Hard SAT and CSP instances with Expander Graphs *

Carlos Ansótegui, Ramón Béjar, César Fernàndez, Carles Mateu
{carlos,ramon,cesar,carlesm}@diei.udl.cat,
Dept. of Computer Science, Universitat de Lleida, SPAIN

Abstract

In this paper we provide a new method to generate hard k-SAT instances. Basically, we construct the bipartite incidence graph of a k-SAT instance where the left side represents the clauses and the right side represents the literals of our Boolean formula. Then, the clauses are filled by incrementally connecting both sides while keeping the girth of the graph as high as possible. That assures that the expansion of the graph is also high. It has been shown that high expansion implies high resolution width w. The resolution width characterizes the hardness of an instance F of n variables since if every resolution refutation of F has width w then every resolution refutation requires size $2^{\Omega(w^2/n)}$. We have extended this approach to generate hard n-ary CSP instances. The experimental investigation conducted on complete and incomplete solvers confirms that the expansion of the graph is indeed a key factor in order to obtain harder instances than other approaches.

1 Introduction

Providing challenging benchmarks for the SAT and the CSP problems is of a great significance for both the experimental evaluation of SAT and CSP solvers and for the theoretical computer science community. Every year new benchmarks are submitted to the SAT and CSP competitions. Our aim is to provide a method for generating hard k-SAT and n-ary CSP instances.

In order to do that we look at the field of propositional proof complexity, where it turns out that graph expansion has been established as a key to hard formulas for resolution (e.g. (Ats04)), but also for other proof systems like the polynomial calculus (AR01). Roughly speaking, an expander graph is a graph $G=(V,E)$ that, for any, not too big, subset of vertices S, its set of neighbors in $V \setminus S$ is big, compared with $|S|$. In other words, if we view G as a network transmitting information (where information retained by some vertex propagates, say in 1 unit of time, to neighboring vertices), then the expansion measures the quality of G as a communication network. If the expansion is high, information propagates well (DSV03).

Basically, our approach is based on creating a bipartite graph with a high expansion, and then from this graph we generate the k-SAT and n-ary CSP instances. In particular, for the k-SAT instances one of the partitions of the graph represents the set of clauses and the other one the set of literals. Edges represent which literals belong to which clauses. We call this graph the literal incidence graph of a SAT instance. Analogously, for the CSP instances on partition represents nogood tuples and the other one pairs (variable, value).

The way our method tries to get a high expansion on the bipartite graph is to incrementally build the graph while keeping the girth as high as possible. The girth is the length of the shortest cycle of the graph. It is known that high girth implies high expansion (Kah93).

The instances we generate with this method can be used to test the efficiency of SAT and CSP solvers. Moreover, expander graphs have many other applications, like efficient communication networks (Chu78; KM06) and linear-time decodable low density parity check codes (SS96). A very interesting recent one is the use of expander graphs to define secure cryptographic hash functions, (i.e. that are collision resistant), where particular families of expander graphs are considered as candidates to build hash functions due to their high expansion and high girth (CGL07).

We have compared our approach against other methods in the SAT Community (BS96; BDIS05) which try to get hard SAT instances by balancing the occurrences of literals, and thus the degrees of the vertices at the literal incidence graph become also balanced. Previous results, e.g. (SS96), show that balanced bipartite graphs also tend to have a high expansion. Our empirical results confirm that our method generates harder instances. Besides, we get a higher lower bound on the expansion, which we compute using spectral graph theory methods. Computing the expansion of a graph is a co-NP Complete problem (BKO$^+$81), so it is intractable to get the exact expansion.

In the CSP field there are four standard methods,

*Research partially supported by projects TIN2004-07933-C03-03, TIN2006-15662-C02-02 funded by the *Ministerio de Educación y Ciencia*.

denoted A, B, C and D, for generating hard random binary CSPs (SD96; GMP+01). In (AKK+97) the model E was introduced in order to overcome some deficiencies of the previous models. For n-ary CSPs some extensions of random binary CSPs models have been defined (XBHL05). At the section of experimental results we compare our method against the n-ary version of Model E, because the set of parameters in model E (domain size, number of variables and total number of nogoods) is the same as in our high-girth model, thus giving a natural comparison.

The rest of the paper is organized as follows. Section 2 introduces a set of previous definitions. Section 3 discusses the related work. Section 4 presents our method for generating hard k-SAT and n-ary CSP instances. Finally section 5 shows the experimental investigation on SAT and CSP solvers.

2 Preliminaries

We first introduce some definitions from graph theory in order to explain the graph expansion concept.

Definition 1 *An* undirected graph *G is a pair (V, E) where V is the set of vertices and E is the set of undirected edges $\{u, v\}$. The* degree *$d(u)$ of a vertex u is the number of edges with an endpoint in u. A* k-regular graph *is a graph where the degree of any vertex is k.*

Definition 2 *A* bipartite graph *G is a pair $(L \cup R, E)$, where L is the left partition and R is the right partition of the set vertices, such that any edge is of the form (l, r) with $l \in L$ and $r \in R$. A* (k_1, k_2)-regular bipartite graph *is a bipartite graph $(L \cup R, E)$ such the degree of any l from L is k_1 and the degree of any r from R is k_2. Observe that $|L|k_1 = |R|k_2$. We have a $(k_1, -)$-regular bipartite graph if we only fix the degree of vertices in L to k_1, but the degrees for R are unfixed.*

Definition 3 *The girth of a graph G ($g(G)$) is the length of the shortest circuit in G. If G is acyclic then, by definition, $g(G) = \infty$.*

There is a limit on how large the girth can be, for a graph with V vertices and minimum degree d. This limit is $2 \log_{d-1}(|V|)$ (DSV03).

