

LINEAR UPPER BOUNDS FOR RANDOM WALK ON SMALL DENSITY RANDOM 3-CNFs*

MIKHAIL ALEKHNovich[†] AND ELI BEN-SASSON[‡]

In memory of Mikhail (Misha) Alekhovich—friend, colleague and brilliant mind

Abstract. We analyze the efficiency of the random walk algorithm on random 3-CNF instances and prove *linear* upper bounds on the running time of this algorithm for small clause density, less than 1.63. This is the first subexponential upper bound on the running time of a *local improvement* algorithm on random instances. Our proof introduces a simple, yet powerful tool for analyzing such algorithms, which may be of further use. This object, called a *terminator*, is a weighted satisfying assignment. We show that any CNF having a good (small weight) terminator is assured to be solved quickly by the random walk algorithm. This raises the natural question of the *terminator threshold* which is the maximal clause density for which such assignments exist (with high probability). We use the analysis of the pure literal heuristic presented by Broder, Frieze, and Upfal [*Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1993, pp. 322–330] and Luby, Mitzenmacher, and Shokrollahi [*Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 364–373] and show that for small clause densities good terminators exist. Thus we show that the pure literal threshold (≈ 1.63) is a lower bound on the terminator threshold. (We conjecture the terminator threshold to be in fact higher.) One nice property of terminators is that they can be found efficiently via linear programming. This makes tractable the future investigation of the terminator threshold and also provides an efficiently computable certificate for short running time of the simple random walk heuristic.

Key words. SAT solving, random CNF, SAT heuristics, random walk algorithm

AMS subject classifications. 68Q25, 68W20, 68W40

DOI. 10.1137/S0097539704440107

1. Introduction. The phenomena we seek to explain is best described by Figure 1.

RWalkSAT, originally introduced by Papadimitriou [35], tries to find a satisfying assignment for a CNF \mathcal{C} by the following method. We start with a random assignment, and as long as the assignment at hand does not satisfy the CNF, an unsatisfied clause $C \in \mathcal{C}$ is picked, and the assignment to a random literal in this clause is flipped. The new assignment satisfies C but may “ruin” the satisfiability of other clauses. We repeat this process (of flipping a bit in the current assignment according to some unsatisfied clause) until either a satisfying assignment is found (success) or we get tired and give up (failure).

The lower batch in Figure 1 (plus sign) was obtained by selecting 810 random 3-CNF formulas¹ with a clause density (i.e., clause/variable ratio) of 1.6 and running RWalkSAT on each instance. The y -axis records the number of assignments used before finding a satisfying one. In particular, the algorithm found an assignment in all

*Received by the editors January 27, 2004; accepted for publication (in revised form) March 24, 2006; published electronically December 21, 2006.

<http://www.siam.org/journals/sicomp/36-5/44010.html>

[†]The author is deceased. Former address: Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112. This work was done while the author was a graduate student at MIT. This author was supported in part by NSF award CCR 0205390 and MIT NTT award 2001-04.

[‡]Department of Computer Science, Technion-Israel Institute of Technology, Technion City, Haifa 32000, Israel (eli@cs.technion.ac.il). This work was done while the author was a Postdoctoral Fellow at MIT and Harvard University. This author was supported by NSF grants CCR-0133096 and CCR-9877049, NSF award CCR 0205390, and NTT award MIT 2001-04.

¹Ten formulas per $n = 2000, 2050, 2100, \dots, 6000$ were selected.

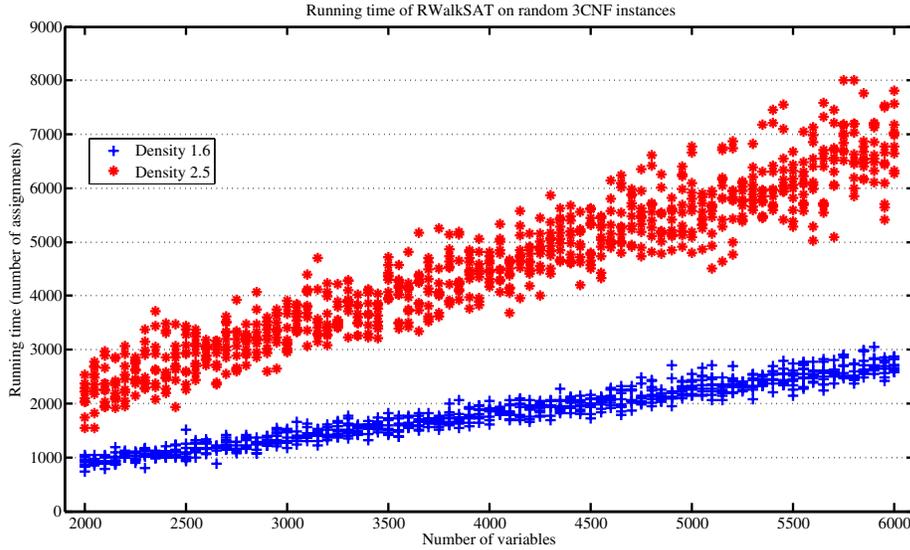


FIG. 1. Running time of RWalkSAT on random 3-CNF instances with clause densities 1.6 and 2.5.

instances. The upper batch (star sign) was similarly obtained by running RWalkSAT on 810 random 3-CNF instances with a higher clause density of 2.5.

Figure 1 raises the conjecture that for clause density 1.6 the running time is linear. Actually, it is even less than the number of variables (and clauses) and seems to have a slope of $\approx 1/2$. In this paper we offer an explanation for the seemingly linear running time of Figure 1. We prove that random 3-CNFs with clause density less than 1.63 take (with high probability) a *linear* number of RWalkSAT steps. (We leave the explanation of the running time displayed in the upper batch of Figure 1 as an interesting open problem.)

1.1. Techniques: Terminators. Our technique can be viewed as a generalization of the analysis of RWalkSAT on satisfiable 2-CNF formulas [35], so we briefly review this result. Papadimitriou showed that the Hamming distance of the assignment at time t from some fixed satisfying assignment α is a random variable that decreases at each step with probability at least $1/2$. Thus, in at most $O(n^2)$ steps this random variable will reach 0, implying we have found α . (The algorithm may succeed even earlier by finding some other satisfying assignment.)

We look at *weighted satisfying assignments*; i.e., we give nonnegative weights to the bits of α . Instead of Hamming distance, we measure the *weighted distance* between α and the current assignment α^t . We show that in some cases, one can find a satisfying assignment α and a set of weights w such that for *any* unsatisfied clause at time t , the expectation of the weighted distance (between α and α^t) decreases by at least 1. Moreover, the maximal weight given to any variable is constant. In this case the running time of RWalkSAT will be linear with high probability (even better than the quadratic upper bound of [35] for 2-CNFs). We call such weighted assignments *terminators*, as their existence assures us that RWalkSAT will terminate successfully in linear time.

Two parameters of a terminator bound the running time of RWalkSAT. The *total weight* (sum of weights of all variables) bounds the distance needed to be traversed by the random walk, because the weighted distance of α^0 from α can be as large as this

sum. The second parameter is the maximal weight of a variable, which bounds the variance of our random walk. Thus we define the termination weight of \mathcal{C} (denoted $\text{Term}(\mathcal{C})$) to be the minimal product of these two parameters, taken over all terminators for \mathcal{C} . As stated above, the running time of `RWalkSAT` is linear (at most) in the termination weight of \mathcal{C} . Not all satisfiable CNFs have these magical terminators, and if \mathcal{C} has no terminator, we define its termination weight to be ∞ .

