Uniquely Satisfiable k-SAT Instances with Almost Minimal Occurrences of Each Variable

William Matthews and Ramamohan Paturi[*]

University of California, San Diego

Abstract. Let (k, s)-SAT refer the family of satisfiability problems restricted to CNF formulas with exactly k distinct literals per clause and at most s occurrences of each variable. Kratochvíl, Savický and Tuza [6] show that there exists a function $f(k)$ such that for all $s \leq f(k)$, all (k, s)-SAT instances are satisfiable whereas for $k \geq 3$ and $s > f(k)$, (k, s)-SAT is NP-complete. We define a new function $u(k)$ as the minimum s such that uniquely satisfiable (k, s)-SAT formulas exist. We show that for $k \geq 3$, unique solutions and NP-hardness occur at almost the same value of s: $f(k) \leq u(k) \leq f(k) + 2$.

We also give a parsimonious reduction from SAT to (k, s)-SAT for any $k \geq 3$ and $s \geq f(k) + 2$. When combined with the Valiant–Vazirani Theorem [8], this gives a randomized polynomial time reduction from SAT to UNIQUE-(k, s)-SAT.

1 Introduction

Let (k, s)-SAT refer to the family of satisfiability problems restricted to CNF formulas with exactly k distinct literals per clause and at most s occurrences of each variable. Since $(2, s)$-SAT is in P for all s, we restrict our attention to $k \geq 3$.

Tovey [7] first observed that $(3, 3)$-SAT was trivial since every instance is satisfiable, and showed that $(3, 4)$-SAT was NP-hard. This was generalized to larger k by Kratochvíl, Savický and Tuza [6] who showed that for each $k \geq 4$ there exists a threshold $f(k)$ such that for all $s \leq f(k)$, (k, s)-SAT is trivial whereas for all $s > f(k)$, (k, s)-SAT is NP-hard.

Using Hall's Theorem, Tovey [7] showed that every (k, k)-SAT instance is satisfiable, giving the first lower bound $f(k) \geq k$. This was improved by Kratochvíl, Savický and Tuza [6] who used the Lovász local lemma to show that all $(k, \lfloor 2^k/ek \rfloor)$-SAT instances are satisfied by random assignments with positive probability, implying $f(k) \geq \lfloor 2^k/ek \rfloor$.

Trivially, $f(k) < 2^k$ since enumerating all 2^k possible clauses for k variables gives an unsatisfiable formula. Kratochvíl, Savický and Tuza [6] proved that $f(k + 1) \leq 2f(k) + 1$. Combined with the fact that $f(3) = 3$, we get $f(k)$

[*] This research is supported by NSF grant CCF-0947262 from the Division of Computing and Communication Foundations. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

O. Strichman and S. Szeider (Eds.): SAT 2010, LNCS 6175, pp. 369–374, 2010.

$\leq 2^{k-1} - 1$ (this may be improved slightly by using a base case of $f(k)$ for larger k). Subsequently, this has been improved to $f(k) = \Theta\left(2^k/k\right)$ [4]. However the exact value of $f(k)$, or even whether $f(k)$ is computable, remains unknown.

Valiant and Vazirani [8] showed that deciding whether a SAT formula has zero or one solution is essentially as hard as SAT in general. In particular, they prove the following theorem:

Theorem 1 (Valiant–Vazirani Theorem [8]). *There exists a randomized polynomial time reduction from SAT to UNIQUE-SAT.*

By the standard parsimonious reduction from SAT to k-SAT, the Valiant–Vazirani Theorem implies the same hardness for UNIQUE-k-SAT. However, what happens when the number of occurrences of each variable is also limited? Specifically, what can be said about UNIQUE-(k, s)-SAT for various values of s?

We give a parsimonious reduction from 3-SAT to (k, s)-SAT, for any $k \geq 3$ and $s \geq f(k) + 2$. Thus, UNIQUE-$(k, f(k) + 2)$-SAT is as hard as UNIQUE-SAT. In contrast, UNIQUE-$(k, f(k))$-SAT is trivial since every formula is satisfiable.

Calabro et al. [3] give additional evidence that UNIQUE-k-SAT is no easier than k-SAT, not just for polynomial time algorithms (as shown by Valiant and Vazirani), but for super-polynomial time algorithms. They show that if UNIQUE-3-SAT is in randomized subexponential time ($\bigcap_{\epsilon>0}\mathrm{RTIME}(2^{\epsilon n})$), then so is k-SAT for all $k \geq 3$. Our parsimonious reduction from 3-SAT to (k, s)-SAT combined with their result implies that if UNIQUE-(k, s)-SAT is in randomized subexponential time for some $k \geq 3$ and $s \geq f(k) + 2$, then so is k'-SAT for all $k' \geq 3$. We omit the details which follow fairly straightforwardly from [3,5].

