Locality in Random SAT Instances*

Jesús Giráldez-Cru
KTH Royal Institute of Technology
Stockholm, Sweden
giraldez@kth.se

Jordi Levy
IIIA-CSIC
Bellaterra, Spain
levy@iiia.csic.es

Abstract

Despite the success of CDCL SAT solvers solving industrial problems, there are still many open questions to explain such success. In this context, the generation of random SAT instances having computational properties more similar to real-world problems becomes crucial. Such generators are possibly the best tool to analyze families of instances and solvers behaviors on them.

In this paper, we present a random SAT instances generator based on the notion of *locality*. We show that this is a decisive dimension of attractiveness among the variables of a formula, and how CDCL SAT solvers take advantage of it. To the best of our knowledge, this is the first random SAT model that generates both scale-free structure and community structure at once.

1 Introduction

It is well known that most industrial SAT instances used in SAT competitions have a great variability in the number of occurrences of variables. In fact, Ansótegui et al. [2009a] showed that, in most real-world industrial SAT instances, the number of occurrences k of a randomly selected variable follows a power-law distribution $P(k) \sim k^{-\delta}$ where δ is between 2 and 3. In many cases, the clauses size also shows this kind of distribution. This means that a solver that would assign preferably those very frequent or *popular* variables would decrease the size of the formula very quickly. Variants of this strategy have been used in the past [Marques-Silva, 1999], in the Bohm's heuristic, Maximum Occurrences on clauses of Minimum Size (MOMS) [Freeman, 1995] and Jeroslow-Wang heuristic [Jeroslow and Wang, 1990], for instance. However, nowadays most modern SAT solvers use the Variable State Independent Decaying Sum (VSIDS) heuristic [Moskewicz *et al.*, 2001]. The intuition is that this heuristic *focuses* the solver on some *local area* of the formula. In large formulas, sometimes, it is preferable to assign variables *closer* to other recently assigned variables, than to assign popular but *distant* variables [Katsirelos and Simon, 2012]. In this direction, Giráldez-Cru and Levy [2015; 2016] define a model of random SAT instances with high modularity [Ansótegui *et al.*, 2012; Ansótegui *et al.*, 2015; 2016], and showed that VSIDS tends to focus the search inside *communities* of variables closely connected by clauses. Ansótegui et al. [2009b] define another model of random formulas with *scale-free structure*, i.e. where the number of occurrences of variables follow a power-law distribution, as observed in real-world SAT instances. However, there is no model of random SAT formulas that captures both properties at once: a power-law distribution in the number of variable occurrences and a notion of *locality*.

We can represent SAT instances as graphs; either representing variables as nodes, and the coexistence of two variables in a clause as an edge; or variables and clauses both as nodes, and the occurrence of a variable in a clause as an edge between them. This approach has allowed to apply many ideas from complex networks to SAT, e.g. the study of the fractal dimension of these formulas [Ansótegui *et al.*, 2014].

Most real-world networks (or graphs), such as Internet, the Web, or many social and metabolic networks, have the same *scale-free* structure observed in industrial SAT instances. This means that the degree (number of connections) of a random node follows a power-law distribution, where a few popular nodes attract a significant fraction of the edges. *Preferential attachment* [Barabási and Albert, 1999] is a model where the probability of a node to get new connections is proportional to its *popularity* (defined as the number of connections it already has). It has been proposed as the force that governs the growth of scale-free networks. However, it has also been observed that most real-world networks have also high clustering factor[1] and modularity, whereas graphs generated with the preferential attachment model have a low clustering factor and a modularity that tends to zero as the number of nodes tends to infinite.

Papadopoulos et al. [2012] showed that, although *popular-*

*This work was partially supported by the MINECO/FEDER project RASO (TIN2015-71799-C2-1-P), the CSIC project LOGAL (201450E045), and the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611.

[1]The clustering factor is the fraction of neighbors of a node that are connected, i.e. the conditional probability $P(a \leftrightarrow b \mid \exists c.a \leftrightarrow c \wedge b \leftrightarrow c)$.

ity and preferential attachment is one of the forces ruling real-world networks, another force is *similarity*. In other words, new nodes tend to connect to *popular* nodes (that already have a lot of connections), and also to *similar* nodes (that are close with respect to some metric). They define a model where at every time instants $t \in \{1, \ldots, n\}$ a new node with index t is created. The probability of an older node $s < t$ to get a connection from the new node t depends on some *energy* or *distance* $e_{st} = r_s + r_t + \log(\theta_{st}/2)$, where $r_s = \log(s)$ is the popularity[2] of node s, and θ_{st} is the similarity between s and t. To model similarity, they assign a random angle θ_t to each node t, and compute θ_{st} as the minimum distance between angles θ_t and θ_s, i.e., $\theta_{st} = \pi - |\pi - |\theta_t - \theta_s||$. Preferential attachment generates scale-free graphs with nodes degree following a power-law distribution $P(k) \sim k^{-\delta}$, where $\delta = 2$. In order to obtain greater values of δ, they assume a *popularity fading*, where popularity of node s depends on the time t as $r_s(t) = \beta \log(s) + (1 - \beta) \log(t)$, and $\beta \in [0, 1]$ is smaller as fading is faster. Finally, as edges in a graph are exclusive (we cannot have more than one edge between two nodes), they assume that they behave as *fermions*[3]. Therefore, they use the Fermi-Dirac probability distribution for the expected number of edges n in a given energy level e_{st}:

$$E[n_{st}] = \frac{1}{1 + e^{\frac{e_{st} - \mu}{k\,T}}}$$

where μ is the total chemical potential, k is the Boltzmann's constant, and T is the temperature[4].