Definition 4 *We say that a family F of k-regular graphs has high girth if, for some constant $0 < C < 2$, $\forall G \in F$, $g(G) \geq (C + o(1)) \log_{k-1} |V|$.*

Random k-regular graphs have an expected girth slightly greater than 3 (MWW04), but there exist constructions of graphs with high girth. The one with the highest girth is that of (ALS88) where they achieve girth $(4/3) \log_{k-1}(|V|)$.

Definition 5 *The expansion of a subset $X \subseteq V$ in $G = (V, E)$ is defined to be the ratio $|N(X)|/|X|$, where $N(X) = \{w \in V \setminus X \mid \exists v \in X, \{v, w\} \in E\}$ is the set of outside neighbors of X.*

When all the neighbors of X are inside X, we have expansion 0. We consider a set high expanding when its expansion is greater than 1, that means that the set of different outside neighbors of X is larger than X, so it is well connected with the rest of the graph.

Definition 6 *An* (α, c)-expander *is a graph (V, E) such that every subset of size at most $\alpha|V|$ has expansion at least c.*

Usually, smaller sets will have better expansion, the limit being for $\alpha \geq 0.5$, where expansion cannot be greater than 1. For bipartite graphs we are mainly interested on the expansion of subsets of the left part. So, we have the next definition.

Definition 7 *A* left (α, c)-expander *is a bipartite graph $(L \cup R, E)$ such that every subset of L of size at most $\alpha|L|$ has expansion at least c.*

Next, we give the definitions of the SAT and CSP problems.

Definition 8 *A* constraint satisfaction problem (CSP) *instance is defined as the triplet $\langle X, D, C \rangle$, where $X = \{x_1, \ldots, x_n\}$ is a set of variables, $D = \{d(x_1), \ldots, d(x_n)\}$ is a set of domains containing the values the variables may take, and $C = \{C_1, \ldots, C_m\}$ is a set of constraints. Each constraint $C_i = \langle S_i, R_i \rangle$ is defined as a relation R_i over a subset of variables $S_i = \{x_{i_1}, \ldots, x_{i_k}\}$, called the* constraint scope. *The relation R_i may be represented extensionally as a subset of the Cartesian product $d(x_{i_1}) \times \cdots \times d(x_{i_k})$. Elements $\in R_i$ are called* good *tuples, and elements $\in ((d(x_{i_1}) \times \cdots \times d(x_{i_k})) \setminus R_i)$ are called* nogood *tuples.*

Definition 9 *An* assignment *v for a CSP instance $\langle X, D, C \rangle$ is a mapping that assigns to every variable $x_i \in X$ an element $v(x_i) \in d(x_i)$. An assignment v satisfies a constraint $\langle \{x_{i_1}, \ldots, x_{i_k}\}, R_i \rangle \in C$ iff $\langle v(x_{i_1}), \ldots, v(x_{i_k}) \rangle \in R_i$.*

Definition 10 *Propositional variables are denoted $p_1, \ldots, p_n$ and can be assigned truth values 0 (or F) or 1 (or T). A* literal *is an expression of the form p_i or $\neg p_i$, where p_i is a propositional variable. The* complement *of a literal l of the form p_i ($\neg p_i$), denoted by $\overline{l}$, is $\neg p_i$ (p_i). A* clause *is a disjunction of literals. A* CNF formula *is a conjunction of clauses.*

Definition 11 *A* truth assignment *for a CNF formula is a mapping that assigns to every propositional variable to value T or F. An truth assignment I* satisfies *a literal p_i ($\neg p_i$) iff $p_i = T$ ($p_i = F$),* satisfies *a clause C iff it satisfies at least one of the literals in C, and* satisfies *a CNF formula Γ iff it satisfies all clauses in Γ. The SAT problem consists of deciding whether there exists a truth assignment to the variables such that the formula becomes satisfied.*

For this work, the following three concepts are the main tools used to link complexity with structural properties of k-SAT and n-ary CSP instances.

Definition 12 *Given a k-SAT instance F with set of clauses C, set of variables V and set of literals L, $G(F) = (C \cup V, E)$ is its bipartite variable incidence graph such that $(c, v) \in E$ if and only if variable v appears in clause c. $LG(F) = (C \cup L, E)$ is its bipartite literal incidence graph such that $(c, l) \in E$ if and only if literal l appears in clause c.*

Observe that if $LG(F) = (C \cup L, E)$ is a left (α, c)-expander, then $G(F) = (C \cup V, E)$, will be, at least, a left $(\alpha, c/2)$-expander.

Definition 13 *Given a CSP instance $P = \langle X, D, C \rangle$, we define the literal incidence graph as the bipartite graph $LG(P) = (NG \cup L, E)$, where for every constraint C_i and every nogood tuple associated with C_i there is a vertex $\in NG$, and vertices from L represent all the pairs (x_i, j), where $x_i \in X$ and $j \in d(x_i)$. Edges represent the pairs (x_i, j) associated which each nogood tuple.*

3 Related Work

In this section we survey some previous theoretical results about the expansion of random graphs, and the related work in the SAT and CSP communities.

