electronics

MDPI

Article

The Transition Phenomenon of (1,0)-d-Regular (k,s)-SAT

Zufeng Fu [1,2], Haiying Wang [1], Jinjiang Liu [1,2], Jincheng Zhou [3,4,*], Daoyun Xu [5] and Yihai Pi [6]

1 School of Computer Science and Technology, Nanyang Normal University, Nanyang 474500, China
2 Henan Engineering Research Center of Intelligent Processing for Big Data of Digital Image, Nanyang 473061, China
3 School of Computer and Information, Qiannan Normal University for Nationalities, Duyun 558000, China
4 Key Laboratory of Complex Systems and Intelligent Optimization of Guizhou Province, Duyun 558000, China
5 College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
6 School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China
* Correspondence: zjc81@sgmtu.edu.cn

Abstract: For a d-regular (k,s)-CNF formula, a problem is to determine whether it has a (1,0)-super solution. If so, it is called (1,0)-d-regular (k,s)-SAT. A (1,0)-super solution is an assignment that satisfies at least two literals of each clause. When the value of any one of the variables is flipped, the (1,0)-super solution is still a solution. Super solutions have gained significant attention for their robustness. Here, a d-regular (k,s)-CNF formula is a special CNF formula with clauses of size exactly k, in which each variable appears exactly s-times, and the absolute frequency difference between positive and negative occurrences of each variable is at most a nonnegative integer d. Obviously, the structure of a d-regular (k,s)-CNF formula is much more regular than other formulas. In this paper, we certify that, for $k \geq 5$, there is a critical function $\varphi(k,d)$ such that, if $s \leq \varphi(k,d)$, all d-regular (k,s)-CNF formulas have a (1,0)-super solution; otherwise (1,0)-d-regular (k,s)-SAT is NP-complete. By the Lopsided Local Lemma, we get an existence condition of (1,0)-super solutions and propose an algorithm to find the lower bound of $\varphi(k,d)$.

Keywords: d-regular (k,s)-CNF; Lopsided Local Lemma; SAT-problem; transition phenomenon; (1,0)-super solutions

Citation: Fu, Z.; Wang, H.; Liu, J.; Zhou, J.; Xu, D.; Pi, Y. The Transition Phenomenon of (1,0)-d-Regular (k,s)-SAT. *Electronics* **2022**, *11*, 2475. https://doi.org/10.3390/electronics11152475

Academic Editor: Neal N. Xiong

Received: 29 June 2022
Accepted: 6 August 2022
Published: 8 August 2022

1. Introduction

In recent years, a great deal has been done to improve the efficiency of SAT solvers. Most techniques assume that all constraints in the SAT problem are fixed and inflexible. In many real world problems, conditions are partially known, imprecise and dynamic. For example, renewable energy modeling and prediction are often disturbed by many natural factors, and so are wireless sensor networks. The stability of prediction schemes has received attention in [1–3]. In the process of problem solving, not only may some activities not achieve expected results, but also some unexpected circumstances may disrupt the execution of the solution. For instance, it is important to guarantee robustness in the face of changing operating conditions for context-aware and smart systems. Therefore, Powell pointed out that dealing with uncertainty in optimization is increasingly recognized as a key necessity for tackling real-world problems in [4]. In a dynamic, uncertain or interactive environment, once some constraints are changed or the implementation of the solution encounters unexpected difficulties, the solution no longer works. Accordingly, it is worth sacrificing some optimality for a robust solution that is resilient to change. The robust solution is much less sensitive to small changes in uncertain and dynamic environments, and guarantees that some small fixes can meet the challenge of the future changes. In order to quantify the robustness of a solution, the concept of the (a,b)-super solution was introduced in [5]. For given a solution, if the values of any a variables are no longer available, flipping values of the a variables and no more than b other variables can tackle the problem. The solution is called an (a,b)-super solution. A (1,0)-super solution is a special case of an (a, b)-super solution. If the value of any one of the variables is flipped, a (1,0)-super solution is still a solution. That is, for all clauses, at least two literals can be satisfied by a (1,0)-super solution. When a CNF formula has a (1,0)-super solution, we denote that the CNF formula is (1,0)-satisfiable; otherwise it is considered (1,0)-unsatisfiable.

We observed a growing need to extend SAT to deal with more sophisticated problems in the real world. For example, the Super SAT problem is a gap version of SAT in which each assignment is attached integer weights, and was introduced in [6–8] to prove the hardness of approximation of some popular lattice problems. The promise SAT problem is a promise version of SAT, such as Unique SAT with a promise of a unique satisfying assignment in [9], (a, g, k)-SAT with a promise of g-satisfying assignment in [10,11], etc. In [10], it was shown that a simple random walk algorithm can solve a g-satisfiable k-CNF formula in expected polynomial time for $g \geq k/2$. A g-satisfiable k-CNF formula implies there is an assignment that satisfies at least g literals of every clause of the formula. It is observed that these formulas with special solutions must have their own characteristics.

Analysing regular structures of CNF formulas is often cited as the starting point of studying SAT problems. For a (k, s)-CNF formula, each clause is the disjunction of k distinct literals and each variable appears in at most s clauses. For a regular (k, s)-CNF formula, it is modified so that each variable appears in exactly s clauses. In [12–14], we introduced the d-regular (k, s)-CNF formula. For a d-regular (k, s)-CNF formula, it is further restricted that the absolute difference between positive and negative occurrences of each variable is no more than a nonnegative integer d. Obviously, for regular (k, s)-CNF formulas and d-regular (k, s)-CNF formulas, their constrained density α (the clause-to-variable ratio) is fixed. This renders some methods based on the constrained density no longer effective. The structure of a d-regular (k, s)-CNF formula is much more regular than (k, s)-CNF formula. In [12], we proved that a d-regular (k, s)-SAT problem has the Transition Phenomenon from triviality (output the affirmative answer without computation) to NP-completeness. In [14], it was shown that the random d-regular (3,s)-SAT problem has an SAT-UNSAT (satisfiable-unsatisfiable) phase transition. In [15], the structural information of formulas was used to solve the SAT problem. These more regular structures contribute to analyzing SAT problems.

We focus on (1,0)-d-regular (k, s)-SAT. It is to determine if a given d-regular (k, s)-CNF formula is (1,0)-satisfiable. For the special SAT problems, their NP-completeness deserves further research.

In this paper, our main contributions are described below.

(i) We analyze the structure of the d-regular (k, s)-CNF formula, investigate the NP-completeness of (1,0)-d-regular (k, s)-SAT and give some conditions for retaining the NP-completeness.
(ii) We propose some reduction methods, and prove that (1,0)-d-regular (k, s)-SAT has the Transition Phenomenon under the certain conditions.
(iii) By the Lopsided Local Lemma, we get an existence condition for a (1,0)-super solution and propose an algorithm to obtain a better lower bound of $\varphi(k, d)$.

This paper is structured as follows: related works are described in Section 2. All necessary preliminary definitions and lemmas appear in Section 3. The NP-completeness of (1,0)-d-regular (k, s)-SAT is discussed in Section 4. The Transition Phenomenon of (1,0)-d-regular (k, s)-SAT is in Section 5.

