entropy

MDPI

Article

Uniquely Satisfiable d-Regular (k,s)-SAT Instances

Zufeng Fu [1,2] and Daoyun Xu [1,*]

[1] College of Computer Science and Technology, Guizhou University, Guiyang 550025, China; fuzufeng@outlook.com
[2] Department of Electronics and Information Engineering, Anshun University, Anshun 561000, China
* Correspondence: dyxu@gzu.edu.cn; Tel.: +85-851-83620667

Received: 22 March 2020; Accepted: 17 May 2020; Published: 19 May 2020

Abstract: Unique k-SAT is the promised version of k-SAT where the given formula has 0 or 1 solution and is proved to be as difficult as the general k-SAT. For any $k \geq 3$, $s \geq f(k,d)$ and $(s+d)/2 > k-1$, a parsimonious reduction from k-CNF to d-regular (k,s)-CNF is given. Here regular (k,s)-CNF is a subclass of CNF, where each clause of the formula has exactly k distinct variables, and each variable occurs in exactly s clauses. A d-regular (k,s)-CNF formula is a regular (k,s)-CNF formula, in which the absolute value of the difference between positive and negative occurrences of every variable is at most a nonnegative integer d. We prove that for all $k \geq 3$, $f(k,d) \leq u(k,d)+1$ and $f(k,d+1) \leq u(k,d)$. The critical function $f(k,d)$ is the maximal value of s, such that every d-regular (k,s)-CNF formula is satisfiable. In this study, $u(k,d)$ denotes the minimal value of s such that there exists a uniquely satisfiable d-regular (k,s)-CNF formula. We further show that for $s \geq f(k,d)+1$ and $(s+d)/2 > k-1$, there exists a uniquely satisfiable d-regular $(k, s+1)$-CNF formula. Moreover, for $k \geq 7$, we have that $u(k,d) \leq f(k,d)+1$.

Keywords: d-regular (k,s)-CNF; SAT-problem; uniquely satisfiable

1. Introduction

Satisfiability Problem (SAT) is a central problem in theoretical computer science of deciding whether a given Conjunction Normal Formula (CNF) is satisfiable. The k-SAT is a satisfiability problem where every clause has exactly k distinct variables, and was proved to be a NP-complete problem for $k \geq 3$ in [1]. That is, SAT problem should be a computationally hard problem. However, modern SAT solvers are able to efficiently solve some formulas with millions of variables, such as MiniSat [2], Glucose [3], Maple [4]. The conflict-driven clause learning technique is an important algorithm to improve the efficiency of these SAT solver. Yet, how these solvers can be so successful has remained elusive. In order to analyze and improve SAT solvers, some random SAT models were propose.

A natural measure of the solution space is the number of solutions. Unique k-SAT denotes the promise search problem of k-SAT where the number of solutions is either 0 or 1. The harder instances should have fewer solutions. But Calabro and Paturi in [5] proved that the exponential complexity of deciding whether a k-CNF formula has a solution is the same as that of deciding whether it has exactly one solution, both when it is promised and when it is not promised that the input formula has a solution. Thus, the research of uniquely satisfiable SAT instances is a very significant work.

The (k,s)-SAT denotes the family of satisfiability problems restricted to CNF formulas with exactly k distinct variables per clause and at most s occurrences of each variable. Regular (k,s)-SAT is a class of special (k,s)-SAT which each variable occurs in exactly s clauses. By some polynomial time reductions, it is discovered that some SAT problems with regular structures are NP-complete, such as (3,4)-SAT problem in [6] and regular (3,4)-SAT problem in [7]. Experimental results and theoretical analysis on a random k-SAT problem showed that the constrained density α of a CNF formula is an important parameter affecting the formula satisfiability and the solving difficulty in [8–11]. There is a phase transition point $\alpha(k)$ on a random k-SAT problem such that

(i) all random k-CNF instances with $\alpha < \alpha(k)$ are satisfiable with high probability;
(ii) all random k-CNF instances with $\alpha > \alpha(k)$ are unsatisfiable with high probability.

But every regular (k,s)-CNF formula has a fixed constrained density α (the clause-to-variable ratio), such as regular (3,4)-CNF formula corresponding to 4/3. The constrained density of the regular (3,4)-CNF is much smaller than the SAT-UNSAT phase transition point of the random 3-SAT problem $\alpha(3) \approx 4.267$ in [12]. This shows that a random regular (3,4)-CNF formula is satisfiable with high probability, but the regular (3,4)-SAT problem is NP-complete. Obviously, it is not enough to describe structural features of the CNF formula merely by the constrained density α.

In [13,14], M. Wahlström presented a definition of (a,b)-variable to classify all variables in a CNF formula, and designed two algorithms for solving a CNF formula with at most d occurrences per variable. Here, an (a,b)-variable is a variable which occurs positively in a clauses and negatively in b clauses. In [15], Johannsen, Razgon and Wahlström presented an algorithm for solving a CNF formula in which the number of occurrences of each literal is at most d. Their results demonstrated that the CNF formulas with some restrictions on the number of occurrences (positive or negative) of each variable have its own characteristics.

In order to further study SAT problems with regular structures, we introduced d-regular (k,s)-CNF formula in [16,17]. The regular (k,s)-CNF formula requires that each clause contains exactly k variables and each variable occurs in exactly s clauses. The d-regular (k,s)-CNF formula also requires that the absolute value of the difference between positive and negative occurrences of each variable is no more than a nonnegative integer d. In this paper, we investigate the existence condition of uniquely satisfiable d-regular (k,s)-SAT Instances, and present a method to construct a uniquely satisfiable d-regular (k,s)-formula. We also give a parsimonious reduction from k-CNF to d-regular (k,s)-CNF, and further explain the constrained density is not enough to describe the structural features of a CNF formula.

2. Related Works

Unique SAT is the promised version of the SAT, where a given CNF formula has 0 or 1 solution. Valiant and Vazirani in [18] gave a randomized polynomial time reduction from SAT to Unique SAT, and showed that deciding whether a CNF formula has zero or one solution is essentially as difficult as SAT in general. Calabro et al. in [19] proved that Unique k-SAT is no easier than k-SAT, not just for polynomial time algorithms but also super-polynomial time algorithms. They in [5] pointed out it does not matter whether there has a promise that a formula has a solution. Matthews in [20] studied the complexity of UNIQUE-(k,s)-SAT and proved that $f(k) \leq u(k) \leq f(k) + 2$ for $k \geq 3$, where $u(k)$ is the minimal value of s so that uniquely satisfiable (k,s)-CNF formulas exist and $f(k)$ represents the maximal value of s such that all (k,s)-CNF formulas are satisfiable. The exact values of $f(k)$ are only known for $k = 3$ and $k = 4$, because $f(3) = 3$, $f(4) = 4$ were shown in [21]. In [22–25], it showed that the upper and lower bounds for $k = 5, 6, \ldots, 9$, $f(k)$ are described as follows

$$5 \leq f(5) \leq 7, 7 \leq f(6) \leq 11, 13 \leq f(7) \leq 17, 24 \leq f(8) \leq 29, 41 \leq f(9) \leq 51.$$

Encoding into a CNF formula is a common way to solve a practical problem. These CNF formulas often have some special structures and properties. It is important to design some random SAT models that are similar to reality. Markström in [26] proposed a constructor method of SAT instance based on Eulerian graphs, and discussed how a solver can try to avoid at least some of the pitfalls presented by these instances. Giraldez-Cru and Levy in [27] proposed a new model of generation of random SAT instances with community structure, and showed that modern solvers do actually exploit this community structure. In [28], they presented a random SAT instances generator based on the notion of locality, and showed that CDCL SAT solvers take advantage of both popularity and similarity.

