An Algorithm for SAT Above the Threshold

Hubie Chen

Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
hubes@cs.cornell.edu

Abstract. We study algorithms for finding satisfying assignments of randomly generated 3-SAT formula. In particular, we consider distributions of highly constrained formulas (that is, "above the threshold" formulas) restricted to satisfiable instances. We obtain positive algorithmic results, showing that such formulas can be solved in low exponential time.

1 Introduction

Randomly generated instances of the boolean satisfiability problem have been used to study typical-case problem complexity. Consider the model for randomly generating a 3-SAT formula over a variable set of size n which includes each of the $2^3\binom{n}{3}$ clauses independently with probability p. For very "low" p, such a formula tends not to have many clauses, and is usually satisfiable; for very "high" p, such a formula tends to have many clauses, and is likely to be unsatisfiable. A fundamental conjecture, the *satisfiability threshold conjecture*, states that there is a transition point at which such random formulas abruptly change from being almost always satisfiable to almost always unsatisfiable. Precisely, this conjecture posits the existence of a constant c such that a random formula is satisfiable almost always if $p < cn/2^3\binom{n}{3}$, that is, the expected number of clauses is less than cn; and is unsatisfiable almost always if $p > cn/2^3\binom{n}{3}$. (By *almost always*, we mean with probability tending to one as n approaches infinity.) There is ample empirical evidence for this conjecture [23], and theoretical work has provided rigorous lower and upper bounds on the value of the purported constant c [16, 6, 15, 1].

There are a number of positive algorithmic results for finding satisfying assignments of randomly generated formulas *below* the threshold – namely, results giving polynomial time algorithms which find satisfying assignments *almost always* on below-threshold formula distributions. In contrast, satisfiability algorithms for formulas *above* the threshold have seen relatively little attention. The main result of this paper is the analysis of a novel random walk algorithm for finding satisfying assignments of 3-SAT formulas from above the threshold. Of course, deciding satisfiability of such formulas is rather uninteresting on the described distribution: the trivial algorithm which outputs "unsatisfiable" is correct almost always. Hence, we restrict the described distribution of formulas to satisfiable formulas, and seek algorithms which output satisfying assignments almost always. In essence, our result shows that almost all satisfiable random 3-SAT formulas with enough clauses have a *tractable structure* that can be exploited by an algorithm to obtain a satisfying assignment. This paper thus forms a contribution to the long-term research program aiming to fully understand the typical complexity of random satisfiability formulas.

1.1 Three Formula Distributions

Before formally stating our results and comparing them to previous work, it will be useful to identify three *distinct* 3-SAT formula distributions.

Standard distribution, $P_{n,p}[\cdot]$. Formulas over a variable set of size n are generated by including each of the $2^3\binom{n}{3}$ clauses independently with probability p. Let $\mathcal{F}$ be a formula over n variables with exactly m clauses; clearly, $P_{n,p}[\mathcal{F}] = p^m(1-p)^{2^3\binom{n}{3}-m}$.

Satisfiable distribution, $P^{\mathrm{sat}}_{n,p}[\cdot]$. This is the standard distribution, but conditioned on the property of satisfiability. That is, unsatisfiable formulas have probability zero, and satisfiable formulas have probability proportional to their probability in the standard distribution. Formally, we define, for all formulas $\mathcal{F}$ over n variables, $P^{\mathrm{sat}}_{n,p}[\mathcal{F}] = P_{n,p}[\mathcal{F}|S]$, where S is the event that a formula is satisfiable.

Planted distribution, $P^{\mathrm{plant}}_{n,p}[\cdot]$. Formulas over a variable set of size n are generated by first selecting an assignment α, uniformly at random from the set of all 2^n assignments. Then, each of the $7\binom{n}{3}$ clauses under which α is true are included independently with probability p; intuitively, α is a "forced" or "planted" satisfying assignment of the resulting formula. Formally, we define, for all formulas $\mathcal{F}$ over n variables, $P^{\mathrm{plant}}_{n,p}[\mathcal{F}] = \sum_\alpha \frac{1}{2^n} P_{n,p}[\mathcal{F}|S(\alpha)]$, where the summation is over all 2^n assignments to the variables of $\mathcal{F}$, and $S(\alpha)$ is the event that α is a satisfying assignment.

