Satisfiability and the Giant Component in Online Variants of the Classical Random Models

David Paul Kravitz

Department of Mathematical Sciences

Carnegie Mellon University

kravitz@cmu.edu

Advised by Thomas Bohman

Department of Mathematical Sciences

Carnegie Mellon University

tbohman@moser.math.cmu.edu

Committee

Alan Frieze

Dept. of Math Sciences

Carnegie Mellon University

af1p@andrew.cmu.edu

Mike Molloy

Dept. of Computer Science

University of Toronto

molloy@cs.toronto.edu

Oleg Pikhurko

Dept. of Math Sciences

Carnegie Mellon University

pikhurko@andrew.cmu.edu

ACKNOWLEDGEMENTS

This work was supported in part by a VIGRE grant from the NSF, DMS-9819950. I would like to express my gratitude both to Carnegie Mellon University and to the NSF for providing me with such generous support.

I am indebted to the extremely dedicated faculty at Carnegie Mellon University and the University of Delaware who made reaching my academic goals possible. My graduate advisor, Tom Bohman, showed an incredible amount of patience with me while he constantly guided me in the right direction. Alan Frieze and Felix Lazebnik were also huge contributors as well, as were many others.

I would also like to thank the staff and the faculty in the Department of Mathematical Sciences at Carnegie Mellon University, and my parents, who went out of their way to make sure that difficult situations for me went as smoothly as possible, and allowed me to concentrate on the work required to finish this thesis.

TABLE OF CONTENTS

ABSTRACT

We introduce and study online versions of two classical random structures. The first is a variation on the classical random graph model, the second is the satisfiability model.

We begin with the random graph. Let c be a constant and $(e_1, f_1), (e_2, f_2), \ldots, (e_{cn}, f_{cn})$ be a sequence of ordered pairs of edges on vertex set $[n]$ chosen uniformly and independently at random. Let A be an algorithm for the online choice of one edge from each presented pair, and for $i = 1, \ldots, cn$ let $G_A(i)$ be the graph on vertex set $[n]$ consisting of the first i edges chosen by A. We prove that all algorithms in a certain class have a critical value c_A for the emergence of a giant component in $G_A(cn)$ (i.e. if $c < c_A$ then with high probability the largest component in $G_A(cn)$ has $o(n)$ vertices and if $c > c_A$ then with high probability there is a component of size $\Omega(n)$ in $G_A(cn)$). We show that a particular algorithm in this class with high probability produces a giant component before $0.385n$ steps in the process (i.e. we exhibit an algorithm that creates a giant component relatively quickly). The fact that another specific algorithm that is in this class has a critical value resolves a conjecture of Spencer. In addition, we establish a lower bound on the time of emergence of a giant component in any process produced by an online algorithm and show that there is a phase transition for the offline version of the problem of creating a giant component.

Now we consider satisfiability. Given n Boolean variables $x_1, \ldots, x_n$, a k-clause is a disjunction of k literals, where a literal is a variable or its negation. Suppose random k-clauses are generated one at a time and an online algorithm

accepts or rejects each clause as it is generated. Our goal is to accept as many randomly generated k-clauses as possible with the condition that it must be possible to satisfy every clause which is accepted. When cn random k-clauses on n variables are given, a natural online algorithm known as *Online-Lazy* accepts an expected $(1-\frac{1}{2^k})cn + z_k n$ clauses for some constant z_k. If these clauses are given offline, it is possible to do much better, an expected $(1-\frac{1}{2^k})cn + \Omega(\sqrt{c})n$ can be accepted. The question of closing the gap between $(1-\frac{1}{2^k})cn + z_k n$ and $(1-\frac{1}{2^k})cn + \Omega(\sqrt{c})n$ for the online version was posed by Coppersmith, Gamarnik, Hajiaghayi, and Sorkin. We show that for any $k \geq 1$, any online algorithm will accept less than $(1-\frac{1}{2^k})cn + (\ln 2)n$ k-clauses **whp**, furthermore we show that this bound is asymptotically tight as $k \to \infty$.

We also introduce a new random model for random $2-SAT$. It is well-known that in the standard model there is a sharp phase transition, the probability of satisfiability quickly drops as the number of clauses exceeds the number of variables. The location of this phase transition suggests that there is a direct connection between the appearance of a giant in the corresponding $2n$-vertex graph and satisfiability. Here we show that the giant has nothing to do with satisfiability, and in fact the expected degree of a randomly chosen vertex is the important parameter.

