# Satisfiability and Algorithms for Non-uniform Random k-SAT

**Oleksii Omelchenko, Andrei A. Bulatov**
Simon Fraser University, Burnaby BC, Canada
oomelche@sfu.ca, abulatov@sfu.ca

## Abstract

Solving Satisfiability is at the core of a wide range of applications from Knowledge Representation to Logic Programming to Software and Hardware Verification. One of the models of Satisfiability, the Random Satisfiability problem, has received much attention in the literature both, as a useful benchmark for SAT solvers, and as an exciting mathematical object. In this paper we tackle a somewhat nonstandard type of Random Satisfiability, the one where instances are not chosen uniformly from a certain class of instances, but rather from a certain nontrivial distribution. More precisely, we use so-called Configuration Model, in which we start with a distribution of degrees (the number of occurrences) of a variable, sample the degree of each variable and then generate a random instance with the prescribed degrees. It has been proposed previously that by properly selecting the starting distribution (to be, say, power law or lognorm) one can approximate at least some aspect of 'industrial' instances of SAT. Here we suggest an algorithm that solves such problems for a wide range of degree distributions and obtain a necessary and a sufficient condition for the satisfiability of such formulas.

## Introduction

The Satisfiability (SAT) problem is one of the classical and most useful in practice computational problems. Solving Satisfiability is at the core of a wide range of applications from Knowledge Representation to Logic Programming to Software and Hardware Verification. Although NP-complete in general, SAT proved to be solvable in reasonable time in a great number of applications. SAT solvers have seen a great deal of progress over the last 3-4 decades. A number of standard methods such as DPLL (Davis and Putnam 1960; Davis, Logemann, and Loveland 1962), and also more advanced heuristics such as conflict analysis (Marques-Silva and Sakallah 1999), VSIDS heuristics (Marques-Silva and Sakallah 1999; Moskewicz et al. 2001), and others (Jarvisalo and Biere 2012; Eén and Biere 2005) have allowed for creation of solvers capable of solving industrial scale instances of SAT, which nowadays may include millions of variables and clauses. Such solvers have become reliable tools in solving a wide variety of real-world computational problems.

Random SAT and its restriction, Random $k$-SAT, are often seen as a model of 'typical case' instances of SAT, and have been intensively studied for decades. Apart from algorithmic questions related to the Random SAT, much attention has been paid to such problems as satisfiability thresholds and the structure of the solution space. While it makes sense to consider random instances as 'typical' SAT instances, in reality it turns out that random instances tend to be much more difficult than the instances seen in practice (Cook and Mitchell 1996). Even rather small random formulas containing only a few hundreds variables often cannot be solved by even the most successful SAT solvers in reasonable time, in stark contrast with huge practical instances of millions of variables. It is therefore an interesting research question, what properties make practical instances amenable to SAT solving methods.

A number of candidates have been suggested to explain this phenomenon such as certain regularities in the instances, low values of graph parameters associated with the instance (say, treewidth), backdoor variables, etc. For example, it is known that SAT solvers use restarts with great success, while restarts help very little when solving random SAT instances. In (Li et al. 2020) they attempt to find a broad class of SAT instances where restarts help. Another proposed condition, (Ansótegui, Bonet, and Levy 2009a,b), is the distribution of the degree (the number of occurrences) of variables in an instance. In this paper we follow this approach and study the Random SAT problem for a range of distributions, as explained in detail below. Although results obtained here and in previous research do not support the proposition that practical problems can be thought of as instances of some Random SAT over a nonstandard distribution, such problems seem to be of significant interest in their own rights as they allow for representation of very diverse types of SAT instances.

Probably the central theme in the SAT research apart from algorithms is identifying necessary and sufficient condition for satisfiability of formulas. Some of such conditions are structural, e.g., so-called bicycles and contradictory paths (Chvátal and Reed 1992; Cooper, Frieze, and Sorkin 2007) determining the satisfiability of 2-CNFs (see more about that in Section ). In the cases of Random SAT such satisfiability conditions manifest themselves as the phenomenon of satisfiability threshold and phase transition.

One of the most widely studied distributions of random SAT is the *uniform model*. To sample $k$-CNF formulas with

$n$ variables and $m$ clauses from this model, we sample u.a.r. (uniformly at random) $m$ clauses with replacement from the set of all $\binom{n}{k}2^k$ possible clauses having exactly $k$ literals. Hence, each instance is sampled equiprobably among all formulas with $n$ variables and $m$ clauses. The satisfiability of a random $k$-CNF then depends heavily on the quantity $\rho = \frac{m}{n}$, called *density*. It was proved in (Friedgut 1999) that for each $k$ (and possibly the number of variables $n$) there exists a critical density $\rho_k(n)$, such that random uniform formulas with density less than $\rho_k(n)$ are satisfiable with high probability (w.h.p.), while instances with $\rho$ above the critical density are unsatisfiable w.h.p. Moreover, a recent work (Friedrich and Rothenberger 2019), which may be regarded as an extension of Fridgut's result to non-uniform random SAT instances, shows that if a distribution of variable's occurrence in random formulas satisfies some criteria, then such formulas must undergo a sharp satisfiability threshold.

The satisfiability threshold phenomena produced an impressive line of research. It includes designing and analyzing SAT algorithms (Achlioptas 2001; Coja-Oghlan 2010; Coja-Oghlan and Panagiotou 2013; Ding, Sly, and Sun 2015) and applications of the second moment method (Achlioptas and Moore 2006) to find lower bounds, and a variety of probabilistic and proof complexity tools to obtain upper bounds (Dubios and Boufkhad 1997; Kirousis et al. 1998; Dubois, Boufkhad, and Mandler 2003). The culmination of this research so far has been the result by (Ding, Sly, and Sun 2015) that finds the precise location of the satisfiability threshold for large $k$.

To see how the subject of this paper is related to the aforementioned research, note that Random $k$-SAT can be formulated using one of the three models whose statistical properties are very similar. The model with fixed density $\rho$ is described above (Franco and Paull 1983; Selman, Mitchell, and Levesque 1996). Alternatively, to produce a random formula, for selected variables every possible $k$-clause is included with probability tuned up so that the expected number of clauses equals $\rho n$. Finally, (Kim 2004) showed that one can also use the Configuration Model we study in this paper, which he called Poisson Cloning model. In this model for each variable $v_i$ we first select a positive integer $d_i$ accordingly to the Poisson distribution with expectation $\rho k$, the degree of the variable. Then we create $d_i$ *clones* of variable $v_i$, and choose a random partition of the set of clones into $(d_1 + \cdots + d_n)/k$ $k$-element subsets, then converting them into clauses randomly. The three models are largely equivalent and can be used whichever suits better to the task at hand.