In this paper, we propose a new model of random SAT instances that captures the notion of *locality* whereas preserves the power-law distribution in the number of variable occurrences. Additionally, this model also allows the lengths of the clauses following another power-law distribution. We will adapt many ideas of Papadopoulos et al. [2012]. One of the problems is that the *growing* process imposes some limitations on the graph. For instance, a minimum degree on nodes. Therefore, we will also adapt some ideas of Ansótegui et al. [2009b], where instead of a growing process as in preferential attachment, all nodes are created at once, and a distinct probability is used for every possible edge, ensuring that the expected node degree is $E[k_i] \sim i^{-\beta}$. This produces a power-law distribution of degrees on random nodes $P(k) \sim k^{-\delta}$, where $\delta = 1 + 1/\beta$. Moreover, we have to generate bi-partite graphs with variable and clause-nodes, where the degree of a variable-node is the number of occurrences of this variable, and the degree of a clause-nodes is the size of this clause. Although in most real-world industrial instances both values follow power-law distributions, their exponent δ is different in each case.

Finally, we show that CDCL SAT solvers take advantage of both popularity and similarity. In particular, we show that when formulas have these two properties, VSIDS quickly focuses its decisions on both popular variables, and variables similar to them. We also show that CDCL solvers perform better in formulas with popularity and similarity, than other solvers *specialized* in classical random formulas. On the contrary, the absence of these two properties makes these last solvers more efficient than CDCL ones. This suggests that both popularity and similarity are two crucial properties to understand the success of CDCL on industrial benchmarks.

[2] A smaller value of r_s means a greater popularity.

[3] Fermions are particles that obey the Pauli exclusion principle. Therefore, we can not have two fermions in exactly the same physical state.

[4] In fact, they integrate the Boltzmann's constant inside the definition of temperature.

2 The Popularity-Similarity SAT Model

In this section we define our Popularity-Similarity model (PS for short), adapting ideas from [Papadopoulos *et al.*, 2012] and [Ansótegui *et al.*, 2009b].

Definition 1 (Popularity-Similarity Random SAT Instance) *In order to generate a random SAT instance with locality over n variables with m clauses of average size k, we first assign a random angle $\theta_i \in [0, 2\pi]$, to every variable $i \in \{1, \ldots, n\}$, and a random angle $\theta'_j \in [0, 2\pi]$, to every clause $j \in \{1, \ldots, m\}$, with uniform probability distributions.*

Then, we construct a bi-partite random graph with n variable-nodes and m clause-nodes, where every possible edge $i \leftrightarrow j$ between a variable-node i and a clause-node j, is selected with probability:

$$P(i \leftrightarrow j) = \frac{1}{1 + \left(\dfrac{i^{\beta} \cdot j^{\beta'} \cdot \theta_{ij}}{R} \right)^{1/T}} \tag{1}$$

where β, β' and T are parameters of the model, θ_{ij} is the minimal distance between angles θ_i and θ'_j:

$$\theta_{ij} = \pi - \left| \pi - |\theta_i - \theta'_j| \right|$$

and R is the normalizing constant ensuring that, on average, the number of selected edges is $k\,m$:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} P(i \leftrightarrow j) = k \cdot m \tag{2}$$

Finally, we construct a random SAT formula from the graph as follows. For every edge $i \leftrightarrow j$ in the bi-partite graph, we add to the clause C_j the literal x_i with probability $1/2$, or the literal $\neg x_i$ otherwise.

This model corresponds to defining the *energy* of the edge $i \leftrightarrow j$ as $e_{ij} = \beta \cdot \log(i) + \beta' \cdot \log(j) + \log(\theta_{ij})$, the chemical potential as $\mu = \log R$, and using the Fermi-Dirac probability distribution. The first term in the energy e_{ij} represents the popularity of variable i, the second the popularity of clause j and the third the *similarity* between them. The probability of selecting an edge could be any function of this energy $P(i \leftrightarrow j) = f(e_{ij})$. However, following the same argument as [Papadopoulos *et al.*, 2012], we decide to use the Fermi-Dirac probability distribution, and a simplification of it in Subsection 2.3.