The standard distribution is clearly different from each of the other two, since in the standard distribution, every formula has a non-zero probability of occurring, whereas in the other two distributions, unsatisfiable formulas have zero probability of occurring. Moreover, the satisfiable and planted distributions are different, as the planted distribution is biased towards formulas with many satisfying assignments. For instance, the empty formula containing no clauses has a $(1-p)^{7\binom{n}{3}}$ probability of occurring in the planted distribution, whereas the empty formula has a lower probability of occurring in the satisfiable distribution, namely, probability $(1-p)^{8\binom{n}{3}}/P_{n,p}[S]$. (It is easy to see that $P_{n,p}[S] > (1-p)^{\binom{n}{3}}$.)

1.2 Our results

Roughly speaking, we show that there are "low" exponential time algorithms for random formulas (from $P^{\rm sat}_{n,p}[\cdot]$ and $P^{\rm plant}_{n,p}[\cdot]$) with "high" constant clause-to-variable ratios. More precisely, our main theorem is that there is a monotonically increasing function $d : (0,1) \to (0,\infty)$ such that for any $\epsilon > 0$, our algorithm finds a satisfying assignment in time $2^{\epsilon n}$ for almost all formulas according to the distribution $P^{\rm sat}_{n,p}[\cdot]$ with $p \geq d(\epsilon)n/2^3\binom{n}{3}$ – that is, p is set so that the expected number of clauses is $d(\epsilon)n$. Put differently, for any $b > 1$, we demonstrate that there is a constant d' such that an algorithm running in time b^n finds a satisfying assignment for almost all formulas from $P^{\rm sat}_{n,p}[\cdot]$, with $p \geq d'(b)n/2^3\binom{n}{3}$. This rephrasing follows from the initial description by defining $d'(b)$ to be $d(\epsilon)$ for an $\epsilon \in (0,1)$ such that $2^\epsilon < b$. We also prove that this theorem holds for almost all formulas from $P^{\rm plant}_{n,p}[\cdot]$.

An intriguing corollary we obtain is that if $p(n)$ is set so that the expected number of clauses is super-linear – formally, if $p(n)$ is $\omega(1/n^2)$ – then, for every $b > 1$, there is an algorithm finding a satisfying assignment for almost all formulas from $P^{\rm sat}_{n,p}[\cdot]$ (as well as from $P^{\rm plant}_{n,p}[\cdot]$) in time b^n. In other words, for such a setting of $p(n)$, we obtain a distribution of formulas which can be solved almost always in time b^n – for *all* $b > 1$! The obvious question suggested by this corollary is whether or not such distributions can be solved almost always in polynomial time, and we conjecture this to be the case.

1.3 Related work

Our results are most directly comparable to the works of Koutsoupias and Papadimitriou [17], Gent [9], and Flaxman [7]. Koutsoupias and Papadimitriou proved that the "greedy" algorithm almost always finds a satisfying assignment when $p(n)$ is a sufficiently large constant times $1/n$ – that is, the expected number of clauses is a sufficiently large quadratic function of n.

Theorem 1. *[17] There exists a constant c_{KP} such that when $p(n) \geq c_{KP}/n$, the greedy algorithm finds a satisfying assignment in polynomial time, for almost all formulas drawn according to $P^{\rm sat}_{n,p}[\cdot]$.*

Their theorem also holds for $P^{\rm plant}_{n,p}[\cdot]$ in place of $P^{\rm sat}_{n,p}[\cdot]$.

Using a similar analysis, Gent showed that a simple algorithm almost always finds a satisfying assignment when the expected number of clauses is larger than a particular constant times $n \log n$. However, his result is only shown to hold for the planted distribution.

Theorem 2. *[9] There exists a constant c_G and a polynomial time algorithm A such that when $p(n) \geq c_G \log n/n^2$, the algorithm A finds a satisfying assignment for almost all formulas drawn according to $P^{\rm plant}_{n,p}[\cdot]$.*

Flaxman also studied the planted distribution, using spectral techniques to demonstrate polynomial time tractability at a sufficiently high linear clause density.

Theorem 3. *[7] There exists a constant c_F and a polynomial time algorithm A such that when $p(n) \geq c_F/n^2$, the algorithm A finds a satisfying assignment for almost all formulas drawn according to $P^{\text{plant}}_{n,p}[\cdot]$.*

In fact, Flaxman [7] studies a more general model of "planted" random formulas which includes our definition of planted distribution as a particular parameterization; we refer the reader to his paper for more details.

