Random 2-SAT And Unsatisfiability

Yann Verhoeven
LRI, Bat 490, Université Paris Sud,
91405, Orsay
France
e-mail :yannv@ lri.fr

Abstract

It is known that almost all 2-CNF formulas with $n(1+\varepsilon)$ clauses are not satisfiable, $\varepsilon > 0$ a constant and n the number of variables. We strengthen this result, showing that it still holds with $\varepsilon = \varepsilon(n) \geq w(n)n^{-1/4}$, where $w(n)$ is any function that tends to infinity as n tends to infinity.

1 Introduction

Let $\{x_1, \ldots, x_n\}$ be a set of n propositional variables. The corresponding set of literals is $\{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\}$. We consider all the 2-literals clauses made with the above mentioned literals. Let F be the random formula in which each of the clauses appears with probability p. Let $|F| \sim p\binom{2n}{2}$ be the number of clauses contained in F.

Goerdt [6] and Chvàtal and Reed [5] proved independently the following: if F is a formula defined as previously, then for any fixed $\varepsilon > 0$, as n tends to infinity,

1. if $|F| = (1-\varepsilon)n$, the probability that F is satisfiable tends to 1;
2. if $|F| = (1+\varepsilon)n$, the probability that F is satisfiable tends to 0.

More recently, Goerdt [7] refined the size of F in the case 1, showing that the result still holds if $\varepsilon \geq (\ln n)^{-1/2}$.

The purpose of this paper is to refine the case 2 by giving an approximation of the function $\varepsilon(n)$. We take $p = \frac{1+\varepsilon}{2n}$, so $E|F| = (1+\varepsilon)n$. We prove the following theorem,

Theorem 1 *Let $w(n)$ be any function, tending to infinity arbitrarily slowly as n tends to infinity. Then almost every system of $n(1 + w(n)n^{-\frac{1}{4}})$ random 2-clauses on n logical variables is not satisfiable.*

Theorem 2 *Let $w(n)$ be any function, tending to infinity arbitrarily slowly as n tends to infinity. Then almost every system in which each clause occurs independently with probability $p = \frac{1+\varepsilon}{2n}$, so $E|F| = (1+\varepsilon)n$, is not satisfiable.*

We shall prove theorem 1. Theorem 2 follows from theorem 1 by standard arguments (see for instance [4]).

In order to prove this result, we first present an algorithm and a procedure derived from an algorithm presented in [1]. Then, we analyze the behaviour of the algorithm by using results on random processes and more precisely on Galton-Watson processes.

2 Basics

Let F be a random formula as defined above. In order to apply our algorithm, we first translate F in terms of a directed graph $G(F)$. For this purpose, we use the method proposed by Asphall, Plass and Tarjan [2]. The graph has $2n$ nodes, labeled by the literals. Then each clause $u \vee v$ gives 2 edges, namely $\overline{u} \rightarrow v$ and $\overline{v} \rightarrow u$. In [2], it was shown that F is insatisfiable iff the graph $G(F)$ contains at least one cycle in which a literal and its complement appear. Such a cycle is called a contradictory cycle.

In the following, $w(n)$ is an arbitrary function increasing to infinity as n tends to infinity, δ is a constant given later and $\varepsilon = \varepsilon(n)$ is also defined later. Finally, we set $N = 2n(1-\delta)$ and define θ by $1+\theta = (1+\varepsilon)(1-\delta)$.