In [29,30], it showed that SAT instances with less solutions tend to be harder for stochastic local search methods. In [31], Žnidarič gave an experimental evaluation of uniquely satisfiable 3-SAT instances obtained by simply filtering randomly generated formulas.

In this paper, we investigate a uniquely satisfiable d-regular (k,s)-SAT Instances, and show that $f(k,d) \leq u(k,d)+1$, $f(k,d+1) \leq u(k,d)$ for $k \geq 3$, and $u(k,d) \leq f(k,d)+1$ for $k \geq 7$. Here $u(k,d)$ denotes the minimal value of s such that uniquely satisfiable d-regular (k,s)-CNF formulas exist, and $f(k,d)$ denotes the maximal value of s such that all d-regular (k,s)-CNF formulas are satisfiable. We demonstrate that for $s \geq f(k,d)+1$ and $(s+d)/2 > k-1$, there is a uniquely satisfiable d-regular (k,s)-CNF formula. We also reveal that for $k \geq 7$, if a d-regular (k,s)-CNF formula is unsatisfiable, then $(s+d)/2 > k-1$. Finally, for $k \geq 3$, $s \geq f(k)+1$ and $(s+d)/2 > k-1$, we give a parsimonious reduction from a k-CNF formula to a d-regular (k,s)-CNF formula. Constructing uniquely satisfiable d-regular $(k,s+1)$-CNF formulas from an unsatisfiable d-regular (k,s)-CNF formula is a key component of our reduction.

3. Notations

A literal is a boolean variable x or a negated boolean variable $\neg x$. x is called a positive literal, and $\neg x$ is called a negative literal. A clause C is a disjunction of literals, $C = L_1 \vee L_2 \vee \ldots \vee L_k$ or $C = \{L_1, L_2, \ldots, L_k\}$. A formula F in the conjunctive normal formula is a conjunction of clauses, $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ or $F = \{C_1, C_2, \ldots, C_m\}$. $var(F)$ denotes the set of boolean variables occurring in a formula F, and $\#var(F)$ refers to the number of variables occurring in F. $\#cl(F)$ denotes the number of clauses of F, and $pos(F,x)$ ($neg(F,x)$) refers to the number of positive (negative) occurrences of a variable x in F. $pos(F)$ ($neg(F)$) denotes the number of positive (negative) literals in F, and $pos(F,X)$ ($neg(F,X)$) refers to the number of positive (negative) occurrences of all variables of the variable set X in F.

A truth assignment τ is a function which assigns to each boolean variable v a unique value $\tau(v) = \{0,1\}$. A CNF formula F is satisfiable, if a truth assignment τ with $\tau(F) = 1$ exists. Such a truth assignment is called a satisfying assignment. We divide boolean variables in these formulas into forced variables or unforced variables. If every satisfying assignment of a formula sets a variable to the same value, we call it a forced variable. Otherwise, the variable is regarded as an unforced variable.

If the formulas Φ and Ψ are either satisfiable at the same time or not, they are called SAT-*equivalents*. This implies that, Φ is satisfiable if and only if Ψ is satisfiable. A formula F' is called the disjoint copy of a CNF formula F, if F' is a copy of F and their variable sets are disjoint. A uniquely satisfiable d-regular (k,s)-CNF formula is a d-regular (k,s)-CNF with only one solution. A CNF formula F is a minimal unsatisfiable formula (MU), if F is unsatisfiable and $F - \{C\}$ is satisfiable for any clause $C \in F$. For a given unsatisfiable formula F, a minimal unsatisfiable formula can be obtained by removing some clauses from F.

Definition 1. *For each $k \geq 3$, $f(k)$ is defined as the maximal value of s such that all (k,s)-CNF formulas are satisfiable, $f(k,d)$ is defined as the maximal value of s such that all d-regular (k,s)-CNF formulas are satisfiable, $u(k)$ is defined as the minimal value of s such that uniquely satisfiable (k,s)-CNF formulas exist, and $u(k,d)$ is defined as the minimal value of s such that uniquely satisfiable d-regular (k,s)-CNF formulas exist.*

Definition 2. *A k-CNF formula F is called a k-forced-once d-regular (k,s)-CNF formula if*

(i) *there exist k variables $x_1, x_2, \ldots, x_k$ that only occur once;*

(ii) *except for the k variables, every variable occurs in exactly s clauses, and the absolute value of the difference between positive and negative occurrences of every variable is no more than the nonnegative integer d.*

(iii) *F is satisfiable and for any truth assignment τ satisfying F, it holds that*

$$\tau(x_1) = \tau(x_2) = \ldots = \tau(x_k) = true.$$

We can represent a CNF formula as a matrix. Each variable x_i corresponds to a row of the matrix and each clause C_j corresponds to a column of the matrix. For each variable x_i, if its positive (resp., negative) literal is in the clause C_j, then $a_{i,j} = +$ (resp., $a_{i,j} = -$); otherwise, 0.

Let F is a CNF formula with 15 variables $x_1, x_2, \ldots, x_{15}$ and 25 clauses $C_1, C_2, \ldots, C_{25}$. The representation matrix of the formula F is

$$\begin{array}{cc}
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{array} &
\begin{pmatrix}
+ & - & + & + & & & - & - \\
+ & - & - & - & + & + \\
+ & - & & & - & - & + & + \\
& & + & - & + & - & + & & & & & & & & & & & & & & & & & & - \\
& & & & & & & + & & & & & & & & & - & - & - & + & + \\
& & & & & & & & + & - & + & + & & & - & - \\
& & & & & & & & + & - & - & - & + & + \\
& & & & & & & & + & - & & & - & - & + & + \\
& & & & & & & & & & + & - & + & - & + & & & & & & & & & - \\
& & & & & & & & & & & & & & & + & - & - & - & & & + & + \\
& & & & & & & & & & & & & & & & & & & - & - & - & + & + & + \\
& & & & & & & & & & & & & & & & & & & + & + & + & - & - & - \\
& & & & & & & & & & & & & & & & + \\
& & & & & & & & & & & & & & & & & + \\
& & & & & & & & & & & & & & & & & & +
\end{pmatrix}
\end{array}.$$

Clearly, F is a 3-forced 0-regular (3,6)-CNF formula. Each of the three variables x_{13}, x_{14}, x_{15} occurs in exactly one clause in F and is forced to be *true*.

Definition 3. *In the context of SAT, a reduction M is identified to be parsimonious if x and $M(x)$ have the same number of satisfying assignments for any one formula x.*

Lemma 1 ([32]). *Let (k, s)-CNF be a class of satisfiable formulas, then all $(k + r, s + r[s/k])$-CNF formulas are satisfiable for any nonnegative integer r ($[x]$ denotes the integral part of x).*

Lemma 2 ([17]). *If the representation matrix of a formula F is*

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

then the formula is satisfiable and every satisfying assignment forces all variables to a same value.

