International Journal of Innovative Computing, Information and Control
Volume **12**, Number **2**, April **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 477–489

UPPER BOUND ON THE SATISFIABILITY THRESHOLD OF REGULAR RANDOM (k, s)-SAT PROBLEM

Jincheng Zhou[1,2] and Daoyun Xu[1,*]

[1]College of Computer Science and Technology
Guizhou University
Xibei Road, Huaxi District, Guiyang 550025, P. R. China
guideaaa@126.com; *Corresponding author: dyxu@gzu.edu.cn

[2]School of Mathematics and Statistics
Qiannan Normal College for Nationalities
Longshan Road, Kaifa District, Duyun 558000, P. R. China

Received July 2015; revised November 2015

Abstract. *We consider a strictly regular random (k, s)-SAT problem and propose a GSRR model for generating its instances. By applying the first moment method and the asymptotic approximation of the γth coefficient for generating function $f(z)^\lambda$, where λ and γ are growing at a fixed rate, we obtain a new upper bound $2^k \log 2 - (k+1)\log 2/2 + \epsilon_k$ for this problem, which is below the best current known upper bound $2^k \log 2 + \epsilon_k$. Furthermore, it is also below the asymptotic bound of the uniform k-SAT problem, which is known as $2^k \log 2 - (\log 2 + 1)/2 + o_k(1)$ for large k. Thus, it illustrates that the strictly regular random (k, s)-SAT instances are computationally harder than the uniform one in general and it coincides with the experimental observations. Experiment results also indicate that the threshold for strictly regular random (k, s)-SAT problem is very close to our theoretical upper bound, and the regular random (k, s)-SAT instances generated by model GSRR are far more difficult to solve than the uniform one in each threshold point.*
Keywords: Strictly regular random (k, s)-SAT problem, Hard instances generation model, Upper bound, Phase transition, Asymptotic approximation

1. **Introduction.** Numerous computational problems encountered in science and industry can be viewed as Constraint Satisfaction Problems (CSPs), which have been intensively studied in theoretical computer science and combinatorics. In general, CSP tasks are computationally intractable [1]. A particular problem in the class of CSPs is the so-called Satisfiability (SAT) problem. Given a Boolean formula $\mathcal{F}$ in conjunctive normal form (CNF), the SAT problem consists in answering the question whether an assignment of Boolean values to the variables exists, such that the formula $\mathcal{F}$ evaluates to true. When $\mathcal{F}$ has exactly k literals in each clause, it is known as k-SAT problem and it was the first constraint satisfaction problem shown to be $\mathcal{NP}$-complete for $k \geq 3$ by Cook in [2]. This $\mathcal{NP}$-completeness property entails that every problem from the complexity class $\mathcal{NP}$ can be efficiently transformed into a SAT problem. Therefore, SAT is a fundamental problem in combinatorial discrete optimization and it is the root problem in complexity theory [3].

Cook's work, as well as most of the work on CSPs that followed, focuses on the worst-case complexity of the problems; however, experimental studies illustrate that many instances in SAT problem are invariably surprisingly easy, even for naive heuristic algorithms. Thus, the introduction of new methods for generating random hard instances in SAT problem is crucial both for understanding the complexity of the SAT problem and for providing challenging benchmarks for experimental evaluation of algorithms [4]. Particularly, a clear connection has been established between so-called phase transition phenomena and the computational hardness of $\mathcal{NP}$-complete problems [5-7]. A great amount of experimental and theoretical studies indicate that a phase transition in solvability is a very paramount feature to many decision problems. More interestingly, the hardest instances are concentrated at the sharp transition region. Moreover, it is widely believed that the ensemble of random k-SAT problem with N variables where each formula is generated by randomly choosing $M = \alpha N$ clauses of k literals has the phase transition phenomenon, and it has been the focus of intensive theoretical studies by computer scientists and statistical physicists in the last twenty years [8-17]. Specifically, as the constraint density α (the ration of clauses to variables) increases, the number of satisfying assignments decreases. More precisely, in the limits of $N \to \infty$, the system is known to have a sharp threshold in constraint density $\alpha_s(k)$, for $\alpha < \alpha_s(k)$ the probability that a randomly generated k-SAT instance is satisfiable goes to 1 and for $\alpha > \alpha_s(k)$ it vanishes [18]. This phenomenon is particularly interesting because it turns out the really difficult instances, from the algorithmic point of view, are those where α is close to $\alpha_s(k)$.

Using discrete Fourier analysis, a slightly weaker statement was proved by Friedgut and Bourgain in [12]. They showed that there exists a sharp threshold sequence $\alpha_k(N)$ in the random k-SAT problem such that when the number of clauses M is around $\alpha_k(N)N$, the probability of the formula having a satisfying assignment drops abruptly from near 1 to near 0 as $N \to \infty$. Friedgut and Bourgain demonstrated that there is a function $\alpha_k(N)$ for which it is true, but we still cannot obtain the exact location of the phase transition point from this approach. There is a multitude of work devoted to the study of the exact threshold where the formula becomes unsatisfiable; however, except in the case of $k = 2$ [19-21], the exact threshold of the random k-SAT problem is currently unknown.

Recently, using a heuristic method called the 'one step replica symmetry breaking' (1RSB) cavity method [9, 10, 15], the threshold has been conjectured to be $2^k \log 2 - (\log 2 + 1)/2 + o_k(1)$ for large k and $\alpha_s \simeq 4.2667$ for $k = 3$ [10]. In addition, it has been very recently proved in [16, 17] that for k is large enough, the SAT-UNSAT threshold $\alpha_s(k)$ exists and the threshold coincides with the prediction from the cavity method in [10]. A widely accepted conjecture is that the SAT-UNSAT threshold $\alpha_s(k)$ exists for any value of k; however, it seems to be very difficult to obtain the exact location of the transition point for the random k-SAT problem. Thus, several k-SAT variations problems by restricting the formula structure have been considered, and the critical thresholds have been obtained for some of these problems, such as 2-SAT problem [19-21], k-NAESAT problem [22], Regular 2-SAT problem [23] and Regular k-NAE-SAT problem [24].