3 The algorithm

Our algorithm finds a contradictory cycle with high probability if $|F| = (1+\varepsilon)n$, $\varepsilon \geq w(n)n^{-1/4}$. First, choose a vertex u at random in $G = G(F)$; next, construct a set A_1 of vertices of $G(F)$ directly reachable from u and a set B_1 of vertices from which $\overline{u}$ can be directly reached. In a graph like $G(F)$, if two sets of vertices are big enough, then they are likely to be connected. Therefore the algorithm stops if A_1 and B_1 are of sufficient size (formula F is then unsatisfiable with high probability). If A_1 and B_1 are not big enough, construct A_2 and then B_2. A_2 is A_1 plus the vertices reachable from a randomly chosen vertex of A_1 (that are reachable in the graph minus A_1) minus that particular vertex. We construct B_2 similarly. We have to note that once we have created the set A_i, we then create the set B_i so we stop as soon as we reach a dead end on either end. We repeat the procedure until we find two sets A_j and B_j that have attained the right size, or that one of them is empty (in this latter case we start again from new vertices u and $\overline{u}$). We denote by $X(u)$ the first set A_j big enough, and by $Y(\overline{u})$ the first set B_j big enough. This method enables us to modelize the algorithm via a *Galton-Watson branching process* because the distribution of the generations don't change, due to our special management of the vertices to be described.
Assume that we construct $X(u)$ first and let us describe this construction. The construction of $X(\overline{u})$ is similar except for the fact that it begins with a smaller number of available vertices. The construction of $X(u)$ is similar to the one presented in [1].

Algorithm MAIN

1. Let H be a random subset of $V(F)$, containing $2n-N$ vertices.
2. $T := V(F) - H$, $INF := false$
3. **while** $INF = false$
   (a) Let u be a random vertex of T such that $\overline{u} \in T$; set $A_0 := \{u\}$;
   (b) Run TREE for $u \to X(u),\ X(\overline{u})$;
4. Test for a path between $X(u)$ and $Y(\overline{u})$;
5. Test for a path between $X(\overline{u})$ and $Y(u)$;

Algorithm TREE

1. $END := false$; $i := 0$; $A_0 := \{u\}$; $B_0 := \{\overline{u}\}$;
2. **while** $END = false$,
   - Let $m_1 := |A_i|$, $A_i := \{x_1, x_2, \ldots, x_{m_1}\}$, $A_{i+1} := \emptyset$ and $m_2 := |B_i|$, $B_i := \{y_1, y_2, \ldots, y_{m_2}\}$, $B_{i+1} := \emptyset$
   - **if** $(|A_i| \geq w(n)\sqrt{2n}$ **and** $|B_i| \geq w(n)\sqrt{2n})$ **then** $END := true$ **and** $INF := true$
   - **if** $(m_1 = 0$ **or** $m_2 = 0)$ **then** $END := true$
   - **else**
     - $A_{i+1} := A_{i+1} \cup (\Gamma^+(x_1)) \cap T$
     - $B_{i+1} := B_{i+1} \cup (\Gamma^+(y_1)) \cap T$
     - replace the sets $\Gamma^+(x_1)$ and $\Gamma^+(y_1)$ in T by $|\Gamma^+(x_j)| + |\Gamma^+(y_1)|$ vertices of H.

This method of construction yields both $X(u)$ and $Y(\overline{u})$.

It will be shown that the success probability of 4 and 5 (given that 3b is successful) is $1 - o(1)$.

In the following, the sets (A_i) and $B_i)$ will be denoted as boundaries of processes.

4 Analysis of a branching process

We consider a Galton-Watson branching process with reproduction distribution $\mathrm{Bin}(N, (1+\varepsilon)/2n)$, N, n, ε are fixed and are defined as in the rest of the paper. Z_n and S respectively denote the size of the n^{th} generation and the total size of the process ($S = \sum_{n \geq 0} Z_n$), which can be infinite. An introduction and further results about such processes can be found in [3, 8].

Aside from the customary process like representation $(Z_i)_{i \geq 0}$ of a branching process, where Z_i denotes the size of the i^{th} generation, we also think in terms of sequences of family trees $(T_i)_{i \geq 0}$ where T_0 is the tree consisting of precisely one vertex and, for each $i \geq 0$, T_{i+1} is obtained from T_i by "expanding" one not yet expanded vertex. (Thus, if all vertices have been expanded, T_{i+1} is empty.) We refer to the set of as yet not expanded vertices as the *boundary* of T_i and denote it by B_i. Clearly, there is much latitude concerning the order in which the vertices are expanded. As a rule, we expand the vertices in the order in which they are produced, thereby keeping track of the relation between these two presentations: for each i, Z_{i+1} is the number of vertices of depth $i+1$ in our family tree when the vertices of depth i have just all been expanded.