4. Uniquely Satisfiable d-Regular (k, s)-CNF Formula

The d-regular (k, s)-CNF formula has stronger regular constraints than the regular (k, s)-CNF formula. It limits the absolute value of the difference between positive and negative occurrences of each variable. The uniquely satisfiable d-regular (k, s)-CNF formula refers to a d-regular (k, s)-CNF formula with only one solution. We investigate the existence conditions of the uniquely satisfiable d-regular (k, s)-CNF formula.

Theorem 1. *For all $k \geq 3$, $f(k, d) \leq u(k, d) + 1$ and $f(k, d + 1) \leq u(k, d)$.*

Proof. Because $f(k, d)$ denotes the maximal value of s such that all d-regular (k, s)-CNF formulas are satisfiable, we usually construct an unsatisfiable d-regular (k, s)-CNF formula to find the upper bound of $f(k, d)$.

Let $s = u(k, d)$. Because $u(k, d)$ denotes the minimal value of s such that uniquely satisfiable d-regular (k, s)-CNF formulas exist, there must be a uniquely satisfiable d-regular (k, s)-CNF formula F. Obviously, by adding a clause to F which is violated by the unique satisfying assignment, the formula

F can become an unsatisfiable formula. Suppose the formula F has n variables. We give two methods to construct unsatisfiable instances.

Method 1: We introduce $\lceil n/k \rceil k - n$ new variables and add $\lceil n/k \rceil (s+2) - ns/k$ new clauses to F, which contains at least one clause violated by the unique satisfying assignment. Let each original variable occurs twice in the new clauses (one negative occurrence and another positive occurrence), and each new variable occurs $s+2$ times in the new clauses (the number of positive and negative occurrences of every new variable is nearly equal). That is, each variable occurs $s+2$ times in F and the absolute value of the difference between positive and negative occurrences of each variable is no more than d. Therefor, F is turned into an unsatisfiable d-regular $(k, s+2)$-CNF formula. It can be seen that $f(k,d) \leq s+1 = u(k,d)+1$.

Method 2: We introduce $\lceil n/k \rceil k - n$ new variables and add $\lceil n/k \rceil (s+1) - ns/k$ new clauses to F, which contains at least one clause violated by the unique satisfying assignment. Let each original variable occur once in the new clauses, and each new variable occurs $s+1$ times in the new clauses (the number of positive and negative occurrences of every new variable is nearly equal). That is, each variable occurs $s+1$ times in F and the absolute value of the difference between positive and negative occurrences of each variable is no more than $d+1$. Therefor, F is turned into an unsatisfiable $(d+1)$-regular (k,$s+1$)-CNF formula. It can be seen that $f(k, d+1) \leq s = u(k,d)$. □

Lemma 3. *If $k \geq 3$ and s are two nonnegative integers such that an unsatisfiable d-regular (k,s)-CNF formula exists, there exists a k-forced-once d-regular (k,s)-CNF formula.*

Proof. Let Φ be an unsatisfiable d-regular (k,s)-CNF formula. Obviously, the number of positive occurrences and negative occurrences of every variable in Φ are all no more than $(s+d)/2$. By removing some clauses of Φ, a minimal unsatisfiable (k,s)-CNF formula Φ_1 can be obtained. It is easy to get that, the number of positive occurrences and negative occurrences of every variable in Φ_1 are all no more than $(s+d)/2$.

Let $\Phi = \Phi_1 \wedge \Phi_2$, where Φ_1 is the unsatisfiable (k,s)-CNF formula obtained by removing some clauses of Φ, and Φ_2 is a conjunction of the removed clauses. Suppose Φ_2 contains $m \geq 0$ clauses and mk literals. Let C_1 be the clause set of Φ_1 and C_2 be the clause set of Φ_2. A variable y of $var(\Phi_1)$ and a clause c containing $\neg y$ are randomly selected. Define $C = (C1 \backslash \{c\}) \cup \{\tilde{c}\}$, with $\tilde{c} = (c \backslash \{\neg y\}) \cup \{x\}$, where x is a new extra variable that does not occur in Φ. Define $\Phi_1' = \wedge_{c \in C} c$. Clearly, the variable x is forced to be *true*.

Let Φ_{1i} be disjoint copies of the formula Φ_1' with the variable x, y of Φ_1' being renamed as x_i, y_i in Φ_{1i}, and Φ_{2i} be disjoint copies of the formula Φ_2, for $1 \leq i \leq k$. In addition, we ensure that every variable occurring both in Φ_1' and Φ_2 is renamed as a same new variable in Φ_{1i} and Φ_{2i}, respectively, for $1 \leq i \leq k$.

Introduce a new boolean variable set $Z = z_1, z_2, \ldots, z_{tk}$ which does not occur in Φ, $t > 2mk/s$. The k-CNF formula Φ_3 is constructed using $\neg y_i$, the literals of Φ_{2i} and the variables of Z, for $1 \leq i \leq k$. And it shall meet the following limits.

(i) Every variable of Z occurs positively in $\lceil s/2 \rceil$ clauses and negatively in $\lfloor s/2 \rfloor$ clauses;
(ii) All literals of Φ_{2i} and $\neg y_i$ occur exactly once in Φ_3, $1 \leq i \leq k$;
(iii) Every clause of Φ_3 must have at least one positive occurrence of any one of Z.

Define $\Phi' = \Phi_3 \wedge \Phi_{11} \wedge \Phi_{12} \wedge \ldots \wedge \Phi_{1k}$.

Obviously, condition (i) and (ii) of Definition 2 hold in Φ' (note $s > 3$ from the unsatisfiability of Φ). Φ_{1i} is satisfiable and forces the variable x_i to be *true*. Because every variable of Z does not occur in Φ_1', Φ_3 is satisfiable (let the value of every variable of Z be *true*) without affecting $\Phi_{11}, \Phi_{12}, \ldots, \Phi_{1k}$. So it can be concluded that Φ' is satisfiable and forces $x_1, x_2, \ldots, x_k$ to be *true*. Φ_1', Φ_2 and $\neg y$ only contain x and all literals of Φ. Except for x, every variable of Φ_1', Φ_2 and $\neg y$ occur in s clause, and meet the d-regularity (Φ is a d-regular (k,s)-CNF formula). Hence, Φ_{1i}', Φ_{2i} and $\neg y_i$ meet these requirements (by the definition of disjoint copy). Every variable of Z occurs positively in $\lceil s/2 \rceil$ clauses and negatively in $\lfloor s/2 \rfloor$ clauses. Thus, $x_1, x_2, \ldots, x_k$ occur only once in Φ'. Except for the k variables, every variable occurs in exactly s clauses, and the absolute value of the difference between positive and negative occurrences of every variable is no more than d. Therefore, we claim that Φ' is a k-forced-once d-regular (k, s)-CNF formula.