1.2. Results. With the terminator concept in hand, we examine the running time of `RWalkSAT` on random 3-CNF formulas. If \mathcal{C} is a random 3-CNF, then $\text{Term}(\mathcal{C})$ is a random variable. Understanding this variable and bounding it from above bounds the running time of `RWalkSAT`. Our main result (Theorem 4.1) is that for clause density ≤ 1.63 , a random 3-CNF has *linear* termination weight (hence `RWalkSAT` succeeds in linear time). This matches the behavior depicted in Figure 1 up to a multiplicative constant. We also present a deterministic version of `RWalkSAT` and show it finds a satisfying assignment in linear time for the same clause density (section 3.1).

Our result relies on previous analysis done for bounding a different SAT heuristic, called the *pure literal heuristic* [10] (see also [31] for a different and shorter analysis). This heuristic is known to succeed up to a clause density threshold of 1.63 and fails above this density. We conjecture terminators should exist even beyond the pure literal threshold, as (unreported) experimental data seems to indicate. However, at clause density ≥ 2 only a negligible fraction of random CNFs has terminators (see section 5), meaning we need to develop new techniques for explaining the observed linear running time at (say) density 2.5 depicted in (the upper part of) Figure 1.

A terminator is a solution to a linear system of inequalities, and thus linear programming can be used to find it. Thus, the existence of a terminator for random \mathcal{C} can be decided efficiently, and an upper bound on $\text{Term}(\mathcal{C})$ can be computed efficiently. (However, obtaining the exact value of $\text{Term}(\mathcal{C})$ is not known to be efficiently computable.) This may allow us to gain a better empirical understanding of the behavior of `RWalkSAT` and its connection to the termination weight parameter.

The success of the pure literal heuristic does not necessarily imply polynomial running time for `RWalkSAT`. Indeed, in section 6 we provide a counterexample that requires exponential time from `RWalkSAT`, although a solution can be found using the pure literal heuristic in linear time. Furthermore, for a random planted SAT instance with large enough clause density, `RWalkSAT` takes exponential time (section 7). This is in contrast to the efficient performance of spectral algorithms for planted SAT presented by Flaxman [18].

1.3. History and related results.

Local improvement algorithms. `RWalkSAT` was introduced by Papadimitriou, who showed it has quadratic running time on satisfiable 2-CNFs [35]. An elegant upper bound was given by Schoning, who showed that the expected running time of `RWalkSAT` on any k -CNF is at most $(1 + 1/k)^n$ (compared with the exhaustive search upper bound of 2^n) [38]. The (worst case) upper bound of [38] was improved in a sequence of results [15, 21, 8, 22, 37], and the best upper bound for 3-SAT is $(1.324)^n$, given by the recent paper [22].

`RWalkSAT` is one of a broad family of *local improvement* algorithms, (re)introduced in the 1990s with the work of [41]. Algorithms in this family start with an assignment to the input formula, and gradually change it one bit at a time, by trying to locally optimize a certain function. These algorithms (the most famous of which is WalkSAT) are close relatives of the simulated annealing method and were found to compete with DLL-type algorithms (also known as Davis–Putnam algorithms). Empirical

results on random 3CNFs with up to 100,000 variables seem to indicate that **RWalkSAT** terminates successfully in linear time up to clause density ≤ 2.6 [36, 40]. More advanced algorithms such as WalkSAT (a Metropolis algorithm that is related to **RWalkSAT**) appear empirically to solve random 3CNF instances with clause density ≤ 4 in quadratic time, and there is data indicating polynomial running time up to density ≤ 4.2 (the empirical SAT threshold is ≈ 4.26) [39].

Random 3-CNFs. Random CNFs have received much interest in recent years, being a natural distribution on NP-complete instances that seems (empirically as well as theoretically) computationally hard for a wide range of parameters. This distribution is investigated in such diverse fields as physics [30, 32], combinatorics [24], proof complexity [13], algorithm analysis [3], and hardness of approximation [17], to mention just a few. One of the basic properties of random 3-CNFs is that for small density ($\Delta < 3.52\dots$ (see [20, 29])) almost all formulas are satisfiable, whereas for large density ($\Delta > 4.506\dots$ (see [16])) they are almost all unsatisfiable. Another interesting property is that the threshold between satisfiability and unsatisfiability is sharp [24]. It is conjectured that a *threshold constant* exists, and empirical experiments estimate it to be ≈ 4.26 [14]. The analysis of SAT solving algorithms on random CNFs has been extensively researched empirically, and random CNFs are commonly used as test cases for analysis and comparison of SAT solvers. From a theoretical point of view, several upper bounds were given on the running time of DPLL-type algorithms of increasing sophistication [1, 2, 3, 10, 31, 11, 12, 19, 20, 29]. The best rigorous upper bound for random 3-CNFs is given by the recent papers [20, 29]. An exponential lower bound on a wide class of DPLL algorithms for density ≈ 3.8 and above was given by [3]. Recently, Mézard et al. presented the survey propagation algorithm and showed that nonrigorous arguments based on replica symmetry and experimental results indicate it efficiently solves large random 3CNF instances very close to the empirical satisfiability [32, 33].

Upper bounds for algorithms imply lower bounds on the satisfiability threshold, and in fact, all lower bounds on the threshold (for $k = 3$) so far have come from analyzing specific SAT solving algorithms. Most of the algorithms for which average case analysis has been applied so far are DPLL algorithms (and typically, with the exception of the recent papers [20, 29], when proving upper bounds on these algorithms, myopic² versions are considered). Much less is known about non-DPLL algorithms, in particular local improvement ones. Our result is (to the best of our knowledge) the first rigorous theoretical analysis of a non-DPLL algorithm on random CNFs.

Paper outline. After giving the necessary formal definition in section 2, we discuss terminators in section 3. Using terminators we prove our upper bound in section 4. In section 5 we give some theoretical upper bounds on the terminator threshold. We then discuss the tightness of the terminator method (section 6). We conclude with exponential lower bounds on the running time of **RWalkSAT** on random CNFs from the “planted-SAT” distribution (section 7).

2. Preliminaries.

Random 3-CNFs. For x_i a Boolean variable, a *literal* ℓ_i over x_i is either x_i or \bar{x}_i (the negation of x_i), where x_i is called a *positive literal* and \bar{x}_i is a *negative* one. A *clause* is a disjunction of literals, and a CNF formula is a set of clauses. Throughout this paper we reserve calligraphic notation for CNF formulas. For \mathcal{C} a CNF, let $Vars(\mathcal{C})$ denote the set of variable appearing in \mathcal{C} (we will always assume $Vars(\mathcal{C}) =$

²See [3] for the definition and tightest analysis of myopic algorithms.

$\{x_1, \dots, x_n\}$ for some n). An *assignment* to \mathcal{C} is some Boolean vector $\alpha \in \{0, 1\}^n$. A literal ℓ_i is satisfied by α iff $\ell_i(\alpha_i) = 1$. We study the following distribution.

DEFINITION 2.1. Let \mathbb{F}_Δ^n be the probability distribution obtained by selecting Δn clauses uniformly at random from the set of all $8 \cdot \binom{n}{3}$ clauses of size 3 over n variables. $\mathcal{C} \sim \mathbb{F}_\Delta^n$ means that \mathcal{C} is selected at random from this distribution. We call such a \mathcal{C} a random 3-CNF

The algorithm. RWalkSAT is described by the following pseudocode. \mathcal{C} is the input CNF and T is the time bound; i.e., if no satisfying assignment is found in T steps, we give up. We use the notation $UNSAT(\mathcal{C}, \alpha)$ for the set of clauses of \mathcal{C} that are unsatisfied by α .

RWalkSAT(\mathcal{C}, T)

```

Select  $\alpha \in \{0, 1\}^n$  (uniformly) at random
Initialize  $t = 0$ 
While  $t < T$  {
  If  $\mathcal{C}(\alpha) = 1$  Return (“INPUT SATISFIED BY”  $\alpha$ )
  Else {
    Select  $C \in UNSAT(\mathcal{C}, \alpha)$  at random
    Select literal  $\ell \in C$  at random
    Flip assignment of  $\alpha$  at  $\ell$ 
     $t++$ 
  }
}

```

Return “FAILED TO FIND SATISFYING ASSIGNMENT IN T STEPS”

Martingales and Azuma’s inequality. Below we state Azuma’s inequality for martingales. We refer the reader to [34] for the definition of conditional expectation and for more information about martingales.

