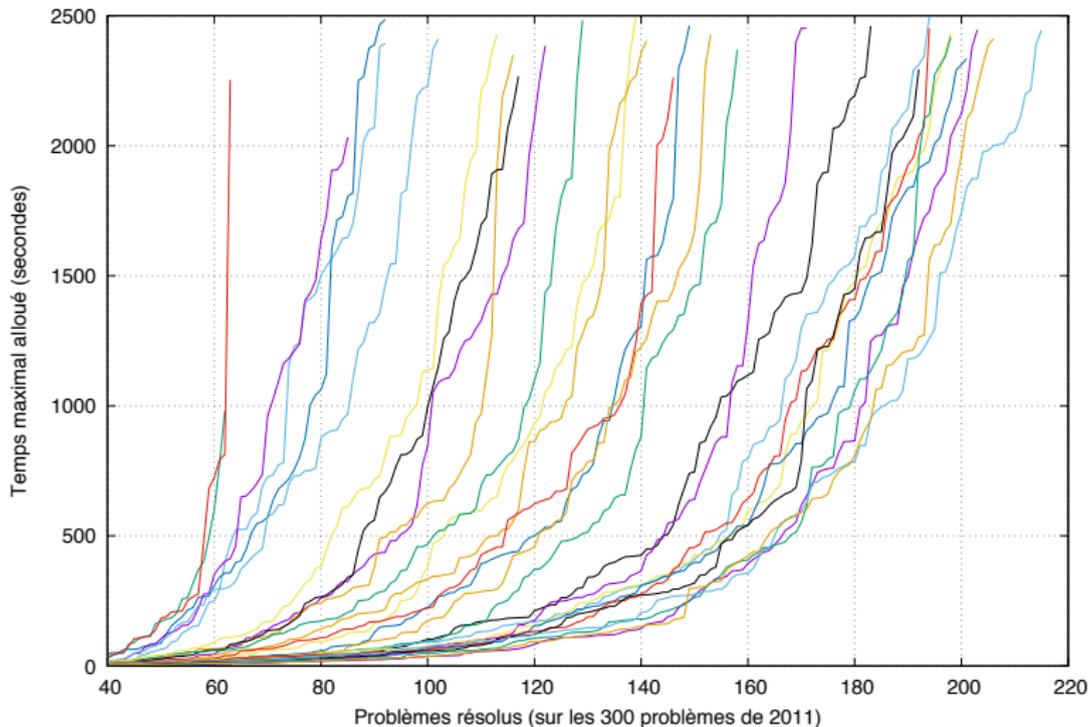
2009

Performances après 2001



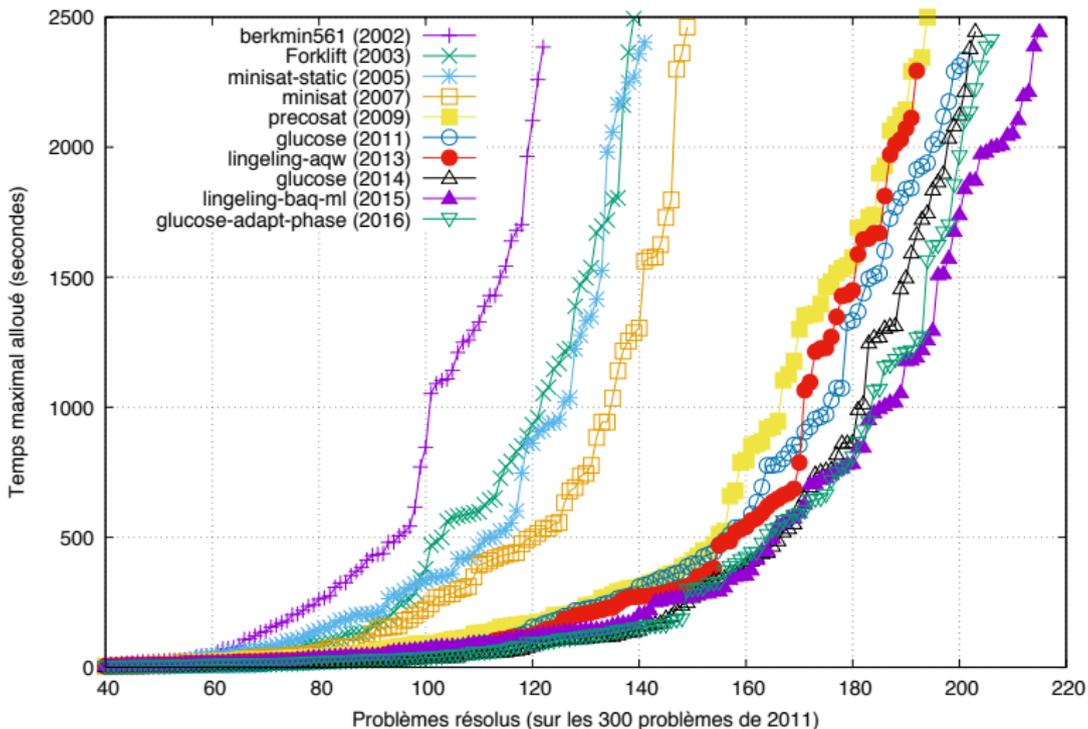
2014

Performances après 2001



2016

Performances après 2001



Les gagnants

Dans ce séminaire

L'avant révolution

- ➔ Impossible de passer à l'échelle

Chaff, le basculement :

« *careful engineering of all aspects of the search* »

- ➔ **La structure de données dirige** l'algorithmique
(*tous les ingrédients étaient là (SATO, GRASP, DLIS, ...)*)

L'après révolution (les SAT solveurs « modernes »)

- ➔ **Simplifier, Améliorer et ...comprendre**

Les challenges

- ➔ **Comprendre, Paralléliser**

SAT avant l'heure

1958 : Hilary Putnam et Martin Davis cherchent un financement pour travailler autour de la logique propositionnelle.

« *What we're interested in is good algorithms for propositional calculus* » (NSA)

Avant, des méthodes particulièrement inefficaces (tables de vérités, ...).

Premiers articles

- *Computational Methods in The Propositional calculus*
[Davis Putnam 1958]²
- *A Computing Procedure for Quantification Theory*
[Davis Putnam 1960]

2. Rapport interne NSA

1960, déjà une (quasi) compétitions !

« *The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes **the machine to compute for 21 minutes** without obtaining a result was worked successfully by **hand computation using the present method in 30 minutes** »*

[Davis et Putnam 1960], page 202.

Résoudre SAT avec DP

DP-60 : oublier les variables les unes après les autres.

Exemple : oublier x_1 .

$$x_1 \vee x_4$$

$$\overline{x_1} \vee x_4 \vee x_{14}$$

$$\overline{x_1} \vee \overline{x_3} \vee \overline{x_8}$$

$$x_1 \vee x_8 \vee x_{12}$$

$$x_1 \vee x_5 \vee \overline{x_9}$$

$$x_2 \vee x_{11}$$

$$\overline{x_3} \vee \overline{x_7} \vee x_{13}$$

$$\overline{x_3} \vee \overline{x_7} \vee \overline{x_{13}} \vee x_9$$

$$x_8 \vee \overline{x_7} \vee \overline{x_9}$$

Résoudre SAT avec DP

DP-60 : oublier les variables les unes après les autres.

Exemple : oublier x_1 .

$$x_1 \vee x_4$$

$$x_1 \vee x_8 \vee x_{12}$$

$$x_1 \vee x_5 \vee \overline{x_9}$$

$$\overline{x_1} \vee x_4 \vee x_{14}$$

$$\overline{x_1} \vee \overline{x_3} \vee \overline{x_8}$$

$$x_2 \vee x_{11}$$

$$\overline{x_3} \vee \overline{x_7} \vee x_{13}$$

$$\overline{x_3} \vee \overline{x_7} \vee \overline{x_{13}} \vee x_9$$

$$x_8 \vee \overline{x_7} \vee \overline{x_9}$$

Résoudre SAT avec DP

DP-60 : oublier les variables les unes après les autres.

Exemple : oublier x_1 .

$$x_1 \vee \begin{pmatrix} x_4 \\ x_8 \vee x_{12} \\ x_5 \vee \overline{x_9} \end{pmatrix}$$

$$\overline{x_1} \vee \begin{pmatrix} x_4 \vee x_{14} \\ \overline{x_3} \vee \overline{x_8} \end{pmatrix}$$

$$\begin{aligned} &x_2 \vee x_{11} \\ &\overline{x_3} \vee \overline{x_7} \vee x_{13} \\ &\overline{x_3} \vee \overline{x_7} \vee \overline{x_{13}} \vee x_9 \\ &x_8 \vee \overline{x_7} \vee \overline{x_9} \end{aligned}$$

Résoudre SAT avec DP

DP-60 : oublier les variables les unes après les autres.

Exemple : oublier x_1 .

$$\left(\begin{array}{l} x_4 \\ x_8 \vee x_{12} \\ x_5 \vee \overline{x_9} \end{array} \right) \vee \left(\begin{array}{l} x_4 \vee x_{14} \\ \overline{x_3} \vee \overline{x_8} \end{array} \right)$$

$$x_2 \vee x_{11}$$

$$\overline{x_3} \vee \overline{x_7} \vee x_{13}$$

$$\overline{x_3} \vee \overline{x_7} \vee \overline{x_{13}} \vee x_9$$

$$x_8 \vee \overline{x_7} \vee \overline{x_9}$$

Résoudre SAT avec DP

DP-60 : oublier les variables les unes après les autres.

Exemple : oublier x_1 .

$$x_4 \vee x_{14}$$

$$x_4 \vee \bar{x}_3 \vee \bar{x}_8$$

$$x_8 \vee x_{12} \vee x_4 \vee x_{14}$$

$$x_5 \vee \bar{x}_9 \vee x_4 \vee x_{14}$$

$$x_5 \vee \bar{x}_9 \vee \bar{x}_3 \vee \bar{x}_8$$

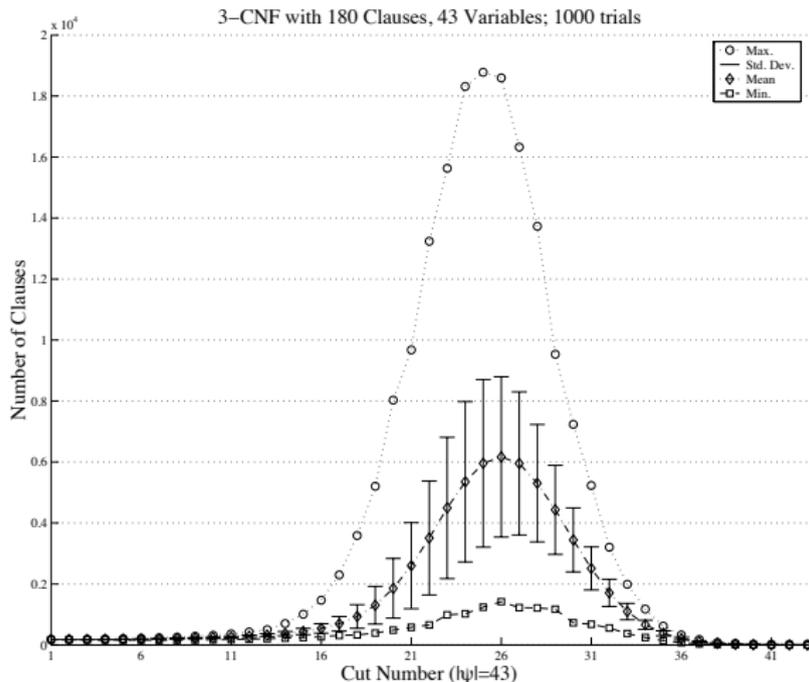
$$x_2 \vee x_{11}$$

$$\bar{x}_3 \vee \bar{x}_7 \vee x_{13}$$

$$\bar{x}_3 \vee \bar{x}_7 \vee \bar{x}_{13} \vee x_9$$

$$x_8 \vee \bar{x}_7 \vee \bar{x}_9$$

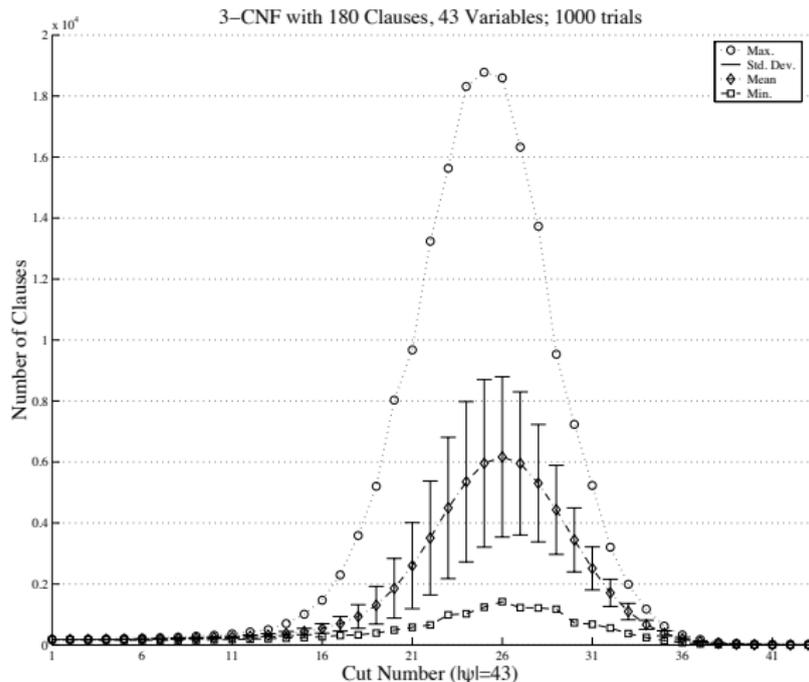
Gros problème de complexité spatiale



Impossible en pratique, même sur de petits exemples !

Mais possible dans certains cas spéciaux (prétraitement SAT)

Gros problème de complexité spatiale



Impossible en pratique, même sur de petits exemples !

Mais possible dans certains cas spéciaux (prétraitement SAT)

1962-2001 : le règne de DPLL

Explorer systématiquement l'espace des modèles partiels

- Choisir un littéral
- Essayer de trouver une solution avec ce littéral assignée à Vrai
- Si on ne trouve pas :
Trouver une solution avec ce littéral assigné à Faux

Parcours avec retour arrière des modèles partiels
La complétude est garantie par l'ordre de parcours

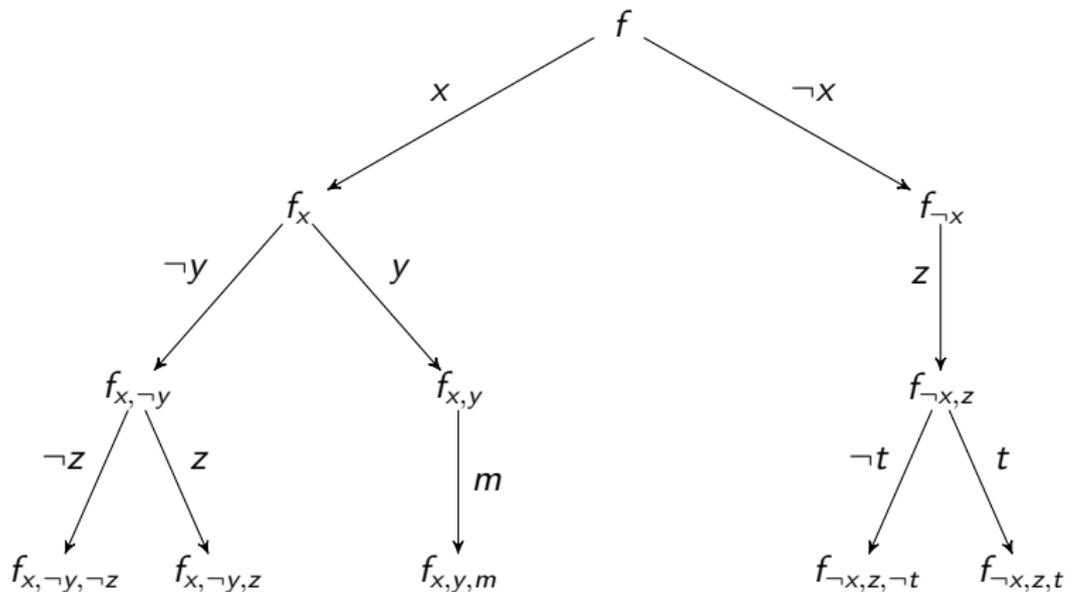
1962-2001 : le règne de DPLL

Explorer systématiquement l'espace des modèles partiels

- Choisir un littéral
- Essayer de trouver une solution avec ce littéral assignée à Vrai
- Si on ne trouve pas :
Trouver une solution avec ce littéral assigné à Faux

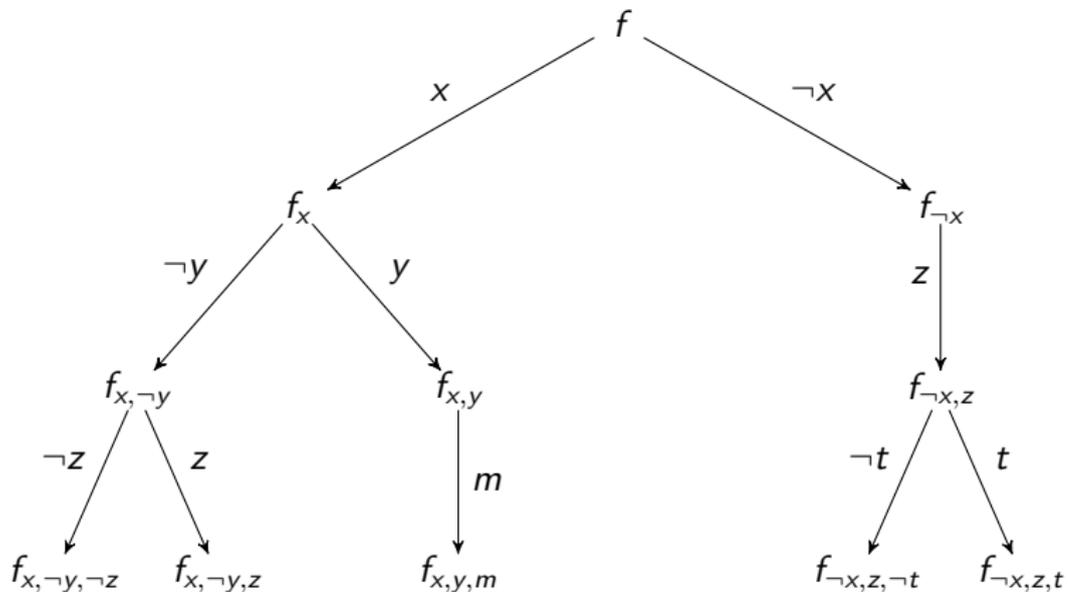
Parcours avec retour arrière des modèles partiels
La complétude est garantie par l'ordre de parcours

Recherche de modèle avec retour arrière



- Comment choisir la bonne variable (et sa valeur?)
- Chercher une solution ou une contradiction d'abord?