Lemma 1 *Consider a Galton-Watson branching process GW defined as above. Then for any function $w(n)$ tending to infinity with n,*

$$\Pr[GW \textit{ does not get extinct}] \geq \frac{2\theta}{(1+\theta)^2} + o(\theta) \ .$$

Conditionally on the extinction of the process, we have

$$E[|S|] \leq \frac{1+O(\theta)}{\theta} \ .$$

Proof: The probability that such a process doesn't reach size $w(n)\sqrt{2n}$ can be upper bounded by the probability of extinction q of this process. We know that if f is the generating function of the probability law $\mathrm{Bin}(N, (1+\varepsilon)/2n)$, q is the root in $]0,1[$ of the equation $f(s) = s$. f is given by

$$f(s) \quad = \quad \left(1 - \frac{1+\varepsilon}{2n} + \frac{1+\varepsilon}{2n}s\right)^{2n(1-\delta)} \ .$$

So we must solve the following equation,

$$s = \left(1 - \frac{1+\varepsilon}{2n}(1-s)\right)^{2n(1-\delta)} \ .$$

One can show that the solution of this equation satisfies

$$q = 1 - \frac{2\theta}{(1+\theta)^2} + o(\theta) \ .$$

We now study the mean size of the generations of a process, when extinction occurs. Theorem 3 page 53 in [3], which tells us that given its extinction, our process has the same evolution as a process with generating function $f^*(s) = q^{-1}f(sq)$. The mean branching number in this extincting process is $f^{*\prime}(0) = m^* = (1+\theta)q$, which is lower than 1. We then deduce the mean size of our process, given its extinction:

$$\begin{aligned}
\mathrm{E}\left[S | \textit{ extinction } \right] &= \sum_n (m^*)^n \\
&= \sum_n ((1+\theta)q)^n \\
&= \frac{1}{1-(1+\theta)q} \\
&= \frac{1+o(\theta)}{\theta} \ .
\end{aligned}$$

□

We next need to evaluate the size of the boundary of a Galton-Watson tree provided its non-extinction. Let's consider a random tree T and its sequence of boundaries $(B_i)_{i\geq 0}$ defined as above.

Lemma 2 *Consider a sequence $(|B_k(T)|)_{k\geq 0}$ of random variables defined as above. Then, for every T conditionned on survival,*

$$|B_k(T)| \geq \mathit{Bin}(kN, (1+\varepsilon)/2n) \quad \textit{stochastically} \qquad (1)$$

Proof In order to prove the theorem, we study a particular sequence of random variables and prove the inequality 1 via a coupling method.
We consider a sequence $(Y_k)_k$ of random variables such that $Y_k = Y_{k-1} + Z_k - 1$,, where for every $k \geq 0$, Z_k has distribution law $\text{Bin}(N, (1+\varepsilon)/2n)$.
We now consider a non-extincting Galton-Watson tree. It is known that a non-extincting tree behaves like a size-biased tree (see [9]), in which in each generation a distinguished vertex is forced to have number of offspring is at least 1. We define $(C_k)_{k\geq 0}$ as the sequence of random variables defined as the size of the boundaries of our non-extincting tree. The C_k are distributed as follow,

- if Z_k is strictly positive, then $C_k = C_{k-1} + Z_k - 1$,
- else if the considered vertex is a distinguished one then $C_k = C_{k-1}$, and if it is not a distinguished vertex, $C_k = C_{k-1} - 1$.