Next, we will assess the feasibility of the construction of Φ'. If an unsatisfiable d-regular (k, s)-CNF formula Φ exists, Φ'_1 should be easily constructed. The number of literals of Φ_3 is $mk^2 + tks + k$, and the number of positive occurrences of the variables of Z in Φ_3 is $kt\lceil s/2 \rceil$. The number of clauses of Φ_3 is $mk + ts + 1$. For $t > 2mk/s$, we obtain $mk < ts/2$, $mk + ts < 3ts/2$. For $k \geq 3$, we obtain $mk + ts + 1 \leq kts/2 \leq kt\lceil s/2 \rceil$. As a result, the number of positive occurrences of Z in Φ_3 is greater than that of clauses of Φ_3. The construction of Φ_3 is almost random (First let each clause get a positive literal of Z, then randomly arrange other literals). Therefore, Φ_3 can be constructed in polynomial time. □

Lemma 4. *For $k \geq 3$, $(s + d)/2 > k - 1$ and $m \geq 1$, we can transform a k-forced-once d-regular (k, s)-CNF formula with n unforced variables into a $(m + 1)k$-forced-once d-regular (k, s)-CNF formula with n unforced variables.*

Proof. Let Φ be a k-forced-once d-regular (k, s)-CNF formula with n unforced variables, and $x_1, x_2, \ldots, x_k$ denote k forced variables that only occur once. That is, $x_1, x_2, \ldots, x_k$ are forced to be *true*. Let

$$H_0 = \wedge_{i=1}^{k}(\neg x_1 \vee \neg x_2 \vee \ldots \vee \neg x_{k-1} \vee y_{1,i}),$$
$$H_j = \wedge_{i=1}^{k}(\neg y_{j,1} \vee \neg y_{j,2} \vee \ldots \vee \neg y_{j,k-1} \vee y_{j+1,i}), j = 1, 2, \ldots, mk - 1,$$

where every $y_{j,i}$ is a fresh variable.

We construct a k-CNF formula Ψ with the variable set $X = \{x_i\}$ and the variable set $Y = \{y_{j,i}\}$, for $i = 1, 2, \ldots, k - 1$, $j = 1, 2, \ldots, mk - 1$, which meets the following restrictions.

(i) every variable of X and Y occurs in exactly $s - k - 1$ clauses of Ψ,

$$\text{if } s \leq 2k,\ pos(\Psi, x_i) = s - k - 1, pos(\Psi, y_{j,i}) = s - k - 1;$$

$$\text{if } s > 2k,\ pos(\Psi, x_i) = \lceil s/2 \rceil - 1, pos(\Psi, y_{j,i}) = \lceil s/2 \rceil - 1.$$

(ii) Every clause of Ψ must have at least one positive occurrence of any one of these variables.

Define $\Phi' = \Phi \wedge H_0 \wedge H_1 \wedge \ldots \wedge H_{mk-1} \wedge \Psi$.

Obviously, x_i and $y_{j,i}$ are forced to be *true* for $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, mk$ (this ensures that Ψ is satisfiable). In these forced variables, $x_k, y_{1,k}, y_{2,k}, \ldots, y_{mk-1,k}$ and $y_{mk,1}, y_{mk,2}, \ldots, y_{mk,k}$ occur exactly in one clause of Φ'. Except for the $(m + 1)k$ variables, every variable occurs in exactly s clauses, and the absolute value of the difference between positive and negative occurrences of every variable is at most d. The number of unforced variables in Φ' is still n. So Φ' is a $(m + 1)k$-forced-once d-regular (k, s)-CNF formula with n unforced variables.

Next, we will prove that the construction of Ψ is feasible. We focus on the satisfiability of the condition ii.

The variables of Ψ consists of two parts: X and Y. The variable set Y has $(mk - 1)(k - 1)$ variables. The variable set X has $k - 1$ variables. Every variable of X and Y occurs in exactly $s - k - 1$ clauses of Ψ. Obviously, the number of literals of Ψ is $(mk - 1)(k - 1)(s - k - 1) + (k - 1)(s - k - 1)$. The number of clauses of Ψ is

$$\begin{aligned}\#cl(\Psi) &= \frac{(mk-1)(k-1)(s-k-1) + (k-1)(s-k-1)}{k} \\ &= m(k-1)(s-k-1).\end{aligned}$$

When $s \leq 2k$, all literals in Ψ are positive literal and must satisfy the condition iii. When $s > 2k$, the number of positive occurrences of the variables in Ψ is

$$\begin{aligned} pos(\Psi) &= (mk-1)(k-1)(\lceil s/2 \rceil - 1) + (k-1)(\lceil s/2 \rceil - 1) \\ &= mk(k-1)(\lceil s/2 \rceil - 1). \end{aligned}$$

For $k \geq 3$,

$$\begin{aligned} mk(k-1)(\lceil s/2 \rceil - 1) &= m(k-1)(k\lceil s/2 \rceil - k) \geq m(k-1)(3\lceil s/2 \rceil - k) \\ &\geq m(k-1)(3s/2 - k) > m(k-1)(s-k) \\ &> m(k-1)(s-k-1). \end{aligned}$$

So $pos(\Psi) > \#cl(\Psi)$. That indicates that the number of positive literals is more than that of clauses. That is, we can arrange a positive literal for every clause of Ψ, then randomly arrange other literals. Hence, Ψ can be constructed. □

Theorem 2. *For $k \geq 3$ and $s \geq f(k,d)+1$ and $(s+d)/2 > k-1$, there exists a uniquely satisfiable d-regular (k,s)-CNF formula.*

Proof. We will show a way to construct a uniquely satisfiable d-regular (k,s)-CNF formula.

By Lemma 3 and Lemma 4, for $k \geq 3$, $s \geq f(k,d)+1$, $(s+d)/2 > k-1$ and $m \geq 1$, we can construct a $(m+1)k$-forced-once d-regular (k,s)-CNF formula Ψ. It is assumed that the $(m+1)k$ forced variables which occur only once are $X = \{x_1, x_2, \ldots, x_{(m+1)k}\}$. Without loss of generality, we assume that forcing n of unforced variables to be *true* can turn Ψ into a uniquely satisfiable formula. Let $Y = \{y_1, y_2, \ldots, y_n\}$ denote the n unforced variables. Let $t = \lceil n/(k-1) \rceil$. Constructing a uniquely satisfiable d-regular (k,s)-CNF formula is based on four stages, which are described as follows.

Step 1 Divide the variables $y_1, y_2, \ldots, y_n$ arbitrarily into t variable sets $Y_1, Y_2, \ldots, Y_t$ of size $k-1$. Some variables of Ψ forced to be *true* are added, so that every variable set contains exactly $k-1$ variables (a variable forced to be $false$ can be transformed to a variable forced to be *true* by flipping all occurrences of the variable). The variables $x_1, x_2, \ldots, x_{(m+1)k}$ are arbitrarily divided into $4t+1$ variable sets $X_1, X_2, \ldots, X_{4t+1}$. Moreover, it should be guaranteed that any one of $X_1, X_2, \ldots, X_{3t}$ has $k-2$ variables, any one of $X_{3t+1}, \ldots, X_{4t}$ has $k-1$ variables and X_{4t+1} includes the rest. When m is appropriately chosen, the partition is feasible. Now assume X_{4t+1} contains r variables.

Step 2 For each $1 \leq i \leq t$, we will construct a formula H_i using the variable sets $Y_i, X_i, X_{t+i}, X_{2t+i}$ and X_{3t+i}.