A key component in our reduction is a construction of uniquely satisfiable $(k, s + 1)$-SAT formulas from unsatisfiable (k, s)-SAT formulas. Starting with unsatisfiable $(k, f(k) + 1)$-SAT formulas allows us work with uniquely satisfiable formulas with almost the minimum number of occurrences of each variable, and also argue about the transition where uniquely satisfiable formulas first occur. Since the smallest s we argue about for each k is $f(k) + 2$, the questions of whether there exists a uniquely satisfiable $(k, f(k) + 1)$-SAT formula and the complexity of UNIQUE-$(k, f(k) + 1)$-SAT remain open.

Since our reduction requires the existence of an unsatisfiable (k, s)-SAT formula, we require that k and $s > f(k)$ be constants. In this case we know there exists an unsatisfiable formula of constant size. If we could give an upper bound on the size of this formula in terms of k and s it would imply that $f(k)$ was computable, which would be an independently interesting result.

Let $F_k(n, m)$ refer to a random k-SAT formula with n variables and m clauses. Just as there is a transition in (k, s)-SAT as s increases from trivial to NP-hard, there is a similar transition in $F_k(n, rn)$ as r increases from satisfiable with high probability to unsatisfiable w.h.p. (See [1] and its references). It is conjectured that for each k the transition occurs at a sharp threshold r_k. Achiloptas and Ricci-Tersenghi [2] show that for sufficiently large k and $r < r_k$, w.h.p., $F_k(n, rn)$ has exponentially many, widely separated, small clusters of solutions. In some ways, small, widely separated clusters of solutions are similar to unique solutions. In both cases, they seem to be some of the hardest instances for algorithms. While we don't consider random SAT formulas in this paper, we view the similarities as additional motivation.

2 Definitions and Results

Definition 1. *We let SAT refer to the satisfiability problem restricted to formulas in conjunctive normal form; k-SAT to SAT restricted to formulas with* exactly *k literals per clause; and (k, s)-SAT to k-SAT restricted to formulas where each variable occurs in at most s clauses.*

Definition 2. *We let UNIQUE-SAT refer to the promise problem of deciding whether a SAT formula is unsatisfiable or has a unique satisfying assignment. UNIQUE-k-SAT and UNIQUE-(k, s)-SAT are defined similarly.*

Definition 3 (Valiant and Vazirani [8]). *A* randomized polynomial time reduction *M from a problem A to a problem B is a randomized polynomial time Turing machine such that for all $x \notin A$ we are guaranteed that $M(x) \notin B$, and for all $x \in A$ we get $M(x) \in B$ with probability at least $1/poly(|x|)$.*

Definition 4. *In the context of SAT, a reduction M is said to be* parsimonious *if the formulas x and $M(x)$ have the same number of satisfying assignments.*

In particular, parsimonious reductions preserve the existence of unique satisfying assignments.

Definition 5 (Kratochvíl, Savický and Tuza [6]). *For each $k \geq 3$, $f(k)$ is defined as the largest value of s such that all (k, s)-SAT instances are satisfiable.*

Equivalently, we may think of $f(k) + 1$ as the smallest value of s such that there exist unsatisfiable (k, s)-SAT instances.

Definition 6. *For each $k \geq 3$, we define $u(k)$ as the smallest value of s such that there exist (k, s)-SAT instances with exactly one satisfying assignment.*

Theorem 2. *For all $k \geq 3$, $f(k) \leq u(k) \leq f(k) + 2$.*

This theorem follows directly from the following two lemmas:

Lemma 1. *For all $k \geq 3$ and $s \geq u(k)$, there exist unsatisfiable $(k, s + 1)$-SAT instances.*

Lemma 2. *For all $k \geq 3$ and $s \geq f(k) + 1$, there exist uniquely satisfiable $(k, s + 1)$-SAT instances.*

To prove that $(k, f(k) + 1)$-SAT is NP-hard, Kratochvíl, Savický and Tuza [6] give a reduction from k-SAT to (k, s)-SAT for any $s > f(k)$. Combining their proof with Lemma 2 we get the following lemma:

Lemma 3. *For any constants $k \geq 3$ and $s \geq f(k) + 2$, there is a* parsimonious *polynomial time reduction from 3-SAT to (k, s)-SAT.*

Composing the Valiant–Vazirani Theorem (Theorem 1), the standard parsimonious reduction from SAT to 3-SAT, and Lemma 3, we get the following:

Corollary 1. *For any constants $k \geq 3$ and $s \geq f(k) + 2$, there is a randomized polynomial time reduction from SAT to UNIQUE-(k, s)-SAT.*

