

Hard Satisfiable Clause Sets for Benchmarking SAT Solvers with Equivalence Reasoning Techniques

Ilkka Niemelä

Ilkka.Niemela@tkk.fi, <http://www.tcs.hut.fi/~ini/>

Laboratory for Theoretical Computer Science
Department of Computer Science and Engineering
Helsinki University of Technology

(Joint work with Harri Haanpää, Matti Jarvisalo, and Petteri Kaski)

Introduction

- Solver development benefits from challenging sets of benchmarks.
- Industrial benchmarks often most relevant from a practical perspective.
- But they come typically as small collections of individual instances.
☞ Limited usability for solver development (hard to measure scalability).
- Needed: families of benchmarks for measuring progress



Outline

- Introduction
- SAT solvers and Equivalence Reasoning Techniques
- Regular XORSAT
- Introducing Nonlinearity
- Experiments
- Conclusions

Interesting Benchmark Families

- Have control parameters—such as size—that provide control over the difficulty of a test case.
- Test cases with the same parameter values should have similar computational characteristics.
- Already fairly small test cases should be challenging.
- Test cases should have structure that could be exploited by an advanced solver.
- It should be easy to generate large numbers of guaranteed satisfiable (unsatisfiable) test cases of the same size.



SAT Benchmark Families

- A number of families of hard *unsatisfiable* instances are known (Tseitin, Haken, Chvátal/Szemerédi, Urquhard, ...).
- Generating hard *guaranteed satisfiable* instances more challenging
- Random k -SAT model:
 - Special techniques to guarantee satisfiability
 - Relatively easy for local search
- Quasigroup completion (Achlioptas et al.)
 - Large clause sets needed for hard test cases



Benchmarking Equivalence Reasoning

- In many applications solving *satisfiable instances* is important.
- The goal is to devise a family of benchmarks to address, e.g., the following questions:
 - Can small satisfiable instances of XORSAT be difficult for state-of-the-art SAT solvers?
 - How well do current solvers with equivalence reasoning techniques cope with basic XORSAT?
 - How sensitive are they if a *mixed problem* is considered where some nonlinearity is introduced to basic XORSAT?
- Main result: *regular XORSAT* family
 - ➔ *Small hard satisfiable* benchmarks



SAT and Equivalence Reasoning

- In application for SAT solvers XORs and equivalences (systems of linear equations modulo 2) occur frequently: hardware design, cryptanalysis, ...
Note: $x_1 \oplus x_2 \oplus x_3 \doteq x_1 + x_2 + x_3 \equiv 1 \pmod{2}$
- Typical approach is to translate the equations to clauses and use a clausal solver.
- Some SAT solvers include *equivalence reasoning techniques*, i.e., special methods for solving linear equations modulo 2 presented in clausal form.
- **XORSAT**: systems of linear equations modulo 2 presented in CNF
- Unsatisfiable instances of XORSAT can be very hard for DPLL/resolution (Tseitin)



Regular XORSAT

Instances of basic regular XORSAT are constructed as follows:

- A d -regular constraint graph is selected uniformly at random.
- A system of linear equations modulo 2 based on the graph is constructed.
- The equations are transformed into an equivalent set of clauses.

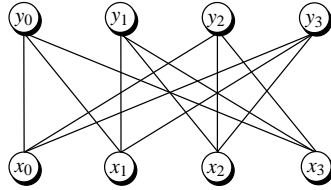
For simplicity we look at the case $d = 3$ but generalization is straightforward.



3-Regular Constraint Graph

- Let n be the number of variables in the instance.

A constraint graph $G = (V, E)$ with bipartition (X, Y) gives the occurrences of variables $X = \{x_0, x_1, \dots, x_{n-1}\}$ in equations $Y = \{y_0, y_1, \dots, y_{n-1}\}$.



- A graph is 3-regular if every node has exactly three neighbors.
- A 3-regular constraint graph G is selected uniformly at random from the set of all 3-regular graphs with bipartition (X, Y) .

A System of Linear Equations

- Now we have a system of linear equations

$$A\vec{x} \equiv \vec{b} \pmod{2}$$

where $\vec{x} = (x_0, x_1, \dots, x_{n-1})$ is a column vector of variables.