For simplicity, let $Y_i = \{y_1, \ldots, y_{k-1}\}$, $X_i = \{x_1, \ldots, x_{k-2}\}$, $X_{t+i} = \{x_{t+1}, \ldots, x_{t+k-2}\}$, $X_{2t+i} = \{x_{2t+1}, \ldots, x_{2t+k-2}\}$ and $X_{3t+i} = \{x_{3t+1}, \ldots, x_{3t+k-1}\}$. For each $1 \leq i \leq t$, we introduce a new boolean variable set $Z_i = \{z_{1,0}, z_{1,1}, z_{2,0}, z_{2,1}, \ldots, z_{k-1,0}, z_{k-1,1}\}$ which does not occur in Ψ and perform the following steps to construct H_i.

(i) Let $z_{j,0}$ replace any one of positive occurrences of y_j, and $\neg z_{j,1}$ replace any one of negative occurrences of y_j in Ψ, for $j = 1, 2, \ldots, k-1$. If y_j does not occur as a positive literal, then we let $z_{j,0}$ replace one of other negative occurrences of y_j in Ψ and flip all occurrences of $z_{j,0}$ in the following formulas $H_{i1}, H_{i2}, H_{i3}, H_{i4}$. If y_j does not occur as a negative literal, then we perform similar operations.

(ii) Let

$$\begin{aligned}
H_{i1} &= \wedge_{j=1}^{k-1}(y_j \vee \neg z_{j,0} \vee \neg x_1 \vee \ldots \vee \neg x_{k-2}),\\
H_{i2} &= \wedge_{j=1}^{k-1}(z_{j,0} \vee \neg z_{j,1} \vee \neg x_{t+1} \vee \ldots \vee \neg x_{t+k-2}),\\
H_{i3} &= \wedge_{j=1}^{k-1}(z_{j,1} \vee \neg y_j \vee \neg x_{2t+1} \vee \ldots \vee \neg x_{2t+k-2}),\\
H_{i4} &= \wedge_{j=1}^{k-1}(z_{j,1} \vee \neg x_{3t+1} \vee \ldots \vee \neg x_{3t+k-1}).
\end{aligned}$$

Define $H_i = H_{i1} \wedge H_{i2} \wedge H_{i3} \wedge H_{i4}$. The new formula with all substitutions performed on Ψ is denoted as Ψ_1.

Step 3 We will make up the gap of the number of occurrences of every variable. Using the variables in sets X and $Z = \{Z_i, i = 1, 2, \ldots, t\}$, we construct a formula Ψ_2 that satisfies the following conditions.

(i) For $i = 1, \ldots, t$, each $z_{j,0}, j = 1, 2, \ldots, k-1$ in the variable set Z_i occurs in exactly $s-3$ clause of Ψ_2 and $pos(\Psi_2, z_{j,0}) + 1 - neg(\Psi_2, z_{j,0}) = min(d, 1)$.

(ii) For $i = 1, \ldots, t$, each $z_{j,1}, j = 1, 2, \ldots, k-1$ in the variable set Z_i occurs in exactly $s-4$ clauses of Ψ_2 and $pos(\Psi_2, z_{j,1}) - neg(\Psi_2, z_{j,1}) = min(d, 1)$.

(iii) Each variable x in $X_1, X_2, \ldots, X_{4t}$ occurs in exactly $s-k$ clauses of Ψ_2,

$$pos(\Psi_2, x) = s - k \text{ for } s < 2k \text{ or } pos(\Psi_2, x) = \lceil s/2 \rceil - 1 \text{ for } s \geq 2k.$$

(iv) Each variable x in X_{4t+1} occurs in exactly $s-1$ clauses of Ψ_2 and

$$pos(\Psi_2, x) + 1 - neg(\Psi_2, x) = min(d, 1).$$

(v) Every clause of Ψ_2 must have at least one positive occurrence of any one of the variables.

Step 4 Let $\Phi = \Psi_1 \wedge (\wedge_{i=1}^{t} H_i) \wedge \Psi_2$.

Clearly, Φ is a d-regular (k, s)-CNF formula. All variables in the set X are forced to be *true*. Hence, $z_{j,1}, j = 1, 2, \ldots, k-1$ is forced to be *true* by H_{i4}. By Lemma 2, $z_{j,0}, z_{j,1}$ and y_j are forced to be the same value. Given that, every variable in Y and Z is forced to be *true*, too. Because all variables in X and Z are forced to be *true*, Ψ_2 is apparently satisfiable. Thus, it can be concluded that Φ has only forced variables and the unique solution. That is, Φ is a uniquely satisfiable d-regular (k, s)-CNF formula.

Next, we will discuss the feasibility of constructing Φ. We focus on the formula Ψ_2. For Ψ_2, the number of positive literals should be more than that of clauses.

The variable set Z generates $t(k-1)(s-3) + t(k-1)(s-4)$ literals in Ψ_2. The variable set X generate $3t(k-2)(s-k) + t(k-1)(s-k) + r(s-1)$ literals in Ψ_2. The number of clauses of Ψ_2 is

$$\#cl(\Psi_2) = \frac{t(k-1)(2s-7) + 3t(k-2)(s-k) + t(k-1)(s-k) + r(s-1)}{k}.$$

Every variable of Z generates $\lceil s/2 \rceil - 2$ positive literals in Ψ_2, and very variable of X_{4t+1} generates $\lceil s/2 \rceil - 1$ positive literals in Ψ_2. About the number of positive literals of Ψ_2, there are two situations. When $s < 2k$, the number of positive literals of Ψ_2 is

$$pos(\Psi_2) = t(k-1)(2\lceil s/2 \rceil - 4) + 3t(k-2)(s-k) + t(k-1)(s-k-1) + r(\lceil s/2 \rceil - 1).$$

When $s \geq 2k$, the number of positive literals of Ψ_2 is

$$pos(\Psi_2) = t(k-1)(2\lceil s/2 \rceil - 4) + 3t(k-2)(\lceil s/2 \rceil - 1) + t(k-1)(\lceil s/2 \rceil - 1) + r(\lceil s/2 \rceil - 1).$$

Since $k \geq 3$ and $s > k$, we get $pos(\Psi_2) > \#cl(\Psi_2)$. To construct Ψ_2, We first arrange a positive literal for every clause, then randomly arrange other literals. That is, Ψ_2 can be constructed in polynomial time.

Ψ_1 and H_i can obviously be constructed in polynomial time. Therefore, we can construct a uniquely satisfiable d-regular (k,s)-CNF formula Φ in polynomial time. □

In the previous proof, we construct a uniquely satisfiable d-regular (k,s)-CNF formula Φ by using a $(m+1)k$-forced-once d-regular (k,s)-CNF formula Ψ. m determines the number of forced variables of Ψ that only occurs once. If let m be 1 more than our demand, then Φ can preserve k forced variables that only occurs once. Therefore, we get the following lemma.