Moreover, experimental results state that the balanced instances of random combinatorial problems are often much more difficult to solve than the uniformly random instances. Thus, a regular random k-SAT problem, in which each literal occurs approximately the same number of times in the formula clauses was proposed in [23] for the first time, where the authors experimentally observed that the regular random 3-SAT formulas are computationally harder than the uniform random 3-SAT instances. In [23], the authors also derived the sharp threshold for the regular random 2-SAT problem and an upper bound threshold where $\alpha < 3.78$ for the strictly regular random 3-SAT problem by using the standard Lagrange maximization method. Then they analyzed a greedy algorithm on regular random 3-SAT formulas and showed that for $\alpha < 2.46$ the algorithm can find a satisfying assignment with positive probability. Furthermore, a strictly regular random (k, s)-SAT problem, in which each literal occurs either $\lfloor s/2 \rfloor$ times or $\lfloor s/2 \rfloor + 1$ in the formula clauses was proposed in [25]. Based on counting the number of satisfying assignments and the saddle point method to the approximation of generating function coefficients, the authors derived upper and lower bounds for the strictly regular random (k, s)-SAT formulas, which are $2^k \log 2 - (k+1)\log 2/2 - 1 - \delta_k \leq \alpha^*_{reg} \leq 2^k \log 2 + \epsilon_k$ for $k \geq 3$, where δ_k and ϵ_k

hide a term that tends to 0 in the limit of large k. Bapst and Coja-Oghlan [26] proposed a non-rigorous approach from physics for harnessing Belief Propagation, and obtained a rigorous proof for the existence and location of a condensation phase transition in the strictly regular random (k,s)-SAT problem.

2. **Technical Definitions.**

2.1. **Basic notations.** A CNF formula over the variables $v_1,\ldots,v_N$ is a conjunction of clauses $\mathcal{C}_1 \wedge \mathcal{C}_2 \wedge \ldots \wedge \mathcal{C}_M$ where each clause $\mathcal{C}_i$ is a disjunction of literals $\ell_1 \vee \ell_2 \vee \ldots \vee \ell_k$. Each literal ℓ_i is either a variable v_i or its negation $\neg v_i$. A formula is said to be in k-CNF form if every clause contains exactly k literals. A CNF formula is satisfiable if there is a Boolean assignment $\sigma = \{0,1\}^N$ to the variables $v_1,\ldots,v_N$, such that every clause contains at least one literal which evaluates to true. A random k-CNF formula consists of M clauses chosen uniformly at random from the set of all C_N^k possible ones. A regular k-SAT formula [23] is denoted on N Boolean variables and M clauses, in which each of the $2N$ literals $\{v_1,\neg v_1,\ldots,v_N,\neg v_N\}$ occurs approximately the same number of times and each clause has exactly k distinct literals.

Suppose each literal occurs precisely r ($r \in Z^+$) times, i.e., each variable occurs precisely $s = 2r$ times, then in any regular k-SAT formula, it implies that $2Nr = kM$ and $\alpha = 2r/k = s/k$; thus, s must be an even number. To circumvent this, a strictly regular random (k,s)-SAT problem was introduced in [25]. In this problem, the authors allowed each literal to take two possible occurrence times. Specifically, for a given s ($s \in Z^+$), let $r = kM/(2N)$, if s is an even number, then each literal occurs precisely $\lfloor r \rfloor = s/2$ times; else each literal occurs either $\lfloor r \rfloor$ or $\lfloor r \rfloor + 1$ times. Thus, a strictly regular random (k,s)-SAT formula is denoted on N Boolean variables and M clauses, in which each of the N variables $\{v_1,\ldots,v_N\}$ occurs precisely s times, each of the $2N$ literals $\{v_1,\neg v_1,\ldots,v_N,\neg v_N\}$ occurs either $\lfloor s/2 \rfloor$ times or $\lfloor s/2 \rfloor + 1$ times, chosen at random among all such formulas with uniform probability.

A strictly regular (k,s)-SAT formula $\mathcal{F}$ can be represented as a (k,s)-regular bipartite graph [27] $I(\mathcal{F})$. The incidence graph $I(\mathcal{F})$ is defined as follows: $\mathcal{V}_1(I(\mathcal{F}))$ (circles in the graphical representation) consists of the variables $v_1,\ldots,v_N$ of $\mathcal{F}$ and $\mathcal{V}_2(I(\mathcal{F}))$ (squares in the graphical representation) consists of the clauses $\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_M$ of $\mathcal{F}$, a variable v and a clause $\mathcal{C}$ are adjacent if and only if v occurs (positively or negatively) in $\mathcal{C}$. In general, we use a full line between v and $\mathcal{C}$ whenever the variable v appearing in the clause is v, a dashed line whenever the variable v appearing in the clause is $\neg v$. As an example, Figure 1 exhibits the bipartite graph representation of the strictly regular (3,6)-SAT formula $\mathcal{F} = (v_1 \vee \neg v_2 \vee v_3) \wedge (\neg v_1 \vee \neg v_2 \vee \neg v_3) \wedge (v_1 \vee v_2 \vee \neg v_3) \wedge (\neg v_1 \vee \neg v_2 \vee v_3) \wedge (v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee \neg v_3)$.

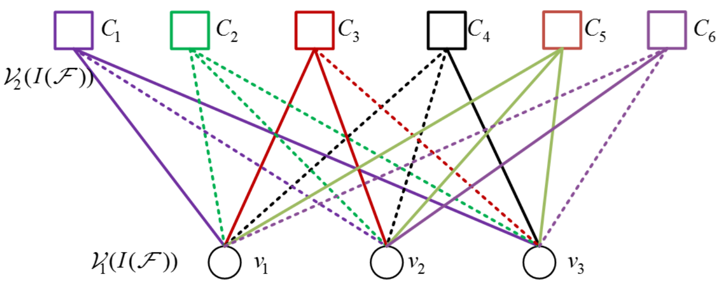

FIGURE 1. Bipartite graph representation of the strictly regular $(3,6)$-SAT formula $\mathcal{F}$, in which each of the 6 literals $\{v_1,\neg v_1,v_2,\neg v_2,v_3,\neg v_3\}$ occurs precisely 3 times and each clause has exactly 3 distinct literals

To generate a regular random (k,s)-SAT formula, Rathi et al. [25] assigned the labels from a set E where $E = \{1, 2, \ldots, sN\}$ to edges on both sides of the bipartite graph, and then generated a random permutation Π on E, connected each edge i on the variable node side to each edge $\Pi(i)$ on the clause node side. Thus, they got a regular random (k,s)-SAT formula $\mathcal{F}$. However, it was shown in [23] that the number of illegal or repeated clauses is $o(N)$ with high probability by this method.