A *martingale* is a sequence $X_0, X_1, X_2, \dots, X_m$ of random variables such that for $0 \leq i < m$ holds

$$\mathbf{E}[X_{i+1}|X_i] = X_i.$$

The following version of Azuma’s inequality [7, 27] may be found in [6].

THEOREM 2.2 (Azuma’s inequality). Let $0 = X_0, \dots, X_m$ be a martingale with $|X_{i+1} - X_i| \leq 1$ for all $0 \leq i < m$. Let $\lambda > 0$ be arbitrary. Then

$$\Pr[X_m > \lambda\sqrt{m}] < e^{-\lambda^2/2}.$$

3. Terminators. In this section we develop the tools needed to bound the running time of RWalkSAT on various interesting instances.

Intuition. Suppose a k -CNF \mathcal{C} over n variables has a satisfying assignment α such that each clause of \mathcal{C} is satisfied by at least $k/2$ literals under α . In this case RWalkSAT will terminate in quadratic time (with high probability). The reason is that if a clause C is unsatisfied at time t by α^t , then α^t must disagree with α on at least half of the literals in C . So with probability $\geq 1/2$ we decrease the Hamming distance between our current assignment and α . If we let sim^t be the similarity of α^t and α , i.e., the number of bits that are identical in both assignments (notice $0 \leq \text{sim}^t \leq n$), then sim^t is a submartingale, i.e., $E(\text{sim}^t | \text{sim}^1, \dots, \text{sim}^{t-1}) \geq \text{sim}^{t-1}$. Standard techniques from the theory of martingales show that sim reaches n (so α^t reaches α) within $O(n^2)$ steps. One elegant example of this situation is when \mathcal{C} is a satisfiable 2-CNF. Papadimitriou [35] proved quadratic upper bounds on the running time of RWalkSAT in this case, using the proof method outlined above.

For a general 3-CNF we do not expect a satisfying assignment to have two satisfying literals per clause. Yet all we need in order to prove good running time is to set up a measure of similarity between α^t and some fixed satisfying assignment α such that (i) if sim^t reaches its maximal possible value, then $\alpha^t = \alpha$; and (ii) the random variables $\text{sim}^1, \text{sim}^2, \dots$ are a submartingale. We achieve both these properties by giving *nonnegative weights* w_1, \dots, w_n to the variables x_1, \dots, x_n . Instead of similarity, we measure the *weighted similarity* between α and α^t , defined by $\text{sim}_w(\alpha, \alpha^t) \stackrel{\text{def}}{=} \sum_{\alpha_i^t = \alpha_i} w_i$. Now suppose there exists a satisfying assignment α such that for any clause C , the expected change in sim_w , conditional on C being unsatisfied, is nonnegative. Suppose, furthermore, that all w_i are bounded by a constant and every clause has a variable with nonzero weight bounded below by another constant. Then we may conclude as above that α^t will reach its maximal value $W = \sum_i w_i$ in time $O(W^2)$.

In some cases we can do even better. We set up a system of weights such that (for any clause C) the expected change in sim_w (conditional on C being unsatisfied) is *strictly positive*. In this case the running time is *linear* in $W = \sum w_i$ (instead of quadratic). As we shall later see, such a setting of weights is possible (with high probability) for random 3-CNFs. But first we formalize our intuition.

Notation. In what follows Boolean variables range over $\{-1, 1\}$. A CNF \mathcal{C} with n variables and m clauses is represented by an $m \times n$ matrix $A^{\mathcal{C}}$ with $\{-1, 0, 1\}$ -entries. The i th clause is represented by $A_i^{\mathcal{C}}$ (the i th row of $A^{\mathcal{C}}$) and has a -1 -entry in the j th position if \bar{x}_j is a literal of the i th clause of \mathcal{C} , a 1 -entry if x_j is a literal of C_i , and is zero otherwise. Thus, if \mathcal{C} is a k -CNF, then the support size of each row $A_i^{\mathcal{C}}$ is at most k . A Boolean assignment is $\alpha \in \{-1, 1\}^n$, and we say α satisfies \mathcal{C} iff for all $i \in [m]$

$$(1) \quad \langle A_i^{\mathcal{C}}, \alpha \rangle > - \|A_i^{\mathcal{C}}\|_1,$$

where $\langle \alpha, \beta \rangle$ is the standard inner product over \mathbb{R}^n (defined by $\sum_{i=1}^n \alpha_i \cdot \beta_i$) and $\|\cdot\|_1$ is the ℓ_1 norm (defined by $\|\beta\|_1 = \sum_{i=1}^n |\beta_i|$). It is easy to see that this definition of satisfiability coincides with the standard one.

Terminator: Definition. A terminator is a generalization of a satisfying assignment. On the one hand, we allow α to be any vector in \mathbb{R}^n , but we require a stronger satisfying condition than (1).

DEFINITION 3.1 (terminators). *Let \mathcal{C} be a k -CNF with n variables and m clauses represented by the matrix $A^{\mathcal{C}}$. $\alpha \in \mathbb{R}^n$ is a terminating satisfying assignment (or terminator) if for all $i \in [m]$*

$$(2) \quad \langle A_i^{\mathcal{C}}, \alpha \rangle \geq 1.$$

The termination weight of \mathcal{C} is

$$\text{Term}(\mathcal{C}) \stackrel{\text{def}}{=} \min\{\|\alpha\|_1 \cdot \|\alpha\|_{\infty} : \alpha \text{ terminator for } \mathcal{C}\}.$$

In case \mathcal{C} has no terminator, we define $\text{Term}(\mathcal{C})$ to be ∞ .

One may think of $\text{sign}(\alpha_i)$ as the Boolean assignment to variable x_i (where $\text{sign}(\alpha_i)$ is 1 if $\alpha_i \geq 0$ and is -1 otherwise) and $|\alpha_i|$ as the weight given to x_i . Notice that if α is a terminator, then the $\{-1, 1\}$ -vector $\text{sign}(\alpha)$ satisfies \mathcal{C} . This is because by property (2) in each clause there is at least one literal that agrees in sign with α .

The decisive name given in the previous definition is justified by the following claim, which is the main theorem of this section.

THEOREM 3.2 (terminator theorem). *If a k -CNF \mathcal{C} has a terminator α , then RWalkSAT succeeds on \mathcal{C} in time $O(\|\alpha\|_1 \cdot \|\alpha\|_\infty)$ with probability $\geq 1 - \exp(-\Omega(\|\alpha\|_1 / \|\alpha\|_\infty))$.*

Notice that we do not claim that when RWalkSAT terminates, it finds the assignment $\text{sign}(\alpha)$, but rather the existence of any terminator of small weight implies short running time. We can say that RWalkSAT is “drawn to” α but only when using the weighted distance measure given by $|\alpha|$. If $|\alpha_i| = 1$, this means RWalkSAT indeed approaches α (as is the case when each clause is satisfied by two literals). But in general, being “close” according to the weighted measure $|\alpha|$ does not imply small Hamming distance.