Lemma 5. *For $k \geq 3$, $s \geq f(k,d)+1$ and $(s+d)/2 > k-1$, there exists a k-forced-once d-regular (k,s)-CNF formula Ψ where every variable is forced.*

Lemma 6. *For $k \geq 7$, if a d-regular (k,s)-CNF formula is unsatisfiable then $(s+d)/2 > k-1$.*

Proof. By $13 \leq f(7) \leq 17$ in [22], if a (7,s)-CNF formula is unsatisfiable, then $s \geq 14$. That is, for any integer $d \geq 0$, if a d-regular (7,s)-CNF formula is unsatisfiable, then $s \geq 14$. It implies that for $k = 7$, if a d-regular (k,s)-CNF formula is unsatisfiable, we can obtain that $(s+d)/2 \geq 14/2 > k-1$.

By $24 \leq f(8) \leq 29$ in [22], all (8,24)-CNF formulas are satisfiable. That is, for any integer $d \geq 0$, if a d-regular (8,s)-CNF formula is unsatisfiable, then $s > 24$. As for $k = 8$, if a d-regular (k,s)-CNF formula is unsatisfiable, we get $(s+d)/2 > k-1$ again.

Using Lemma 1, all $(8+r, 24+3\times r)$-CNF formulas are satisfiable for any nonnegative integer r. That is to say, if a $(8+r,s)$-CNF formula is unsatisfiable, then $s > 24+3\times r$ for any nonnegative integer r. For $(24+3\times r)/2 > 8+r-1$, we obtain that for $k \geq 7$ if a (k,s)-CNF formula is unsatisfiable, then $(s+d)/2 > k-1$. □

Theorem 3. *For all $k \geq 7$ and $s \geq f(k,d)+1$, there exist uniquely satisfiable d-regular (k,s)-CNF formulas.*

Proof. By the definition of $f(k,d)$, if $k \geq 7$ and $s \geq f(k,d)+1$, there exists an unsatisfiable d-regular (k,s)-CNF formula. Using Lemma 6, we get $(s+d)/2 > k-1$. By Theorem 2, we obtain that there exist uniquely satisfiable d-regular (k,s)-CNF formulas. □

By Theorem 3, for $k \geq 7$, we get $u(k,d) \leq f(k,d)+1$. Matthews in [20] showed that $f(k) \leq u(k) \leq f(k)+2$. Using Theorem 2, it is easy to achieve $f(k) \leq u(k) \leq f(k)+1$.

Theorem 4. *For all $k \geq 3$, $f(k) \leq u(k) \leq f(k)+1$.*

Proof. Let d be a infinite integer. That is, any one of (k,s)-CNF formulas is a d-regular (k,s)-CNF formula and any one of d-regular (k,s)-CNF formulas is a (k,s)-CNF formula. It holds that $f(k) = f(k,d)$ and $(s+d)/2 > k-1$. Using Theorem 2, for a infinite integer d, $k \geq 3$ and $s \geq f(k)+1$, there exists a uniquely satisfiable d-regular (k,s)-CNF formula. Obviously, a uniquely satisfiable d-regular (k,s)-CNF formula must be a uniquely satisfiable (k,s)-CNF formula. In other words, for $k \geq 3$ and $s \geq f(k)+1$, there exists a uniquely satisfiable (k,s)-CNF formula. By $f(k) \leq u(k) \leq f(k)+2$ in [20], we obtain that $k \geq 3$, $f(k) \leq u(k) \leq f(k)+1$. □

Corollary 1. *For $k \geq 7$ and $s \geq f(k,d)+1$, there exists a k-forced-once d-regular (k,s)-CNF formula Ψ that has exactly one satisfying assignment.*

Proof. The statement follows directly from Lemmas 5 and 6. □

5. A Parsimonious Polynomial Time Reduction

In [20], Matthews presented a parsimonious reduction from SAT to (k,s)-SAT for any $k \geq 3$ and $s \geq f(k)+2$. We will transform parsimoniously a k-CNF formula into a d-regular (k,s)-CNF formula.

Theorem 5. *For any constants $k \geq 3$, $s \geq f(k) + 1$ and $(s + d)/2 > k - 1$, there exists a parsimonious polynomial time reduction from k-CNF to d-regular (k, s)-CNF.*

Proof. Let Ψ be an arbitrarily k-CNF formula. It is supposed that Ψ contains m clauses. Obviously, Ψ contains mk literals $L_{1,1}, L_{1,2}, \ldots, L_{m,k}$. We will construct a d-regular (k, s)-CNF formula Ψ' that is SAT-equivalent with the formula Ψ, and they have the same number of solutions. Based on Lemma 5, we first construct a k-forced-once d-regular (k, s)-CNF formula Φ where every variable is forced. It is assumed that k forced variables that occur only once are $x_1, x_2, \ldots, x_k$.

The reduction method has five steps, which are described as follows.

Step 1 We introduce a new boolean variable set $Z = \{z_{i,j} : 1 \leq i \leq m, 1 \leq j \leq k\}$ to replace mk literals in Ψ in order to construct a new formula Ψ_1.

$$\Psi_1 = \underset{1 \leq i \leq m}{\wedge} \underset{1 \leq j \leq k}{\vee} L'_{i,j}, \;\; L'_{i,j} = \begin{cases} z_{i,j}, \; if \;\; L_{i,j} = v \\ \neg z_{i,j}, \; if \;\; L_{i,j} = \neg v \end{cases}, v \in \mathrm{var}(\Psi).$$

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

Step 2 Let $\Phi_i, 1 \leq i \leq m(k-1)$ be disjoint copies of the formula Φ with the variables $x_j, 1 \leq j \leq k$ of Φ being renamed as $x_{i,j}$ in Φ_i. All of $x_{i,j}$ are renumbered and formed a variable set $X = \{x_i, 1 \leq i \leq mk(k-1)\}$. Let $\Psi_2 = \wedge_{1 \leq i \leq m(k-1)} \Phi_i$.

Step 3 Let $\Psi_3 = \wedge_{1 \leq i \leq m, 1 \leq j \leq k} d_{i,j}$, and $d_{i,j} = z_{i,j} \vee \neg z'_{i,j} \vee_{l=1}^{k-2} \neg x_{((i-1)m-j-1)(k-2)+l}$. Here $z_{i,j}, z'_{i,j} \in Z$ and if $z_{i,j}$ replaces a variable v in Ψ, then $z'_{i,j}$ will point to the next variable in Z that replaces v (if $z_{i,j}$ is the last variable in Z that replaces v, then $z'_{i,j}$ will point to the first variable in Z that replaces v). The variables in Z are sorted by their subscripts.

Step 4 We construct a k-CNF formula Ψ_4 with two variable sets X and Z, satisfying the following conditions.

(i) Every variable $z_{i,j}$ of the variable set Z occurs in exactly $s - 3$ clauses of Ψ_4, and if $z_{i,j}$ occurs negatively in Ψ_1,

$$pos(\Psi_4, z_{i,j}) - neg(\Psi_4, z_{i,j}) = min(d, 1).$$

Otherwise

$$neg(\Psi_4, z_{i,j}) - pos(\Psi_4, z_{i,j}) = min(d, 1).$$

(ii) For $1 \leq i \leq mk(k-2)$, every variable x_i of X occurs in exactly $s - 2$ clauses of the formula Ψ_4, and

$$pos(\Psi_4, x_i) - neg(\Psi_4, x_i) = min(d, 1).$$

(iii) For $mk(k-2) + 1 \leq i \leq mk(k-1)$, every variable x_i of the variable set X occurs in exactly $s - 1$ clauses of the formula Ψ_4, and

$$pos(\Psi_4, x_i) + 1 - neg(\Psi_4, x_i) = min(d, 1).$$

(iv) Every clause of Ψ_4 must have at least one positive occurrence of any one of the variable set X.