3.1 Expansion of random graphs

The problem of checking whether a graph is an expander is co-NP complete (BKO$^+$81). However, lower and upper bounds on the expansion of a graph have been obtained using spectral graph theory results. Given the adjacency matrix $A(G)$ of $G = (V, E)$ we denote their eigenvalues by $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{n-1}$, where $n = |V|$. When G is k-regular, then $\lambda_0 = k$. If G is connected then $\lambda_0 > \lambda_1$.

The (combinatorial) Laplacian $L(G)$ is the matrix $D - A(G)$, where D is the diagonal matrix in which $D_{i,i}$ is the degree of v_i. We denote the eigenvalues of $L(G)$ by $\mu_0 = 0 \leq \mu_1 \leq \ldots \leq \mu_{n-1}$. When G is k-regular, then we have that $\mu_i = k - \lambda_i$. But in general, the two sets of eigenvalues can be very different. For non regular graphs, the spectrum of $L(G)$ provides more information about the connectivity of the graph. For example, for general graphs we have the following expansion lower bound (CS03):

$$\frac{|N(X)|}{|X|} \geq \frac{4\mu_1(1-\alpha)}{\Delta + 4\mu_1\alpha} \quad (1)$$

where $\alpha = |X|/|V|$. Observe that for $\alpha \leq 0.5$, the higher μ_1 is, the higher this lower bound is. There are also upper bounds on expansion based on the eigenvalues of the *normalized* Laplacian of the graphs (Chu96).

For k-regular graphs, we have more precise expansion lower bounds that depend on λ_1, such that the lower λ_1 is the higher the expansion. Asymptotically (as $n \to \infty$), there is a limit $(2\sqrt{k-1})$ on how small λ_1 can be for a family of k-regular graphs (AB88). The value $\lambda_0 - \lambda_1$ is called the spectral gap, and asymptotically for k-regular graphs the best we can hope for this value is to tend to $k - 2\sqrt{k-1}$. Friedman (Fri03; Fri04) shows that for random k-regular graphs and any $\epsilon > 0$:

$$lim_{n\to\infty} Pr(\lambda_1 \leq 2\sqrt{k-1} + \epsilon) = 1$$

k-regular graphs with $\lambda_1 \leq 2\sqrt{k-1}$ are called Ramanujan graphs (ALS88). So, asymptotically, a random k-regular graph will get very close to be a Ramanujan graph. Kahale (Kah95) gave an expansion lower bound that shows that Ramanujan k-regular graphs can have expansion as high as $k/2$ for small sets. Thus, Ramanujan graphs (and random k-regular graphs) with $k \geq 3$ are excellent expander graphs.

Independently of what can be proved thanks to eigenvalue methods, probabilistic methods have been used to show that regular graphs are almost surely very good expanders (see for example (SS96; Bol01; AS00)). The particular case of k-regular or (k_1, k_2)-regular bipartite graphs have received special attention in the communications community (e.g. (Chu78; SS96; Tan84)), and such bipartite graphs are good expanders almost always. By contrast, we do not have similar results for sparse $G(n, p)$ and $G(n, M)$ graphs. So, it seems that regular graphs are more promising towards obtaining good expanders, although we will see that *almost* regular graphs can also be excellent good expanders, even better than regular graphs. For the case of bipartite graphs we have, for example, that a random $(k, -)$-regular bipartite graph $(L \cup R, E)$ with $|L| = |R|$ will be a good expander *with probability* $> 1/2$. So, when only the vertices of one part have the same degree, the expansion properties seem to degrade. Observe that this last graph can represent the incidence graph of a random k-SAT instance.

3.2 Theoretical results in SAT community

Concerning the resolution complexity of a 3SAT instance F, Ben-Sasson and Wigderson (BSW01) proved that if every resolution refutation of F requires width w, then every resolution refutation of F requires size $2^{\Omega(w^2/n)}$. The width of a resolution refutation is the length of the longest clause in the refutation. Thus, lower bounds on width imply lower bounds on size. Finally, there is a connection between graph expansion and 3SAT resolution complexity based on this width-size relationship. Consider a 3SAT instance F with set of clauses C and set of variables V and its bipartite variable incidence graph $G(F) = (C \cup V, E)$. Results presented in (Ats04) imply that any resolution refutation will have width lower bounded by $\lfloor((c-1)\alpha|C|)/((2+c)d)\rfloor$, where d is the maximum right-degree of $G(F)$, if $G(F)$ is a left (α, c)-expander. So, any resolution refutation of F will have exponential size if $d = o(|C|)$ and $c > 1, \alpha > 0$, given the width-size relationship. So, the higher the expansion of the graph and the smaller the maximum right-degree d, the higher the refutation size lower bound.

The result is basically established through a connection between two different combinatorial games, the

matching game and the existential $k-$pebble game, and trough a connection between the $k-$pebble game and the width of resolution proofs. To summarize, the results indicate that the higher the expansion, the higher the number of fingers needed to win the matching game, and the higher the number of pebbles k needed to win the k-pebble game. Then, by considering the encoding of 3SAT as a CSP problem, Atserias connects the needed number of pebbles k with the width of resolution proofs thanks to the relation between $k-$Datalog programs and the $k-$pebble game (KV95).