2.2. **Generating strictly regular random (k,s)-SAT instances.** In order to avoid the appearances of the illegal or repeated clauses, here we propose a new type of instances generating model for the strictly regular random (k,s)-SAT problem, called model GSRR. This model contains three control parameters, the clause length k, the variable size N and the same occurrence times s for every variable. We denote r_{i_1} as the number of occurrences for variable v_i positively and r_{i_2} as the number of occurrences for variable v_i negatively in a formula. The generation of an instance $\mathcal{F}_k(N, \alpha_{reg}N)$ for the strictly regular random (k,s)-SAT problem in model GSRR is done in the following five steps:

- **Step *1*** Set $j := 0$, $\mathcal{C} := \mathbf{\Phi}$.
- **Step *2*** For each variable v_i, $i \in \{1, 2, \ldots, N\}$, if s is an even number, then set $r_{i_1} = r_{i_2} := s/2$; else set $r_{i_1} := \lfloor s/2 \rfloor$ or $r_{i_1} := \lfloor s/2 \rfloor + 1$ with equal probability, and set $r_{i_2} := s - r_{i_1}$.
- **Step *3*** Put r_{i_1} copies of variable v_i and r_{i_2} copies of its negation $\neg v_i$ into box A.
- **Step *4*** Randomly selected k literatures $\ell_{j_1}, \ell_{j_2}, \ldots, \ell_{j_k}$ from box A:

  (1) If these k literatures $\ell_{j_1}, \ell_{j_2}, \ldots, \ell_{j_k}$ can constitute a correct clause (without repetition for the corresponding variables of the k literatures and without repetition clauses in clauses $\mathcal{C}_q$ where $q = 1, 2, \ldots, j$), then we set $\mathcal{C}_j = \{\ell_{j_1} \vee \ell_{j_2} \vee \ldots \vee \ell_{j_k}\}$ and connect the clause $\mathcal{C}_j$ into $\mathcal{C}$ with conjunction norm form; else put back these k literatures $\ell_{j_1}, \ell_{j_2}, \ldots, \ell_{j_k}$ into the box A, and go to Step 4.

  (2) Set $A := A \setminus \{\ell_{j_1}, \ell_{j_2}, \ldots, \ell_{j_k}\}$, $j = j + 1$.
- **Step *5*** If $j < Ns/k$, then go to Step 4; else output formula $\mathcal{C}$ and we stop.

Hence, it is easy to see, in each strictly regular random formula $\mathcal{F}_k(N, \alpha_{reg}N)$ generated by model GSRR, each clause has exactly k different literatures, each variable occurs exactly s times and each literal occurs either $\lfloor s/2 \rfloor$ times or $\lfloor s/2 \rfloor + 1$ times.

3. **Main Results.** In this paper, we propose a GSRR model to generate the strictly regular random (k,s)-SAT formulas. By applying the first moment method and the asymptotic approximation of the coefficient of order γ for a generating function $f(z)^\lambda$, where λ and γ are growing at a fixed rate, we derive a new upper bound on the satisfiability threshold for the strictly regular random (k,s)-SAT formulas for $k \geq 3$. We show that our upper bound is $2^k \log 2 - (k+1)\log 2/2 + \epsilon_k$, which is below the current best known upper bound $2^k \log 2 + \epsilon_k$ in [25]. Our new upper bound is also below the asymptotic threshold of the uniform k-SAT model obtained very recently in [16, 17], which is $2^k \log 2 - (\log 2 + 1)/2 + o_k(1)$ for large k. Thus, we give a theoretical explanation why the regular random (k,s)-SAT formulas instances are computationally harder than the uniform k-SAT instances in general, which coincides with the experimentally observed in [23] and our paper. Together with the lower bound of [25], we establish the following sharp satisfiability threshold in Theorem 3.1. Therefore, we just left an additive gap of a constant 1 in the regular random (k,s)-SAT problem.

Theorem 3.1. *For each $k \geq 3$, the satisfiability threshold of strictly regular random (k,s)-SAT formula satisfies*

$$2^k \log 2 - (k+1)\log 2/2 - 1 - \delta_k \leq \alpha^*_{reg} \leq 2^k \log 2 - (k+1)\log 2/2 + \epsilon_k, \tag{1}$$

where ϵ_k hides a term that tends to 0 in the limit of large k.

4. Upper Bound on Threshold.

4.1. Asymptotic approximation of coefficients.

The generating functions with the type $G(z) = f(z)^\lambda$ often appear in several combinatorial enumeration problems, where $f(z)$ is a given function with positive coefficients and λ is a parameter that tends to infinity. We often need to estimate the γth coefficient of $f(z)^\lambda$, which we denote by $[z^\gamma]\{f(z)^\lambda\}$ for large γ and λ.

First, we define two operators on a generating function $f(z)$ as the following.

$$\Delta f(z) = z\frac{d}{dz}\log(f(z)) = z\frac{f'(z)}{f(z)};\ \delta f(z) = \frac{f''(z)}{f(z)} - \frac{f'(z)^2}{f(z)^2} + \frac{f'(z)}{zf(z)}. \tag{2}$$

For any analytic function G with positive coefficients, its coefficient of order γ can be given by the Cauchy's formula, where the integration contour is a closed curve around the origin of the complex plane, inside the domain of its convergence, note that

$$[z^\gamma]\{G(z)\} = \frac{1}{2\pi i}\oint G(z)\frac{dz}{z^{\gamma+1}}. \tag{3}$$

Thus, we get an upper bound $|[z^\gamma]\{G(z)\}| \leq \frac{1}{2\pi}\oint |G(z)\frac{1}{z^{\gamma+1}}|dz$ from (3). Therefore, while integrating on a circle of radius ρ, which is smaller than the radius of convergence of G, we have

$$[z^\gamma]\{G(z)\} \leq G(\rho)\rho^{-\gamma}, \tag{4}$$

and when ρ satisfies $\rho G'(\rho)/G(\rho) = \gamma$, the smallest upper bound is obtained. Asusume that X is a random varible for the generating function G. Setting $\rho = \exp(t)$ and using the fact that $G(\exp(t)) = \mathbb{E}[\exp(tX)]$, by Chernoff's bound, we have

$$\Pr(X = (1+\delta)\mu) \leq \frac{\mathbb{E}[\exp(tX)]}{\exp(t(1+\delta)\mu)}. \tag{5}$$

In addition, if we assume that the random variable X is obtained by summing λ independent random variables with distribution $G(z) = f(z)^\lambda$, then we have $\Pr(X = \gamma) \leq f(\rho)^\lambda\rho^{-\gamma}$, and the upper bound (4) can be refined to give an approximation of $[z^\gamma]\{G(z)\}$.

Instead of bounding G on the integration circle, Gardy [28] looked closely at the points that give the main contribution to the integral. He showed that it is the basis of the saddle point method for applications to the approximation of generating function coefficients. Therefore, if we can choose the point ρ defined by the equation $\rho G'(\rho)/G(\rho) = \gamma$ for radius of the integration circle, the majority of the integral often comes from the proximity of ρ. Indeed, it is a saddle point. Define $h(z) = \log(G(z)) - (\gamma+1)\log(z)$, then

$$[z^\gamma]\{G(z)\} = \frac{1}{2i\pi}\oint e^{h(z)}dz = \frac{e^{h(\rho)}}{\sqrt{2\pi h''(\rho)}}(1+o(1)) = \frac{G(\rho)}{\rho^{\gamma+1}\sqrt{2\pi h''(\rho)}}(1+o(1)). \tag{6}$$

For a larger class of functions this result is actually valid, such as functions defined on an open disk or entire functions.