Step 5 We construct the formula $\Psi' = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4\}$.

Obviously, every variable of Ψ' occurs in exactly s clauses, and the absolute value of the difference between positive and negative occurrences of every variable of Ψ' is at most d. Therefore, Ψ' is a d-regular (k, s)-CNF formula. Next, we will evaluate the feasibility of Ψ', SAT-equivalent with Ψ' and Ψ, the parsimony of the reduction.

First, we focus on the feasibility of Ψ'. The formulas Ψ_1, Ψ_2, Ψ_3 apparently can be constructed in polynomial time. With respect to the formula Ψ_4, we need to consider the condition iv. That is to say, the number of positive occurrences of X in Ψ_4 should be more than the number of clauses of Ψ_4.

The variables of Ψ_4 consists of two parts: X and Z. The variable set X generates $mk(k-2)(s-3)+mk(s-1)$ literals in Ψ_4. The variable set Z generates $mk(s-3)$ literals in Ψ_4. The number of clauses of Ψ_4

$$\begin{aligned}\#cl(\Psi) &= \frac{mk(s-3)+mk(k-2)(s-2)+mk(s-1)}{k} \\ &= m(s-3)+m(k-2)(s-2)+m(s-1) \\ &= m(k-2)(s-2)+m(2s-4).\end{aligned}$$

For $1 \leq i \leq mk(k-2)$, $pos(\Psi_4, x_i) - neg(\Psi_4, x_i) = min(d,1)$ and $pos(\Psi_4, x_i) + neg(\Psi_4, x_i) = s-2$. So,

$$pos(\Psi_4, x_i) = \lceil (s-2)/2 \rceil = \lceil s/2 \rceil - 1.$$

For $mk(k-2)+1 \leq i \leq mk(k-1)$, $pos(\Psi_4, x_i) + 1 - neg(\Psi_4, x_i) = min(d,1)$ and $pos(\Psi_4, x_i) + neg(\Psi_4, x_i) = s-1$. So, $pos(\Psi_4, x_i) + 1 = \lceil s/2 \rceil$, $pos(\Psi_4, x_i) = \lceil s/2 \rceil - 1$. The number of positive occurrences of X in Ψ_4

$$pos(\Psi_4, X) = mk(k-2)(\lceil s/2 \rceil - 1) + mk(\lceil s/2 \rceil - 1).$$

For $k \geq 3$ and $s > k$, we get

$$\begin{aligned}pos(\Psi_4, X) &> 3m(k-2)(\lceil s/2 \rceil - 1) + 3m(\lceil s/2 \rceil - 1) \\ &> m(k-2)(s-2) + m(k-2)(\lceil s/2 \rceil - 1) + 3m(\lceil s/2 \rceil - 1) \\ &> m(k-2)(s-2) + 4m(\lceil s/2 \rceil - 1) \\ &> m(k-2)(s-2) + m(2s-4) = \#cl(\Psi_4).\end{aligned}$$

Obviously, the number of positive literals of X is more than the number of clauses in Ψ_4. To construct Ψ_4, We first arrange a positive literal for every clause, then randomly arrange other literals. That is, the formula Ψ_4 can be constructed in polynomial time.

Second, we will prove that the formula Ψ is satisfiable if and only if the formula Ψ' is satisfiable.

It is assumed that Ψ is satisfied by a truth assignment τ on $var(\Psi)$ and Φ_i is satisfied by a truth assignment τ_i on $var(\Phi_i)$ for $1 \leq i \leq m(k-1)$. Because Φ_i forces the variable $x_{i,j}$ to be *true*, $\tau_i(x_{i,j})$ must be *true*. A truth assignment τ' is defined by

$$\tau'(v) = \begin{cases} \tau(z), & if\ v \in var(\Psi_1)\ and\ a\ variable\ z\ of\ var\ \Psi\ is\ replaced\ with\ v \\ \tau_i(v), & if\ v \in var(\Phi_i) \end{cases}.$$

Obvious, the truth assignment τ' can satisfy these formulas Ψ_1, Ψ_2, Ψ_3. Every clause of Ψ_4 must have at least one positive occurrence of any one of X. As a result, τ' also can satisfy the formula Ψ_4. The formula Ψ' is a conjunction of $\Psi_1, \Psi_2, \Psi_3, \Psi_4$. Thus, τ' can satisfy the formula Ψ' certainly.

It is assumed that Ψ' is satisfied by a truth assignment τ over $var(\Psi')$. Obviously, the truth assignment τ can satisfy these formulas $\Psi_1, \Psi_2, \Psi_3, \Psi_4$. For $\Psi_2 = \wedge_{1 \leq i \leq m(k-1)} \Phi_i$, the truth assignment τ can satisfy these formulas $\Phi_i, 1 \leq i \leq m(k-1)$. Because Φ_i forces the variable $x_{i,j}$ to be *true*,

$$\tau(\neg x_{i,j}) = false, 1 \leq i \leq mk, 1 \leq j \leq k-2. \quad (1)$$

We substitute Equation (1) into Ψ_3, and simplify Ψ_3. The simplified Ψ_3 contains some similar structure that are mentioned in Lemma 2. According to Lemma 2, if z_i and z_j replace the same variable of Ψ, $\tau(z_i) = \tau(z_j)$. Therefore, we define a truth assignment τ' on $var(\Psi)$ by

$$\tau'(v) = \tau(z), \text{ if a variable } v \text{ of } \Psi \text{ is replaced with a variable } z \text{ in } \Psi_1.$$

Obviously, the truth assignment τ' can satisfy the formula Ψ, and the formula Ψ is satisfiable.

Therefore, Ψ' is SAT-equivalent with Ψ.

Finally, we will explain why the polynomial-time reduction is parsimonious. If Ψ' is satisfiable, all variables in X are forced to be *true*. Due to the formula Ψ_3, all variables of Z that replaced the same variable of Ψ are forced to be the same value in every satisfying assignment. Thus, the number of satisfying assignments cannot be changed by introducing new variable set Z. Due to only one solution of Φ, Ψ_2 must not influence the number of satisfying assignments. Therefore, Ψ has as many satisfying assignments as the formula Ψ'. □

6. Conclusions

For $k \geq 3$, k-SAT problem is a NP-complete problem. From Theorem 5, it demonstrates that there exists a polynomial time reduction from k-SAT to d-regular (k, s)-SAT for any constants $k \geq 3$, $s \geq f(k) + 1$ and $(s + d)/2 > k - 1$. That is to say, d-regular (k, s)-SAT problem is NP-complete in this case. For example, the 2-regular (3,4)-SAT problem is NP-complete. In other words, there exists a parsimonious polynomial time reduction from 3-CNF to 2-regular (3,4)-CNF. Although the parsimonious reduction does not increase the number of solutions, it adds numerous new variables to the original formula. That is, The new formula has bigger solution space than the original formula. It seems that the parsimonious reduction diluted these solutions and make the SAT problem harder to solve. This explains why a random regular (3,4)-SAT instance is satisfiable with high probability and can be easily solved, but the 2-regular (3,4)-SAT problem is NP-complete.