The effect of temperature T is regulating the entropy of the resulting formula. In the limit $T \to 0$, when we approach the absolute zero temperature, the number of fermions in state i is one when $e_i < \mu$, and zero when $e_i > \mu$. For other temperatures, the average number of fermions is between zero and

one. Therefore, at $T = 0$, the total chemical potential is equal to the energy of the n-th less energetic state, where n is the number of particles. In our case, for $T = 0$, the value R is the minimum value such that

$$|\{i \leftrightarrow j \mid e_{ij} < \log(R)\}| = k \cdot m$$

Therefore, for $T = 0$, the generated bi-partite graph consists of the $k \cdot m$ edges with smallest energy. At $T = \infty$, we get the graph with maximal entropy, which corresponds to the classical Erdös-Rényi random model, i.e., classical random SAT formulas. The chemical potential μ, or in our case R, depends on the temperature. In Subsection 2.2 we will see how to approximate R for $T > 0$.

The model generates SAT instances with $k\,m$ literals *on average*. If we want to generate instances of *exactly* $k\,m$ literals, we can select a random variable i and clause j with uniform probability, add the corresponding edge with probability given by Eq. 1, and repeat the process until we get the desired number of edges.

2.1 Generating Scale-free Instances

Ansótegui et al. [2009b] proved that, for scale-free random SAT instances, if the probability of a clause is given by $P(x_{i_1} \vee \cdots \vee x_{i_n}) \sim \prod_{j=1}^{n}(i_j)^{-\beta}$, then the number of occurrences of a variable chosen at random follows a power-law distribution $P(k) \sim k^{-\delta}$, where $\delta = 1/\beta + 1$. Here we prove a similar result:

Lemma 2 *For Popularity-Similarity random SAT instances, if the probability of an edge is $P(i \leftrightarrow j) = f(i^\beta\, j^{\beta'}\, \theta_{ij})$, and the function f decreases fast enough, then the resulting SAT instance is scale-free with a number of variable occurrences distributed as $P(k) \sim k^{-\delta}$, where $\delta = 1 + 1/\beta$. Similarly, clauses size also follows a power-law distribution with exponent $\delta' = 1 + 1/\beta'$.*

PROOF: The expected degree of node i can be computed integrating $P(i \leftrightarrow j)$ for all possible values of j. Since θ_i and θ'_j are uniformly distributed in $[0, 2\pi]$, the minimal distance between both angles will be uniformly distributed in $[0, \pi]$, and the integral be can computed as:

$$\begin{aligned}
E[k_i] &= \frac{1}{\pi}\int_1^m \int_0^\pi f(i^\beta j^{\beta'}\theta)\,\mathrm{d}\theta\,\mathrm{d}j \\
&= \frac{1}{\pi}\int_1^m i^{-\beta} j^{-\beta'} \int_0^{i^\beta j^{\beta'}\pi} f(x)\,\mathrm{d}x\,\mathrm{d}j \\
&\approx \frac{1}{\pi}\int_1^m i^{-\beta} j^{-\beta'} \int_0^{\infty} f(x)\,\mathrm{d}x\,\mathrm{d}j \\
&= \frac{1}{\pi}\left(\int_0^\infty f(x)\,\mathrm{d}x\right)\frac{m^{-\beta'+1}}{1-\beta'} i^{-\beta}
\end{aligned}$$

The approximation is correct when $\int_{i^\beta j^{\beta'}\pi}^{\infty} f(x)\,\mathrm{d}x$ is negligible.

Now, we basically have to reproduce the proof in [Ansótegui *et al.*, 2009b], to prove that $E[k_i] \sim i^{-\beta}$ implies $P(k) \sim k^{-\delta}$ with $\delta = 1 + 1/\beta$. Similarly, we prove that the expected size of clause j is $E[k_j] \sim j^{-\beta'}$. From this, we deduce that clause size also follows a power-law distribution.

In our case the approximation above is correct for small values of T. For instance, for $T = 0$, the function f defined by the Fermi-Dirac distribution returns one for the smallest $k\,m$ values of energy, and zero elsewhere. Therefore, $\int_{i^\beta j^{\beta'}\pi}^{\infty} f(x)\,\mathrm{d}x$ is exactly zero, because there are less edges than possible levels of energy. For small values of T, there is some error. ■

2.2 Normalizing the Probability Distribution

In order to normalize the probability distribution to ensure that the sum of probabilities is equal to the number of edges, as stated in Eq. (2), we have to integrate it. As the distance θ_{ij} between angles is uniformly distributed, we can compute the probability of an edge between nodes i and j as:

$$\begin{aligned}
P(i \leftrightarrow j) &= \frac{1}{\pi}\int_0^\pi \frac{1}{1+\left(\frac{i^\beta j^{\beta'}\theta}{R}\right)^{1/T}}\,\mathrm{d}\theta \\
&= \frac{R}{\pi} i^{-\beta} j^{-\beta'} \int_0^{i^\beta j^{\beta'}\pi} \frac{1}{1+x^{1/T}}\,\mathrm{d}x \\
&\approx \frac{R}{\pi} i^{-\beta} j^{-\beta'} \int_0^{\infty} \frac{1}{1+x^{1/T}}\,\mathrm{d}x = \frac{R}{\pi}\,\frac{T\pi}{\sin(T\pi)}\, i^{-\beta} j^{-\beta'}
\end{aligned}$$

