

Des évolutions récentes dans la résolution du problème SAT

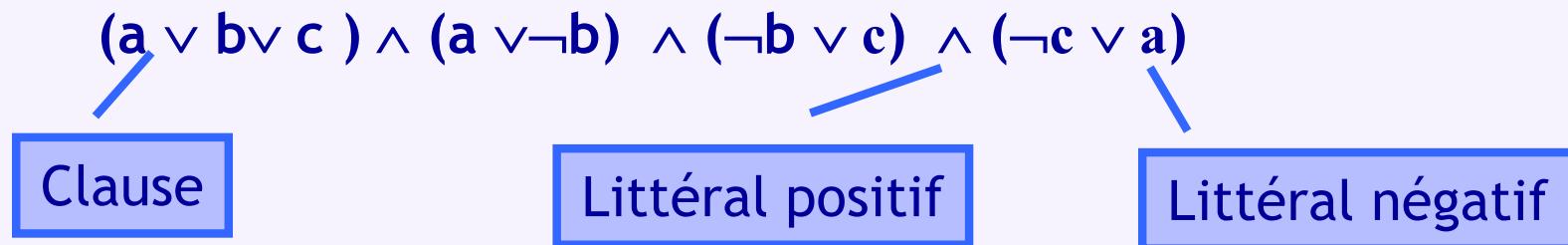
Lakhdar Saïs
CRIL CNRS, Université d'Artois

Plan

- ▶ Introduction
- ▶ De la résolution au solveurs SAT modernes
- ▶ ... résolution parallèle de SAT
- ▶ Conclusion

Problème SAT

- ▶ Etant donnée une formule booléenne (CNF)



- ▶ Admet-elle une valuation qui la rende vraie?
 - Oui : a = vrai, b = vrai et c = vrai (modèle, F est Satisfiable)

Problème SAT

- ## ▶ Etant donnée une formule booléenne (CNF)

$$(a \vee b \vee c) \wedge (\neg a \vee b) \wedge (\neg b \vee c) \wedge (\neg c \vee a) \wedge (\neg a \vee \neg b \vee \neg c)$$

- ## ▶ Admet-elle une valuation qui la rende vraie?

- Non : elle est insatisfiable

SAT- théorie de la complexité

- ▶ Problème NP-Complet de Référence [Cook 71]
- ▶ Utilisé pour prouver la NP-Compleétude d'autres problèmes (Π) - notion réduction polynomiale
 - $\Pi \in \text{NP}$ - certificat polynomiale
 - Réduction polynomiale de SAT vers Π
 - a Π est NP-Complet (pas d' algorithme polynomiale [sous hyp P!=NP])

SAT- applications

► De nombreux domaines d'applications

- Vérification formelle de matériels et de logicielles
- Démonstration automatique
- Bioinformatique
- Et même en cryptographie
 - ▶ F. MASSACCI [Using walk-SAT and rel-sat for cryptographic key search](#). In *IJCAI-99*
 - ▶ F. MASSACCI and L. MARRARO. [Logical cryptanalysis as a SAT-problem: Encoding and analysis of the U.S.S. Data Encryption Standard](#). *Journal of Automated Reasoning* 24(1-2), (2000).
 - ▶ C. FIORINI, E. MARTINELLI, and F. MASSACCI. [How to fake an RSA signature by encoding modular root finding as a SAT problem](#). *Discrete Applied Mathematics* 130(2), (2003).

► Progrès spectaculaires : des instances industrielles

- avec des centaines de milliers de variables
 - et de millions de clauses
- sont résolues en quelques secondes

Exemple : une instance

Nom de l'instance : post-cbmc-zfcpc-2.8-u2.cnf

p cnf 11 483 525 (vars) 32 697 150 (clauses)

1 -3 0

2 -3 0

$$x_3 = (x_1 \wedge x_2)$$

-1 -2 3 0

...(1 million de pages)...

-11482897 -11483041 -11483523 0

11482897 11483041 -11483523 0

$$x_3 \Leftrightarrow x_4 \Leftrightarrow x_5$$

11482897 -11483041 11483523 0

-11482897 11483041 11483523 0

-11483518 -11483524 0

-11483519 -11483524 0

$$x_6 = (x_7 \wedge x_8 \wedge x_9 \wedge x_{10} \wedge x_{11} \wedge x_{12})$$

-11483520 -11483524 0

-11483521 -11483524 0

-11483522 -11483524 0

-11483523 -11483524 0

11483518 11483519 11483520 11483521 11483522 11483523 11483524 0

-8590303 -11483524 -11483525 0

8590303 11483524 -11483525 0

8590303 -11483524 11483525 0

$$x_{13} \Leftrightarrow x_{14} \Leftrightarrow x_{15}$$

-8590303 11483524 11483525 0

-11483525 0

Résolu en moins d'1 minute!

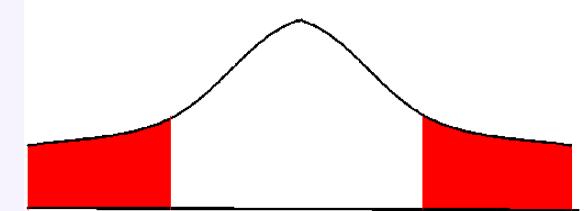
Quelques propriétés structurelles

- ▶ Les longues traînes (heavy-tailed phenomena [Gomes et al:97])
- ▶ Notion de variables essentielles (backdoor) [Willams:03]

Heavy-tailed distribution

- ▶ a une queue de la forme Paréto Lévy:

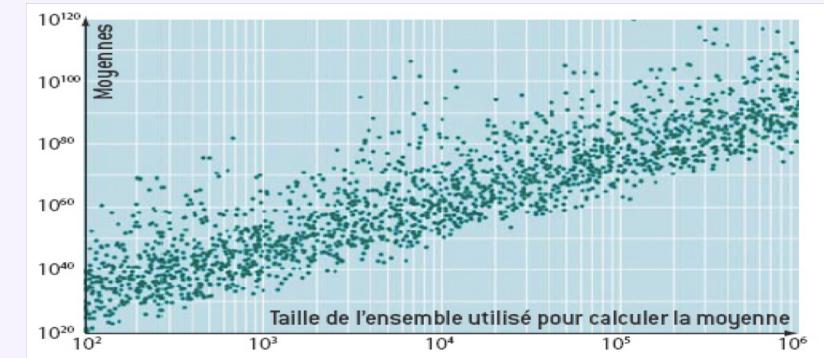
$$\Pr[X > x] = c x^{-\alpha}, \quad x > 0$$



HEAVY TAILED DISTRIBUTION
(infinite mean & variance)

