

Regular random k -SAT: properties of balanced formulas

Yacine Boufkhad¹, Olivier Dubois², Yannet Interian³, and Bart Selman⁴

¹ *LIAFA, CNRS-Université Denis Diderot- Case 7014, 2, place Jussieu F-75251 Paris Cedex 05 France*

`boufkhad@liafa.jussieu.fr`

² *LIP6, Box 169, CNRS-Université Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France*

`Olivier.Dubois@lip6.fr`

³ *Center for Applied Mathematics. Cornell University, Ithaca, New York 14853*

`interian@cam.cornell.edu`

⁴ *Department of Computer Science. Cornell University, Ithaca, New York 14853*

`selman@cs.cornell.edu`

Abstract. We consider a model for generating random k -SAT formulas, in which each literal occurs approximately the same number of times in the formula clauses (regular random k -SAT). Our experimental results show that such regular random k -SAT instances are much harder than the usual uniform random k -SAT problems. This is in agreement with other results that show that more balanced instances of random combinatorial problems are often much more difficult to solve than uniformly random instances, even at phase transition boundaries. There are almost no formal results known for such problem distributions. The balancing constraints add a dependency between variables that complicates a standard analysis. Regular random 3-SAT exhibits a phase transition as a function of the ratio α of clauses to variables. The transition takes place at approximately $\alpha = 3.5$. We will show that for $\alpha > 3.78$ w.h.p.¹ random Regular 3-SAT formulas are unsatisfiable. We will also show that the analysis of a greedy algorithm proposed in Kaporis *et al* (KKL02) for the uniform 3-SAT model can be adapted for regular random 3-SAT. In particular, we show that for formulas with ratio $\alpha < 2.46$, a greedy algorithm finds a satisfying assignment with positive probability.

1. Introduction

The introduction of new methods for generating random hard instances is an important factor in the development of new search algorithms for satisfiability testing (SAT) (LBS04). In addition, randomly generated SAT problems provide important insights into typical case complexity.

The most popular model for generating random SAT problems is the uniform k -SAT model, formed by selecting uniformly and independently m clauses from the set of all $2^k \binom{n}{k}$ k -clauses on a given set of n variables. Such randomly generated instances exhibit a “phase transition” as a function of the ratio α of clauses to variables (MSL92). Uniform k -SAT problems with a small α value typically have one or more satisfying assignments, whereas problems with a large α value have too many constraints and become unsatisfiable.

Experimental results showing the phase transition phenomenon motivated theoretical interest in understanding uniform k -SAT. The main open question for uniform

¹ The events \mathcal{E}_n hold with high probability (w.h.p.) if $\Pr(\mathcal{E}_n) \rightarrow 1$ when $n \rightarrow \infty$.



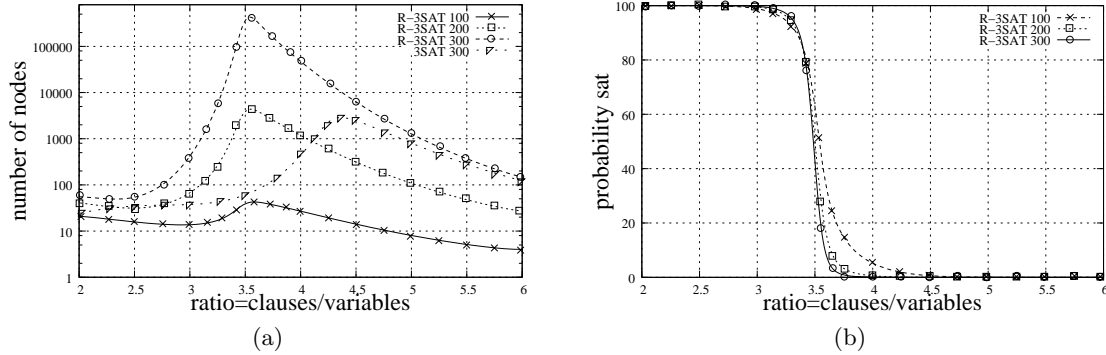


Figure 1. (a) Median of the number of branches needed to solve Reg 3-SAT versus 3-SAT as a function of the ratio α . We consider problems with 100, 200 and 300 variables for Reg 3-SAT and 300 variables for 3-SAT (triangle data points). The plot is in log scale. (b) Phase transition in Reg 3-SAT. Probability that a Reg 3-Sat problem has at least one satisfying assignment as a function of the ratio.

k -SAT concerns the existence of a sharp threshold as the ratio of clauses to variables increases. More precisely, the question is whether there exists constants α_k such that a random formula with $\alpha < \alpha_k$ is satisfiable w.h.p., whereas a random formula with $\alpha > \alpha_k$ is unsatisfiable w.h.p. For $k = 2$, Chvatal and Reed (VB92), Goerdt (Goe96) and Fernandez de la Vega (dlV92) independently proved the existence of the sharp threshold at $\alpha_2 = 1$. For $k \geq 3$, much less is known. Friedgut (Fri99) proved the existence of a sharp threshold around a critical sequence of values. In particular, he showed that there exists a function $\alpha_k(n)$ such that when the number of clauses is around $\alpha_k(n)n$ the satisfiability of the formula drops abruptly from near 1 to near 0. However, these results do not provide information about the value of $\alpha_k(n)$ and its dependence on n .

For uniform 3-SAT there has been a number of results on bounds for the threshold α_3 (see (Ach01) for a survey); the best known result for the lower bound proves that a random uniform instance for 3-SAT is satisfiable w.h.p. if $\alpha < 3.52$ (KKL03; HS04). The best known result for upper bounds states that for $\alpha > 4.506$, random uniform 3-SAT formulas are unsatisfiable w.h.p. (DBM00) (for a survey of upper bounds see (Dub01)). For general k -SAT, the best known bounds are in (AP04; AM02) for lower bounds and in (DB97) and with a slightly less precise method in (KKKS98) for upper bounds.

In this paper we give experimental and theoretical results for a different model for random satisfiability, which we call regular k -SAT (Reg k -SAT). In this model, each literal has nearly the same number of occurrences in the formula. More specifically, given α , the expected ratio of clauses to variables, and n , the number of variables, let $r = \frac{k\alpha}{2}$ be the expected number of occurrences of each literal in the formula. We will generate instances such that each literal appears $\lfloor r \rfloor$ or $\lfloor r \rfloor + 1$ times in the formula.