Recherche de modèle avec retour arrière



- Comment choisir la bonne variable (et sa valeur?)
- Chercher une solution ou une contradiction d'abord?

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv. Lit. Retour?

x_1 apparait dans 4 clauses dont 1 clause binaire

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)

x_4 apparaît dans une clause unaire

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
+	x_4	

x_3 apparaît dans 3 clauses dont une clause (nouvellement) binaire

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
	+	x_4
2	x_3	(d)

$\overline{x_8}$ apparaît dans une clause unaire

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
	+ x_4	
2	x_3	(d)
	+ $\overline{x_8}$	

x_{12} apparait dans une clause unaire

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
+	x_4	
2	x_3	(d)
+	$\overline{x_8}$	
+	x_{12}	

$x_{13}, \overline{x_7}$ apparaissent dans des clauses unaires

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
	+ x_4	
2	x_3	(d)
	+ $\overline{x_8}$	
	+ x_{12}	
	+ x_{13}	

$x_7, \overline{x_7}$ apparaissent dans des clauses unaires

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
+	x_4	
2	x_3	(d)
+	$\overline{x_8}$	
+	x_{12}	
+	x_{13}	
+	$\overline{x_7}$	

Conflit ! Défaire tout jusqu'à la dernière décision

DPLL sur un exemple

Formule

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Formule Simplifiée

$$\begin{aligned}
 &x_1 \vee x_4 \\
 &\overline{x_1} \vee x_4 \vee x_{14} \\
 &x_1 \vee \overline{x_3} \vee \overline{x_8} \\
 &x_1 \vee x_8 \vee x_{12} \\
 &x_2 \vee x_{12} \\
 &\overline{x_3} \vee \overline{x_{12}} \vee x_{13} \\
 &\overline{x_3} \vee x_7 \vee \overline{x_{13}} \\
 &x_8 \vee \overline{x_7} \vee \overline{x_{12}}
 \end{aligned}$$

Modèle partiel

Niv.	Lit.	Retour ?
1	$\overline{x_1}$	(d)
+	x_4	
*	$\overline{x_3}$	

Maintenant, $\overline{x_3}$ n'est plus une décision

Réfléchir ou explorer ?

Maintenir l'état courant de la formule :

- **Indispensable pour faire les bons choix**
- **Très coûteux en pratique**
- La majorité du temps est passé dans le maintien des compteurs heuristiques

Assigner une valeur à une variable :

- Annuler le poids des variables dans les clauses devenues vraies
- Mettre à jour le poids des variables dans les clauses raccourcies

Libérer la variable :

- Mettre à jour tous les compteurs !

Fin des années 90

Sur de petites instances

- Importance d'heuristiques bien informées (coûteuses)
- Les heuristiques *miment* ce que nous ferions
 - Variables apparaissant le plus dans les clauses les plus courtes
 - Variables permettant de propager le plus d'autres variables

Mais techniques non adaptées aux grosses instances

Mieux équilibrer Recherche/Réflexion

- DLIS donne de bons résultats sur les grosses instances
- Simplement choisir les variables les plus fréquentes

Séparation des méthodes pour SAT aléatoire des méthodes SAT applicatives



Plan du séminaire

Petite Histoire

2001-2005, la révolution CDCL

- La révolution
- Zchaff

Compétitions et problèmes typiques

2006-2016, vers des systèmes complexes

Problèmes ouverts

1999, les prémices d'une révolution

Des problèmes gigantesques arrivent : Planning & Bounded Model Checking

- Planning as Satisfiability. [Kautz and Selman, 92]
- Symbolic Model Checking using SAT procedures instead of BDDs. [Biere & al. 99]
- Quasi-incapacité des solveurs SAT à gérer autant de clauses

La suprématie DPLL se fissure

- GRASP : **apprentissage de clauses** dans les solveurs SAT
- DLIS : **nouvelle heuristique simplissime**
- SATO : **structure de donnée paresseuse** pour les clauses unitaires

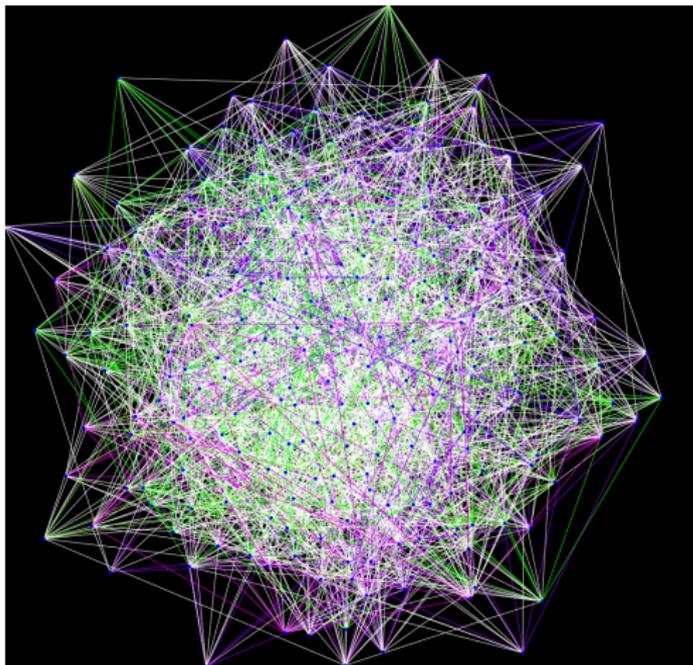
Les ingrédients de la révolution

BMC, GRASP, DLIS, SATO

Arrivée des problèmes gigantesques

Outil utilisé : satgraf <http://satbench.uwaterloo.ca/site/satgraf>

Instance aléatoire

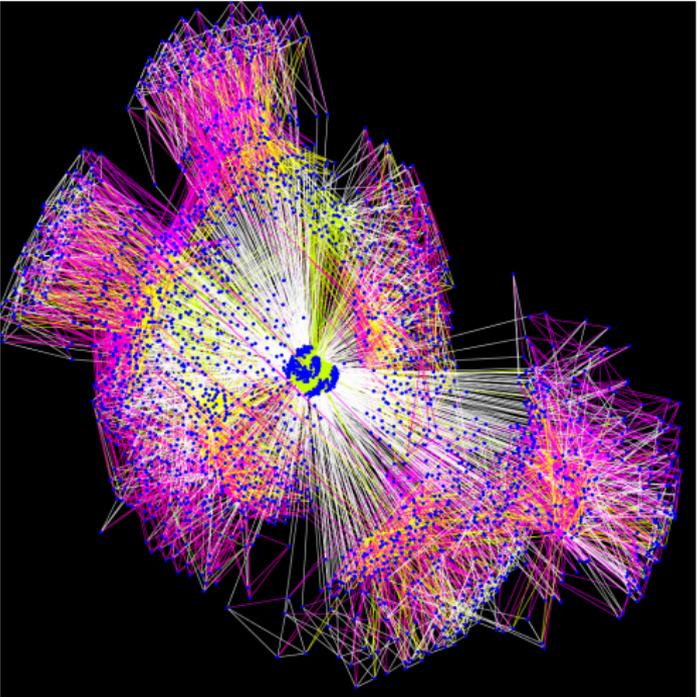




Arrivée des problèmes gigantesques

Outil utilisé : satgraf <http://satbench.uwaterloo.ca/site/satgraf>

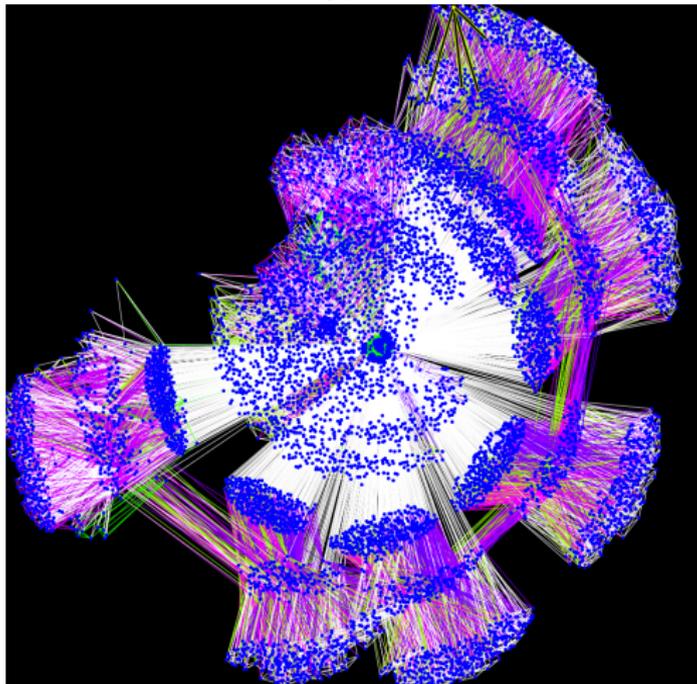
Instances Structurées de type Bounded Model Checking



Arrivée des problèmes gigantesques

Outil utilisé : satgraf <http://satbench.uwaterloo.ca/site/satgraf>

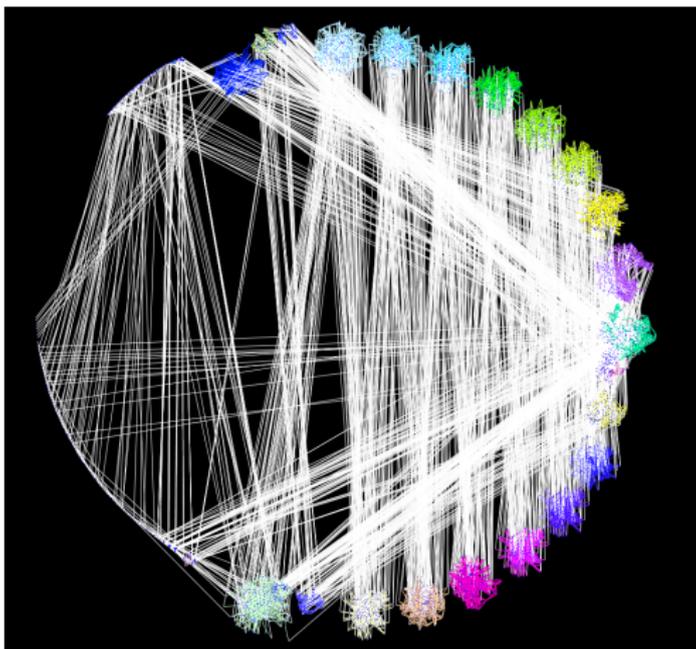
Instances Structurées de type Bounded Model Checking



Arrivée des problèmes gigantesques

Outil utilisé : satgraf <http://satbench.uwaterloo.ca/site/satgraf>

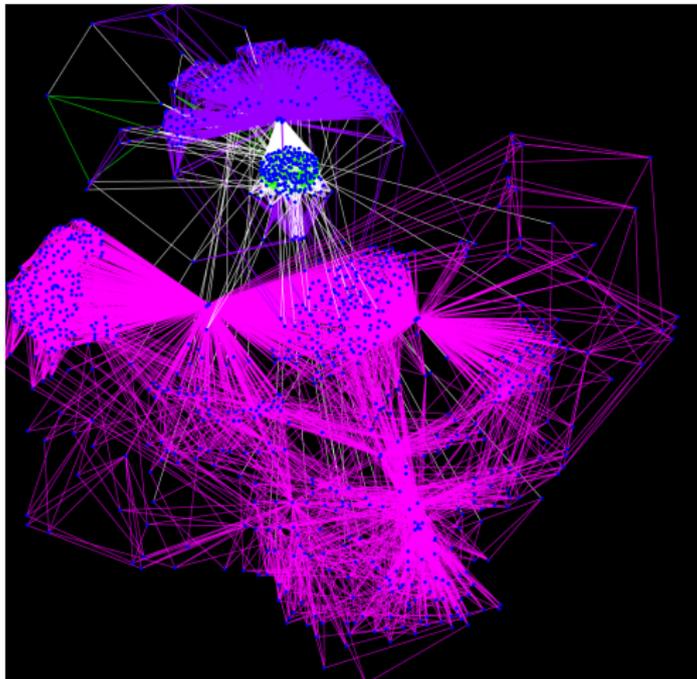
Instances Structurées de type Bounded Model Checking



Arrivée des problèmes gigantesques

Outil utilisé : satgraf <http://satbench.uwaterloo.ca/site/satgraf>

Instances Structurées de type Bounded Model Checking



Une structure de données révolutionne les algorithmes

Jusque là :

- Les heuristiques *miment* ce que l'on ferait
- la structure de donnée au service de l'algorithme

Article de référence

« *Chaff : Engineering an Efficient SAT Solver* » [Moskewicz & al. '01]

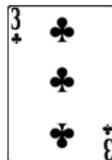
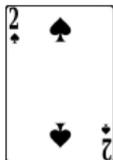
- ➔ Optimiser les algorithmes à base d'apprentissage de clauses

Priorité absolue : Détecter efficacement les propagations unitaires

Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes

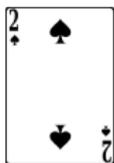


6

Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



6



Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



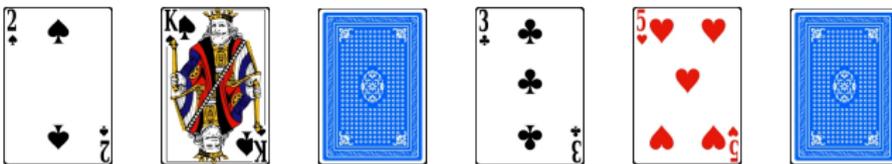
5



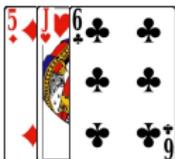
Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



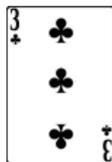
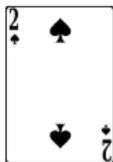
4



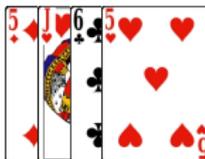
Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



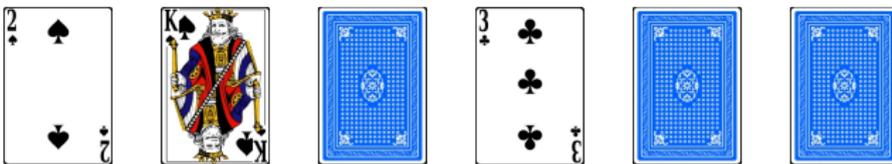
4



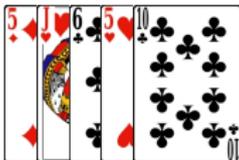
Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



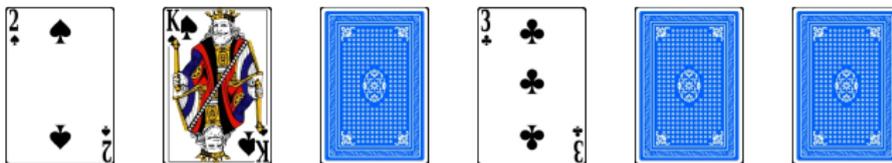
3



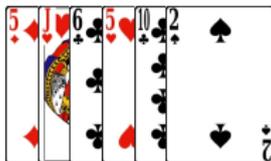
Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



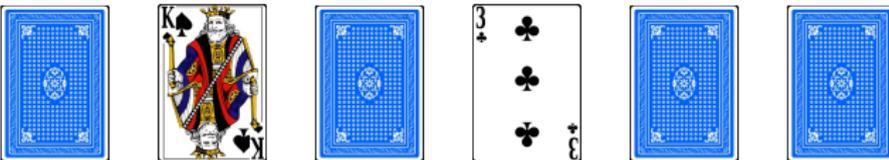
3



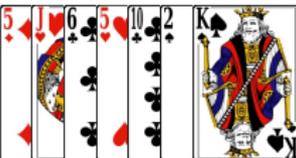
Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



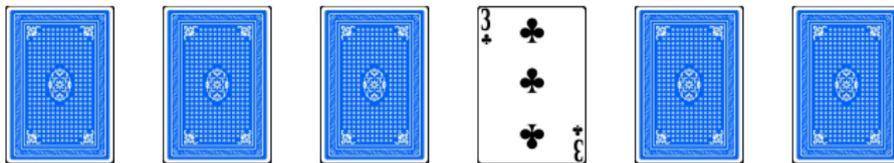
2



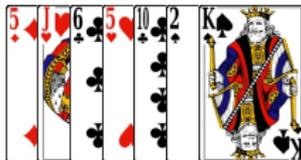
Jouons...à en faire le minimum

Jouer à détecter une main à 1 carte.