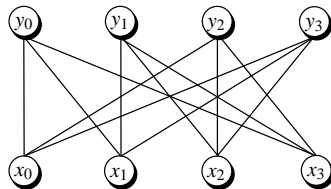
- Note: by construction $A\vec{z} \equiv \vec{b} \pmod{2}$
 the system always has at least one solution
- If a unique solution is required, then the matrix A must be invertible modulo 2.



A System of Linear Equations

- For a given G , construct a corresponding system of linear equations $A\vec{x} \equiv \vec{b} \pmod{2}$ as follows:
- Construct matrix A according to constraint graph G :

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$



- Choose \vec{b} as follows:
 - Select uniformly at random $\vec{z} \in \{0, 1\}^n$
 - Let $\vec{b} \in \{0, 1\}^n$ so that $\vec{b} \equiv A\vec{z} \pmod{2}$.

Example.

- For A and a randomly chosen vector \vec{z} , we get \vec{b} :

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad \vec{z} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \rightsquigarrow \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- The corresponding equation system:

$$\begin{cases} x_0 + x_1 + x_3 = 0 \\ x_1 + x_2 + x_3 = 0 \\ x_0 + x_2 + x_3 = 1 \\ x_0 + x_1 + x_2 = 0 \end{cases} \pmod{2}$$



Transformation to CNF

- Done using no extra variables
- For every equation $x_{j_1} + x_{j_2} + x_{j_3} \equiv b_i \pmod{2}$ we take four clauses forbidding the combinations of truth values violating the equation.

$$\begin{aligned}x_0 + x_1 + x_3 &= 0 && \{\{\bar{x}_0, \bar{x}_1, \bar{x}_3\}, \{\bar{x}_0, x_1, x_3\}, \{x_0, \bar{x}_1, x_3\}, \{x_0, x_1, \bar{x}_3\}\}, \\x_1 + x_2 + x_3 &= 0 && \{\{\bar{x}_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, x_2, x_3\}, \{x_1, \bar{x}_2, x_3\}, \{x_1, x_2, \bar{x}_3\}\}, \\x_0 + x_2 + x_3 &= 1 && \{\{\bar{x}_0, \bar{x}_2, x_3\}, \{\bar{x}_0, x_2, \bar{x}_3\}, \{x_0, \bar{x}_2, \bar{x}_3\}, \{x_0, x_2, x_3\}\}, \\x_0 + x_1 + x_2 &= 0 && \{\{\bar{x}_0, \bar{x}_1, \bar{x}_2\}, \{\bar{x}_0, x_1, x_2\}, \{x_0, \bar{x}_1, x_2\}, \{x_0, x_1, \bar{x}_2\}\}.\end{aligned}$$

Limiting the Pruning Power of BCP

- A random regular graph G provides a potential highly connected graph because it has a constant nonzero lower bound for the expansion coefficient $h(G)$:

$$h(G) = \min \left\{ \frac{|\partial_G U|}{|U|} : U \subseteq V, 1 \leq |U| \leq \frac{|V|}{2} \right\}$$

where $\partial_G U$ is the *boundary* of U .

- For such a graph it can be shown that the number of variables determined by BCP is linearly bounded by the number of split variables (choice points).
- Hence, potentially many choice points are required and thus the DPLL search tree will be large.



Why Regular XORSAT Difficult for DPLL?

- Regular XORSAT uses similar constructions as previous XORSAT families such as
 - random XORSAT (Ricci-Tersenghi at al.) and
 - “spin glass” XORSAT (Jia et al.)
- The key difference is the use of a random 3-regular constraints graph
- The idea is to limit the pruning power of Boolean Constraint Propagation (BCP) by choosing a *constraint graph that is highly connected*.

Introducing Nonlinearity

- XORSAT instances are challenging for DPLL and local search.
- However, linear equations $\pmod{2}$ can be solved in polynomial time using Gaussian elimination.
- For solvers with equivalence reasoning techniques we need schemes to make instances of XORSAT gradually more and more challenging.
- A solution: schemes for introducing more and more nonlinearity into the equations.
 - ☞ Naive scheme, covering scheme, ...



Naive Scheme

Idea: Make all linear equations conditional on a single variable x .