Moreover, the results also imply that more powerful proof algorithms based on strong k-consistency (for a bounded level of consistency) will also require exponential time for solving the 3SAT instance under the same circumstances.

3.3 Related results in CSP community

For binary CSPs, in (ABFM07) new methods for generating hard instances were presented, based on balancing both the constraint language and the constraint graph. Also, a method for generating a high girth constraint graph was introduced and it generated the hardest instances. In that work they link the hardness of the instances to the fact that more balanced graphs tend to have a higher treewidth, thanks to the result of (CS03) were the treewidth is linked with the graph expansion. Previous work has considered the generation of hard balanced CSPs (see for example (KRA$^+$01; ABF$^+$06)), but without linking the balance of the constraint graphs to their treewidth.

Given the relation between existential $k-$pebble games for CSPs and strong k-consistency presented in (KV00), we can reasonably think that there is a relation between the expansion of the CSP incidence graph and the level of $k-$consistency needed to solve the CSP. That would be a similar result to the one we have mentioned at the previous SAT section. As we will see at the section of experimental results, our method increases the hardness of n-ary CSPs.

4 Hard SAT and n-ary CSP instances

In this section we introduce our generation method for hard k-SAT and n-ary CSP instances.

4.1 Expansion, balance and girth

To get an idea about what is the typical structure of a good expander graph, consider the expansion of subsets of the left partition of the two bipartite graphs of Figure 1. As the vertices in the left partition of both graphs have degree 3, the expansion when $|S| = 1$ is 3. Consider now sets with $|S| = 2$. In the graph (a), the set $N(S)$ for any left subset S with $|S| = 2$ is always the entire right partition, so the expansion is 4/2. But for graph (b) the set $N(\{1, 4\})$ does not contain the vertex 7, and so the expansion is only 3/2 due to the poor connectivity of vertex 7. For $|S| = 3$ the situation

(a) $(3, 3)$-regular (b) $(3, -)$-regular

Figure 1: Example of two bipartite graphs with different expansion

is similar. For graph (a) any left subset with $|S| = 3$ is connected to the whole right partition (its expansion is 4/3), but for graph (b) $N(\{2, 1, 4\}) = \{5, 6, 8\}$, so the expansion is 1. Thus, we observe that, due to the unbalanced degrees of the right partition of graph (b), the vertex expansion of the left subsets is not as good as in graph (a), where all the degrees are equal.

However, the balance of the degrees does not provide a complete characterization of good expander graphs. Kahale (Kah93) shows that high girth ($O(\log_{k-1}(|V|))$ implies high expansion, at least for subsets of size at most $|V|^\delta$, with $\delta < 1$. So, one way to obtain graphs with good expansion is to get high girth graphs. In (Cha03) it is presented an algorithm for building graphs with degrees $k - 1$, k and $k + 1$ and high girth. The algorithm we present in the next subsection follows the same approach to build bipartite graphs with high girth. This graph will be used to build hard k-SAT and n-ary CSPs instances .

4.2 High girth bipartite graphs

The algorithm presented in (Cha03) works for general (non-bipartite) graphs. It builds the graph by introducing edges one by one, connecting vertices which are at large distances in the current graph, in such a way that the degrees are maintained almost balanced and the girth obtained is $O(\log_{k-1}(|V|))$. The algorithm initiates the construction by building a matching between the vertices, if $|V|$ is even, or a $|V|$-length cycle, if $|V|$ is odd.

For building the literal incidence graph of a k-SAT formula with C clauses and L literals (and similarly for a k-ary CSP formula), we need to build a $(k, -)$-regular bipartite graph $(V_1 \cup V_2, E)$, where V_1 is C and V_2 is L. The algorithm of Figure 1 does this, but trying to keep the girth as high as possible, using the same technique of linking vertices which are at large distances in the current graph. It starts the process by creating a random matching from V_1 to V_2, such that every vertex from V_1 will have degree 1 and every vertex from V_2 will have degree either $\lfloor |V_1|/|V_2| \rfloor$ or $\lfloor |V_1|/|V_2| \rfloor+1$. Because this matching does not create any cycles, it starts with girth equal to ∞. Then, at every step it selects an edge from the subset of edges (u, v) with $u \in V_1$ and $v \in V_2$, such that $degree(u) < k$ and $degree(v)$ is minimum among all the current degrees in V_2. From this subset of edges, it selects one (u', v') with the maximum distance

between u' and v', because this way the new created cycle is of maximum length. This process ends when the graph has $|V_1|k$ edges.

Additionally, the degrees of the right vertices (V_2) of the bipartite graph will be almost balanced. Observe that, when selecting the edge (u', v'), if we are always able to pick a vertex v' from V_2 of minimum degree, then at the end all the vertices of V_2 will have degree either $\lfloor |V_1|k/|V_2| \rfloor$ or $\lfloor |V_1|k/|V_2| \rfloor + 1$. So, only when is not possible to pick any minimum degree vertex from V_2 the algorithm introduces a third degree. However, this situation can only occur when all the available vertices from V_1 are already linked with all the current minimum degree vertices from V_2, and this can only occur when we are at the very end of the process, i.e. when the number of available vertices in V_1 is very small and the number of remaining minimum degree vertices in V_2 is $< k$ (observe that an available vertex from V_1 can be linked at the same time with at most $k-1$ vertices from V_2). Actually, the instances we have obtained with this method almost always have only two distinct degrees in V_2, and only exceptionally three distinct degrees.