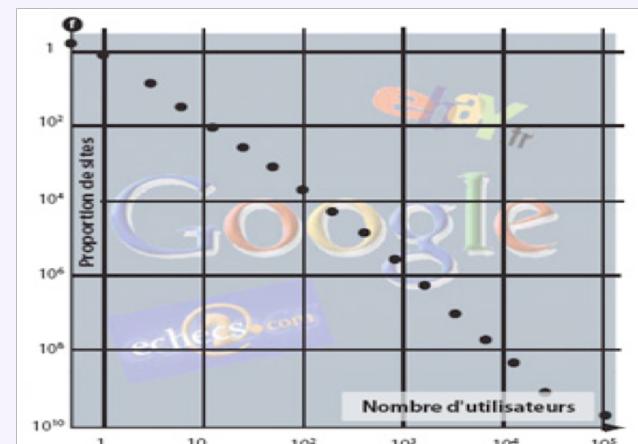
- ▶ Variance et Moyenne infinie

- ▶ Introduit par Pareto en 1920



- ▶ Mandelbrot l'a utilisé pour modéliser des phénomènes fractales

- ▶ De nombreux exemples :
marchés, météo, trafic internet

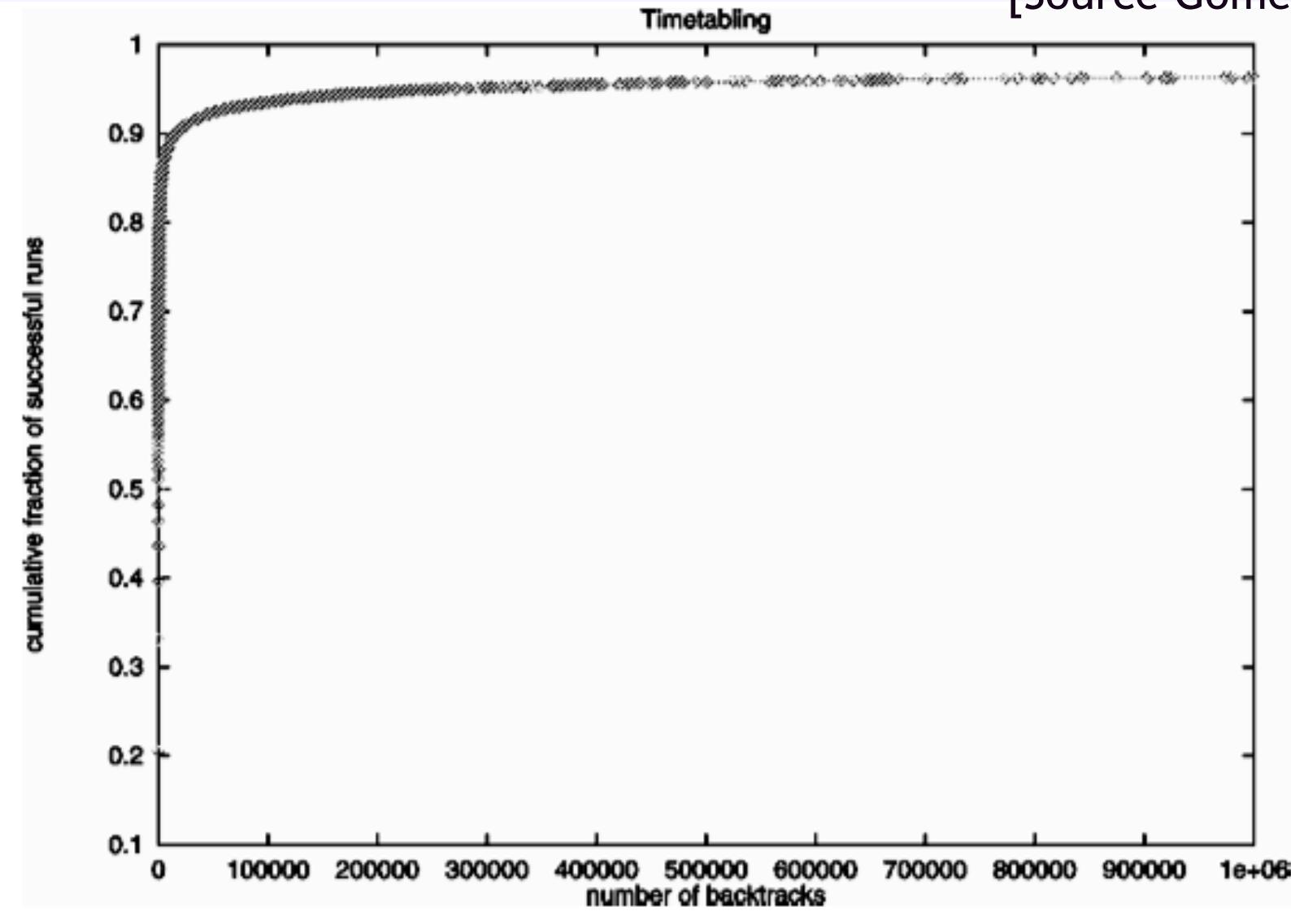


Phénomène longue traîne en SAT

- ▶ Observé sur de nombreuses familles d'instances SAT
 - Grande variabilités en temps entre différentes exécutions d'un algorithme
 - Un mauvais choix dans les premières décisions peut conduire à des arbres de tailles exponentielles
 - Un choix chanceux de bonnes décisions peut conduire à des arbres de tailles polynomiales

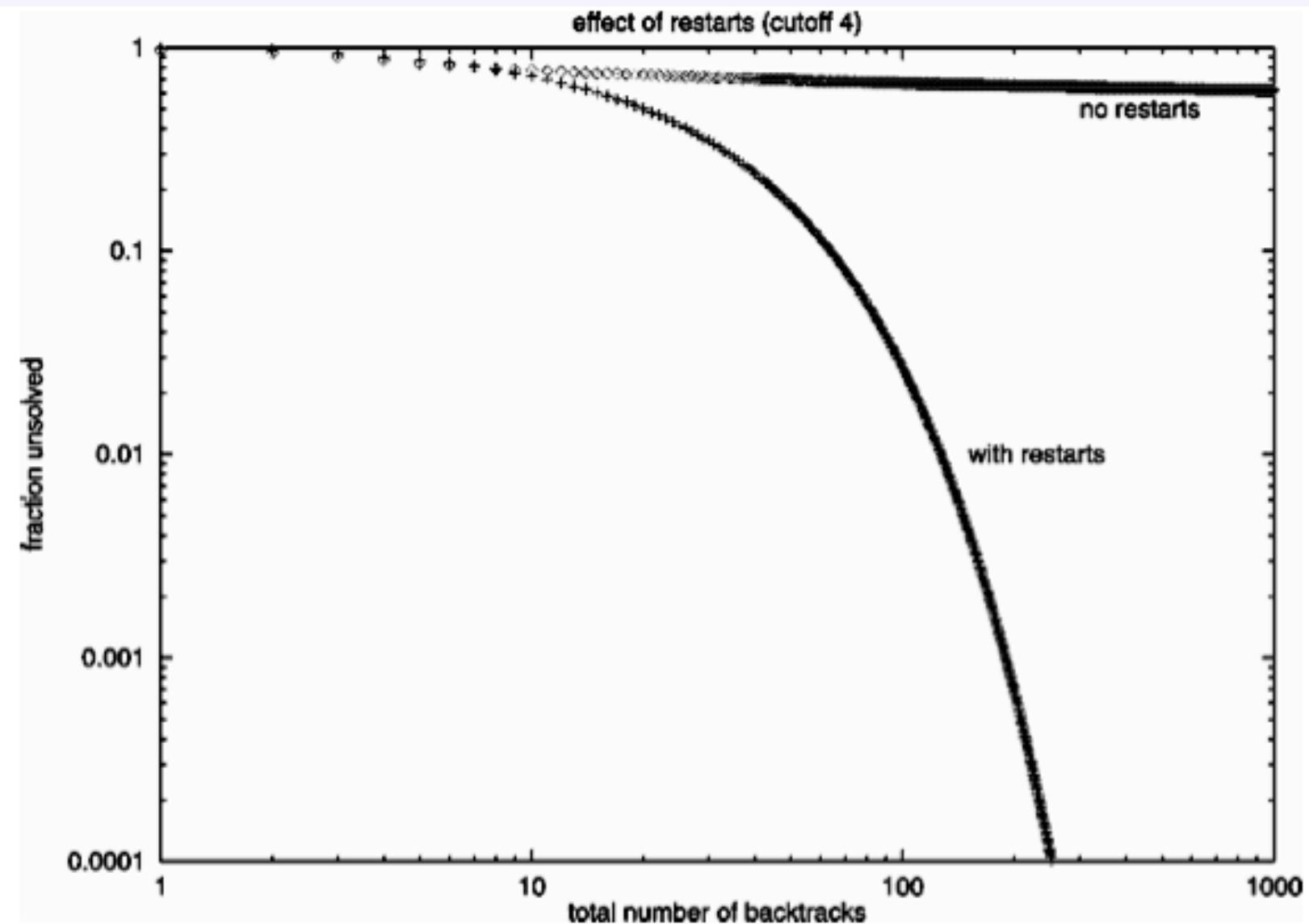
Longue traîne : exemple

[Source Gomes-Selman]