where we are using the same approximation as in Lemma 2, and the equality $\int_0^\infty \frac{1}{1+x^{1/T}} = \frac{T\pi}{\sin(T\pi)}$ only holds for $0 < T < 1$. Imposing the condition of Eq. 2, we get:

$$R \approx \frac{\sin(T\pi)}{T}\,\frac{1-\beta}{n^{1-\beta}-1}\,\frac{1-\beta'}{m^{1-\beta'}-1}\,k\,m$$

which is a good approximation of R for small values of T.

Unfortunately, we have to compute the value of R for values $T > 1$, and we do not know how to do this analytically. Therefore, we use in our algorithm the Newton-Raphson method. Let $F = \sum_{i=1}^{n}\sum_{j=1}^{m} P(i \leftrightarrow j)$. We want to find the value of R such that $F(R) = k\,m$. Therefore, we compute:

$$\frac{\partial F}{\partial R} = \frac{\left(\frac{i^\beta j^{\beta'}\theta_{ij}}{R}\right)^{1/T}}{R\,T\left(1+\left(\frac{i^\beta j^{\beta'}\theta_{ij}}{R}\right)^{1/T}\right)^2}$$

and compute R as the limit of the serie:

$$R_{i+1} = R_i + \left.\frac{k\,m - F}{\frac{\partial F}{\partial R}}\right|_{R=R_i}$$

Experimentally, we observe that in 4 or 5 iterations the value of F is $k\,m \pm 0.1$.

2.3 Simplifying the Model

In the previous subsection we have seen that computing the value of R is a difficult task. Given an unnormalized probability distribution $P(x) \sim f(x)$ where $x = 1, \ldots, n$, a trivial way to normalize it is defining $P(x) = f(x)/\sum_{i=1}^{n} f(x)$, which ensures that $\sum_{i=1}^{n} P(x) = 1$. However, in the case of Equation (1), the function is already normalized (its value is always in the range $[0, 1]$), and Equation (2) is used to compute R, not as a proper probability normalization. However,

if we slightly modify Equation (1), we can transform the computation of R in a proper probability normalization.

The expected number of bosons[5] in an energy state e_i is given by the Bose-Einstein probability distribution:

$$E[n_i(e_i)] = \frac{1}{e^{\frac{e_i-\mu}{kT}} - 1}$$

Compared to the Fermi-Dirac distribution, we observe that a "plus one" has changed to "minus one". In our case, we propose to simply remove the "plus one". With this change, if we replace $R^{1/T}$ by simply R, and we impose an upper bound of 1 to the probability, in order to get a single edge for every pair of nodes, we get:

$$P(i \leftrightarrow j) = \min\left\{1\,,\ \frac{R}{(i^\beta j^{\beta'}\,\theta_{ij})^{1/T}}\right\} \tag{3}$$

Now, R is *almost* a normalizing constant, since it multiplies one of the cases in the $\min$. This probability distribution is quite similar to Eq. (1), and it results into SAT formulas quite similar to the ones obtained with the other model.

Since the probability is proportional to $i^{-\beta}$, adapting Lemma 2, we can see that the expected degree of node i is also $P(k_i) \sim i^{-\beta}$ and the generated formulas are also scale-free.

The computation of R is also quite simple. We have to compute a set S of pairs of nodes whose probability is 1, and iterate the computation of R as follows. Initially, we set:

$$S_0 = \emptyset$$
$$R_0 = k\,m \sum_{i\in\{1,\ldots,n\}, j\in\{1,\ldots,m\}} (i^\beta j^{\beta'}\,\theta_{ij})^{1/T}$$

and, at every iteration, compute

$$S_{i+1} = \{(i,j) \mid R_i/(i^\beta j^{\beta'}\,\theta_{ij})^{1/T} \geq 1\}$$
$$R_{i+1} = (k\,m - |S_{i+1}|) \sum_{\substack{i\in\{1,\ldots,n\}, j\in\{1,\ldots,m\} \\ (i,j)\notin S_{i+1}}} (i^\beta j^{\beta'}\,\theta_{ij})^{1/T}$$

until $S_i = S_{i+1}$ and the series converges[6]. In practice, it converges in just 2 or 3 steps. At this point, Eq. (2) holds.