2. Related Works

In order to deal with uncertainty of constraints, various approaches have been proposed by researchers. Fault tolerant solutions were introduced in [16] by R. Weigel and C. Bliek for constraint programming (CP), and Wallace and Freuder in [17] introduced the concept of stable solutions for the dynamic constraint satisfaction problem. For the propositional satisfiability problem, Ginsberg, Parkes and Roy in [18] came up with Supermodels to measure solution robustness. An (a,b)-super solution was introduced in [5] to

SAT problems. It is a generalization of Supermodels. In order to find super solutions, some algorithms were presented in [19–21], including some local search algorithms.

The robustness is a valuable property of solutions, particularly for decision problems and combinatorial optimization problems. However, robustness and reliability all have a cost. We expect slight reconfiguration is enough to cope with changes in the environment. A (1,0)-super solution just meets the requirement. Even if the value of any one variable on a (1,0)-super solution is inverted, it still is a satisfying assignment. A (1,0)-super solution can be able to cope with assignment breach of only one variable. The problem to determine if a CNF formula is (1,0)-satisfiable is represented as (1,0)-SAT, and this corresponds to a k-CNF formula. It was shown in [22] that, for $k \leq 3$, (1,0)-k-SAT is in P; otherwise it is in NP-complete. They also proved that, for Constrained Density (which is the clause-to-variable ratio) $\alpha < 1/3$, a random 3-CNF formula is (1,0)-satisfiable with high probability, and not (1,0)-satisfiable with high probability for $\alpha > 1/3$. Moreover, the cutoff point is called the phase transition point. For $k = 3$, the phase transition point is equal to $1/3$. For $k \geq 4$, its upper bound is $2kln2/(k+1)$. A better lower bound of it was obtained in [23] by utilizing an enhanced weighting scheme.

Kratochvíl, Savický and Tusa in [24] studied the transition phenomenon of (k,s)-SAT. They indicated that, for $k \geq 3$, a critical function $f(k)$ can be found such that

(i) any one of the (k,s)-CNF formulas is satisfiable for $s \leq f(k)$;
(ii) The (k,s)-SAT problem is NP-complete for $s \geq f(k) + 1$.

The critical function $f(k)$ equals just the maximum of s that can be set to ensure that any one of the (k,s)-CNF formulas has a solution. They showed that $f(k) \geq \lfloor 2^k/ek \rfloor$ by using the Lovász Local Lemma in [24]. Berman, Karpinski and Scott in [25] obtained a better lower bound of $f(k)$ by using the Lopsided Local Lemma. In [26], it was shown that $f(k) \geq \lfloor 2^{(k+1)}/e(k+1) \rfloor$. In [12], we proved that d-regular (k,s)-SAT also has the Transition Phenomenon from triviality (output the affirmative answer without computation) to NP-completeness, and gave some favorable properties of the critical function $f(k,d)$. In [27], we gave some existence conditions of a (1,0)-super solution, and pointed out that if there is an unsatisfiable (1,0)-(k,s)-SAT instance, then (1,0)-(k,s)-SAT problem is NP-complete for $k > 3$. That paper shows that the (1,0)-(k,s)-SAT problem also exhibits the Transition Phenomenon. The corresponding critical function is marked as $\varphi(k)$.

The Lovász Local Lemma proposed in [28] is regarded as a classical tool in probabilistic combinatorics. It provides a sufficient condition to avoid all events deemed to be 'bad' in a probability space. In [29], an algorithm based on the variable framework of the Lovász Local Lemma was discovered to sample satisfying assignments of k-CNF formulas with bounded variable occurrences. In [30], by utilizing the Lovász local lemma, some algorithms based on Markov chains were presented to sample and approximately count satisfying assignments of (k,s)-CNF formulas. The Lopsided Local Lemma was proposed in [31] based on a lopsidependency graph.

For the study, we put forward a new reduction method to transform from k-SAT to (1,0)-d-regular $(k+1,s)$-SAT if $k \geq 4$ and an unsatisfiable (1,0)-d-regular $(k+1,s)$-SAT instance is given. It shows that, for $k \geq 5$, a critical function $\varphi(k,d)$ can be found such that for $s \leq \varphi(k,d)$ all d-regular (k,s)-CNF formulas are (1,0)-satisfiable, and for $s > \varphi(k,d)$ (1,0)-d-regular (k,s)-SAT is NP-complete. It is observed that, for $k \geq 5$, (1,0)-d-regular $(k+1,s)$-SAT also has the Transition Phenomenon from triviality to NP-completeness. Furthermore, we give some characteristics of the critical function $\varphi(k,d)$ and show a better lower bound of $\varphi(k,d)$ by Lopsided Local Lemma.

3. Notations

Given a propositional variable x, the variable has two corresponding literals: positive literal (the variable itself x) and negative literal (negation of the variable is $\neg x$). A clause $C = L_1 \vee L_2 \vee \ldots \vee L_k$ is an elementary disjunction of these literals, and is also simply written as $C = L_1, L_2, \ldots, L_k$. A CNF formula $\Psi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ is the conjunction of a set of clauses, and is also simply written as $\Psi = [C_1, C_2, \ldots, C_m]$. For a formula

Ψ, $var(\Psi)$ denotes the set of variables occurring in F, the number of these variables is $\#var(\Psi)$, and the number of clauses is $\#cl(\Psi)$. For a formula Ψ and a variable x, $Occ(\Psi, x)$ is the number of occurrences of x in the formula Ψ, $Pos(\Psi, x)$ is the number of positive occurrences of x in the formula Ψ, and $Neg(\Psi, x)$ corresponds to negative occurrences. That is, $Pos(\Psi, x) = Neg(\Psi, x) + Pos(\Psi, x)$.

If two formulas Φ and Ψ are SAT-*equivalents*, then the two formulas are either simultaneously satisfiable or simultaneously unsatisfiable. Given a CNF formula Ψ, if Ψ' is a copy of Ψ but does not have the same variable with Ψ, then Ψ' is regarded as the disjoint copy of Ψ. Variables can be divided into two categories: forced variables and unforced variables. Given a formula, if all satisfying assignments force a variable to be a same value, then the variable is termed a forced variable.

Definition 1. *A regular (k, s)-CNF is a CNF formula such that the length of each clause is exactly k and each variable appears exactly s-times. If the absolute difference between positive and negative occurrences of every variable is no more than $d \geq 0$, such a regular (k, s)-CNF formula is called a d-regular (k, s)-CNF formula.*

Definition 2. *A minimal (1,0)-unsatisfiable formula is a CNF formula such that it is not (1,0)-satisfiable but is (1,0)-satisfiable as soon as any of its clauses is removed.*

Definition 3. *For $k \geq 3$, the critical function of (k, s)-SAT, denoted by $f(k)$, is the maximum of s such that any one of (k, s)-CNF formulas has a solution. The critical function of d-regular (k, s)-SAT is denoted $f(k, d)$. The critical function of $(1, 0)$-(k, s)-SAT, denoted by $\varphi(k)$, is the maximum of s such that any one of (k, s)-CNF formulas must be (1,0)-satisfiable.*

Definition 4. *A (k, s)-CNF formula Φ is called a forced-d-regular (k, s)-CNF formula if*
(i) there are two variables x and y such that $Occ(\Phi, x) = 1$, $Occ(\Phi, y) = s - 1$ and

$$|Pos(\Phi, y) - Neg(\Phi, y) - 1| \leq d;$$

(ii) except for x, y, each variable exactly occurs s-times, and the absolute difference between positive and negative occurrences of every variable is at most $d \geq 0$;
(iii) Φ must be (1,0)-satisfiable and for any one (1,0)-super solution τ, it is tenable that $\tau(x) = \tau(y) = true$.