1. compter les cartes restantes



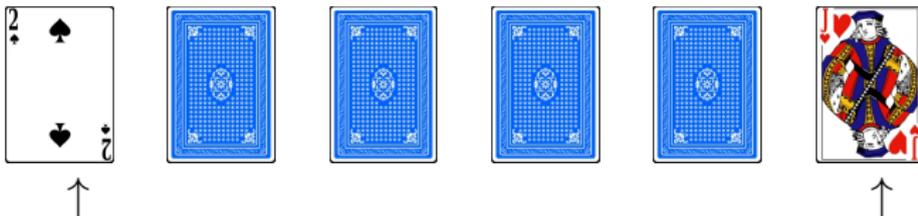
Clause Unitaire !!



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

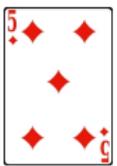
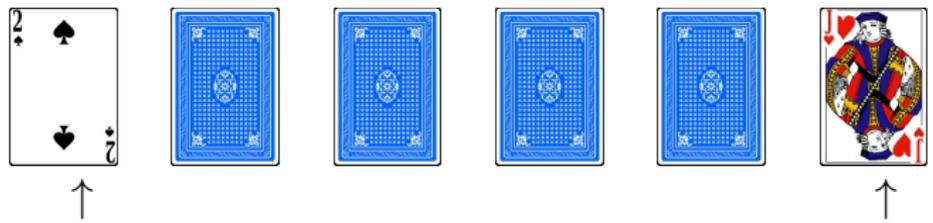
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

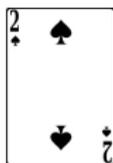
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

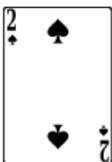
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

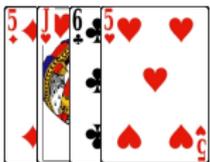
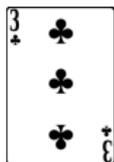
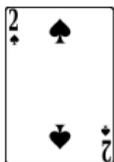
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

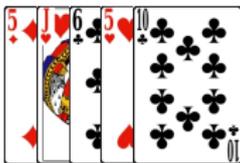
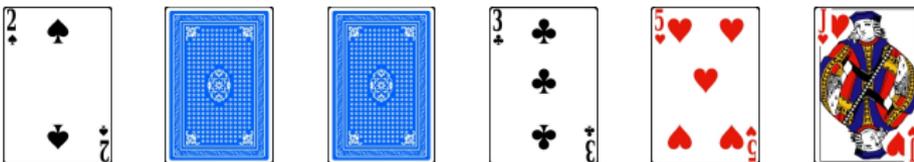
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

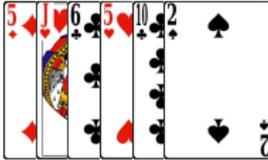
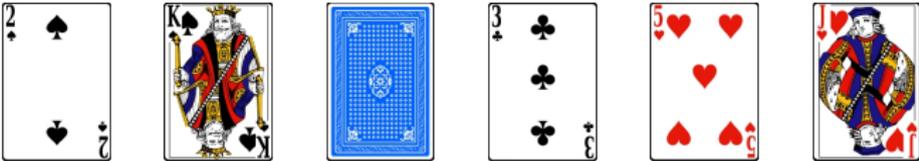
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

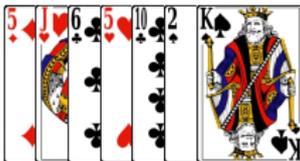
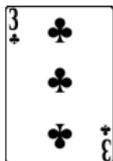
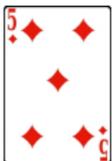
1. Surveiller les extrémités (SATO)



Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

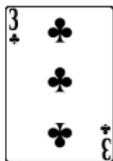
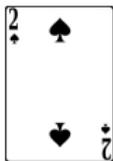
1. Surveiller les extrémités (SATO)



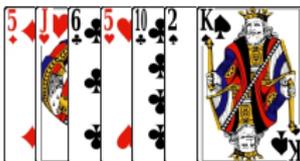
Jouons...à en faire le minimum – 2

Jouer à détecter une main à 1 carte.

1. Surveiller les extrémités (SATO)



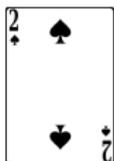
Clause Unitaire !!



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

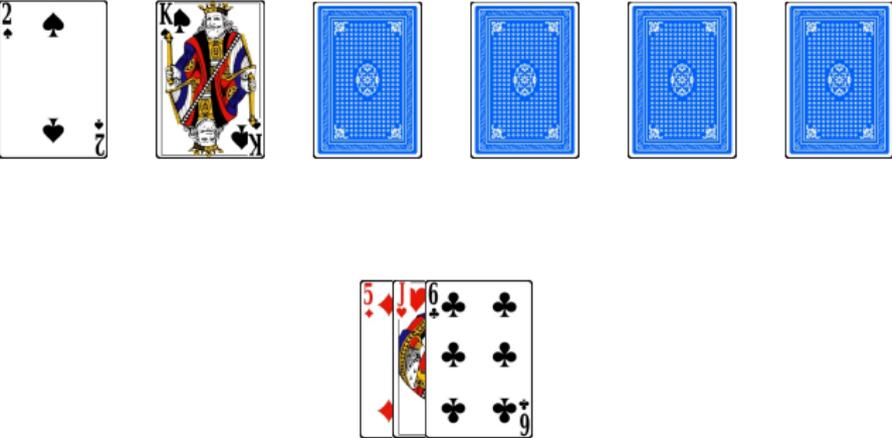
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

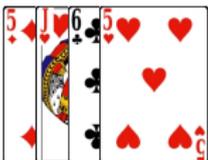
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

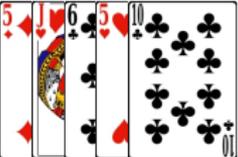
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

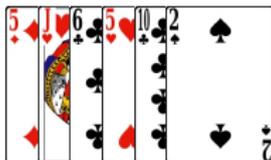
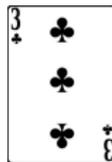
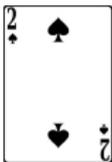
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

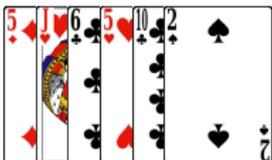
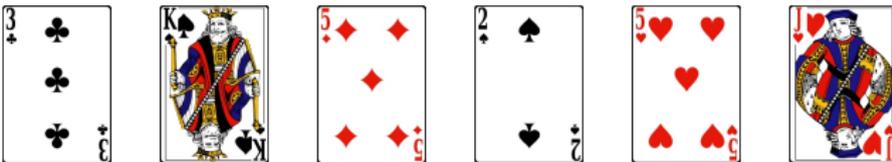
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

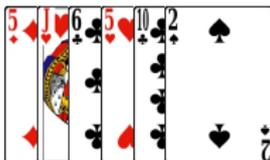
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

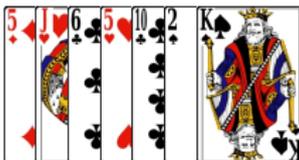
3. Garder deux témoins non appelés (Chaff)



Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

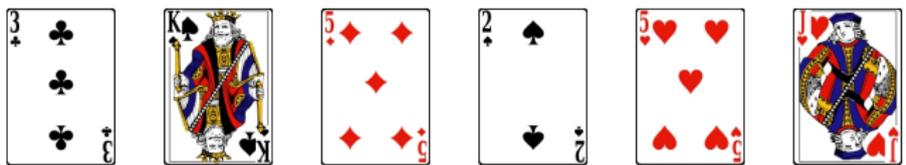
3. Garder deux témoins non appelés (Chaff)



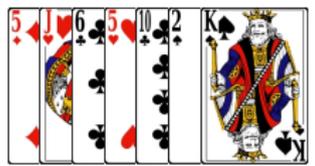
Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

3. Garder deux témoins non appelés (Chaff)



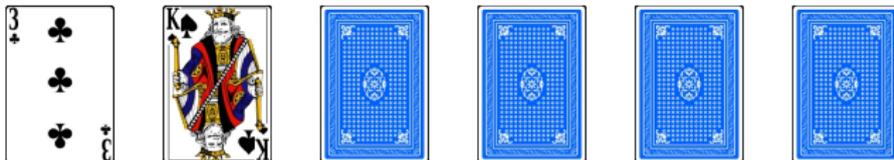
Clause Unitaire !!



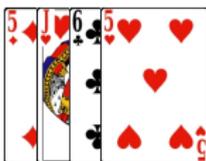
Jouons...à en faire le minimum – 3

Jouer à détecter une main à 1 carte.

3. Garder deux témoins non appelés (Chaff)



Rien à faire au retour arrière !!



Algorithmique au service de cette structure de données

Impossible de connaître l'état courant de la formule

- Combien de clauses réduites ? satisfaites ?
- Apparitions de variables pures ?

Seule garantie : toutes les clauses unitaires ou vides sont détectées.

Trouver un modèle : toutes les variables sont affectées sans conflit (on ne sait pas si toutes les clauses sont satisfaites).

Heuristique : Comment choisir une variable à l'aveugle ?

- On ne doit se baser que sur le passé
- Lien avec l'apprentissage de clauses basé sur les conflits

VSIDS : l'heuristique à battre

Variable State Independent Decaying Sum

Idée : L'heuristique est indépendante de l'état courant

On récompense les variables vues

- dans les dernières clauses apprises
- pendant les analyses de conflits les plus récents

Où brancher au début ?

L'heuristique est **hautement dynamique**

- Déclassement important d'une variable en 1/30s

Le solveur devient plus difficile à appréhender / comprendre / prédire

Graphe d'implication et heuristique

¬x₁

x₂

¬x₃

1

$$x_1 \vee x_2$$

$$\neg x_2 \vee \neg x_4 \vee \neg x_5 \vee x_7 \vee \neg x_6 \vee \neg x_8$$

$$x_{10} \vee \neg x_9 \vee x_{11}$$

$$\neg x_6 \vee x_{12} \vee x_{15}$$

$$\neg x_2 \vee \neg x_3$$

$$x_3 \vee x_5 \vee x_6$$

$$\neg x_4 \vee x_8 \vee x_9$$

$$\neg x_{11} \vee x_8 \vee \neg x_{12}$$

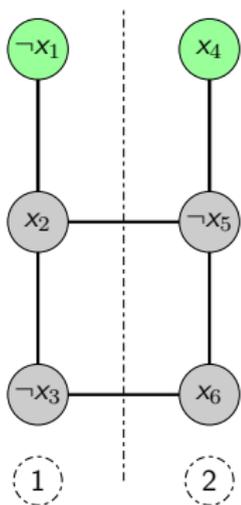
$$x_{13} \vee \neg x_{14} \vee \neg x_{16}$$

$$x_{12} \vee \neg x_{13}$$

$$\neg x_{15} \vee \neg x_{14} \vee x_{16}$$

$$x_7 \vee x_{12} \vee x_{14}$$

Graphe d'implication et heuristique



$$x_1 \vee x_2$$

$$\neg x_2 \vee \neg x_4 \vee \neg x_5 \vee x_7 \vee \neg x_6 \vee \neg x_8$$

$$x_{10} \vee \neg x_9 \vee x_{11}$$

$$\neg x_6 \vee x_{12} \vee x_{15}$$

$$\neg x_2 \vee \neg x_3$$

$$x_3 \vee x_5 \vee x_6$$

$$\neg x_4 \vee x_8 \vee x_9$$

$$\neg x_{11} \vee x_8 \vee \neg x_{12}$$

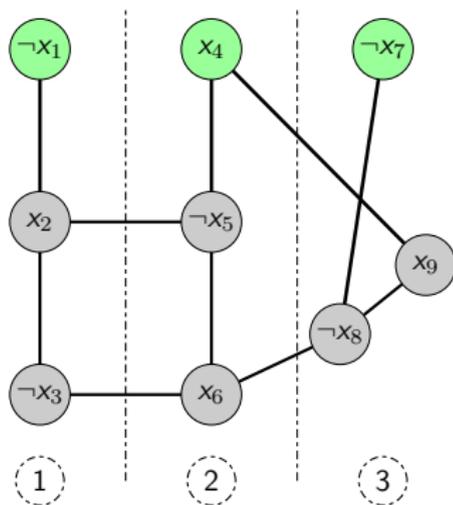
$$x_{13} \vee \neg x_{14} \vee \neg x_{16}$$

$$x_{12} \vee \neg x_{13}$$

$$\neg x_{15} \vee \neg x_{14} \vee x_{16}$$

$$x_7 \vee x_{12} \vee x_{14}$$

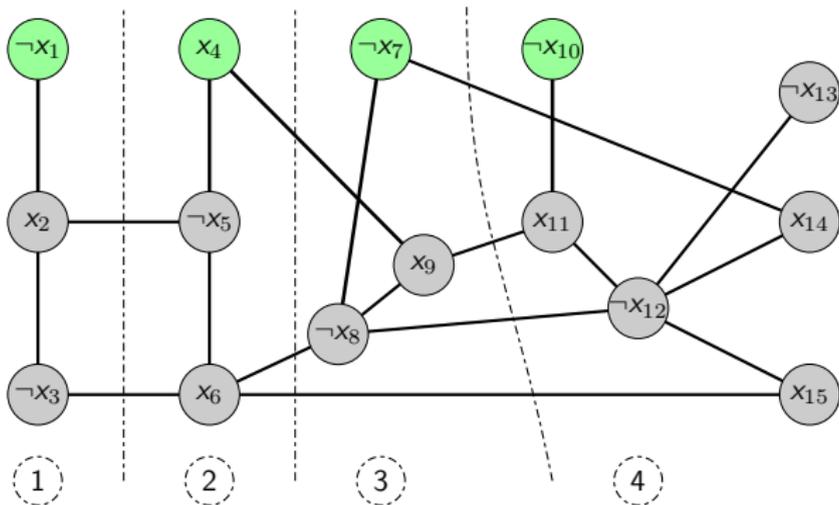
Graphe d'implication et heuristique



$$\begin{array}{llll}
 x_1 \vee x_2 & \neg x_2 \vee \neg x_4 \vee \neg x_5 & x_7 \vee \neg x_6 \vee \neg x_8 & x_{10} \vee \neg x_9 \vee x_{11} & \neg x_6 \vee x_{12} \vee x_{15} \\
 \neg x_2 \vee \neg x_3 & x_3 \vee x_5 \vee x_6 & \neg x_4 \vee x_8 \vee x_9 & \neg x_{11} \vee x_8 \vee \neg x_{12} & x_{13} \vee \neg x_{14} \vee \neg x_{16} \\
 & & & x_{12} \vee \neg x_{13} & \neg x_{15} \vee \neg x_{14} \vee x_{16}
 \end{array}$$

$$x_7 \vee x_{12} \vee x_{14}$$

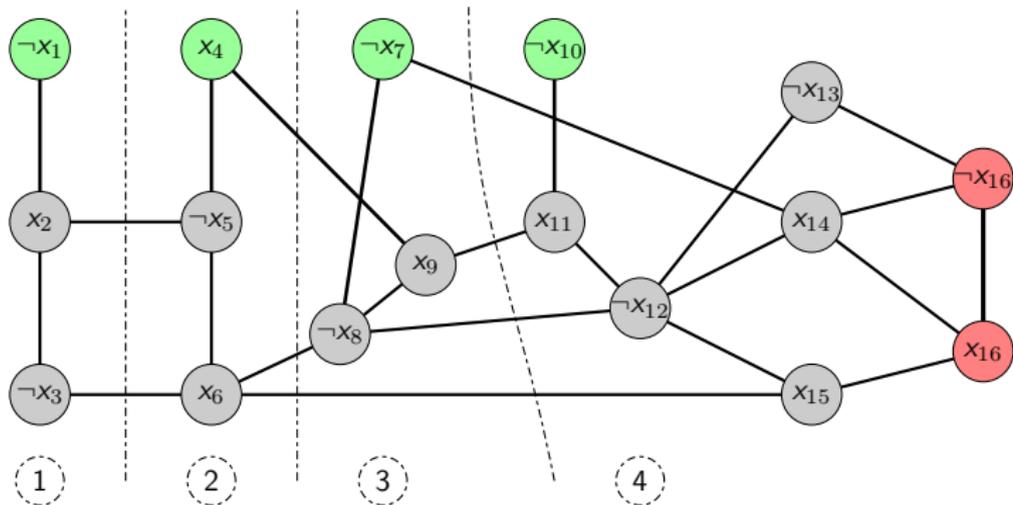
Graphe d'implication et heuristique



$$\begin{array}{llll}
 x_1 \vee x_2 & \neg x_2 \vee \neg x_4 \vee \neg x_5 & x_7 \vee \neg x_6 \vee \neg x_8 & x_{10} \vee \neg x_9 \vee x_{11} & \neg x_6 \vee x_{12} \vee x_{15} \\
 \neg x_2 \vee \neg x_3 & x_3 \vee x_5 \vee x_6 & \neg x_4 \vee x_8 \vee x_9 & \neg x_{11} \vee x_8 \vee \neg x_{12} & x_{13} \vee \neg x_{14} \vee \neg x_{16} \\
 & & & x_{12} \vee \neg x_{13} & \neg x_{15} \vee \neg x_{14} \vee x_{16}
 \end{array}$$

$$x_7 \vee x_{12} \vee x_{14}$$

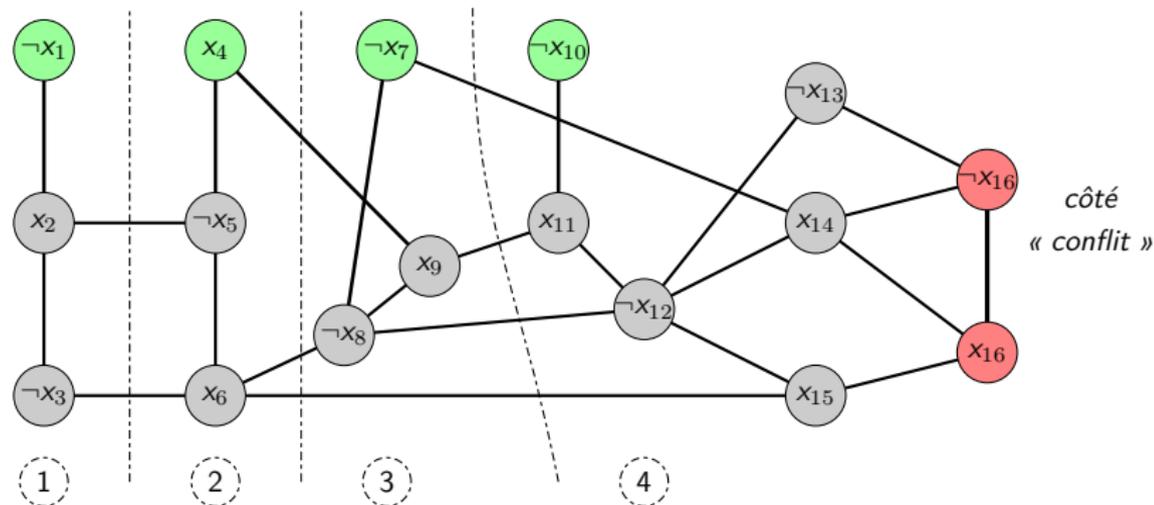
Graphe d'implication et heuristique



$x_1 \vee x_2$	$\neg x_2 \vee \neg x_4 \vee \neg x_5$	$x_7 \vee \neg x_6 \vee \neg x_8$	$x_{10} \vee \neg x_9 \vee x_{11}$	$\neg x_6 \vee x_{12} \vee x_{15}$
$\neg x_2 \vee \neg x_3$	$x_3 \vee x_5 \vee x_6$	$\neg x_4 \vee x_8 \vee x_9$	$\neg x_{11} \vee x_8 \vee \neg x_{12}$	$x_{13} \vee \neg x_{14} \vee \neg x_{16}$
			$x_{12} \vee \neg x_{13}$	$\neg x_{15} \vee \neg x_{14} \vee x_{16}$

