Delaying satisfiability for random 2SAT

Alistair Sinclair[1] and Dan Vilenchik[2]

[1] Computer Science Division, University of California, Berkeley CA 94720–1776*
[2] Department of Mathematics, University of California, Los Angeles, CA 90095**

Abstract. Let $(C_1, C_1'), (C_2, C_2'), \ldots, (C_m, C_m')$ be a sequence of ordered pairs of 2CNF clauses chosen uniformly at random (with replacement) from the set of all $4\binom{n}{2}$ clauses on n variables. Choosing exactly one clause from each pair defines a probability distribution over 2CNF formulas. The choice at each step must be made on-line, without backtracking, but may depend on the clauses chosen previously. We show that there exists an on-line choice algorithm in the above process which results *whp* in a satisfiable 2CNF formula as long as $m/n \leq (1000/999)^{1/4}$. This contrasts with the well-known fact that a random m-clause formula constructed without the choice of two clauses at each step is unsatisfiable *whp* whenever $m/n > 1$. Thus the choice algorithm is able to *delay* satisfiability of a random 2CNF formula beyond the classical satisfiability threshold. Choice processes of this kind in random structures are known as "Achlioptas processes." This paper joins a series of previous results studying Achlioptas processes in different settings, such as delaying the appearance of a giant component or a Hamilton cycle in a random graph. In addition to the on-line setting above, we also consider an off-line version in which all m clause-pairs are presented in advance, and the algorithm chooses one clause from each pair with knowledge of all pairs. For the off-line setting, we show that the two-choice satisfiability threshold for k-SAT for any fixed k coincides with the standard satisfiability threshold for random $2k$-SAT.

1 Introduction

The random graph process, introduced by Erdős and Rényi in the 1960's, begins with an empty graph on n vertices and adds a single new edge to the graph in each round $i = 1, \ldots, m$. Each new edge is chosen uniformly at random from all unchosen edges. The resulting distribution over graphs is commonly denoted $G_{n,m}$. One of the most fundamental random graph properties to be studied is the emergence of a giant component: at what density m/n does a connected component of size $\Omega(n)$ first appear? A classical result by Erdős and Rényi [9]

* Email: sinclair@cs.berkeley.edu. Supported in part by NSF grant CCF-0635153 and by a UC Berkeley Chancellor's Professorship.

** Email: vilenchik@math.ucla.edu. This work was done while the author was a postdoctoral researcher at UC Berkeley, supported by NSF grants CCF-0635153 and DMS-0528488.

asserts that for $m/n = c$, $c > 1/2$ a constant, a random graph with m edges will have a unique giant component *whp*[3].

Inspired by the celebrated "power of two choices" phenomenon for balls-and-bins [2] (n balls are randomly thrown into n bins, each ball inspecting two random bins and choosing the less heavily loaded of the two, resulting in a significant decrease in the maximum bin load), Dimitris Achlioptas posed the following question for the random graph process. Suppose that edges arrive in pairs, i.e., in round i the pair (e_i, e'_i) appears, and one of these edges is chosen for inclusion in the graph. The decision is to be made on-line, possibly based on the history of the process, but with no backtracking. Does there exist an algorithm A that delays the appearance of the giant component? Frieze and Bohman answered this question positively [3], describing an on-line algorithm A whose greedy rule postpones the appearance of the giant component until $m/n \geq 0.53$. Spencer and Wormald [18] improved this result to $m/n \leq 0.83$. An upper bound of $m/n = 0.964$ for every on-line algorithm was proved in [4].

Quite a few subsequent papers have addressed various other facets of the above model, including speeding up the appearance of the giant component, delaying the appearance of certain fixed subgraphs, speeding up the appearance of a Hamilton cycle, and so on (see, e.g., [4, 10, 6, 16, 17]).

Another class of random structures that has been widely studied is that of random 2CNF formulas. To generate a random 2CNF formula with m clauses over n variables, choose uniformly at random m clauses out of all $4\binom{n}{2}$ possible ones. We call the resulting distribution $F_{n,m}$. Goerdt [13], and independently Chvátal and Reed [8], showed that whenever $m/n < 1$ a random $F_{n,m}$ formula is *whp* satisfiable, while if $m/n > 1$ it is *whp* unsatisfiable. In this paper we consider an Achlioptas process for random 2CNF formulas. Specifically, we answer the question whether one can delay the sat/unsat threshold if at each step two random clauses are available to choose from. As far as we are aware, this is the first time that an Achlioptas process for random formulas has been studied.

Formally, we examine the following process: at each round $i = 1, \ldots, m$, generate two random clauses independently and uniformly at random (with replacement) out of all $4\binom{n}{2}$ possible clauses; then choose one of the two to be included in the formula. The decision is to be made on-line, without backtracking, but possibly dependent on the clauses seen so far. We note that, to avoid technical complications, our distribution is slightly different from the Achlioptas process for $F_{n,m}$ because in our distribution some clauses may appear twice. However, a simple calculation shows that the number of pairs of identical clauses among the $2m = \Theta(n)$ clauses appearing in the process is $o(n)$. Disregarding steps involving such clauses we get exactly the Achlioptas process for $F_{n,m-o(n)}$. But removing $o(n)$ clauses from the formula doesn't change our result as our advantage over the threshold will be of order $\Theta(n)$.