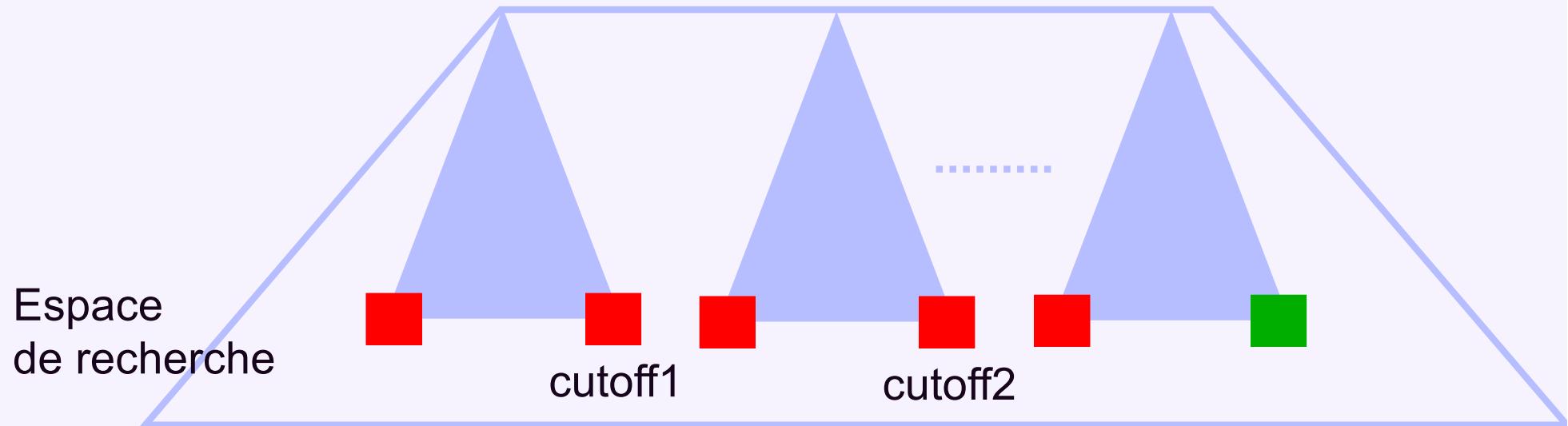


Instance de problème d'emploi du temps (10 000 exécutions)

Restarts et Heavy tailed dist.



Les restarts : première brique des solveurs SAT modernes



► Cutoff_i : progression

- Arithmétique
- Géométrique
- Suite de Loby
- **Dynamique** (avec Youssef Hamadi et Said jabbour)

Plan

- ▶ Introduction
- ▶ De la résolution au solveurs SAT modernes
- ▶ ... résolution parallèle de SAT
- ▶ Conclusion

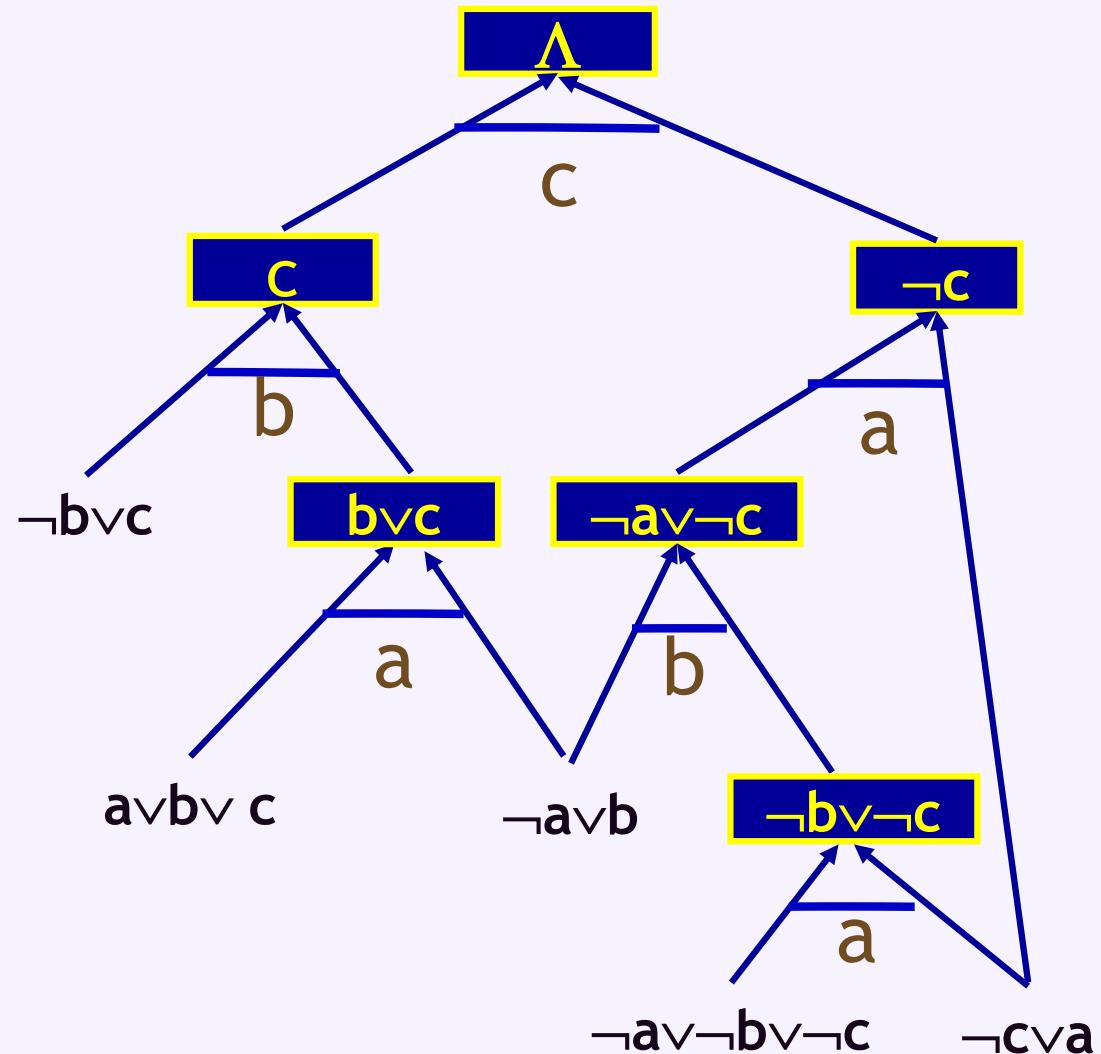
Résolution

- ▶ Soit une formule CNF F
- ▶ Règle de résolution
 - $(A \vee x), (B \vee \neg x)$ déduire $(A \vee B)$
 - Notation : $\eta[x, (A \vee x), (B \vee \neg x)] = (A \vee B)$
- ▶ La clause vide est dérivable (réfutation)
 $\Leftrightarrow F$ est unsatisfiable
- ▶ Taille de la preuve = # clauses utilisées