To review, our result is that for every $b > 1$, there exists a constant c' such that when $p(n) \geq c'/n^2$, an algorithm running in time b^n finds a satisfying assignment for almost all formulas drawn according to $P^{\text{sat}}_{n,p}[\cdot]$ (as well as $P^{\text{plant}}_{n,p}[\cdot]$). Thus, we basically lower by a factor of n the clause density required to give the result of Koutsoupias and Papadimitriou, but give an algorithm which requires exponential time. Although our algorithms require more time than those of Gent and Flaxman, our results on the satisfiable distribution are not directly comparable to their results, which concern only the planted distribution. We emphasize that Flaxman's Theorem 3 does not imply our results on the satisfiable distribution. This theorem does imply the results we give on the planted distribution; however, we still consider our proof technique of analyzing a simple random walk to be of independent interest.

We briefly discuss three other groups of results related to our study.

"Below the threshold" algorithms. For randomly generated formulas from below the threshold, polynomial time algorithms that work almost always have been given [2, 8]. It is worth noting that at the times these results were initially presented, they gave the best threshold lower bounds to date.

Worst-case algorithms. The problem of finding a satisfying assignment for a 3-SAT formula has the trivial exponential time upper bound of 2^n; a sequence of results has improved this trivial upper bound [21, 5, 22]. For instance, [22] gives a probabilistic algorithm solving 3-SAT in time 1.3302^n. That is, they exhibit an algorithm which solves *any* given 3-SAT formula in time 1.3302^n with high probability. Hirsch [13] took the slightly different route of performing a worst-case analysis of particular local search algorithms that had been shown to work well in practice.

Our results do not by any means strictly improve these results, nor are they implied by these results. These upper bounds establish specific values of $b' > 1$ such that any formula from the entire class of all 3-SAT formulas can be solved in time b'^n. In contrast, we establish that for *any* $b > 1$, there is a *particular* class of distributions that can be solved *almost* always in time b^n.

Proof complexity. Formulas from above the threshold are almost always unsatisfiable. A natural question concerning such formulas is how difficult or easy it is to certify their unsatisfiability. The lengths of unsatisfiability proofs from above-threshold formula distributions is taken up in [3, 11, 10, 12], for example; we refer the reader thither for further pointers into the current literature.

2 Preliminaries

We first note a lemma; both it and its proof are from [17].

Lemma 1. *Let M, M' be independent binomial random variables. If $m \geq \epsilon n$, then $P[M(p, \binom{n-1}{2} - \binom{m}{2}) \geq M'(p, \binom{n-1}{2})] \leq e^{-g(\epsilon)pn^2}$, where $g : (0,1) \to (0,\infty)$ is monotonically increasing.*

Proof. For any a, the probability is bounded above by $P[M(p, \binom{n-1}{2} - \binom{m}{2}) \geq a] + P[M'(p, \binom{n-1}{2}) \leq a]$. Choosing $a = p\sqrt{[\binom{n-1}{2} - \binom{m}{2}]\binom{n-1}{2}}$, this sum can be bounded above by routine computations using the Chernoff bounds $P[M(p,N) \leq (1-\theta)pn] \leq e^{-\theta^2 np/2}$ and $P[M(p,N) \geq (1+\theta)pn] \leq e^{-\theta^2 np/3}$. □

We now present the notation and definitions used throughout the paper. Let $\mathcal{F}$ denote a 3-SAT formula, and let A and α denote true/false assignments to the variable set of $\mathcal{F}$.

Let S denote the event that a formula is satisfying, and let $S(\alpha)$ denote the event that a formula is satisfied by α. A property X holds for *almost all satisfiable formulas* if $\lim_{n\to\infty} P^{\mathrm{sat}}_{n,p}[X] \to 1$. A property X holds for *almost all planted formulas* if $\lim_{n\to\infty} P^{\mathrm{plant}}_{n,p}[X] \to 1$. Define the distribution $P^{\alpha}_{n,p}[\cdot]$ so that for all formulas $\mathcal{F}$, $P^{\alpha}_{n,p}[\mathcal{F}] = P_{n,p}[\mathcal{F}|S(\alpha)]$.

Let $T(A,\alpha)$ be the set of variables v such that $A(v) = \alpha(v)$, and $F(A,\alpha)$ be the set of variables v such that $A(v) \neq \alpha(v)$. (Notice that when α is the all-true assignment $T(A,\alpha)$ is the set of variables that are true in A, and $F(A,\alpha)$ is the set of variables that are false in A.) Let $\overline{A}$ denote the complement of assignment A, so that $\overline{A}$ assigns a variable to false if and only if A assigns the variable to true. Let $B(A,d)$ denote the set of assignments within Hamming distance d of A.