Proof of Theorem 3.2. Let \mathcal{C} be a k -CNF and α be a terminator of minimal weight for \mathcal{C} , i.e., $\text{Term}(\mathcal{C}) = \|\alpha\|_1 \cdot \|\alpha\|_\infty < \infty$. Let $\beta^t \in \{-1, 1\}^n$ be the sequence of assignments traversed by RWalkSAT(\mathcal{C}) starting from the random assignment β^1 , where $t \leq T = c \cdot k \cdot \|\alpha\|_1 \cdot \|\alpha\|_\infty$ (c will be fixed later). For $t \geq 1$ let Y^t be the random variable $\langle \beta^t, \alpha \rangle$. If RWalkSAT fails to find a satisfying assignment in T steps, then the following event occurs:

$$(3) \quad Y^t < \|\alpha\|_1 \quad \text{for all } t < T.$$

This is because $\langle \beta^t, \alpha \rangle = \|\alpha\|_1$ implies $\beta^t = \text{sign}(\alpha)$ and $\text{sign}(\alpha)$ satisfies \mathcal{C} . Thus we need only to bound the probability of event (3). Suppose clause C_i is picked at time t (i.e., C_i is unsatisfied by β^{t-1}). We claim the expected change in Y^t (with respect to Y^{t-1}) is precisely

$$(4) \quad \frac{2}{k} \cdot \langle A_i^{\mathcal{C}}, \alpha \rangle.$$

With probability $1/k$ we flip the assignment to each literal x_j of C_i , which amounts to multiplying β_j^{t-1} by -1 . Thus the expected change in Y^t is $\frac{-2}{k} \cdot \langle \beta^{t-1}|_i, \alpha \rangle$, where $\beta^{t-1}|_i$ is the restriction of β^{t-1} to support of $A_i^{\mathcal{C}}$. But C_i being unsatisfied by β^{t-1} implies $\beta^{t-1}|_i = -A_i^{\mathcal{C}}$, so (4) is proved. Thus by property (2) in Definition 3.1

$$E[Y^t | Y^1, \dots, Y^{t-1}] = Y^{t-1} + \frac{2}{k} \langle A_i^{\mathcal{C}}, \alpha \rangle \geq Y^{t-1} + \frac{1}{k}.$$

We claim that the sequence of random variables

$$X_t \stackrel{\text{def}}{=} \sum_{\ell=1}^t (Y^\ell - \mathbf{E}[Y^\ell | Y^1, \dots, Y^{\ell-1}])$$

is a martingale satisfying $\mathbf{E}X_1 = 0$. Indeed,

$$\begin{aligned} \mathbf{E}[X_t | X_1, \dots, X_{t-1}] &= \mathbf{E}[X_t | Y^1, \dots, Y^{t-1}] \\ &= \mathbf{E} \left[\sum_{\ell=1}^t (Y^\ell - \mathbf{E}[Y^\ell | Y^1, \dots, Y^{\ell-1}]) \mid Y^1, \dots, Y^{t-1} \right] \\ &= \mathbf{E}[Y^t | Y^1, \dots, Y^{t-1}] - \mathbf{E}[\mathbf{E}[Y^t | Y^1, \dots, Y^{t-1}] | Y^1, \dots, Y^{t-1}] \end{aligned}$$

$$\begin{aligned}
 & + \mathbf{E} \left[\sum_{\ell=1}^{t-1} (Y^\ell - \mathbf{E}[Y^\ell | Y^1, \dots, Y^{\ell-1}]) | Y^1, \dots, Y^{t-1} \right] \\
 & = 0 + \mathbf{E} \left[\sum_{\ell=1}^{t-1} (Y^\ell - \mathbf{E}[Y^\ell | Y^1, \dots, Y^{\ell-1}]) | Y^1, \dots, Y^{t-1} \right] \\
 & = \sum_{\ell=1}^{t-1} (Y^\ell - \mathbf{E}[Y^\ell | Y^1, \dots, Y^{\ell-1}]) = X_{t-1}.
 \end{aligned}$$

Also $E[X_1] = E[Y^1 - E[Y^1]] = 0$. For all t , $|X_{t+1} - X_t| = Y^{t+1} - \mathbf{E}[Y^{t+1} | Y^1, \dots, Y^t] \leq \|\alpha\|_\infty$. Note that

$$X_t = Y^t - \sum_{\ell=1}^t (\mathbf{E}[Y^\ell | Y^1, \dots, Y^{\ell-1}] - Y^\ell) - \mathbf{E}Y^1 \leq Y^t - t/k + \|\alpha\|_1.$$

In order to bound the probability of event (3), it suffices to bound the probability of the event “ $X_T < 2\|\alpha\|_1 - T/k$ ” (if this event does not occur, then $Y^T \geq -\|\alpha\|_1 + X_T \geq \|\alpha\|_1$). Recalling $T = c \cdot k \cdot \|\alpha\|_1 \cdot \|\alpha\|_\infty$ we will pick $c > \frac{4}{k}$ so that

$$2\|\alpha\|_1 - \frac{T}{k} = 2\|\alpha\|_1 - c \cdot \|\alpha\|_1 \cdot \|\alpha\|_\infty < -\frac{ck}{2} \|\alpha\|_1 \cdot \|\alpha\|_\infty.$$

We now apply Azuma’s inequality and get

$$\begin{aligned}
 (3) & \leq \Pr \left[X_T < -\frac{ck}{2} \|\alpha\|_1 \|\alpha\|_\infty \right] \\
 & = \Pr \left[\frac{X_T}{\|\alpha\|_\infty} < -\frac{ck}{2} \|\alpha\|_1 \right] \\
 & \leq \exp \left(-\frac{(\frac{ck}{2} \|\alpha\|_1)^2}{2T} \right) \\
 & \leq \exp \left(-\frac{c^2 k^2 (\|\alpha\|_1)^2}{8ck \cdot \|\alpha\|_1 \|\alpha\|_\infty} \right) \\
 & \leq \exp \left(-\frac{ck \|\alpha\|_1}{8 \|\alpha\|_\infty} \right) = \exp \left(-\Omega \left(\frac{\|\alpha\|_1}{\|\alpha\|_\infty} \right) \right).
 \end{aligned}$$

The theorem is proved. \square

3.1. A deterministic variant of RWalkSAT. Consider the following deterministic variant of RWalkSAT, which we will call DWalkSAT. Fix an ordering on clauses in \mathcal{C} . Initialize α_0 to be (say) the all zero assignment. At each step t , select the smallest clause unsatisfied by α_t and flip the assignment to *all* literals in it. Repeat this process until all clauses are satisfied. Naturally, one can introduce a time bound T and declare failure if a satisfying assignment is not found within T steps. We immediately get the following result.

THEOREM 3.3. *If a CNF \mathcal{C} has a terminator α , then DWalkSAT succeeds on \mathcal{C} within $2 \cdot \|\alpha\|_1$ steps.*

Proof. We closely follow the proof of the terminator theorem, Theorem 3.2. Let β^1, \dots be the (deterministic) sequence of assignments traversed by the algorithm.

Let $Y^t = \langle \beta^t, \alpha \rangle$ (noticing Y^t is no longer random). Clearly, $Y^1 \geq -\|\alpha\|_1$, and if $Y^t = \|\alpha\|_1$, then β^t (equals $\text{sign}(\alpha)$, hence) satisfies \mathcal{C} . So we have to only show for all t

$$(5) \quad Y^t \geq Y^{t-1} + 2.$$

This follows from the fact that the clause \mathcal{C}_i flipped at time t was unsatisfied at time $t - 1$. Flipping all variables in \mathcal{C}_i amounts to adding to Y^{t-1} the amount $\langle A_i^{\mathcal{C}}, \alpha \rangle$, and this, by definition of terminator, is at least one. We have proved (5) and with it the theorem. \square

4. Linear upper bounds on random CNFs. In this section we show that for clause densities for which the pure literal heuristic succeeds, there exist linear weight terminators. Our current analysis uses insights into the structure of such pure CNFs, but we see no reason to believe that the terminator threshold is linked to the pure literal threshold. The main theorem of this section is the following.