In this paper, we consider the asymptotic approximation of $[z^\gamma]\{f(z)^\lambda\}$ for large γ and λ growing at a fixed rate. Moreover, it can be improved to give further terms of an asymptotic development. We give the main result in [29] as follows:

Theorem 4.1. *Let $f(z) = f_0 + f_1x + f_2x^2 + \dots$ be a generating function which has real positive coefficients with $f_0 \neq 0$, $f_1 \neq 0$, and its radius of convergence R is strictly positive. Assume that γ/λ belongs to an interval $[a, b]$, $0 < a < b$, and $\gamma, \lambda \to +\infty$. Define ρ and δ^2 by $\Delta f(\rho) = \gamma/\lambda$ and $\sigma^2 = \rho^2 \delta f(\rho)$. If $\rho < R$, then*

$$[z^\gamma]\{f(z)^\lambda\} = \frac{f(\rho)^\lambda}{\sigma \rho^\gamma \sqrt{2\pi\lambda}}(1 + o(1)). \tag{7}$$

Thus, note that for suitable constants $A = f(\rho_0){\rho_0}^{-\kappa}$, $B = \sigma\sqrt{2\pi}$, and with ρ_0 being the solution (independent of γ and λ) of $\Delta f(z) = \kappa$. Indeed, σ is a constant as $\sigma^2 = {\rho_0}^2 \delta f(\rho_0)$. If $\gamma = \kappa\lambda$, we can get the following corollary immediately.

Corollary 4.1. *Under the assumptions of Theorem 4.1, if there exists a strictly positive real constant κ such that $\gamma = \kappa\lambda$, then we have*

$$[z^\gamma]\{f(z)^\lambda\} = \frac{A^\lambda}{B\sqrt{\lambda}}(1 + o(1)). \tag{8}$$

4.2. **Upper bound on threshold via first moment.** Let $\mathcal{Z}$ be a non-negative integer-valued random variable with expected value of $\mathbb{E}[\mathcal{Z}]$. Using one of the most popular techniques in the probabilistic method, namely the first moment method, we have $\Pr(\mathcal{Z} \geq 1) \leq \mathbb{E}[\mathcal{Z}]$. The implementation of the first moment method makes use of Markov's inequality. Consequently, by estimating the expected number of solutions we can obtain an upper bound on the threshold beyond which no solution exists with high probability.

Let $\mathcal{N}(k)$ be the number of satisfying assignments for a randomly generated strictly regular formula $\mathcal{F}_k(N, \alpha_{reg}N)$ by model GSRR. For any assignment $\xi \in \{0,1\}^N$ to the variables $v_1, v_2, \dots, v_N$ for formula $\mathcal{F}_k(N, \alpha_{reg}N)$, let $\mathcal{D}$ denote the event that an assignment ξ satisfies $\mathcal{F}_k(N, \alpha_{reg}N)$, let $\mathcal{H}$ denote the event that the assignment $\sigma = \{1, 1, \dots, 1\}$ satisfies $\mathcal{F}_k(N, \alpha_{reg}N)$, and let $\mathcal{I}_\xi$ be an indicator variable that ξ is a satisfying assignment for formula $\mathcal{F}_k(N, \alpha_{reg}N)$. Due to the symmetry of the strictly regular random (k, s)-SAT formula generation by model GSSR, the occurrence of each literal has the same distribution. That is to say, any assignment of variables has the same probability of being a solution. Thus, it implies that the probability

$$\Pr[\mathcal{D}] = \Pr[\mathcal{H}]. \tag{9}$$

Therefore, we obtain

$$\mathbb{E}[\mathcal{N}(k)] \; = \; \sum\nolimits_{\xi \in \{0,1\}^N} \mathbb{E}[\mathcal{I}_\xi] = 2^N \times \Pr[\mathcal{H}]. \tag{10}$$

In any strictly regular random formula $\mathcal{F}_k(N, \alpha_{reg}N)$, there are rN positive literals and the same amount of negative literals from the clauses. Thus, the total number of formulas is $(2rN)!$. Note that the total numbers of positive literals and negative literals are equal, so all the permuting among positive literals and negative literals is $(rN)! \times (rN)!$. Thus, the total number of formulas for which $\{1, 1, \dots, 1\}$ is a solution is given by

$$(rN)! \times (rN)! \times \text{the total ways of satisfying } M \text{ clauses.} \tag{11}$$

We denote $g(z)$ to be the generating function to satisfy a clause. It corresponds to placing at least one positive literal in a clause. Hence, we have,

$$g(z) = C_k^1 z^1 + C_k^2 z^2 + \dots + C_k^k z^k = (1 + z)^k - 1, \; z \in (0, 1). \tag{12}$$

Therefore, the generating function to satisfy M clauses is $g(z)^M$, the total way of satisfying M clauses is $\left[z^{rN}\right]\left\{g(z)^M\right\}$, where $\left[z^{rN}\right]\left\{g(z)^M\right\}$ denotes the coefficient of z^{rN}

in the expansion of $g(z)^M$. Thus, by (10), (11) and (12), we have

$$\mathbb{E}\left[\mathcal{N}(k)\right] = 2^N \times \frac{((rN)!)^2}{(2rN)!} \times \left([z^{rN}]\left\{g(z)^M\right\}\right). \tag{13}$$

We now use the asymptotic approximate of the coefficients in Section 4.1 to solve $\left[z^{rN}\right]\left\{g(z)^M\right\}$, and then compute the expectation of the total number of solutions. By Theorem 4.1, the generating function should have real positive coefficients with $f_0 \neq 0$ and $f_1 \neq 0$. Note that if we let

$$f(z) = \frac{g(z)}{z} = \frac{(1+z)^k - 1}{z}, \tag{14}$$

then Function $f(z)$ satisfies the assumptions of Theorem 4.1. By (13), (14) and $M/N = 2s/k = \alpha_{reg}$, we have

$$\begin{aligned}\mathbb{E}[\mathcal{N}(k)] &= 2^N \times \frac{((rN)!)^2}{(2rN)!} \times \left(\left[z^{rN-M}\right]\left\{[g(z)/z]^M\right\}\right)\\ &= 2^N \times \frac{((rN)!)^2}{(2rN)!} \times \left(\left[z^{(k/2-1)\alpha_{reg}N}\right]\left\{f(z)^{\alpha_{reg}N}\right\}\right).\end{aligned} \tag{15}$$

Set $\gamma = (k/2-1)\,\alpha_{reg}N$, $\lambda = \alpha_{reg}N$, we have $\kappa = \gamma/\lambda = k/2-1$; thus the coefficient term coincides with the situation of Corollary 4.1 in (8). Thus, by the definition of the $\Delta f(z)$ and $\delta f(z)$ in (3), we have

$$\Delta f(z) = z\frac{f'(z)}{f(z)} = \frac{1 + kz(1+z)^{k-1} - (1+z)^k}{(1+z)^k - 1}, \tag{16}$$

$$\delta f(z) = \frac{\Delta f'(z)}{z} = \frac{k(1+z)^{k-2}\left[(1+z)^k - kz - 1\right]}{z\left[(1+z)^k - 1\right]^2}. \tag{17}$$