A random $F_{n,m}$ formula can be viewed in an obvious way as a random $G_{2n,m}$ graph on the set of $2n$ literals (i.e., variables and their negations), in which a

[3] Throughout, we shall take the phrase *whp* (with high probability) to mean "with probability tending to 1 as $n \to \infty$."

clause $(\ell \vee \ell')$ is translated into an edge between ℓ and ℓ'. With this picture, one might naively think that an application of the Bohman-Frieze greedy rule [3] for delaying the giant component suffices to delay the sat/unsat threshold for 2SAT. After all, until the emergence of the giant component the connected components of $F_{n,m}$ are simple (mostly trees, plus a few unicyclic components), and hence one may think that the formula is likely to be satisfiable.

However, this intuition turns out to be false. A recent result of Kravitz [15] implies that, if one only tampers with the degrees of literals in the random formula (leaving the parities of the variables uniformly random), then once the average degree exceeds 1 the formula will be unsatisfiable *whp*. The underlying reason, of course, is that the satisfiability of a 2CNF formula is determined not by the literal graph above, but by the *implication graph*. The vertices of the implication graph are again the $2n$ literals; however, for every clause $(\ell \vee \ell')$ in the formula, two *directed* edges are added to the implication graph: $\bar{\ell} \to \ell'$ and $\bar{\ell'} \to \ell$. As is well known [1], a 2CNF formula is satisfiable iff in its implication graph there is no variable x such that x and $\bar{x}$ belong to the same strongly connected component. Thus the fact that the literal graph has a simple structure does not exclude the possibility of contradictory cycles in the implication graph. This tells us that any rule we use to determine which clause to choose at each step must take into account the parities of the variables, and thus the Achlioptas process for 2SAT will depart from the realm of pure random graph structure that has been explored in previous such results.

Before stating our result, let us mention that alongside the on-line version, an *off-line* version of the Achlioptas process has also been studied. In the off-line version (formulated for random 2SAT), m random pairs of clauses are generated; then an algorithm chooses one clause from each pair, given full information about all the pairs. For the analogous off-line version of the random graph process, Bohman and Kim [5] prove an exact threshold for avoiding the giant component, whose value is roughly $m = 0.97677n$. As we shall see shortly (Theorem 1), we are able to obtain the exact threshold for the off-line k-SAT process.

1.1 Our contributions

Let us first state our threshold result for the off-line version. Observe that it is not *a priori* clear that such a process will have a threshold, in the sense of [11]. However, we show that it does have a threshold, and that this threshold coincides with that of random $2k$-SAT. In what follows, we denote by d_k the satisfiability threshold for random k-SAT. (This threshold exists by virtue of [11]; note that d_k may depend on n as well as on k).

Theorem 1. *Given m pairs $(C_1, C_1'), (C_2, C_2'), \ldots, (C_m, C_m')$ of random k-SAT clauses over n variables, if $m/n < d_{2k}$ there exists* whp *an off-line choice of one clause per pair so that the resulting formula is satisfiable. If $m/n > d_{2k}$* whp *every such choice will result in an unsatisfiable formula.*

The proof uses a somewhat similar idea to that used in [4] in the context of avoiding the giant component in the random graph process, and can be found in Section 2.

We turn now to the on-line case, which is rather more challenging. Of course, the off-line threshold provides an upper bound for the on-line setting, so we immediately deduce from Theorem 1 that no on-line choice algorithm can delay satisfiability beyond $m/n = d_{2k}$. In particular, for 2SAT this upper bound is d_4, which is predicted experimentally to be about 9.25 [14]. A rigorous upper bound on d_4 is obtained by plugging $k = 4$ into the first moment bound $2^k \ln 2$, giving 11.09. Thus Theorem 1 proves that no on-line choice algorithm can delay satisfiability for 2SAT beyond $m/n = 11.09$.

What about the more interesting question of a lower bound? Is it possible to delay satisfiability beyond the threshold at all? We are able to answer this question affirmatively for the case $k = 2$, and this is the main technical contribution of our paper.

Before presenting our on-line algorithm we give a few definitions. We denote the set of variables by $x_1, x_2, \ldots, x_n$. Throughout we use ℓ to denote a literal (i.e., $\ell = x_i$ or $\ell = \bar{x}_i$), and $\bar{\ell}$ its negation.

Definition 1. *A clause $C = (\ell \vee \ell')$ is **bad** with respect to a set F of clauses if either $\bar{\ell}$ appears in F or $\bar{\ell}'$ appears in F. A clause is **good** if it is not bad.*

The procedure in Figure 1 specifies our on-line choice rule.

For each round $i = 1, \ldots, m$ do:

1. Pick two clauses C_1, C_2, with replacement, independently at random out of all $4\binom{n}{2}$ possible clauses.
2. Set $F_i = \{D_1, D_2, \ldots, D_{i-1}\}$, where D_j is the clause chosen in round j.
3. If C_1 is good with respect to F_i, choose it, otherwise choose C_2.