In figure 1(a), we first consider the computational properties of the Reg k -SAT model. We plot the complexity of experimentally solving uniform 3-SAT and Reg 3-SAT as a function of the ratio α , using the **knfs** solver (DD01). The hardest problems with 300 variables for uniform 3-SAT require less than 4000 branches (median cost) while for the same number of variables Reg 3-SAT requires around $5e + 05$ branches

— more than two orders of magnitude difference. The same hardness is observed with an other complete SAT solver *satz* (Li) and with incomplete solver WalkSAT. Bayardo and Schrag (BS96) reported comparable results on a model similar to the one we present here. (In the Bayardo and Schrag (BS96) model each literal has at least $\lfloor r \rfloor$ but in general could have more than $\lfloor r \rfloor + 1$ occurrences.) Reg 3-SAT also exhibits a phase transition similar to that of uniform 3-SAT. However, the transition is at a quite different ratio: around $\alpha = 3.5$, Reg 3-SAT instances change from satisfiable to unsatisfiable (see figure 1(b)). As one might expect, the figures show that the complexity peak and the phase transition coincide.

Achlioptas *et al.* (AGKS00) introduced a generator of satisfiability formulas based on Latin squares that creates only satisfiable instances. More recently, that model was modified to obtain a more “balanced” version (KRA⁺01), thereby significantly increasing the difficulty of the instances. As in the comparison of uniform 3-SAT versus Reg 3-SAT, in these generators the effect of balancing dramatically increases the hardness of the problem. Another example of this phenomenon appears in coloring random graphs. When considering the Erdős-Rényi model $G(n, p = \frac{r}{n})$ versus the regular graphs $G(n, r)$ with the same average degree r , regular graphs are much harder to color than graphs in $G(n, p)$.

It is interesting to consider for a moment why solvers have so much more trouble with regular or balanced problem instances. The key issue appears to be that in the standard uniform random formula and graph models, solvers can exploit variations between variable occurrences (or node degrees). In particular, most solvers will first focus on variables that occur relatively frequently or nodes with relatively high degree. In the uniform k -SAT model, literal occurrences range from 0 to $\log(n)$, in n variable instances. This is a rather significant range and heuristics for variable selection exploit these differences quite successfully. In the Reg k -SAT model, on the other hand, each literal occurs either r times or $r + 1$ times for some small constant r (independent of n). So, one cannot exploit obvious differences in the frequency of literal occurrences. Setting variables and simplifying the formula may disturb the precise balance of literal occurrences. However, since the maximum literal occurrence is only $r + 1$, the formulas remain nearly balanced with the maximum range of literal occurrences between 0 and $r + 1$. Because of the lack of variation between literal occurrences, these balanced models require the development of solvers with new branching heuristics to tackle them more effectively. We hope that our work will stimulate the development of such new solvers.

Aside from the complexity differences, the fact that the thresholds for the regular and the uniform k -SAT model occur at significantly different locations also suggests that there are interesting differences between the two models. In terms of the bounds on the threshold phenomena in the regular SAT model, we will see how one can exploit the properties of the limited degree variation to obtain bounds that are tighter than the bounds obtained for the uniform random formula model. A deeper understanding as to why these bounds in the regular SAT model are better may also lead us to new insights into the analysis of the uniform SAT model.

An interesting direction for future research is to consider what happens when one pushes the uniform random k -SAT model in the other direction: instead of making them more balanced, make the literal occurrences even less balanced. In particular, one could consider power-law distributions in terms of literal occurrences. This would

be analogous to the work on random graphs, where one has found that power-law distributed node degrees are most prevalent in real-world networks (*e.g.*, the World Wide Web). Real world SAT instances, such as derived from bounded model-checking, also exhibit large variations in literal occurrences. So, a random formula model with power-law literal occurrence distribution would provide an interesting complement to our results for regular SAT.

We begin the next section with a precise definition of our model. We use the results of Cooper *et al.* (CFS02) to derive the sharp threshold for Reg 2-SAT. The threshold for regular 2-SAT is at the same ratio of $\alpha = 1$ as for the uniform random 2-SAT model. So, only for $k > 2$, do the properties of the models diverge in an interesting way. In Section 3, we use the first moment method combined with a subtle argument based on literal occurrences to prove that for $\alpha > 3.78$ a Reg 3-SAT formula is unsatisfiable w.h.p. In Section 4, we analyze a greedy algorithm on Reg 3-SAT formulas to prove that for $\alpha < 2.46$ the algorithm finds a satisfying assignment with positive probability.

2. The model

A k -SAT formula is a finite set of clauses, each clause being a disjunction of k literals over the set of boolean variables.

We are interested in generating random k -SAT formulas where each literal appears in approximately the same number of clauses. For the case $k = 2$, this problem is very similar to the problem of generating a regular random graph. A generalization of the usual procedure to generate random regular graphs is used here to generate random regular k -SAT formulas. For simplicity, suppose we want to generate a random 3-SAT formula in which each literal appears exactly 4 times. We take a box in which we place 4 copies of each literal. If n is the number of variables, we have $4 \times 2n$ literals in the box. To form a clause, we take 3 literals from the box without replacement. We continue until we have $m = 8n/3$ clauses. The problem with this procedure is that we may obtain “illegal” clauses, that is, clauses in which a variable appears more than once. If that happens, we start the process again. In practice, instead of restarting, we can also just erase the illegal clauses.

With the algorithm described above one can obtain formulas in which all literals appear exactly r times, for nonnegative integers r . Therefore, we get just some values of the ratio $\alpha = 2r/3$. We generalize this procedure to obtain formulas with average ratio α for every real α . In essence, to obtain a balanced SAT formula with a ratio α that lies in between $2r/3$ and $2(r+1)/3$, for some value of r , we will create a random balanced formula where each literal has either r occurrences or $r+1$ occurrences. The ratio of the number of literals with r occurrences to the number with $r+1$ occurrences will be chosen carefully to obtain the desired value of α . Our model is inspired by the way random graphs with prescribed literal degrees have been defined.

We first introduce the notion of the literal degree sequence of a formula. Let n be the number of variables, $m = \lfloor \alpha n \rfloor$, $\alpha > 0$, the number of clauses in a k -CNF formula F . We say that a literal x has degree l if x appears l times in the formula. Let $r = k\alpha/2$ be the average literal degree. The degree sequence associated with a formula F is the sequence $\{d_1, d_{-1} \dots d_n, d_{-n}\}$ where d_x is the number of clauses in which the literal

x occurs. So, the degree sequence simply tells us how often each literal occurs in the formula.

The actual formula generation process will consist of two steps. First, we randomly generate a desired degree sequence for our formula. Then, to obtain our balanced formula, we randomly generate a set of clauses that satisfies this degree sequence.

Let $\{p_l\}_{l \geq 0}$ with $\sum_{l \geq 0} p_l = 1$ be a sequence of non negative real numbers, where p_l is the probability of a literal having degree l . In our regular SAT model, this sequence of probabilities is very simple: $p_{\lfloor r \rfloor} = p$ and $p_{\lfloor r \rfloor + 1} = 1 - p$, where p is defined so that the expected number of clauses is $m = \alpha n$, i.e., $p = \lfloor r \rfloor + 1 - r$, and all the other values for p_l are zero. Given these probabilities for each degree, we can generate a sequence of actual literals degrees $\{d_1, d_{-1} \dots d_n, d_{-n}\}$ drawn independently from that distribution and conditioned on the event that the sum of all degrees is a multiple of k . Note that in our regular SAT model, each literal will either have degree $\lfloor r \rfloor$ or $\lfloor r \rfloor + 1$ in this degree sequence.