Set ρ_k to be the solution of the equation $\Delta f(z) = \gamma/\lambda$, i.e., ρ_k is the solution of the following equation

$$\frac{1 + kz(1+z)^{k-1} - (1+z)^k}{(1+z)^k - 1} = \frac{k}{2} - 1. \tag{18}$$

Simplifying the equation of (18), we obtain

$$(1+\rho_k)^{k-1}(1-\rho_k) - 1 = 0,\ \ \rho_k \in (0,1). \tag{19}$$

Lemma 4.1. *Let $\mathcal{N}(k)$ denote the total number of satisfying assignments of a strictly regular random (k,s)-SAT formula, then,*

$$\mathbb{E}\left[\mathcal{N}(k)\right] = 2^{N-k\alpha_{reg}N-1}\frac{\left[(1+\rho_k)^k - 1\right]^{\alpha_{reg}N+1}}{\sqrt{(1+\rho_k)^{k-2}\left[(1+\rho_k)^k - k\rho_k - 1\right]\rho_k{}^{k\alpha_{reg}N+1}}}(1+o(1)), \tag{20}$$

where ρ_k is the positive solution of (19).

Proof: According to the (8) in Corollary 4.1 and $\sigma = \rho_k \cdot \sqrt{\delta f(\rho_k)}$, we have

$$\begin{aligned}
\left[\rho_k^{\frac{(k-2)\alpha_{reg}N}{2}}\right]\left\{f(\rho_k)^{\alpha_{reg}N}\right\} &= \frac{[f(\rho_k)]^{\alpha_{reg}N}\rho_k^{-\frac{k-2}{2}\alpha_{reg}N}}{\sigma\sqrt{2\pi}\cdot\sqrt{\alpha_{reg}N}}(1+o(1)) \\
&= \frac{[f(\rho_k)]^{\alpha_{reg}N}}{\sigma\cdot\rho_k^{(k-2)\alpha_{reg}N/2}\sqrt{2\pi\alpha_{reg}N}}(1+o(1)) \\
&= \frac{[f(\rho_k)]^{\alpha_{reg}N}}{\rho_k\cdot\sqrt{\delta f(\rho_k)}\cdot\rho_k^{(k-2)\alpha_{reg}N/2}\sqrt{2\pi\alpha_{reg}N}}(1+o(1)) \\
&\sim \frac{\left((1+\rho_k)^k-1\right)^{\alpha_{reg}N+1}}{\sqrt{2\pi k\alpha_{reg}N}\sqrt{(1+\rho_k)^{k-2}\left[(1+\rho_k)^k-k\rho_k-1\right]\rho_k^{k\alpha_{reg}N+2}}}.
\end{aligned} \tag{21}$$

Now using the Stirling's approximation [30] of $N! \sim \sqrt{2N\pi}\left(\frac{N}{e}\right)^N$ when $N \to \infty$, it implies that

$$((rN)!)^2 \sim 2rN\pi\left(\frac{rN}{e}\right)^{2rN}, \quad (2rN)! \sim \sqrt{4rN\pi}\left(\frac{2rN}{e}\right)^{2rN}. \tag{22}$$

Finally, with $\alpha_{reg} = M/N = 2s/k$, using direct simplification, Lemma 4.1 holds from the equations of (22), (15) and (21). □

Next, we will derive the upper bound to the satisfiability threshold for the strictly regular random (k,s)-SAT formulas.

Lemma 4.2. *Let α^*_{reg} be the satisfiability threshold for the strictly regular random (k,s)-SAT formulas. Let α^u_{reg} be the upper bound on α^*_{reg} obtained by the first moment method. Then we have,*

$$\alpha^*_{reg} \le \alpha^u_{reg} = 2^k \log 2 - (k+1)\log 2/2 + \epsilon_k. \tag{23}$$

Proof: Note that ρ_k is the solution of Equation (19), together with (20), we have

$$\log\left(\mathbb{E}\left[\mathcal{N}(k)\right]\right) = \log\left[2^{N-k\alpha N-1}\frac{[(1+\rho_k)^k-1]^{\alpha_{reg}N+1}}{\sqrt{(1+\rho_k)^{k-2}[(1+\rho_k)^k-k\rho_k-1]\rho_k^{k\alpha_{reg}N+2}}}(1+o(1))\right].$$

Observe that

$$\lim_{N\to\infty}\frac{\log(\mathbb{E}\left[\mathcal{N}(k)\right])}{N} = (1-k\alpha_{reg})\log 2 + \alpha_{reg}\log((1+\rho_k)^k-1) - \frac{k\alpha_{reg}}{2}\log(\rho_k), \tag{24}$$

where ρ_k satisfies the equation of (19).

Therefore, via the first moment method, we know that if $\mathbb{E}\left[\mathcal{N}(k)\right] < 1$, a randomly generated strictly regular random formula $\mathcal{F}_k(N, \alpha_{reg}N)$ is unsatisfiable with high probability. This implies that if $\lim_{N\to\infty}(\log(\mathbb{E}\left[\mathcal{N}(k)\right])/N) < 0$, a randomly generated strictly regular random formula $\mathcal{F}_k(N, \alpha_{reg}N)$ is unsatisfiable with high probability. Thus, set $\lim_{N\to\infty}(\log(\mathbb{E}\left[\mathcal{N}(k)\right])/N) \ge 0$, and then we can get the upper bound of the satisfiability for the strictly regular random (k,s)-SAT problem from Equation (24). Furthermore, from Equation (24), using the approximate solution of $\rho_k \sim 1 - 1/2^k$ in (19) to substitute the equation of (24), we obtain the following upper bound on the satisfiability threshold of strictly regular random (k,s)-SAT problem, which is

$$\alpha^*_{reg} \le \alpha^u_{reg} = 2^k \log 2 - (k+1)\log 2/2 + \epsilon_k. \tag{25}$$

□

Consequently, by the result of Lemma 4.2 and the lower bound in [25], we have finished the proof of Theorem 3.1. □

5. **Numerical Analysis.** In order to illustrate the reliability of the approximate solution for the upper bound on the satisfiability threshold in (25) via $\rho_k \sim 1 - 1/2^k$, we calculate the numerical solutions of ρ_k in (19) for $3 \leq k \leq 18$, and get the numerical upper bounds of the α_{reg}^u from Equation (24). In Table 1, the upper bound $\alpha_{reg}^u = 2^k \log 2 - (k+1) \log 2/2$ is obtained by our method, the term $2^k \log 2 - \alpha_u^*$ means the gap between the upper bound in [25] and our result. We can observe that the numerical results fit the upper bound $2^k \log 2 - (k+1) \log 2/2$ well. Furthermore, Figure 2 shows the numerical analysis results fit the curve of our upper bound function $2^k \log 2 - (k+1) \log 2/2$ fairly well.