The Configuration Model opens up a possibility for a wide range of different distributions of $k$-CNFs arising from different degree distributions. Starting with any random variable $\xi$ that takes strictly positive integer values one obtains a distribution $\mathbb{C}_k^n(\xi)$ on $k$-CNFs as above using $\xi$ in place of the Poisson distribution. Note that $\xi$ may depend on $n$, the number of variables, and even be different for different variables. One 'extreme' case of such a distribution is Poisson Cloning described above. Another case is studied by Cooper, Frieze, and Sorkin (Cooper, Frieze, and Sorkin 2007). In their case each variable of a 2-SAT instance has a prescribed degree, which can be viewed as assigning a degree to every variable according to a random variable that only takes one value. Also, (Boufkhad et al. 2005) considered another case of this kind — regular Random $k$-SAT.

Therefore, it may be beneficial to study what similarities and differences the random models have, what inner structures they possess, and how they affect the running time of SAT solvers, in order to create better and more universal search heuristics.

In this paper we consider Random $k$-SAT in the configuration model given by distribution $\mathbb{C}_k^n(\xi)$, where $\xi$ is distributed according to the power law distribution in the following sense. Let $F_\xi(\ell) = \Pr[\xi \geq \ell]$ denote the tail function of a positive integer valued random variable $\xi$. We say that $\xi$ is distributed according to the power law with parameter $\alpha$ if there exist constants $V, W$ such that

$$W\ell^{-\alpha} \leq F_\xi(\ell) \leq V\ell^{-\alpha}. \tag{1}$$

Power law type distributions have received much attention in the literature. They have been widely observed in natural phenomena (Newman 2005; Clauset, Shalizi, and Newman 2009), as well as in more artificial structures such as networks of various kinds (Barabási and Albert 1999). Apart from the Configuration Model, graphs and hypergraphs (and therefore CNFs), whose degree sequences are distributed accordingly to a power law of some kind can also be generated in a number of ways. These include preferential attachment (Aiello, Graham, and Lu 2001; Barabási and Albert 1999; Bollobás et al. 2001; Bollobás and Riordan 2002), hyperbolic geometry (Krioukov et al. 2010), and others (Ansótegui, Bonet, and Levy 2009a,b). Although graphs, hypergraphs, and CNFs resulting from all such processes satisfy the power law distributions of their degrees, other properties can be very different.

The approach most closely related to this paper was suggested by (Ansótegui, Bonet, and Levy 2009a,b). Given the number of variables $n$, the number of clauses $m$, and a parameter $\beta$, the first step in their construction is to create $m$ $k$-clauses without naming the variables. Then for every variable-place $X$ in every clause, $X$ is assigned to be one of the variables $v_1, \ldots, v_n$ according to the distribution

$$\Pr[X = v_i, \beta, n] = \frac{i^{-\beta}}{\sum_{j=1}^{n} j^{-\beta}}.$$

Ansotegui et al. argue that this model often well matches the experimental results on industrial instances, see also (Ansótegui et al. 2016; Giráldez-Cru and Levy 2016; Ansótegui et al. 2008).

The satisfiability conditions for this model has been studied in (Friedrich et al. 2017). Since the model has two parameters, $\beta$ and $\rho = m/n$, the resulting picture is complicated. Friedrich et al. proved that a random CNF is unsatisfiable with high probability if $\rho$ is large enough (although constant), and if $\beta < \frac{2k-1}{k-1}$. If $\beta \geq \frac{2k-1}{k-1}$, the formula is satisfiable with high probability provided $\rho$ is smaller than a certain constant. The unsatisfiability results in (Friedrich et al. 2017) are mostly proved using the local structure of a formula.

Here we seek to obtain similar results for the Configuration Model, in which distribution $\xi$ satisfies condition (1). The case $k = 2$ was considered in (Omelchenko and Bulatov 2019), where a tight bound on the satisfiability/unsatisfiability was obtained. Specifically, they proved that a formula $\phi$ sampled from $\mathbb{C}_2^n(\xi)$ is satisfiable w.h.p. if $\xi$ satisfies the inequality $\mathbb{E}\xi^2 < 3\mathbb{E}\xi$ and unsatisfiable w.h.p. otherwise.

Here we consider the case when $k \geq 3$. Although we have not obtained a complete satisfiability classification, we obtained strong necessary and sufficient conditions for the satisfiability of instances in the Configuration Model. First, we show that if $\alpha$ is small enough (and therefore the formulas tend to be dense) w.h.p. the formula contains a small unsatisfiable subformula, and hence is unsatisfiable itself.

**Theorem 1** *Let $\phi$ is sampled according to the Configuration Model for a distribution $\xi$ satisfying condition* (1) *for $0 < \alpha < \frac{k}{k-1}$. Then w.h.p. $\phi$ is unsatisfiable.*

Interestingly, the density of formulas satisfying (1) can be much lower than the density of unsatisfiable instances sampled uniformly, which of course suggests that heavy tail distributions allow for unevenly concentrated formulas.

For a sufficient condition for satisfiability we suggest an algorithm (Algorithm 2) that w.h.p. finds a satisfying assignment provided $\xi$ satisfy certain property (see condition (2) in the section on a sufficient condition) that is not difficult to verify.

**Theorem 2** *Let $\phi$ be sampled according to the Configuration Model for a r.v. $\xi$ satisfying condition* (1) *for $\alpha \geq \frac{k}{k-1}$, and*

$$2C_1\,\mathbb{E}\left[\left(\deg(l)\deg(\bar{l})\right)^{1+\frac{(1-\alpha)(k-2)}{2}}\right] < 1,$$

*where $C_1 := \left[\frac{2V}{\alpha-1}\right]^{k-2}\frac{k}{[\mathbb{E}\xi]^{k-1}}$. Then w.h.p. $\phi$ is satisfiable.*

We will argue later that the gap between the necessary condition from Theorem 1 and the sufficient condition from Theorem 2 is relatively small. However, experiments suggest that the actual satisfiability/unsatisfiability borderline lies somewhere between the two conditions, see the section on experiments.