Graphe de résolution : DAG

$F =$

- c1 ($a \vee b \vee c$)
- c2 ($a \vee \neg b$)
- c3 ($\neg b \vee c$)
- c4 ($\neg c \vee a$)
- c5 ($\neg a \vee \neg b \vee \neg c$)



Formes restreintes de résolution

- ▶ “Tree Resolution”
 - Le graphe de résolution (DAG) doit être un arbre binaire
 - ▶ i.e., pour utiliser une clause une seconde fois, il faudra la dériver une seconde fois.
- ▶ Résolution ordonnée
 - Sur chaque branche du graphe de résolution (DAG), les variables sur lesquelles on a effectué la résolution respectent un ordre fixé (statique).
- ▶ Résolution régulière
 - Sur chaque branche du graphe de résolution (DAG), l’ensemble des variables sur lesquelles on a effectué la résolution est sans répétition.
 - Plus général que la résolution ordonnée et la “Tree Resolution”

Résoudre SAT (DP & DPLL)

► DP : Davis Putnam 60

- Elimination de variables par résolution
- Notation : $F(x) = \{c \mid c \in F \wedge x \in c\}$
- $\forall x \in V_F, F = F - [F(x) \cup F(\neg x)] \cup \eta[x, F(x), F(\neg x)]$
- DP \Leftrightarrow Résolution ordonnée

► DPLL : Davis Logemann Loveland (1962)

- Algorithme énumératif (choix+propagation et retour arrières = « backtrack search »)
- Notation : $F|x = \{c \mid c \in F, x \notin c\} \cup \{c - (\neg x) \mid c \in F, \neg x \in c\}$
- $F = (F|x) \vee (F|\neg x)$
- DPLL \Leftrightarrow Tree resolution

DPLL

DPLL(F)

while ($\exists c = (x) \in F$), $F \leftarrow F|_x // F \leftarrow F^*$

if $F = \{\}$ return true

if $\lambda \in F$ return false

else choix du littéral x à affecter

return DPLL($F|_x$) \vee DPLL($F|_{\neg x}$)

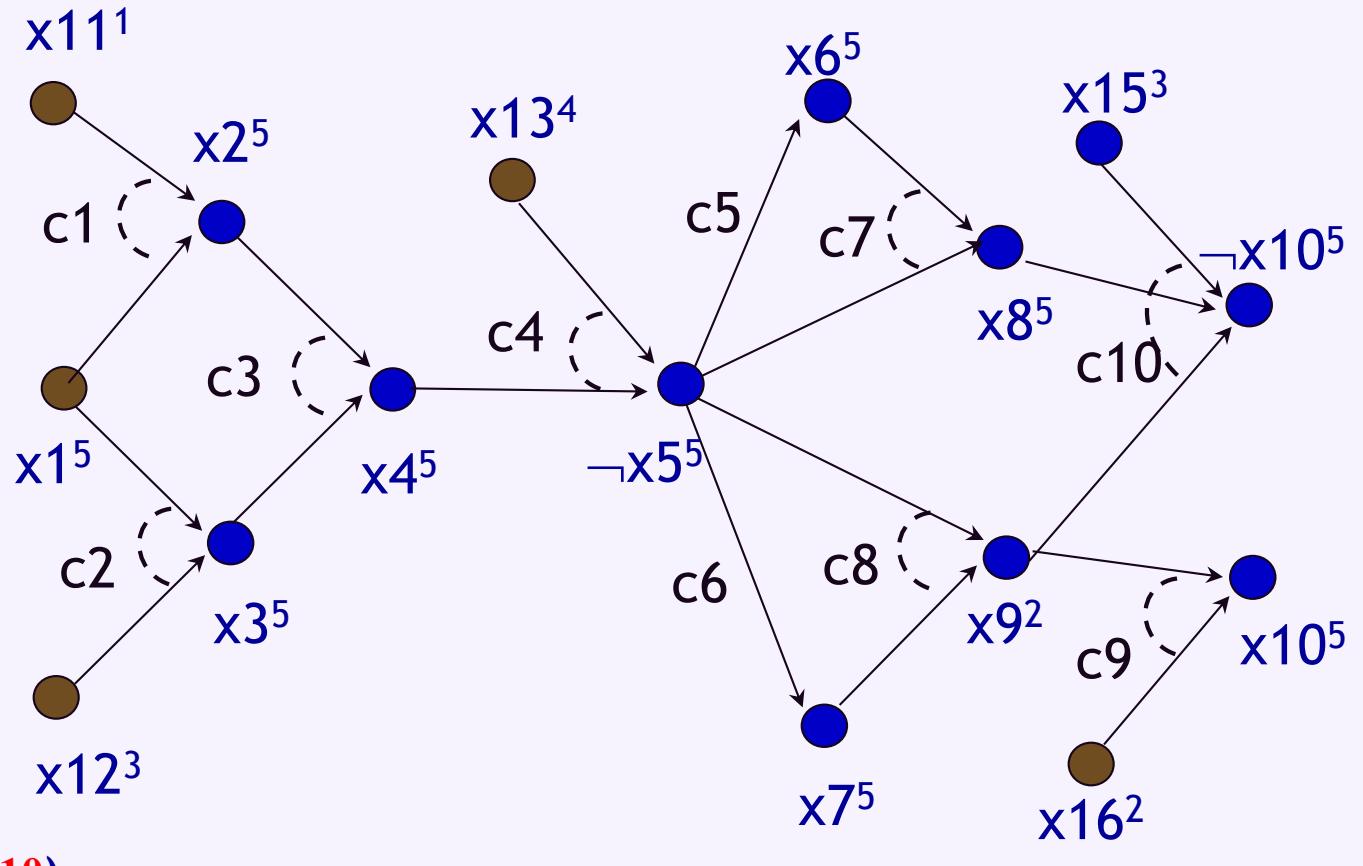
DPLL + Learning

- ▶ GRASP [Marques-Silva et al. 1996]
- ▶ À chaque conflit, analyse, déduction d'une clause représentant les causes du conflit, ajout de la clause et retour arrière non chronologique.
- ▶ Learning : critique pour l'efficacité des solveurs SAT modernes (**seconde brique**)
- ▶ La puissance de DPLL + Learning ?
?= résolution régulière ?= la résolution générale