Chapter 1

THE ACHLIOPTAS PROCESSES

We begin with vertex set $[n] = \{1, 2, \ldots, n\}$. We are primarily interested in what happens as $n \to \infty$, and we will say something happens *with high probability*, or **whp**, if it happens with probability $1 - o(1)$. A random edge is a pair of vertices chosen uniformly at random from $\binom{[n]}{2}$.

We first examine $G_{n,cn}$, which is the graph with n vertices and cn random edges for some constant c. It is well known, a classical result of Erdős and Rényi [19], that for any $c < \frac{1}{2}$ the largest component is of size $O(\log n)$ **whp**, and for $c > \frac{1}{2}$ there is a giant component (i.e. a component of size $\Omega(n)$) **whp**. The c values in which this huge jump in the largest component size occurs is called the phase transition, or double-jump threshold.

Although we concentrate on $c - \frac{1}{2}$ being a constant, there has also been a significant amount of work done when c differs from $\frac{1}{2}$ by less than a constant. It has been shown in [4, 31, 34] and several other places that a phase transition still exists in this much smaller window of time. If $M = \frac{n}{2} - s$ for $n \gg s(n) \gg n^{2/3}$, then **whp** the largest component of $G_{n,M}$ has $\Theta(n^2 s^{-2} \log(s^3 n^{-2}))$ vertices. On the other hand, if $M = \frac{n}{2} + s$ for $s(n) \gg n^{2/3}$, then **whp** the largest component of $G_{n,M}$ has $\frac{2(s+\overline{s})n}{n+2s} \pm O(n^{2/3})$ vertices, where $\overline{s}$ satisfies

$$\left(1 - \tfrac{2\overline{s}}{n}\right) \exp\left(\tfrac{2\overline{s}}{n}\right) \;=\; \left(1 + \tfrac{2s}{n}\right) \exp\left(-\tfrac{2s}{n}\right).$$

Furthermore, all other components are either trees or unicyclic components, with less than $n^{2/3}$ vertices.

Now we add a variation on the theme. Let c be a constant and let

$$(e_1, f_1), (e_2, f_2), \ldots, (e_{cn}, f_{cn})$$

be a sequence of pairs of edges on vertex set $[n]$ chosen uniformly and independently at random. We will examine algorithms which create random graphs by choosing one edge from each of these cn pairs. We consider two versions:

- *The online version*, where pairs appear sequentially and the choice of edge from the pair (e_t, f_t) is made without knowledge of future edges.
- *The offline version*, where the choice of edge from the pair (e_t, f_t) is made with complete knowledge of the full set of edges.

The online version is called an **Achlioptas process**, after Dimitris Achlioptas. He asked if there exists a $c > \frac{1}{2}$ and an online algorithm which chooses one edge from each of cn presented pairs such that **whp** a graph without a giant component is formed (this is often called the problem of avoiding a giant component). Recently, a lot of work has been done on this problem. Here, the interesting case is $c > \frac{1}{2}$ because otherwise the Erdős and Rényi result shows that the trivial algorithm which always chooses edge e_i will avoid a giant **whp**.

Bohman and Frieze introduced an online algorithm and proved that **whp** it produces a graph with no giant component for any $c < 0.535$ [11]. Spencer and Wormald claim that $c = 0.89$ can be achieved by an online algorithm [36]. Bohman and Kim showed that the offline version of the avoiding a giant component problem has a threshold at c_{off} roughly equal to 0.976 [13]. (If $c > c_{\text{off}}$ then **whp** every graph that consists of at least on edge from each pair $(e_1, f_1), \ldots, (e_{cn}, f_{cn})$ has a giant component and if $c < c_{\text{off}}$ then the **whp** there exists a choice of one edge from each pair that produces a graph in which the largest component has size $o(n)$.) This threshold is strictly greater than an upper bound on the online version of the

avoiding a giant component problem that was established by Bohman, Frieze and Wormald [12]; in other words, there exist values of c for which any online algorithm **whp** produces a graph with a giant but there exists a way to make an offline choice of one edge from each pair that succeeds **whp** in producing a graph with no giant component.

In this thesis we primarily do two things with these processes. First, we analyze general processes and look at the largest component in the graph they create. Second, we try to create a giant component in cn rounds of an Achlioptas process for c as small as possible. The result of Erdős and Rényi shows that for any $c > \frac{1}{2}$ we can create a giant component **whp** by choosing e_i for all i, and for any $c < \frac{1}{4}$ **whp** we can not create a giant even if we are allowed to choose all $2cn$ edges. Thus, the question is interesting for $c \in (\frac{1}{4}, \frac{1}{2})$.