Definition 5 ([31])**.** *Given an undirected graph $G = (X, E)$ and its vertex set X. A collection of events in a probability space is denoted as $A = \{A_x\}_{x \in X}$. G is called a lopsidependency graph for the events A if*

$$Pr\left[A_x | \bigcap_{y \in Y(x)} \bar{A_y}\right] \leq Pr[A_x].$$

Here $Y(x) = X/N^+(x)$, $N^+(x) = \{N(x)/x\}$ and $N(x)$ is the set of neighbours of x.

Lemma 1 ([31])**.** *(The Lopsided Local Lemma) Suppose a graph $G = (X, E)$ is the lopsidependency graph for a collection of events $A = \{A_x\}_{x \in X}$. If there are real numbers $\{p_x\}_{(x \in X)}$ such that, for each $x \in X$*

$$Pr[A_x] \leq p_x \prod_{y \in N(x)} (1 - p_y),$$

then $\Pr\left[\bigcap_{x \in X} \bar{A_x}\right] > 0.$

Lemma 2 ([12])**.** *For given $k \geq 3$ and $s > 0$, if an unsatisfiable d-regular (k, s)-CNF formula can be found, then the d-regular (k, s)-SAT problem is NP-complete.*

Lemma 3 ([12]). $f(k,d) \leq f(k+1,d)$.

Lemma 4. *Given a formula F, if its representation matrix is*

$$\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{array} \begin{pmatrix} + & & & & - \\ - & + & & & \\ & - & & & \\ & & \ddots & & \\ & & & + & \\ & & & - & + \end{pmatrix},$$

then F is satisfiable and any one of variables is forced to be a same value by all satisfying assignments.

In the representation matrix of a CNF formula, each row corresponds to a variable and each column corresponds to a clause. If an element a_{ij} is $+$ (i is row number and j is column number), this implies the ith variable appears as positive literal in the jth clause. If $a_{ij} = -$, then the ith variable appears as negative literal in the jth clause. Otherwise, $a_{ij} = 0$. Evidently, F implies that

$$x_1 \to x_2 \to \ldots \to x_{n-1} \to x_n \to x_1.$$

4. The NP-Completeness of (1,0)-d-Regular (k,s)-SAT

In the field of complexity theory, the NP-completeness of a hard problem is an important attribute. In this section, we will determine some requirements such that (1,0)-d-regular (k,s)-SAT problem is a NP-complete problem.

Theorem 1. *Given $k > 3$ and $s > d \geq 0$, (1,0)-d-regular $(k+1,s)$-SAT is an NP-complete problem if d-regular (k,s)-SAT is an NP-complete problem.*

Proof. Because d-regular (k,s)-SAT is an NP-complete problem, we only need to construct a reduction method to transform a d-regular (k,s)-SAT problem to a (1,0)-d-regular $(k+1,s)$-SAT problem.

Assume we have a d-regular (k,s)-CNF formula Ψ which has $nk > 0$ variables and ns clauses, $n \geq 1$. There are four steps to our reduction method, as described below.

Step 1. Introduce a fresh set of variables $X = \{x_i : 1 \leq i \leq t\}$. Here, $t > max\{ns, \dfrac{2n(sk-s)}{sk-3s}\}$, and $ts - ns$ is a multiple of $k+1$.

Step 2. Let $\Psi_1 = \wedge_{C_i \in \Psi}(C_i \vee x_i)$.

Step 3. Build a new k-CNF formula Ψ_2 consisting of X, and satisfying the following requirements.

(i) For $1 \leq i \leq ns$ and each variable $x_i \in X$, $Pos(\Psi_2, x_i) = \lceil s/2 \rceil - 1$ and $Occ(\Psi_2, x_i) = s - 1$;
(ii) For $ns + 1 \leq i \leq t$ and each variable $x_i \in X$, $Occ(\Psi_2, x_i) = s$ and $Pos(\Psi_2, x_i) = \lceil s/2 \rceil$;
(iii) Any one of clauses of Ψ_2 includes no less than two positive literals from X.

Step 4. Construct a new CNF formula $\Phi = \Psi_1 \wedge \Psi_2$.

Obviously, Φ is just a d-regular $(k+1,s)$-CNF formula. Next, the top concern is the feasibility of constructing Φ. Because $ts - ns$ is a multiple of $k+1$, we only focus on whether the condition (iii) is going to be met.

In the subformula Ψ_2 constructed in Step 3, the number of clauses is

$$\#cl(\Psi_2) = \frac{ts - ns}{k+1},$$

and the number of non-negative occurrences of X is

$$Pos(\Psi_2, X) = t\lceil s/2 \rceil - ns.$$

For $k > 3$, we get $sk - 3s > 0$. For $t > max\{ns, \frac{2n(sk-s)}{sk-3s}\}$, we get

$$\begin{aligned} t(sk-3s) &> 2n(sk-s), \\ tsk - 2nsk + st - 2ns &> 4st - 4ns, \\ ts - 2ns &> \frac{4ts - 4ns}{k+1}, \\ ts/2 - ns &> 2\frac{ts - ns}{k+1}, \\ Pos(\Psi_2, X) &> 2\#cl(\Psi_2). \end{aligned}$$

From this, there are more than twice as many non-negative literals from X in Ψ_2 as clauses. That is, the formula Ψ_2 can be constructed in polynomial time by simple placement. Hence, constructing Φ is practicable.

Finally, we will show that Ψ is satisfiable iff Φ is (1,0)-satisfiable.

First, we presume that an assignment τ satisfies Ψ. Define a new truth assignment $\tau'(x)$ as

$$\tau'(x) := \begin{cases} \tau(x), & if\ x \in var(\Psi) \\ true, & if\ x \in X \end{cases}$$

Obviously, at least two literals of every clause of Φ can be satisfied by the truth assignment $\tau'(x)$. This would mean that $\tau'(x)$ is a (1,0)-super solution of Φ.

Next, let us suppose Φ has a (1,0)-super solution τ. What that means is that no less than two literals of every clause of Ψ_1 are satisfied by τ. Because $\Psi_1 = \wedge_{C_i \in \Psi}(C_i \vee x_i)$, there must be no less than two literals satisfied by τ for any clause of $C_i \vee x_i$. It is obvious that at least one literal of $C_i \in \Psi$ is satisfied by τ. We get that τ satisfies Ψ, and Ψ is satisfiable.

Based on the reduction method, we obtain that if the d-regular (k,s)-SAT problem is an NP-complete problem, then the (1,0)-d-regular $(k+1,s)$-SAT also is an NP-complete problem for $k > 3$.