The following definitions are relative to a formula $\mathcal{F}$, but we do not explicitly include $\mathcal{F}$ in the notation, as the relevant formula will be clear from context. Let $L(A,v)$ denote the set of clauses that are "lost" at assignment A if variable v is flipped, that is, the set of clauses in $\mathcal{F}$ that are true under A but false under A with v flipped. For $\beta \in (0,1)$, let $E_\beta(A)$ denote the set of variables v which have one of the βn lowest values of $|L(A,v)|$. We assume that ties are broken in an arbitrary but fixed manner, so that $E_\beta(A)$ is always of size βn. Define the set $F'_\beta(A,\alpha)$ to be $F(A,\alpha) \setminus E_\beta(A)$, and define the set $T'_\beta(A,\alpha)$ to be $T(A,\alpha) \cap E_\beta(A)$. The set $T'_\beta(A,\alpha)$ contains every variable which is in $E_\beta(A)$ but would move the assignment A further away from the assignment α, if flipped.

Let us say that a formula $\mathcal{F}$ is *β-bad relative to* α if $\beta \in (0,1)$, α is a satisfying assignment, and there exists $A \notin B(\alpha,\beta n) \cup B(\overline{\alpha},\beta n)$ such that $|T'_\beta(A,\alpha)| \geq \frac{\beta n}{2}$. Let us say that a formula $\mathcal{F}$ is *β-bad* if it is β-bad relative to every satisfying assignment α. Let us say that a formula $\mathcal{F}$ is *β-good* if it is not β-bad, that is, $\beta \in (0,1)$ and there exists a satisfying assignment α such that for all $A \notin B(\alpha,\beta n) \cup B(\overline{\alpha},\beta n)$, the inequality $|T'_\beta(A,\alpha)| < \frac{\beta n}{2}$ holds.

The next two sections constitute the technical portion of the paper. Roughly speaking, these sections will establish that the algorithm succeeds on any good formula, and that almost all satisfiable formulas are good – hence, the algorithm can handle almost all satisfiable formulas.

3 Algorithm

In this section, we present our algorithm, and show that it finds a satisfying assignment on all good formulas. The algorithm, which is parameterized by $\beta \in (0,1)$ and takes a 3-SAT formula $\mathcal{F}$ as input, is as follows:

- Pick a random assignment A
- While no assignment in $B(A, \beta n) \cup B(\overline{A}, \beta n)$ satisfies $\mathcal{F}$:
  Randomly pick a variable from $E_\beta(A)$, and flip it to obtain a new A
- Output satisfying assignment

The key property of this algorithm is given by the following theorem. Roughly, the theorem states that the property of β-goodness is sufficient to direct the algorithm to a satisfying assignment in a polynomial number of loop iterations.

Theorem 4. *Suppose that $\mathcal{F}$ is a β-good formula. Then, the expected number of flips required by the algorithm (parameterized with β) to find a satisfying assignment for $\mathcal{F}$ is $O(n^2)$.*

Proof. Consider a Markov chain with state set $\{0, \ldots, n\}$ such that when the chain is in state i (for $i \in \{1, \ldots, n-1\}$), it is only possible to make transitions to the two states $i \pm 1$. If the probability of transitioning from state i to $i-1$ is $\geq 1/2$ (for all $i \in \{1, \ldots, n-1\}$), then, from any starting point, it is known that the expected time to hit either state 0 or state n is $O(n^2)$ [19, pp. 246-247].

Since $\mathcal{F}$ is β-good, there exists a satisfying assignment α such that for all $A \notin B(\alpha, \beta n) \cup B(\overline{\alpha}, \beta n)$, $|T'_\beta(A, \alpha)| < \frac{\beta n}{2}$.

We can consider the algorithm as taking a random walk on the Markov chain described above, where its state is the Hamming distance to the assignment α. Each time the loop test fails on an assignment A, we have $\alpha \notin B(A, \beta n) \cup B(\overline{A}, \beta n)$, which implies that $A \notin B(\alpha, \beta n) \cup B(\overline{\alpha}, \beta n)$. Thus, $|T'_\beta(A, \alpha)| < \frac{\beta n}{2}$, and randomly picking a variable from $E_\beta(A)$ will result in picking a variable from $F(A, \alpha)$ at least $1/2$ of the time, and thus the Hamming distance of A to α has a probability of at least $1/2$ of decreasing. If the Hamming distance of A to α is less than βn or greater than $n - \beta n$, the loop test is true, and the satisfying assignment α is found. By the initially stated fact on the type of Markov chain considered, this will occur in expected $O(n^2)$ flips. □

4 Analysis: almost all satisfiable formulas are good

We now prove a complement to the theorem of the last section: that almost all satisfiable formulas are β-good, when $p(n) \geq h(\beta)/n^2$, for a trade-off function h that we give. This will allow us to conclude that the algorithm "works" for almost all satisfiable formulas.