$x_7 \vee x_{12} \vee x_{14}$

Graphe d'implication et heuristique



$$x_1 \vee x_2$$

$$\neg x_2 \vee \neg x_4 \vee \neg x_5 \vee x_7 \vee \neg x_6 \vee \neg x_8$$

$$x_{10} \vee \neg x_9 \vee x_{11}$$

$$\neg x_6 \vee x_{12} \vee x_{15}$$

$$\neg x_2 \vee \neg x_3$$

$$x_3 \vee x_5 \vee x_6$$

$$\neg x_4 \vee x_8 \vee x_9$$

$$\neg x_{11} \vee x_8 \vee \neg x_{12}$$

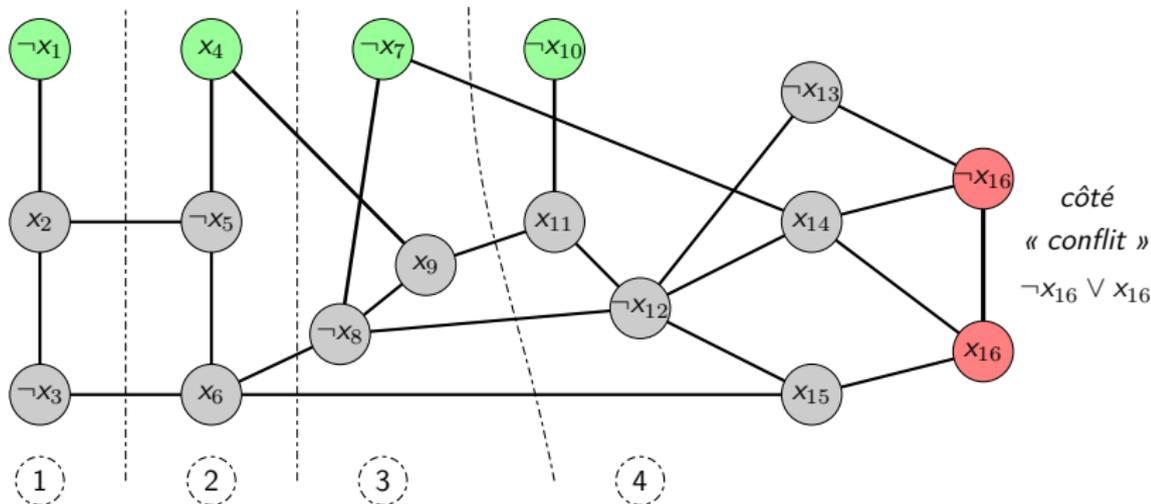
$$x_{13} \vee \neg x_{14} \vee \neg x_{16}$$

$$x_{12} \vee \neg x_{13}$$

$$\neg x_{15} \vee \neg x_{14} \vee x_{16}$$

$$x_7 \vee x_{12} \vee x_{14}$$

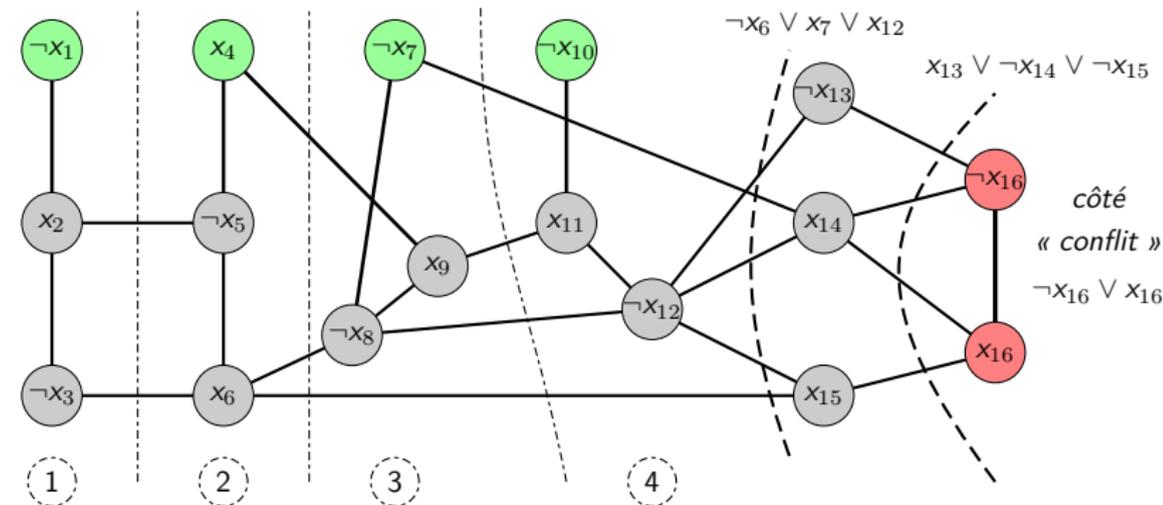
Graphe d'implication et heuristique



$x_1 \vee x_2$	$\neg x_2 \vee \neg x_4 \vee \neg x_5$	$x_7 \vee \neg x_6 \vee \neg x_8$	$x_{10} \vee \neg x_9 \vee x_{11}$	$\neg x_6 \vee x_{12} \vee x_{15}$
$\neg x_2 \vee \neg x_3$	$x_3 \vee x_5 \vee x_6$	$\neg x_4 \vee x_8 \vee x_9$	$\neg x_{11} \vee x_8 \vee \neg x_{12}$	$x_{13} \vee \neg x_{14} \vee \neg x_{16}$
			$x_{12} \vee \neg x_{13}$	$\neg x_{15} \vee \neg x_{14} \vee x_{16}$

$x_7 \vee x_{12} \vee x_{14}$

Graphe d'implication et heuristique



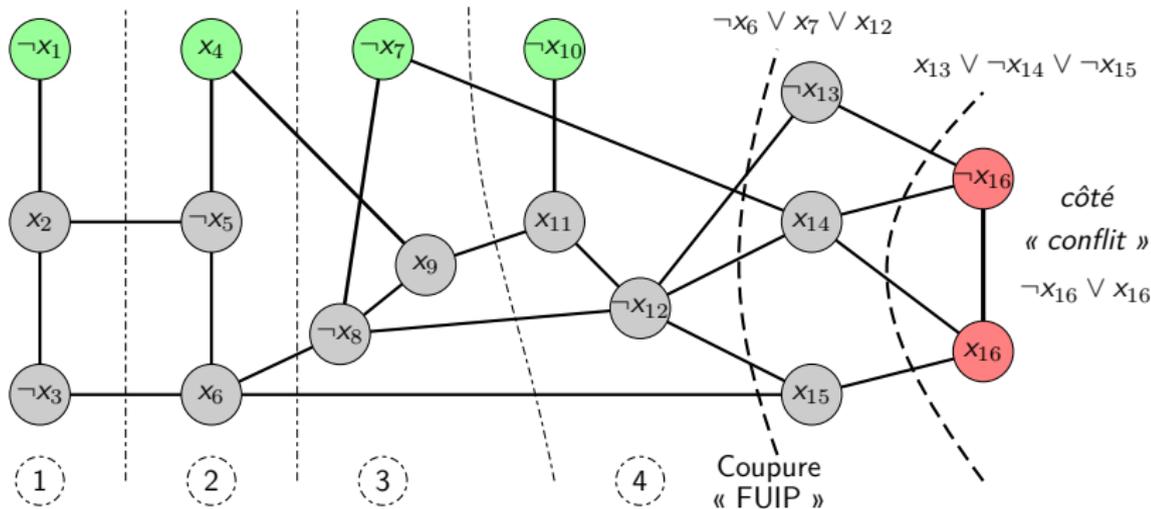
$$x_1 \vee x_2 \quad \neg x_2 \vee \neg x_4 \vee \neg x_5 \quad x_7 \vee \neg x_6 \vee \neg x_8 \quad x_{10} \vee \neg x_9 \vee x_{11} \quad \neg x_6 \vee x_{12} \vee x_{15}$$

$$\neg x_2 \vee \neg x_3 \quad x_3 \vee x_5 \vee x_6 \quad \neg x_4 \vee x_8 \vee x_9 \quad \neg x_{11} \vee x_8 \vee \neg x_{12} \quad x_{13} \vee \neg x_{14} \vee \neg x_{16}$$

$$x_{12} \vee \neg x_{13} \quad \neg x_{15} \vee \neg x_{14} \vee x_{16}$$

$$x_7 \vee x_{12} \vee x_{14}$$

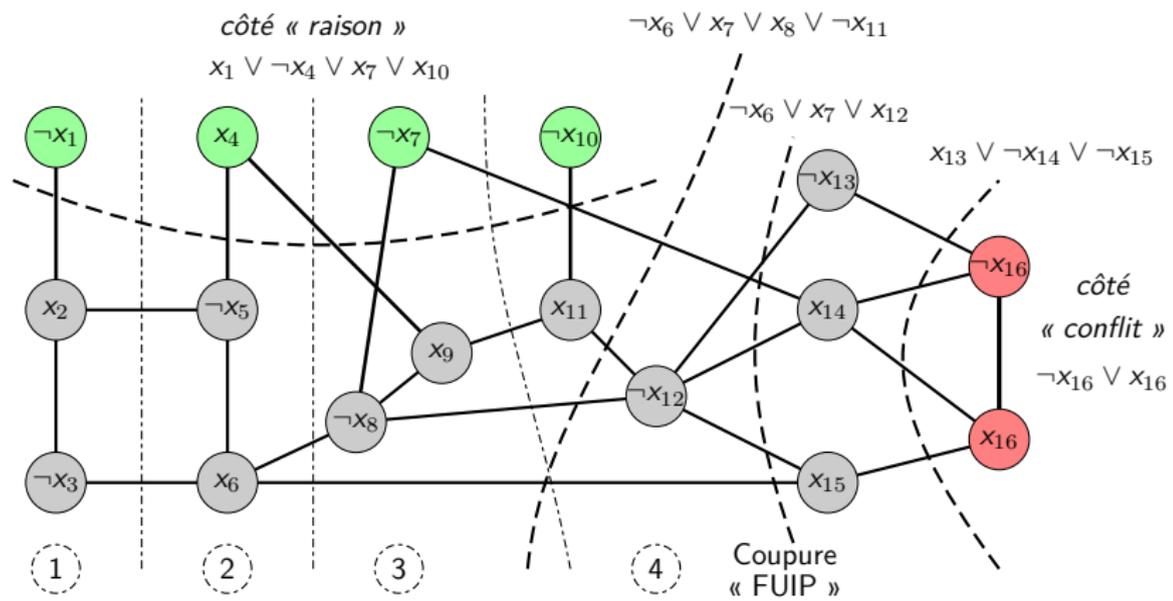
Graphe d'implication et heuristique



$x_1 \vee x_2$	$\neg x_2 \vee \neg x_4 \vee \neg x_5$	$x_7 \vee \neg x_6 \vee \neg x_8$	$x_{10} \vee \neg x_9 \vee x_{11}$	$\neg x_6 \vee x_{12} \vee x_{15}$
$\neg x_2 \vee \neg x_3$	$x_3 \vee x_5 \vee x_6$	$\neg x_4 \vee x_8 \vee x_9$	$\neg x_{11} \vee x_8 \vee \neg x_{12}$	$x_{13} \vee \neg x_{14} \vee \neg x_{16}$
			$x_{12} \vee \neg x_{13}$	$\neg x_{15} \vee \neg x_{14} \vee x_{16}$

$x_7 \vee x_{12} \vee x_{14}$

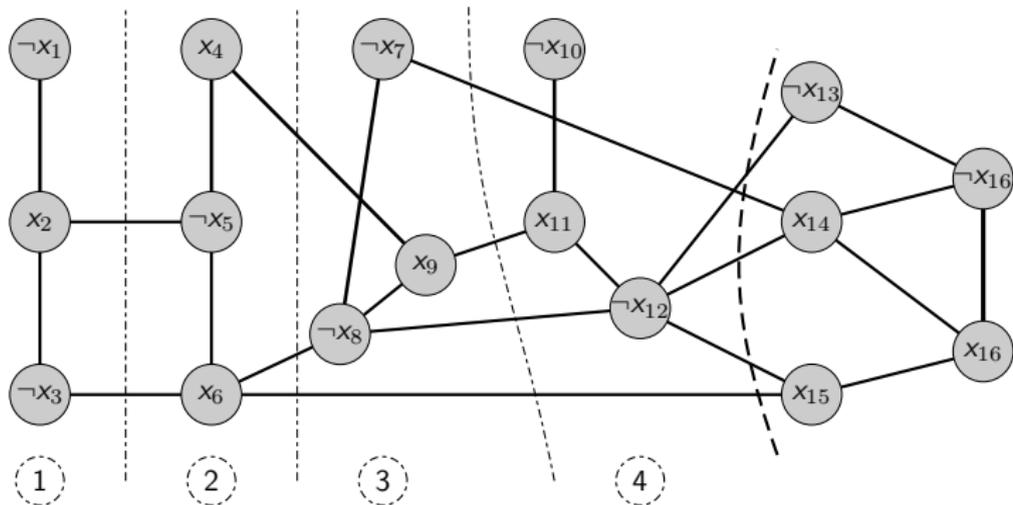
Graphe d'implication et heuristique



- $x_1 \vee x_2$
- $\neg x_2 \vee \neg x_4 \vee \neg x_5$
- $x_7 \vee \neg x_6 \vee \neg x_8$
- $x_{10} \vee \neg x_9 \vee x_{11}$
- $\neg x_6 \vee x_{12} \vee x_{15}$
- $\neg x_2 \vee \neg x_3$
- $x_3 \vee x_5 \vee x_6$
- $\neg x_4 \vee x_8 \vee x_9$
- $\neg x_{11} \vee x_8 \vee \neg x_{12}$
- $x_{13} \vee \neg x_{14} \vee \neg x_{16}$
- $x_{12} \vee \neg x_{13}$
- $\neg x_{15} \vee \neg x_{14} \vee x_{16}$

$x_7 \vee x_{12} \vee x_{14}$

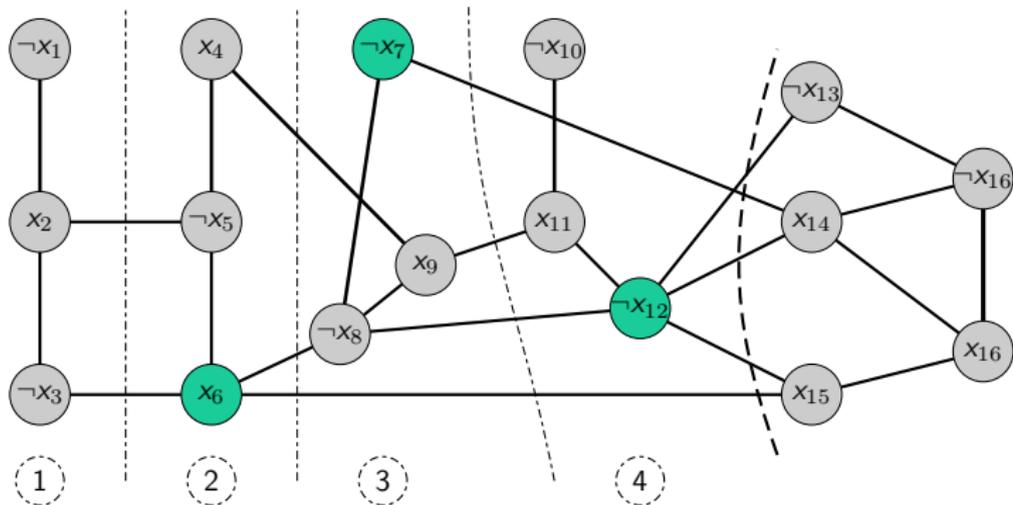
Graphe d'implication et heuristique



$x_1 \vee x_2$	$\neg x_2 \vee \neg x_4 \vee \neg x_5$	$x_7 \vee \neg x_6 \vee \neg x_8$	$x_{10} \vee \neg x_9 \vee x_{11}$	$\neg x_6 \vee x_{12} \vee x_{15}$
$\neg x_2 \vee \neg x_3$	$x_3 \vee x_5 \vee x_6$	$\neg x_4 \vee x_8 \vee x_9$	$\neg x_{11} \vee x_8 \vee \neg x_{12}$	$x_{13} \vee \neg x_{14} \vee \neg x_{16}$
			$x_{12} \vee \neg x_{13}$	$\neg x_{15} \vee \neg x_{14} \vee x_{16}$