After having obtained a literal degree sequence for our random formula, we generate a random formula with this degree sequence. To do so, we generalize the example discussed at the beginning of this section. Let W_d be the set of literals associated with a degree sequence d where literal l appears d_l times, i.e., $\mathcal{D} = |W_d| = \sum_l d_l$. A *configuration* F is a partition of W_d into \mathcal{D}/k groups of k literals. For each configuration we obtain a formula with the desired degree sequence by assigning literals in one group to literals in a clause. The problem with that mapping is that some of these clauses may not be “legal”. A legal clause is one in which there are no repeated or complementary literals. Call a *configuration formula* a formula that is not necessarily legal as opposed to a *simple formula*, one with legal clauses. In the context of regular graphs, this procedure is known as the *configuration model* (JLR00).

For the analysis of section 4, we need a slightly more general configuration model. For simplicity consider the case in which we have a 3-SAT formula. After we set some variables, and remove unit clauses by unit propagation, the formula will consist of a mixture of 2 and 3-clauses and a certain degree sequence. Let W_d be the set of literals associated with a degree sequence d . To obtain a configuration formula with degree sequence d , C_2 2-clauses, and C_3 3-clauses such that $\mathcal{D} = |W_d| = \sum_l d_l = 2C_2 + 3C_3$, we partition W_d randomly in C_2 groups of 2 and C_3 groups of 3 literals and associate each group with a clause.

Next lemmas will help us to extend properties of the configuration formulas to properties of simple formulas. Let $\Pr(\text{SIMPLE})$ be the probability that a configuration formula is simple.

Lemma 1. *Let $m_2 = an$, $m_3 = bn$, $a \geq 0$, $b \geq 0$, $a+b > 0$, and let $d = \{d_1, d_{-1} \dots d_n, d_{-n}\}$, a bounded degree sequence $d_i + d_{-i} < \Delta$, for some constant Δ . Let F be a configuration formula with n variables, m_i i -clauses $i = 1, 2$, and degree sequence d (where $\sum_i d_i + d_{-i} = 2an + 3bn$), then there exists a constant $\delta > 0$ such that,*

$$\Pr(F \text{ is SIMPLE}) \rightarrow \delta > 0 \text{ as } n \rightarrow \infty$$

Applying previous result for $a = 0$ we get the following corollary.

Corollary 1. *If F is a 3-Reg formula there exist $\delta > 0$ such that*

$$\Pr(F \text{ is SIMPLE}) \rightarrow \delta > 0 \text{ as } n \rightarrow \infty$$

Lemma 2. *Let F as in the hypothesis of lemma 1, let y a fixed variable, the probability that we have a clause with 2 occurrences of the variable y bounded by C/n .*

The proofs of these lemmas are in the appendix.

2.1. REG 2-SAT

Let $d = \{d_1, d_{-1} \dots d_n, d_{-n}\}$, a degree sequence corresponding to a 2-SAT formula. In the following theorem, we limit the maximum degree in the degree sequence. To do so, we say that d is Δ -proper, if $d_i < \Delta$ for $i \in \{1, -1, \dots, n, -n\}$, where Δ is a constant depending on n .

The location of the threshold for the Reg 2-SAT model can be derived using the following theorem.

Theorem 1. (CFS02) *Let $0 < \epsilon < 1$ and $n \rightarrow \infty$. Let d be any Δ -proper degree sequence over n variables, with $\Delta = n^{1/11}$, and let F be a uniform random simple formula with degree sequence d , then*

$$\text{If } D < (1 - \epsilon)m \text{ then } P(F \text{ is satisfiable}) \rightarrow 1$$

$$\text{If } D > (1 + \epsilon)m \text{ then } P(F \text{ is satisfiable}) \rightarrow 0$$

where m is the number of clauses and $D = \sum_{i=1}^n d_i d_{-i}$

Corollary 2. *The Reg 2-SAT formulas have a threshold at $\alpha = 1$.*

Proof. We prove that w.h.p. degree sequences generated with our Reg 2-SAT model have the property that $\frac{D}{m} \rightarrow \alpha$. Using theorem 1 we can conclude that $\alpha = 1$ is the value of the threshold.

Let $D = \sum_{i=1}^n d_i d_{-i}$ a random variable. Note that the expected value $E(D)$ of D is $\alpha^2 n$ and $E(m) = \alpha n$. Note that $2m = \sum_{i=1}^n d_i + d_{-i}$; the variables d_i , $i \in \{1, -1, \dots, n, -n\}$ are independent identically distributed random variables. The variance of the variables D and m are easy to compute, and there exist constants c, c' such that $Var(D) = cn$ and $Var(m) = c'n$.

Using Chebyshev's inequality, we get that $P(|D - \alpha^2 n| \geq n^{1/2+\delta}) \rightarrow 0$ as n goes to infinity for any $\delta > 0$. A similar property follows for the variable m , $P(|m - \alpha n| \geq n^{1/2+\delta}) \rightarrow 0$ as n goes to infinity. Therefore the property follows and then the claim. \square

3. Upper bound on the threshold

In order to estimate the probability that a random formula is satisfiable, we bound that probability by the expected number of solutions, *i.e.*,

$$\Pr(F \text{ is sat}) \leq \mathbf{E}(\# \text{ solutions } F) = 2^n \Pr(x \text{ is a solution}) \quad (1)$$

The last equality follows from the fact that the occurrences of each literal has the same distribution. All assignments $x \in \{0, 1\}^n$ have the same probability of being a solution. The use of the first inequality in (1) is known as the first moment method.

For a clauses to variables ratio α , let $q = \lfloor 3\alpha n/2 \rfloor - \lfloor 3\alpha/2 \rfloor n$. In a configuration formula, a subset of q among n variables chosen uniformly at random will have $\lfloor 3\alpha/2 \rfloor + 1$ positive copies and $\lfloor 3\alpha/2 \rfloor + 1$ negative copies. The remaining $n - q$ variables will have $\lfloor 3\alpha/2 \rfloor$ copies for each sign. (If $3\alpha n$ is odd then a literal chosen randomly will have a positive or negative copy more than the copies of opposite sign but this has a negligible effect on the calculation of the expectation). For the following we define $\mu = q/n$.

Thanks to lemma 1, it is sufficient to compute a bound on configuration formulas. Let $\Pr(\text{SIMPLE})$ be the probability that a configuration formula is simple.