We begin with general online processes. Let A be an online algorithm for the choice of one edge from each presented pair (i.e. the choice of edge from the pair (e_i, f_i) is made without knowledge of pairs (e_j, f_j) such that $j > i$). Let $G_A(i)$ be the graph on vertex set $[n]$ consisting of the first i edges chosen by A. This produces a random graph process $G_A(1), G_A(2), \ldots, G_A(cn)$. Note that this class includes processes produced by algorithms that might be designed to influence the emergence of a giant component in a variety of ways.

1.1 Size Algorithms

Now we discuss two broad classes of processes that were introduced by Spencer [36]. A **size algorithm** A makes the choice between edges e_{t+1} and f_{t+1} based on the sizes of the components in $G_A(t)$ that are joined by e_{t+1} and f_{t+1}. We will denote $e_{t+1} = \{u_{t+1}, v_{t+1}\}$ and $f_{t+1} = \{x_{t+1}, y_{t+1}\}$, for $t = 0, \ldots, cn-1$. We assume (for convenience) that $u_{t+1}, v_{t+1}, x_{t+1}, y_{t+1}$ is an ordered sequence of vertices and let $a_{t+1}, b_{t+1}, c_{t+1}$, and d_{t+1} be the sizes of the components containing $u_{t+1}, v_{t+1}, x_{t+1}$, and y_{t+1}, respectively, in $G_A(t)$.

Formally, a size algorithm chooses the edge e_{t+1} if the 4-tuple

$$(a_{t+1}, b_{t+1}, c_{t+1}, d_{t+1}) \tag{1.1}$$

lies in some fixed set (this subset of $\{1, 2, \dots\}^4$ defines the algorithm) and chooses f_i otherwise.

Conjecture 1 (Spencer)**.** *Any size algorithm A has a critical value t_0 such that for any $\epsilon > 0$, at $c = t_0 - \epsilon$ the largest component of $G_A(cn)$ is $O(\log n)$, and at $c = t_0 + \epsilon$ the largest component of $G_A(cn)$ is $\Omega(n)$. Furthermore, at $c = t_0 + \epsilon$ the second largest component of $G_A(cn)$ is $O(\log n)$.*

This is Conjectures 3 and 4 in [36]. The **product rule**, which accepts edge e_{t+1} if $a_{t+1}b_{t+1} \leq c_{t+1}d_{t+1}$, is an example of a size algorithm. Very little is known about the graph evolution given by the product rule, even the analysis of the location and nature of the phase transition is open. It is natural to think that the product rule is the optimal size algorithm with respect to avoiding a giant component (based on the analysis introduced in Section 3.3), however this does not seem to be true [37].

A **bounded size algorithm** is a size algorithm that makes no distinction between components larger than some fixed constant m. Formally, a bounded size algorithm is defined by a fixed subset of the finite set

$$(1, 2, \dots, m, \overline{m})^4,$$

where we abuse notation by letting $\overline{m}$ denote 'all integers larger than m', and a bounded size algorithm will choose edge e_{t+1} or f_{t+1} depending on whether the 4-tuple (1.1) lies in this subset or not.

Spencer also made a sequence of conjectures regarding these processes [36]. The issues here are the existence, location and nature of a phase transition in the size of the largest component of $G_A(t)$. For each bounded size algorithm there is a natural candidate for the location of this critical value: the blow-up point in the

differential equation for the sum of the squares of the component sizes (this quantity is known as the susceptibility in statistical physics and is discussed in Section 3.3). If A is a bounded size algorithm let this value be c_A (the formal definition of c_A is given in Chapter 4).