We begin by showing that formulas are almost never β-bad on the planted distribution $P^{\text{plant}}_{n,p}[\cdot]$.

Lemma 2. *If $\beta \in (0,1)$, and A, α are assignments such that $A \notin B(\alpha, \beta n) \cup B(\overline{\alpha}, \beta n)$, then $P^{\alpha}_{n,p}[|T'_\beta(A,\alpha)| \geq \frac{\beta n}{2}] \leq (\frac{4}{e^{g(\beta)\beta p n^2/2}})^n$, where $g : (0,1) \to (0,\infty)$ is the same function given by Lemma 1.*

Proof. We assume without loss of generality that α is the all-true assignment. Let $k = \beta n/2$ (which is equal to $|E_\beta(A)|/2$). Since $A \notin B(\alpha, \beta n) \cup B(\overline{\alpha}, \beta n)$, $|F(A,\alpha)| \geq \beta n = 2k$, and $|T'_\beta(A,\alpha)| \geq k$ implies that $|F'(A,\alpha)| \geq k$ (with respect to any formula). This in turn implies that there exist distinct variables $t_1, \ldots, t_k$ in $T'_\beta(A,\alpha)$ and distinct variables $f_1, \ldots, f_k$ in $F'(A,\alpha)$ such that for all $i = 1, \ldots, k$, $|L(A,t_i)| \leq |L(A,f_i)|$.

Thus, we have

$$P^{\alpha}_{n,p}[|T'_\beta(A,\alpha)| \geq k] = P^{\alpha}_{n,p}[[|T'_\beta(A,\alpha)| \geq k] \wedge [|F'(A,\alpha)| \geq k]]$$

which is bounded above by

$$\leq P^{\alpha}_{n,p}[\cup(|L(A,t_1)| \leq |L(A,f_1)| \wedge \ldots \wedge |L(A,t_k)| \leq |L(A,f_k)|)]$$

where the union is over all sequences $t_1, \ldots, t_k, f_1, \ldots, f_k$ such that the t_i are distinct elements of $T'_\beta(A,\alpha)$, and the f_i are distinct elements of $F'(A,\alpha)$. Applying the so-called union bound, we see that the previous expression is

$$\leq \sum P^{\alpha}_{n,p}[|L(A,t_1)| \leq |L(A,f_1)| \wedge \ldots \wedge |L(A,t_k)| \leq |L(A,f_k)|]$$

where the summation is over the same set of sequences as the union in the previous expression.

Observe that the clause set $L(A,t_i)$ is a subset of those clauses containing t_i positively and two other variables occurring positively if and only if they are false in A. Also, the clause set $L(A,f_i)$ is a subset of those clauses containing f_i negatively and two other variables occurring positively if and only if they are false in A, but without clauses containing three negative literals (as they are not satisfied by the forced all-true assignment).

Consequently, for a variable t_i, of all the $L(A,v)$, only $L(A,t_i)$ can contain a clause with t_i appearing positively, and all clauses of $L(A,t_i)$ do; likewise, for a variable f_i, only $L(A,f_i)$ can contain a clause with f_i appearing negatively, and all clauses of $L(A,f_i)$ do. Hence, the quantities $|L(A,v)|$ are independent random variables! It follows that the previous expression is

$$= \sum P^{\alpha}_{n,p}[|L(A,t_1)| \leq |L(A,f_1)|] \cdots P^{\alpha}_{n,p}[|L(A,t_k)| \leq |L(A,f_k)|].$$

As just described, $|L(A,t_j)|$ is the sum of $\binom{n-1}{2}$ Bernoulli trials, and $|L(A,f_j)|$ is the sum of $\binom{n-1}{2} - \binom{m}{2}$ Bernoulli trials, where m is the number of variables assigned to the value true in A – that is, $|T(A,\alpha)|$. Using Lemma 1 along with the developed chain of inequalities, we can upper bound the expression of interest, $P^{\alpha}_{n,p}[|T'_\beta(A,\alpha)| \geq k]$, by $4^n[e^{-g(\beta)pn^2}]^k = [\frac{4}{e^{g(\beta)\beta p n^2/2}}]^n$. □

Lemma 3. *There exists a monotonically decreasing function* $h : (0,1) \to (0,\infty)$ *such that* $p(n) \geq h(\beta)/n^2$ *implies* $\lim_{n\to\infty} 2^n P^{\alpha}_{n,p}[\beta\text{-bad relative to } \alpha] = 0$*, for all assignments* α *on* n *variables.*