Learning : exemple

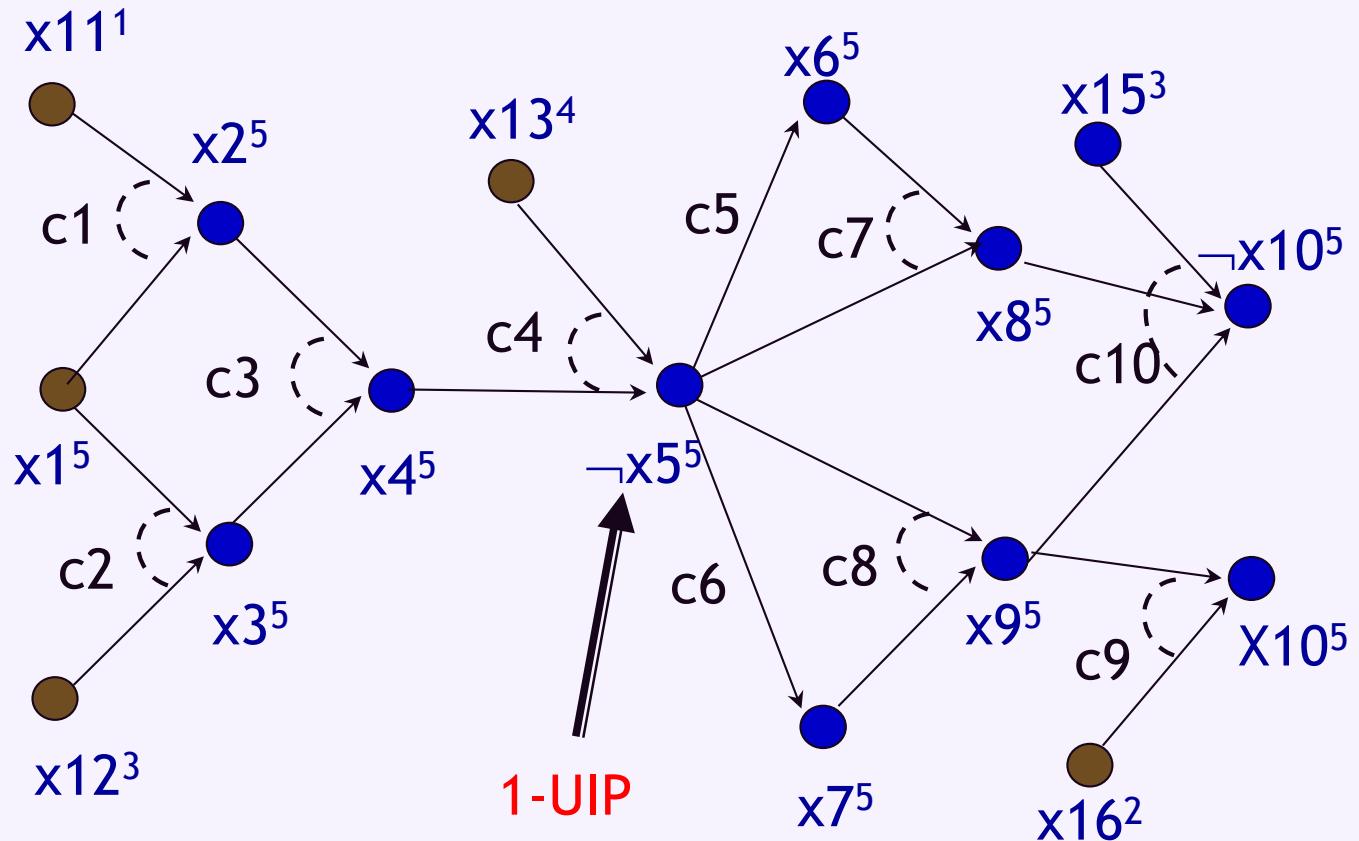
$F \supset$

- c1 ($\neg x_1 \vee \neg x_{11} \vee x_2$)
- c2 ($\neg x_1 \vee \neg x_{12} \vee x_3$)
- c3 ($\neg x_2 \vee \neg x_3 \vee x_4$)
- c4 ($\neg x_4 \vee \neg x_{13} \vee \neg x_5$)
- c5 ($x_5 \vee x_6$)
- c6 ($x_5 \vee x_7$)
- c7 ($x_5 \vee \neg x_6 \vee x_8$)
- c8 ($x_5 \vee \neg x_7 \vee x_9$)
- c9 ($\neg x_9 \vee \neg x_{16} \vee x_{10}$)
- c10 ($\neg x_{15} \vee \neg x_8 \vee \neg x_9 \vee \neg x_{10}$)



$$\rho = \langle (x_{11}^1), (x_{16}^2), (x_{12}^3), (x_{15}^3), (x_{13}^4), \\ \langle (x_1^5), (x_2^5), (x_3^5), (x_4^5), (\neg x_5^5), (x_6^5), (x_7^5), (x_8^5), (x_9^5), (x_{10}^5) \rangle \rangle$$

Learning : exemple



$$R1 = \eta[x10, c9, c10] = (\neg x15^3 \vee \neg x16^2 \vee \neg x8^5 \vee \neg x9^5)$$

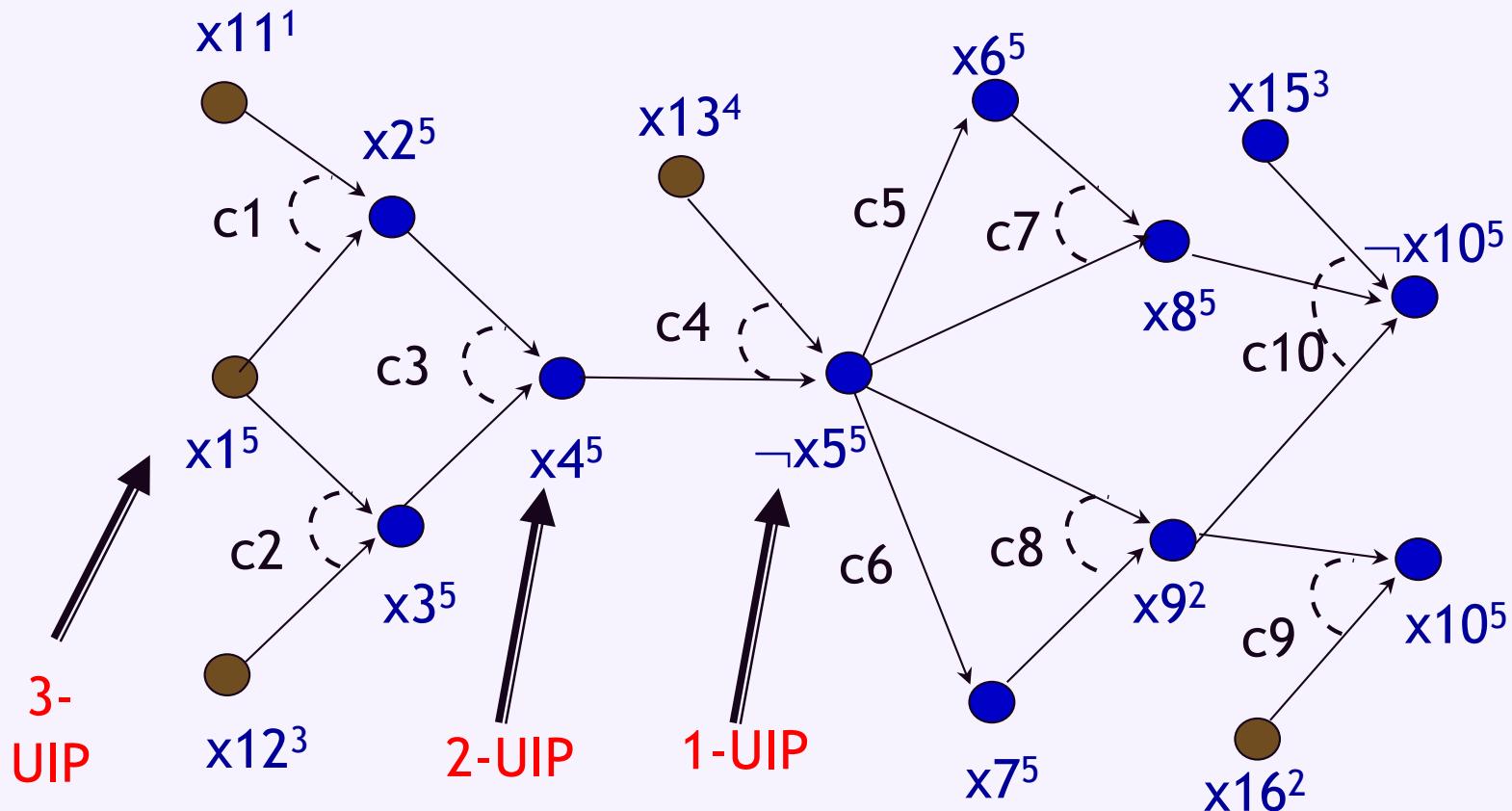
$$R2 = \eta[x9, R1, c8] = (\neg x15^3 \vee \neg x16^2 \vee \neg x7^5 \vee x5^5 \vee \neg x8^5)$$

$$R3 = \eta[x8, R2, c7] = (\neg x15^3 \vee \neg x16^2 \vee \neg x6^5 \vee x5^5 \vee \neg x7^5)$$

$$R4 = \eta[x7, R3, c6] = (\neg x15^3 \vee \neg x16^2 \vee x5^5 \vee \neg x6^5)$$

$$A1 = \eta[x6, R4, c5] = (\neg x15^3 \vee \neg x16^2 \vee \underline{x5^5}) \text{ (Clause assertive)}$$