- Introduce three new variables x, y, z ,
- insert the literal x into each original clause,
- add 7 clauses to force x to 0 and y, z into unique values

Benchmarking simple equivalence reasoning techniques (such as preprocessing): detecting the clausal representation of a set of linear equations conditional on a single variable x .



Covering Scheme

Idea: add a nonlinear term ($x \wedge y$) to every equation

- Select a minimal set of variables such that every clause contains at least one selected variable.
- For each selected variable, x , introduce a new variable, y , and then substitute each occurrence of x (respectively, \bar{x}) in the clauses with $x \wedge y$ (respectively, $\overline{x \wedge y} \equiv \bar{x} \vee \bar{y}$).
- After all the substitutions have been performed, expand any conjuncts inside disjuncts to obtain a set of clauses.

For benchmarking dynamic equivalence reasoning techniques that are applied *during search*.



Generalizations

These two basic schemes can be extended as follows

■ k -Nonlinear Covering Scheme:

- Instead of introducing one new variable y for each selected variable x , introduce k new variables y_1, \dots, y_k for each such x .
- Substitute each occurrence of x (respectively, \bar{x}) in the clauses with $x \wedge y_1 \wedge \dots \wedge y_k$ (respectively, $\overline{x \wedge y_1 \wedge \dots \wedge y_k} \equiv \bar{x} \vee \bar{y}_1 \vee \dots \vee \bar{y}_k$), and expand any resulting conjuncts to obtain clauses.



Generalizations

■ p -Covering Scheme.

Select a minimal set of variables such that $p\%$ of the clauses contain at least one selected variable. Apply the k -nonlinear covering scheme on these selected variables only.

■ p -Mixed Covering/Naive Scheme.

As in p -covering scheme, but additionally apply the naive scheme for the remaining $(100 - p)\%$ clauses not containing any selected variables.



Experiments

Two sets of experiments:

- Evaluate the hardness of the basic 3-regular XORSAT against other satisfiable families of instances
- Comparing solvers with equivalence reasoning techniques