The proof is completed. □

Corollary 1. *If there is a d-regular (k,s)-CNF formula without a solution, then the (1,0)-d-regular $(k+1,s)$-SAT problem is an NP-complete problem, for $k > 3$ and $s > d \geq 0$.*

Proof. The corollary can be directly derived from Lemma 2 and Theorem 1. □

Lemma 5. *For $k > 4$ and $s > d \geq 0$, if a d-regular (k,s)-CNF formula is satisfiable but is (1,0)-unsatisfiable, then we can construct a forced-d-regular (k,s)-CNF formula.*

Proof. Given a d-regular (k,s)-CNF formula Φ that meets the requirements. A (1,0)-unsatisfiable formula can change into a (1,0)-satisfiable formula just by removing some clauses. Because it has a solution but no (1,0)-super solution, we remove some clauses of Φ to obtain a minimal (1,0)-unsatisfiable formula (marked as Φ_1). According to definition, for every variable in Φ_1, the number of positive occurrences is at most $(s+d)/2$, and so is the number of negative occurrences. A conjunction of the removed clauses form a formula Φ_2. Then we construct $\Phi = \Phi_1 \wedge \Phi_2$. Assuming Φ_2 has $m \geq 0$ clauses, and this showed that it has mk literals. The clause set of Φ_1 is denoted as C_1 and the clause set of Φ_2 is denoted as C_2.

We will put forward a construction method to generate some forced-d-regular (k,s)-CNF formulas. There are three steps to our method, as described below.

Step 1. A clause c of Φ_1 is randomly selected. It is supposed that a new formula obtained by removing the clause c from Φ_1, has a (1,0)-super solution τ. Obviously, for the clause c, only one literal can be satisfied by τ. Let $\neg y$ be a literal of c unsatisfied by τ. Let x be a new extra variable that does not appear in Φ. We generate $C = (C_1 \backslash \{c\}) \cup \{\tilde{c}\}$, with $\tilde{c} = (c \backslash \{\neg y\}) \cup \{x\}$. Define $\widetilde{\Phi} := \wedge_{c_i \in C} c_i$.

Step 2. Introduce a fresh set of variables $Z = \{z_1, z_2, \ldots, z_{tk}\}$. Here, let $t > 2m/(ks - 4s)$. Using mk literals of Φ_2 and the variable set Z, we generate a k-CNF formula Φ_3 that satisfies the following requirements.

(i) For each variable z of Z, z appears exactly s-times, and $Pos(\Phi_3, z) = \lceil s/2 \rceil$;
(ii) Every literal of Φ_2 occurs only once in Φ_3;
(iii) Any one of the clauses of Φ_3 includes no less than two positive literals from Z.

Step 3. Construct a new CNF formula $\Phi' = \widetilde{\Phi} \wedge \Phi_3$.

Define a new truth assignment $\tau'(u)$ as

$$\tau'(u) := \begin{cases} \tau(u), & u \in var(\Phi) \\ true, & u \in Z \text{ or } u = x \end{cases}.$$

For the formula obtained by removing the clause c from Φ_1, τ is a (1,0)-super solution of it. Therefore, at least two literals are satisfied by τ' for any clause of Φ'. That is, τ' is a (1,0)-super solution of Φ'.

Given any (1,0)-super solution of Φ', if it forces x to be $false$ or y to be $false$, then it must be a (1,0)-super solution of Φ_1. This is in contradiction with the claim that Φ_1 is the minimal (1,0)-unsatisfiable formula. Therefore, (1,0)-super solutions of Φ' all force two variables x and y to be $true$.

Each variable in $var(\Phi')$, except for x, y, occurs in exactly s clauses. Moreover, for any one of variables, the absolute difference between positive and negative occurrences is at most d. Hence, the formula Φ' we constructed is a forced-d-regular (k, s)-CNF formula.

Next, we will show that the construction of Φ_3 is feasible. In the subformula Φ_3, the number of clauses is $m + ts$, and the number of non-negative literals from Z is $tk\lceil s/2 \rceil$. For $k > 4$ and $t > 4m/(ks - 4s)$, we obtain that

$$tks - 4ts > 4m, \quad tks/2 > 2m + 2ts, \quad tk\lceil s/2 \rceil > 2(m + ts).$$

Obviously, there are more than twice as many non-negative literals from Z in Φ_3 as clauses. That is, the formula Φ_3 can be constructed in polynomial time. Hence, in polynomial time the construction of Φ' can be done.

The proof is completed. □

Theorem 2. *For $k \geq 4$ and $s > d \geq 0$, if there is a (1,0)-d-regular $(k+1, s)$-SAT instance that is (1,0)-unsatisfiable, then the (1,0)-d-regular $(k+1, s)$-SAT problem is an NP-complete problem.*

Proof. Given a (1,0)-unsatisfiable d-regular $(k+1, s)$-CNF formula F. There are two cases to consider.

Case 1: Any assignment cannot satisfy the formula F. That is, we obtain $f(k+1, d) < s$. By Lemma 3, we also get $f(k, d) \leq f(k+1, d) < s$. Therefore, we can found an unsatisfiable d-regular (k, s)-SAT instance. By Corollary 1, it means that (1,0)-d-regular $(k+1, s)$-SAT problem is a NP-complete problem.

Case 2: There exists an assignment that can satisfy the formula F but any one of the assignments is not a (1,0)-super solution of it. We first generate a forced-d-regular $(k+1, s)$-CNF formula Φ by using the method in Lemma 5. Given any k-CNF formula Ψ with m clauses and mk literals. Next, we will put forward a reduction method that can transform k-SAT to (1,0)-d-regular $(k+1, s)$-SAT in polynomial time. Table 1 shows some variable sets that we will introduce in the method. There are five steps to our reduction method, as described below.

Table 1. The variable sets introduced in the method.

Variable Set	Size
X_1	$mk(k-4)$
X_2	mk
X_3	m
X_4	$mk-m$
Y	$mk(k-2)$
Z	mk
U	$m+3mk+t(k+1)$

Step 1. With reference to the formula Φ, we generate some disjoint copies of it, denoted as Φ_{ij}, $1 \le i \le mk$, $1 \le j \le k-2$. The two variables x and y in Φ are renamed as $x_{i,j}$ and $y_{i,j}$ in Φ_{ij}. In addition, sets of variables of these formulas are pairwise disjoint. Define

$$\begin{aligned} X_1 &= \{x_{i,j}\},\ 1 \le i \le mk,\ 3 \le j \le k-2, \\ X_2 &= \{x_{i,1}\},\ 1 \le i \le mk, \\ X_3 &= \{x_{i,2}\},\ 1 \le i \le m, \\ X_4 &= \{x_{i,2}\},\ m+1 \le i \le mk, \\ Y &= \{y_{i,j}\},\ 1 \le i \le mk,\ 1 \le j \le k-2. \end{aligned}$$

Construct a new formula $\Psi_1 = \wedge_{1\le i\le mk} \wedge_{1\le j\le k-2} \Phi_{ij}$.

Step 2. Introduce a new set of variables $Z = \{z_{i,j}\}$, $1 \le i \le m$, $1 \le j \le k$. Replace mk literals of Ψ with some variables of Z, and generate a new formula Ψ_2.

$$\Psi_2 = \wedge_{1\le i\le m}(x_{i,2} \vee (\vee_{1\le j\le k} L'_{i,j})), \quad L'_{i,j} = \begin{cases} z_{i,j}, & if\ L_{i,j} = v \\ \neg z_{i,j}, & if\ L_{i,j} = \neg v \end{cases},\ v \in var(\Psi).$$

Here, $L_{i,j}$ is denoted as the jth literal in the ith clause of Ψ.