Learning : exemple



$$A1 = \eta[x_6, R4, c5] = (\neg x_{15}^3 \vee \neg x_{16}^2 \vee x_5^5) \text{ (Clause assertive)}$$

$$A2 = \eta[x_5, A1, c4] = (\neg x_{13}^4 \vee \neg x_{15}^3 \vee \neg x_{16}^2 \vee \neg x_4^5)$$

$$\ddots A3 = (\neg x_{13}^4 \vee \neg x_{12}^4 \vee \neg x_{15}^3 \vee \neg x_{16}^2 \vee \neg x_{11}^1 \vee \neg x_1^5)$$

Schéma 1-UIP : quelques propriétés

Propriété [Audemard-etal:08]

Le schéma first UIP est optimal en

- ▶ saut
- ▶ nombre de points de décisions
- ▶ Un cadre général pour l'analyse de conflit [Audemard-etal:08]

DPLL + Learning & Résolution

- ▶ DPLL + Learning >> Résolution régulière (sur certaines familles de formules) [Sabharwal-Kautz:04]
- ▶ [Darwiche 2011]
 - DPLL + Learning = Résolution générale

Propagation unitaire (PU)

- ▶ ~90% des littéraux affectées sont fait par PU
- ▶ Améliorer la PU : Approche 2-watch
[Moskewicz et al. 2001] (**Troisième brique**)



Propagation unitaire (PU)

- ▶ ~90% des littéraux affectées sont fait par PU
- ▶ Améliorer la PU : Approche 2-watch
[Moskewicz et al. 2001] (**Troisième brique**)



- $x_3 = \text{faux}$, $x_5 = \text{vrai}$: on ne fait rien

Propagation unitaire (PU)

- ▶ ~90% des littéraux affectées sont fait par PU
- ▶ Améliorer la PU : Approche 2-watch
[Moskewicz et al. 2001] (**Troisième brique**)



- $x_3 = \text{faux}$, $x_5 = \text{vrai}$: on ne fait rien
- $x_1 = \text{vrai}$: on cherche un autre watched

Propagation unitaire (PU)

- ▶ ~90% des littéraux affectées sont fait par PU
- ▶ Améliorer la PU : Approche 2-watch
[Moskewicz et al. 2001] (**Troisième brique**)



- $x_3 = \text{faux}$, $x_5 = \text{vrai}$: on ne fait rien
- $x_1 = \text{vrai}$: on cherche un autre watched (x_4)

Propagation unitaire (PU)

- ▶ ~90% des littéraux affectées sont fait par PU
- ▶ Améliorer la PU : Approche 2-watch
[Moskewicz et al. 2001] (**Troisième brique**)



- $x3 = \text{faux}$, $x5 = \text{vrai}$: on ne fait rien
- $x1 = \text{vrai}$: on cherche un autre watched ($x4$)
- $x4 = \text{faux}$: on cherche un autre watched

Propagation unitaire (PU)

- ▶ ~90% des littéraux affectées sont fait par PU
- ▶ Améliorer la PU : Approche 2-watch
[Moskewicz et al. 2001] (**Troisième brique**)



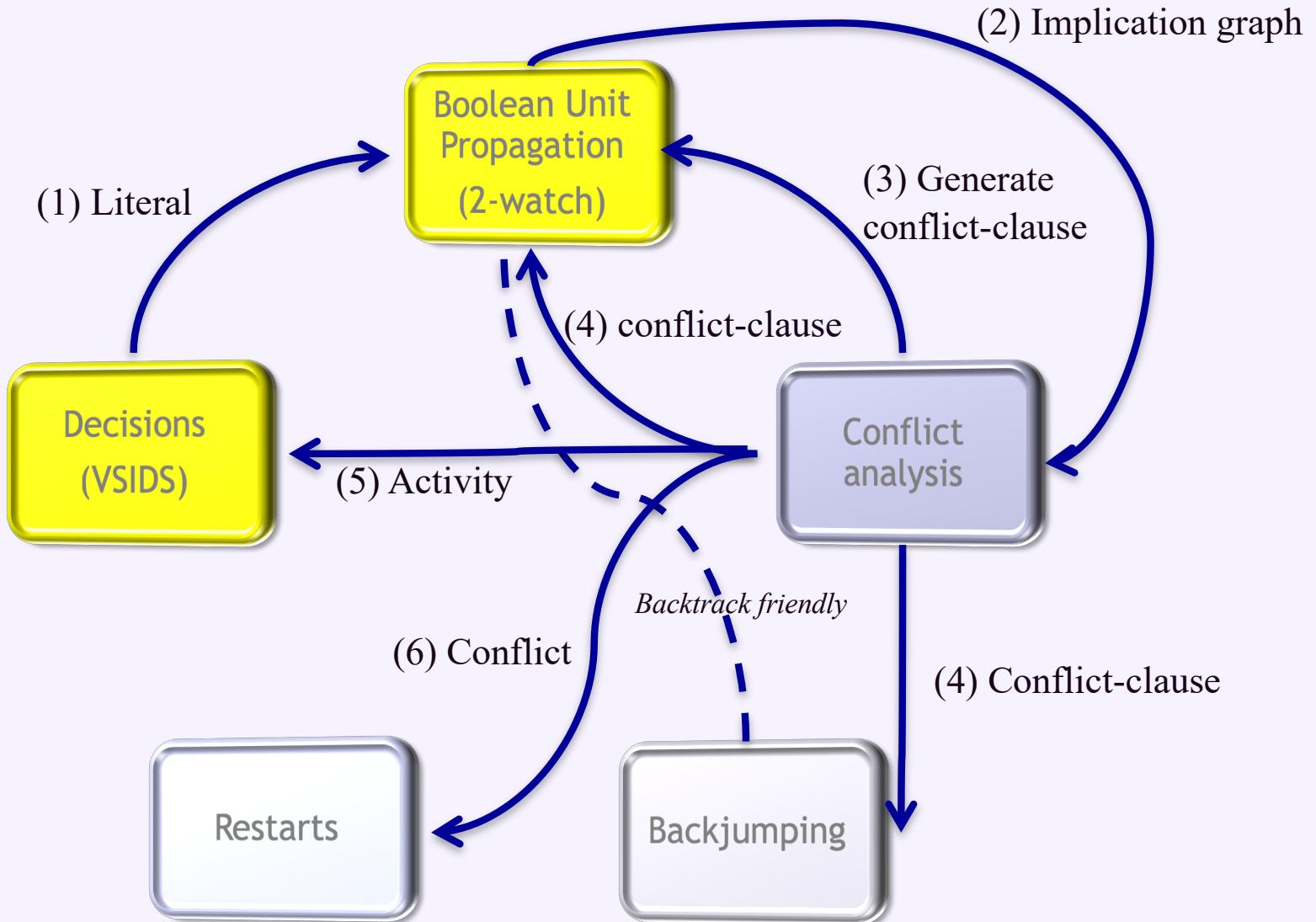
- $x3 = \text{faux}$, $x5 = \text{vrai}$: on ne fait rien
- $x1 = \text{vrai}$: on cherche un autre watched ($x4$)
- $x4 = \text{faux}$: on cherche un autre watched
 - $x2$ est unitaire => $x2 = \text{vrai}$ (UP)