Let $\Pr(\text{SAT})$ be the probability that a configuration formula is satisfiable and $\Pr_0(\text{SAT})$ the probability that a simple formula is satisfiable. We have :

$$1 - \frac{1 - \Pr(\text{SAT})}{\Pr(\text{SIMPLE})} \leq \Pr_0(\text{SAT}) \leq \frac{\Pr(\text{SAT})}{\Pr(\text{SIMPLE})}$$

Thanks to Lemma 1 and the preceding inequalities, the threshold for simple formulas has the same location as the threshold for configuration formulas, if it exists.

We consider a truth assignment T and compute the number of Reg-3-SAT formulas satisfied by T . A clause is said to be of type i if it contains i true literals with respect to T . Denote by δ_i the fraction of clauses of type i in a formula.

In the uniform 3-SAT model, formulas are typically satisfied by some fixed truth assignment in such a way that the proportion of each clause type concentrates around its mean, namely, $\delta_1 = 3/7$, $\delta_2 = 3/7$ and $\delta_3 = 1/7$. This follows from the following observation. A random uniform 3-SAT formula F is obtained by taking uniformly at random $m = \alpha n$ clauses out of the $8 \binom{n}{3}$ possible clauses. A formula that satisfied a certain assignment T is formed by taking $m = \alpha n$ clauses from the $7 \binom{n}{3}$ possible clauses that satisfy T . From these $7 \binom{n}{3}$ clauses $3 \binom{n}{3}$ are of type 1, $3 \binom{n}{3}$ are of type 2 and $\binom{n}{3}$ are of type 3.

Thus, the number of true literals is $\delta_1 \alpha n + 2\delta_2 \alpha n + 3\delta_3 \alpha n = \frac{12}{7} \alpha n$ and the number of false literals is $2\delta_1 \alpha n + \delta_2 \alpha n = \frac{9}{7} \alpha n$. So in the uniform model the typical formulas are satisfied by truth assignments that skew true and false literals in favor of true ones. Clearly, the random regular formulas are not among these typical formulas because whatever truth assignment one considers, the number of true and false literals is equal to $3\alpha n/2$ (simply because each variable and its negation occur equally often in the

formula). Thus, by restricting the δ_i to take only values that satisfy the balance between signs, one can expect to get a better bound than 5.19. Indeed, we establish:

Theorem 2. Let $\beta = \frac{9-3\sqrt{5}}{4}$ and $\alpha^* = \frac{\log(2)}{3\log(2)+\beta\log(\frac{2}{3})+(\frac{3}{2}-2\beta)\log(\frac{1}{2}-\frac{2}{3}\beta)+(\beta-\frac{1}{2})\log(\beta-\frac{1}{2})} \simeq 3.7822$ If $\alpha > \alpha^*$ then w.h.p. every formula is unsatisfiable.

Proof. In the following, a configuration formula is viewed as $3\alpha n$ ordered cells such that cells numbered 1,2, and 3 form the first clause, 4, 5 and 6 the second clause and so on... A formula is then built by assigning a literal to each cell. Then the total number of formulas is:

$$\binom{n}{\mu n} \frac{(3\alpha n)!}{([\lfloor 3\alpha/2 \rfloor + 1]!)^{2\mu n} [\lfloor 3\alpha/2 \rfloor!]^{2(1-\mu)n}}$$

Now, we count the number of Reg-3-SAT formulas satisfied by T . There are

$$\frac{(\alpha n)!}{(\delta_1 \alpha n)! (\delta_2 \alpha n)! (\delta_3 \alpha n)!} 3^{\delta_1 \alpha n} 3^{\delta_2 \alpha n}$$

ways to choose clauses of each type and to choose the cells for the i true literals in clauses of type i with the δ_i subject to the following constraints :

$$\delta_1 + \delta_2 + \delta_3 = 1 \quad (2)$$

and

$$\delta_1 + 2\delta_2 + 3\delta_3 = 3/2 \quad (3)$$

At this point, the cells that will be filled with true and false literals are fixed. It remains to fill them with literals. There are :

$$\binom{n}{\mu n} \left(\frac{[\lfloor 3\alpha n/2 \rfloor!]}{([\lfloor 3\alpha/2 \rfloor + 1]!)^{\mu n} [\lfloor 3\alpha/2 \rfloor!]^{(1-\mu)n}} \right)^2$$

ways to fill correctly these cells with literals.

To sum up the probability that T satisfies a random Reg-3-SAT formula is :

$$\Pr(T \text{ satisfies } F) = \left(\frac{[\lfloor 3\alpha n/2 \rfloor!]}{([\lfloor 3\alpha/2 \rfloor + 1]!)^{\mu n} [\lfloor 3\alpha/2 \rfloor!]^{(1-\mu)n}} \right)^2 \frac{([\lfloor 3\alpha/2 \rfloor + 1]!)^{2\mu n} [\lfloor 3\alpha/2 \rfloor!]^{2(1-\mu)n}}{(3\alpha n)!} \\ \sum_{(\delta_1, \delta_2, \delta_3) \in A} \frac{3^{\delta_1 \alpha n} 3^{\delta_2 \alpha n} (\alpha n)!}{(\delta_1 \alpha n)! (\delta_2 \alpha n)! (\delta_3 \alpha n)!}$$

where

$$A = \{(\delta_1, \delta_2, \delta_3) / \text{for some integers } \Delta_1, \Delta_2, \Delta_3 \\ \delta_1 = \frac{\Delta_1}{n}, \delta_2 = \frac{\Delta_2}{n} \text{ and } \delta_3 = \frac{\Delta_3}{n} \text{ subject to constraints (2) and (3)}\}$$

After simplification :

$$\Pr(T \text{ satisfies } F) = \frac{([\lfloor 3\alpha n/2 \rfloor!])^2}{(3\alpha n)!} \sum_{(\delta_1, \delta_2, \delta_3) \in A} \frac{3^{\delta_1 \alpha n} 3^{\delta_2 \alpha n} (\alpha n)!}{(\delta_1 \alpha n)! (\delta_2 \alpha n)! (\delta_3 \alpha n)!}$$

We use the fact that $|A| = O(n^3)$ (i.e. the number of terms of the sum is bounded by a polynome) and the following inequality :

$$\left(\frac{p}{e}\right)^p \sqrt{2\pi p} \leq p! \leq \left(\frac{p}{e}\right)^p \sqrt{2\pi p} \left(1 + \frac{1}{12p-1}\right)$$

we get the exponential order of expectation of the number of solutions :

$$\begin{aligned} \mathbf{E}(\# \text{ solutions of } F) &= 2^n \mathbf{Pr}(T \text{ satisfies } F) \\ &\asymp \max_{(\delta_1, \delta_2, \delta_3) \in A} \left(2^{-3\alpha} 2 \left(\frac{3}{\delta_1}\right)^{\delta_1 \alpha} \left(\frac{3}{\delta_2}\right)^{\delta_2 \alpha} \left(\frac{1}{\delta_3}\right)^{\delta_3 \alpha} \right)^n \end{aligned}$$

Let $f_\alpha(\delta_1, \delta_2, \delta_3) = 2^{-3\alpha} 2 \left(\frac{3}{\delta_1}\right)^{\delta_1 \alpha} \left(\frac{3}{\delta_2}\right)^{\delta_2 \alpha} \left(\frac{1}{\delta_3}\right)^{\delta_3 \alpha}$. The problem amounts to maximize $f_\alpha(\delta_1, \delta_2, \delta_3)$ subject to constraints¹ (2) and (3). (The calculation for general Reg- k -SAT gives an analogous function $f_\alpha(\delta_1, \delta_2, \dots, \delta_k) = 2^{-k\alpha} 2 \prod_{i=1..k} \left(\frac{k}{\delta_i}\right)^{\delta_i \alpha}$, the maximization is subject to analogous constraints and can be solved using standard Lagrange maximization method.)