3 Models with Clauses of Distinct Lengths

The Popularity-Similarity model we have described generates clauses of distinct sizes, following a power-law distribution with exponent $\delta' = 1 + 1/\beta'$, as it has been observed in real-world industrial instances. However, the satisfiability of formulas is very sensitive to the size of the clause. In particular, if the formula contains (many) small –unary or binary– clauses, it becomes trivially unsatisfiable. Moreover, one of the properties we appreciate in the classical random model is the existence of a SAT-UNSAT phase transition phenomenon. This allows to generate formulas with 50% probability of being satisfiable, and hence, more adequate for competitions. However, it is difficult to define a model with both distinct clause sizes and a phase transition. In order to exemplify this fact, we are going to study what we can consider a *classical model* of random formulas with distinct sizes. This is the simplest model with maximal entropy –hence, generating the most difficult instances– that we can define. Even though, we show that the existence of small clauses in this model makes formulas trivial.

Given a number of variables n, number of clauses m and average clause size k, we can generate a random formula choosing a random variable $i \in \{1,\ldots,n\}$ with uniform probability, a random clause $j \in \{1,\ldots,m\}$ with uniform probability, and a random sign $s \in \{1,-1\}$ with probability $1/2$, adding $s \cdot i$ in j, and repeating this process $k\,m$ times. Finally, we remove empty clauses, if there is any.

We have studied the phase transition and the hardness of formulas for this model, for distinct values of k, and clause/variable m/n fractions. We show the results in Fig. 1 for $n = 10^4$ variables. We observe that the transition between SAT and UNSAT is not so sharp as in regular random k-SAT formulas. The clause/variable fraction at the transition point increases with k, like the phase transition fraction for regular k-SAT. However, the value is not exactly the same. Moreover, if we repeat the experiment with $n = 10^5$ variables, we observe that almost all formulas in the represented ranges are trivially unsatisfiable. This means that the clause/variable fraction at the transition point is not constant. Finally, if we observe the number of conflicts needed to solve the formula, we see that, although the hardest formulas are located in the phase transition region, like in regular random formulas, they are trivially solvable for a modern SAT solver.

The only way to avoid all these problems is not only removing empty clauses, but also unary (and probably binary) clauses. Alternatively, we can also force clauses to contain a minimum of K literals (now k and K will be parameters of the model). In this second case, we would assign K literals to each clause like in the classical model, and then we would add $k\,m$ more literals shared among all clauses, following the above method. This way, the average clause size would be $k + K$. This is exactly what we do in our model. However, to assign the K literals to each clause, instead of the classical method, we use the following method:

For every clause, and for 1 to K, we select a random variable i with probability $P(i) = 1/(1 + (i^\beta \theta_{ij}/R)^{1/T})$ and a random sign with probability $1/2$. This is exactly the same probability distribution described in Eq. (1) removing the popularity of j. After that, we generate the $k\,m$ additional literals as described in Section 2, avoiding repetitions of variables in clauses.

4 Locality and CDCL SAT Solving

Let us recall first the parameters of the PS model for random SAT instances. The number of variables and clauses are n and m. K is the minimum clause size, and $(K + k)$ is the average clause size. The exponent of the power-law distributions of number of variable occurrences and clause size are respectively $\delta = 1 + 1/\beta$ and $\delta' = 1 + 1/\beta'$. When β and β' are very close to 0 (resp. to 1), there is almost no (resp.

[5]Bosons are particles that are not restricted by the Pauli's exclusion principle. Thus, we may have several bosons in the same state.

[6]Notice that $R_{i+1} \leq R_i$ and $S_{i+1} \supseteq S_i$, for any i. Therefore, convergence is ensured.

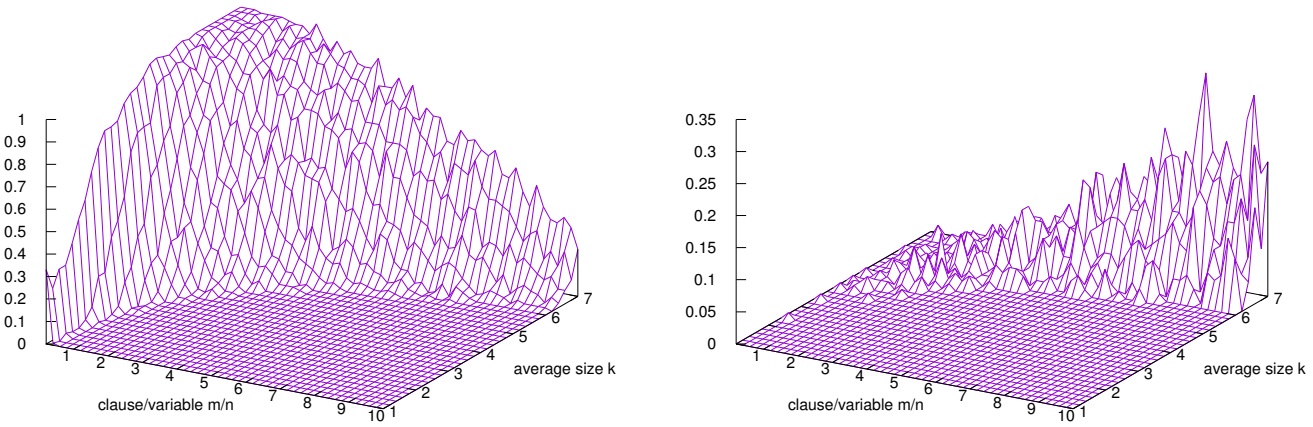

Figure 1: Fraction of unsatisfiable formulas for the variable clause length model (left) and number of conflicts required by Minisat to solve them (right), for $n = 10^4$ variables. Every point is the mean for 100 instances.

a high) variability. Finally, T is the temperature. When T is small, variables are chosen according to their popularity and similarity. In contrast, when the temperature is high enough, all variables have almost the same probability to be chosen at a certain moment, and hence, the model behaves very similar to the classical random model.