Note also that while Algorithm 2 serves theoretical purposes, actual SAT solvers have no difficulties with power-law distributed CNFs even of very large size, as our experiments suggest.

Due to space restrictions, we only explain here the main ideas behind the majority of proofs. Detailed proofs can be found in supplemental materials.

## Notation and Preliminaries

**Random $k$-Satisfiability and Configuration Model.** As usual, a $k$-CNF is a conjunction $\phi = \bigwedge_{1\leq i\leq m} c_i$ of *clauses*, and every clause $c_i$ is a disjunction of exactly $k$ *literals*, that is, a propositional variable or its negation. It will be convenient to think of a $k$-CNF as a set of its clauses, i.e. $\phi = \{c_1, c_2, \ldots, c_m\}$. We call the *degree of a variable* $v$ the number of times it appears in $\phi$. Likewise, the degree of a literal $l$ is the number of times it appears in $\phi$. To avoid any confusion, we use letter $v$ (possibly with a subscript) when we refer to a variable, and letter $l$ (again, possibly with a subscript) to refer to a literal. Then the degree of a variable $v$ will be denoted $\deg(v)$ and the degree of a literal $l$ will be denoted $\deg(l)$.

In the *$k$-Satisfiability* problem ($k$-SAT for short) the goal is to decide the satisfiability of a $k$-CNF. In the *Random $k$-Satisfiability* the goal is the same, while instances are sampled from a certain distribution of $k$-CNFs.

In this paper we study Random $k$-SAT that does not have limitations on instances such as fixed density, but different instances appear with vastly different probability. The *Configuration Model* for $k$-SAT is defined as follows. Let $\xi$ be positive integer-valued random variable, this will be the distribution of the degree of each variable in our $k$-CNFs. Let $n$ be the number of variables we want in the formula, let the variables be $v_1, \ldots, v_n$. Then construct a $k$-CNF in the following three steps.

STEP 1. Sample the degree $d_i$ of each $v_i$ from $\xi$. If the sum of the $d_i$'s is not a multiple of $k$, discard the sampled numbers and start over. Otherwise, for each $i$, create $d_i$ copies of $v_i$; we call these copies *clones* of $v_i$.

STEP 2. Generate a partition $p_1, \ldots, p_m$ of the set $\mathcal{S}$ of all clones into $k$-element subsets uniformly at random. These will be the clauses of our $k$-CNF after we choose the polarity for each clone in the next step. It may happen that some of the $k$-element subsets in fact contain multiple clones of the same variable. We treat such clauses as regular $k$-clauses, and add them into the formula.

STEP 3. For each $p_i$ and every clone $x \in p_i$, negate $x$ with probability 1/2. The resulting clause is denoted $c_i$.

If clone $x$ is a clone of a variable $v$ and it was assigned negative sign, then we say that $x$ is also a clone of the literal $\bar{v}$, otherwise it is clone of the literal $v$. To avoid any confusion that may arise from this convention, we will explicitly mention whether we talk about literals or variables. If a literal $l$ was formed from a clone of some variable $v$, we say $l$ is *associated* with $v$. Algorithm 1 gives a more formal description of this procedure.

We denote the fact that a $k$-CNF formula $\phi$ having $n$ variables was sampled from the Configuration Model by $\phi \sim \mathbb{C}_n^k(\xi)$.

**Example 3 (Poisson Cloning (Kim 2004))** *In the standard uniform Random $k$-SAT we generate a random instance with $n$ variables and $m$ clauses by sampling the required number of random clauses from the set of all $k$-clauses. If the density $\rho = m/n$ is low (say, constant) the distribution of the degree of each variable is well approximated by the Poisson distribution. Therefore, (Kim 2004) argued that the Configuration Model in which $\xi$ has the Poisson distribution with the mean $k\rho$ shares many common properties with the standard Random $k$-SAT. Kim called this kind of Configuration Model* Poisson Cloning *and demonstrated that a number of*

**Algorithm 1** Configuration Model $\mathbb{C}_n^k(\xi)$

**procedure** CONFIGURATIONMODELCNF($n$, $k$, $\xi$)
  Form a sequence of $n$ numbers $\{d_i\}_{i=1}^n$, each sampled independently from $\xi$
  **if** $S_n := \sum_{i=1}^n d_i$ is not a multiple of $k$ **then**
    discard the sequence, and go to step 2
  **end if**
  Otherwise, let $\mathcal{S} \leftarrow \bigcup_{i=1}^{n} \{\underbrace{v_i, v_i, \ldots, v_i}_{d_i \text{ times}}\}$
  Let $\phi \leftarrow \emptyset$
  **while** $\mathcal{S} \neq \emptyset$ **do**
    Pick u.a.r. $k$ elements $\{x_1, x_2, \ldots, x_k\}$ from $\mathcal{S}$
    Let $c \leftarrow \{x_1, x_2, \ldots, x_k\}$, $\mathcal{S} \leftarrow \mathcal{S} - c$
    Negate each element in $c$ independently with probability 1/2
    $\phi \leftarrow \phi \cup \{c\}$
  **end while**
  **return** $\phi$
**end procedure**

*results can be obtained much easier in the Poisson Cloning model.*

**Example 4 ($d$-Regular SAT)** *If $\xi$ is constant, that is, $\Pr[\xi = d] = 1$, we obtain so-called $d$-*Regular $k$-SAT*, in which the degree of every variable equals d. (Boufkhad et al. 2005) uses another model in which all literals have almost the same degree. However, the later model can be simulated to some extent by the Configuration Model as well.*

**Example 5 (Power Law SAT)** *The following distribution plays a very important role in this paper, see, (Borovkov 2008; Omelchenko and Bulatov 2018, 2019). Random variable $\xi$ is said to be distributed according to the* zeta distribution *with parameter $\beta$ if $\Pr[\xi = \ell] = \frac{\ell^{-\beta}}{\zeta(\beta)}$, where $\zeta(\beta)$ is the Riemann zeta function. Power-law distributed structures naturally appear as the result of many discrete processes, and the Configuration Model with a zeta distributed $\xi$ often approximates such structures very well.*

**Tail Conditions and Power-Law Distributions.** The conditions on random variables (r.v.) we use are in terms of tail functions. The tail function of a positive integer-valued r.v. $\xi$ is $F_\xi(\ell) = \Pr[\xi \geq \ell]$, where $\ell \geq 1$.