From (2) and (3), we have $\delta_2 = 3/2 - 2\delta_1$ and $\delta_3 = \delta_1 - 1/2$. f_α can then be expressed in terms of δ_1 alone. By applying \log , we have to maximize the following function of δ_1 :

$$\begin{aligned} g_\alpha(\delta_1) = & (1 - 3\alpha)\log(2) + \alpha(3/2 - \delta_1)\log(3) - \alpha\delta_1\log(\delta_1) \\ & - \alpha(3/2 - 2\delta_1)\log(3/2 - 2\delta_1) - \alpha(\delta_1 - 1/2)\log(\delta_1 - 1/2) \end{aligned}$$

which attains its maximum at $\delta_1^* = \frac{9-3\sqrt{5}}{4}$.

The upper bound is obtained by solving w.r.t. α the equation $g_\alpha(\delta_1^*) = 0$. \square

4. Lower bound on the threshold

In this section, we analyze a greedy algorithm and prove that it finds a satisfying assignment with positive probability for formulas with $\alpha < 2.46$. This result by itself does not give a lower bound on the threshold. To prove a lower bound we need that property to hold w.h.p. In previous work on lower bound for uniform 3-SAT, this problem was solved by using the result in (Fri99) that implies that if such property holds with positive probability it also holds w.h.p. That result is not known for our model.

We first prove that configuration formulas with $\alpha < 2.46$ are satisfiable with positive probability. At the end of the section we discuss how to modify the proof to get the same result for simple formulas.

¹ By ignoring the constraint of balancing between signs $\delta_1 + 2\delta_2 + 3\delta_3 = 3/2$, the maximum is at $\delta_1 = 3/7$, $\delta_2 = 3/7$, $\delta_3 = 1/7$ and the bound is 5.19 as for the uniform 3-SAT model. Surprisingly, the fact that all variables have almost the same number of occurrences disappears in the expectation. There remains only the constraint that each variable have the same number of positive and negative occurrences.

To achieve our claim, it is enough to prove that with positive probability the algorithm does not generate empty clauses. The algorithm we analyze makes n iterations setting one variable at each iteration. Keeping track of the number of 2-clauses at each iteration of the algorithm and checking that for $\alpha < 2.46$ the density of 2-clauses is bounded below 1 is sufficient to obtain the results.

The method of differential equations proposed in (Wor95) is used in this analysis to keep track of the number of 2-clauses at each time. In particular, our analysis follows closely the one done by Kaporis *et al* (KKL02) for random uniform 3-SAT formulas. This is a well known approach that has been used in most of the lower bounds for the threshold of random uniform 3-SAT (e.g. see (Ach01; KKL02; KKL03; HS04)).

For a fixed ratio α and the corresponding $r = 3\alpha/2$, let h be the smallest integer greater than r . Let \mathcal{X}_j for $j = 0, \dots, h$ be the current collection of literals of degree j . We consider an algorithm that at each time sets a literal with the higher occurrence in the formula and sets unit clauses anytime they appear. Let's call a *round* the first while loop of the following algorithm. In a round we assign a random literal and unit clauses that may appear.

Greedy algorithm

```

begin
  let j=h
  while unset literals exists
    while  $\mathcal{X}_j \neq \emptyset$ 
      set an arbitrary literal from  $\mathcal{X}_j$  to TRUE
      and its negation to FALSE and Del&Shrink
      while unit clauses exists
        set an arbitrary unit clause to True
        its negation to FALSE and Del&Shrink
      end
    end
     $j = j - 1$ 
  end
end

```

Note that in the process of choosing the literal to be assigned to true at the beginning of each round the greedy algorithm does not use any information about the negation of that literal. Therefore the negated literal is random, in particular it's degree has the same distribution as any other literal in the formula.

As we already mentioned, the analysis of the algorithm relies on the method of differential equations described in (Wor95). The idea is as follows: suppose $Y^t = (Y_1^t, \dots, Y_s^t)$ are stochastic parameters related to a formula. In our case, these parameters are the number of variables (literals), the number of 2-clauses and 3-clauses, and the number of literals of degree i , $0 \leq i \leq h$, in the formula at time t . We want to estimate the trajectory of Y^t through the duration of our algorithm. In a restricted version, the theorem states that if

(a) $\Pr(|Y^{t+1} - Y^t| > n^{1/5}) = o(n^{-3})$

(b) $\mathbf{E}(Y^{t+1} - Y^t \mid Y^t) = f(t/n, Y^t/n) + o(1)$

(c) the function f is continuous and satisfies a Lipschitz condition on some set D , then

$$Y^t = ny(t/n) + o(n) \quad (4)$$

where $y(x)$ is the solution of the system of differential equations

$$\frac{dy}{dt} = f(x, y) \quad y(0) = \frac{Y^0}{n} \quad (5)$$

The precise statement of the theorem is given in the appendix.

In order to make use of the previous theorem to analyze an algorithm one needs to choose the algorithm, the set of parameters Y^t , and the random model, to satisfy the following property: after each round the resulting formula is random, given the values of the parameters in Y^t . Precisely, let $Y^t = (L(t), C_3(t), C_2(t), X_1(t), \dots, X_h(t))$, where $L(t)$ is the number of unset literals at time t , $C_3(t)$ and $C_2(t)$ is respectively the number of 3-clauses and 2-clauses, and X_i is the number of literal with degree i , $0 \leq i \leq h$, *i.e.*, we need the following lemma.

Lemma 3. (KKL02) *During the evolution of the algorithm, the formula remains random conditional on the current value of the parameters Y^t .*

Using lemma 3 we are able to compute the expression in (b). In the appendix we give the expression for the function f in our case. Equation (4) allows us to use the solution of the differential equations (5) to trace our parameter Y^t . In particular we use the values of $C_2(t)$ to prove that with positive probability the algorithm does not generate empty clauses.