Proof. Let $\mathcal{A}$ denote the set of assignments A such that $A \notin B(\alpha, \beta n) \cup B(\overline{\alpha}, \beta n)$. We establish an upper bound on the probability of interest.

$$\begin{aligned}
2^n P^{\alpha}_{n,p}[\beta - \text{bad relative to } \alpha] &\leq 2^n P^{\alpha}_{n,p}[\cup_{A\in\mathcal{A}} |T'_{\beta}(A,\alpha)| \geq \tfrac{\beta n}{2}] \\
&\leq 2^n \textstyle\sum_{A\in\mathcal{A}} P^{\alpha}_{n,p}[|T'_{\beta}(A,\alpha)| \geq \tfrac{\beta n}{2}] \\
&\leq 2^n \textstyle\sum_{A\in\mathcal{A}} [\frac{4}{e^{g(\beta)\beta p n^2/2}}]^n \\
&\leq 2^n 2^n [\frac{4}{e^{g(\beta)\beta p n^2/2}}]^n \\
&= [\frac{16}{e^{g(\beta)\beta p n^2/2}}]^n
\end{aligned}$$

The third inequality holds by Lemma 2.

It suffices to choose h to be a sufficiently large monotonically decreasing function so that $[\frac{16}{e^{g(\beta)\beta h(\beta)/2}}] < 1$, for all $\beta \in (0,1)$. □

The next lemma connects the $P^{\text{sat}}_{n,p}[\cdot]$ distribution to the $P^{\text{plant}}_{n,p}[\cdot]$ distribution, using the prior two lemmas to show the rareness of β-bad formulas in $P^{\text{sat}}_{n,p}[\cdot]$.

Lemma 4. *There exists a monotonically decreasing function* $h : (0,1) \to (0,\infty)$ *such that* $p(n) \geq h(\beta)/n^2$ *implies* $\lim_{n\to\infty} P^{\text{sat}}_{n,p}[\beta\text{-bad}] = 0$.

Proof. Let $\mathcal{A}$ denote the set containing all 2^n assignments on n variables, and let h be as in Lemma 3. Fix $\beta \in (0,1)$, and let D denote the event that a formula is β-bad. We have the following chain of inequalities.

$$\begin{aligned}
P^{\text{sat}}_{n,p}[D] &= P_{n,p}[D \wedge S]/P_{n,p}[S] \\
&\leq \textstyle\sum_{\alpha\in\mathcal{A}} P_{n,p}[D \wedge S(\alpha)]/P_{n,p}[S] \\
&\leq \textstyle\sum_{\alpha\in\mathcal{A}} (P_{n,p}[D \wedge S(\alpha)]/P_{n,p}[S(\alpha)]) \\
&\leq \textstyle\sum_{\alpha\in\mathcal{A}} P_{n,p}[D|S(\alpha)] \\
&= \textstyle\sum_{\alpha\in\mathcal{A}} P^{\alpha}_{n,p}[D] \\
&\leq \textstyle\sum_{\alpha\in\mathcal{A}} P^{\alpha}_{n,p}[\beta\text{−bad relative to } \alpha]
\end{aligned}$$

By Lemma 3, the last expression approaches zero as n approaches infinity. □

Having established that almost all formulas are β-good, we are now in a position to combine this result with the theorem which showed that the algorithm "works" on β-good instances.

Theorem 5. *There exists a monotonically decreasing function* $h' : (0,1) \to (0,\infty)$ *such that when* $p \geq h'(\epsilon)/n^2$*, the given randomized algorithm finds a satisfying assignment for almost all satisfiable formulas in expected* $O(2^{\epsilon n})$ *time (where "expected time" is with respect to the random bits used by the algorithm).*

Proof. For all $\epsilon \in (0,1)$, define $h'(\epsilon) = h(\beta)$, where β is sufficiently small so that $\binom{n}{\beta n} \leq 2^{\frac{\epsilon n}{2}}$. (Such a value β exists, for $\binom{n}{\beta n} \leq (\frac{ne}{\beta n})^{\beta n} = (\frac{e}{\beta})^{\beta n}$; see e.g. [18, p. 434].) By Lemma 4, when $p \geq h'(\epsilon)/n^2 = h(\beta)/n^2$, $\lim_{n\to\infty} P^{\text{sat}}_{n,p}[\beta - \text{good}] \to 1$.

Thus, for such p, almost all satisfiable formulas are β-good, and by Theorem 4, the algorithm will find a satisfying assignment for almost all satisfiable formulas in $O(n^2)$ flips.