Conjecture 2 (Spencer)**.** *Let A be the algorithm that takes e_{t+1} if it joins two isolated vertices in $G_A(t)$ and otherwise takes f_{t+1}. If $c < c_A$ then* **whp** *all components of $G_A(cn)$ have size $O(\log n)$.*

Conjecture 3 (Spencer)**.** *Let A, c_A be as from Conjecture 2. If $c > c_A$ then* **whp** *$G_A(cn)$ has a component of size $\Omega(n)$.*

Conjecture 4 (Spencer)**.** *Let A be any bounded size algorithm. If $c < c_A$ then* **whp** *all components of $G_A(cn)$ have size $O(\log n)$, and if $c > c_A$ then* **whp** *$G_A(cn)$ has a component of size $\Omega(n)$.*

Conjectures 2, 3, and 4 are Conjectures 1,2 and 6, respectively, of [36]. The last conjecture from [36] guesses that any size algorithm can be approximated by bounded size algorithms:

Conjecture 5 (Spencer)**.** *Given a size algorithm A, a restriction to K is a bounded size algorithm that agrees with A when all 4 component sizes are less than K. For any size algorithm A with critical value t_0 and any positive δ, there exists K_0 such that all restrictions with $K \geq K_0$ have critical value within δ of t_0.*

1.2 Our Main Result

This is a general theorem that establishes the existence of a critical value for the emergence of a giant component for a class of Achlioptas processes. We call an algorithm A a **bounded first-edge algorithm** if it chooses between e_{t+1} and f_{t+1} by observing the sizes of the components in $G_A(t)$ connected by e_{t+1}, making no distinction between components larger than some fixed constant m. In other words,

such an algorithm looks at the first edge in the pair of random edges and either accepts it based on the sizes of the components involved or rejects it in favor of the (not yet observed and therefore purely random) second edge. Note that the class of bounded first-edge algorithms is contained in the class of bounded size algorithms. We formally define a bounded first-edge algorithm A with a fixed set

$$\mathcal{S}_A \subseteq \{1, 2, \dots\}^2,$$

such that

$$(i,j) \in \mathcal{S}_A,\ m < i \ \text{ and } \ m < i' \ \Rightarrow \ (i',j) \in \mathcal{S}_A\ , \ \text{ and}$$
$$(i,j) \in \mathcal{S}_A,\ m < j \ \text{ and } \ m < j' \ \Rightarrow \ (i,j') \in \mathcal{S}_A$$

The algorithm A accepts e_{t+1} if and only if $(a_{t+1}, b_{t+1}) \in \mathcal{S}_A$ (recall that a_{t+1} and b_{t+1} are the sizes of the components in $G_A(t)$ that contain the vertices in e_{t+1}).

Theorem 6. *Let A be a bounded first-edge algorithm. There exists a constant c_A such that*

(a) If $c < c_A$ then **whp** *the largest component in the graph $G_A(cn)$ has $O\left((\log n)n^{12/13}\right)$ vertices, and*

(b) If $c > c_A$ then **whp** *the graph $G_A(cn)$ has a component of size $\Omega(n)$.*

The algorithm in Conjectures 2 and 3 is a bounded first edge algorithm (defined by $\mathcal{S}_A = \{(1,1)\}$). Thus, Theorem 6 resolves Conjectures 2 and 3.

The proof of Theorem 6 is given in Section 4.1. The main tool in the proof of Theorem 6 is the differential equations method for random graph processes (see Theorem 18 in Chapter 3). The critical value c_A is given by the blow-up point in the differential equation for the sum of the squares of the component sizes, as predicted above in Conjectures 2, 3, 4, 5.

Spencer and Wormald independently proved Theorem 6(b), and they showed that for any bounded size algorithm and any $c < c_A$ **whp** the largest component in $G_A(cn)$ has $O(\log n)$ vertices [37]. Note that this resolves Conjecture 2, and this is stronger than part (a) of Theorem 6, both in terms of the class of algorithms considered and in the bound on the component sizes in part (a). They used an exponential tail method, which roughly shows that the susceptibility is determined by the smaller components only. Formally, for fixed positive integers K and c, we say that a graph G on n vertices has a $\mathbf{K, c}$ **component tail** if

$$\tfrac{1}{n}\,|\{v \;:\; |C(v)| \geq s\}| \quad \leq \quad K\mathbf{e}^{-cs},$$

where $C(v)$ is the component in G containing vertex v.

This same exponential tail method was also used by Beveridge, Bohman, Frieze, and Pikhurko [7]. Here they analyzed a two-player game where each player is alternately presented with a pair of edges and chooses one of them, one players objective is to create a giant and the others is to avoid one. They showed that the product rule is an asymptotically optimal strategy for both players.

The machinery introduced in the proof of Theorem 6 can be applied to other situations. Note that bounded first-edge algorithms are static, that is, they employ the same rule throughout the process. The proof goes through, for example, for certain algorithms that always make the choice between e_{t+1} and f_{t+1} without observing f_{t+1} but allow the rule to change some bounded number of times during the process. The main limitation of the proof of Theorem 6 is that it requires a supply of chosen edges around the critical point that are purely random.