From Lemma 6, this suggests that for $k \geq 7$ and $(s + d)/2 < k$, all d-regular (k, s)-CNF formulas are satisfiable. Consider a regular (k, s)-CNF formula F in which the positive and negative occurrences number of every variable do not exceed $k - 1$. Obviously, $s \leq 2k - 2$ and the formula F is a $(2k - s - 2)$-regular (k, s)-CNF formula. For $(s + d)/2 = (s + 2k - s - 2)/2 = k - 1 < k$, we obtain that the formula F must be satisfiable for $k \geq 7$. That is, all regular (k, s)-CNF formulas in which the positive and negative occurrences number of every variable are less than k, can be satisfiable for $k \geq 7$. However, it is unknown whether this phenomenon exists for $k < 7$.

We present the construction method of a uniquely satisfiable d-regular (k, s)-formula. Uniquely satisfiable d-regular (k, s)-SAT instances have their own characteristics. How to use uniquely satisfiable SAT instances to evaluate, analyze and improve some SAT solvers will be considered in the future.

Author Contributions: Formal analysis, Z.F.; Investigation, Z.F. and D.X.; Methodology, Z.F. and D.X.; Writing—Original Draft, Z.F.; Writing—Review & Editing, Z.F. and D.X. 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 No.61762019,61862051.

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

References

1. Cook, S.A. The complexity of theorem-proving procedures. In Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights, OH, USA, 3-5 May 1971; pp. 151–158. [CrossRef]
2. Eén, N.; Sorenssön, N. An Extensible SAT-solver. In *Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2003. [CrossRef]
3. Audemard, G.; Simon, L. GLUCOSE2.1: Aggressive-but Reactive-Clause Database Management, Dynamic Restarts. In Proceedings of the International Workshop of Pragmatics of SAT, Trento, Italy, 16 June 2012.
4. Luo, M.; Minli, M.; Xiao, F.; Manyá, F.; Zhipeng, L. An Effective Learnt Clause Minimization Approach for CDCL SAT Solvers. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017. [CrossRef]
5. 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]
6. Tovey, C.A. A simplified NP-complete satisfiability problem. *Discret. Appl. Math.* **1984**, *8*, 85–89. [CrossRef]

7. Daoyun, X.; Xiaofeng, W. A Regular NP-Complete Problem and Its Inapproximability. *J. Front. Comput. Sci. Technol.* **2013**, *7*, 691–697. [CrossRef]
8. Crawford, J.M.; Auton, L.D. Experimental Results on the Crossover Point in Satisfiability Problems. *Artif. Intell.* **1996**, *81*, 31–57. [CrossRef]
9. Kirkpatrick, S.; Selman, B. Critical behavior in the satisfiability of random boolean expressions. *Science* **1994**, *264*, 1297–1301. [CrossRef]
10. Jincheng, Z.; Daoyun, X.; Youjun, L. Satisfiability Threshold of the Regular Random (k,r)-SAT Problem. *J. Softw.* **2016**, *27*, 2985–2993. [CrossRef]
11. Jincheng, Z.; Daoyun, X.; Youjun, L. Satisfiability threshold of regular (k,r)-SAT problem via 1RSB theory. *J. Huazhong Univ. Sci. Technol.* **2017**, *45*, 7–13. [CrossRef]
12. Mézard, M.; Parisi, G.; Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science* **2002**, *297*, 815–814. [CrossRef]
13. Wahlström, M. Faster exact solving of SAT formulae with a low number of occurrences per variable. In *Theory and Applications of Satisfiability Testing (SAT-2005)*; Springer: Berlin/Heidelberg, Germany, 2005. [CrossRef]
14. Wahlström, M. An algorithm for the SAT problem for formulae of linear length. In *European Conference on Algorithms*; Springer: Berlin/Heidelberg, Germany, 2005. [CrossRef]
15. Johannsen, D.; Razgon, I.; Wahlström, M. Solving SAT for CNF Formulas with a One-Sided Restriction on Variable Occurrences. In *Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
16. Fu, Z.; Xu, D. The NP-completeness of d-regular (k,s)-SAT problem. *J. Softw.* **2020**, *31*, 1113–1123. [CrossRef]
17. Fu, Z.; Xu, D. (1,0)-Super Solutions of (k,s)-CNF Formula. *Entropy* **2020**, *22*, 253. [CrossRef]
18. Valiant, L.; Vazirani, V. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.* **1986**, *47*, 85–93. [CrossRef]
19. Calabro, C.; Impagliazzo, R.; Kabanets, V.; Paturi, R. The complexity of unique k-SAT: An isolation lemma for k-CNFs. *Comput. Syst. Sci.* **2008**, *74*, 386–393. [CrossRef]
20. Matthews, W.; Paturi, R. Uniquely Satisfiable k-SAT Instances with Almost Minimal Occurrences of Each Variable. In *Theory and Applications of Satisfiability Testing*; Springer: Berlin/Heidelberg, Germany, 2010. [CrossRef]
21. Kratochvíl, J.; Savický, P.; Tuza, Z. One more occurrence of variables makes satisfiability jump from trivial to NP-complete. *Acta Inform.* **1993**, *22*, 203–210. [CrossRef]
22. Hoory, S.; Szeider, S. Computing unsatisfiable k-SAT instances with few occurrences per variable. *Theor. Comput. Sci.* **2004**, *337*, 347–359. [CrossRef]
23. Hoory, S.; Szeider, S. Families of unsatisfiable k-CNF formulas with few occurrences per variable. *SIAM J. Discret. Math.* **2006**, *20*, 523–528. [CrossRef]
24. Savický, P.; Sgall, J. DNF tautologies with a limited number of occurrences of every variable. *Theor. Comput. Sci.* **2007**, *238*, 495–498. [CrossRef]
25. Gebauer, H.; Szabo, T.; Tardos, G. The Local Lemma is asymptotically tight for SAT. *ACM* **2016**, *63*, 664–674. [CrossRef]
26. Markström, K. Locality and Hard SAT-Instances. *J. Satisf. Boolean Modeling Comput.* **2006**, *2*, 221–227. [CrossRef]
27. Giráldez-cru, J.; Levy, J. Generating SAT instances with community structure. *Artif. Intell.* **2016**, *238*, 119–134. [CrossRef]
28. Giráldez-cru, J.; Levy, J. Locality in Random SAT Instances. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017. [CrossRef]
29. Clark, D.; Frank, J.; Gent, I.; MacIntyre, E.; Tomov, N.; Walsh, T. Local search and the number of solutions. In *The Principles and Practices of Contraint Programming (CP96)*; Springer, Berlin/Heidelberg, Germany, 1996. [CrossRef]
30. Singer, J.; Gent, I.P.; Smaill, A. Backbone fragility and the local search cost peak. *J. Artif. Intell. Res.* **2000**, *12*, 235–270. [CrossRef]

31. Znidaric, M. Single-solution Random 3-SAT Instances. *arXiv* **2005**, arXiv:cs/0504101.
32. Dubois, O. On the r,s-SAT satisfiability problem and a conjecture of Tovey. *Discret. Appl. Math.* **1990**, *26*, 51–60. [CrossRef]