Table 1. The numerical analysis results about the upper bound on the satisfiability threshold of strictly regular random (k, s)-SAT problem

k	numerical result	$\alpha_{reg}^u -$numerical result	$2^k \log 2$ [25]	$2^k \log 2 - \alpha_{reg}^u$
3	3.78	0.38	5.558	1.39
4	9.11	0.25	11.09	1.73
5	19.93	0.17	22.18	2.08
6	41.83	0.11	44.36	2.43
7	85.88	0.07	88.72	2.77
8	174.28	0.05	177.45	3.12
9	351.40	0.03	354.89	3.47
10	705.95	0.02	709.78	3.81
11	1415.40	0.01	1419.57	4.16
12	2834.62	0.01	2839.13	4.51
13	5673.41	0.00	5678.26	4.85
14	11351.30	0.00	11356.50	5.20
15	22707.50	0.00	22713.00	5.55
16	45420.20	0.00	45426.10	5.90
17	90845.90	0.00	90852.20	6.30
18	181698.00	0.00	181704.37	6.37

6. **Experimental Results.** In this section, we present numerical experiments to demonstrate the correctness of our theoretical upper bound on the satisfiability threshold of the strictly regular random (k, s)-SAT problem, and verify that the random (k, s)-SAT instances generated by model GSRR are much more difficult to solve than the uniform random k-SAT instances generated by the uniform random k-SAT model in each phase transition region. Furthermore, as Zchaff algorithm [31] is currently the best complete algorithm for solving the SAT problem, in our experiment, we choose Zchaff algorithm to solve these two kinds of random k-SAT instances. To simplify the experiment, we chose $k = 3$.

(1) In strictly regular random $(3, s)$-SAT problem, for variables size $N = 60, 90, \ldots, 210$, firstly we generate 100 random instances for each $s \in \{6, 7, \ldots, 18\}$ by model GSSR (the total number of instances is $6 \times 13 \times 100 = 7800$). Then for each $N \in \{60, 90, \ldots, 210\}$, we compute each 100 random instances for each s by Zchaff algorithm, record the corresponding computation time and whether it is satisfiable for each formula.

(2) In general uniform random 3-SAT problem, for variables size $N = 60, 90, \ldots, 210$, we generate 100 random instances for each α with $\alpha_{start} = 2$, $\triangle\alpha = 0.1$, $\alpha_{end} = 6$ by uniform

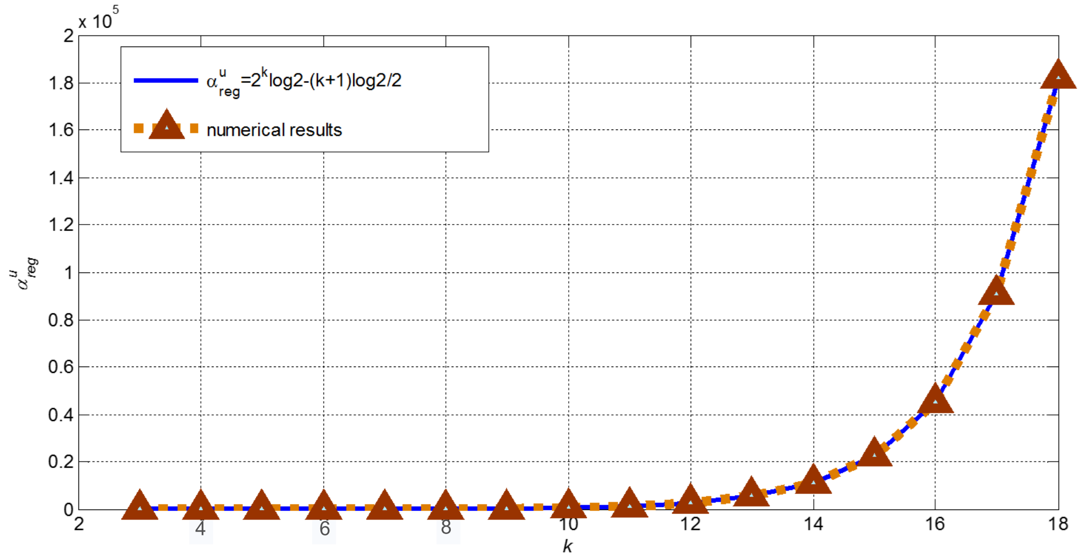

FIGURE 2. The numerical analysis results versus the approximate upper bound of $2^k \log 2 - (k+1)\log 2/2$ for $k = 3, 4, \dots, 18$

random k-SAT model (the total number of instances is $6 \times 100 \times [1 + (\alpha_{end} - \alpha_{start})/\Delta\alpha] = 24600$). Then for each $N \in \{60, 90, \dots, 210\}$, we compute 100 random generated instances for each α by Zchaff algorithms, record the corresponding computation time and whether it is satisfiable for each formula.

Experimental results show that, for the strictly regular random 3-SAT instances generated by model GSRR, with variables size $N \geq 150$ and $s > 11$, all the $(3, s)$-SAT instances are unsatisfied; however, with $s < 11$, all the $(3, s)$-SAT instances are satisfied, that is, the threshold point of the strictly regular random $(3, s)$-SAT instances is located at $s = 11$ (i.e., $\alpha^*_{reg} = 11/3 \simeq 3.6667$), which is very close to the theoretical upper bound 3.78 in our paper. Since α_{reg} can only take values from a discrete set of possible values, it indicates that our upper bound is very close to the real phase transition point in this problem.

Figure 3 illustrates the empirical phase transitions results for uniform random 3-SAT instances and strictly regular random $(3, s)$-SAT instances with different-sized variables. It is easy to see that the empirical threshold point for the uniform random 3-SAT instances is very close to 4.2667, which coincides with the conjecture in [10] and the threshold point of the strictly regular random $(3, s)$-SAT problem is close to 11/3, which is smaller than the uniform 3-SAT problem. It implies that the regular SAT problem is harder to satisfy than the uniform one.

TABLE 2. The average solution time for $\alpha_{reg} \simeq 3.6667$ in strictly regular random $(3, s)$-SAT problem and $\alpha_s \simeq 4.2667$ in uniform 3-SAT problem in every 100 random instances with different-sized variables. T_1 is the average solution time for strictly regular random $(3, s)$-SAT instances and T_2 is the average solution time for uniform random 3-SAT instances.