Algorithm 1: Algorithm for generation of high girth $(k, -)$-regular bipartite graphs $(V_1 \cup V_2, E)$

input : V_1, V_2, k
output: a bipartite $(k, -)$-regular graph $(V_1 \cup V_2, E)$
Initialize E with a random matching from V_1 to V_2 (every vertex from V_1 will have degree 1)
for $i = |V_1| + 1$ **to** $k|V_1|$ **do**
  $L_T := \{u \in V_1 \mid degree(u) < k\}$
  $R_T := \{u \in V_2 \mid degree(u) \leq degree(v), \forall v \in V_2\}$
  $P := \emptyset$
  while $P = \emptyset$ **do**
    $T := \{(u, v) \mid (u, v) \in L_T \times R_T \; distance(u, v) \geq distance(x, y) \forall (x, y) \in L_T \times R_T\}$
    $d_{min} := degree(u)$, where $u \in R_T$
    $P := \{(u, v) \in T \mid (u, v) \notin E\}$
    if $P \neq \emptyset$ **then**
      randomly select a pair (u, v) from P
      $E := E \cup (u, v)$
    else
      $R_T := \{u \in V_2 \mid degree(u) = d_{min} + 1\}$

Regarding the girth, our empirical results show that it is of the order of $\log_{\overline{k}-1}(|V|)$, where $\overline{k}$ is the average degree of the graph, so it is comparable to the girth achieved by the algorithm for general graphs.

Observe that not every subset of k vertices from V_2 gives a valid clause of k literals, since if two selected literals share the same variable, we get a tautological clause. However, the number of tautological clauses is as low as the expected number obtained with a random k-SAT model where tautological clauses are not excluded, that is $O(|C|/|V|)$. So, since the hardest instances occur always around a fixed ratio $|C|/|V|$, that depends on k, we simply discard the tautological clauses obtained. For the literal incidence bipartite graph of k-ary CSPs, we have an analogous situation. Not every subset of k vertices from V_2 give a valid $k-$ary nogood. A valid subset cannot contain two vertices of the form (x_i, j) and (x_i, l) with $j, l \in d(x_i)$. So, invalid nogoods are also discarded.

5 Experimental investigation

We have divided our experimental investigation into three sections. The first one presents a comparison of our method against the most recent k-SAT generators and the classical random k-SAT generator. The second one shows a comparison with between model E and our method high-girth.

Finally, we have checked whether our generation method produces instances which incidence graph have a higher expansion than the other generators used in the experimental investigation.

5.1 Hard k-SAT instances

For generating the k-SAT instances we have used four methods: the classical random k-SAT (random), the method described in (BS96)(lit-bal-1), the method described in (BDIS05)(lit-bal-2), and our method (high-girth).

The generation method lit-bal-1 described in (BS96) for 3-SAT problems can be generalized to k-SAT as follows. Inputs are the number of variables (n), the number of clauses (m) and the clauses arity (k). There are $2n$ possible literals given n variables, so $\lfloor \frac{k \cdot m}{2n} \rfloor$ occurrences of each literal are placed in a bag. A random set of unique literals is then added to the bag so that there are exactly $k \cdot m$ literals in it. To construct each clause, k literals on distinct variables are removed from the bag. If there are less than k distinct variables mentioned in literals remaining the bag, additional distinct variables are randomly selected from the set of all variables and negated with probability $\frac{1}{2}$. The generation method lit-bal-2 described in (BDIS05) is very similar to lit-bal-1, being the main difference that with lit-bal-2 every literal in the resulting formula appears exactly $\lfloor \frac{k \cdot m}{2n} \rfloor$ or $\lfloor \frac{k \cdot m}{2n} \rfloor + 1$ times. By contrast with lit-bal-1 the occurrences of literals can be less balanced.

We have solved the instances with four SAT solvers: satz (LA97), minisat (ES03), kcnfs (DD01) and WalkSAT (SKC94).

Figure 3 shows the results for the complete SAT solver kcnfs on 4-SAT and 5-SAT instances. As we can see high-girth is the best generator, while lit-bal-1 and lit-bal-2 are almost identical. Observe that the differences are more significative for 5-SAT. This can be due to the differences in the expansion of the bipartite graphs of the different models, because as we increase k is possible to obtain more drastic differences in the expansion of the bipartite graphs of the different models. That is, the higher k, the higher the maximum size of the set of neighbours $N(S)$ of a subset of clauses S. We only report the results for the SAT solver kcnfs since

it was the fastest and it reported the least difference between the two best generators. For example, for satz solver we can see in Figure 2 results for 4-SAT instances with 130 variables. As with kcnfs high-girth is also the best generator, but performance of satz solver is worse than that of kcnfs.