3 Proofs

Proof (Lemma 1). Since $s \geq u(k)$, there exists a uniquely satisfiable (k, s)-SAT formula F. Add a single clause to F which is violated by the unique satisfying assignment. We add at most 1 occurrence of each variable, so this gives an unsatisfiable $(k, s + 1)$-SAT formula. □

To prove Lemma 2, we will construct a uniquely satisfiable $(k, s + 1)$-SAT formula in a sequence of steps from an unsatisfiable (k, s)-SAT formula. We classify variables in each of these formulas as either *forced* or *unforced.* If every satisfying assignment for a formula sets a variable to the same value, we say that the variable is forced. Otherwise, we say that the variable is unforced. We will be particularly interested in forced variables that occur exactly once in the formula. Without loss of generality, we will always assume that forced variable must be set to false in all satisfying assignments (otherwise replace every occurrence of the variable with its negation). Note that uniquely satisfiable formulas are equivalent to formulas where every variable is forced.

Our construction can be broken down into 3 steps formalized by the following lemmas: Lemma 4 constructs a formula with a few forced variables. Lemma 5 increases the number of forced variables without increasing the number of unforced variables. Lemma 6 uses the newly created forced variables to force all of the unforced variables.

Lemma 4. *We can transform an unsatisfiable (k, s)-SAT formula into a satisfiable (k, s)-SAT formula with k forced variables that only occur once.*

Lemma 5. *We can transform a (k, s)-SAT formula with n unforced variables and $t \geq k - 1$ forced variables that only occur once (and possibly other variables that are forced but occur more than once) into a (k, s)-SAT formula with n unforced variables and $t + (s - k) > t$ forced variables that only occur once.*

Lemma 6. *We can transform a (k, s)-SAT formula with n unforced variables and at least $n + k$ forced variables that only occur once into a $(k, s + 1)$-SAT formula where every variable is forced.*

Proof (Lemma 4). Let F be a minimal unsatisfiable (k, s)-SAT formula. (A formula is minimally unsatisfiable if removing any clause would make it satisfiable.) Transform F by renaming variables and replacing variables with their negations so that F can be written as $(x_1 \vee x_2 \vee \cdots \vee x_k) \wedge G$, where G is satisfied by the all-false assignment. Within G, the variables $x_1, \ldots, x_k$ each occur at most $s - 1$

times and are forced to false (any satisfying assignment that didn't set them to false would also satisfy F).

Let $G^{(1)}, \ldots, G^{(k-1)}$ be $k-1$ disjoint copies of G. Let $x_1^{(i)}, \ldots, x_k^{(i)}$ denote the copy of $x_1, \ldots, x_k$ occurring in $G^{(i)}$. Return the formula $G^{(1)} \wedge \cdots \wedge G^{(k-1)} \wedge H$, where $H = \bigwedge_{i=1}^{k}(x_i^{(1)} \vee \cdots \vee x_i^{(k-1)} \vee \overline{y_i})$ and $y_1, \ldots, y_k$ are fresh variables. Each variable y_i occurs in exactly 1 clause and must be set to false to satisfy that clause since all of the other variables in the clause are already forced. □

Proof (Lemma 5). Let G be a (k,s)-SAT formula with n unforced variables and $t \geq k-1$ forced variables that only occur once. Let $y_1, \ldots, y_{k-1}$ denote $k-1$ of these forced variables. Let $H = \bigwedge_{i=1}^{s-1}(y_1 \vee \cdots \vee y_{k-1} \vee \overline{z_i})$, where $z_1, \ldots z_{s-1}$ are fresh variables. Return the formula $G \wedge H$. Each of the variables $y_1, \ldots, y_{k-1}$ is still forced, but now each occurs s times. In their place, we have $s-1$ new forced variables $z_1, \ldots, z_{s-1}$ which each only occur once, for a total of $t+(s-k)$ such variables. Whether other variables are forced remains unchanged. Note that $s > f(k) \geq k$ since unsatisfiable (k,s)-SAT instances exist. □

Proof (Lemma 6). Let F be a (k,s)-SAT formula with n unforced variables and $n+k$ forced variables that only occur once. Let $x_1, \ldots, x_n$ denote the unforced variables. Let $y_1, \ldots, y_{n+k}$ denote the forced variables that only occur once. Let $m = \lceil \frac{n}{k-1} \rceil$. Arbitrarily partition the variables $x_1, \ldots, x_n$ into m sets $X_1, \ldots, X_m$ of size $k-1$. Add new variables as needed so that every set contains exactly $k-1$ variables. Arbitrarily partition the variables $y_1, \ldots, y_{(k-1)m}$ into m sets $Y_1, \ldots, Y_m$ of size $k-1$.