Step 3. The variables of Z are sorted by their subscripts. Let $d_i = z_i \vee \neg z_j \vee \neg x_{i,1} \vee \neg x_{i,2} \vee \ldots \vee \neg x_{i,k-2} \vee x_{j,1}$, and generate $\Psi_3 = \bigwedge_{1\le i\le mk} d_i$. Here $z_i, z_j \in Z$, and if z_i replaces a variable v of Ψ then z_j is set to be the next variable of Z which replaces v (if z_i is the last variable in the variable set Z which replaces v, z_j is set to be the first).

Step 4. Introduce a new variable set $U = \{u_1, u_2, \ldots, u_{m+3mk+t(k+1)}\}$. Here, $t > \dfrac{mks+3ms-4mk-4m}{ks-3s}$. Using these variable sets $X_1, X_2, X_3, X_4, Y, Z, U$, we construct a $(k+1)$-CNF formula Ψ_4 satisfying the following conditions.

(i) Every variable x of X_1 and X_4 occurs exactly in $s-2$ clauses and

$$0 \le |\ Neg(\Psi_4, x) - Pos(\Psi_4, x)\ | \le d.$$

(ii) Every variable x of X_2 and X_3 occurs exactly in $s-3$ clauses and

$$0 \le |\ Neg(\Psi_4, x) + 1 - Pos(\Psi_4, x)\ | \le d.$$

(iii) All variables of Y occur negatively exactly once.

(iv) Each variable z of Z appears exactly $(s-3)$-times, and if z occurs positively in Ψ_2,

$$0 \le |\ Neg(\Psi_4, z) + 1 - Pos(\Psi_4, z)\ | \le d.$$

Otherwise

$$0 \le |\ Neg(\Psi_4, z) - Pos(\Psi_4, z) - 1\ | \le d.$$

(v) Every variable u of U occurs exactly in s clauses and

$$Pos(\Psi_4, u) - Neg(\Psi_4, u) \le min(1, d).$$

(vi) Any one of the clauses of Ψ_4 includes no less than two positive literals from U.

Step 5. Construct a new formula $\Psi' = [\Psi_1,\ \Psi_2,\ \Psi_3,\ \Psi_4]$.

It is easy to see that the formula Ψ' constructed by us is a d-regular $(k+1,s)$-CNF formula. Our biggest concern is the feasibility of constructing Ψ'. Constructing three formulas Ψ_1, Ψ_2 and Ψ_3 apparently must be done in polynomial time. We will mainly analyze Ψ_4.

In the subformula Ψ_4, the number of clauses equals

$$\begin{aligned}\#cl(\Psi_4) &= \frac{(m+3mk+t(k+1))s+(2mk+m)(s-3)}{k+1} \\ &\quad+\frac{(mk(k-3)-m)(s-2)+mk(k-2)}{k+1} \\ &= \frac{mk^2s-mk^2+2mks-2mk+ms+tks+ts-m}{k+1} \\ &= mks-mk+ms+ts-m,\end{aligned}$$

and the number of non-negative occurrences of U equals

$$Pos(\Psi_4,U) = (m+3mk+t(k+1))\lceil s/2\rceil.$$

For $k \geq 4$ and $t > \dfrac{mks+3ms-4mk-4m}{ks-3s}$,

$$\begin{aligned} tks-3ts &> mks+3ms-4mk-4m, \\ ms+3mks+tks+ts &> 4mks-4mk+4ms+4ts-4m, \\ (m+3mk+tk+t)\lceil s/2\rceil &> 2(mks-mk+ms+ts-m), \\ Pos(\Psi_4,U) &> 2\#cl(\Psi_4,U).\end{aligned}$$

Obviously, there are more than twice as many non-negative literals from U in Ψ_4 as clauses. First arrange any two positive literals of U to each clause of Ψ_4, and then randomly arrange others. The simple placement method can construct Ψ_4. It can be seen that Ψ_4 can be constructed in polynomial time.

Finally, we will show that iff Ψ is satisfiable, Ψ' is (1,0)-satisfiable.

Let's suppose that a (1,0)-super solution of Ψ' is just an assignment τ. That is, for any clause of Ψ', no less than two literals are satisfied by τ. Because all Φ_{ij} are forced-d-regular $(k+1,s)$-CNF formulas, it is significant that $\tau(x_{i,j}) = true$, $\tau(y_{i,j}) = true$, for $1 \leq i \leq mk$, $1 \leq j \leq k-2$.

Substitute $\tau(x_{i,j})$ into Ψ_3, and simplify Ψ_3. We obtain by Lemma 4 that in any one (1,0)-super solution the simplified Ψ_3 can express n cyclic of implication. This suggests, if the same variable of Ψ is replaced by z_i and z_j, we get $\tau(z_i) = \tau(z_j)$. Because of this, a new truth assignment τ' is defined as

$$\tau'(v) := \tau(z),\ \ if\ a\ variable\ v\ of\ \Psi\ is\ replaced\ with\ a\ variable\ z\ in\ Z.$$

Because no less than two literals of every clause of Ψ_2 are satisfied by the (1,0)-super solution τ, for any clause of Ψ, τ' is guaranteed to satisfy at least one literal. Therefore, Ψ is satisfiable.

Let's suppose that a truth assignment τ satisfies Ψ and an assignment $\tau_{i,j}$ is precisely a (1,0)-super solution of $\Phi_{i,j}$, $1 \leq i \leq mk$, $1 \leq j \leq k-2$. We define a new truth assignment τ' as

$$\tau'(v) := \begin{cases} \tau(x), & if\ v \in Z\ and\ a\ variable\ x\ of\ var(\Psi)\ is\ replaced\ with\ v \\ \tau_{i,j}(v), & if\ v \in var(\Phi_{ij}) \\ true, & if\ v \in U \end{cases}.$$

As a (1,0)-super solution of $\Phi_{i,j}$, $\tau_{i,j}$ is certain to satisfy two literals of any clause in $\Phi_{i,j}$, it means that, for any clause of Ψ_1, there are two literals is satisfied by τ'. Because $\Phi_{i,j}$ is a forced-d-regular $(k+1,s)$-CNF formula, we infer that $\tau'(x_{i,j}) = true$, $\tau'(y_{i,j}) = true$, $1 \le i \le mk$, $1 \le j \le k-2$. In addition, τ satisfies at least one literal of every clause of Ψ, so at least two literals of every clause of Ψ_2 and Ψ_3 are satisfied by τ'. Thus, τ' is a (1,0)-super solution of Ψ'.

For $k \ge 4$, k-SAT problem is a NP-complete problem. Therefore, (1,0)-d-regular $(k+1,s)$-SAT problem is also a NP-complete problem.

The proof is completed. □

Theorem 4 shows that for $k \ge 5$, if a d-regular (k,s)-CNF formula is (1,0)-unsatisfiable, then (1,0)-d-regular (k,s)-SAT problem is a NP-complete problem. But it is not known whether the conclusion is valid for $k = 4$. When s is an odd number, $2\lceil s/2 \rceil \ge s$. By modifying the reduction method used to proof Theorem 2, we can get that (1,0)-1-regular (4,5)-SAT, (1,0)-2-regular (4,4)-SAT, (1,0)-2-regular (4,6)-SAT are NP-complete.

Given a (4,4)-CNF formula F, its representation matrix is

$$\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{pmatrix} + & + & - & - \\ + & - & + & - \\ - & + & + & - \\ + & - & - & + \end{pmatrix}.$$

Obviously, F has no (1,0)-super solution. However, we have no idea whether (1,0)-0-regular (4,4)-SAT and (1,0)-1-regular (4,6)-SAT are NP-complete.