We also wanted to check if we could observe the same behavior when using a local search SAT solver. We filtered out the unsatisfiable instances from sets of 100 instances. Table 1 reports results for the random, lit-bal-1 and high-girth generation methods on 3-SAT instances located at the underconstrained and the phase transition zones and solved with the local search SAT solver WalkSat. We tuned the noise parameter for the heuristic *best* to be 30. As we can see, high-girth and lit-bal-1 behave similarly and clearly outperform the random generation method.

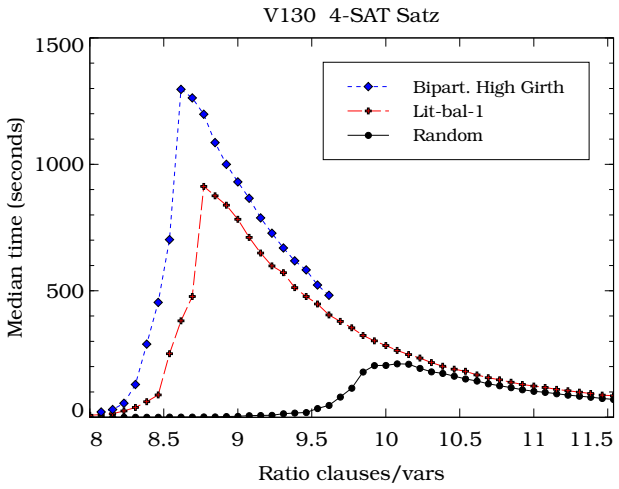

Figure 2: Results for 4-SAT with satz for 130 variables.

5.2 Hard n-ary CSP instances

For generating the n-ary CSP instances we have used two methods: the model E described in (AKK+97) and our method high-girth. We have solved the n-ary CSP instances with the CSP solver minion (GJM06) using the dynamic heuristic *sdf* (smaller domain first). We also report results on the direct SAT encoding[1] of the n-ary CSP instances for the SAT solvers minisat and kcnfs (some competitive solvers submitted to the CSP competition are built on top of minisat). We have generated two set of instances, one of 25 variables, domain 3, and arity 4, and the other set of 40 variables, domain 3 and arity 3. At Figure 4 we can see again that our generation method high-girth produces the hardest instances. In this figure, the results are shown in log-scale, in contrast with Figure 3 for $k-$SAT, because here the differences are even more significative than in Figure 3.

[1]For more details see (Wal00).

	330 vars			
	C	m	md	# inst.
rand	1340/1420	0.14/2.31	0.01/0.07	99/43
lit	1090/1170	0.09/161	0.00/8.24	100/50
hg	1090/1160	0.16/49	0.06/10.5	100/57

	400 vars			
	C	m	md	# inst.
rand	1620/1720	1/6.38	0.02/0.45	100/36
lit	1304/1404	0.083/283	0.001/30.2	100/76
hg	1304/1404	0.03/385	0.03/46.48	100/48

Table 1: 3-SAT instances, 330 and 400 variables. Results for WalkSat, heuristic = Best, noise = 30. Each instance solved 30 times. No cutoff. *C*, *m*, *md* and *# inst.* stand for clauses, mean, median and number of satisfiable instances, respectively. Both values (x/y) represent the underconstrained and phase transition region.

However, observe that we do not have previous existing balanced models for n-ary CSPs, like lit-bal-1 and lit-bal-2 for $k-$SAT, that are the ones that are closer to our high girth model for $k-$SAT.

5.3 Girth and Expansion

In this subsection we compare the girth of the bipartite literal incidence graph $LG = (C \cup L, E)$ of the k-SAT formulas obtained with the different models.

We also lower bound the left-expansion $|N(S)|/|S|$ of the bipartite graphs using results from spectral graph theory found in (Chu97). As small subsets will expand similarly in all the models (because the left degree is the same for all), where one may find the biggest differences is with the biggest subsets. In particular, the highest possible expansion would be $N(S) = L$ for subsets S of size $|L|/k$. So, we focus on the case $|S| = |L|/k$. The results, obtained for two different C/V ratios, are shown in Table 2. The ratio of the left subcolumn corresponds to the peak of hardness for the high-girth model and the other corresponds to the peak of hardness for the random model. As we can see, the girth computed for our generation method high-girth is higher. Concerning the expansion we also get a higher lower bound. Notice that this result is significant since, as we have already mentioned, in previous results, e.g. (SS96), it is shown that balanced bipartite graphs tend also to have a high expansion. The generation methods lit-bal-1 and lit-bal-2 tend to produce almost balanced bipartite literal incidence graphs. That might suggest that balancing the incidence graph is not enough to get the highest possible expansion. Therefore one should also look at other parameters as the girth.