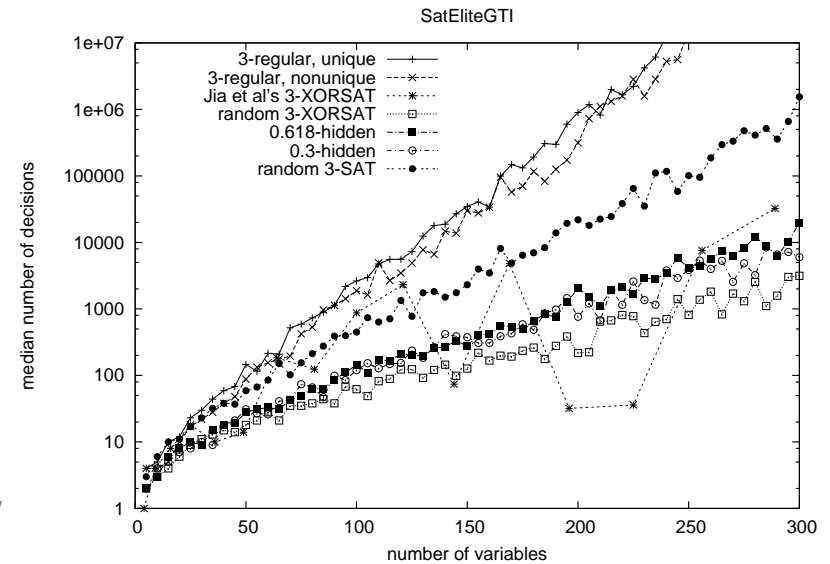


Hardness of basic 3-regular XORSAT

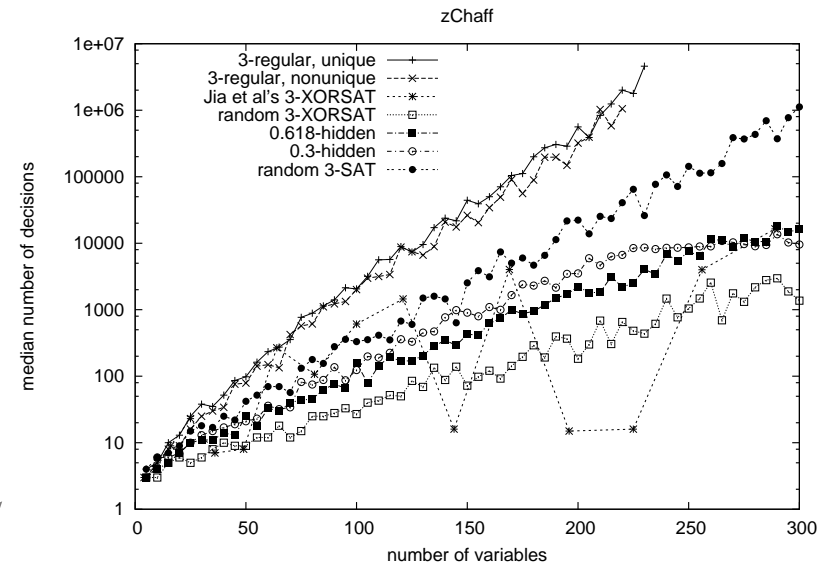
- Evaluate the hardness of the basic construction against other satisfiable families of benchmarks
 - 3-regular, unique
 - 3-regular, nonunique
 - random 3-XORSAT (at the phase transition point)
 - Jia et al's 3-XORSAT ("spin glass")
 - random 3-SAT (at the phase transition point)
 - q -hidden ("deceptive" satisfiable random 3-SAT)
- using complete solvers:
SatEliteGTI, zChaff, and Satz
- local search solvers:
WalkSAT and AdaptiveNovelty+