Fig. 1. Generating a random 2CNF instance

Our main result is formally stated in the following theorem, which says that the above choice rule succeeds in delaying the sat/unsat phase transition for random 2CNF formulas by a constant factor:

Theorem 2. *Let F be a random 2CNF formula generated by the procedure in Figure 1. If $m/n \leq (1000/999)^{1/4}$ then F is* whp *satisfiable.*

Experimental results predict that the right critical value of m/n when using the above algorithm is approximately 1.2. However, to keep the analysis clear we did not try to optimize the constant. (We do not claim that simply optimizing over the constants in our proof will yield the value 1.2.)

One may also try other greedy rules, similar in flavor to the one we use, and get different threshold values. The best experimental threshold we achieved with a simple rule was approximately 1.5; this is discussed in more detail in Section 7.

Another way of extending the result is the following: suppose that in each round one is allowed to choose from T clauses rather than just from two. Our analysis easily implies that, using the same rule (but now choose C_T only if $C_1, \ldots, C_{T-1}$ are all bad), the sat/unsat threshold scales as β^T for some fixed $\beta > 1$. We omit the details.

We also remark that the techniques we develop here to prove Theorem 2 may be applicable in other settings. One such setting is delaying the threshold for the pure-literal procedure in random 3SAT formulas. Broder et al [7] showed a tight threshold of 1.63 for this model. We conjecture that using our techniques it is possible to show that, given a choice of two clauses in each round, one can delay the threshold for the pure literal procedure in 3SAT beyond this point. More details are given in Section 7.

Finally, let us state a result concerning k-SAT for $k = \omega(\log n)$.

Theorem 3. *Given a choice of two clauses in the Achlioptas process for random k-SAT with $k = \omega(\log n)$, there exists an on-line algorithm that delays the satisfiability threshold by a factor of* $0.99/\ln 2$.

Note that the sat/unsat threshold for k-SAT is (for any k) at most $m/n = 2^k \ln 2$ (this follows from a simple first moment calculation). Actually, for $k = \omega(\log n)$ this upper bound is tight [12]. The proof of Theorem 3 is self-contained and short, and uses a different (even simpler) choice rule from that in Figure 1.

The remainder of the paper is organized as follows. In the next section, we give the short proof of the off-line threshold, Theorem 1. Then we turn to the proof of our main result, Theorem 2: in Section 3 we give an outline of the proof, then in Section 4 we establish some useful properties of the distribution induced by the algorithm, and finally we use these properties to prove Theorem 2 in Section 5. Section 6 gives the short proof of Theorem 3. We conclude with a brief discussion in Section 7.

2 The off-line setting: Proof of Theorem 1

Consider m pairs of random k-SAT clauses $(C_1, C_1'), (C_2, C_2'), \ldots, (C_m, C_m')$, and from each pair generate a $2k$-SAT clause by setting $D_i = C_i \vee C_i'$. Set $F^* = D_1 \wedge D_2 \wedge \cdots \wedge D_m$. Observe that the $2k$-SAT formula F^* may contain some "illegal" clauses in which some variable repeats. As we shall see, this is a technicality that is readily overcome. Hence, in what follows we allow such clauses.

The following is a general lemma that does not assume any randomness in the choice of clauses.

Lemma 1. *F^* is satisfiable iff there exists a choice of a satisfiable formula in $(C_1, C_1'), (C_2, C_2'), \ldots, (C_m, C_m')$.*

Proof. If there exists a choice of satisfiable formula, concatenating the unchosen clause in every pair obviously lifts this to a satisfiable $2k$-SAT formula. So F^* is satisfiable. Conversely, if F^* is satisfiable, let φ be a satisfying assignment and consider the following choice rule: evaluate $D_i = C_i \vee C'_i$ under φ, and choose the clause (C_i or C'_i) that contains at least one true literal under φ (breaking ties arbitrarily). Since φ satisfies F^*, every D_i contains such a choice. Clearly, φ satisfies the k-SAT formula that we chose. □

As mentioned above, the distribution of F^* does not coincide exactly with that of $F_{2k,n,m}$, random $2k$-SAT. The reason is that, for some i, C_i and C'_i may share a variable, so D_i will be an illegal $2k$-SAT clause. However, the following lemma asserts that the satisfiability threshold for the two distributions is the same (both these distributions have a threshold by [11]).

Lemma 2. *Let F^* be distributed as above. Let d^*_{2k} be the satisfiability threshold for that distribution, and let d_{2k} be the threshold for $F_{n,m,2k}$. Then $d^*_{2k} = d_{2k}$ for any fixed k.*

Proof. First let us estimate the probability of a shared variable in a pair (C_i, C'_i). This probability is easily seen to be $O(k^2/n)$. Let T be a random variable that counts the number of such pairs. Since the regime that is relevant for us is $m = O(2^{2k}n)$, we have $E[T] = O(k^2 2^{2k}) = O(1)$ (as k is fixed). Note that T is binomially distributed, and so with constant probability $T = 0$. Hence if for example we assume that $d^*_{2k} < d_{2k}$, then for $d^*_{2k} < m/n < d_{2k}$ we get that with some constant probability, the resulting random formula F^* is a random $F_{n,m,2k}$ formula, and hence satisfiable *whp*. Thus in turn, with constant probability F^* is satisfiable above the threshold d^*_{2k}, which contradicts the definition of a threshold. The same argument shows that $d^*_{2k} > d_{2k}$ cannot occur. □

Theorem 1 follows immediately from the above two lemmas.