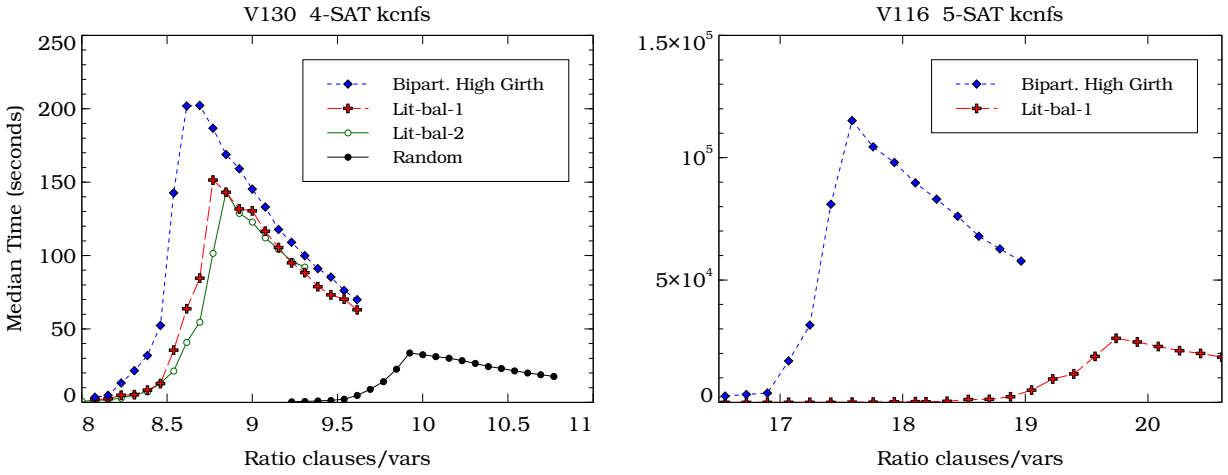

Figure 3: Comparison of SAT generators: random-k-sat, lit-bal-1, lit-bal-2 and high-girth. 3-SAT, 4-SAT and 5-SAT instances of 300, 130 and 116 variables respectively.

	C=1160		C=1420	
	g	$\|N(S)\|$	g	$\|N(S)\|$
random	4	272.6	4	287.7
lit-bal-1	4	272.8	4	288.2
lit-bal-2	4	272.1	4	287.3
high girth	10	283.9	8	303.0

Table 2: Girth and lower bound on expansion $|N(S)|$ for a left subset S of size $|L|/k$ for the bipartite literal incidence graphs of 3SAT instances with 330 variables in the peak of hardness

6 Conclusions

We have proposed a new method for generating hard k-SAT and n-ary CSP instances. This method is based on the results that link problem hardness with the expansion of the incidence graph of the instances. In particular, in our method we achieve high expansion by maintaining a high girth during the construction process of the incidence graph.

References

N. Alon and P. Boppana. Ramanujan graphs. *Combinatorica*, 8:261 – 277, 1988.

C. Ansótegui, R. Béjar, C. Fernández, C. Gomes, and C. Mateu. The impact of balance in a highly structured problem domain. In *Proc. of AAAI06*, pages 438–443. AAAI Press, 2006.

C. Ansótegui, R. Béjar, C. Fernández, and C. Mateu. On balanced CSPs with high treewidth. In *Proceedings of AAAI'2007*, 2007.

D. Achlioptas, L.M. Kirousis, E. Kranakis, D. Krizanc, M. Molloy, and Y.C. Stamatiou. Random Constraint Satisfaction: a More Accurate Picture. In *Proceedings CP'97*, pages 107–120, Linz, Austria, 1997.

R. Phillips A. Lubotzky and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261 – 277, 1988.

M. Alekhnovich and A. Razborov. Lower bounds for polynomial calculus: non-binomial case. In *Proceedings of 42nd Annual Symposium on Fondutations of Computer Science*, pages 190 – 199, 2001.

Noga Alon and Joel H. Spencer. *The Probabilistic Method. Second Edition.* Discrete Mathematics and Optimization. Wiley Inter-Science, 2000.

Albert Atserias. On sufficient conditions for unsatisfiability of random formulas. *Journal of the ACM*, 51(2):281–311, 2004.

Yacine Boufkhad, Olivier Dubois, Yannet Interian, and Bart Selman. Regular random -sat: Properties of balanced formulas. *J. Autom. Reasoning*, 35(1-3):181–200, 2005.

M. Blum, R.M. Karp, O.Vornberger, C.H. Papadimitriou, and M. Yannakakis. The complexity of testing whether a graph is a superconcentrator. *Inf. Process. Letters*, 13(4/5):164–167, 1981.

Béla Bollobás. *Random Graphs. Second Edition.* Number 73 in Cambridge studies in advanced mathematics. Cambridge University Press, 2001.

R.J. Bayardo and Robert Schrag. Using CSP look-back techniques to solve exceptionally hard sat instances. In *CP'96*, pages 46–60, 1996.

E. Ben-Sasson and A. Wigderson. Short proofs are narrow-resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

D. X. Charles, E. Z. Goren, and K. E. Lauter. Cryptographic hash functions from expander graphs. *To appear in Journal of Cryptology*, 2007.

L. Sunil Chandran. A high girth graph construction. *SIAM journal on Discrete Mathematics*, 16(3):366–370, 2003.

F. R. K. Chung. On concentrators, superconcentrators, generalizers and nonblocking networks. *Bell Systems Tech. Journal*, 58:1765–1777, 1978.