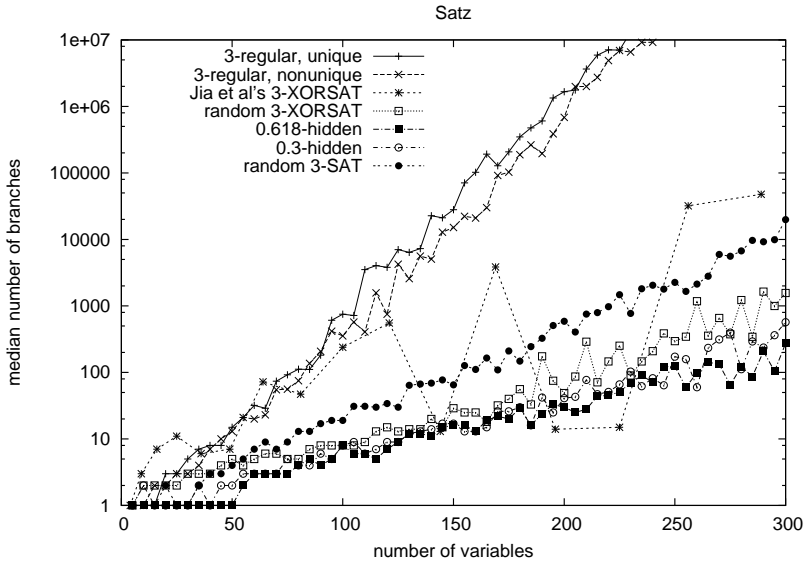
Hardness: SatEliteGTI



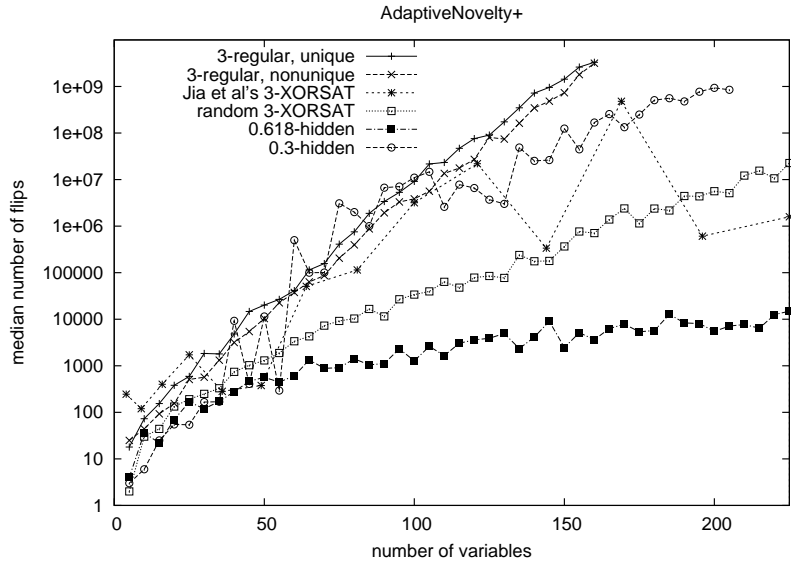
Hardness: zChaff



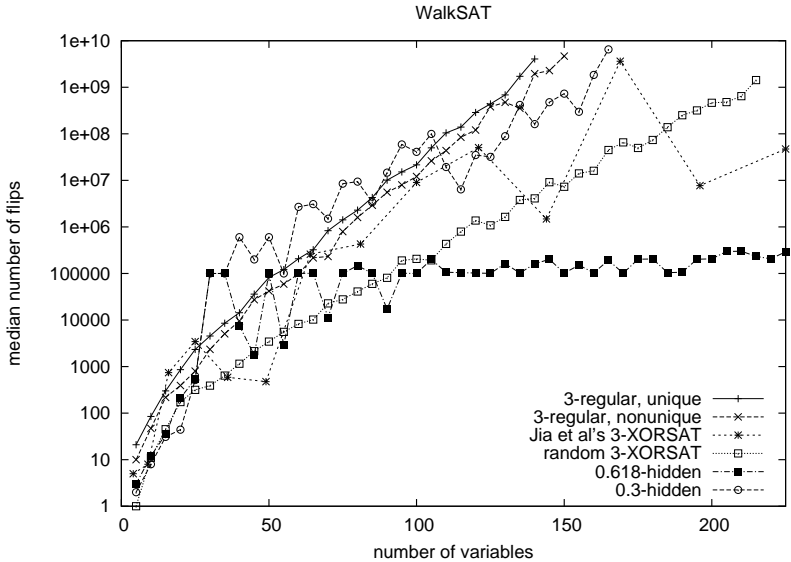
Hardness: Satz



Hardness: AdaptiveNovelty+



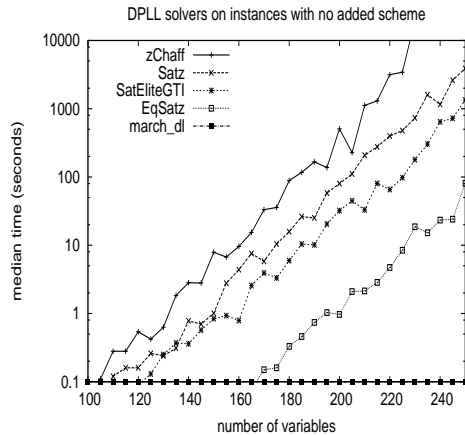
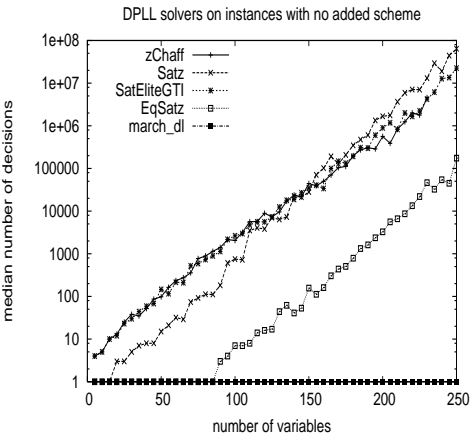
Hardness: WalkSAT