**Definition 6** *An integer-valued positive r.v. $\xi$ has power-law probability distribution, if $F_\xi(\ell) = \Theta\left(\ell^{-\alpha}\right)$, where $\alpha > 0$. We denote this fact as $\xi \sim P(\alpha)$. Clearly, if $\xi \sim P(\alpha)$, then there exist constants $V, W > 0$, such that $W\,\ell^{-\alpha} \leq F_\xi(\ell) \leq V\,\ell^{-\alpha}$ for every $\ell \geq 1$.*

The prime example of a power-law distributed r.v. is the zeta distributed r.v. introduced in Example 5. To verify this we only need to compute its tail function. Let $\xi$ be a zeta distributed r.v. with parameter $\beta > 1$. A rough estimation shows that

$$F_\xi(\ell) = \Pr[\xi \geq \ell] = \sum_{d=\ell}^{\infty} \frac{d^{-\beta}}{\zeta(\beta)} = \Theta\left(\ell^{1-\beta}\right).$$

Thus, a zeta-distributed r.v. with parameter $\beta$ in our terms is power-law distributed with parameter $\alpha = \beta - 1$.

Suppose that $\phi \in \mathbb{C}_k^n(\xi)$ is a $k$-CNF with $n$ variables produced by the Configuration Model Algorithm 1. The set of all clones will usually be denoted by $\mathcal{S}_n$ and $S_n = |\mathcal{S}_n|$ will denote the total number of clones in $\phi$. Clearly, the number of clauses in $\phi$ is $|\phi| = \frac{S_n}{k}$. We denote by $L(\phi)$ the set of all literals in $\phi$, while $V(\phi)$ denotes the set of all variables in $\phi$. Then $|V(\phi)| = n$ and $|L(\phi)| \leq 2n$.

The next theorem shows that $S_n$ does not deviate too far from its expected value.

**Theorem 7 ((Omelchenko and Bulatov 2018))** *Let $S_n = \sum_{i=1}^n \xi_i$, where $\xi_i$'s are independent copies of a positive integer-valued random variable $\xi$ with $\Pr[\xi \geq \ell] \leq V\,\ell^{-\alpha}$, where $V > 0$ and $\alpha > 0$ are constants.*

*(1) If $0 < \alpha \leq 1$, then $S_n = O\left(n^{\frac{1}{\alpha}}\right)$ w.h.p.*

*(2) If $\alpha > 1$, then $S_n = n\mathbb{E}\xi + o(n)$ w.h.p.*

We complete this section with a useful bound on the maximal degrees of variables in $\phi$.

**Lemma 8** *Let $\phi \sim \mathbb{C}_n^k(\xi)$, where $\xi$ is some positive integer-valued r.v. with the right tail satisfying $F_\xi(\ell) \leq V\,\ell^{-\alpha}$ for some $V, \alpha > 0$. Let also $\Delta = \max_{v \in V(\phi)} \left(\deg(v)\right)$ be the maximal degree of variables in $\phi$. Then $\Delta = O\left(n^{\frac{1}{\alpha}}\right)$ w.h.p.*

## A Necessary Condition for Satisfiability

We begin with identifying a necessary condition for satisfiability of instances of our Random $k$-SAT. To this end we establish a simple combinatorial property of $k$-CNFs sampled from $\mathbb{C}_n^k(\xi)$, $\xi \sim P(\alpha)$, when $\alpha$ is small enough.

**Theorem 9** *Let $\phi \sim \mathbb{C}_n^k(\xi)$, where $\xi \sim P(\alpha)$ and $k \geq 2$. If $0 < \alpha < \frac{k}{k-1}$, then w.h.p. $\phi$ is unsatisfiable.*

To prove the theorem we show that $\phi$ contains a set of $k$ variables that w.h.p. appear together in at least $\frac{1}{2}(k-1)!\log^k n$ clauses. Then, observing that if we have a formula with $k$ variables, we only need $2^k$ clauses (that is a constant number) to make the formula unsatisfiable. In the case $0 < \alpha < \frac{k-1}{k}$ in $\phi \sim P(\alpha)$ we have a subformula with $(k-1)!\log^k n$ clauses. Clearly, w.h.p. such subformula is not satisfiable.

In the case of power law distributed r.vs. in addition to unsatisfiability several other events happen at $\alpha = \frac{k}{k-1}$.

**Theorem 10** *Let $\xi \sim P(\alpha)$. Then the following are equivalent:*

1. *$\alpha < \frac{k}{k-1}$;*
2. *for $\phi \in \mathbb{C}_n^k(\xi)$ let $\phi^\ell$ denote the formula obtained from $\phi$ by removing all variables of degree less than $\ell$ and clauses they occur in; then w.h.p. the density of $\phi^\ell$ goes to infinity as $n$ and $\ell$ grow;*
3. *for any $k$-SAT formula $\psi$ with a fixed number of variables with constant degrees w.h.p. $\phi \in \mathbb{C}_n^k(\xi)$ contains $\psi$ as subformula.*

**Algorithm 2**

**procedure** IsSAT($\phi$)
  $\phi' \leftarrow \emptyset$
  **for** $c \in \phi$ **do**
    Let $l_1, l_2$ be two literals from $c$, such that $\deg(l) \geq \max(\deg(l_1), \deg(l_2))$ for any $l \in c$
    $c' \leftarrow \{l_1, l_2\}$
    $\phi' \leftarrow \phi' \cup \{c'\}$
  **end for**
  **if** $\phi'$ is SAT **then**
    **return** SAT
  **end if**
  **return** FAILED
**end procedure**

## A Sufficient Condition for Satisfiability

In this section we present a sufficient condition for a Random $k$-SAT to be satisfiable. More specifically, we show that if the distribution $\xi$ satisfies some extra conditions on the mean of a certain expression in terms of degrees of literals, then a random instance from $\mathbb{C}_n^k(\xi)$ is satisfiable with high probability.

