Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

Generating Hard SAT/CSP Instances Using Expander Graphs*

Carlos Ansótegui, Ramón Béjar, Cèsar Fernández, 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. We incrementally construct a high girth bipartite incidence graph of the k-SAT instance. Having high girth assures high expansion for the graph, and high expansion implies high resolution width.

We have extended this approach to generate hard n-ary CSP instances and we have also adapted this idea to increase the expansion of the system of linear equations used to generate XORSAT instances, being able to produce harder satisfiable instances than former generators.

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. (Atserias 2004)), but also for other proof systems like the polynomial calculus. Roughly speaking, an expander graph is a graph G=(V,E) that, for any, not too big, subset of vertexes S, its set of neighbors in $V \setminus S$ is big, compared with $|S|$.

We have compared our approach against other methods in the SAT Community (Bayardo & Schrag 1996; Boufkhad *et al.* 2005) which try to get hard SAT instances by balancing the occurrences of literals, and thus the degrees of the vertexes at the literal incidence graph become also balanced.

Our empirical results confirm that our method generates harder instances. We have also modified the underlying system of linear equations used in regular k-XORSAT (Järvisalo 2006) by using our High girth bipartite graphs, instead of the original random regular bipartite graphs, and show that the hardness of the instances increases by orders of magnitude.

Preliminaries

Definition 1 *The expansion of a subset X from the vertexes of $G = (L \cup R, E)$ is defined to be the ratio $|N(X)|/|X|$, where $N(X) = \{w \in (L \cup R) \setminus X \mid \exists v \in X, \{v, w\} \in E\}$ 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 2 *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.*

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.

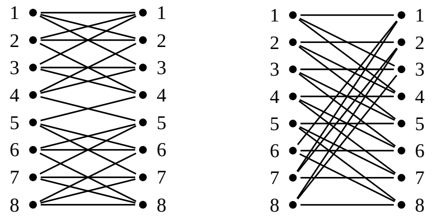
Definition 3 *Given a k-SAT instance F with set of clauses C, set of variables V and set of literals L, $G(F) = (C \cup V, E)$ is its bipartite variable incidence graph such that $(c, v) \in E$ if and only if variable v appears in clause c. $LG(F) = (C \cup L, E)$ is its bipartite literal incidence graph where $(c, l) \in E$ 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.

High girth bipartite graphs

Probabilistic methods have been used to show that regular graphs are almost surely very good expanders.The particular case of k-regular or (k_1, k_2)-regular bipartite graphs have received special attention in the communications community and such bipartite graphs are good expanders almost always. However, the balance of the degrees does not provide a complete characterization of good expander graphs. Consider the graphs (a) and (b) of Figure 1, that are both equally balanced. Graph (a) has several cycles of length 4, and thus girth 4, and its expansion for some left subsets of size 4 is $5/4$. By contrast, in graph (b) the minimum expansion for

*Research partially supported by projects TIN2006-15662-C02-02, TIN2007-68005-C04-02 and José Castillejo 2007 program funded by the *Ministerio de Educación y Ciencia*



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

Figure 1: Bipartite graphs with same balance and different expansion

Table 1: Median time (seconds) for satisfiable 3-SAT, 4-SAT and 5-SAT with High girth, regular XORSAT ($\oplus$-sat) and with High girth-XORSAT (HG-$\oplus$sat) (* denotes more than $2 \cdot 10^4$ s.)

3-SAT						
Num. vars	200	250	270	300	330	350
High girth	0	7	14	91	368	1125
$\oplus$-sat	14	386	2322	19778	*	*
HG-$\oplus$sat	642	*	*	*	*	*

	4-SAT			5-SAT		
Num. vars	100	130	150	80	90	100
High girth	3	59	1180	64	405	2839
$\oplus$-sat	4	201	2543	51	290	2528
HG-$\oplus$sat	66	8018	*	586	3186	*

left subsets of size 4 is $7/4$. The main structural difference with the graph (a) is actually its girth, that in this case is 6.

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 $(C \cup L, E)$. Algorithm 1 does this, but trying to keep the girth as high as possible, using the technique of linking vertexes which are at large distances in the current graph. It starts the process by creating a random matching from C to L, such that every vertex from C will have degree 1 and every vertex from L will have degree either $\lfloor |C|/|L| \rfloor$ or $\lfloor |C|/|L| \rfloor + 1$. 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 $u \in C$ and $v \in L$, such that $degree(u) < k$ and $degree(v)$ is minimum among all the current degrees in L. 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.

Experimental investigation

For generating k-SAT instances we used five methods: classical random k-SAT (Random), (Bayardo & Schrag 1996)(Lit-bal-1), (Boufkhad *et al.* 2005)(Lit-bal-2), regular k-XORSAT and ours (High girth). Lit-bal-1 and Lit-bal-2 were almost identical. Best SAT solver among minisat, kcnfs, and satz used at every data point. Table 2 shows the

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: 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**
  $L_T := \{u \in V_1 \mid degree(u) < k\}$
  $R_T := \{u \in V_2 \mid degree(u) \le degree(v), \forall v \in V_2\}$
  $maxdist := 1$
  while $(maxdist = 1)$ **do**
    $T := \{(u,v) \mid (u,v) \in L_T \times R_T \; dist(u,v) \ge dist(x,y) \forall (x,y) \in L_T \times R_T\}$
    $d_{min} := degree(u)$, where $u \in R_T$
    $maxdist := dist(u,v)$, where $(u,v) \in T$
    if $(maxdist = 1)$ **then**
      $R_T := \{u \in V_2 \mid degree(u) = d_{min} + 1\}$
  $E := E \cup (u,v)$, where $(u,v) \in T$

Table 2: Ratio of median time on peak of hardness

	3-SAT		4-SAT		5-SAT	
Num. vars.	300	330	130	150	70	100
HG/Lit	1.29	1.44	1.34	1.39	1.64	3.09
Lit/Ran	80.2	132	4.58	5.59	1.73	2.09

higher arity (k) is, the higher the ratio HG/Lit results, particularly for larger number of variables. The ratio Lit/Ran seems to decrease. As we increase k, more variables may be needed in order to observe a difference. The XORSAT instances are harder than satisfiable ones from High girth, but generating the system of linear equations for k-XORSAT with our High girth method (HG-XORSAT) we obtain the hardest instances (see Table 1).

For n-ary CSPs, we have obtained similar results when generating their literal incidence graph with our High girth algorithm. For example, for instances with $|V| = 25, |D| = 3$ and arity 4, our High girth based CSP instances are about 1 order of magnitude harder to solve than random CSP instances.

References

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

Bayardo, R., and Schrag, R. 1996. Using CSP look-back techniques to solve exceptionally hard sat instances. In *Principles and Practice of Constraint Programming - CP 1996*, 46–60.

Boufkhad, Y.; Dubois, O.; Interian, Y.; and Selman, B. 2005. Regular random *k*-sat: Properties of balanced formulas. *Journal of Automated Reasoning* 35(1-3):181–200.

Järvisalo, M. 2006. Further investigations into regular xorsat. In *Proceedings of the AAAI 2006,*. AAAI Press / The MIT Press.