If the density of 2-clauses $\frac{2C_2(t)}{L(t)}$ is bounded below 1 that property holds. A proof for that result can be found in (KKL02). The argument goes as follows: The number of unit clauses generated in one of the while loops in the algorithm can be approximated by a branching process. Suppose we start by satisfying a unit clause. That assignment can produce some new unit clauses that can be seen as the offspring of the first unit clause. The process continues until all unit clauses are satisfied. The expected number of unit clauses produced by one assignment has mean $\frac{2C_2(t)}{L(t)}$. If $\mu = \frac{2C_2(t)}{L(t)} < \rho < 1$ for some constant ρ and for all t , the process is subcritical, and its expected size is $1/(1-\mu)$. The probability of the appearance of a literal b and its complement \bar{b} in a single round can be proved to be less than C/n for some constant $C > 0$ independent of t and therefore the probability of not having an empty clause is bounded by $(1 - O(1/n))^n$, which is greater than $e^{-C'}$ for some constant C' .

Equation (4) holds for values of t such that the scaled number of literals $l(t) = L(t)/n > \epsilon$ for any fixed $\epsilon > 0$. Using the previous argument we are able to prove that for $t < t^*$ our main claim holds with positive probability. To finish the proof we use theorem 1 from section 2 to prove that the remaining formula is satisfiable w.h.p. We choose t^* such that the degree sequence at time t^* satisfies the conditions of theorem 1. Note that if we delete one literal from the every 3-clause we get a 2-SAT formula satisfiable w.h.p. and so the original formula is also satisfiable.

We solve the differential equations (see appendix B.1), associated with the equation (5), numerically using the *ode45* function of matlab. The results are in agreement with

simulation of the algorithm on randomly generated formulas. We find that for $\alpha = 2.46$, $\frac{2c_2(x)}{l(x)}$ is bounded below 1 for $x < x^*$. We can conclude the following result.

Theorem 3. *Let F be a configuration formula with ratio $\alpha < 2.46$. The greedy algorithm finds a satisfying assignment for F with positive probability.*

To extend theorem 3 for simple formulas we just have to introduce some minor changes in the proof. By *exposing* a variable x we mean disclosing the information pertaining to this variable (*e.g.* the number of occurrences of x and \bar{x} in i -clauses). At each step we expose the variable we want to assign, some of the information we expose is whether two literals from the same variable are in a single clause. Let us call such event a *bad* event. Also the appearance of an empty clause is considered a bad event. We are going to prove that the probability of a bad event in a round t is bounded by C/n for some constant $C > 0$ independent of t .

Claim: At any round the probability of having a bad event is less than C/n , for some $C > 0$.

If T is the number of steps in a round, at that round we expose the information about T variables. By lemma 2, after exposing one variable the probability of getting a cycle is bounded by C'/n , where C' is independent of the round. The probability that we get a cycle in the round conditioned on the number of steps in a round being T , is bounded by TC/n . The unconditional probability is less than $\mathbf{E}(T)C/n$ ($\mathbf{E}(T)$ can also be bounded independently of the round). These facts, together with the fact that the probability of getting an empty clause is bounded by C''/n independently of the round completes the claim.

To finish we have to argue that the final subformula is simple and satisfiable with positive probability. From our previous discussion we know it is satisfiable. It is simple with positive probability by lemma 1. We conclude the following result.

Theorem 4. *Let F be a Reg 3-SAT formula with ratio $\alpha < 2.46$. The greedy algorithm finds a satisfying assignment for F with positive probability.*

5. Conclusions

We proposed a new model for random k -SAT, in which every literal appears in approximately the same number of clauses. Experimental results show that the new model leads to formulas that are substantially more difficult to solve than the well known uniform k -SAT model. Experiments also show that the model exhibits a phase transition as a function of α , the ratio of clauses to variables. The hardest instances are concentrated around the value $\alpha = 3.5$, where the probability that a formula is satisfiable falls from 1 to 0. We provide the first rigorous bounds for this model. In particular, we obtain an upper bound of 3.78 for the location of the phase transition. This bound was obtained by exploiting the special balanced nature of the formulas. Our analysis of a greedy algorithm shows that we can find satisfying assignments for formulas with $\alpha < 2.46$ with positive probability.

The underlying theme of this research is to develop interesting alternatives to the uniform random k -SAT model. In this paper, we have changed the almost Poisson degree

distribution of the uniform random k -SAT to an almost constant degree distribution of the regular random k -SAT model. As we discussed in the introduction, given the limited variance in degrees, solvers have much more difficulty identifying good variables to branch on. It will require new ideas for branching heuristics or different techniques altogether to make progress on these formulas. (We did some preliminary experiments with WalkSAT and survey propagation. Again, these formulas appear much harder than uniform random k -SAT instances.) Eventually, we hope to develop other analyzable models that are closer to real world instances. One interesting possibility in this regard is to consider random formulas with power law distributed literal degree distributions.

There are still other results to pursue for regular SAT. To get a proper lower bound using our result for the greedy algorithm, one has to extend the proof of Friedgut (Fri99) of the existence of a sharp threshold around some critical sequence of values. Another interesting problem is to extend the results of Chvatal and Szemerédi (CS88). Chvatal and Szemerédi proved that for the uniform k -SAT model for $k \geq 3$, unsatisfiable formulas need an exponential refutation proof. Their proof extends to our model. But, more interestingly, by exploiting the additional balanced structure in our model as we did for the upper bound result, we may be able to obtain sharper results.

References

- Dimitris Achlioptas. Lower bounds for random 3-sat via differential equations. *Theoretical Computer Science*, 265:159–185, 2001.
- Dimitris Achlioptas, Carla P. Gomes, Henry A. Kautz, and Bart Selman. Generating satisfiable problems instances. In *Proceedings of 17th National Conference on Artificial Intelligence*, pages 256–261, 2000.
- Dimitris Achlioptas and Christopher Moore. The asymptotic order of the random k -sat threshold. In *43th Annual Symposium on Foundations of Computer Science*, pages 779–788, Vancouver, 2002.
- Dimitris Achlioptas and Yuval Peres. The Threshold for Random k -SAT is $2^k \ln 2 - O(k)$. *Journal of the AMS*, 17:947–973, 2004.
- Bela Bollobas. *Random Graphs Second Edition*. Cambridge University Press, United Kingdom, 2001.
- Roberto J. Bayardo and Robert Schrag. Using csp look-back techniques to solve exceptionally hard sat instances. In *Proc. of the Second Int. Conf. on Principles and Practice of Constraint Programming*, pages 46–60, 1996.
- Colin Cooper, Alan Frieze, and Gregory B. Sorkin. A note on random 2-sat with prescribed literal degrees. *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2002.
- V. Chvatal and E. Szemerédi. Many hard examples for resolution. *Journal of the Association for Computing Machinery*, 35:759–768, 1988.
- Olivier Dubois and Yacine Boufkhad. A general upper bound for the satisfiability threshold of random r -sat formulae. *J. Algorithms*, 24(2):395–420, 1997.
- Olivier Dubois, Yacine Boufkhad, and Jacques Mandler. Typical random 3-sat formulae and the satisfiability threshold. In *SODA*, pages 126–127, 2000. Full version in Electronic Colloquium on Computational Complexity (ECCC 2003).
- Olivier Dubois and Gilles Dequen. A backbone search heuristic for efficient solving of hard 3-sat formulae. In *Proceedings of 17th International Joint Conference on Artificial Intelligence*, pages 248–253, Seattle, 2001.
- W. Fernandez de la Vega. On random 2-sat. *Manuscript*, 1992.
- Olivier Dubois. Upper bounds on the satisfiability threshold. *Theoretical Computer Science Volume*, 265:187–197, 2001.