We then obtain, for every $k \geq 0$, $C_k \geq Y_k$ *a.s.* But it is easy to see that Y_k has distribution law $\text{Bin}(kN, (1+\varepsilon)/2n) - k$, and the theorem follows. □

5 Analysis of the algorithm

We claim that the former analysis of the Galton-Watson branching process provides an analysis of the first steps of the search process used in section 3 for the construction of set A_j. At stage i, the algorithm chooses a particular vertex v from A_{i-1}, determines the offspring $O(v)$ of v and constructs the set $A_i = A_{i-1}\backslash\{v\} \cup O(v)$. Since $|O(v)|$ follows a distribution law of type $\text{Bin}(N, p)$, the algorithm mirrors a Galton-Watson tree. Note that the updating of $G(F)$ makes p constant. In the algorithm, we compensated the loss of vertices by adding new ones we had previously stocked. Recall that originally we had $2n$ vertices in the graph, but in the algorithm we considered only N vertices for each iteration of the loop. From this, it follows that we have to consider at each step the binomial law $\text{Bin}(N, (1+\varepsilon)/2n)$. Thus, the behaviour of the search process is similar to the behaviour of the Galton-Watson process studied in section 4.

We must show that the algorithm enables us to find two trees in $G(F)$, each of them containing a boundary of size at least $w(n)\sqrt{2n}$. First we study a tree given non-extinction and show that there exists n_0 such that the n_0^{th} boundary of the tree reaches size $w(n)\sqrt{2n}$ with high probability. It is done in lemma 3. We then show that the algorithm succeeds with high probability with no more than $1/4\theta^2$ executions of the algorithm MAIN (section 3). This is the object of lemma 4. Then we demonstrate via lemma 5 that the algorithm doesn't need more than $1/2\theta^3$ vertices during the trials, with high probability. We can then find the correct values for ε and δ.

Lemma 3 *Given a tree constructed by the algorithm MAIN, given its non-extinction, there exists k_0 such that the k_0^{th} boundary reaches size $w(n)\sqrt{2n}$ with high probability.*

Proof The lemma 2 enables us to lower bound the size of the k^{th} boundary of the tree. We can deduce from this relation that if k_0 is greater than

$2w(n)n^{3/4}$ then the k_0^{th} boundary has size greater than $w(n)\sqrt{2n}$ with high probability. $\square$

Lemma 4 *The algorithm presented in section 3 finds 2 subsets of $G(F)$ of boundary at least $w(n)\sqrt{2n}$ after at most $1/4\theta^2$ runs almost surely.*

Proof According to lemma 1, we have a tree with boundary size greater than $w(n)\sqrt{2n}$ with probability greater than the probability of non-extinction of a Galton-Watson tree, *i.e.* $2\theta + o(\theta)$. The algorithm presented in section 3 develops 2 such trees independently, so we succeed with probability greater than $4\theta^2 + o(\theta^2)$. Denoting by $\#trials$ the number of executions of the algorithm until success, we have

$$\begin{aligned} \Pr\left[\#trials \geq \frac{w(n)}{4\theta^2}\right] &\leq \left(1 - 4\theta^2 + o(\theta^2)\right)^{\frac{w(n)}{4\theta^2}} \\ &\leq \exp\left(-3w(n) + o(w(n))\right) \ . \end{aligned}$$

$\square$

Lemma 5 *The algorithm given in section 3 uses no more than $w(n)/2\theta^3$ runs, with high probability, before finding 2 subsets of $G(F)$ with boundary size at least $w(n)\sqrt{2n}$.*

Proof At each trial, if the tree is not big enough, we no longer consider the vertices explored by the procedure. So, each new trial of the procedure is done on a graph that has lost $2/\theta$ vertices. We claim that we have lost no more than $1/2\theta^3$ vertices with high probability at the end of the algorithm. The control of this value is also given by Markov inequality. The algorithm will run at most $w(n)/4\theta^2 = m$ times with high probability before stopping and construct trees of size respectively, $T_1, T_1', \dots, T_m, T_m'$, all of size less than $w(n)\sqrt{2n}$ vertices. The sum of the sizes of these trees is the sum of $w(n)/4\theta^2$ independent random variables of mean $\mathrm{E}T = 2/\theta$. We have

$$\mathrm{E}\left[T_1 + T_1' + \ldots + T_m + T_m'\right] = \frac{w(n)}{4\theta^2}\mathrm{E}T = \frac{w(n)}{2\theta^3} \ .$$