Heuristique VSIDS

Variable State Independent Decaying Sum (VSIDS) [Moskewicz et al:01, Brisoux-etal:99] (**Quatrième brique**)

- ▶ On maintient pour chaque variable un compteur comptabilisant son activité lors de l’analyse des conflits (#fois utilisée dans une opération de résolution)
- ▶ Les valeurs de tous les compteurs sont périodiquement divisées par une constante $c > 1$ - préférence donnée au conflits les plus récents
- ▶ Choix de la variable ayant l’activité maximale

Architecture



Plan

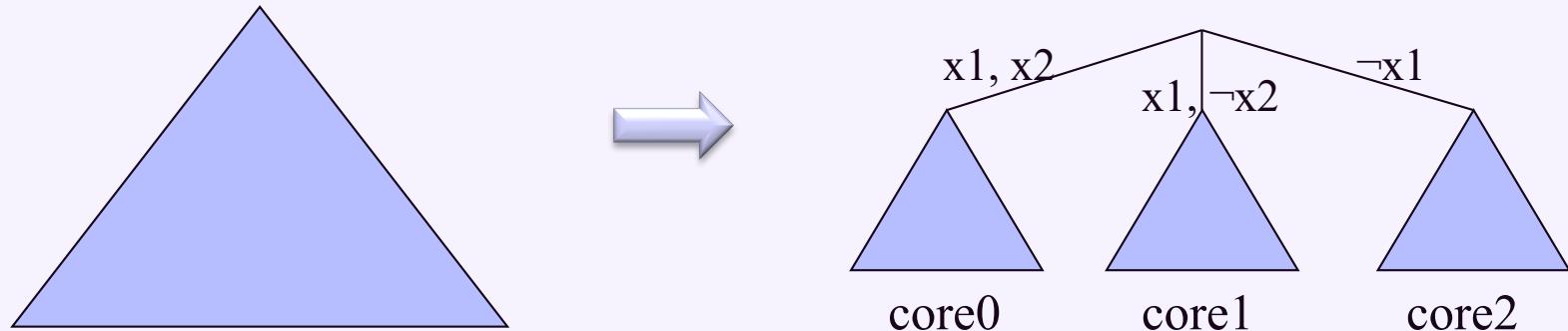
- ▶ Introduction
- ▶ De la résolution au solveurs SAT modernes
- ▶ ... Résolution parallèle de SAT
- ▶ Conclusion

... résolution parallèle de SAT

- ▶ Résolution séquentielle de SAT : des progrès notables mais...
 - Sur les instances industrielles résolues
 - ~90% des instances < 15 minutes et ~10% en 1 h et plus
 - Un large panel d'instances non résolues!
 - Amélioration mineurs (pas d'ordre de grandeurs) des meilleurs algorithmes SAT séquentielles
- ▶ Résoudre les instances les plus difficiles en profitant des nouvelles architectures multicores.

Approches “diviser pour régner”

- ▶ **Principe:** allouer des sous arbres indépendant aux différentes ressources
 - Utiliser le concept de “guidng paths” pour diviser l'espace de recherche.



- ▶ **Problème:** équilibrage de charges

Diviser pour régner dans SAT

	Base algorithm	Parallel architecture	Knowledge sharing
Psato [Zhang et al. 1996] [Bohm et al. 1996]	Sato	workstations	Load-balancing
Gradsat [Chrabakh et al. 2003]	zChaff	workstations	Load-balancing, clause sharing
[Blochinger et al. 2003]	zChaff	workstations	Load-balancing, restricted clause sharing
MiraXT [Lewis et al. 2007]	Minisat	multicore	Load-balancing, systematic clause sharing
Pminisat [Chu et al. 2008]	Minisat	multicore	Load-balancing, restricted clause sharing

ManySAT : philosophie

Au lieu d'une approche « diviser pour régner »

ManySAT

► intègre

- un portfolio de stratégies différentes et complémentaires
- pour explorer différemment l'espace de recherche

► exploite

- les faiblesses des solveurs modernes
 - Sensibilités aux réglages des paramètres
 - Manque de robustesse

► partage

- des clauses apprises pour éviter des explorations redondantes

(collaboration avec Youssef Hamadi, MSR, cambridge)

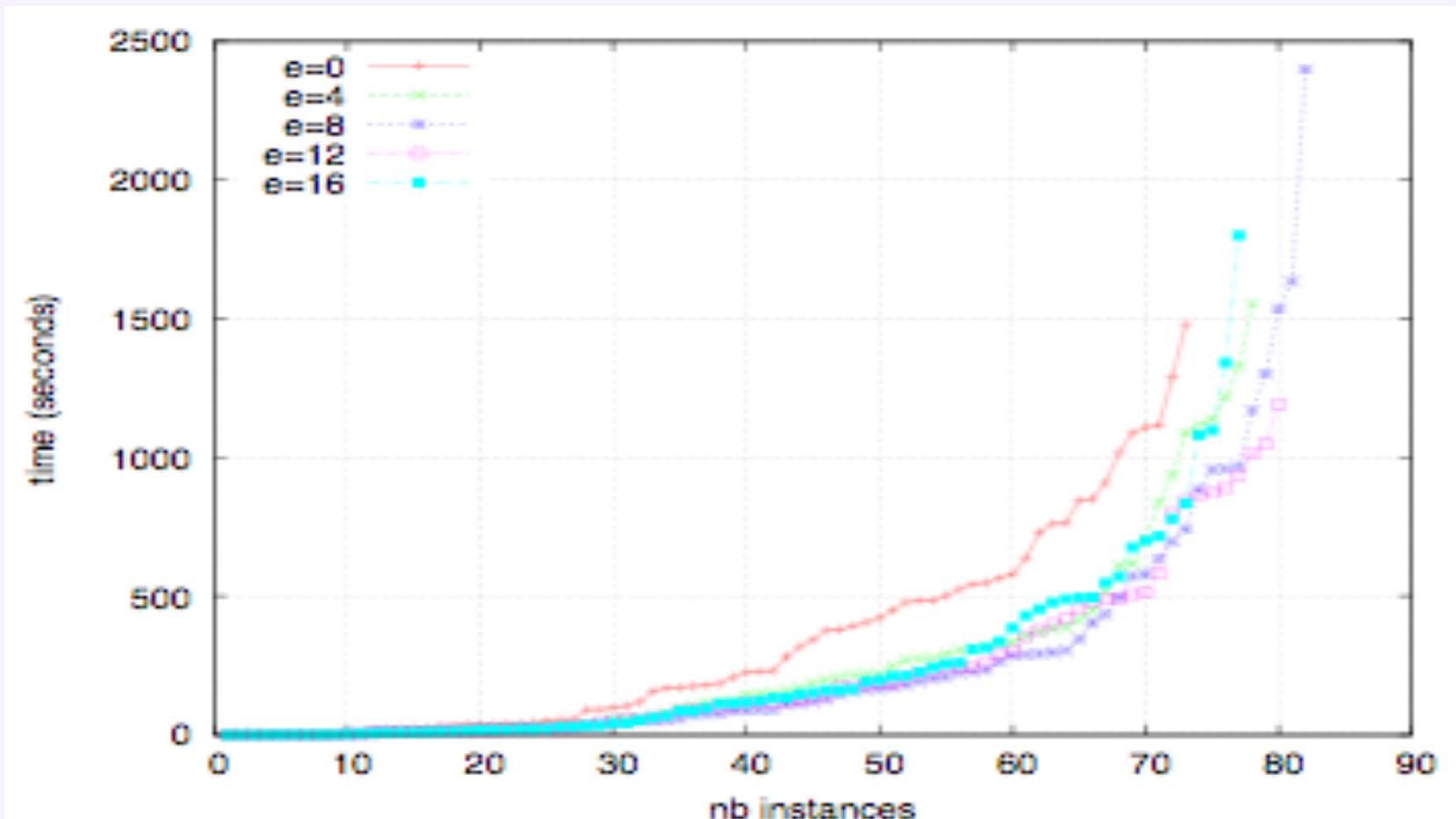
ManySAT : diversification

- ▶ Politique de restarts dynamique
 - Exploite des mesures liées à la difficulté relative des instances (hauteur moyenne des sauts, des arbres, etc)
- ▶ Nouveaux schémas d'analyse de conflits (cf. papier SAT'2008)
- ▶ ...