- E Friedgut. Sharp thresholds for graph properties and the k -sat problem. *Journal of the American Mathematical Society*, 12:1017–1054, 1999.
- A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53(3):469–486, 1996.
- M. Hajiaghayi and G. Sorkin. The satisfiability threshold of random 3-sat is at least 3.52. www-math.mit.edu/hajiagha/3satRC1.ps, 2004.
- S. Janson, T. Luczak, and A. Rucinski. *Random Graphs*. John Wiley and Sons, New York, 2000.
- Lefteris M. Kirousis, Evangelos Kranakis, Danny Krizanc, and Yannis C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Struct. Algorithms*, 12(3):253–269, 1998.
- Alexis C. Kaporis, Lefteris M. Kirousis, and Efthimios G. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. In *10th Annual European Symposium on Algorithms*, 2002.
- Alexis C. Kaporis, Lefteris M. Kirousis, and Efthimios Lalas. Selecting complementary pairs of literals. *Electronic Notes in Discrete Mathematics*, 16, 2003.
- H. Kautz, Y. Ruan, D. Achlioptas, C.P. Gomes, B. Selman, and M. Stickel. Balance and filtering in structured satisfiable problems. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.
- D. Le Berre and L Simon. 2004 sat competitions. <http://www.lri.fr/simon/contest/results/>, 2004.
- Chu-Min Li. Satz. <http://www.laria.u-picardie.fr/cli/EnglishPage.html>.
- D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of sat problems. In *Proc. 10-th National Conf. on Artificial Intelligence*, pages 459–465, 1992.
- Chvatal V. and Reed B. Mick gets some (the odds are on his side). *Proceedings of 33rd FOCS*, pages 620–627, 1992.
- N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5(4):1217–1235, 1995.

Appendix

A. Proof of Lemma 1 and Lemma 2

In this section we prove that the probability that a configuration formula F , with parameter as in lemma 1, is simple goes to a constant as the number of variables goes to infinity.

Denote $D_i = d_i + d_{-i}$, the number of occurrences of the variable i in the formula F . Let $\lambda = \sum_{i=1}^n D_i(D_i - 1)/n$. Note that if x_l is the fraction of variables with degree (number of occurrences) l then $\lambda = \sum_{l=2}^{\Delta} l(l-1)x_l$. To be able to take the limit we are assuming when n goes to infinity the densities x_l $0 \leq l \leq \Delta$ are fixed. We prove the following result.

$$\Pr(F \text{ is SIMPLE}) \rightarrow e^{-\Lambda} > 0 \text{ as } n \rightarrow \infty \quad (6)$$

where $\Lambda = 2\lambda \frac{a+b}{(2a+3b)^2} (1 + \frac{3}{2a+3b})$.

Let us call a *cycle* a non legal clause in F . Let Z be the number cycles in F . A configuration formula F is simple if and only if $Z = 0$.

Our proof follows a standard procedure (see (JLR00) chapter 9, (Bol01) chapter 2). We are going to prove that the distribution of Z converges to a Poisson distribution with mean Λ . Then equation (6) follows, because the right hand site is just the probability of the event $Z = 0$.

To prove that Z converges in distribution to a Poisson we use the method of moments (see theorem 6.10 in (JLR00)). Let $(Z)_k$ be the number of ordered k disjoint cycles in

F. We are going to prove that for every $k \geq 1$

$$\mathbf{E}(Z)_k \rightarrow \Lambda^k \text{ as } n \rightarrow \infty$$

to conclude that the distribution of Z approaches the distribution of a Poisson random variable with mean Λ .

Let $\beta(m_2, m_3)$ be the number of configurations with m_2 2-clauses and m_3 3-clauses. Note that $\beta(m_2, m_3) = \frac{(2m_2+3m_3)!}{2^{m_2}6^{m_3}(m_2+m_3)!}$. The probability $q_{1,0}$ that a particular 2-clause is present in a configuration is $q_{1,0} = \frac{\beta(m_2-1, m_3)}{\beta(m_2, m_3)}$. Moreover, the probability $q_{k,j}$ that any disjoint k 2-clause and j 3-clauses are present in a configuration is $q_{k,j} = \frac{\beta(m_2-k, m_3-j)}{\beta(m_2, m_3)}$. For fixed, k and j one can check that ²

$$q_{k,j} \sim \frac{2^k 6^j}{n^{k+2j} (2a+3b)^{2k+3j}} (a+b)^{k+j}$$

To compute $\mathbf{E}(Z)$, note that we can divide cycles into two groups: the ones coming from clauses with two different variables (for example $x \vee \bar{x} \vee y$, $x \vee x \vee y$, $x \vee x$), or the ones involving one variable ($x \vee \bar{x} \vee x$). We prove that the last ones have a very small probability and they do not contribute to our calculation. Let $Z = Y' + Y''$, where Y' are the number of cycles in the first group and Y'' the ones in the second.

Now we have to count how many clauses form cycles. Consider the first case. Here we have the 2-clause cycles and 3-clause cycles. The number of cycles coming from 2-clauses is

$$a_{1,0} = \sum_{i=1}^n D_i(D_i - 1) = \lambda n$$

The one coming from 3-clauses are

$$a_{0,1} = \sum_{i=1}^n D_i(D_i - 1)(n - D_i) \sim \lambda n^2$$

Therefore $\mathbf{E}(Y') = a_{1,0}q_{1,0} + a_{0,1}q_{0,1} \sim \Lambda$. Note now that if we count how many edges give us cycles of the second type, there are just $\sum_{i=1}^n D_i(D_i - 1)(D_i - 2) < \Delta^3 n$, as $q_{0,1}n$ goes to 0 then $\mathbf{E}(Y'') \rightarrow 0$ as $n \rightarrow \infty$.

Next, we compute $\mathbf{E}((Z)_s)$. Recall the $(Z)_s$ is the number of ordered s disjoint cycles. This time divide $(Z)_s$ into two sets: cycles in Y' not only are cycles on the first kind described before but also each cycle corresponds to different variable, cycles in Y'' have either a cycle of the second type or two or more cycles correspond to the same variable. Similar to what we have done before one can prove that $\mathbf{E}(Y'') \rightarrow 0$ as $n \rightarrow \infty$.