To obtain this result we present an algorithm that given an instance from $\mathbb{C}_n^k(\xi)$, where $\xi$ satisfies the aforementioned condition, with high probability finds a satisfying assignment for such an instance. Clearly, the existence of such algorithm implies that instances from $\mathbb{C}_n^k(\xi)$ must be satisfiable with high probability as well.

Given a $k$-CNF $\phi$ the algorithm works as follows: first, for each clause it identifies two literals with the smallest degrees (making a random choice if such literals are not unique) and removes all the remaining literals, thus converting $\phi$ into a 2-CNF formula $\phi'$. Second, it solves the resulting 2-SAT instance that can be done in linear time. If $\phi'$ is satisfiable, then clearly the original formula $\phi$ is satisfiable as well. However, if $\phi'$ is not satisfiable, we cannot infer that $\phi$ is not satisfiable, and the algorithm declares it fails to answer whether $\phi$ is satisfiable or not. For a more formal description of the algorithm, see Algorithm 2. We show that when $\xi$ satisfies some condition, Algorithm 2 finds a satisfying assignment of $\phi \in \mathbb{C}_n^k(\xi)$ w.h.p.

Originally this algorithm was proposed for a different model, which was used to study satisfiability of random $k$-CNFs with power-law degree distribution (Friedrich et al. 2017). Although the model studied in (Friedrich et al. 2017) and the Configuration Model exhibit some level of similarity, there are significant differences too. For instance, the latter model is able to simulate directly random d-Regular CNF, while the former model cannot do this; on the other hand, model from (Friedrich et al. 2017) can be used to produce instances of random uniform model, while the Configuration Model can only approximate such instances. What is also more important for this section, in (Friedrich et al. 2017) Algorithm 2 proves a necessary and sufficient conditions for satisfiability, while in our case it can only prove a sufficient condition that does not quite match the necessary condition from the previous section.

**Theorem 11** *Let $\phi \sim \mathbb{C}_n^k(\xi)$, where $\xi$ is some positive integer-valued r.v. with the right tail function $F_\xi(\ell) \leq V\,\ell^{-\alpha}$ for some constants $V > 0$, $\alpha > \frac{k}{k-1}$ and any $\ell \geq 1$. Then Algorithm 2 finds a satisfying assignment of $\phi$ w.h.p., when $\xi$ satisfies*

$$2C_1\,\mathbb{E}\left[\Big(\deg(l)\deg(\bar{l})\Big)^{1+\frac{(1-\alpha)(k-2)}{2}}\right] < 1, \qquad (2)$$

*where $C_1 := \left[\frac{2V}{\alpha-1}\right]^{k-2} \frac{k}{[\mathbb{E}\xi]^{k-1}}$.*

**Remark 12** *Observe that when $\alpha > \frac{k}{k-2}$ and $k \geq 3$ it also holds that $1 + \frac{(1-\alpha)(k-2)}{2} < 0$ and the expression in the expectation decreases exponentially in $k$ and $\alpha$, and much faster than the denominator in $C_1$. This shows that the satisfiability threshold of $\mathbb{C}_n^k(\xi)$ is equal $\frac{k}{k-1} + o_k(1)$, when $n \to \infty$ and $k$ is large.*

In the rest of this section we outline a proof of Theorem 11.

In order to prove Theorem 11 we employ a well-known structural property of satisfiable 2-CNFs. A *bicycle* of length $s$ is a sequence of 2-clauses $(u, l_1)(\bar{l}_1, l_2)\ldots\ \ldots(\bar{l}_s, w)$, where variables associated with literals $l_1, l_2, \ldots, l_s \in L(\phi)$ are distinct, and $u, w \in \{l_1, \bar{l}_1, l_2, \bar{l}_2, \ldots, l_s, \bar{l}_s\}$. It was shown in (Chvátal and Reed 1992) that if a 2-CNF does not contain bicycles then it is satisfiable. Therefore it suffices to show that the 2-CNF $\phi'$ produced in the first phase of Algorithm 2 contains no bicycles.

Let $\phi'$ be a 2-CNF formula obtained during the execution of Algorithm 2 from $\phi \sim \mathbb{C}_n^k(\xi)$, where $\xi$ is a r.v. satisfying condition (2) with tail function $F_\xi(\ell) \leq V\,\ell^{-\alpha}$ with $\alpha > \frac{k}{k-1}$ and $V > 0$. It turns out that formula $\phi'$ w.h.p. does not contain not only bicycles of length greater than $\frac{4}{\epsilon}\log n$, but even paths of such length. This can be shown by estimating the expectation of the number of such paths.

On the other hand, $\phi'$ may contain 'short' paths (of length $s \in O(\log n)$), and potentially they can be a part of some short bicycles. However, a bicycle is not only a path, it also needs to sort of close on itself. We show that for short paths this is almost improbable. This again can be done by estimating the expectation of the number of bicycles of such length.

**Lemma 13** *Let $\phi'$ be a 2-CNF formula obtained during the execution of Algorithm 2 from $\phi \sim \mathbb{C}_n^k(\xi)$, where $\xi$ is an r.v. with tail function $F_\xi(\ell) \leq V\,\ell^{-\alpha}$ for some $\alpha > \frac{k}{k-1}$, and which also satisfies condition* (2)*. Then $\phi'$ has no bicycles w.h.p.*

As it was mentioned at the beginning of this section, 2-SAT formulas with no bicycles are satisfiable, hence, $\phi'$ is satisfiable, which means that the original $k$-CNF formula $\phi$ must be satisfiable w.h.p. This finalizes proof of Theorem 11.