Since we are just interested in analyzing the effects of popularity and similarity of variables in SAT solving performance –and not the effects of different clause sizes–, we limit our experimentation to the case $\beta' = 0$, $K = 3$ and $k = 0$, i.e., all clauses of size 3. We always use $m/n = 4.25$.

In a first experiment, we show how VSIDS exploits both the popularity and the similarity of variables by analyzing the variables selected by the branching heuristic of the well-known CDCL SAT solver Minisat. VSIDS rewards the variables occurring in the last conflicts, and hence they are more likely to be selected in the next branching steps. We generate some extreme cases of our model to show these results.

In Fig. 2, we represent the variables chosen by VSIDS along the search, for some formulas of the PS model generated with $n = 5000$. Each point (i, j) represents that variable j was decided between the i-th and the $(i + 1)$-th conflicts. We represent three cases. First, a formula with high popularity ($\beta = 0.8$) and small temperature ($T = 1.50$); variables ordered by popularity (see top). Second, a formula with high similarity ($\beta = 0.1$) and small temperature ($T = 0.75$); variables ordered by similarity (see center). Finally, a third formula with high temperature ($T = 100$). In this last case, $\beta = 0.1$ and variables are ordered by similarity; similar results occur with other β or if ordered by popularity. Although we only represent the first 30,000 conflicts, the same results can be observed during the whole execution.

We can observe that in the formula with popularity, the solver decides in popular variables (low variable indexes). Ansótegui et al. [2009a] already showed that VSIDS likes branching on popular variables. Interestingly, in the formula with similarity, we observe that the solver is focused on a certain area of the angular space, i.e., similar variables. This suggests that the solver is indeed also taking advantage of the notion of locality in the formula. Finally, if the formula has neither popularity nor similarity –because the temperature is too high–, decisions occur everywhere (as expected) because the formula is more similar to a classical random CNF.

When a SAT formula has popularity and similarity, the CDCL solver is able to exploit these two structures, finding a good balance between both of them. A possible explanation is that the solver is finding conflicts faster, and these conflicts relate less variables. In the formula with popularity, the average clause size of the learned clauses is 18.2 literals, and conflicts occur at a rate of 584 conf/s; in the formula with similarity 20.3 literals and 510 conf/s. On the other hand, in the formula with high temperature, it is observed that after each restart (vertical lines), the solver makes many decisions, suggesting that it is harder to find conflicts in this formula (as it happens in random SAT instances). In fact, the average clause size for the learned clauses is 120.6 literals, and the learning rate is 154 conf/s.

In a second experiment, we show that CDCL SAT solvers indeed over-perform other solvers in PS random formulas. To this purpose, we evaluate the performances of the CDCL SAT solver Glucose [Audemard and Simon, 2009] and the look-ahead SAT solver March [Heule *et al.*, 2004] –more efficient in classical random instances–, on some families of random PS formulas with different temperatures T. The size of the formulas is $n = 5000$ when $T < 2$, and $n = 300$ otherwise. Notice that actual industrial formulas have millions of variables, but $n = 5000$ is enough to show the effects of popularity and similarity. On the contrary, when the temperature is high, the formulas are similar to random CNF. No complete solver could even solve a hard random instance with $n = 5000$ variables (in a reasonable timeout). So, we need to decrease the formula size to analyze these cases.

In Fig. 3, we plot the results. It can be seen that when the temperature is low (see $T = 1.40$), all formulas are very easy, and both solvers solve them in a few seconds. Interestingly, as we increase the temperature T, formulas become harder (as expected), but the CDCL solver is much more efficient solving most of the formulas (see $T = 1.50$). This happens until the temperature is very high, when both solvers fail to solve any formula (see $T = 1.55$). Finally, if the temperature keeps growing, the formulas become easier for March (see $T = 10$), as expected.