5. The Transition Phenomenon of (1,0)-d-Regular (k,s)-SAT

We have proved (1,0)-(k,s)-SAT problem has a Transition Phenomenon for $k \ge 3$. In this section, we will focus on the transition phenomenon of (1,0)-d-regular (k,s)-SAT.

Theorem 3. *For $k \ge 5$ and $d \ge 0$, a critical function $\varphi(k,d)$ can be found such that*

(i) for $s \le \varphi(k,d)$ any one of d-regular (k,s)-CNF formulas is (1,0)-satisfiable and
(ii) for $s > \varphi(k,d)$ (1,0)-d-regular (k,s)-SAT problem is a NP-complete problem.

Proof. The corollary can be directly derived from Theorem 2. □

It is clear that $\varphi(k,d)$ is the maximum s can be set to ensure that all d-regular (k,s)-CNF formulas must be (1,0)-satisfiable. Then, we will give some properties of the critical function $\varphi(k,d)$.

If all (k,s)-CNF formulas can find a (1,0)-super solution, all d-regular (k,s)-CNF formulas have a (1,0)-super solution. Therefore, we get $\varphi(k,d) \ge \varphi(k)$. Because the critical function $\varphi(k)$ is an increasing function, proved in [27], we obtain $\varphi(k,d) \ge \varphi(k-1)$. Any d-regular (k,s)-CNF formula obviously should be a $(d+1)$-regular (k,s)-CNF formula, so we get $\varphi(k,d) \ge \varphi(k,d+1)$. For each variable of any one of regular (k,s)-CNF formulas, the absolute difference between positive and negative occurrences is at most $s-2$. Therefore, if $\varphi(k) < s$, then $\varphi(k,s-2) < s$. For $d = \varphi(k) - 1$, we get $\varphi(k,d) = \varphi(k)$. According to some properties of the critical function $f(k,d)$ and $\varphi(k)$ presented in [12,27], we can obtain the following corollaries.

Corollary 2. *For $k \ge 5$ and $d \ge 0$, $\varphi(k,d) \ge 2$.*

Corollary 3. *For $k \ge 5$ and $d \ge 0$, $\varphi(k+1,d) \ge \varphi(k,d)$.*

Corollary 4. *For $k \ge 5$ and $d \ge 0$, $\varphi(k+1,2d) \ge 2\varphi(k,d) + 1$.*

By Corollary 3, it can be known that $\varphi(k,d)$ is an increasing function of k and a decreasing function of d. By Corollary 4, we can obtain a peculiar result that $\varphi(k+1,0) \geq 2\varphi(k,0)+1$.

Theorem 4. *For $k \geq 5$ and $d \geq 0$, we get $\varphi(k,d) \leq f(k-1,d)$.*

Proof. We can find an unsatisfiable d-regular $(k-1, f(k-1,d)+1)$-CNF formula, because $f(k-1,d)$ is the critical function of d-regular $(k-1,s)$-SAT. By Corollary 1, we get (1,0)-d-regular $(k, f(k-1,d)+1)$-SAT is NP-complete. That is, $\varphi(k,d) \leq f(k-1,d)$. □

Next, we will put forward a property of the critical function of $\varphi(k)$ and then extend it to $\varphi(k,d)$.

Theorem 5. *$\varphi(2k) \geq f(k)$.*

Proof. Given a $(2k, f(k))$-CNF formula Φ with m clauses, we break up every clause C_i of Φ into two parts with the same size, namely C_{i1} and C_{i2}. Then, a new formula Ψ is formed by these divided clauses. Obviously, Ψ is a $(k, f(k))$-CNF formula. Because $f(k)$ is the critical function of $(k-1,s)$-SAT, Ψ must be satisfiable. Let an assignment τ be a satisfying assignment of Ψ. That is, τ satisfies every pair of C_{i1} and C_{i2}. Every clause of Φ is formed a pair of C_{i1} and C_{i2}. Therefore, τ is (1,0)-super solution of Φ.

Every $(2k, f(k))$-CNF formula must have (1,0)-super solution, so $\varphi(2k) \geq f(k)$. □

The clauses split method used to prove Theorem 5 keeps the number of positive occurrences or negative occurrences of any variable unchanged, so we can derive the following corollary by the method.

Corollary 5. *For $k \geq 5$ and $d \geq 0$, $\varphi(2k,d) \geq f(k,d)$.*

By this corollary mentioned above, we derive some bounds of the critical function $\varphi(k,d)$.

$$
\begin{aligned}
&2 \leq \varphi(5,0) \leq 8,\ 2 \leq \varphi(5,1) \leq 7,\ 2 \leq \varphi(5,2) \leq 6,\ 2 \leq \varphi(5,3) \leq 4,\\
&3 \leq \varphi(6,0) \leq 10,\ 3 \leq \varphi(6,1) \leq 9,\ 3 \leq \varphi(6,4) \leq 8,\\
&3 \leq \varphi(7,0) \leq 16,\ 3 \leq \varphi(7,1) \leq 15,\ 3 \leq \varphi(7,2) \leq 14,\ 3 \leq \varphi(7,3) \leq 13,\ 3 \leq \varphi(7,4) \leq 12,\\
&4 \leq \varphi(8,d) \leq \varphi(9,d),\ 5 \leq \varphi(10,d) \leq \varphi(11,d),\\
&7 \leq \varphi(12,d) \leq \varphi(13,d),\ 13 \leq \varphi(14,d) \leq \varphi(15,d),\\
&24 \leq \varphi(16,d) \leq \varphi(17,d),\ 41 \leq \varphi(18,d) \leq \varphi(19,d).
\end{aligned}
$$

Finally, we will give some better results about the lower bound of $\varphi(k,d)$ by The Lopsided Local Lemma.

Theorem 6. *For three positive integers $k \geq 4$, s and $(s+1)/2 \geq t \geq 1$, if there is a real number $0 < p < 1$ such that*

$$\frac{p(1-p)^{kt}2^k}{k+1} \geq 1, \tag{1}$$

then every (k,s)-CNF formula in which each literal occurs in at most t clauses must be (1,0)-satisfiable.

Proof. Given a (k,s)-CNF formula Ψ that each literal occurs at most t times. Let C be the set of clauses in Ψ and X be the set of variables. We define an undirected graph G with vertex set C. For any two clauses $c_1, c_2 \in C$, if there is an edge between c_1 and c_2 if and only if there exists a variable x that occurs negated in one clause and without negation in the other. We pick a clause $c \in C$ at random and denote a clause set $S = C \backslash N^{+}(c)$. We generate a random assignment τ by setting each variable independently to be true with a probability of $1/2$. Let A_c be the event that at most one literal of c is satisfied by the assignment τ, B_c be the event that at least two literals of each clause of S are satisfied by the assignment τ.

We first show that the graph G is the lopsidependency graph for the events $A = \{A_c\}_{c\in C}$. Let X_1 be the set of variables that occur both in the clause c and in some clauses of S. By the definition of G, we obtain that for each $x \in X_1$, x occurs with the same sign in c and S. That is, if x occurs positively in c, then x must not occur negatively in any one clause of S, and vice versa. Obviously, $\overline{A_c}$ and B_c are both increasing because of these variables of X_1. By the FKG inequality on [32], we get that $\overline{A_c}$ and B_c are positively correlated, and A_c and B_c are negatively correlated. That is to say,

$$\Pr[A_c|B_c] = Pr\left[A_c \middle| \bigcap_{i\in S} \bar{A}_i\right] \leq Pr[A_c].$$

Thus, the graph G is the lopsidependency graph for the events $A = \{A_c\}_{c\in C}$.