$x_7 \vee x_{12} \vee x_{14}$

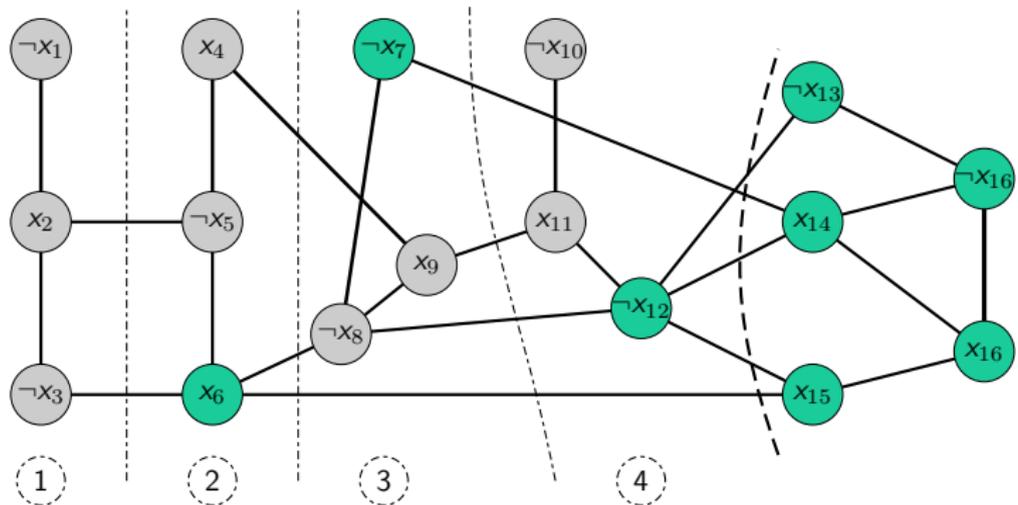
Graphe d'implication et heuristique



$x_1 \vee x_2$	$\neg x_2 \vee \neg x_4 \vee \neg x_5$	$x_7 \vee \neg x_6 \vee \neg x_8$	$x_{10} \vee \neg x_9 \vee x_{11}$	$\neg x_6 \vee x_{12} \vee x_{15}$
$\neg x_2 \vee \neg x_3$	$x_3 \vee x_5 \vee x_6$	$\neg x_4 \vee x_8 \vee x_9$	$\neg x_{11} \vee x_8 \vee \neg x_{12}$	$x_{13} \vee \neg x_{14} \vee \neg x_{16}$
			$x_{12} \vee \neg x_{13}$	$\neg x_{15} \vee \neg x_{14} \vee x_{16}$

$x_7 \vee x_{12} \vee x_{14}$

Graphe d'implication et heuristique



$$x_1 \vee x_2$$

$$\neg x_2 \vee \neg x_4 \vee \neg x_5 \quad x_7 \vee \neg x_6 \vee \neg x_8$$

$$x_{10} \vee \neg x_9 \vee x_{11}$$

$$\neg x_6 \vee x_{12} \vee x_{15}$$

$$\neg x_2 \vee \neg x_3$$

$$x_3 \vee x_5 \vee x_6$$

$$\neg x_4 \vee x_8 \vee x_9$$

$$\neg x_{11} \vee x_8 \vee \neg x_{12}$$

$$x_{13} \vee \neg x_{14} \vee \neg x_{16}$$

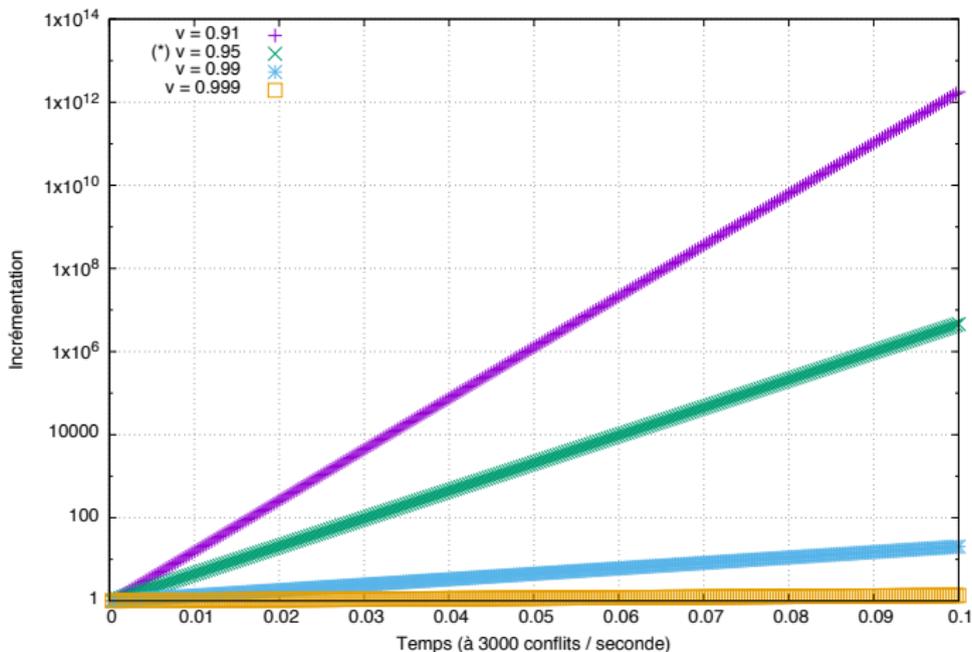
$$x_{12} \vee \neg x_{13}$$

$$\neg x_{15} \vee \neg x_{14} \vee x_{16}$$

$$x_7 \vee x_{12} \vee x_{14}$$

Dynamacité de l'heuristique

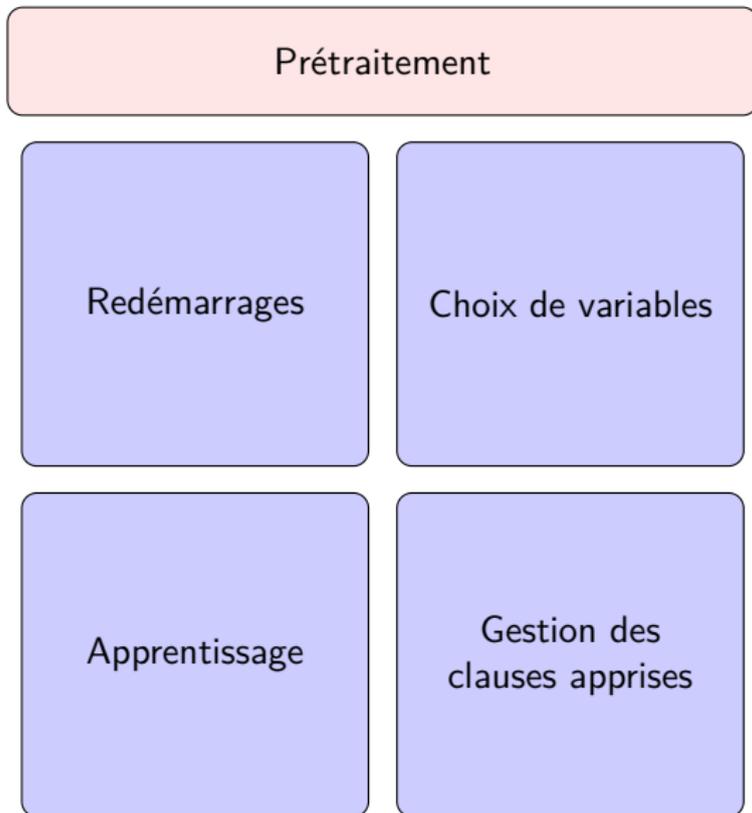
Croissance exponentielle de l'incrément



Après chaque conflit, incrément multiplié par $1/v$ ($v = 0.95$)



Ingrédients des CDCL



Ingrédients des CDCL – 2

Prétraitement

- Avant de plonger à l'aveugle, éliminer les cas triviaux
- Élimination des variables pures
- Simplification de la formule de départ

Gestion des clauses apprises

- Garder les clauses garantit la complétude
- Rapidement des problèmes de mémoire
- Régulièrement, les clauses apprises les moins *actives* sont effacées

Redémarrages

- On peut redémarrer la recherche sans tout recommencer
- Les clauses apprises permettent de ne pas refaire du travail



Plan du séminaire

Petite Histoire

2001-2005, la révolution CDCL

Compétitions et problèmes typiques

- Compétitions SAT
- Problématiques Classiques
- Problèmes typiques

2006-2016, vers des systèmes complexes

Problèmes ouverts

Une comparaison impartiale des solveurs

« *The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program* »

[Davis et al., 1962]

Un premier cycle de compétitions

- **1991** : Paderborn
- **1993** : Second DIMACS Challenge
- **1996** : Beijing Competition

Un nouveau cycle depuis 2002

- trois juges qui décident sur des classements anonymes
- des sous catégories (aléatoire, combinatoires, applications, ...)
- >100 solveurs / >40 équipes (2011)
- 10 ans de temps CPU (2011)

Impacts de la compétition

Accélérer le progrès

- Imposer la disponibilité des sources pour participer
- Identifier les problèmes les plus difficiles
- Les articles scientifiques se sont structurés autour de la compétition
 - Si un article ne compare pas sur les problèmes classiques, il y a un problème
 - Si un article se compare avec un solveur vieux de 4 ans, il y a un problème
- Hack track : Autoriser uniquement des modifications mineures sur un solveur de référence (Minisat, Glucose).

Accélère l'incorporation de nouvelles problématiques

- Temps plus long (jusque 10,000s)
- Certificats pour UNSAT
- Parallelisation, SAT Incrémental

Impacts de la compétition – 2

Fournir une réponse simple à l'extérieur de la communauté

- Quel est le meilleur solveur pour mon problème ?

Principale critique : un seul paradigme semble survivre

- Nouvelle partition des problèmes

Ne pas trop se prendre au sérieux

- Motiver les participants (étudiants, chercheurs)
- Rester léger : les gagnants sont chanceux

Problématiques classiques

Dès 1992 : « *Planning as Satisfiability* » [Kautz and Selman '92]

Problèmes de recherches de plans

- Encodage de l'état courant
- Encodage de déroulement du temps
- Encodage des actions possibles et de leurs effets

Model Checking Borné

$$I \wedge T_0 \wedge T_1 \wedge T_2 \wedge \dots \wedge T_k \wedge BUG_k$$

- Si UNSAT, *BUG* n'est pas accessible en *k* itérations
- Si SAT, on a trouvé un bug (et sa trace)

Model Checking non borné

$$I \wedge T_1 \wedge T_2 \wedge \dots \wedge T_k \wedge BUG_k$$

Comment garantir que BUG ne soit jamais accessible ?

Idée : trouver un invariant Inv t.q. BUG non atteignable en $k > 0$ étapes

- Inv caractérise un sur ensemble des états atteignables en j étapes :

$$I \wedge T_1 \wedge \dots \wedge T_j \rightarrow Inv$$

- Inv est inductif :

$$Inv \wedge T_1 \rightarrow Inv_1$$

- BUG n'est pas accessible depuis Inv en k étapes :

$$Inv \wedge T_1 \wedge T_2 \wedge \dots \wedge T_k \wedge BUG_k \equiv \perp$$

Model Checking non borné : Interpolants de Craig

$$\underbrace{I \wedge T_1 \wedge T_2}_A \wedge \underbrace{T_3 \wedge T_4 \wedge \dots \wedge T_k \wedge BUG_k}_B$$

Si on a $A \wedge B \equiv \perp$, alors il existe Itp un interpolant de Craig t.q. :

- $A \rightarrow Itp$
- $Itp \wedge B \equiv \perp$
- $Vars(Itp) \subseteq Vars(A) \cap Vars(B)$

Dès que l'on a $A \rightarrow \neg B$, il existe un C t.q. $A \rightarrow C$ et $C \rightarrow \neg B$

Lien avec les solveurs CDCL

- On peut extraire facilement un interpolant de la trace d'un CDCL
- Cet interpolant est lié à la preuve d'insatisfiabilité
- La construction de l'invariant nécessite de nombreux appels SAT

Model Checking non borné : Interpolants de Craig

$$\underbrace{I \wedge T_1 \wedge T_2}_A \rightarrow ltp, \quad ltp \rightarrow \underbrace{T_3 \wedge T_4 \wedge \dots \wedge T_k \wedge BUG_k}_B$$

Si on a $A \wedge B \equiv \perp$, alors il existe ltp un interpolant de Craig t.q. :

- $A \rightarrow ltp$
- $ltp \wedge B \equiv \perp$
- $Vars(ltp) \subseteq Vars(A) \cap Vars(B)$

Dès que l'on a $A \rightarrow \neg B$, il existe un C t.q. $A \rightarrow C$ et $C \rightarrow \neg B$

Lien avec les solveurs CDCL

- On peut extraire facilement un interpolant de la trace d'un CDCL
- Cet interpolant est lié à la preuve d'insatisfiabilité
- La construction de l'invariant nécessite de nombreux appels SAT

Divergence des sommes partielles (Conjecture Erdős 1932)

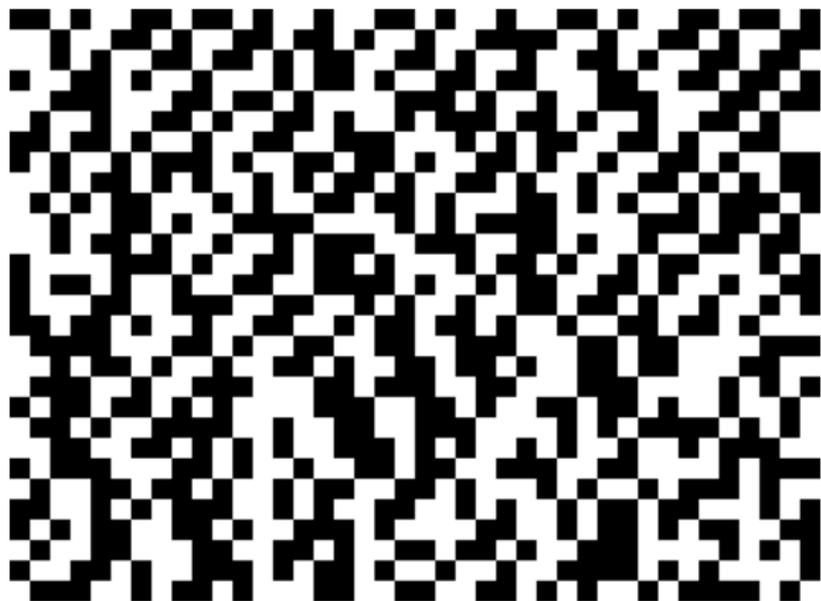
- Suite infinie de $+1$ et -1 : $\langle -1, 1, 1, -1, -1, 1, 1, \dots \rangle$
- $\forall C \exists k, d$ t.q. $|\sum_{i=1}^k x_{i,d}| \geq C$

$$\begin{array}{cccccccccccc|c}
 + & - & - & + & + & - & - & - & + & + & - & -1 \\
 & & & & + & - & - & & & + & & -1 \\
 & & & - & & - & & & + & & & -1 \\
 & & & & + & & - & & & & & 0 \\
 & & & & & + & & & + & & & +2
 \end{array}$$

- Conjecture *prouvée* en 2014 pour $C = 2$ ($k=1161$)
- La *preuve* : certificat UNSAT de Glucose (13 Go)³
- Conjecture démontrée par Terence Tao il y a quelques mois

3. cgi.csc.liv.ac.uk/~konev/SAT14/

Solution pour $C=2$, 1160 pas



(Pour $C=3$, toujours pas de maximum (solutions jusque 13900 pas!))