## Experiments

In this section we report the results of some computational experiments we conducted with Configuration Model, in

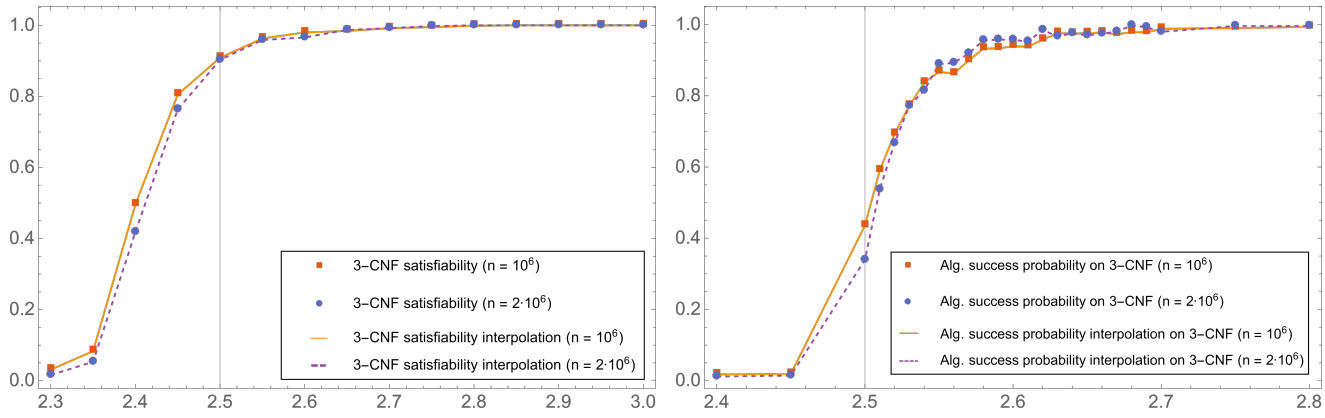


Figure 1: (a) Experimentally obtained satisfiability of $\mathbb{C}_n^k(\xi)$ formulas. (b) Experimentally obtained dependence of the Algorithm 2 success probability (y-axis) vs. $\beta$ (x-axis). Solid gray vertical line at $\beta = 2.5$ marks our lower bound on satisfiability.

which $\xi$ is the zeta distribution. The experiments pursued the following goals:

(a) to estimate the actual location of the satisfiability/unsatisfiability borderline;

(b) to evaluate the success rate of Algorithm 2 for different values of the parameter $\alpha$.

Let $\xi$ be an r.v. distributed according to the zeta distribution $\Pr[\xi = l] = \frac{\ell^{-\beta}}{\zeta(\beta)}$. We ran our experiments for different values of the parameter $\beta > 1$. As it was noted earlier, zeta distributed r.v. $\xi$ is from the class of power-law distributions $P(\alpha)$ with tail function satisfying $F_\xi(\ell) = \Theta\left(\ell^{1-\beta}\right)$.

(a) By the definition of power-law distributions the exponent of the tail $\alpha = \beta - 1$. Hence, since $\xi \sim P(\beta - 1)$, by Theorem 9 we do not expect formulas from $\mathbb{C}_n^k(\xi)$ to be satisfiable, when $\beta - 1 < \frac{k}{k-1}$.

We generated two collections of 3-SAT formulas, one with $n = 10^6$ variables and another one with $n = 2 \cdot 10^6$ variables. For each $n \in \{10^6, 2 \cdot 10^6\}$ and every $\beta \in \{2.30, 2.35, 2.40, \ldots, 2.95, 3.00\}$ we created 2048 formulas from $\mathbb{C}_n^3(\xi)$, which we solved using CryptoMiniSAT 5.6.6 with default parameters, and calculated the fraction of satisfiable formulas among those instances (see Figure 1a). Experiments were performed on machines with 32Gb of RAM and Intel Core i7-6700 CPU, and we had 6 threads working in parallel generating instances and solving them on each computer.

Observe that when we doubled the number of variables, the satisfiability dropped for $\beta < 2.5$, which agrees with our lower bound for $\alpha < \frac{k}{k-1}$ (or equivalently, in terms of $\beta$ for zeta-distributed $\xi$, when $\beta < \frac{2k-1}{k-1}$). Also the satisfiability somewhat dropped at $\beta = 2.5$, which supports our belief that the true satisfiability/unsatisfiability borderline lies somewhere above the $\frac{k}{k-1}$ mark. However, it could also be that 2048 instances were not enough to see how satisfiability evolves in the neighborhood to the right of $\beta = 2.5$.

(b) We ran Algorithm 2 on the same instances used for (a) and (b), and also extra 2048 instances for each $n \in \{10^6, 2 \cdot 10^6\}$ and $\beta \in \{2.51, 2.52, \ldots, 2.68, 2.69\}$ (see Figure 1b). Notice that overall the algorithm is quite effective solving $\mathbb{C}_n^k(\xi)$ formulas with zeta-distributed $\xi$. When we doubled the number of variables, its success probability somewhat dropped for $\beta \leq 2.52$, however, after the 2.53 mark the algorithm demonstrates reliable increase in success probability comparing to its efficiency when $n = 10^6$.

Also, looking for an explanation of such a good performance of SAT-solvers on heavy-tailed distributions we followed (Boufkhad et al. 2005) who ran experiments, which suggested that DPLL based SAT-solvers explore many more branches while solving random d-Regular SAT (see Example 4) , in which every literal has approximately the same degree, than a random formula from the uniform model. The hypothesis is that heuristics in the solvers have hard times deciding which branch to pick in d-Regular formulas, since all literals "look" almost the same, while in the uniform model there is some variability in the literals' degree. Naturally, Boufkhad et al. posed an interesting question whether the number of explored branches decreases when the range of degrees increases.

We conducted a similar experiment, where we compared the number of branches that CryptoMiniSAT solver must explore before concluding whether a given formula is SAT/UNSAT, and formulas were sampled from three different random models, i.e. d-Regular SAT, uniform model, and the Configuration Model with zeta distribution. Zeta distribution may be viewed as one of the canonical discrete heavy-tailed distributions, and while in the uniform model the largest degree is $O(\log n)$ w.h.p., heavy-tailed distributions can produce much larger degrees.

The experiment's set up was the following. We created 10,000 instances for each model, for each average variable's degree from the set $\{1.1, 1.2, 1.3, \ldots, 13.2, 13.3, 13.4\}$, and for the number of variables $n \in \{100, 200, 300\}$ (except we did not test d-Regular SAT with $n = 300$, since it was taking too long to solve such instances to collect any reasonable amount of good measurements). Each instance then was solved using CryptoMiniSAT 5.6.6 solver, and the num-

| k | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|
| **Lower bound** | 2.500 | 2.333 | 2.250 | 2.200 | 2.167 | 2.143 | 2.125 | 2.111 |
| **Upper (algorithmic) bound** | 2.616 | 2.400 | 2.294 | 2.231 | 2.190 | 2.160 | 2.139 | 2.122 |
| **Relative gap** | 4.64% | 2.86% | 1.95% | 1.41% | 1.06% | 0.81% | 0.64% | 0.51% |

Table 1: Estimated bounds of the satisfiability threshold $\beta_0$ for $\phi \sim \mathbb{C}_n^k(\xi)$, where $\xi$ is a zeta-distributed r.v.