3 The on-line setting: Proof outline

As we have already mentioned, the satisfiability of a 2CNF formula F is determined by certain structures in its implication graph $G(F)$: namely, directed paths from x to $\bar{x}$ and from $\bar{x}$ to x. The formula is unsatisfiable iff for some x paths of both these types exist.

We may view a simple directed path p from x to $\bar{x}$ in $G(F)$ as a sequence of clauses $C_1, C_2, \ldots, C_t$, where $C_1 = (\bar{x} \vee \ell_1)$, $C_i = (\bar{\ell}_{i-1} \vee \ell_i)$ for $2 \le i \le t-1$, and $C_t = (\bar{\ell}_{t-1} \vee \bar{x})$. Observe that every variable in p appears twice, and except for the variable x, each variable appears both positively and negatively.

Our proof is a first moment calculation, estimating the number of pairs of simple paths ($x \rightsquigarrow \bar{x}$, $\bar{x} \rightsquigarrow x$) in the implication graph. We will show that this number is $o(1)$ for our choice of m/n, thus proving Theorem 2.

The main challenge is to estimate $\Pr[C_1, C_2, \ldots, C_t]$, the probability of occurrence of a sequence of clauses as above. Note that in the standard random 2SAT

model $F_{n,m}$ this is straightforward: $\Pr[C_1, C_2, \ldots, C_t] \sim \left(m/4\binom{n}{2}\right)^t$. However, under the new distribution we will need to do much better in order to achieve a first moment of $o(1)$ for values of m with $m/n > 1$. In fact, the key point is that our choice rule for clauses "punishes" paths in the implication graph, thus enabling us to delay the sat/unsat threshold.

The key observation in the analysis is the following: given that a clause $(\ell' \vee \ell)$ has been included already, then if $(\bar{\ell} \vee \ell'')$ is to be included later, it will have to be as the *second* clause (C_2 in the procedure in Figure 1): it cannot be included as the first clause because it is "bad" by our definition. This in turn means that the first clause, C_1, must itself be bad in order to allow us to choose C_2. This fact reduces the probability of some clauses in the path (at least half of them, as we shall see), which allows us to achieve satisfiability even when $m/n > 1$.

One challenge in the analysis is the fact that the choice of a new clause depends on the history of clauses chosen so far. Thus instead of dealing with a standard "product" space[4] like $F_{n,m}$, we have to analyze a more complicated conditioned probability space.

4 Properties of the distribution

In this section we establish some properties of the distribution over formulas induced by our choice rule. We will use these in our proof of Theorem 2 in the next section.

Definition 2. *Two clauses C and C' **threaten** each other if there exists a variable x that appears positively in one and negatively in the other.*

Proposition 1. *Consider a simple path of length t in the implication graph $G(F)$. In every ordering π of the clauses on the path, at least $(t/2) - 1$ clauses are threatened by clauses that appear before them in π.*

Proof. Fix an arbitrary ordering π of the clauses; let T be the set of clauses that are threatened by some clause before them in π, and N the other clauses of the path. Every clause, except possibly for the first and last clauses of the path, threatens exactly two other clauses, and every clause in the path is threatened by at most two clauses. Also, no clause in N can threaten another clause in N, since if so the one appearing first in π would threaten the second clause and the latter would be in T ("threatening" is a symmetric relation). To conclude, the clauses in N threaten at least $2(|N| - 2)$ clauses in T (we assume the worst case for us: both the first and last clauses are in N), each of which was counted at most twice. All in all, $|T| \geq 2(|N| - 2)/2 = |N| - 2$. Further, $|T| + |N| = t$, and therefore $|T| \geq (t/2) - 1$ as desired. □

Consider now a possible (simple) path in $G(F)$, corresponding to the clauses $C_1, \ldots, C_t$. Our next task is to derive an upper bound on the probability that

[4] Technically $F_{n,m}$ is not a product space, but standard methods allow it to be viewed as one.

this set of clauses is chosen by the algorithm. This probability depends on the order in which the clauses are chosen, which motivates the following definition. Here π is an arbitrary permutation of the clause labels $1, \ldots, t$.

Definition 3. *The clauses $C_1, \ldots, C_t$ are **chosen according to** π if the algorithm described in Figure 1 chooses these clauses in the order specified by the permutation π. The clauses are chosen according to π **in rounds** $k_1, \ldots, k_t$ if in addition the algorithm chooses clause C_i in round k_i. (Note that the k_i must **respect** π in the sense that $k_i < k_j$ iff $\pi(i) < \pi(j)$.)*

Let us fix a simple path in $G(F)$ and a corresponding set of clauses $C_1, \ldots, C_t$, as well as an associated permutation π and a set of rounds $k_1, \ldots, k_t$ respecting π. Let $\mathcal{A}_i^{(k)}$ denote the event that clause C_i is chosen in round k, and $\mathcal{A}_i^{(<k)}$ the event that clause C_i is chosen in some round $k' < k$. Then we have

$$\begin{aligned} &\Pr[C_1, C_2, \ldots, C_t \text{ are chosen according to } \pi \text{ in rounds } k_1, \ldots, k_t] \qquad (1) \\ &\quad = \prod_{i=1}^{t} \Pr\Big[\mathcal{A}_i^{(k_i)} \;\Big|\; \bigcap_{j:\pi(j)<\pi(i)} \mathcal{A}_j^{(<k_i)}\Big]. \end{aligned}$$

To analyze the conditional probabilities appearing in (1), we partition the clauses into three categories as follows:

1. the first and last clauses of the path (as they appear in the implication graph, not in π);
2. the inner clauses which are threatened by a clause that precedes them in π;
3. the inner clauses which are not threatened by a clause that precedes them in π.