Since the critical value c_A given in Theorem 6 is given by a blow-up point in a system of differential equations, it can be estimated by numerically solving the system. In Section 4.4 we estimate the critical points for two bounded first-edge algorithms. The algorithm A_1 accepts e_{t+1} if neither u_{t+1} nor v_{t+1} is isolated in

$G_A(t)$; that is,

$$\mathcal{S}_{A_1} = \{(i,j) : i,j \in \{2,3,\dots\}\}\,.$$

This algorithm is designed to and creates a giant component relatively quickly.

Theorem 7. *If $c > 0.385$ then* **whp** *$G_{A_1}(cn)$ has a component of size $\Omega(n)$.*

In Section 4.4 we also estimate the critical value of the algorithm of Conjectures 2 and 3, which was introduced for the purpose of avoiding a giant.

1.3 Lower Bounds

Besides analyzing the performance of online algorithms, we establish a lower bound on the online version of the creating a giant component problem and establish a phase transition for the offline version of the problem. We show that all online algorithms **whp** fail to create a giant for c slightly larger than $\frac{1}{4}$ (recall that the interesting interval for the online version of creating a giant component is $\frac{1}{4} < c < \frac{1}{2}$).

Theorem 8. *If $c < 0.2544$ then for any Achlioptas process,* **whp** *all of the components of the graph created in cn steps will be of size $O(\log n)$.*

Finally, we consider the offline version of the problem of creating a giant component. Here we are given pairs of random edges $(e_1, f_1), \dots, (e_{cn}, f_{cn})$, and we try to create a giant by choosing one edge from each pair. In this case we establish a phase transition.

Theorem 9. *Let c be a constant and let $(e_1, f_1), (e_2, f_2), \dots, (e_{cn}, f_{cn})$ be a sequence of ordered pairs of edges on vertex set $[n]$ chosen independently and uniformly at random. If $c > \frac{1}{4}$ then* **whp** *there exists a collection of edges E such that $|E \cap \{e_i, f_i\}| \leq 1$ for all i and the graph $([n], E)$ has a component of size $\Omega(n)$.*

Using exactly the same method of proof we were able to extend this to $c > \frac{1}{2k}$ when given k edges at a time, instead of just 2 edges at a time for any $c > \frac{1}{4}$.

Note that it follows from Theorems 8 and 9 that there exist values of c between 0.25 and 0.2545 for which any online algorithm **whp** produces a graph with no giant but there exists an offline choice of one edge from each pair that succeeds **whp** in producing a graph with a giant component (which is analogous to the problem of avoiding a giant component [13]). Results similar to Theorem 8 and Theorem 9 were obtained independently by Flaxman, Gamarnik and Sorkin [21].

Section 4.2 consists of the proof of Theorem 8, and Section 4.3 consists of the proof of Theorem 9. Finally, in Section 4.5 we mention a very simple Achlioptas process that succeeds in creating a giant component for $c < 0.46$.

Chapter 2

RANDOM K-SAT

Let $\{x_1, x_2, \ldots, x_n\}$ be a set of n Boolean variables. The corresponding set of literals is

$$\mathbf{X} := \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\},$$

A *k-clause* is a set of k literals from $\mathbf{X}$. We say a clause is *satisfied* by an assignment of the variables if and only if at least one of its literals is true. The question of RANDOM k-SAT takes a family of k-clauses chosen at random and asks if there is an assignment to the Boolean variables for which every clause is satisfied. We are interested in what happens as $n \to \infty$.

Notation 10. *For any n, m and k, let $F_k(n, m)$ denote a set of m random k-clauses, where each k-clause is chosen uniformly at random from the set of all $\binom{kn}{k}$ possible k-clauses.*

In Section 2.1, we consider random $2-SAT$. While it appears that the structure of the corresponding graph, in particular the appearance of a giant component in this graph, has a lot to do with satisfiability, we present results that indicate this is not the case. In Section 2.2 we discuss Random $k-SAT$ when $k > 2$. In Section 2.3 we introduce an online version of Random $k-SAT$, present previously known results about this version, and present Theorem 16, which is an asymptotically optimal upper bound.