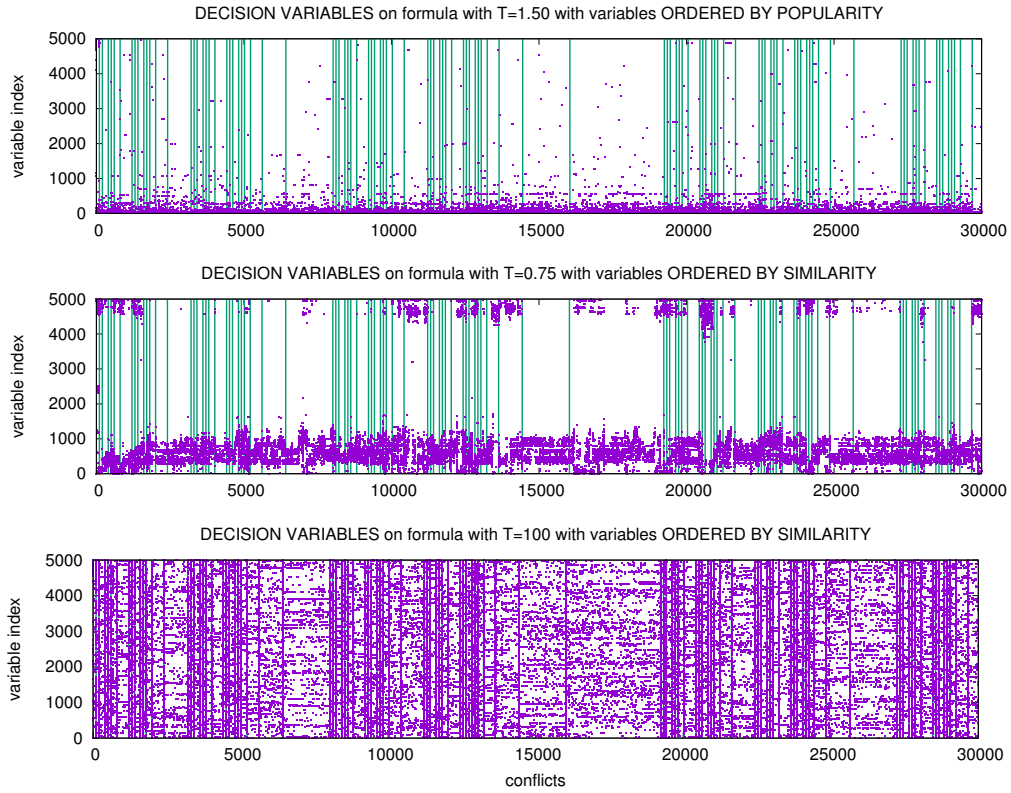

Figure 2: Analysis of the branching variables decisions performed by Minisat in some formulas of the PS model, with high popularity (top), high similarity (center), and none of them (bottom). Vertical lines represent restarts.

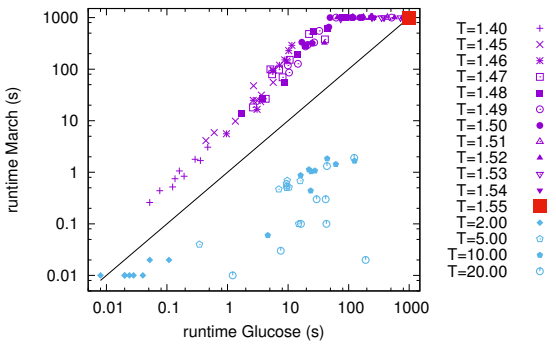

Figure 3: Scatter plot of runtimes of Glucose and March on different PS instances, for different temperatures T.

5 Related Work

The problem of generating random SAT formulas which realistically capture the computational properties of industrial instances has been stated as one of the most challenging problems in propositional search [Selman *et al.*, 1997; Kautz and Selman, 2003; 2007; Dechter, 2003].

Ansótegui et al. [2009b] propose the Scale-free model for random SAT formulas, where the number of variable occurrences follow a power-law distribution. However, this model is unable to generate formulas with high clustering or community structure. Recently, it has been shown that the phase transition point of these formulas depend on the exponent δ of the power-law distribution, in 2-CNF [Friedrich *et al.*, 2017]. Experimental results suggest that this is also the case in k-CNF, with $k > 2$.

Giráldez-Cru and Levy [2016] propose the Community Attachment model for random SAT formulas, which generates formulas with clear community structure. In this model, there is no variability in the occurrences of variables. Although these formulas are experimentally *easy*, it has been proven that in theory they are exponentially hard [Mull *et al.*, 2016].

6 Conclusions

In this paper, we present the Popularity-Similarity model for random SAT instances. It generates formulas with a power-law distribution in the number of variable occurrences (popularity), and good clustering between them (similarity). To the best of our knowledge, this is the first model that generates these two properties at once. Additionally, it also generates formulas with variable clause size. All of these features are typically found in a vast majority of industrial SAT instances.

We check that this model is characterized by a phase transition SAT-UNSAT phenomenon. We show that popularity and similarity are two orthogonal forces exploited by VSIDS. Also, we show that the performance of CDCL SAT solvers (w.r.t. other solving techniques) is related to the existence of popularity and similarity in the SAT formulas. Therefore, this suggests that both popularity and similarity are two crucial properties to understand the success of CDCL on industrial benchmarks.