THEOREM 4.1. *For any $\Delta < 1.63$, there exists a constant c such that with high probability $\mathcal{C} \sim \mathbb{F}_{\Delta}^n$ has a terminator $\alpha \in \mathbb{R}^n$ with $\|\alpha\|_{\infty} \leq c$ and hence $\|\alpha\|_1 \leq c \cdot n$.*

COROLLARY 4.2. *For any $\Delta < 1.63, \epsilon > 0$, there exists a constant c such that with high probability for $\mathcal{C} \sim \mathbb{F}_{\Delta}^n$, **RWalkSAT** succeeds on \mathcal{C} in time $c \cdot n$ with probability $\geq 1 - \epsilon$.*

COROLLARY 4.3. *For any $\Delta < 1.63, \epsilon > 0$, there exists a constant c such that with high probability for $\mathcal{C} \sim \mathbb{F}_{\Delta}^n$, **DWalkSAT** succeeds on \mathcal{C} in time $c \cdot n$.*

To prove our main theorem, we construct small weight terminators for pure and expanding CNFs and then merge the two into one small weight terminator.

4.1. Terminators for pure CNFs. A literal ℓ in \mathcal{C} is called *pure* if it appears only as a positive literal, or only as a negative literal, in \mathcal{C} . A clause in \mathcal{C} is said to be *pure* if it contains a pure literal. When seeking a satisfying assignment, a natural strategy is to start by assigning all pure literals their satisfying assignment and thus remove all pure clauses. The removal of pure clauses may result in the emergence of new pure literals in the restricted CNF, and the process may be repeated. The *pure literal heuristic* is the heuristic that applies this removal process until no pure clauses remain. If the remaining CNF is empty, the pure literal heuristic has found a satisfying assignment, and otherwise it failed.

Let us introduce some notation. For \mathcal{C} a CNF, define $\mathcal{C}_0 = \mathcal{C}$, L_0 to be the set of pure literals in \mathcal{C} , and P_0 to be the set of pure clauses in \mathcal{C} . Recursively define \mathcal{C}_{i+1} to be $\mathcal{C}_i \setminus P_i$, and let L_{i+1}, P_{i+1} be, respectively, the set of pure literals and pure clauses in \mathcal{C}_{i+1} . Finally, let r be the minimal i such that $L_i = \emptyset$. Notice that the pure literal succeeds on \mathcal{C} iff $\mathcal{C}_r = \emptyset$. If $\mathcal{C}_r = \emptyset$, we say \mathcal{C} is *r-pure*.

THEOREM 4.4. *Every r-pure k-CNF over n variables has a terminator $\alpha \in \mathbb{R}^n$ with $\|\alpha\|_{\infty} \leq k^r$ and $\|\alpha\|_1 \leq n \cdot k^r$, so $\text{Term}(\mathcal{C}) \leq n \cdot k^{2r}$. Moreover, α is supported only on $\cup_{i=0}^{r-1} L_i$.*

Notice that invoking Theorem 3.2 we bound the running time of **RWalkSAT** on an *r-pure k-CNF* by $n \cdot k^{2r}$ (with high probability).

Proof. Let L_0, \dots, L_{r-1} be the pure literals in $\mathcal{C}_0, \dots, \mathcal{C}_{r-1}$. Notice that $\cup_{j=0}^{r-1} L_j$ does not necessarily cover all variables in \mathcal{C} , but assigning each pure literal to 1 (i.e., if ℓ_i is pure, then set $\text{sign}(\alpha_i) = \text{sign}(\ell_i)$) and assigning the other variables arbitrarily gives a satisfying assignment α . We now deal with the weights (absolute values) of α . Fix the weight of each variable in L_j to k^{r-j} . For any variable $x_i \notin \cup_{j=0}^{r-1} L_j$ fix its weight to 0.

To see that α is a terminator (of weight nk^r), consider any clause $C_i \in P_j$. By definition of P_j there are no literals from L_0, \dots, L_{j-1} appearing in C . Thus all literals appearing in C have weight $\leq (k)^{r-j}$. There is at least one literal $\ell_s \in C$ that has weight k^{r-j} and agrees with α_s in sign, and any literal disagreeing with α must have weight $\leq (k)^{r-j-1}$. Hence

$$\langle A_i^C, \alpha \rangle \geq k^{r-j} - (k-1) \cdot k^{r-j-1} \geq 1. \quad \square$$

Broder, Frieze, and Upfal showed that with high probability the pure literal heuristic finds a satisfying assignment for a random 3-CNF with clause density < 1.63 [10] (for a simpler analysis of the same heuristic see [31]). In particular, the following theorem follows from the work of [10]. A proof of this theorem can be found in Appendix A.

THEOREM 4.5 (see [10]). *For every $\Delta < 1.63$, there exists a constant c such that with high probability $\mathcal{C} \sim \mathbb{F}_\Delta^n$ is $c \log n$ -pure.*

By applying Theorems 3.2 and 4.4 to Theorem 4.5 we conclude that the running time of RWalkSAT on a random instance (with small enough clause density) is at most polynomial.

4.2. Terminators for expanding CNFs. Our next step in proving Theorem 4.1 starts with the following theorem, which is a combination of a result of Broder, Frieze, and Upfal [10] and (the now) standard analysis of random CNFs, originating in the work of Chvátal and Szemerédi [13]. Being standard and somewhat technical, we defer its proof to Appendix A.

DEFINITION 4.6. *For \mathcal{C} a CNF, we say \mathcal{C} is an (r, c) -expander if for all $\mathcal{C}' \subseteq \mathcal{C}$ $|\mathcal{C}'| \leq r$, $|\text{Vars}(\mathcal{C}')| \geq c \cdot |\mathcal{C}'|$.*

THEOREM 4.7. *For every $\Delta < 1.63$, there exists an integer d such that for $\mathcal{C} \sim \mathbb{F}_\Delta^n$, with high probability \mathcal{C}_d is a $(|\mathcal{C}_d|, 7/4)$ -expander, where \mathcal{C}_d is the CNF remaining of \mathcal{C} after removing the d outermost pure layers.*

This theorem assures us that after removing a constant number of the layers from a random \mathcal{C} (with small clause density), we have in hand a residual CNF \mathcal{C}_d , such that any subset of it, including all of \mathcal{C}_d , has a very large set of neighbors. This in turn implies the existence of small weight terminators for \mathcal{C}_d .

THEOREM 4.8. *If \mathcal{C} is an $(|\mathcal{C}|, 7/4)$ -expanding 3-CNF over n variables, then \mathcal{C} has a terminator $\alpha \in \mathbb{R}^n$ with $\|\alpha\|_\infty \leq 4$ (hence $\|\alpha\|_1 \leq 4n$).*

Proof. Form the following bipartite graph G . On the left-hand side, put one vertex for each clause in \mathcal{C} . On the right-hand side, put 4 distinct vertices for each variable appearing in \mathcal{C} . Connect the vertex labeled by the clause C to all 12 vertices labeled by variables appearing in C . We do not care if the appearance is as positive or negative literals.

Since \mathcal{C} is an $(|\mathcal{C}|, \frac{7}{4})$ -expander, G has expansion factor 7; i.e., for all subsets S on the left-hand side, $|N(S)| \geq 7 \cdot |S|$, where $N(S)$ is the set of neighbors of S . By Hall's matching theorem [26] we conclude that there is a 7-matching from the left-hand side to the right; i.e., each node C on the left-hand side can be associated with a set of seven of its neighbors on the right-hand side (denoted $N'(C)$), such that for all clauses $C \neq D$, $N'(C) \cap N'(D) = \emptyset$. We now use N' to define our terminator α . For any variable x , if there exists a clause C such that $N'(C)$ has at least three members labeled by x , then we say x is associated with C , and the weight of x is the number of copies of x in $N'(C)$ (notice this weight is either 3 or 4). For any variable x_i associated with a clause C , set $\text{sign}(\alpha_i)$ to the value that satisfies C and set $|\alpha_i|$ to the weight of x_i . Set all other variables to zero. α is well defined because a

variable can be associated with at most one clause. We are left with verifying that it is a terminator. This follows by a case analysis, using the fact that each clause has a dozen neighbors, and seven of them are in $N'(C_i)$. There are three cases to consider.