According to Markov inequality,

$$\Pr\left[T_1 + T_1' + \ldots + T_m + T_m' \geq t\right] \quad \leq \quad \frac{w(n)}{2\theta^3 t} \ .$$

We take $t = w(n)/2\theta^3$, and the lemma 5 is then proved. $\square$

Finally, we have to define the values of the constants ε and δ. For the validity of the analysis of the search procedure, we want the mean of the law $\mathrm{Bin}(N, 1+\varepsilon/2n)$ to be lower than the mean number of vertices actually reachable, *i.e.*,

$$N\frac{1+\varepsilon}{2n} \leq \left(\frac{1+\varepsilon}{2n}\right)\left(2n - \frac{w(n)}{2\theta^3}\right)$$

which is equivalent to,

$$4n\theta^4 - 4n\varepsilon\theta^3 + (1+\varepsilon)w(n) \leq 0 \ . \tag{2}$$

The function $\theta \mapsto 4n\theta^4 - 4n\varepsilon\theta^3 + (1+\varepsilon)w(n)$ gets minimal for $\theta = 3\varepsilon/4$. The inequality (2) then becomes,

$$-c\varepsilon^4 n + (1+\varepsilon)w(n) \leq 0$$

with c a constant. And finally, this inequality is verified with:

$$\varepsilon \geq w(n)n^{-\frac{1}{4}} \ .$$

Then, if $\varepsilon \geq w(n)n^{-1/4}$, then the algorithm can develop with high probability 2 trees of size greater than $w(n)\sqrt{2n}$.

6 The probability of a connection

We have to show that, as announced, that two sets $X(u_0)$ and $X(\overline{u_0})$, each containing $w(n)\sqrt{2n}$ vertices, are almost surely connected to each other.

Let u_0 be a vertex of $G(F)$. Assume $|X(u_0)| \geq w(n)\sqrt{2n}$ and $|X(\overline{u_0})| \geq w(n)\sqrt{2n}$. Let u be a vertex of $X(u_0)$. Then, by construction, $\overline{u}$ belongs to set $Y(\overline{u_0})$. So, if there exists an edge between $u \in X(u_0)$ and $v \in Y(\overline{u_0})$, then there also exists an edge between $\overline{u}$ and $\overline{v}$. Since this pair of edges exists with probability p, the probability that there is no such pair is

$$(1-p)^{2nw^2(n)} \quad = \quad \exp(-w^2(n) + O(1/n)) \ .$$

Thus, there exists an edge between $X(u_0)$ and $Y(\overline{u_0})$ almost surely. By the same method, we know there exists an edge between $X(\overline{u_0})$ and $Y(u_0)$ almost surely.

ACKNOWLEDGMENTS

The author thanks S. Boucheron and C. Kenyon for stimulating discussions concerning this paper.

References

[1] M. Ajtai, J. Komlòs, and E. Szemerédi. The longest path in a random graph. *Combinatorica*, 1(1):1–12, 1981.

[2] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 1979.

[3] K. B. Athreya and P. E. Ney. *Branching Processes.* Springer Verlag, New York, 1972.

[4] B. Bollobàs. *Random Graphs.* Academic Press, 1985.

[5] V. Chvátal and B. Reed. Micks gets some (the odds are on his side). In *Proceedings 33rd FOCS*, pages 620–627, 1992.

[6] A. Goerdt. A threshold for unsatisfiability. In J.M. Havel and V. Koubek, editors, *Proceedings MFCS*, pages 264–274. LNCS 692, 1992.

[7] A. Goerdt. A remark on random 2-SAT. submitted, 1997.

[8] T. E. Harris. *The Theory of Branching Processes.* Springer Verlag, New York, 1963.

[9] R. Lyons and Y. Peres. Probability on trees and networks. In press, http://php.indiana.edu/ rdlyons/prbtree/prbtree.html, 1999.