We can write

$$\mathbf{E}(Y') = \sum_{k+j=s} q_{k,j} a_{k,j}$$

where $a_{k,j}$ is the number of s disjoint cycles with k 2-clause cycles and j 3-clause cycles.

² We say that $a(n) \sim b(n)$ when $n \rightarrow \infty$ if $\lim_{n \rightarrow \infty} \frac{a(n)}{b(n)} = 1$.

Note that $a_{k,j}$ has the following expression (if we approximate as before $n - D_i$ by n)

$$a_{k,j} = \binom{k+j}{j} n^j \sum_{i_1, i_2, \dots, i_{k+j}} D_{i_1}(D_{i_1} - 1) \cdots D_{i_{k+j}}(D_{i_{k+j}} - 1) \quad (7)$$

where the sum over $\{i_1, i_2, \dots, i_{k+j}\}$ is over the set of disjoint indexes.

Claim: $a_{k,j} \sim \binom{k+j}{j} n^{2j+k}$.

Note that the difference between the above expression and (7) are the terms in which the same index is repeated but those terms are going to 0 when we divide by n^{k+j} .

Now note that

$$q_{k,j} a_{k,j} \sim \left(\frac{2\lambda(a+b)}{(2a+3b)^2}\right)^{k+j} \left(\frac{3}{2a+3b}\right)^j \binom{k+j}{j}$$

So we get $\mathbf{E}(Y') \sim \Lambda^k$. This completes the proof.

Proof of Lemma 2

The proof uses some of the ideas explained before. Let D be the number of occurrences of variable y in F and Z be the number of clauses with 2 occurrences of the variable y . As is the previous proof,

$$\mathbf{E}(Z) = D(D-1)q_{1,0} + D(D-1)(n-D)q_{0,1} \leq \frac{D^2}{n} \frac{2(a+b)}{(2a+3b)^2} \left[1 + \frac{3}{2a+3b}\right].$$

$P(Z > 1) \leq E(Z) \leq C/n$ the lemma follows, where C is given in the expression above.

Remark: The constant C can be bound uniformly for all the configuration formulas consider though the analysis. Note that we set variables until time t^* as explained in the prove of the section 4, the remaining clauses at that time is of order order n so $2a + 3b = c > 0$ for some constant c . Therefore we have that at any state of the algorithm $D^2 \frac{2(a+b)}{(2a+3b)^2} \left[1 + \frac{3}{2a+3b}\right] < \Delta^2 \frac{\alpha}{c^2} (1 + 3/c)$.

B. Differential equations

In this section we discuss the main theorem used in section 4 to analyze the random process.

We consider here a sequence of random process $Y_t = Y_t(n), n = 1, 2, \dots$, for simplicity the dependence on n is dropped from the notation. Let \mathcal{F}_t be the the σ -field generated by the process up to time t , *i.e.*, $\mathcal{F}_t = \sigma(Y_0, Y_1, \dots, Y_t)$. Our process $Y_t = (Y_t^{(1)}, \dots, Y_t^{(j)})$ is a vector of dimension j , let $\|Y\| = \max(|Y^{(1)}|, \dots, |Y^{(j)}|)$. Suppose that $Y_0 = z_0 n$ the value of the process at time 0.

We say that $X = o(f(n))$ *always* if $\max\{x : \Pr(X = x) \neq 0\} = o(f(n))$. The term *uniformly* means that the convergence implicit in the $o()$ is uniform on t .

Theorem 5. (Wor95) Let $f : \mathfrak{R}^{j+1} \rightarrow \mathfrak{R}^j$. Suppose there exist a constant C such that the process Y_t is bounded by Cn , i.e., $\|Y_t\| < Cn$. Suppose also that for some function $m = m(n)$:

(i) for all $t < m$ and all l

$$\Pr(\|Y_{t+1} - Y_t\| > n^{1/5} \mid \mathcal{F}_t) = o(n^{-3})$$

always;

(ii) for all l and uniformly over all $t < m$,

$$\mathbf{E}(Y_{t+1} - Y_t \mid \mathcal{F}_t) = f(t/n, Y_t/n) + o(1)$$

always;

(iii) The function f is continuous and satisfies a Lipschitz condition on D , where D is some bounded open set containing $(0, z_0^{(1)}, \dots, z_0^{(j)})$.

then:

(a) The system of differential equations

$$\frac{dz}{ds} = f(s, z)$$

has a unique solution in D for $z : \mathfrak{R} \rightarrow \mathfrak{R}^j$ with initial conditions $z(0) = z_0$ and which extends to points arbitrarily closed to the boundary of D .

(b)

$$Y_t = nz(t/n) + o(n) \text{ w.h.p.}$$

uniformly for $0 \leq t \leq \min\{\sigma n, m\}$, where σ is the supremum of those s to which the solution can be extended.

B.1. DIFFERENTIAL EQUATIONS FOR THE LOWER BOUND

Let l scaled number of current unset literals, c_3 , c_2 scaled number of 3-clauses and 2-clauses respectively and x_s scaled number of literals of degree s , $s = 1, \dots, 4$. The equations for round j are:

$$\begin{aligned} \frac{dl}{dt} &= -2 - 4 \frac{c_2}{l - 2c_2} \\ \frac{dc_3}{dt} &= -\frac{3jc_3}{p} - \frac{3c_3}{l} + \left(-\frac{3jc_3}{p} - \frac{3c_3}{l}\right) \frac{2c_2}{l - 2c_2} \\ \frac{dc_2}{dt} &= \frac{3c_3 - 2c_2}{l} - \frac{2jc_2}{p} + \left(\frac{3c_3 - 2c_2}{l} - \frac{2jc_2}{p}\right) \frac{2c_2}{l - 2c_2} \\ \frac{dx_4}{dt} &= -(6c_3 + 2c_2) \frac{4x_4}{p^2} j - \frac{x_4}{l} - \delta_{4,j} \\ &\quad - \left((6c_3 + 2c_2) \frac{4x_4}{p^2} j + \frac{x_4}{l} + \frac{4x_4}{p}\right) \frac{2c_2}{l - 2c_2} \\ \frac{dx_s}{dt} &= (6c_3 + 2c_2) \frac{(s+1)x_{s+1} - (s)x_s}{p^2} d^j - \frac{x_s}{l} - \delta_{s,j} \end{aligned}$$

$$((6c_3 + 2c_2) \frac{(s+1)x_{s+1} - (s)x_s}{p^2} d^j - \frac{x_s}{l} - \frac{sx_s}{p}) \frac{2c_2}{l-2c_2}$$

for $s = 1, 2, 3$

with initial conditions $l = 2, c_3 = c, c_2 = 0, x_4 = 2p, x_3 = 2(1-p), x_2 = 0, x_1 = 0$.