Autres applications

Cryptographie

- Retrouver une clé de cryptage / Inverser une fonction de hachage

Biologie

- Analyse de grands réseaux métaboliques
- Alignement de gènes

Vérification de logiciels

- Implémentation conforme aux spécifications
- Découverte d'invariants

Fouille de données

- Sur de petites bases

Problèmes d'optimisations

- Des milliers d'appels SAT pour résoudre un problème



Plan du séminaire

Petite Histoire

2001-2005, la révolution CDCL

Compétitions et problèmes typiques

2006-2016, vers des systèmes complexes

- SAT incrémental
- Comprendre les CDCL
- Glucose et la *Literal Block Distance*
- Structure des instances industrielles

Problèmes ouverts

Des milliers d'appels SAT

Extension de la problématique SAT

- La réponse Oui/Non n'est parfois pas suffisante
- Le déroulement du temps cache plusieurs appels SAT
- Optimiser peut aussi se faire par de multiples appels

Au lieu d'encoder tout cela en un gigantesque problème SAT

- Le solveur SAT est appelé de nombreuses fois (comme un oracle NP)
- Chaque appel se fait sur quasiment la même formule
 - Quelques clauses/variables peuvent être ajoutées (facile)
 - Quelques clauses/variables peuvent être retirées (plus difficile)

Ces applications sont de plus en plus importantes en pratique
(un prix spécial est apparu dans les dernières compétitions)

Travailler avec des hypothèses

- Une formule F
- Un ensemble d'hypothèses, $\ell_1, \ell_2, \dots, \ell_n$ (des littéraux)
- **Problème** : résoudre $F \wedge \ell_1 \wedge \ell_2 \dots \wedge \ell_n$
- **Résolution incrémentale** : Nombreux appels, hypothèses différentes

Première solution

- Simplifier $F' = F \wedge \ell_1 \wedge \ell_2 \dots \wedge \ell_n$
- Résoudre F'
- Les clauses apprises ne peuvent pas être gardées

Travailler avec des hypothèses

- Une formule F
- Un ensemble d'hypothèses, l_1, l_2, \dots, l_n (des littéraux)
- **Problème** : résoudre $F \wedge l_1 \wedge l_2 \dots \wedge l_n$
- **Résolution incrémentale** : **Nombreux appels, hypothèses différentes**

Seconde Solution

- Considérer les hypothèses comme des variables de décision : le solveur branchera d'abord sur elles
- Lancer le solveur SAT (presque) comme d'habitude
- **Toutes les clauses apprises peuvent être gardées**
- La non satisfiabilité s'exprimera par rapport aux hypothèses

Oublier des clauses en SAT incrémental

- Assigner ℓ_i (hypothèse) à **faux** pour **activer** la clause i
- Assigner ℓ_j (hypothèse) à **vrai** pour **désactiver** la clause j
- Toutes les clauses apprises dépendantes de la clause j seront désactivées du même coup

$$\ell_1 \vee x \vee y \vee z$$

$$\ell_2 \vee x \vee \neg y$$

$$\ell_3 \vee x \vee \neg z$$

$$\ell_4 \vee \neg x \vee y \vee z$$

$$\ell_5 \vee x \vee w$$

$$\ell_6 \vee w \vee z \vee \neg y$$

$$\ell_7 \vee \neg x \vee \neg y$$

$$\ell_8 \vee \neg x \vee \neg z$$

$$\ell_9 \vee w \vee \neg x \vee \neg z$$

$$\ell_1 \vee \ell_2 \vee \vee x \vee z$$

Oublier des clauses en SAT incrémental

- Assigner ℓ_i (hypothèse) à **faux** pour **activer** la clause i
- Assigner ℓ_j (hypothèse) à **vrai** pour **désactiver** la clause j
- Toutes les clauses apprises dépendantes de la clause j seront désactivées du même coup

$$\ell_1 \vee x \vee y \vee z$$

$$\ell_2 \vee x \vee \neg y$$

$$\ell_3 \vee x \vee \neg z$$

$$\ell_4 \vee \neg x \vee y \vee z$$

$$\ell_5 \vee x \vee w$$

$$\ell_6 \vee w \vee z \vee \neg y$$

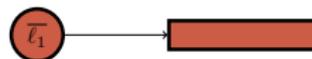
$$\ell_7 \vee \neg x \vee \neg y$$

$$\ell_8 \vee \neg x \vee \neg z$$

$$\ell_9 \vee w \vee \neg x \vee \neg z$$

$$\ell_1 \vee \ell_2 \vee \vee x \vee z$$

Niv 1



Oublier des clauses en SAT incrémental

- Assigner ℓ_i (hypothèse) à **faux** pour **activer** la clause i
- Assigner ℓ_j (hypothèse) à **vrai** pour **désactiver** la clause j
- Toutes les clauses apprises dépendantes de la clause j seront désactivées du même coup

$$\ell_1 \vee x \vee y \vee z$$

$$\ell_2 \vee x \vee \neg y$$

$$\ell_3 \vee x \vee \neg z$$

$$\ell_4 \vee \neg x \vee y \vee z$$

$$\ell_5 \vee x \vee w$$

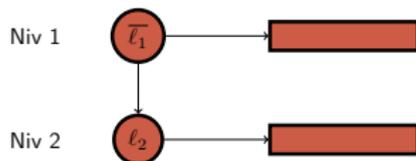
$$\ell_6 \vee w \vee z \vee \neg y$$

$$\ell_7 \vee \neg x \vee \neg y$$

$$\ell_8 \vee \neg x \vee \neg z$$

$$\ell_9 \vee w \vee \neg x \vee \neg z$$

$$\ell_1 \vee \ell_2 \vee \vee x \vee z$$





Les CDCL sont des systèmes complexes

“ Comprendre les algorithmes que l'on a écrit ”

Difficile de comprendre les CDCL :

- Extrêmement rapides et imprédictibles
- Fonctionnent sur les problèmes du monde réel
 - ➔ Comment définir leur caractéristiques ?
- Tous les composants sont intimement connectés
 - ➔ effets de bord partout !
- Il n'y a pas une seule raison expliquant leurs performances
 - ➔ Au moins **nous savons que nous ne savons pas**

Idée de glucose

Une étude expérimentale des solveurs CDCL

Les CDCL sont des systèmes complexes – Illustration

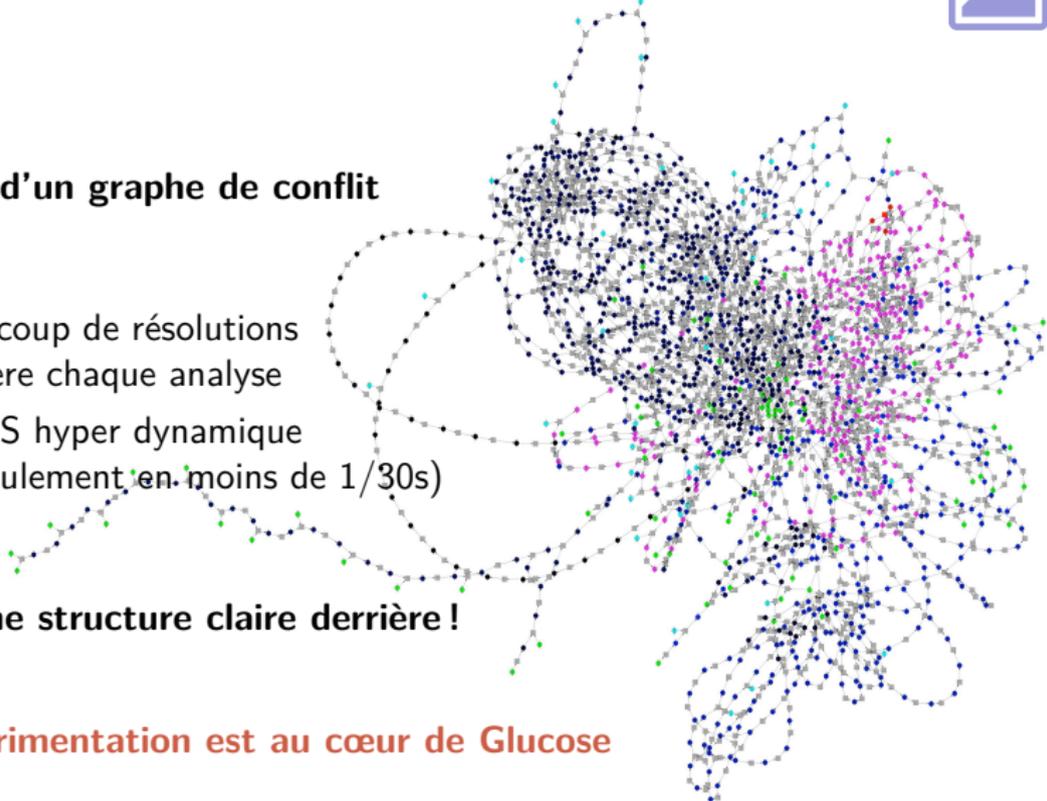


Exemple d'un graphe de conflit

- Beaucoup de résolutions derrière chaque analyse
- VSIDS hyper dynamique (basculement en moins de 1/30s)

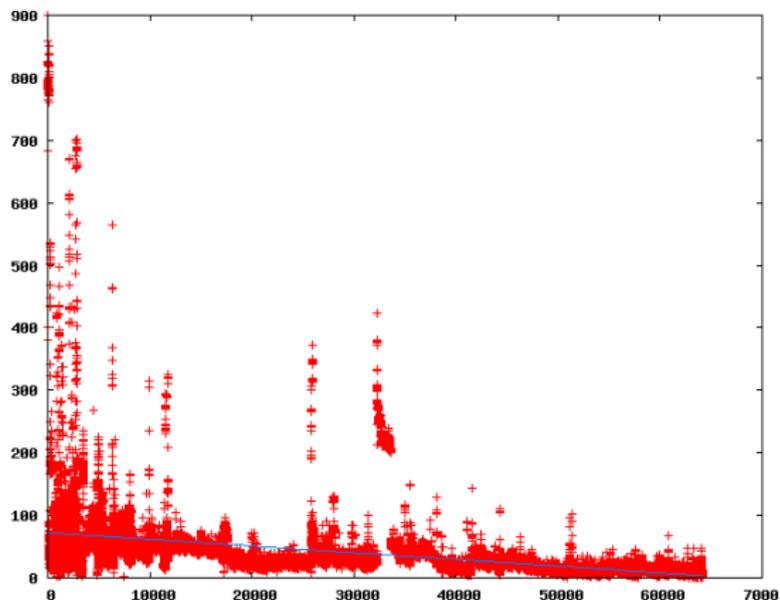
Mais : une structure claire derrière !

L'expérimentation est au cœur de Glucose



Nombre de décisions avant un conflit

een-pico-prop05-50 – UNSAT – 13,000 vars and 65,000 clauses



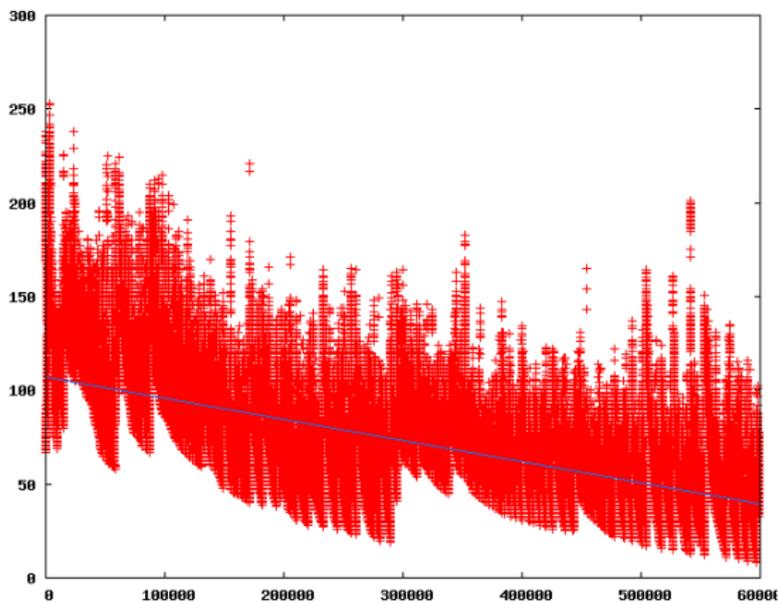
een-pico-prop05-50 – UNSAT – 13,000 vars and 65,000 clauses

Une bonne clause apprise doit réduire le nombre de décisions

Observation vérifiée pour la plupart des instances satisfiables !

Nombre de décisions avant un conflit

grieu-vmc-s05-25 – sat – 625 vars and 76,000 clauses



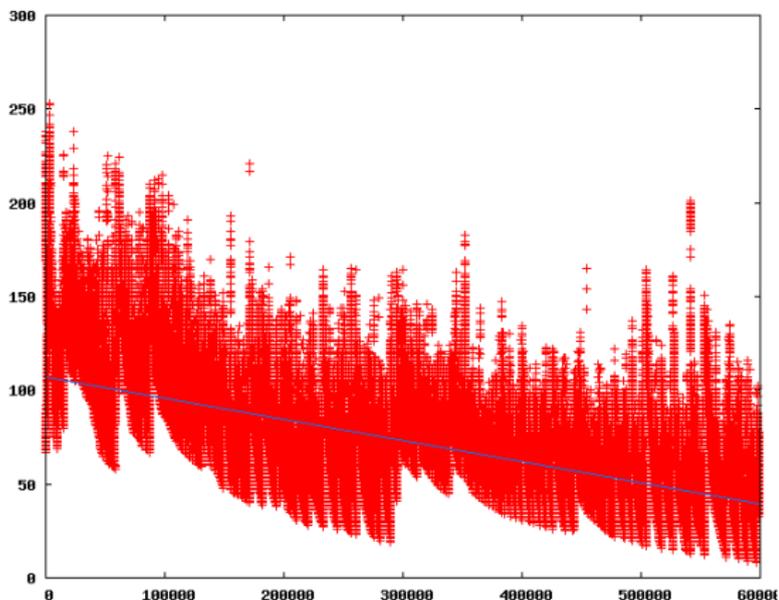
grieu-vmc-s05-25 – SAT – 625 vars and 76,000 clauses

Une bonne clause apprise doit réduire le nombre de décisions

Observation vérifiée pour la plupart des instances satisfiables !

Nombre de décisions avant un conflit

grieu-vmc-s05-25 – sat – 625 vars and 76,000 clauses



grieu-vmc-s05-25 – SAT – 625 vars and 76,000 clauses

Une bonne clause apprise doit réduire le nombre de décisions

Observation vérifiée pour la plupart des instances satisfiables !



Glucose : les blocs de propagations (LBD)

- Une décision déclenche souvent de nombreuses propagations (*blocs de propagations*)
- Ces littéraux seront souvent propagés ensemble (*hypothèse*)
- **Réduire les décisions ?**
 - ➔ Ajouter des dépendances fortes entre ces blocs

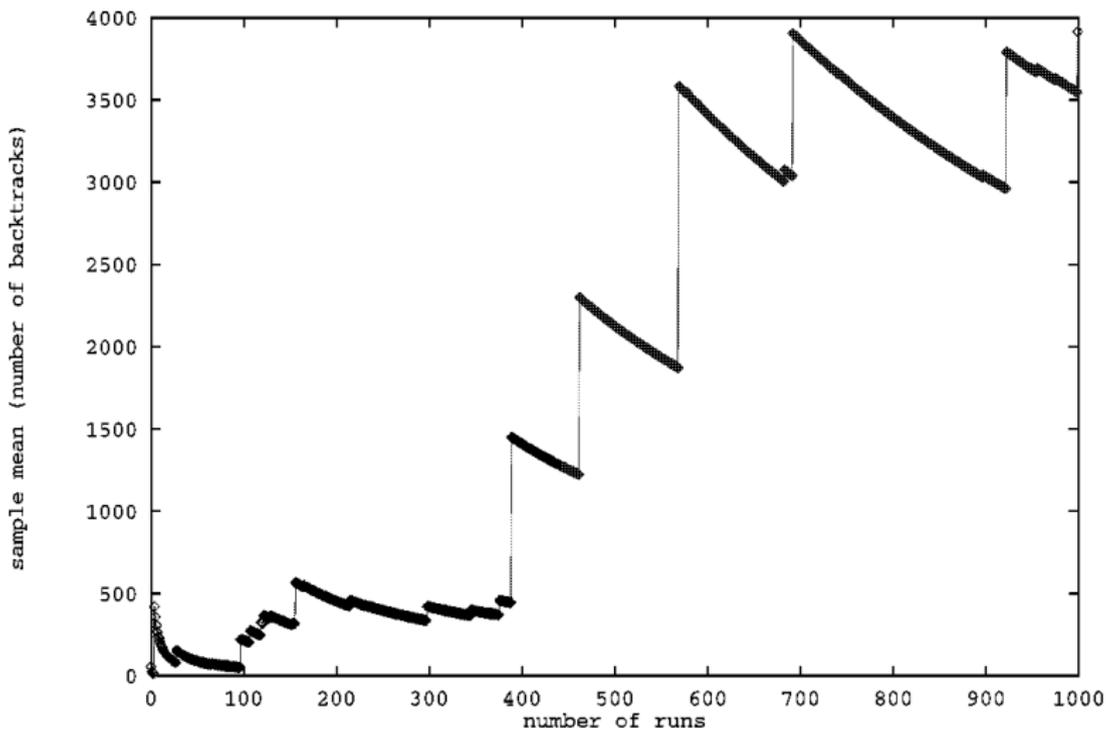
Score LBD d'une clause apprise :
nombre de niveaux de décision la composant

- Les petits LBD sont meilleurs
- Les clauses « *Glues* » sont fondamentales (LBD=2)
 - Un seul littéral du dernier niveau de décision (littéral FUIP)
 - Ce littéral sera **collé** à l'autre bloc
 - Gardées pour toujours dans glucose
- Oublie agressif (95% des clauses apprises)



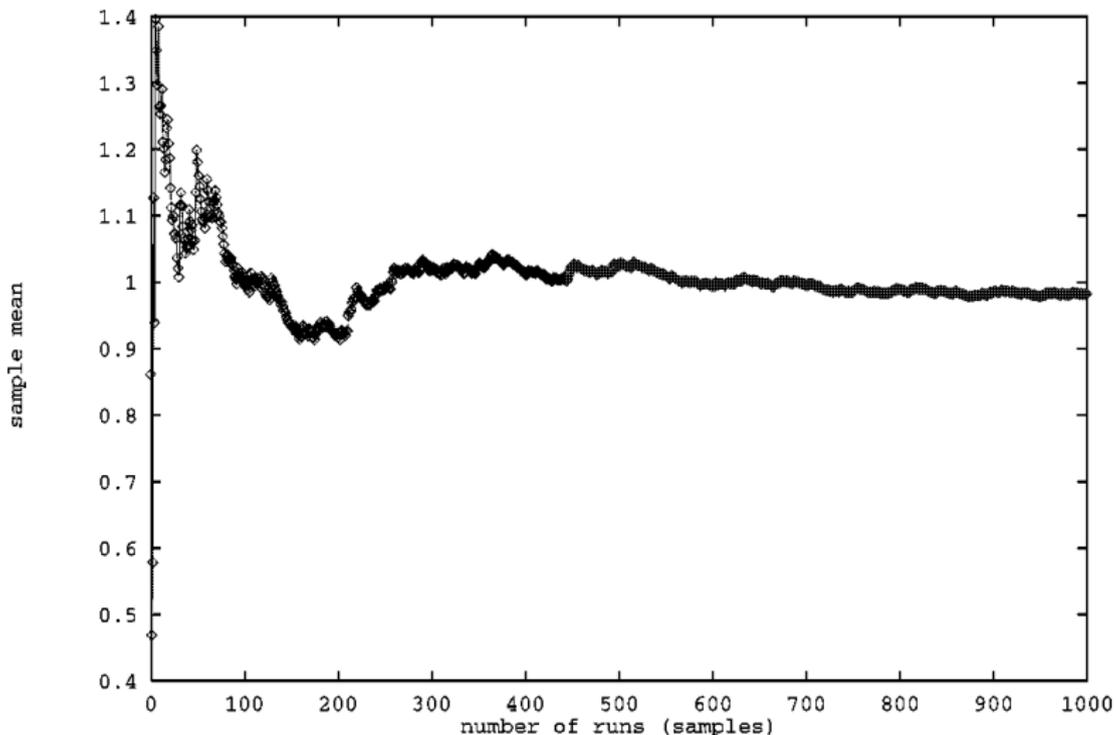
Redémarrages : d'abord un phénomène observé

Évolution de la moyenne des temps CPU sur plusieurs lancements



Redémarrages : d'abord un phénomène observé

Stabilité de la médiane des temps CPU sur plusieurs lancements



Redémarrages de plus en plus rapides

(Luby et autres)

Casser le phénomène de *longue traine* : **redémarrer** régulièrement

Minisat : suite géométrique de raison 2 (100, 200, 400, 800, ...)