Before each flip, every assignment in $B(A, \beta n) \cup B(\overline{A}, \beta n)$ must be checked to see if it is satisfying; there are $2\binom{n}{\beta n} \leq 2 * 2^{\frac{\epsilon n}{2}}$ such assignments, and checking each takes time $t(n)$, for some polynomial t. Thus, $O(n^2)$ flips require time $2 * 2^{\frac{\epsilon n}{2}} t(n) O(n^2)$, which is $O(2^{\epsilon n})$. □

By a similar proof, we obtain the same theorem for planted formulas.

Theorem 6. *There exists a monotonically decreasing function $h'' : (0, 1) \to (0, \infty)$ such that when $p \geq h''(\epsilon)/n^2$, the given randomized algorithm finds a satisfying assignment for almost all planted formulas in expected $O(2^{\epsilon n})$ time (where "expected time" is with respect to the random bits used by the algorithm).*

Proof. Notice that Lemma 4 is true with $P_{n,p}^{\text{plant}}[\cdot]$ in place of $P_{n,p}^{\text{sat}}[\cdot]$; this is immediate from the definition of $P_{n,p}^{\text{plant}}[\cdot]$. (Notice that the extra 2^n factor of Lemma 3 is not needed.) Using this modified version of Lemma 4, the proof of the theorem is exactly the same as the proof of Theorem 5. □

Finally, we observe in the next two corollaries that when the clause density is super-linear, then both the satisfiable and planted distributions are solvable almost always in time b^n – for *all* $b > 1$.

Corollary 1. *Suppose that $p(n)$ is $\omega(1/n^2)$. Then for every $b > 1$, there is a randomized algorithm finding a satisfying assignment for all satisfiable formulas in expected $O(b^n)$ time (where "expected time" is with respect to the random bits used by the algorithm).*

Proof. Let $\epsilon > 0$ be sufficiently small so that $2^\epsilon < b$. Observe that $p(n) \geq h'(\epsilon)/n^2$ for all but finitely many n. Modify the algorithm given by Theorem 5 so on formulas of variable set size n such that $p(n) < h'(\epsilon)/n^2$, the desired output is hard-coded. □

Corollary 2. *Suppose that $p(n)$ is $\omega(1/n^2)$. Then for every $b > 1$, there is a randomized algorithm finding a satisfying assignment for all planted formulas in expected $O(b^n)$ time (where "expected time" is with respect to the random bits used by the algorithm).*

The proof of Corollary 2 is similar to that of Corollary 1.

Example 1. Let $p(n) = .0001(\log\log\log n)/n^2$, that is, p is set so that the expected number of clauses is a small constant times $n \log\log\log n$. Then there is an algorithm finding satisfying assignments for almost all satisfiable formulas in expected time $(1.000001)^n$.

5 Discussion

The work in this paper points to the question of whether or not there is a *polynomial time algorithm* that works almost always, for "above the threshold" satisfiable formulas. A number of SAT algorithms, for example, GSAT [24], can be viewed as performing random walks on a Markov chain with state set equal to the set of all assignments. From this perspective, each SAT formula induces a different Markov chain, and what we have really done here is shown that almost all such induced Markov chains (for the distribution of formulas of interest) have a desired convergence property. Perhaps one can perform a more detailed analysis of the Markov chains for some such random walk algorithm, without collapsing assignments with the same Hamming distance from some "target" assignment, together into one state – as is done here and in the analysis of a 2-SAT algorithm given in [20].

It may also be of interest to attempt to prove time lower bounds on restricted classes of random-walk algorithms. For instance, the algorithm in this paper picks a variable to flip based only on the *number* of clauses that would be lost by flipping each variable (that is, the quantity $|L(A, v)|$ for variable v at assignment A) and not based on, for instance, what the clauses of $L(A, v)$ look like, and at which variables they overlap. Can it be shown that all such algorithms with this property have an exponential expected time before converging on a satisfying assignment, say, for the distribution of formulas studied in this paper?

A broader but admittedly more speculative question we would like to pose is whether or not it is possible to develop a complexity theory for distributions of problem instances which takes *almost always polynomial time* as its base notion of tractability. Along these lines, we are curious whether or not distributions that are in almost always time b^n for *all* $b > 1$ – such as those identified by Corollaries 1 and 2 – are always in almost always polynomial time. Can this hypothesis, perhaps restricted to some class of "natural" or computable distributions, be related to any better-known complexity-theoretic hypotheses?

Acknowledgements. The author wishes to thank Amy Gale, Jon Kleinberg, Riccardo Pucella, and Bart Selman for useful discussions and comments. The author also thanks the anonymous referees for their many helpful comments.