For each $1 \leq i \leq m$, we will construct a formula H_i using the variables in sets X_i and Y_i. For simplicity, let $X_i = \{x_1, \ldots, x_{k-1}\}$ and $Y_i = \{y_1, \ldots, y_{k-1}\}$. For each i, let $H_i = \bigwedge_{j=1}^{k-1}(y_1 \vee \cdots \vee y_{k-1} \vee \overline{x_j})$.

Return the formula $F \wedge H_1 \wedge \cdots \wedge H_m$. Each H_i uses the variables in Y_i to force the variables in X_i. Since each variable in Y_i is forced, the variables in X_i must be false to satisfy the clauses in H_i. This adds $k-1$ occurrences for each variable in Y_i and one occurrence for each variable in X_i. Each variable in Y_i now occurs $k < s$ times and each variable in X_i now occurs at most $s+1$ times. □

Proof (Lemma 2). Since $s \geq f(k)+1$, there exists an unsatisfiable (k,s)-SAT formula F. Use Lemma 4 to construct a (k,s)-SAT formula G with k forced variables that only occur once. Let n denote the number of unforced variables in G. Use Lemma 5 sufficiently many times starting with G to get a formula H with at least $n+k$ forced variables that only occur once. Note that H still contains only n unforced variables. Using Lemma 6 on H gives a uniquely satisfiable $(k,s+1)$-SAT formula. □

By repeating Lemma 5 sufficiently many additional time before using Lemma 6, we get the following corollary:

Corollary 2. *For any constants $k \geq 3$ and $s \geq f(k)+2$, and any $m \geq 0$, we can construct a uniquely satisfiable (k,s)-SAT formula with at least m forced variables that only occur once in time polynomial in m.*

The following proof of Lemma 3 is the same as the reduction given by Kratochvíl, Savický and Tuza [6] to prove that $(k, f(k)+1)$-SAT is NP-hard with one exception. We use Corollary 2 to supply forced variables whereas they used a $(k, f(k)+1)$-SAT formula with potentially many satisfying assignments.

Proof (Lemma 3). For any $k \geq 3$ and $s \geq f(k)+2$, we transform a 3-SAT formula F parsimoniously into a (k, s)-SAT formula in 2 steps:

First, we reduce the number of occurrences of each variable to at most s, which introduces additional 2-variable clauses. For each variable x occurring $t > s$ times, replace each occurrence of x with a new variable x_i, $1 \leq i \leq t$. Add clauses $(\overline{x_i} \vee x_{i+1})$ for $1 \leq i \leq t-1$, and $(\overline{x_t} \vee x_1)$. These clauses ensure that in any satisfying assignment all the variables x_i are assigned the same value. Thus, we maintain exactly the same number of satisfying assignments. Each of these new variables occurs exactly $3 < s$ times. Let G denote the resulting formula, and m the number of clauses in G.

Second, we pad each clause with forced variables so that all clauses contain exactly k variables. Using Corollary 2, there exists a (k, s)-SAT formula H with at least mk forced variables that only occur once. For each clause c of length $\ell < k$ in G, replace c with $(c \vee y_1 \vee \cdots \vee y_{k-\ell})$, where $y_1, \ldots, y_{k-\ell}$ are arbitrary forced variables from H occurring fewer than s times. Let G' denote the result of these replacements. Return the formula $G' \wedge H$. Since the only satisfying assignment to H sets all variables to false, the padded clauses in G' are satisfied by exactly the same assignments that satisfy G. Thus, $G' \wedge H$ has exactly the same number of satisfying assignments as G. □

References

1. Achlioptas, D., Peres, Y.: The threshold for random k-SAT is $2^k \ln 2 - O(k)$. J. AMS 17, 947–973 (2004)
2. Achlioptas, D., Ricci-Tersenghi, F.: On the solution-space geometry of random constraint satisfaction problems. In: Kleinberg, J.M. (ed.) STOC, pp. 130–139. ACM, New York (2006)
3. Calabro, C., Impagliazzo, R., Kabanets, V., Paturi, R.: The complexity of unique k-SAT: An isolation lemma for k-CNFs. J. Comput. Syst. Sci. 74(3), 386–393 (2008)
4. Gebauer, H.: Disproof of the neighborhood conjecture with implications to SAT. In: Fiat, A., Sanders, P. (eds.) ESA 2009. LNCS, vol. 5757, pp. 764–775. Springer, Heidelberg (2009)
5. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. 63(4), 512–530 (2001)
6. Kratochvíl, J., Savický, P., Tuza, Z.: One more occurrence of variables makes satisfiability jump from trivial to NP-complete. SIAM Journal on Computing 22(1), 203–210 (1993)
7. Tovey, C.A.: A simplified NP-complete satisfiability problem. Discrete Applied Mathematics 8(1), 85–89 (1984)
8. Valiant, L.G., Vazirani, V.V.: NP is as easy as detecting unique solutions. Theor. Comput. Sci. 47(1), 85–93 (1986)