We proceed to bound the conditional probability for a clause in each category.

Proposition 2. *For any fixed round $k \le m$,*

$$\Pr\Big[\mathcal{A}_i^{(k)} \;\Big|\; \bigcap_{j:\pi(j)<\pi(i)} \mathcal{A}_j^{(<k)}\Big] \le \frac{2}{4\binom{n}{2}}.$$

The proof is immediate: since we are conditioning only on the past, each candidate clause in round k is uniformly distributed over the set of all $4\binom{n}{2}$ clauses, and therefore $2/4\binom{n}{2}$ is an upper bound on C_i even being a candidate in round k.

In what follows, we use $|\pi|$ to denote the number of clauses ordered by π. (Thus $|\pi| = t$ for a sequence of clauses $C_1, \ldots, C_t$.)

Proposition 3. *For a fixed round $k \le m$, $|\pi| = o(n)$, and $\alpha^* = 969/970$,*

$$\Pr\Big[\mathcal{A}_i^{(k)} \;\Big|\; \bigcap_{j:\pi(j)<\pi(i)} \mathcal{A}_j^{(<k)} \cap \ (C_i \textit{ is threatened by some preceding clause in } \pi)\Big] \le \frac{\alpha^*}{4\binom{n}{2}}.$$

Proof. In order for C_i to be chosen in round k, given that all clauses that precede it in π have been previously chosen and at least one of these threatens C_i, two events need to occur: (a) the first clause in round k is bad; and (b) C_i appears as the second clause in round k. Since the conditioning is only on the past, these two events are independent. The probability of the second event is trivially $1/4\binom{n}{2}$. As for the first event, with probability $1 - o(1)$ the two literals of the first candidate clause will not occur in the set of clauses we condition upon (because, by assumption, they span only $o(n)$ variables). So we may assume this is the case. We now derive a lower bound on the probability of the first clause being good. Suppose w.l.o.g. that this clause is $C = (x \vee y)$. Let us calculate the probability that either x or y appears more than four times up to round k. The expected number of appearances of each such variable up to round k is at most $2k/n \leq 2m/n$. (Here we are using the fact that the distribution induced by our choice rule is symmetric for all variables that do not appear in the clauses conditioned upon.) It is also easy to see that variable appearances are negatively correlated (conditioning on x already appearing in a chosen clause, if x is to appear again then it will be with at most the probability of the first appearance, as now one or both of its parities are "punished").

Using the Chernoff bound (which we can do due to negative correlation), the probability that a variable appears four times is smaller than 0.467, and therefore with probability at least $1 - 2 \times 0.467$, both x and y appear at most three times. Let us assume this is the case. The worst case for us is that both actually appear three times. Next observe that the configurations x, x, x and $\bar{x}, \bar{x}, \bar{x}$ are at least as likely as any other (again, by our choice rule). Therefore, with probability at least $2/2^3 = 1/4$, the appearances of x are either all negative or all positive (that is, x appears in pure form). With probability at least $1/16$ this is true for both x and y (conditioning on x's configuration will not make y's non-pure configurations more likely than its pure ones). To conclude, with probability at least $(1 - 2 \times 0.467)/16$ both x and y appear in pure form. In that case, with probability $1/4$ the clause C is good (x and y appear in C with the correct parities). Overall, then, C is good with probability at least $(1 - 2 \cdot 0.467)/(16 \cdot 4) > 1/970$. To conclude, for sufficiently large n,

$$\Pr\Big[\mathcal{A}_i^{(k)} \;\Big|\; \bigcap_{j:\pi(j)<\pi(i)} \mathcal{A}_j^{(<k)} \cap \ (C_i \text{ is threatened by some preceding clause in } \pi)\Big] \leq \frac{969}{970} \cdot \frac{1}{4\binom{n}{2}}. \qquad \square$$

Proposition 4. *For a fixed round $k \leq m$ and $|\pi| \leq r$,*

$$\Pr\Big[\mathcal{A}_i^{(k)} \;\Big|\; \bigcap_{j:\pi(j)<\pi(i)} \mathcal{A}_j^{(<k)} \cap (C_i \textit{ is an inner clause not threatened by any preceding clause in } \pi)\Big] \leq \frac{1 + O(r/n)}{4\binom{n}{2}}.$$