Because every variable is assigned to true with probability $1/2$ and every clause has exactly t literals, the probability that the clause c has at most one literal satisfied is

$$Pr[A_c] = (1/2)^k + k(1/2)^k = (k+1)/2^k.$$

In the formula Ψ, each literal occurs at most t times. That is, for any variable in Ψ, the number of positive occurrences and negative occurrences are all at most t. Let $deg(c)$ be the degree of c in G. By the definition of G, we obtain that for each $c \in C$, $deg(c) \leq kt$. For $0 < p < 1$, we get

$$p \prod_{i\in N(c)} (1-p) = p(1-p)^{\deg(c)} \geq p(1-p)^{kt}.$$

For $\frac{p(1-p)^{kt}2^k}{k+1} \geq 1$, we get $p(1-p)^{kt} \geq \frac{k+1}{2^k}$. So

$$Pr[A_c] \leq p \prod_{i\in N(c)} (1-p).$$

By the Lopsided Local Lemma, we get $Pr\left[\bigcap_{c\in C} \bar{A}_c\right] > 0$. That is, for any clause of C, an assignment can satisfy at least two literals. Thus, the formula Ψ must be (1,0)-satisfiable.

The proof is completed. $\square$

For given k and t, if we can find a real number $0 < p < 1$ such that $\dfrac{p(1-p)^{kt}2^k}{k+1} \geq 1$, then every (k,s)-CNF formula, in which the number of positive and negative occurrences of all variables are at most t, must have a (1,0)-super solution. So we deduce that all 0-regular $(k,2t)$-CNF formulas, 1-regular $(k,2t-1)$-CNF formulas and 3-regular $(k,2t-3)$-CNF formulas must be (1,0)-satisfiable.

For a given k, the maximum value of t satisfying Equation (1) is useful in searching the lower bound of $\varphi(k,d)$. We design an algorithm that can find rapidly the maximum value of t such that $\dfrac{p(1-p)^{kt}2^k}{k+1} \geq 1$, for $0 < p < 1$. Let $f(k,p,t) = \dfrac{p(1-p)^{kt}2^k}{k+1}$. Figure 1 is the flowchart of the algorithm. When fixing t, the maximum value of the function f can be obtained for $p = \dfrac{1}{tk+1}$.

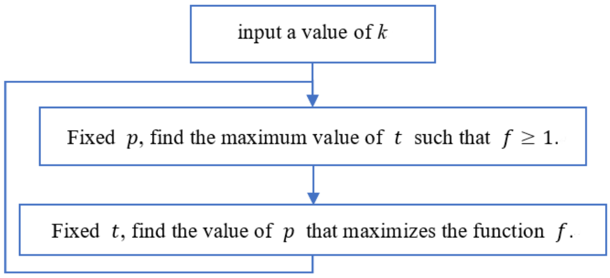

Figure 1. The flowchart of Algorithm 1, here $f(k,p,t) = \frac{p(1-p)^{kt}2^k}{k+1}$.

Utilizing Algorithm 1, we obtain some values of k, p, t that satisfy Equation (1) and show it in Table 2.

Algorithm 1: An algorithm for finding the maximum value of t satisfying Equation (1)

Input: a positive integer $k \geq 4$
Output: a positive integer t

$i \leftarrow 2$;
$t \leftarrow 1$;
$flag \leftarrow 1$;
while $flag = 1$ **do**
  $f \leftarrow 1$;
  while $f \geq 1$ **do**
    $f \leftarrow \frac{1}{i}(1-\frac{1}{i})^{t*k}\frac{2^k}{k+1}$;
    $t \leftarrow t+1$;
  end
  if $i = (t-1)*k+1$ **then**
    $flag \leftarrow 0$;
    $t \leftarrow t-1$;
  else
    $i \leftarrow (t-1)*k+1$;
    $t \leftarrow t-1$;
  end
end
$t \leftarrow t-1$;

Table 2. The values of k, p, t that satisfy Equation (1).

k	p	t
8	1/10	1
9	1/19	2
10	1/34	3
11	1/64	5
12	1/116	9
13	1/215	16
14	1/402	28

Based on these results, we obtain the better lower bound of $\varphi(k,d)$.

$$\begin{aligned}
&\varphi(9,0) \geq 4,\\
&\varphi(10,0) \geq 6,\ \varphi(10,1) \geq 5,\\
&\varphi(11,0) \geq 10,\ \varphi(11,1) \geq 9,\ \varphi(11,2) \geq 8,\ \varphi(11,3) \geq 7,\ \varphi(11,4) \geq 6,\\
&\varphi(12,0) \geq 18,\ \varphi(12,1) \geq 17,\ \varphi(12,2) \geq 16,\ \varphi(12,3) \geq 15,\ \varphi(12,4) \geq 14,\ \varphi(12,5) \geq 13,\\
&\varphi(13,0) \geq 32,\ \varphi(13,1) \geq 31,\ \varphi(13,2) \geq 30,\ \varphi(13,3) \geq 29,\ \varphi(13,4) \geq 28,\ \varphi(13,5) \geq 27,\\
&\varphi(14,0) \geq 56,\ \varphi(14,1) \geq 55,\ \varphi(14,2) \geq 54,\ \varphi(14,3) \geq 53,\ \varphi(14,4) \geq 52,\ \varphi(14,5) \geq 51.
\end{aligned}$$

6. Conclusions and the Future Work

Results of previous research indicate that the (1,0)-k-SAT problem is a P problem for $k = 3$, and the k-SAT problem with a promise of g-satisfying assignment is also a P problem for $g \geq k/2$. We investigate the NP-completeness of (1,0)-d-regular (k,s)-SAT problem, and prove that it has a sudden jump from triviality to NP-completeness for $k \geq 5$. But for $k = 4$, we are not clear about this problem. Similarly, the (3,4)-SAT problem, (4,5)-SAT problem and 0-regular (3,4)-SAT problem all have some unsatisfiable instances, the (7,6)-SAT problem and (8,7)-SAT problem all have an unsatisfiable instance, but we do not know if (5,6)-SAT has an unsatisfiable instance. Obviously, there are some unsolved problems for smaller k.

We obtain the better lower bound of $\varphi(k,d)$ by the Lopsided Local Lemma. Moser and Tardos in [33] showed that, under the same conditions as the Local Lemma, there is an efficient algorithm to find a solution of k-SAT problem, but it is not clear whether this is also the case in (1,0)-k-SAT problem. That is, we expect to find an efficient algorithm for (1,0)-SAT. In addition, there is no good way to construct some unsatisfiable (1,0)-SAT instances. These unsatisfiable (1,0)-SAT instances are conducive to finding a better upper bound of $\varphi(k,d)$. These questions are all worth probing into.