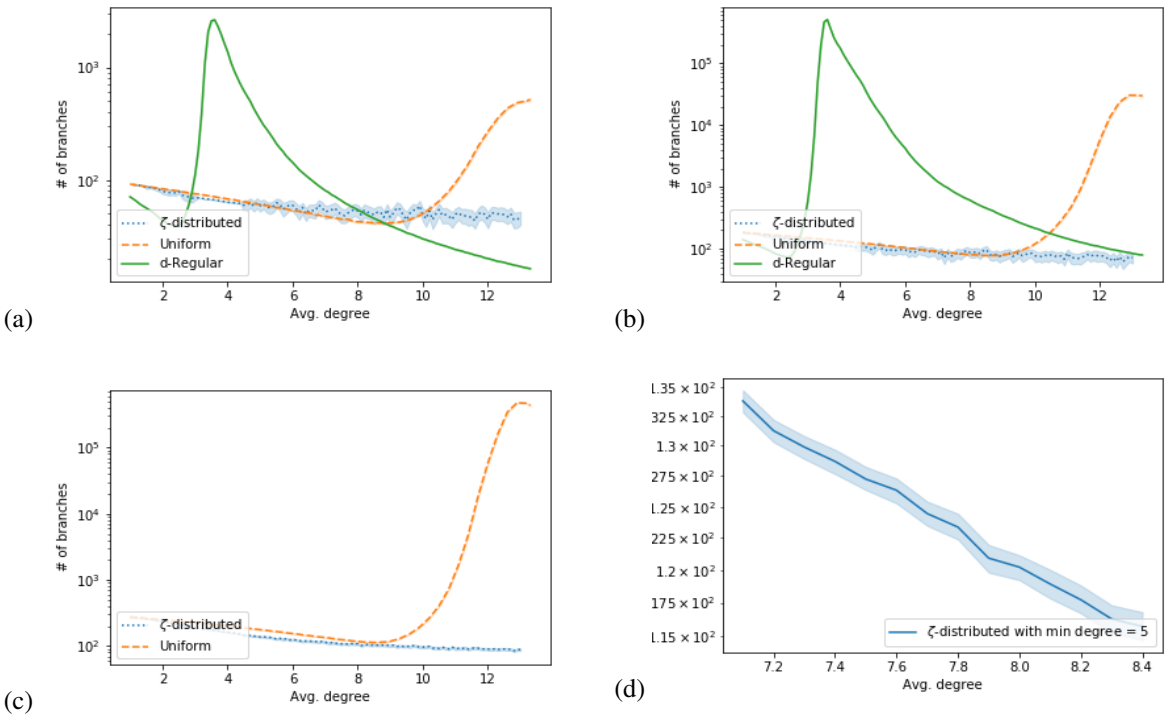


Figure 2: The mean number of explored branches in 3-SAT formulas sampled from different random models. The number of branches is in log scale and differs for each plot; the shaded area around the curves is the 99% confidence interval. (a) Formulas having $n = 100$ variables. (b) Formulas with $n = 200$. (c) Formulas with $n = 300$. (d) The mean number of branches for zeta distributed 3-SAT from the Configuration Model with minimum degree 5 and $n = 300$.

ber of explored branches was recorded for each formula. The aggregated results of the experiment are presented in the Figure 2.

Our results support the hypothesis proposed in (Boufkhad et al. 2005). Even after 15 years after the paper was published, SAT solvers still have troubles solving d-Regular SAT, since the number of decisions they make seem to increase exponentially with the number of variables. Similar troubles (although to slightly lower extent) they have with uniformly distributed instances. However, what strikes the most is the ease solvers have when dealing with the heavy-tailed distributed instances. It seems that the number of branches a DPLL solver explores scales at most linearly with the number of variables. This agrees well with other experiments described above.

However, one may argue that zeta distributed formulas have too many variables of degree 1, which greatly simplifies the search process. However, we also measured the number of branches needed to solve zeta distributed formulas with minimum degree 5, and, as Figure 4(d) shows, it did not make formulas much harder to solve.

Finally, to demonstrate that Algorithm 2 is actually quite effective solving heavy-tailed distributed instances from the Configuration Model despite being myopic, we computationally estimated the lower (Theorem 9) and upper (Theorem 11) bounds of the satisfiability threshold $\beta_0$ for zeta-distributed random $k$-SAT, see Table 1. From the table, it readily follows that the relative gap between the two bounds decreases rapidly, as $k$ grows, and becomes less than 1% even for $k \geq 8$.

## References

Achlioptas, D. 2001. Lower bounds for random 3-SAT via differential equations. *Theor. Comput. Sci.* 265(1-2): 159–185.

Achlioptas, D.; and Moore, C. 2006. Random k SAT: Two

Moments Suffice to Cross a Sharp Threshold. *SIAM Journal on Computing* 36(3): 740–762. ISSN 0097-5397.

Aiello, W.; Graham, F. C.; and Lu, L. 2001. A Random Graph Model for Power Law Graphs. *Experimental Mathematics* 10(1): 53–66.

Ansótegui, C.; Bonet, M. L.; Giráldez-Cru, J.; Levy, J.; and Simon, L. 2016. Community Structure in Industrial SAT Instances. *CoRR* abs/1606.03329. URL http://arxiv.org/abs/1606.03329.

Ansótegui, C.; Bonet, M. L.; and Levy, J. 2009a. On the Structure of Industrial SAT Instances. In *Principles and Practice of Constraint Programming - CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20-24, 2009, Proceedings*, 127–141.

Ansótegui, C.; Bonet, M. L.; and Levy, J. 2009b. Towards Industrial-Like Random SAT Instances. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, 387–392.

Ansótegui, C.; Bonet, M. L.; Levy, J.; and Manyà, F. 2008. Measuring the Hardness of SAT Instances. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, AAAI'08, 222–228.

Barabási, A.-L.; and Albert, R. 1999. Emergence of scaling in random networks. *Science* 286: 509–512.

Bollobás, B.; and Riordan, O. 2002. Mathematical Results on Scale-Free Random Graphs. In *Handbook of Graphs and Networks*, 1–34. Wiley-VCH.