Proof. Since C_i is not threatened by any clause we condition upon, it can be chosen as either the first or the second clause in its pair. Let $\mathcal{N}$ be the event "C_i

is *not* threatened by any preceding clause in π"; then (suppressing throughout the proof the conditioning on $\bigcap_{j:\pi(j)<\pi(i)} \mathcal{A}_j^{(<k)}$), we have

$$\begin{aligned}\Pr[\mathcal{A}_i^{(k)}|\mathcal{N}] = \Pr[&C_i \text{ is chosen as first clause } |\mathcal{N}] \\ &+ \Pr[C_i \text{ is chosen as second clause } |\mathcal{N}].\end{aligned}$$

Say w.l.o.g. $C_i = (\ell_1 \vee \ell_2)$. For C_i to be chosen as the first clause in round k, it must be the case that $\bar{\ell}_1$ and $\bar{\ell}_2$ have not appeared in a chosen clause before round k (otherwise C_i is bad). Therefore,

$$\Pr[C_i \text{ chosen as first clause } |\mathcal{N}] = \frac{\Pr[\bar{\ell}_1,\, \bar{\ell}_2 \text{ don't appear before round } k\ |\mathcal{N}]}{4\binom{n}{2}}. \tag{2}$$

The denominator, $4\binom{n}{2}$, accounts for the probability of the clause $(\ell_1 \vee \ell_2)$ actually appearing. Observe that in this case ℓ_1 and ℓ_2 do not appear in any of the clauses we condition upon (otherwise C_i is threatened by a clause in that set as C_i is an inner clause).

For C_i to be chosen as the second clause in round k, it must be the case that the first clause is bad and C_i appears as the second clause. Again, these two events are independent (as the conditioning is only on the past, and at each round the two candidate clauses are chosen independently). With probability $O(r/n)$, some variable of the first clause appears in the set of clauses conditioned upon. (This is because there are r such clauses by assumption, involving at most $2r$ variables.) Assume this is not the case, and let ℓ, ℓ' be the two literals in the first candidate clause. Then the probability that the clause is bad is

$$\begin{aligned}&\Pr[\bar{\ell} \text{ or } \bar{\ell}' \text{ appears before round } k\ |\mathcal{N}] \\ &\quad = 1 - \Pr[\bar{\ell},\bar{\ell}' \text{ don't appear before round } k\ |\mathcal{N}].\end{aligned}$$

Observe that

$$p^* \equiv \Pr[\bar{\ell},\, \bar{\ell}' \text{ don't appear before round } k\ |\mathcal{N}]$$

is exactly the numerator in Equation (2): in both cases we ask for the probability that two literals (whose variables do not appear among the clauses conditioned upon) have not appeared before round k. By symmetry, the identity of the literals doesn't matter, and therefore the two expressions are equal. To conclude,

$$\Pr[\mathcal{A}_i^{(k)}|\mathcal{N}] \le \frac{p^*}{4\binom{n}{2}} + \frac{O(r/n) + 1 - p^*}{4\binom{n}{2}} = \frac{1 + O(r/n)}{4\binom{n}{2}}. \qquad \square$$

5 Proof of Theorem 2

Recall that, to prove Theorem 2, it suffices to exclude the existence of a directed path from x to $\bar{x}$ and from $\bar{x}$ to x (for any x) in the implication graph $G(F)$ of the formula F constructed by our algorithm. We follow the same approach

as that in [8] for proving the threshold for random 2SAT; the main challenge in our case is to use the "power of two choices" to get a tighter bound on the appearance of a fixed path, using the bounds we derived in the previous section.

We branch into two cases, depending on the length of the path. We will start with the case where one of the paths $x \rightsquigarrow \bar{x}$ or $\bar{x} \rightsquigarrow x$ is of length, say, at least $\log^2 n$. Take a (simple) prefix of length $t = \log^2 n$ of some such path, consisting of clauses $C_1, C_2, \ldots, C_t$. Let us bound the probability of such a prefix occurring:

$$\begin{aligned} &\Pr[\text{prefix of length } t] \qquad (3)\\ &\leq \binom{n}{t+1} \cdot 2^{t+1} \cdot (t+1)! \cdot \sum_{\pi} \Pr[C_1, C_2, \ldots, C_t \text{ are chosen according to } \pi]. \end{aligned}$$

The first factor is the number of ways to choose the $t+1$ variables, the second counts their parities, and the last factor accounts for the probability the clauses are actually chosen, summing over all possible orderings in which they are chosen.

We may bound this final summation over π by $\binom{m}{t}$ (for the number of ways of choosing the rounds) times an upper bound on

$$\Pr[C_1, C_2, \ldots, C_t \text{ are chosen according to } \pi \text{ in rounds } k_1, \ldots, k_t]$$

over all choices of $k_1, \ldots, k_t$. This we obtain via Equation (1), using the bounds on the conditional probabilities derived in Propositions 2, 3 and 4. Note first that, by Proposition 1, at least $(t/2) - 1$ clauses are threatened in every ordering π. To bound the conditional probabilities for these clauses we use Proposition 3. Otherwise, except for the first and last clauses of the prefix, every clause is an inner clause whose probability we bound using Proposition 4. For the first and last clauses we use Proposition 2. Putting all this together yields

$$\begin{aligned} &\sum_{\pi} \Pr[C_1, C_2, \ldots, C_t \text{ are chosen according to } \pi] \\ &\qquad \leq t! \binom{m}{t} \left(\frac{\alpha^*}{4\binom{n}{2}} \right)^{(t/2)-1} \cdot \left(\frac{1 + O(t/n)}{4\binom{n}{2}} \right)^{(t/2)-1} \cdot \left(\frac{2}{4\binom{n}{2}} \right)^{2}. \end{aligned}$$