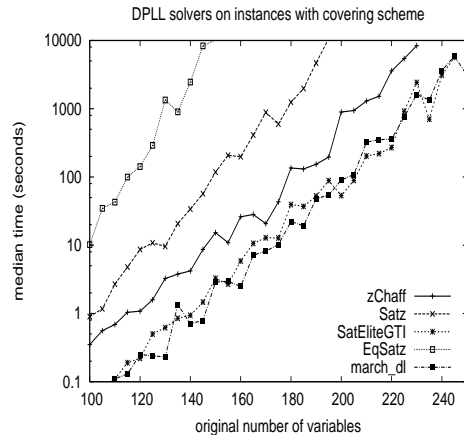
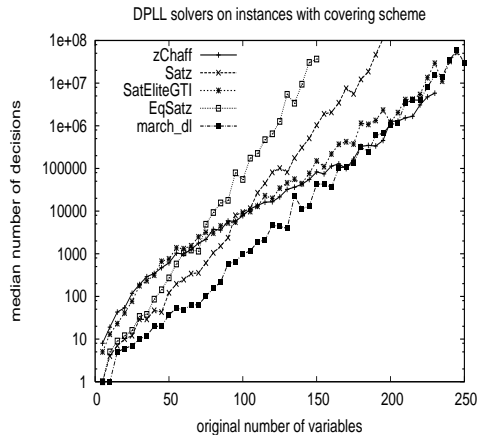
Equivalence Reasoning

- Compared SatEliteGTI, zChaff, and Satz
- against solvers with equivalence reasoning techniques: march_dl and EqSatz
- Measured time and number of decisions
- Three settings
 - Basic 3-regular XORSAT
 - Nonlinearity using the naive scheme
 - Nonlinearity using the covering scheme

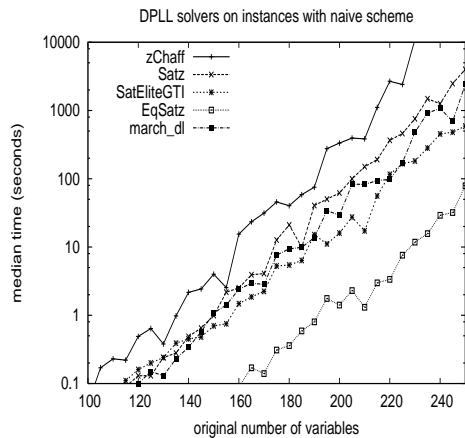
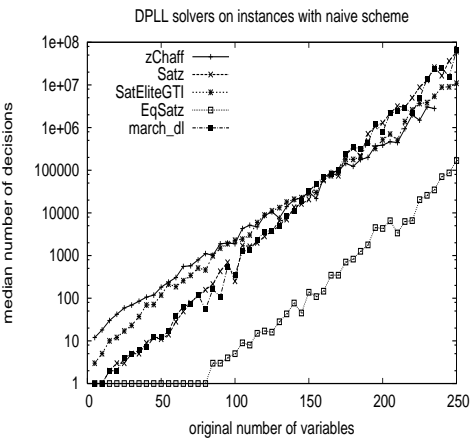
Basic 3-regular XORSAT



Covering Scheme



Naive Scheme



Conclusions

- Regular XORSAT is an interesting family of guaranteed satisfiable instances for benchmarking.
- Number of variables works as a control parameter for the difficulty of an instance.
- Instances of the same size have similar computational characteristics.
- It is easy to generate large numbers of guaranteed satisfiable test cases of the same size.
- Already fairly small test cases are challenging
 - ☞ In SAT Competition 2005 regular XORSAT instances (see [crafted/jarvisalo05](#)) were the smallest guaranteed satisfiable instances not solved in the second stage (smallest had 322 variables).



Conclusions

- For experimental work on the case of d -regular XORSAT with $d > 3$, see (Järvisalo, AAI-2006)
- Current equivalence reasoning techniques work quite well for pure XORSAT
- However, in practice problems are mixed
 - Hardware verification
 - Cryptanalysis: stream ciphers, block ciphers, . . .
- Current techniques do not yet scale up to this case.
- A possible solution:
XOR formulas are not classified
☞ *SMT solver with linear equations mod 2 theory*
See, (Baumgartner/Massacci, CL2000).



References

- Harri Haanpää, Matti Järvisalo, Petteri Kaski, and Ilkka Niemelä. Hard Satisfiable Clause Sets for Benchmarking Equivalence Reasoning Techniques. Journal on Satisfiability, Boolean Modeling and Computation. Volume 2 (2006), 27-46.
- Matti Järvisalo. Further Investigations into Regular XORSAT. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06), pages 1873-1874. AAAI Press, 2006.
- Generators for regular XORSAT:
<http://www.tcs.hut.fi/~mjj/benchmarks/>