2.1 $k = 2$

Random 2-SAT is well understood. The following was proven by Chvátal and Reed in [15] and Goerdt in [23] for any fixed constant $\epsilon > 0$:

1. $F_2(n, (1-\epsilon)n)$ is unsatisfiable **whp**.

2. $F_2(n, (1+\epsilon)n)$ is satisfiable **whp**.

There have also been several other results which strengthened this to the case where $\epsilon = o(1)$. ([10, 38], and others), but from now on we will assume $\epsilon > 0$ is a constant.

In [15], Chvátal and Reed define a *bicycle* as a formula with at least two distinct variables $x_1, \ldots, x_s$ and clauses $C_0, C_1, \ldots, C_s$ that have the following structure: there are literals $w_1, \ldots, w_s$ such that each w_r is either x_r or $\overline{x}_r$, each C_r with $0 < r < s$ is $\{\overline{w}_r, w_{r+1}\}$, and $C_0 = \{u, w_1\}, C_s = \{\overline{w}_s, v\}$ with literals u, v chosen from $\{x_1, \ldots, x_s, \overline{x}_1, \ldots, \overline{x}_s\}$. They prove that every unsatisfiable family of 2-clauses contains a bicycle.

Each family of clauses F is easily seen to correspond to a graph G_F on $2n$ vertices, where each vertex of G_F corresponds to a literal in F and each edge corresponds to a clause. It is well-known ([19], and many others) that G_F undergoes a major change right when the number of clauses exceeds n. When F has $(1-\epsilon)n$ clauses, the largest connected component of G_F has $O(\log n)$ vertices and all components are either trees or unicyclic, making a bicycle extremely unlikely. However, when there are $(1+\epsilon)n$ clauses, a *giant component* of size $\Omega(n)$ appears, this component contains a lot of cycles and has a substantial 2-core.

It is very reasonable to think that the appearance of this complex component has something to do with the first appearance of at least one bicycle, and therefore the change in satisfiability. In [32], Molloy introduces several constraint satisfaction problems where the probability of satisfiability dramatically changes with the appearance of a giant component in the natural n vertex graph (which is the graph

considered here with the vertices for x_i and $\overline{x}_i$ identified). Here, we introduce a natural random model in which there is no connection between the appearance of a giant in G_F and satisfiability.

Given any simple graph G on $2n$ vertices, we will make a family of clauses $S(G)$ by randomly assigning labels from $\mathbf{X}$ to the vertices, then each edge corresponds to one clause. We would like to know the probability that $S(G)$ is satisfiable over the space of all possible assignments to the vertices. This question is equivalent to the one with $F_2(n, m)$ if G is a random graph with m edges, however we allow G to be *anything* (provided $\Delta(G)$ isn't extremely large). This model does allow clauses $x_i \wedge \overline{x}_i$ which are usually excluded in 2-SAT, however **whp** we will have $O(1)$ such clauses, which makes no difference in our results.

Note that $S(G)$ is satisfiable if and only if there are exactly n vertices in G which cover $E(G) \cup M$, where M is a random perfect matching added to G. We must take exactly one vertex from each edge in M for an edge cover of size n, and these n vertices must cover every edge in G. Vertices in the edge cover are "true", while vertices out of the edge cover are "false". We will primarily use this model, in most cases we will expose one matching edge at a time by matching a given vertex with a randomly chosen unmatched vertex.

Theorem 11. *If G is a graph with $2n$ vertices, less than $(1-\epsilon)n$ edges for some $\epsilon > 0$, and $\Delta(G) = o(\frac{n^{1/10}}{\log n})$, then $S(G)$ is satisfiable* **whp**.

This can be thought of as an extension of the result from Chvátal and Reed stated above, in that case G would be a random graph with $2n$ vertices and up to $(1-\epsilon)n$ edges. The necessity of a condition on $\Delta(G)$ is discussed in Section 5.4.

Our result in the case when there are $(1+\epsilon)n$ edges requires an additional condition, namely that enough of the edges come from vertices of a degree less than $O(\log n)$.

Notation 12. *For all $i \geq 0$, define $d_i = d_i(G)$ as the number of vertices of degree i in graph G.*

Theorem 13. *If G is a graph with $2n$ vertices and $\Delta(G) = o(n^{1/8})$, and there is some $\epsilon > 0$ and function $\tau \leq c \log n$ for some constant $c < \frac{3\epsilon}{16}$ such that*

$$\sum_{0}^{\tau} i d_i = (1+\epsilon)2n, \tag{2.1}$$

then $S(G)$ is not satisfiable **whp**.