N	60	90	120	150	180	210
T_1	0.0292	0.4384	4.8541	109.4090	1760.3780	57679.60
T_2	0.0015	0.1030	0.6521	1.3417	8.2935	76.859

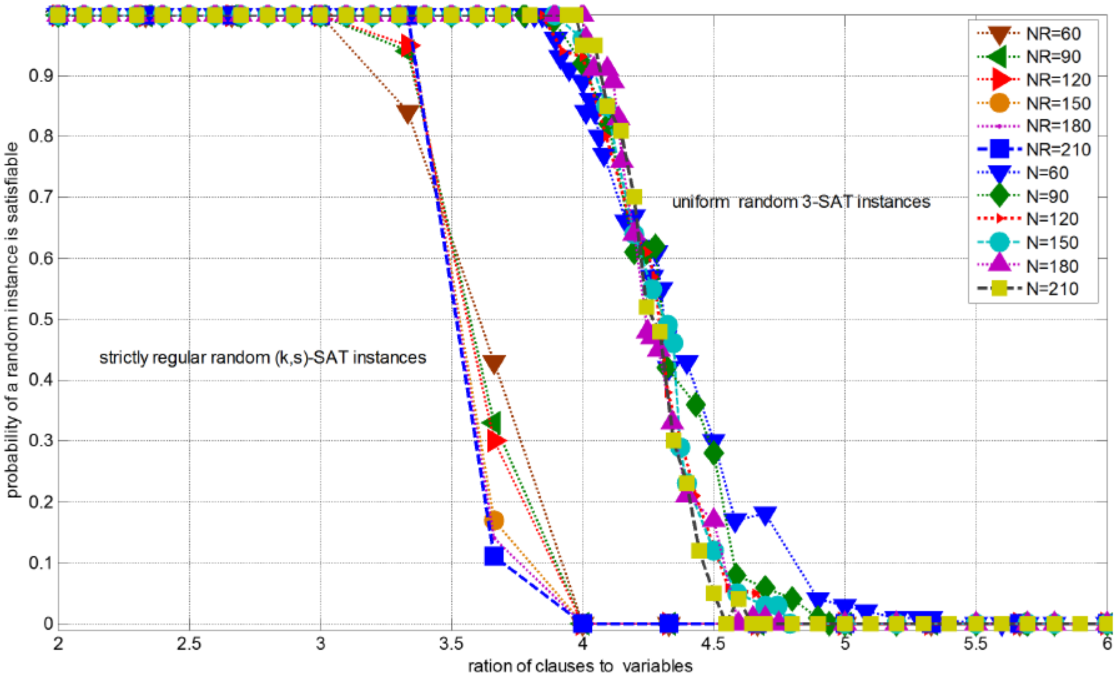

FIGURE 3. Empirical phase transitions for uniform random 3-SAT instances and strictly regular random $(3, s)$-SAT instances with different-sized variables. The figure shows the probability that a random formula is satisfiable for different N, computed over 100 instances with $s = 6, 7, \ldots, 18$ (i.e., $\alpha_{reg} = 6/3, 7/3, \ldots, 18/3$) for the strictly regular random $(3, s)$-SAT problem, and over 100 instances with $\alpha_{start} = 2$, $\triangle\alpha = 0.1$, $\alpha_{end} = 6$ for the uniform random 3-SAT problem.

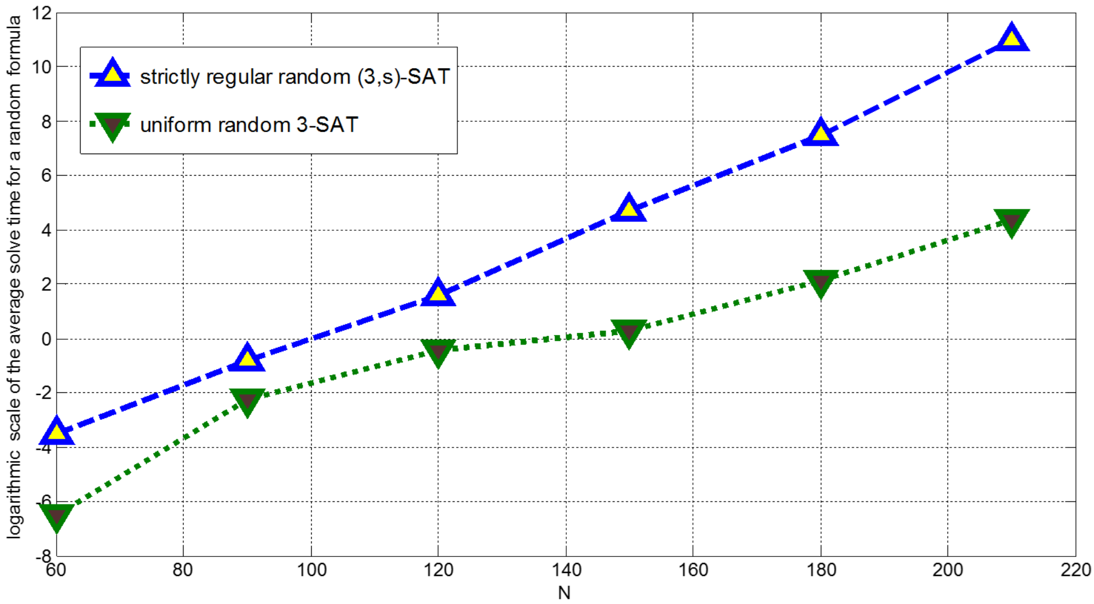

FIGURE 4. The logarithm scale of the average solution time for a strictly regular random $(3, s)$-SAT instance and a uniform random 3-SAT instance with different-sized variables in each phase transition region

Table 2 illustrates that, compared to the uniform random 3-SAT instances around its phase transition point $\alpha_s \simeq 4.2667$, the strictly regular random $(3, s)$-SAT instances are much more difficult to solve at the location $\alpha^*_{reg} \simeq 3.6667$.

In Figure 4, the two curves respectively represent the logarithm scale of the average solution time for a strictly regular random $(3, s)$-SAT instance and a uniform random 3-SAT instance with different-sized variables in each phase transition region. Thus, both the strictly regular random $(3, s)$-SAT hardest instances and the uniform random 3-SAT hardest instances generated in its corresponding threshold point where $\alpha_{reg} \simeq 3.6667$ and $\alpha_s \simeq 4.2667$, the difficulty grows exponentially with N (note the use of a log scale), and clearly, the strictly regular random $(3, s)$-SAT problem instances in its threshold point generated by model GSRR are far more difficult to solve than the uniform 3-SAT problem instances in its threshold point.

7. **Conclusions.** In this paper, we considered a strictly regular random (k, s)-SAT problem and we proposed an instances generating model, named GSRR model for this problem. Based on the asymptotic approximation of $[z^\gamma]\{f(z)^\lambda\}$ for large γ and λ growing at a fixed rate, we calculated the upper bound α^u_{reg} on the satisfiability threshold for the regular random (k, s)-SAT formulas for $k \geq 3$ by counting the number of the solutions. We showed that with the clause density $\alpha^u_{reg} > 2^k \log 2 - (k + 1) \log 2/2 + \epsilon_k$, there is no satisfying assignments with high probability. This bound is also blow the asymptotic bound of the uniform k-SAT problem, which is known as $2^k \log 2 - (\log 2 + 1)/2 + o_k(1)$ in [16, 17] for large k. Thus, it is also shown why the regular random (k, s)-SAT formulas instances are computationally harder than the uniform k-SAT instances theoretically, which coincides with the observation in our experiment. Together with the lower bound of [25], we just left an additive gap of a constant 1 for strictly regular random (k, s)-SAT problem. Moreover, it is quite easy to generate hard random k-SAT instances by our GSRR model. We believe that the GSRR model should be useful both for experimental evaluation of algorithms and theoretical research.