References

[Ansótegui *et al.*, 2009a] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. On the structure of industrial SAT instances. In *Proc. of the 15th Int. Conf. on Principles and Practice of Constraint Programming (CP'09)*, pages 127–141, 2009.

[Ansótegui *et al.*, 2009b] Carlos Ansótegui, Maria Luisa Bonet, and Jordi Levy. Towards industrial-like random SAT instances. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09)*, pages 387–392, 2009.

[Ansótegui *et al.*, 2012] Carlos Ansótegui, Jesús Giráldez-Cru, and Jordi Levy. The community structure of SAT formulas. In *Proc. of the 15th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT'12)*, pages 410–423, 2012.

[Ansótegui *et al.*, 2014] Carlos Ansótegui, Maria Luisa Bonet, Jesús Giráldez-Cru, and Jordi Levy. The fractal dimension of SAT formulas. In *Proc. of the 7th Int. Joint Conf. on Automated Reasoning (IJCAR'14)*, pages 107–121, 2014.

[Ansótegui *et al.*, 2015] Carlos Ansótegui, Jesús Giráldez-Cru, Jordi Levy, and Laurent Simon. Using community structure to detect relevant learnt clauses. In *Proc. of the 18th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT'15)*, pages 238–254, 2015.

[Ansótegui *et al.*, 2016] Carlos Ansótegui, Maria Luisa Bonet, Jesús Giráldez-Cru, and Jordi Levy. Community structure in industrial SAT instances. *arXiv*, 1606.03329, 2016.

[Audemard and Simon, 2009] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern SAT solvers. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09)*, pages 399–404, 2009.

[Barabási and Albert, 1999] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[Dechter, 2003] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[Freeman, 1995] J. W. Freeman. *Improvements to Propositional Satisfiability Search Algorithms*. PhD thesis, University of Pennsylvania, Philadelphia, 1995.

[Friedrich *et al.*, 2017] Tobias Friedrich, Anton Krohmer, Ralf Rothenberger, and Andrew M. Sutton. Phase transition for sclae-free SAT formulas. In *Proc. of the 31st National Conf. on Artificial Intelligence (AAAI'17)*, 2017.

[Giráldez-Cru and Levy, 2015] Jesús Giráldez-Cru and Jordi Levy. A modularity-based random SAT instances generator. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 1952–1958, 2015.

[Giráldez-Cru and Levy, 2016] Jesús Giráldez-Cru and Jordi Levy. Generating SAT instances with community structure. *Artif. Intell.*, 238:119–134, 2016.

[Heule *et al.*, 2004] Marijn J. H. Heule, Joris E. van Zwieten, Mark Dufour, and Hans van Maaren. March_eq: Implementing additional reasoning into an efficient Look-ahead SAT solver. In *Proc. of the 7th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT'04)*, pages 345–359, 2004.

[Jeroslow and Wang, 1990] Robert G. Jeroslow and Jinchang Wang. Solving propositional satisfiability problems. *Annals of Mathematics and Artificial Intelligence*, 1(1):167–187, 1990.

[Katsirelos and Simon, 2012] George Katsirelos and Laurent Simon. Eigenvector centrality in industrial SAT instances. In *Proc. of the 18th Int. Conf. on Principles and Practice of Constraint Programming (CP'12)*, pages 348–356, 2012.

[Kautz and Selman, 2003] Henry A. Kautz and Bart Selman. Ten challenges redux: Recent progress in propositional reasoning and search. In *Proc. of the 9th Int. Conf. on Principles and Practice of Constraint Programming (CP'03)*, pages 1–18, 2003.

[Kautz and Selman, 2007] Henry A. Kautz and Bart Selman. The state of SAT. *Discrete Applied Mathematics*, 155(12):1514–1524, 2007.

[Marques-Silva, 1999] Joao P. Marques-Silva. The impact of branching heuristics in propositional satisfiability algorithms. In *Proc. of the 9th Portuguese Conf. on Artificial Intelligence: Progress in Artificial Intelligence (EPIA'99)*, volume 1695 of *LNCS*, pages 62–74, 1999.

[Moskewicz *et al.*, 2001] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient sat solver. In *Proc. of the 38th Annual Design Automation Conf. (DAC'01)*, pages 530–535, 2001.

[Mull *et al.*, 2016] Nathan Mull, Daniel J. Fremont, and Sanjit A. Seshia. On the hardness of SAT with community structure. In *Proc. of the 19th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT'16)*, pages 141–159, 2016.

[Papadopoulos *et al.*, 2012] Fragkiskos Papadopoulos, Maksim Kitsak, M. Ángeles Serrano, Marián Boguñá, and Dimitri Krioukov. Popularity versus similarity in growing networks. *Nature*, 489:537–540, 2012.

[Selman *et al.*, 1997] Bart Selman, Henry A. Kautz, and David A. McAllester. Ten challenges in propositional reasoning and search. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI'97)*, pages 50–54, 1997.