C_i has at least two associated variables: In this case, $\text{sign}(\alpha)$ agrees with C on at least two variables, and each variable has weight at least 3. The remaining variable has weight at most 4, so $\langle A_i^C, \alpha \rangle \geq 6 - 4 \geq 2$.

C_i has one associated variable of weight three: The remaining four neighbors of $N'(C_i)$ must be evenly split between the two remaining variables of C (otherwise C_i would have two associated variables). So the remaining pair of variables of C_i have weight zero. Hence $\langle A_i^C, \alpha \rangle = 3$.

C_i has one associated variable of weight four: The remaining three neighbors of $N'(C_i)$ are split between the remaining two variables. One variable has two such neighbors (and hence zero weight) and the other has one neighbor, so the weight of this literal is at most 3. Thus, $\langle A_i^C, \alpha \rangle \geq 4 - 3 = 1$.

Theorem 4.8 follows. \square

4.3. Small weight terminators for random CNFs.

Proof of Theorem 4.1. By Theorem 4.7, (with high probability) \mathcal{C} can be partitioned into the d outermost pure layers $\mathcal{C}' \stackrel{\text{def}}{=} \cup_{i=0}^{d-1} P_i$ and the remaining residual inner core $\mathcal{C}'' = \mathcal{C}_d$. This inner core is a $(|\mathcal{C}''|, 7/4)$ -expander. We know (by Theorems 4.4 and 4.8, respectively) how to construct terminators for each of these formulas, so all we need to do is merge them into a single terminator for \mathcal{C} .

Let α', α'' be the respective terminators of $\mathcal{C}', \mathcal{C}''$. By Theorem 4.4 α' has all its support on pure literals, which do not appear in \mathcal{C}'' . Thus the supports of α' and α'' are disjoint. We merge the two assignments by defining α as the assignment that agrees with $9 \cdot \alpha'$ on the support of α' and agrees with α'' otherwise (the reason for multiplying α' by the scalar 9 will soon become clear). By our previous remark (that α' and α'' have disjoint supports) α is well defined, and we now prove it is a terminator.

Consider a clause $C_i \in \mathcal{C}$. If $C_i \in \mathcal{C}''$, then $\langle A_i^C, \alpha \rangle = \langle A_i^C, \alpha'' \rangle \geq 1$, because all literals appearing in \mathcal{C}'' are given zero weight by α' . Otherwise, $C_i \in \mathcal{C}'$ might have some of its (nonpure) literals in $\text{Vars}(\mathcal{C}'')$, but recall that the maximal weight of α'' is 4, so in the worst case C_i has two literals with weight 4 coming from α'' . Thus $\langle A_i^C, \alpha \rangle \geq 9 - 2 \cdot 4 = 1$. We have shown the existence of a terminator of linear total weight, and the proof of Theorem 4.1 is complete. \square

5. Investigating the terminator threshold. When \mathcal{C} is a random CNF, $\text{Term}(\mathcal{C})$ is a random variable. Since $\text{Term}(\mathcal{C})$ bounds the running time of RWalkSAT , investigating this random variable is an interesting question. The property of having a terminator α with $\|\alpha\|_\infty \leq w$ is monotone with respect to addition of new clauses. Thus one can define the *terminator threshold* θ_n^w as the density for which a terminator α , $\|\alpha\|_\infty \leq w$ exists with probability $1/2$.

CLAIM 5.1. *A CNF \mathcal{C} with m clauses and n variables has some terminator iff $0 \notin \text{convex hull}(\{A_i^C : i = 1, \dots, m\})$.*

Proof. Think of a terminator α as the normal of a hyperplane in \mathbb{R}^n passing through zero. This hyperplane partitions \mathbb{R}^n into two parts. $\langle A_i^C, \alpha \rangle > 0$ iff the point A_i^C lies in the positive half of \mathbb{R}^n . Thus $\langle A_i^C, \alpha \rangle > 0, i = 1, \dots, m$, iff zero is not in the convex hull of the points. \square

Füredi proved the following general theorem (he gave a tighter bound than presented here, but the form we quote is sufficient for our purposes). A set of points

$P \subset \mathbb{R}^n$ is symmetric if $p \in P \Rightarrow (-p) \in P$.

THEOREM 5.2 (see [25]). *Let $\{P_n \subset \mathbb{R}^n\}_{n \in \mathbb{N}}$ be an infinite family of finite symmetric sets of points. Suppose $(2 + \epsilon)n$ points are selected uniformly at random from P_n . Then*

$$\lim_{n \rightarrow \infty} \Pr[0 \notin \text{convex hull of points}] = 0.$$

In our case P_n is the symmetric set of $\{-1, 0, 1\}$ -valued points with support size 3. Thus, by Füredi’s theorem when the clause density is greater than 2, with high probability there is no terminator. Notice this upper bound on the terminator threshold holds for any k -CNF, even for nonconstant k (e.g., $k = n$). Combining Theorem 4.1 with Füredi’s theorem gives for $k = 3$ the following bounds:

$$1.63 \leq \theta_n^\infty \leq 2.$$

We leave the resolution of the terminator threshold for $k = 3$ as an interesting open problem.

For the case of 2-CNFs we can bound the terminator threshold from above by 1, because this is the satisfiability threshold for random 2-CNFs (and a terminator implies satisfiability). It seems reasonable to conjecture that for $k = 2$ the satisfiability and terminator threshold coincide. This could be used to prove that for random 2-CNFs below the satisfiability threshold, RWalkSAT terminates in linear time (as opposed to the quadratic upper bound guaranteed for any satisfiable 2-CNF by [35]).

6. Tightness of terminator based bounds. In this section we show that the upper bound derived by the terminator method is tight, even for pure CNFs. We present pure CNFs such that the running time of RWalkSAT on them is exponential in the number of variables and also lower bounded by the terminator weight.

THEOREM 6.1. *For arbitrarily large n , there exist pure 3-CNFs over n variables, with total terminator weight $\geq 2^{n/2}$, and the running time of RWalkSAT on them is $2^{\epsilon n}$ for some $\epsilon > 0$.*

Proof. Use the following formula, which is a slight variation of the X -DAG contradiction used in [9].

DEFINITION 6.2. *Let \mathcal{G}_n be the following CNF over variables $x_1, \dots, x_n, y_1, \dots, y_n, z$:*

$$\{\bar{x}_1\} \wedge \{\bar{y}_1\} \wedge \bigwedge_{i=1}^{n-1} \{x_i \vee y_i \vee \bar{x}_{i+1}\} \wedge \bigwedge_{i=1}^{n-1} \{x_i \vee y_i \vee \bar{y}_{i+1}\} \wedge \{x_n \vee y_n \vee \bar{z}\}.$$

\mathcal{G}_n has a unique satisfying assignment, $\vec{0}$. Moreover, \mathcal{G}_n is n -pure, because \bar{z} appears only in one clause, and once z is satisfied and removed, \bar{y}_n, \bar{x}_n each appear in one clause in the remaining formula. Thus, one can repeatedly remove x_{i-1}, y_{i-1} until all the formula is satisfied. This implies the existence of a terminator of weight 3^n , and it is not hard to see that any terminator must have weight 2^n at least. We claim that RWalkSAT requires exponential time to succeed on \mathcal{G}_n .