Plugging this into Equation (3) and simplifying yields

$$\begin{aligned} &\Pr[\text{prefix of length } t] \\ &\leq m^t n^{t+1} \cdot 2^t \cdot \left(\frac{1}{4\binom{n}{2}} \right)^{t} \cdot (\alpha^*)^{(t/2)-1} \cdot O(1) \leq O(n) \cdot \left(\frac{m}{n} \right)^{t} \cdot (\alpha^*)^{t/2-1}. \end{aligned}$$

By our choice of $m/n \leq (1000/999)^{1/4}$, and the fact that $\alpha^* = 969/970$, for sufficiently large t this last expression is at most $O(n)\beta^t$ for some fixed $\beta < 1$. Since $t \geq \log^2 n$, $\Pr[\text{prefix of length } t] = n^{-\Omega(\log n)}$. Finally, there are at most n ways to choose t (the length of the simple path), so the probability that any path of length greater than $\log^2 n$ exists in the implication graph is $o(1)$.

Let us now move to the case of short paths. We shall bound the probability that, for any variable x, there exists a short path from x to $\bar{x}$ and from $\bar{x}$ to x (recall that we only consider simple paths). Denote the paths from x to $\bar{x}$ and from $\bar{x}$ to x by p_1 and p_2 respectively, and let their respective lengths be t_1 and t_2. Suppose w.l.o.g. that $t_1 \geq t_2$. The path p_2 may contain clauses from p_1: say, s segments of total length r from p_1. The number of variables in p_1 is t_1, and p_2 further introduces $t_2 - 1 - r - s$ new variables. (We do the variable counting in p_2 as follows: for every clause in p_2, we count the variable that appears first along the path. However, some variables were counted already in p_1 and need to be subtracted. We subtract one for the first clause of p_2 (x was already counted), then we subtract one for every shared clause, and one for every clause after a segment ends since that variable was counted in p_1.) As for choosing the segments, there are at most t_1^{2s} ways to choose the shared segments in p_1 (starting and ending points) and at most t_2^s ways to choose their starting points in p_2. Once the segments and starting points are fixed, there are $(t_2 - 1 - r - s)!$ ways to arrange the new variables in p_2, and this completely determines p_2.

Call such a path a (t_1, t_2, s, r)-*path*. We now bound the probability of any such path occurring, for all possible choices of x. In light of the above observations, we have

$$\begin{aligned}\Pr[\ (t_1,t_2,s,r)\text{-path}\ ] \leq\ & \binom{n}{t_1} \cdot 2^{t_1} \cdot t_1! \cdot \binom{n}{t_2 - 1 - r - s} \cdot t_1^{2s} \cdot t_2^s \cdot (t_2 - 1 - r - s)! \cdot 2^{t_2 - r - s} \\ & \times \sum_{\pi} \Pr[C_1, C_2, \ldots, C_{t_1 + t_2 - r} \text{ are chosen according to } \pi].\end{aligned}$$

We may bound the summation over π in analogous fashion to the case of long paths above, with a factor $\binom{m}{t_1+t_2-r}$ to choose the rounds in which the clauses along the paths are chosen. By Proposition 1, at least $(t_1/2) - 1$ clauses in any ordering of p_1 are threatened by a clause that precedes them, and this of course remains true if we order a superset of those clauses. We can bound the conditional probabilities for these clauses using Proposition 3. For the four clauses containing x and $\bar{x}$ we use Proposition 2. For the rest of the clauses we use Proposition 4 (some of these may be threatened as well, but we only want an upper bound). The calculation now proceeds similarly to the single (long) path case, giving

$$\Pr[\ (t_1,t_2,s,r)\text{-path}\ ] \leq O(1) \cdot \left(\frac{m}{n}\right)^{t_1+t_2-r} \cdot (\alpha^*)^{(t_1/2)-1} \cdot \left(\frac{t_1^2 t_2}{n}\right)^s \cdot \frac{1}{n}.$$

Observe that $t_1^2 t_2 \leq \log^6 n$, so $\left(\frac{t_1^2 t_2}{n}\right)^s = O(1)$. Summing over the at most $\log^4 n$ ways to choose s and r gives

$$O\left(\frac{\log^4 n}{n}\right) \cdot \left(\frac{m}{n}\right)^{t_1+t_2} \cdot (\alpha^*)^{(t_1/2)-1}.$$

Summing again over t_1, t_2 such that $t_2 \leq t_1 \leq \log^2 n$, we can bound the probability of a cycle consisting of short paths through any variable and its negation by

$$O\left(\frac{\log^6 n}{n}\right) \cdot \sum_{t_1 \leq \log^2 n} \left(\frac{m}{n}\right)^{2t_1} \cdot (\alpha^*)^{(t_1/2)-1}.$$

Now since $\left(\frac{m}{n}\right)^4 \alpha^* \leq \left(\frac{1000}{999}\right)\left(\frac{969}{970}\right) < 1$, this final summation is a decreasing geometric series and hence bounded by a constant. Hence the probability of any such cycle is $O\left(\frac{\log^6 n}{n}\right) = o(1)$. This completes the proof of the theorem. □

6 Proof of Theorem 3

We employ the following greedy rule: Fix some assignment ψ to the variables, say the all-TRUE assignment. In each round k, if just one of the two clauses is satisfied by ψ, choose that clause; otherwise (if neither or both are satisfied) choose one of the clauses arbitrarily.