In addition, in the uniform random k-SAT problem, 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. However, in the strictly regular random (k, s)-SAT problem, each literal occurs either $\lfloor s/2 \rfloor$ times or $\lfloor s/2 \rfloor + 1$ times. Due to the lack of variation between literal occurrences, one cannot exploit obvious differences in the frequency of literal occurrences to design more efficient algorithms. Therefore, how to develop new algorithms with new branching heuristics to this problem will be our future research direction.

Acknowledgment. The authors would like to thank the anonymous referees for their valuable suggestions and comments. This work is partially supported by the National Natural Science Foundation of China (No. 61262006, No. 61463044, No. 61462001), the Major Applied Basic Research Program of Guizhou Province (No. JZ20142001), the Science and Technology Foundation of Guizhou Province (No. LH20147636, No. LKQS201313) and the Foundation of Qiannan Normal College for Nationalities (QNSY2011QN10, 2014ZCSX13).

REFERENCES

[1] A. K. Mackworth, Consistency in networks of relations, *Artificial Intelligence*, vol.8, no.1, pp.99-118, 1997.

[2] S. A. Cook, The complexity of theorem-proving procedures, *Proc. of the 3rd ACM Symp. on Theory of Comput.*, Shaker Heights, Ohio, pp.151-158, 1971.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completness*, W. H. Freeman and Company, San Francisco, 1979.

[4] S. A. Cook and D. G. Mitchell, Finding hard instances of the satisfiability problem: A survey, *Proc. of the DIMACS Workshop on Satisfiability Problems*, American Mathematical Society, pp.1-17, 2000.
[5] P. Cheeseman, B. Kanefsky and W. M. Taylor, Where the really hard problems are, *Proc. of the 12th Int. Joint Conf. Artif. Intell.*, Sydney, Australia, pp.331-337, 1991.
[6] D. Mitchell, B. Selman and H. Levesque, Hard and easy distributions of SAT problems, *Proc. of the 10th National Conf. Artif. Intell.*, San Jose, California, pp.459-465, 1992.
[7] T. Hogg, B. A. Huberman and C. P. Williams, *Artificial Intelligence: Special Issue on Frontiers in Problem Solving: Phase Transitions and Complexity*, vol.81, 1996.
[8] R. Monasson, R. Zecchina, S. Kirkpatrick et al., Determining computational complexity from characteristic 'phase transitions', *Nature*, vol.400, no.6740, pp.133-137, 1999.
[9] M. Mézard, G. Parisi and R. Zecchina, Analytic and algorithmic solution of random satisfiability problems, *Science*, vol.297, no.5582, pp.812-815, 2002.
[10] S. Mertens, M. Mézard and R. Zecchina, Threshold values of random K-SAT from the cavity method, *Random Structures & Algorithms*, vol.28, no.3, pp.340-373, 2006.
[11] M. Alava, J. Ardelius, E. Aurell et al., Circumspect descent prevails in solving random constraint satisfaction problems, *Proc. of the National Academy of Sciences*, vol.105, no.40, pp.15253-15257, 2008.
[12] E. Friedgut and J. Bourgain, Sharp thresholds of graph properties, and the k-sat problem, *J. of the American mathematical Society*, vol.12, no.4, pp.1017-1054, 1999.
[13] D. Achlioptas and F. Ricci-Tersenghi, On the solution-space geometry of random constraint satisfaction problems, *Proc. of the 38th ACM Symp. on Theory of Comput.*, Seattle, Washington, pp.130-139, 2006.
[14] D. Achlioptas and F. Ricci-Tersenghi, Random formulas have frozen variables, *SIAM J. Comput.*, vol.39, no.1, pp.260-280, 2009.
[15] M. Mézard and R. Zecchina, Random k-satisfiability problem: From an analytic solution to an efficient algorithm, *Physical Review E*, vol.66, no.5, pp.249-264, 2002.
[16] A. Coja-Oghlan, The asymptotic k-sat threshold, *Proc. of the 46th ACM Symp. on Theory of Comput.*, New York, pp.804-813, 2014.
[17] J. Ding, A. Sly and N. Sun, Proof of the satisfiability conjecture for large k, *arXiv Preprint arXiv:1411.0650*, 2014.
[18] S. Kirkpatrick and B. Selman, Critical behavior in the satisfiability of random boolean expressions, *Science*, vol.264, no.5163, pp.1297-1301, 1994.
[19] V. Chvátal and B. Reed, Mick gets some (the odds are on his side), *Proc. of the 33rd IEEE Symp. on Foundations of Computer Science*, Pittsburgh, pp.620-627, 1992.
[20] W. F. de La Vega, *On Random 2-SAT*, Unpublished Manuscript, 1992.
[21] A. Goerdt, A remark on random 2-SAT, *Discrete Applied Mathematics*, vol.96, pp.107-110, 1999.
[22] A. Coja-Oglan and K. Panagiotou, Catching the k-NAESAT threshold, *Proc. of the 44th ACM Symp. on Theory of Comput.*, New York, pp.899-908, 2012.
[23] Y. Boufkhad, O. Dubois, Y. Interian and B. Selman, Regular random k-SAT: Properties of balanced formulas, *J. of Automated Reasoning*, vol.35, pp.181-200, 2005.
[24] J. Ding, A. Sly and N. Sun, Satisfiability threshold for random regular NAE-SAT, *Proc. of the 46th ACM Symp. on Theory of Comput.*, New York, pp.814-822, 2014.
[25] V. Rathi, E. Aurell, L. K. Rasmussen and M. Skoglund, Bounds on threshold of regular random k-SAT, *Proc. of the 12th SAT*, pp.264-277, 2010.
[26] V. Bapst and A. Coja-Oghlan, The condensation phase transition in the regular k-SAT model, *arXiv Preprint arXiv:1507.03512*, 2015.
[27] F. R. Kschischang, B. J. Frey and H. A. Loeliger, Factor graphs and the sum-product algorithm, *IEEE Trans. Info. Theory*, vol.47, no.2, pp.498-519, 2001.
[28] D. Gardy, Some results on the asymptotic behavior of coefficients of large powers of functions, *Discrete Mathematics*, vol.139, pp.189-217, 1995.
[29] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, 2009.
[30] H. Robbins, A remark on Stirling's formula, *American Mathematical Monthly*, pp.26-29, 1995.
[31] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang and S. Malik, Chaff: Engineering an efficient SAT solver, *Proc. of the 39th ACM Design Automation Conference*, New York, pp.530-535, 2001.