Author Contributions: Formal analysis, Z.F., H.W. and J.Z.; Investigation, Z.F., J.L. and D.X.; Methodology, Z.F., J.Z. and D.X.; Writing—original draft, Z.F., H.W. and J.Z.; Writing—review & editing, Z.F., Y.P. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant numbers (Nos. 61862051, 61762019), the Ph.D. Scientific Research Project of Nanyang Normal University (No. 2022ZX022), the Science and Technology Foundation of Guizhou Province under Grant (No. [2019]1299), the Top-notch Talent Program of Guizhou province under Grant (No. KY[2018]080), the program of Qiannan Normal University for Nationalities under Grant (Nos. QNSY2018JS013, QNSYRC201715).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the anonymous referees for their helpful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cao, Y.; Raise, A.; Mohammadzadeh, A.; Rathinasamy, S.; Band, S.; Mosavi, A. Deep learned recurrent type-3 fuzzy system: Application for renewable energy modeling/prediction. *Energy Rep.* **2021**, *7*, 8115–8127. [CrossRef]
2. Guo, W.; Xiong, N.; Chao, H.-C.; Hussain, S.; Chen, G. Design and analysis of self-adapted task scheduling strategies in wireless sensor networks. *Sensors* **2011**, *11*, 6533–6554. [CrossRef] [PubMed]
3. Wan, R.; Xiong, N. An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks. *Hum.-Centric Comput. Inf. Sci.* **2018**, *8*, 1–22. [CrossRef]
4. Powell, W.B. A unified framework for optimization under uncertainty. In Proceedings of the INFORMS 2016, Optimization Challenges in Complex, Nashville, TN, USA, 13–16 November 2016; pp. 45–83. [CrossRef]

5. Hebrard, E.; Hnich, B.; Walsh, T. Super solutions in constraint programming. *Integr. AI OR Tech. Constr. Program. Comb. Optim. Probl.* **2004**, *3011*, 157–172. [CrossRef]
6. Dinur, I.; Kindler, G.; Raz, R.; Safra, S. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica* **2003**, *23*, 205–243. [CrossRef]
7. Dinur, I. Approximating SVP^{∞} to within almost-polynomial factors is NP-hard. *Theor. Comput. Sci.* **2002**, *285*, 55–71. [CrossRef]
8. Mukhopadhyay, P. The Projection Games Conjecture and the hardness of approximation of super-SAT and related problems. *J. Comput. Syst. Sci.* **2022**, *123*, 186–201. [CrossRef]
9. Calabro, C.; Paturi, R. k-SAT Is No Harder Than Decision-Unique-k-SAT. In *Computer Science Symposium in Russia*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
10. Austrin, P.; Guruswami, V.; Håstad, J. (2+ε)-SAT Is NP-hard. *SIAM J. Comput.* **2017**, *46*, 1554–1573. [CrossRef]
11. Brandts, A.; Wrochna, M.; Živný, S. The Complexity of Promise SAT on Non-Boolean Domains. *ACM Trans. Comput. Theory* **2021**, *13*, 1–20. [CrossRef]
12. Fu, Z.; Xu, D. The NP-completeness of d-regular (k,s)-SAT problem. *J. Softw.* **2020**, *31*, 1113–1123. [CrossRef]
13. Fu, Z.; Xu, D. Uniquely Satisfiable d-regular (k,s)-SAT Instances. *Entropy* **2020**, *22*, 569. [CrossRef]
14. Wang, Y.; Xu, D. Properties of the satisfiability threshold of the strictly d-regular random (3,2s)-SAT problem. *Front. Comput. Sci.* **2020**, *14*, 146404. [CrossRef]
15. Zhang, Z.; Xu, D.; Zhou, J. An algorithm for solving satisfiability problem based on the structural information of formulas. *Front. Comput. Sci.* **2021**, *15*, 156405. [CrossRef]
16. Weigel, R.; Bliek, C. On reformulation of constraint satisfaction problems. In Proceedings of the 13th European Conference on Artificial Intelligence (ECAI98), Brighton, UK, 23–28 August 1998; pp. 254–258.
17. Wallace, R.J.; Freuder, E.C. Stable solutions for dynamic constraint satisfaction problems. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, Pisa, Italy, 26–30 October 1998; pp. 447–461. [CrossRef]
18. Ginsberg, M.; Parkes, A.; Roy, A. Supermodels and robustness. In Proceedings of the 15th National Conference on Artificial Intelligence (AAAI 98), Madison, WI, USA, 26–30 July 1998; pp. 334–339.
19. Holland, A.; O'Sullivan, B. Super Solutions for Combinatorial Auctions. *Recent Adv. Constr.* **2004**, *3419*, 187–200. [CrossRef]
20. Hebrard, E.; Wslsh, T. Improved algorithm for finding (a,b)-super solutions. *Princ. Pract. Constr. Program.* **2005**, *3709*, 848. [CrossRef]
21. Holland, A.; O'Sullivan, B. Weighted Super Solutions for Constraint Programs. In Proceedings of the Twentieth National Conference on Artificial Intelligence & the Seventeenth Innovative Applications of Artificial Intelligence Conference, Pittsburgh, PA, USA, 9–13 July 2005; pp. 378–383.
22. Zhang, P.; Gao, Y. A probabilistic study of generalized solution concepts in satisfiability testing and constraint programming. *Theor. Comput. Sci.* **2016**, *657*, 98–110. [CrossRef]
23. Zhou, G.; Kang, R. On the Lower Bounds of (1,0)-Super Solutions for Random k-SAT. *Int. J. Found. Comput. Sci.* **2019**, *30*, 247–254. [CrossRef]
24. Kratochvíl, J.; Savický, P.; Tuza, Z. One more occurrence of variables makes satisfiability jump from trivial to NP-complete. *Acta Inf.* **1993**, *22*, 203–210. [CrossRef]
25. Berman, P.; Karpinski, M.; Scott, A.D. Approximation hardness and satisfiability of bounded occurrence instances of SAT. *Electron. Colloquium Comput. Complex.* **2003**, *10*, TR03-022.
26. Gebauer, H.; Szabo, T.; Tardos, G. The Local Lemma is asymptotically tight for SAT. *ACM* **2016**, *63*, 664–674. [CrossRef]
27. Fu, Z.; Xu, D. (1,0)-Super Solutions of (k,s)-CNF Formula. *Entropy* **2020**, *22*, 253. [CrossRef] [PubMed]
28. Erd˝os, P.; Lovász, L. Problems and results on 3-chromatic hypergraphs and some related questions. *Infin. Finite Sets* **1975**, *10*, 609–627.
29. Guo, H.; Jerrum, M.; Liu, J. Uniform Sampling Through the Lovász Local Lemma. *J. ACM* **2019**, *66*, 1–31. 3310131. [CrossRef]
30. Feng, W.; Guo, H.; Yin, Y.; Zhang, C. Fast Sampling and Counting k-SAT Solutions in the Local Lemma Regime. *J. ACM* **2021**, *68*, 1–42. [CrossRef]
31. Erdős, P.; Spencer, J. Lopsided Lovász local lemma and Latin transversals. *Discret. Appl. Math.* **1991**, *30*, 151–154. [CrossRef]
32. Fortuin, C.M.; Kasteleyn, P.W.; Ginibre, J. Correlation inequalities on some partially ordered sets. *Commun. Math. Phys.* **1971**, *22*, 89–103. [CrossRef]
33. Moser, R.A.; Tardos, G. A constructive proof of the general lovász local lemma. *J. ACM* **2010**, *57*, 1–15. [CrossRef]