Loi de redémarrage efficace : multiple de la suite de Luby :

1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 4, 8, 1, ...

L'optimal est un multiple de 6 (6, 6, 12, 6, 6, 12, 24, ...)

Dans Glucose

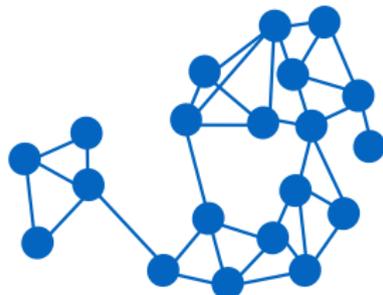
- Tout Glucose est tourné vers les LBD
- Les CDCL sont considérés comme des producteurs de clauses
- Si la qualité des clauses récentes n'est pas bonne, on redémarre



Structure en communautés

Idée centrale :

- Connexions internes denses dans le groupe
- Connexions moins denses à l'extérieur du groupe



[Ansótegui, Girádez-Cru, Levy'12] :

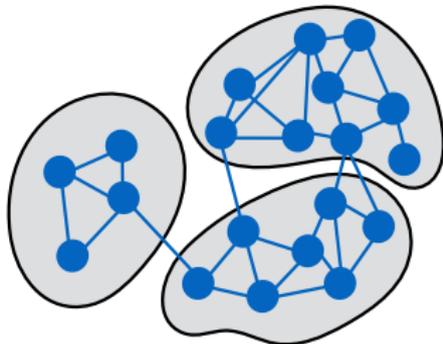
- **Les instances industrielles ont une forte structuration en communautés**
- L'apprentissage ne les détruit pas totalement



Structure en communautés

Idée centrale :

- Connexions internes denses dans le groupe
- Connexions moins denses à l'extérieur du groupe



[Ansótegui, Girádez-Cru, Levy'12] :

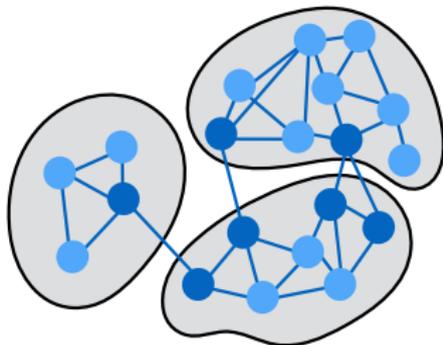
- **Les instances industrielles ont une forte structuration en communautés**
- L'apprentissage ne les détruit pas totalement



Structure en communautés

Idée centrale :

- Connexions internes denses dans le groupe
- Connexions moins denses à l'extérieur du groupe



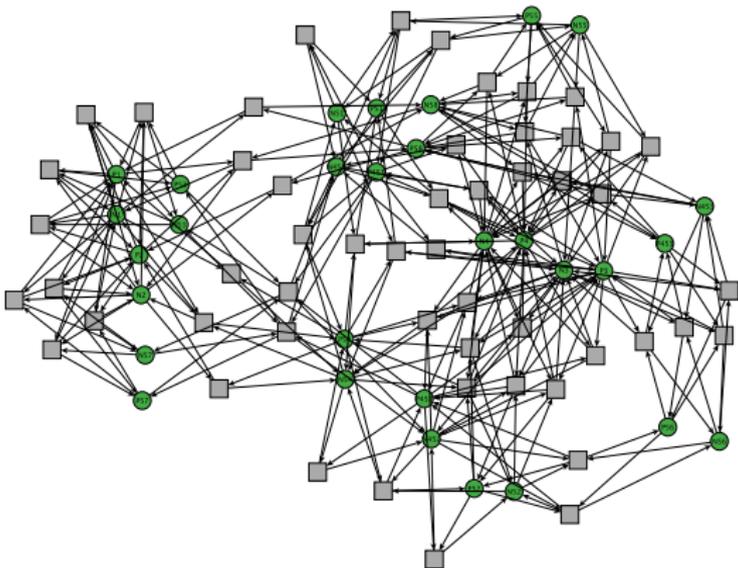
[Ansótegui, Girádez-Cru, Levy'12] :

- **Les instances industrielles ont une forte structuration en communautés**
- L'apprentissage ne les détruit pas totalement

Autre mesure : la centralité des variables

On se base aussi sur une représentation sous forme de graphe de la formule initiale.

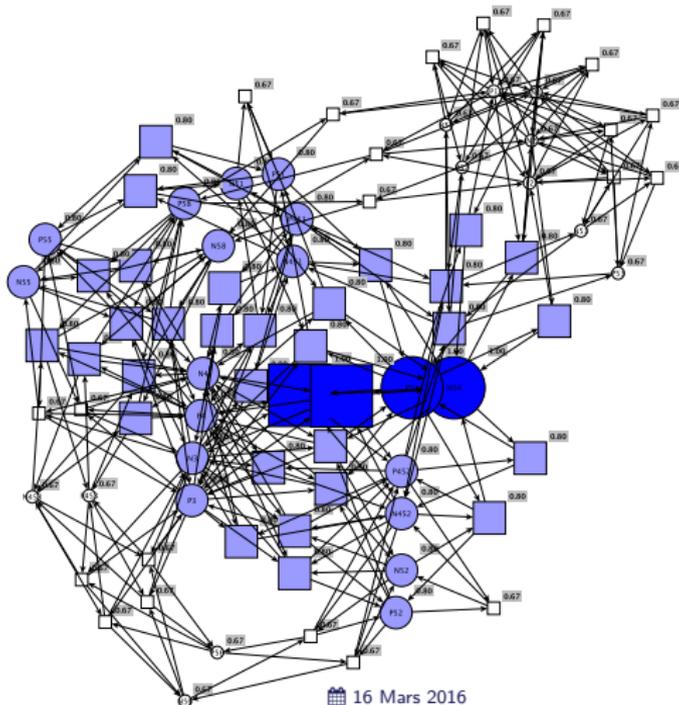
➔ La centralité d'une variable exprime son importance



Autre mesure : la centralité des variables

On se base aussi sur une représentation sous forme de graphe de la formule initiale.

➔ La centralité d'une variable exprime son importance



Centralité et communautés

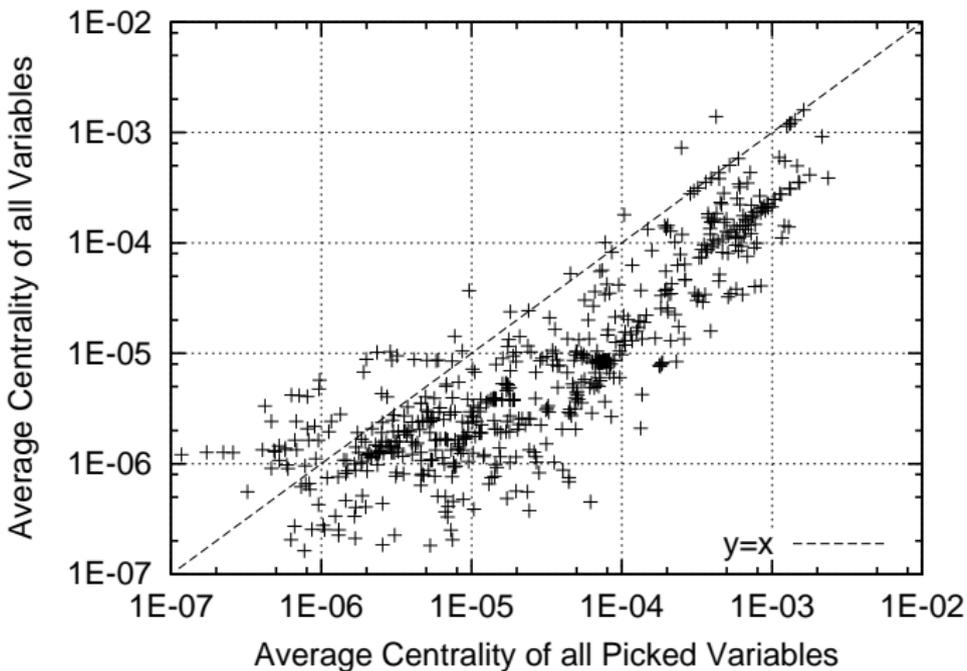
Une expérimentation pour comprendre les CDCL

- Mesurer le centralité de chaque variable (formule initiale)
- Observer le solveur CDCL
- Peut on établir un lien entre centralité et
 - Choix des variables (VSIDS) ?
 - Apprentissage ?

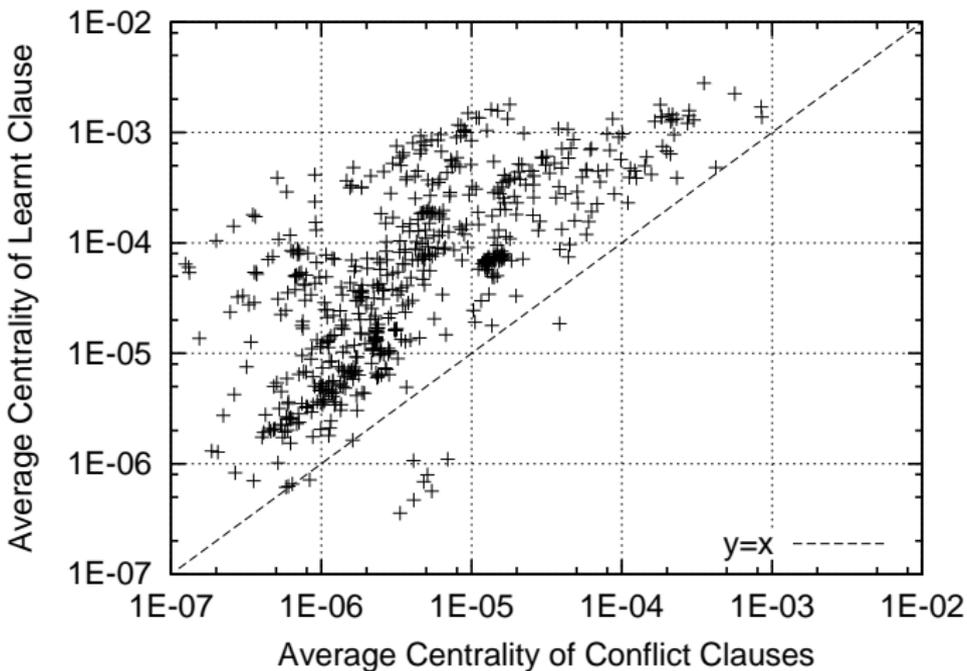
La centralité est liée aux communautés

- Nœuds centraux : ceux qu'il faut retirer pour partitionner le graphe au mieux
- Nœuds centraux : devraient se trouver aux frontières entre communautés
- Nœuds non centraux : plutôt à l'intérieur des communautés

Centralité des variables choisies par VSIDS



Centralité des clauses conflits / clauses apprises





Lien entre LBD et communautés

Expérimentation :

- Partitionnement des variables en communautés (formule initiale)
- À chaque clause apprise, observer la différence entre
 - Le nombre de communautés dans la clause
 - Le nombre de niveaux de décisions (LBD) dans la clause

Un autre bon exemple (on en trouve beaucoup)



Benchmarks

aaai10-plan...-
ipc5-...-17-
step20

Nb Variables

50277

Nb Clauses

283903

Max conflicts

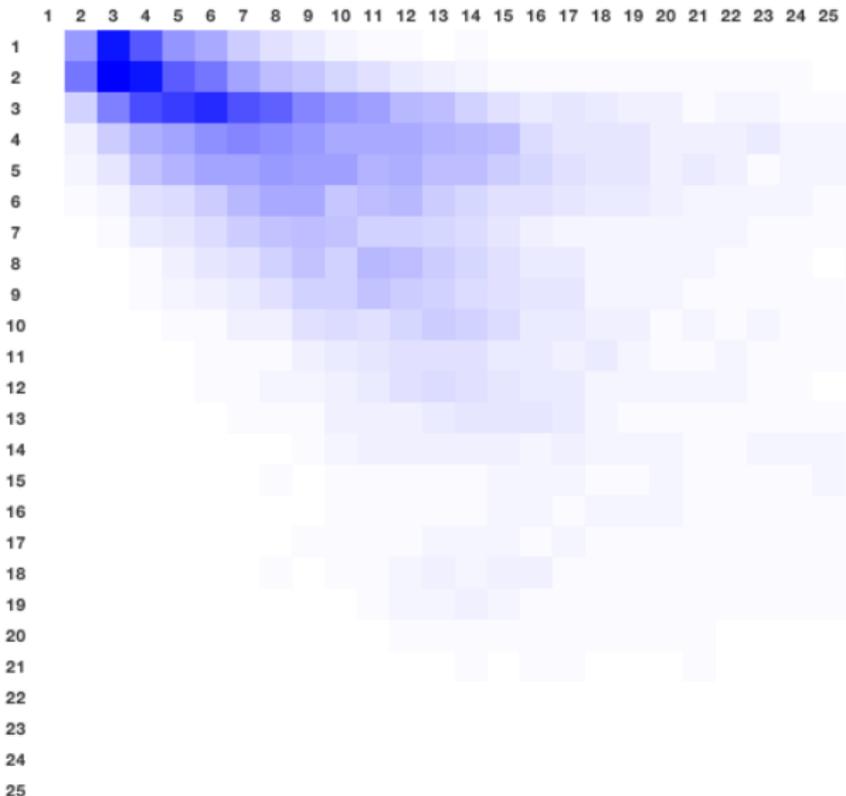
20000

Q

0.913337

Nb Partitions

12776



Un exemple moins bon (il y en a)