Let X_t be the number of ones assigned by α_t to the variables $x_2, \dots, x_n, y_2, \dots, y_n$. With high probability $X_0 > (1 - \epsilon)n$, and if RWalkSAT(\mathcal{G}_n, T) succeeds, we know $X_T = 0$. But for every step t , the probability of X_t decreasing is at most $1/3$. The theorem follows. \square

7. Lower bounds for large density planted SAT. In this section, we state (without proofs) that **RWalkSAT** is not a good algorithm for random CNFs with large clause density. By definition, **RWalkSAT** gives the correct answer on any unsatisfiable formula. For large enough clause density ($\Delta > 4.6$), almost all formulas in \mathbb{F}_Δ^n are unsatisfiable [16]. Thus, one may argue that **RWalkSAT** operates very well for these densities. On second thought, on this distribution, even the constant time algorithm that fails on every input, without reading it, operates well. Thus, it makes sense to discuss the performance of **RWalkSAT** only on the uniform distribution over *satisfiable* formulas with Δn clauses (denoted \mathbf{SAT}_n^Δ). The problem is that for small densities, \mathbf{SAT}_n^Δ is not well characterized, we do not know how to analyze it. Thus, we propose looking at the following pair of *planted SAT* distributions over satisfiable 3-CNFs.

DEFINITION 7.1 (planted SAT). *Let \mathbb{S}_Δ^n be the distribution obtained by selecting at random $\beta \in \{0, 1\}^n$ and selecting at random Δn clauses out of all clauses of size 3 that are satisfied by β . Denote a random formula from this distribution by $\mathcal{C} \sim \mathbb{S}_\Delta^n$.*

Let \mathbb{P}_Δ^n be the distribution obtained by selecting at random $\beta \in \{0, 1\}^n$, and for each clause C satisfied by β , select C to be in \mathcal{C} with independent probability $p_n^\Delta = \frac{6\Delta}{7(n-1)(n-2)}$. Denote a random formula from this distribution by $\mathcal{C} \sim \mathbb{P}_\Delta^n$.

This distribution is highly interesting in its own right. It is the analogue of the *planted clique* and *planted bisection* distributions, studied, e.g., in [5, 23, 28]. There are efficient spectral algorithms for finding the satisfying assignment for the planted SAT distribution [18], and in this section we argue that **RWalkSAT** performs poorly (takes exponential running time) on this distribution. The proofs of this result are fairly straightforward, so we omit them from the paper.

THEOREM 7.2 (main lower bound). *There exists a constant $\Delta_0 > 0$, such that for all $\Delta \geq \Delta_0$ (Δ may be a function of n), with high probability for $\mathcal{C} \sim \mathbb{P}_\Delta^n$, or $\mathcal{C} \sim \mathbb{S}_\Delta^n$*

$$\mathbf{P}[\mathbf{RWalkSAT}(\mathcal{C}, 2^{\epsilon n}) \text{ succeeds}] \leq 2^{-\epsilon n},$$

where $\epsilon > 0$ is some a constant, depending on Δ .

The rest of this section is devoted to a sketch of the proof of Theorem 7.2. We warm up by discussing the case of \mathcal{C} being the maximal size CNF satisfying β and then apply our insights to the case of a random CNF. For the rest of this section we assume without loss of generality that β , the random planted assignment, is the all zero vector, denoted $\vec{0}$.

The full CNF of size n , denoted \mathcal{F}_n , has all clauses of size exactly 3 (without repetition of literals) that are satisfied by $\vec{0}$. Our starting point is the following.

LEMMA 7.3. $\mathbf{P}[\mathbf{RWalkSAT}(\mathcal{F}_n, 2^{n/100}) \text{ succeeds}] \leq 2^{-n/100}$.

Intuitively, the lemma holds because for an assignment that is very close to $\vec{0}$, the fraction of falsified clauses that have two (or more) positive literals is significantly larger than the fraction of falsified clauses with only one positive literal. Thus, a random falsified clause is more likely to lead us away from $\vec{0}$ and hence the exponential running time.

To complete the proof of Theorem 7.2, notice $\mathcal{C} \sim \mathbb{P}_\Delta^n$ is a “random fraction” of \mathcal{F}_n . Additionally, for large Δ all satisfying assignments are close to $\vec{0}$. Thus, when the random walk algorithm reaches an assignment that is close to $\vec{0}$, the fraction of clauses with two or more positive literals is significantly larger than the fraction of falsified clauses with one positive literal. Thus, as in the case of \mathcal{F}_n , **RWalkSAT** is more likely to move away from $\vec{0}$ than to approach it, resulting in exponential running time. This completes the sketch of the proof of Theorem 7.2.

8. Open problems.

1. What is the largest Δ for which one can prove RWalkSAT to have polynomial running time on $\mathcal{C} \sim \mathbb{F}_\Delta^n$?
2. What are the statistics of the random variable $\text{Term}(\mathcal{C})$ as a function of the clause density? Does $\text{Term}(\mathcal{C}) < \infty$ have a sharp threshold? Is there a terminator threshold independent of n ? How does $\text{Term}(\mathcal{C})$ correspond to n (number of variables) above density 1.63 (below 1.63 it is linear)?

Appendix A. Proofs. In this section we prove Theorems 4.5 and 4.7. Our starting point is the following theorem and lemma proved implicitly in [10]. The lemma is a slight generalization of Lemma 4.4 in [10], so we provide its proof. (The original Lemma 4.4 of [10] needed only expansion factor of $3/2$, whereas we need a constant fraction more than $3/2$. The proof is essentially the same.)

THEOREM A.1 (see [10]). *For every $\Delta < 1.63$, there exists an integer d such that with high probability for $\mathcal{C} \sim \mathbb{F}_\Delta^n$, $|\mathcal{C}_d| \leq \frac{n}{600\Delta_0^2}$.*

LEMMA A.2 (see [10]). *Let $\Delta_0 = 1.63$. For any constant $\Delta \leq \Delta_0$ with high probability $\mathcal{C} \sim \mathbb{F}_\Delta^n$ is a $(\frac{n}{600\Delta_0^2}, 3/2 + 10^{-3})$ -expander.*

Proof of Lemma A.2. Set $\epsilon = 10^{-3}$. Let A_k be the event that there exists a set of k clauses having less than $3/2 + \epsilon$ variables. Let us bound the probability of these bad events, using a union bound. Let $r = \frac{n}{600\Delta_0^2}$ and $c = 3/2 + \epsilon$. We make use of the following well-known inequality $\binom{n}{k} \leq (\frac{en}{k})^k$:

$$\begin{aligned} \mathbf{P}[Bad] &\leq \sum_{k=1}^r \mathbf{P}[A_k] \leq \sum_{k=1}^r \binom{\Delta n}{k} \cdot \binom{n}{ck} \cdot \left(\frac{ck}{n}\right)^{3k} \\ &\leq \sum_{k=1}^r \left(\frac{e\Delta_0 n}{k}\right)^k \cdot \left(\frac{en}{ck}\right)^{ck} \cdot \left(\frac{ck}{n}\right)^{3k} \\ &\leq \sum_{k=1}^r \left[(e^{1+c} c^{3-c} \Delta_0) \cdot \left(\frac{k}{n}\right)^{2-c} \right]^k \\ &\leq \sum_{k=1}^r \left[37 \cdot \left(\frac{k}{n}\right)^{\frac{1}{2}-\epsilon} \right]^k = o(1), \end{aligned}$$

where the last inequality holds for $r \leq \frac{n}{600\Delta_0^2}$. \square

Notice that if \mathcal{C}_d is a $(|\mathcal{C}_d|, \frac{3}{2} + \epsilon)$ -expander, then every subset of \mathcal{C}_d (including \mathcal{C}_d itself) has at least ϵ unique neighbors (i.e., literals appearing in exactly one clause), and these unique neighbors are pure. Thus, \mathcal{C}_d is $O(\log n)$ -pure. Hence \mathcal{C} is $O(\log n)$ -pure (remember d is a constant), and this proves Theorem 4.5. In order to prove Theorem 4.7 we need the following lemma from [13].