Let us estimate the probability that in round k both clauses are not satisfied by ψ. W.l.o.g., consider the first clause. Regardless of which variables appear in it, only one of the 2^k possible parities of the variables results in a clause that is not satisfied by ψ. Thus the probability of an unsatisfied clause is 2^{-k}. The probability of both clauses being unsatisfied by ψ is thus 2^{-2k}.

Now suppose $m = 0.99n \cdot 2^k$. The expected number of unsatisfied clauses is

$$2^{-2k} \cdot 0.99n \cdot 2^k = 0.99n \cdot 2^{-k}.$$

When $k = \omega(\log n)$ this quantity is $o(1)$, and the result follows from Markov's inequality together with the fact that the k-SAT threshold is at most $m/n = 2^k \ln 2$. □

7 Discussion

In this paper we considered the Achlioptas process for random 2CNF instances, and provided a simple greedy algorithm that provably delays the sat/unsat threshold for random 2SAT in this model.

As we mentioned in the introduction, there are several natural variations of the greedy rule we used to prove this result. For example, one can define a clause to be bad only if both literals appear already in negated form. Intuitively, this rule "punishes" the path structure in the implication graph even more severely than our rule does, and indeed experiments predict the threshold to be $m/n \approx 1.5$ using this rule. However, this rule is a little harder to analyze than ours, and we expect it to yield similar numerical bounds using our methods. We conjecture that the techniques we developed to prove Theorem 2 can be used in other settings as well. As an example, consider the pure literal heuristic for 3CNF instances. It is known that for random 3SAT, whenever $m/n \leq 1.63$ the pure literal procedure ends *whp* with all clauses satisfied [7], and this value is tight. The main idea of the proof is that if the peeling process of layers of pure literals

gets stuck before the formula is satisfied, then every variable in the remaining formula appears both positively and negatively. This combinatorial structure is similar to the paths that we are "punishing" in the 2SAT case. Therefore one might expect our on-line algorithm (applied to 3CNF formulas) to move the threshold above 1.63, given a choice of two clauses at each step.

Acknowledgments. We thank Benny Sudakov for introducing us to the Achlioptas process, Alan Frieze for a pointer to reference [15], and Uri Feige for useful discussions.

References

1. B. Apswall, M. Plass and R. Tarjan. A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Inf. Proc. Letters*, 8:121–123, 1979.
2. Y. Azar, A. Broder, A. Karlin and E. Upfal. Balanced allocations. *SIAM Journal on Computing*, 29(1):180–200, 1999.
3. T. Bohman and A. Frieze. Avoiding a giant component. *Random Structures & Algorithms*, 19(1):75–85, 2001.
4. T.Bohman, A.Frieze and N.Wormald. Avoidance of a giant component in half the edge set of a random graph. *Random Structures & Algorithms*, 25(4):432–449, 2004.
5. T. Bohman and J.H. Kim. A phase transition for avoiding a giant component. *Random Structures & Algorithms*, 28(2):195–214, 2006.
6. T. Bohman and D. Kravitz. Creating a giant component. *Combinatorics, Probability and Computing*, 15(4):489–511, 2006.
7. A. Broder, A. Frieze and E. Upfal. On the satisfiability and maximum satisfiability of random 3CNF formulas. *Proc. 4th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 322–330, 1993.
8. V. Chvátal and B. Reed. Mick gets some (the odds are on his side). *Proc. 33rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 620–627, 1992.
9. P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* 5:17–61, 1960.
10. A. Flaxman, D. Gamarnik and G. Sorkin. Embracing the giant component. *Random Structures & Algorithms*, 27(3):277–289, 2005.
11. E. Friedgut. Sharp thresholds of graph properties and the k-SAT problem. *Journal of the American Mathematical Society*, 12(4):1017–1054, 1998.
12. A. Frieze and N. Wormald. Random k-Sat: A tight threshold for moderately growing k. *Combinatorica*, 25(3):297–305, 2005.
13. A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53(3):469–486, 1996.
14. S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random Boolean expressions. *Science*, 264:1297–1301, 1994.
15. D. Kravitz. Random 2SAT does not depend on a giant. *SIAM Journal on Discrete Mathematics*, 21(2):408–422, 2007.
16. M. Krivelevich, P. Loh and B. Sudakov. Avoiding small subgraphs in Achlioptas processes. *Random Structures & Algorithms*, 34(1):165–195, 2009.
17. M. Krivelevich, E. Lubetzky and B. Sudakov. Hamiltonicity thresholds in Achlioptas processes. *Random Structures & Algorithms*, to appear.
18. J. Spencer and N. Wormald. Birth control for giants. *Combinatorica*, 27(5):587–628, 2007.