References

1. D. Achlioptas, Y. Peres. The Threshold for Random k-SAT is 2^k (ln 2 + o(1)). 35th ACM Symposium on Theory of Computing (STOC 2003).
2. D. Achlioptas, G. B. Sorkin. Optimal myopic algorithms for random 3-SAT. Proc. 41st IEEE Symp. on Foundations of Comput. Sci., 2000, pp. 590-600.
3. P. Beame, R. Karp, T. Pitassi, M. Saks. On the complexity of unsatisfiability of random k-CNF formulas. In Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pages 561-571, Dallas, TX, May 1998.
4. P. Beame, T. Pitassi. Propositional Proof Complexity: Past, Present and Future. Electronic Colloquium on Computational Complexity (ECCC) 5(067): (1998).

5. E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, P. Raghavan, U. Schning. A Deterministic $(2-2/(k+1))^n$ Algorithm for k-SAT Based on Local Search. Theoretical Computer Science, 289(2002).
6. O. Dubois, Y. Boufkhad, J. Mandler. Typical random 3 SAT formulae and the satisfiability threshold. Proc. 11th ACM SIAM Symp. on Discrete Algorithms, San Franscisco, CA, January 2000, pp. 124-126 and Electronic Colloquium on Computational Complexity, TR03-007 (2003).
7. A. Flaxman. A spectral technique for random satisfiable 3CNF formulas. SODA 2003.
8. A. M. Frieze, S. Suen. Analysis of two simple heuristics on a random instance of k-SAT. J. Comput. System Sci. 53 (1996) 312-355.
9. I. Gent. On the Stupid Algorithm for Satisfiability. APES Technical Report 03-1998.
10. A. Goerdt, T. Jurdzinski. Some Results On Random Unsatisfiable K-Sat Instances and Approximation Algorithms Applied To Random Structures. Combinatorics, Probability & Computing, Volume 12, 2003.
11. A. Goerdt, M. Krivelevich. Efficient recognition of random unsatisfiable k-SAT instances by spectral methods. Proceedings of the 18th International Symposium on Theoretical Aspects of Computer Science (STACS 2001). Lecture Notes in Computer Science, 2010:294-304, 2001.
12. A. Goerdt, A. Lanka. Recognizing more random unsatisfiable 3-SAT instances efficiently. LICS'03, Workshop on Typical case complexity and phase transitions, June 21, 2003, Ottawa, Canada.
13. E. Hirsch. SAT Local Search Algorithms: Worst-Case Study. Journal of Automated Reasoning 24(1/2):127-143, 2000.
14. A. Kamath, R. Motwani, K. Palem, P. Spirakis. Tail Bounds for Occupancy and the Satisfiability Threshold Conjecture. Random Structures and Algorithms 7 (1995) 59-80.
15. A. C. Kaporis, L. M. Kirousis, E. G. Lalas. The Probabilistic Analysis of a Greedy Satisfiability Algorithm. ESA 2002: 574-585.
16. L. M. Kirousis, E. Kranakis, D. Krizanc, Y.C. Stamatiou. Approximating the Unsatisfiablity Threshold of Random Formulas. Random Structures and Algorithms 12 (1998) 253-269.
17. E. Koutsoupias, C. H. Papadimitriou. On the greedy algorithm for satisfiability. Inform. Process. Lett. 43 (1992) 53-55.
18. R. Motwani, P. Raghavan. Randomized Algorithms. Cambridge University Press, Cambridge, England, 1995.
19. C. H. Papadimitriou. Computational Complexity. Addison-Wesley, New York, 1994.
20. C. H. Papadimitriou. On selecting a satisfying truth assignment (Extended Abstract). 32nd IEEE Symposium on Foundations of Comput. Sci., 1991, pp. 163-169.
21. U. Schöning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. 40th IEEE Symposium on Foundations of Comput. Sci., 1999 pp. 410-414.
22. R. Schuler, U. Schöning, O. Watanabe, T. Hofmeister. A probabilistic 3-SAT algorithm further improved. Proceedings of 19th International Symposium on Theoretical Aspects of Computer Science, STACS 2002. Lecture Notes in Computer Science, Springer.
23. B. Selman, H. Levesque, D. Mitchell. Hard and Easy Distributions of SAT Problems. Proc. of the 10th National Conference on Artificial Intelligence, 1992, pp. 459-465.

24. B. Selman, H. Levesque, D. Mitchell. A New Method for Solving Hard Satisfiability Problems. Proc. of the 10th National Conference on Artificial Intelligence, 1992, pp. 440-446.