Benchmarks

rbcl_xits_14_SAT

Nb Variables

2220

Nb Clauses

148488

Max conflicts

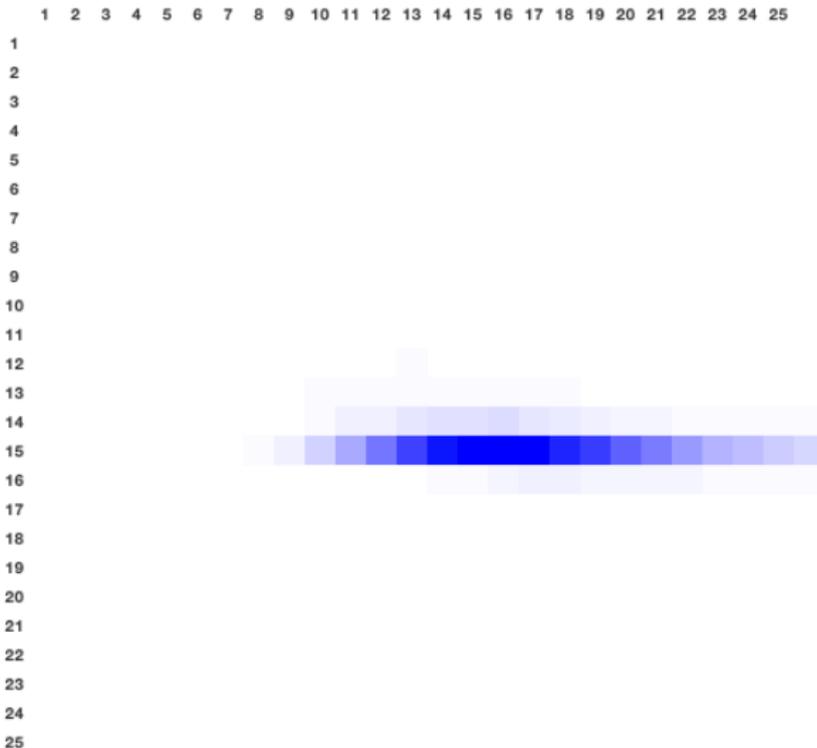
20000

Q

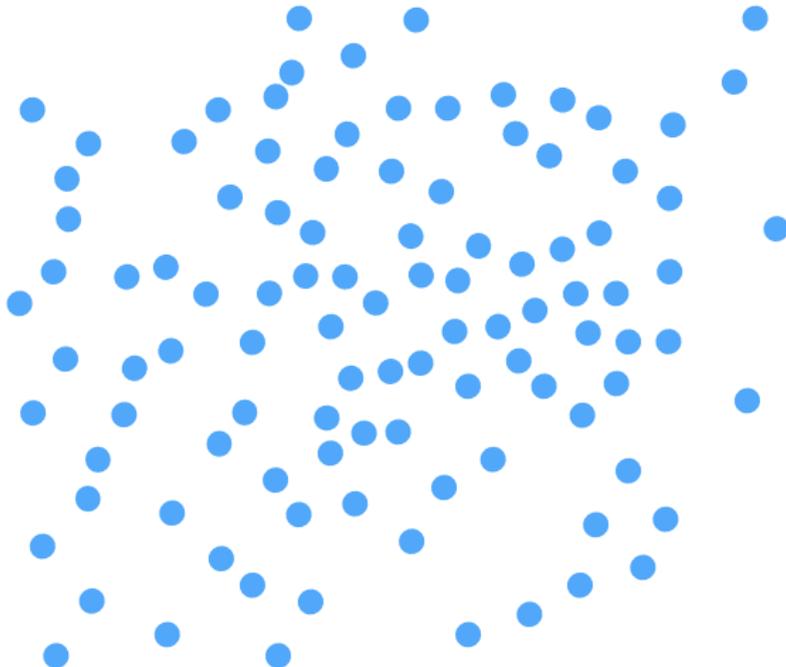
0.531783

Nb Partitions

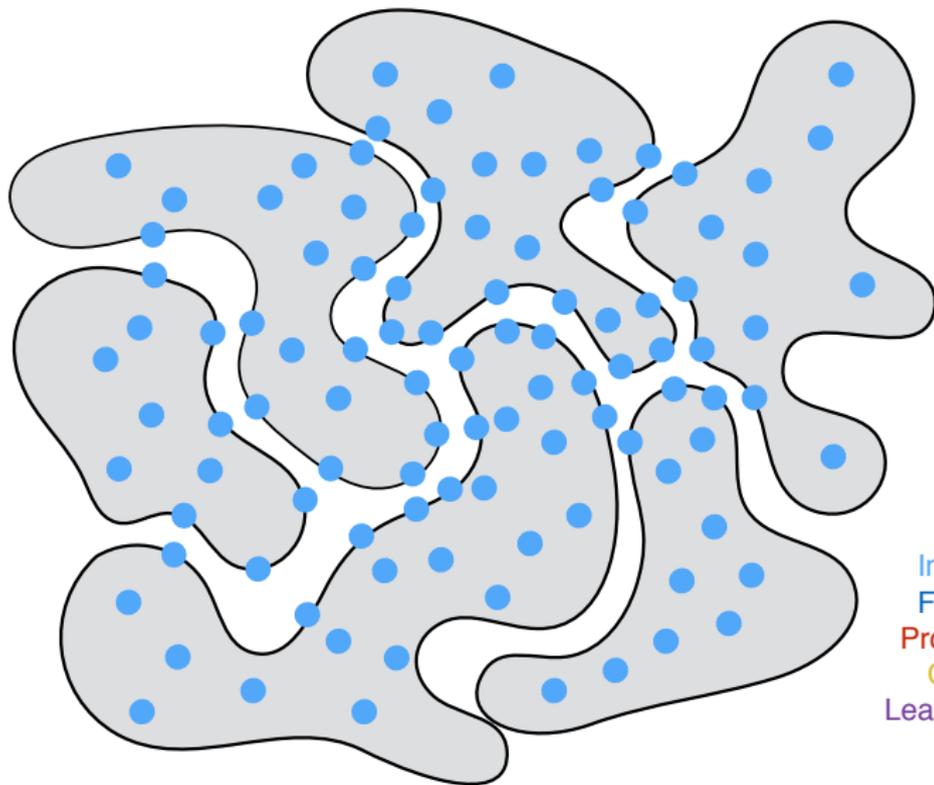
725



En déduire une illustration des CDCL

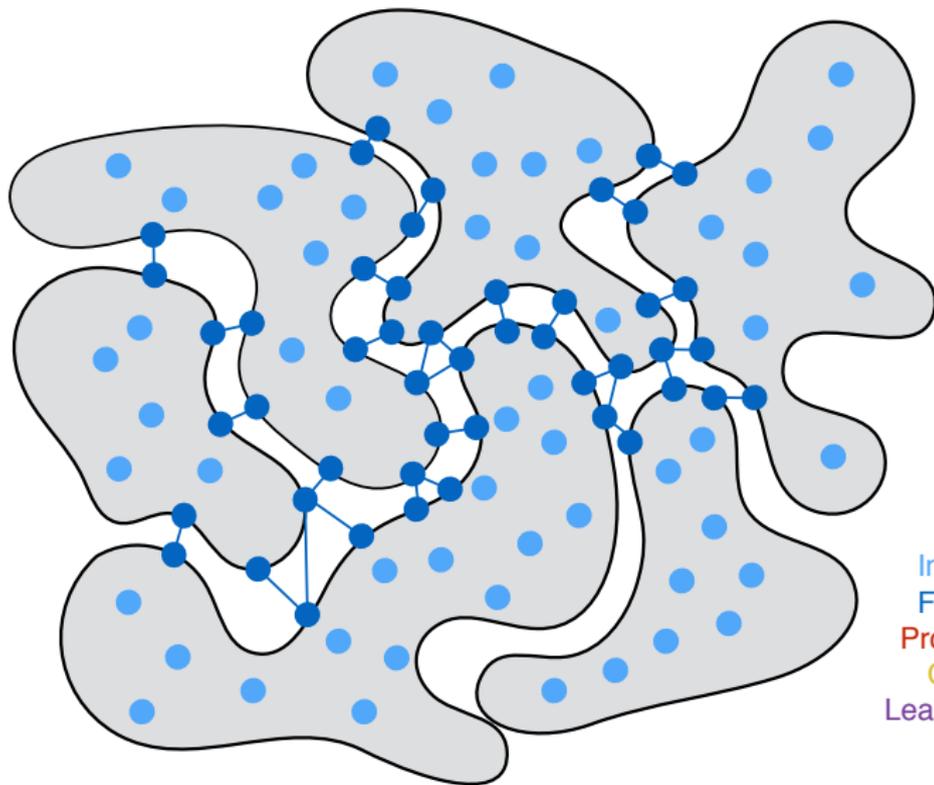


En déduire une illustration des CDCL



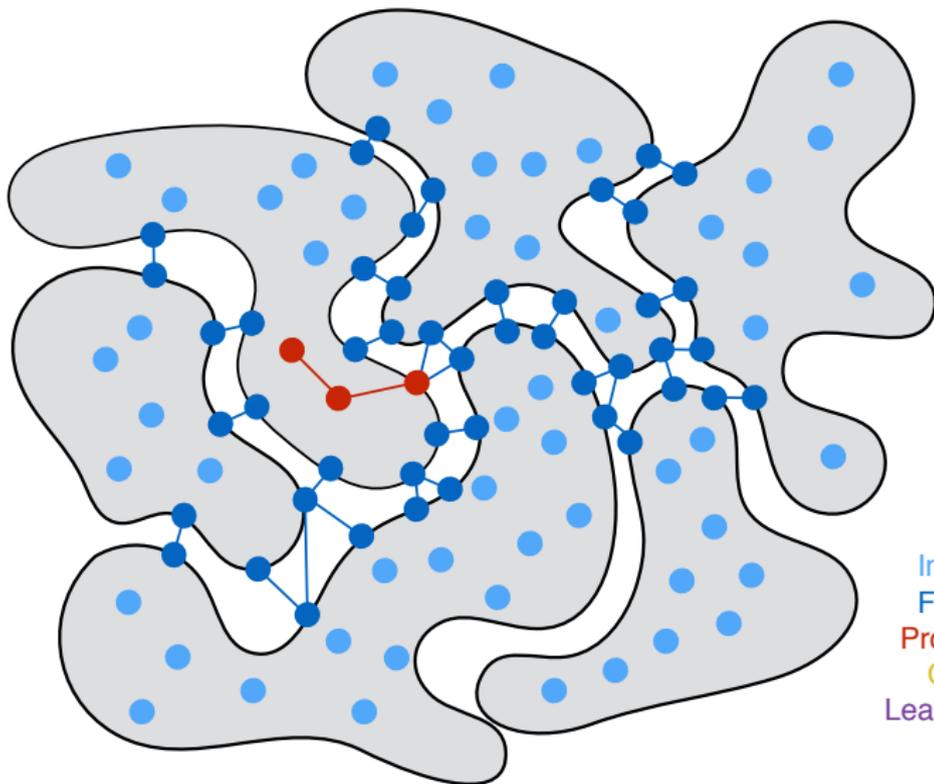
InCluster Variable
 Frontiere Variable
 Propagated Variable
 Conflict Variable
 Learnt Clause Variable

En déduire une illustration des CDCL



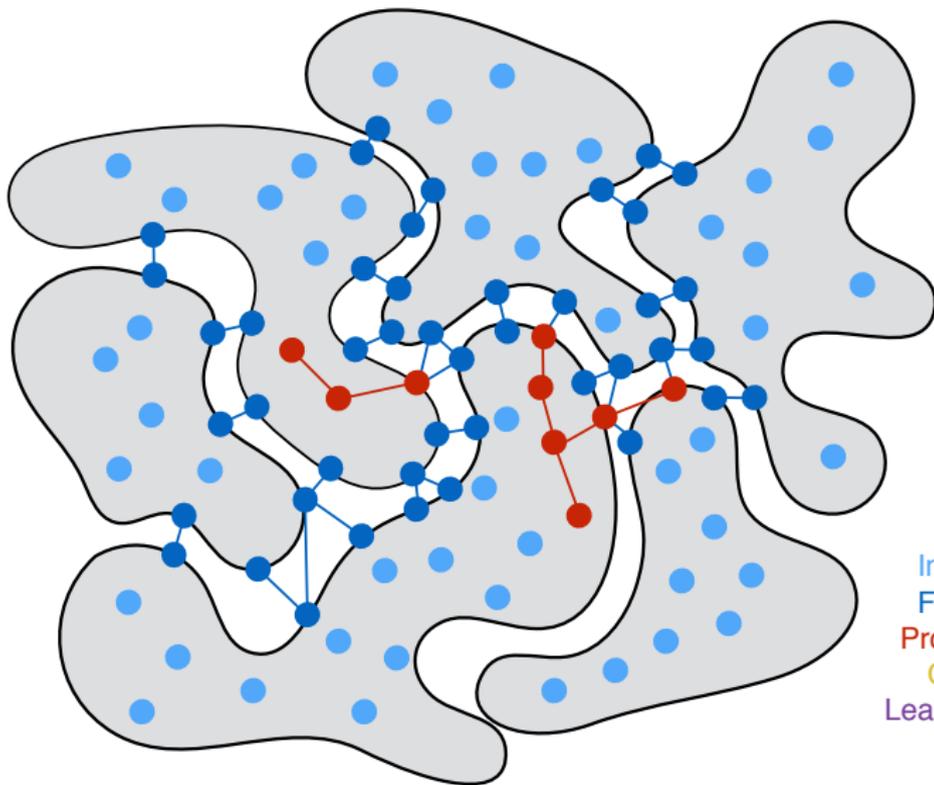
InCluster Variable
 Frontiere Variable
 Propagated Variable
 Conflict Variable
 Learnt Clause Variable

En déduire une illustration des CDCL



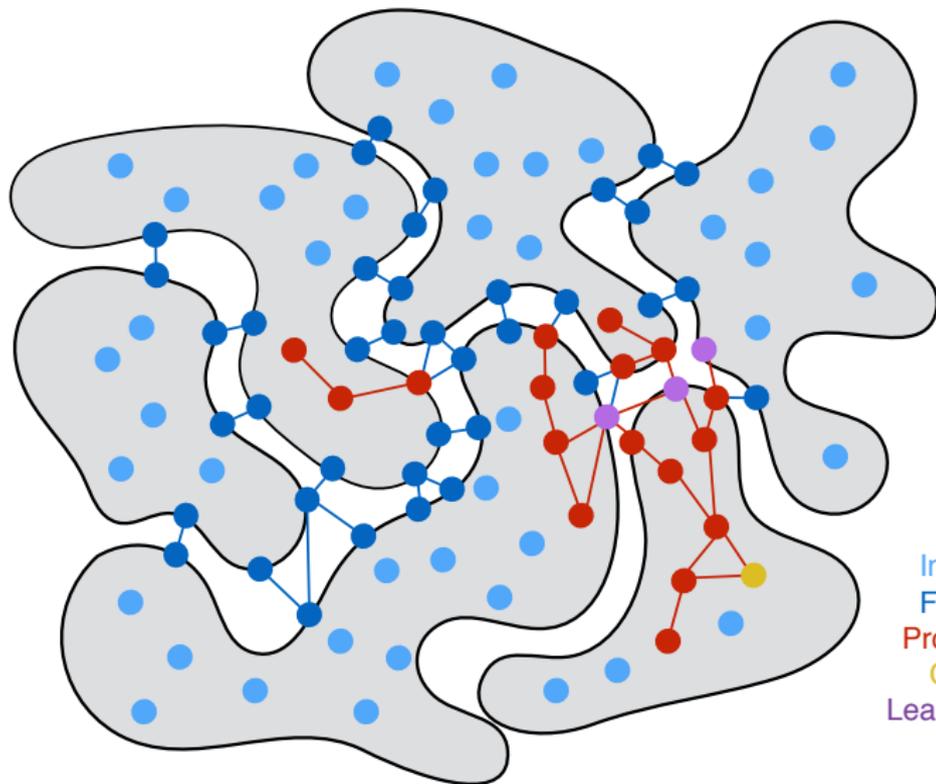
InCluster Variable
 Frontiere Variable
 Propagated Variable
 Conflict Variable
 Learnt Clause Variable

En déduire une illustration des CDCL



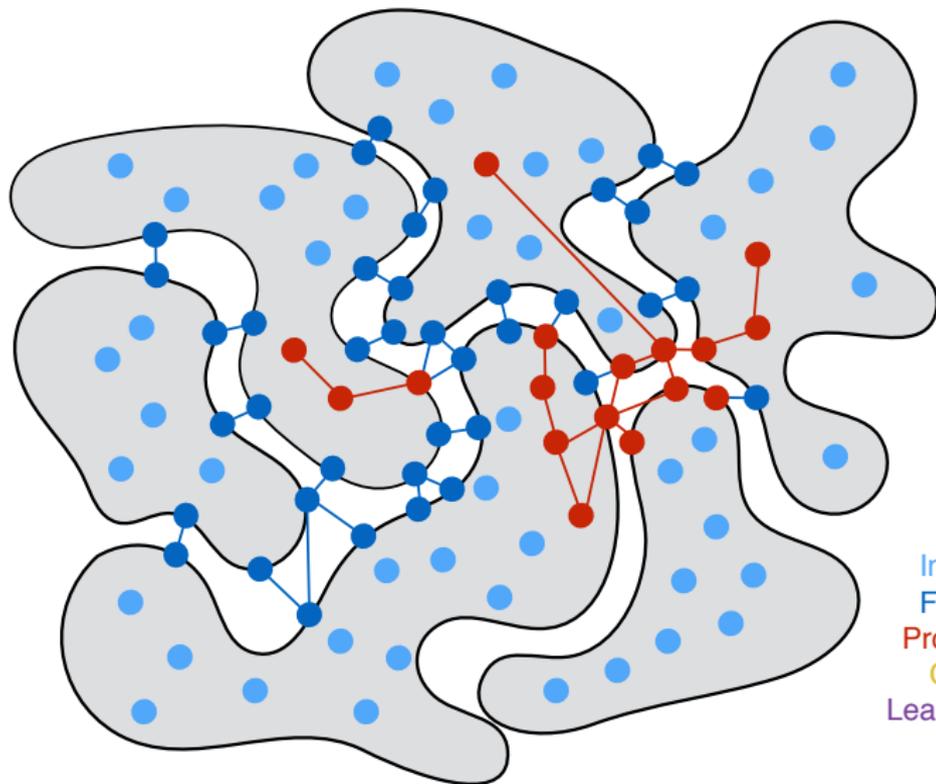
InCluster Variable
 Frontiere Variable
 Propagated Variable
 Conflict Variable
 Learnt Clause Variable

En déduire une illustration des CDCL



InCluster Variable
 Frontiere Variable
 Propagated Variable
 Conflict Variable
 Learnt Clause Variable

En déduire une illustration des CDCL



InCluster Variable
 Frontiere Variable
 Propagated Variable
 Conflict Variable
 Learnt Clause Variable



Plus de questions que de réponses

On sait écrire un solveur efficace mais difficile d'en expliquer les bonnes performances

- CDCL n'est pas DPLL
 - ➔ *redémarrages rapides, oubli agressif des clauses*
- Chercher un modèle ou une preuve UNSAT ?
 - ➔ *que favorise l'heuristique VSIDS ?*
- Apprendre peut être mauvais
 - ➔ *garder toutes les clauses n'est pas le meilleur choix*
- Redémarrer n'est pas redémarrer
 - ➔ *on retourne au même endroit, par un chemin différent*
- Bonnes variables : en haut ou en bas de l'arbre ?
 - ➔ *découper en haut, regrouper en bas*

Encore une marge de progression !

**Quasiment tout le temps des solveurs
est dans la propagation unitaire**

Mais, sur des instances UNSAT :

- Seules 50% des clauses apprises sont utiles pour la contradiction
- Seules 20% des propagations sont utiles pour produire chaque clause

**90% du temps du solveur ne sert pas
la construction finale de la preuve !**

Comprendre les CDCL

La compréhension des CDCL est au cœur des préoccupations de la communauté

De nombreuses questions (simples) n'ont pas encore de réponses

Cette compréhension passe par des études expérimentales

Paralléliser les CDCL

L'un des gros enjeux à venir

Dur de faire mieux (2011) :

- PPFolio a remporté de nombreuses médailles en 2011
- Lance simplement les meilleurs solveurs sans échanges

Progrès récents : Stratégie d'échange paresseuse des clauses

- Attendre qu'une clause soit localement intéressante avant de l'envoyer
- Toute clause venant de l'extérieur est ajoutée après probation

L'incompréhension des CDCL empêche leur parallélisation massive :

- Que fait VSIDS ? Comment découper le problème ?
- Quelles clauses sont bonnes à apprendre ? à Partager ?
- ...