ManySAT (4 cores)

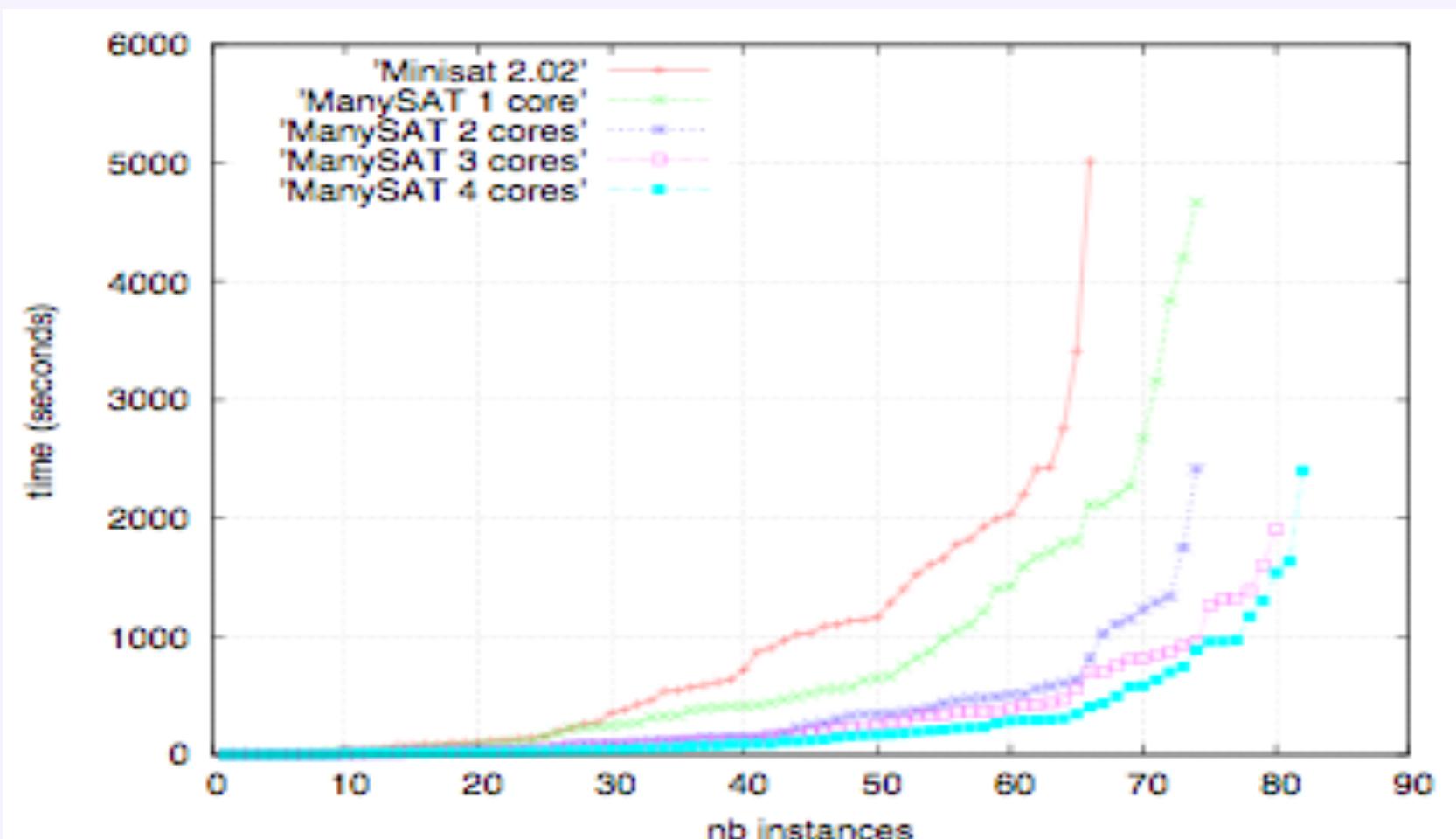
Strategies	Core 0	Core 1	Core 2	Core 3
Restart	Geometric $x_1 = 100$ $x_i = 1.5 \times x_{i-1}$	Dynamic (Fast) $x_1 = 100, x_2 = 100$ $x_i = f(y_{i-1}, y_i), i > 2$ if $y_i > y_{i-1}$ $f(y_{i-1}, y_i) = \frac{\alpha}{y_i} \times \cos(1 + \frac{y_{i-1}}{y_i}) $ else $f(y_{i-1}, y_i) = \frac{\alpha}{y_i} \times \cos(1 + \frac{y_i}{y_{i-1}}) $ $\alpha = 1200$	Arithmetic $x_1 = 16000$ $x_i = x_{i-1} + 16000$	Luby 512
Heuristic	VSIDS (3% rand.)	VSIDS (2% rand.)	VSIDS (2% rand.)	VSIDS (2% rand.)
Polarity	if $\#occ(l) > \#occ(\neg l)$ $l = true$ else $l = false$	Progress saving	false	Progress saving
Learning	CDCL (extended [1])	CDCL	CDCL	CDCL (extended [1])
Cl. sharing	size 8	size 8	size 8	size 8

ManySAT : performance (partage de clauses)



Instances Sat Race 2008

ManySAT : performance (séquentiel, 1, 2, 3 et 4 cores)



Instances Sat Race 2008

ManySAT : comparaison



► SAT-Race 2008

- 100 instances industrielles
- 900 secondes timeout
- 4 cores

	ManySAT	pMinisat	MiraXT
SAT	45	44	43
UNSAT	45	41	30

ManySAT : comparaison



► SAT-Race 2008

vs. Minisat 2.1 (meilleur solveur SAT séquentiel)

	ManySAT	pMinisat	MiraXT
Average speed-up	6.02	3.10	1.83
by SAT/UNSAT	8.84/3.14	4.00/2.18	1.85/1.81
Minimal speed-up	0.25	0.34	0.04
by SAT/UNSAT	0.25/0.76	0.34/0.46	0.04/0.74
Maximal speed-up	250.17	26.47	7.56
by SAT/UNSAT	250.17/4.74	26.47/10.57	7.56/4.26

ManySAT

- ▶ IBM Germany Research & Development GmbH in Boeblingen, Bounded Model Checking
- ▶ Microsoft,
 - Z3 SMT Solver -> //Z3
 - ‘En production’ pour la vérification de logicielles
- ▶ University of Munich, parity games
- ▶ ...

Source youssef hamadi, Microsoft research UK

Conclusion

- ▶ Progrès spectaculaires ces dernières années
- ▶ De plus en plus utilisé en pratique
 - Connexions avec d'autres domaines
 - Problèmes autour de SAT (QBF, Max-SAT, #SAT,...)
- ▶ Problèmes difficiles ?
- ▶ Résolution parallèle de SAT : une direction prometteuse, avec des challenges à relever