Bollobás, B.; Riordan, O.; Spencer, J.; and Tusnády, G. E. 2001. The degree sequence of a scale-free random graph process. *Random Struct. Algorithms* 18(3): 279–290.

Borovkov, A. A.; Borovkov, K. A. 2008. *Asymptotic Analysis of Random Walks : Heavy-Tailed Distributions*. New York: Cambridge University Press. ISBN 9780521881173.

Boufkhad, Y.; Dubois, O.; Interian, Y.; and Selman, B. 2005. Regular Random $k$-SAT: Properties of Balanced Formulas. *J. Autom. Reasoning* 35(1-3): 181–200.

Chvátal, V.; and Reed, B. A. 1992. Mick Gets Some (the Odds Are on His Side). In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, 620–627.

Clauset, A.; Shalizi, C.; and Newman, M. 2009. Power-law distributions in empirical data. *SIAM Review* 51(4): 661–703.

Coja-Oghlan, A. 2010. A Better Algorithm for Random k-SAT. *SIAM J. Comput.* 39(7): 2823–2864.

Coja-Oghlan, A.; and Panagiotou, K. 2013. Going after the k-SAT threshold. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, 705–714.

Cook, S. A.; and Mitchell, D. G. 1996. Finding hard instances of the satisfiability problem: A survey. In *Satisfiability Problem: Theory and Applications, Proceedings of a DIMACS Workshop, Piscataway, New Jersey, USA, March 11-13, 1996*, 1–18.

Cooper, C.; Frieze, A.; and Sorkin, G. B. 2007. Random 2-SAT with Prescribed Literal Degrees. *Algorithmica* 48(3): 249–265.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem-proving. *Communications of the ACM* 5(7): 394–397. ISSN 00010782.

Davis, M.; and Putnam, H. 1960. A Computing Procedure for Quantification Theory. *Journal of the ACM (JACM)* 7(3): 201–215. ISSN 00045411.

Ding, J.; Sly, A.; and Sun, N. 2015. Proof of the Satisfiability Conjecture for Large k. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 59–68.

Dubios, O.; and Boufkhad, Y. 1997. A General Upper Bound for the Satisfiability Threshold of Random r-SAT Formulae. *J. Algorithms* 24(2): 395–420.

Dubois, O.; Boufkhad, Y.; and Mandler, J. 2003. Typical random 3-SAT formulae and the satisfiability threshold. *Electronic Colloquium on Computational Complexity (ECCC)* 10(007).

Eén, N.; and Biere, A. 2005. Effective Preprocessing in SAT Through Variable and Clause Elimination. In *Theory and Applications of Satisfiability Testing: 8th International Conference, SAT 2005, St Andrews, UK, June 19-23, 2005. Proceedings*, volume 3569 of *Lecture Notes in Computer Science*, 61–75. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 9783540262763.

Franco, J.; and Paull, M. C. 1983. Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem. *Discrete Applied Mathematics* 5(1): 77–87.

Friedgut, E. 1999. Sharp thresholds of graph properties, and the $k$-SAT problem. *J. ACM* 12(4): 1017–1054.

Friedrich, T.; Krohmer, A.; Rothenberger, R.; Sauerwald, T.; and Sutton, A. M. 2017. Bounds on the Satisfiability Threshold for Power Law Distributed Random SAT. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, 37:1–37:15.

Friedrich, T.; and Rothenberger, R. 2019. The Satisfiability Threshold for Non-Uniform Random 2-SAT. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, 61:1–61:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Giráldez-Cru, J.; and Levy, J. 2016. Generating SAT instances with community structure. *Artificial Intelligence* 238(C): 119–134. ISSN 0004-3702.

Jarvisalo, M.; and Biere, A. 2012. Inprocessing Rules. In *Automated Reasoning: 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, volume 7364 of *Lecture Notes in Computer Science*, 355–370. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 9783642313646.

Kim, J. H. 2004. The Poisson Cloning Model for Random Graphs, Random Directed Graphs and Random $k$-SAT Problems. In *COCOON*, 2.

Kirousis, L. M.; Kranakis, E.; Krizanc, D.; and Stamatiou, Y. C. 1998. Approximating the unsatisfiability threshold of random formulas. *Random Struct. Algorithms* 12(3): 253–269.

Krioukov, D. V.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; and Boguñá, M. 2010. Hyperbolic Geometry of Complex Networks. *CoRR* abs/1006.5169.

Li, C.; Fleming, N.; Vinyals, M.; Pitassi, T.; and Ganesh, V. 2020. Towards a Complexity-Theoretic Understanding of Restarts in SAT Solvers. In Pulina, L.; and Seidl, M., eds., *Theory and Applications of Satisfiability Testing - SAT 2020 - 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings*, volume 12178 of *Lecture Notes in Computer Science*, 233–249. Springer. doi:10.1007/978-3-030-51825-7_17. URL https://doi.org/10.1007/978-3-030-51825-7_17.

Marques-Silva, J.; and Sakallah, K. 1999. GRASP: a search algorithm for propositional satisfiability. *IEEE Transactions on Computers* 48(5): 506–521. ISSN 0018-9340.

Moskewicz, M.; Madigan, C.; Zhao, Y.; Zhang, L.; and Malik, S. 2001. Chaff: engineering an efficient SAT solver. In *Proceedings of the 38th annual Design Automation Conference*, DAC '01, 530–535. ACM. ISBN 1581132972. ISSN 0738100X.

Newman, M. 2005. Power laws, pareto distributions and Zipf's law. *Contemporary Physics* 46(5): 323–351.

Omelchenko, O.; and Bulatov, A. 2018. Concentration inequalities for sums of random variables, each having power bounded tail Available at https://arxiv.org/abs/1903.02529.

Omelchenko, O.; and Bulatov, A. A. 2019. Satisfiability Threshold for Power Law Random 2-SAT in Configuration Model. In Janota, M.; and Lynce, I., eds., *Theory and Applications of Satisfiability Testing – SAT 2019*, 53–70. Cham: Springer International Publishing. ISBN 978-3-030-24258-9.

Selman, B.; Mitchell, D. G.; and Levesque, H. J. 1996. Generating Hard Satisfiability Problems. *Artif. Intell.* 81(1-2): 17–29.