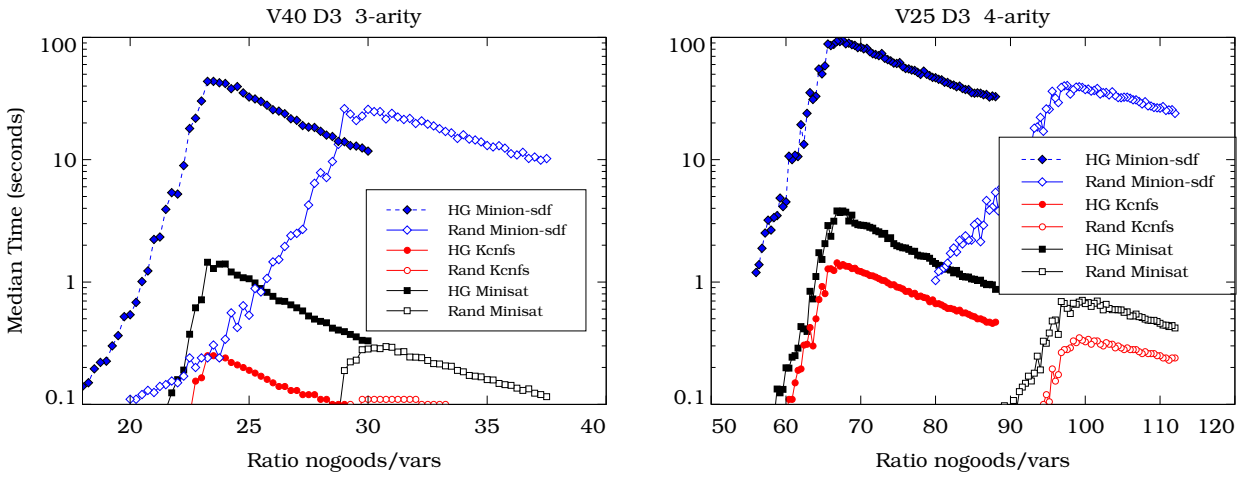

Figure 4: Comparison of CSP generators: Model E and high-girth. 3-ary instances of 40 vars with domain 3, and 4-ary instances of 25 vars with domain 4. CSP solvers: Minion with sdf heuristic and minisat and kcnfs (CSP to SAT direct encoding).

F. R. K. Chung. *Combinatorics, Paul Erdos is eighty*, volume 2, chapter Laplacians of graphs and Cheeger inequalities, pages 157–172. Bolyai Math. Soc., Budapest, 1996.

Fan Chung. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. AMS, 1997.

L. Sunil Chandran and C.R. Subramanian. A spectral lower bound for the treewidth of a graph and its consequences. *Information Processing Letters*, 87(4):195–200, 2003.

Olivier Dubois and Gilles Dequen. A backbone-search heuristic for efficient solving of hard 3-SAT formulae. In *Proc. of IJCAI'01*, pages 248–253, 2001.

Giuliana Davidoff, Peter Sarnark, and Alain Valette. *Elementary Number Theory, Group Theory, and Ramanujan Graphs*. Number 55 in London Mathematical Society Student Texts. Cambridge University Press, 2003.

Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Proc. of SAT'03*, pages 502–518, 2003.

Joel Friedman. A proof of Alon's second eigenvalue conjecture. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 720 – 724, 2003.

Joel Friedman. A proof of Alon's second eigenvalue conjecture and related problems. *Memoirs of the A.M.S.*, 2004.

Ian P. Gent, Christopher Jefferson, and Ian Miguel. Watched literals for constraint propagation in minion. In *Proc. of CP'06*, pages 182–197, 2006.

I. Gent, E. MacIntyre, P. Prosser, B. Smith, and T. Walsh. Random constraint satisfaction: flaws and structure. *Constraints*, 6(4):345–372, 2001.

N. Kahale. *Expander Graphs*. PhD thesis, MIT, 1993.

N. Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42(5):1091–1106, 1995.

S. Kar and J. M. F. Moura. Ramanujan topologies for decision making in sensor networks. In *Forty-Fourth Annual Allerton Conference on Communication, Control and Computing*, 2006.

Henry Kautz, Yongshao Ruan, Dimitris Achlioptas, Carla Gomes, Bart, , and Mark Stickel. Balance and filtering in structured satisfiable problems. In *Proc. of IJCAI-01*, pages 193–200, 2001.

Ph. G. Kolaitis and M. Y. Vardi. On the expressive power of datalog: tools and a case study. *Journal of Computer and System Sciences*, 51:110 – 134, 1995.

Ph. G. Kolaitis and M. Y. Vardi. A game-theoretic approach to constraint satisfaction. In *Proceedings of AAAI'2000*, pages 175 – 181, 2000.

Chu Min Li and Anbulagan. Look-ahead versus look-back for satisfiability problems. In *Proc. of CP'97*, pages 341–355, 1997.

B. D. McKay, N. C. Wormald, and B. Wysocka. Short cycles in random regular graphs. *Elect. J. Combinatorics*, 11:R66, 2004.

B. Smith and M. Dyer. Locating the Phase Transition in Binary Constraint Satisfaction Problems. *Artificial Intelligence*, 81:155–181, 1996.

Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. In *Proc. of AAAI'94*, pages 337–343, 1994.

M. Sipser and D. A. Spielman. Expander codes. *IEEE Trans. on Information Theory*, 43(6):1710–1722, 1996.

R. M. Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal Algorithmic Discrete Mathematics*, 5(3):287–293, 1984.

Toby Walsh. SAT vs CSP. In *Proc. of CP'00*, pages 441–456, 2000.

Ke Xu, Frédéric Boussemart, Fred Hemery, and Christophe Lecoutre. A simple model to generate hard satisfiable instances. In *IJCAI'05*, pages 337–342, 2005.