LEMMA A.3 (see [13]). *For all constants $\Delta > 0, c < 2$, there exists some constant $\delta > 0$ such that with high probability $\mathcal{C} \sim \mathbb{F}_\Delta^n$ is a $(\delta\Delta n, c)$ -expander.*

Let δ be the constant promised by Lemma A.3 for $\Delta = 1.63$ and $c = 7/4$. By Theorem 4.5, $|\mathcal{C}_d| \leq n/(600\Delta_0^2)$ for some constant d . By Lemma A.2, \mathcal{C}_d is a $(|\mathcal{C}_d|, \epsilon)$ -boundary expander for some $\epsilon > 0$. Remove an additional d' layers from \mathcal{C} (each containing at least an $\epsilon/3$ fraction of the remaining clauses) so that $|\mathcal{C}_{d+d'}| \leq \delta n$, and by Lemma A.3 this remaining CNF is (with high probability) a $(|\mathcal{C}_{d+d'}|, 7/4)$ -expander. This proves Theorem 4.7.

Acknowledgments. We thank Madhu Sudan for many useful discussions. We thank Bart Selman and Andrew Parkes for valuable information on the empirical results regarding RWalkSAT and Balint Virag for his help with the analysis of martingales. We thank Jon Feldman for providing code for running LP simulations for empirical investigation of the terminator threshold and Jeong Han Kim (via private communication) for allowing us to include the upper bound on the terminator threshold (section 5) in the paper. The second author thanks Rocco Servedio, Salil Vadhan, and Dimitris Achlioptas for helpful discussions. Finally, we thank the anonymous referees for helpful remarks.

REFERENCES

- [1] D. ACHLIOPTAS, *Setting two variables at a time yields a new lower bound for random 3-SAT*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, 2000, pp. 28–37.
- [2] D. ACHLIOPTAS, *Lower bounds for random 3-SAT via differential equations*, Theoret. Comput. Sci., 285 (2001), pp. 159–185.
- [3] D. ACHLIOPTAS AND G. B. SORKIN, *Optimal myopic algorithms for random 3-SAT*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2000, pp. 590–600.
- [4] D. ACHLIOPTAS, P. BEAME, AND M. MOLLOY, *A sharp threshold in proof complexity*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 2001, pp. 337–346.
- [5] N. ALON, M. KRIVELEVICH, AND B. SUDAKOV, *Finding a large hidden clique in a random graph*, Random Structures Algorithms, 13 (1998), pp. 457–466.
- [6] N. ALON AND J. SPENCER, *The Probabilistic Method*, 2nd ed., Wiley, New York, 2000.
- [7] K. AZUMA, *Weighted sums of certain dependent random variables*, Tôhoku Math. J., 19 (1967), pp. 357–367.
- [8] S. BAUMER AND R. SCHULER, *Improving a probabilistic 3-SAT algorithm by dynamic search and independent clause pairs*, in Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing, Lecture Notes in Comput. Sci. 2919, Springer, Berlin, 2004, pp. 150–161.
- [9] E. BEN-SASSON, *Size space tradeoffs for resolution*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 457–464.
- [10] A. BRODER, A. FRIEZE, AND E. UPFAL, *On the satisfiability and maximum satisfiability of random 3-CNF formulas*, in Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, 1993, pp. 322–330.
- [11] M. T. CHAO AND J. FRANCO, *Probabilistic analysis of two heuristics for the 3-satisfiability problem*, Inform. Sci., 51 (1990), pp. 289–314.
- [12] V. CHVÁTAL AND B. REED, *Mick gets some (the odds are on his side)*, in Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science, 1992, pp. 620–627.
- [13] V. CHVÁTAL AND E. SZEMERÉDI, *Many hard examples for resolution*, J. Assoc. Comput. Mach., 35 (1988), pp. 759–768.
- [14] J. M. CRAWFORD AND L. D. AUTON, *Experimental results on the crossover point in random 3-SAT*, Artificial Intelligence, 81 (1996), pp. 31–57.
- [15] E. DANSTIN, A. GOERDT, E. A. HIRSCH, J. KLEINBERG, C. PAPADIMITRIOU, P. RAGHAVAN, AND U. SCHONING, *A deterministic $2 - \frac{2}{k+1}$ algorithm for k -SAT based on local search*, Theoret. Comput. Sci., 223 (1999), pp. 1–72.
- [16] O. DUBOIS, Y. BOUFGHAD, AND J. MANDLER, *Typical random 3-SAT formulae and the satisfiability threshold*, in Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 126–127.
- [17] U. FEIGE, *Relations between average case complexity and approximation complexity*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 534–543.
- [18] A. FLAXMAN, *A spectral technique for random satisfiable 3CNF formulas*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 357–363.
- [19] A. FRIEZE AND S. SUEN, *Analysis of two simple heuristics for random instances of k -SAT*, J. Algorithms, 20 (1996) pp. 312–355.
- [20] M. T. HAJIAGHAYI AND G. B. SORKIN, *The Satisfiability Threshold of Random 3-SAT Is at Least 3.52*, IBM Research Report RC22942, 2003, submitted.
- [21] T. HOFMEISTER, U. SCHONING, R. SCHULER, AND O. WATANABE, *Probabilistic 3-SAT algorithm*

- further improved*, in Proceedings of the 19th International Symposium on Theoretical Aspects of Computer Science, 2002, pp. 193–202.
- [22] K. IWAMA AND S. TAMAKI, *Improved bounds for 3-SAT*, in Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 321–322.
 - [23] U. FEIGE AND R. KRAUTHGAMER, *Finding and certifying a large hidden clique in a semi-random graph*, Random Structures Algorithms, 16 (2000), pp. 195–208.
 - [24] E. FRIEDGUT, *Sharp thresholds of graph properties, and the k -sat problem*, J. Amer. Math. Soc., 12 (1999), pp. 1017–1054.
 - [25] Z. FÜREDI, *Random polytopes in the d -dimensional cube*, Discrete Comput. Geom., 1 (1986), pp. 315–319.
 - [26] P. HALL, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26–30.
 - [27] W. HEOFFDING, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc., 58 (1963), pp. 13–30.
 - [28] M. JERRUM AND G. B. SORKIN, *Simulated annealing for graph bisection*, in Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, 1993, pp. 94–103.
 - [29] A. KAPORIS, L. M. KIROUSIS, AND E. G. LALAS, *The probabilistic analysis of a greedy satisfiability algorithm*, in Proceedings of the 10th Annual European Symposium on Algorithms, Rome, Italy, 2002.
 - [30] S. KIRKPATRICK, C. D. GELATT, AND M. P. VECCHI, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.
 - [31] M. LUBY, M. MITZENMACHER, AND A. SHOKROLLAHI, *Analysis of random processes via and-or tree evaluation*, in Proceeding of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, pp. 364–373.
 - [32] M. MÉZARD, G. PARISI, AND R. ZECCHINA, *Analytic and algorithmic solution of random satisfiability problems*, Science, 297 (2002), pp. 812–815.
 - [33] M. MÉZARD AND R. ZECCHINA, *Random k -satisfiability: From an analytic solution to an efficient algorithm*, Phys. Rev. E, 66 (2002), 056126.
 - [34] R. MOTWANI AND P. RAGHAVAN, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.
 - [35] C. H. PAPADIMITRIOU, *On selecting a satisfying truth assignment*, in Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science, 1991, pp. 163–169.
 - [36] A. J. PARKES, *private communication*.
 - [37] D. ROLF, *3-SAT in $RTIME(O(1.32793^n))$ —Improving Randomized Local Search by Initializing Strings of 3-Clauses*, ECCC report TR03-054, 2003.
 - [38] U. SCHONING, *A probabilistic algorithm for k -SAT and constraint satisfaction problems*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science, 1999, pp. 410–414.
 - [39] B. SELMAN, *private communication*.
 - [40] B. SELMAN AND H. KAUTZ, *Local search strategies for satisfiability testing*, in Proceedings of the Second DIMACS Challenge on Cliques, Coloring, and Satisfiability, AMS, Providence, RI, 1993, pp. 521–532.
 - [41] B. SELMAN, H. LEVESQUE, AND D. MITCHELL, *A new method for solving hard satisfiability problems*, in Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92), San Jose, CA, 1992, pp. 440–446.