This is also an extension of the Chvátal and Reed result because a random graph with $(1+\epsilon)n$ edges will **whp** satisfy (5.1) with τ equal to some sufficiently large constant. Theorems 11 and 13 are proven in Section 5.2.

If there is a collection of high-degree vertices incident with more than ϵn edges, the structure of the graph is much more important. However, we do believe the following to be true:

Conjecture 14. *Let $\epsilon > 0$. There exists $\phi > 0$ such that if G is a graph with $2n$ vertices and more than $(1+\epsilon)n$ edges, and $\Delta(G) \leq n^{\phi}$, then $S(G)$ is not satisfiable* **whp**.

In section 5.5 we discuss some results that lead us to believe Conjecture 14 is true.

2.2 $k \geq 2$

While Random 2-SAT has a threshold at $c = 1$, no such value is known for $k - SAT$ for any $k > 2$. In fact, it is not even known if such a value exists. In [3], Achlioptas and Peres proved that for $k \geq 2$, the threshold for satisfiability in Random k-SAT is in the range $c = 2^k \ln 2 - O(k)$.

Conjecture 15. *(Satisfiability Threshold Conjecture) For each k there exists a threshold density c_k such that for any positive ϵ, for all $c < c_k - \epsilon$, $F_k(n, (c_k - \epsilon)n)$ is satisfiable* **whp**, *and $F_k(n, (c_k + \epsilon)n)$ is not satisfiable* **whp**.

The closest result to this conjecture is a theorem of Friedgut, proving that each n has its own threshold $c_k(n)$, but these may not converge to a limit as $n \to \infty$ [22].

When $k = 3$ the best knowledge that we currently have is that when $c < 3.42$ that $F_3(n, cn)$ is satisfiable **whp** [25], and when $c > 4.6$ that $F_3(n, cn)$ is not satisfiable **whp** [18].

2.3 The Online Version

The question of MAX 2-SAT looks at $F_2(n, m)$ and asks for the maximum expected number of clauses that can be satisfied. In [17], Coppersmith, Gamarnik, Hajiaghayi, and Sorkin show that for $k = 2$ and c large the expected number is

$$\tfrac{3}{4}cn + \left(\lambda_2\sqrt{c}\right) n,$$

where $\lambda_2 \in (0.344, 0.510)$. It was also shown there that for any $k \geq 1$ we can expect

$$\left(1 - \tfrac{1}{2^k}\right) cn + \left(\lambda(k)\sqrt{c}\right) n$$

clauses to be satisfied, although bounds on $\lambda(k)$ are unknown to within a factor of $\sqrt{k}$.

The online version of RANDOM MAX k-SAT was introduced by Coppersmith, Gamarnik, Hajiaghayi, and Sorkin [17]. Here, cn clauses are presented one at a time, and we must either accept or reject a clause when it is given with no knowledge of future clauses. The goal is to accept as many clauses as possible so that a valid assignment exists on the family of clauses that has been accepted. An assignment is valid only if every single clause which was accepted is satisfied. This can be done either with a fixed c or with $c \to \infty$, here we primarily look at c fixed but large.

It is very easy for an online algorithm to accept an expected $(1 - \frac{1}{2^k})cn$ out of cn clauses; this is done by deciding on the values of the Boolean variables before the clauses are given. Perhaps the most natural improvement of this is known as

Online-Lazy [17]. Here, we start out with all Boolean variables undetermined and set them as the algorithm proceeds. We reject a clause only if all of its literals have already been set false. Any clause which is accepted but has no literals true gets one of its literals set to true immediately after acceptance. It was also shown in [17] that *Online-Lazy* is optimal among algorithms which are forced to satisfy each clause upon acceptance. As $c \to \infty$, this accepts an expected $(1 - \frac{1}{2^k})cn + a_k n$ clauses for some constant a_k. Using the differential equations methods discussed in [40] we are able to determine a_k for small k:

k	1	2	3	4	5	10
$\mathbf{a}_k$	0.5	0.375	0.2842...	0.2209...	0.1765...	0.0809...

The derivation of these numbers is discussed in Section 6.2.

Since the offline version allows $(1 - \frac{1}{2^k})cn + \Theta(\sqrt{c})n$ out of cn clauses to be accepted **whp**, this leaves a gap between $(1 - \frac{1}{2^k})cn + a_k n$ and $(1 - \frac{1}{2^k})cn + \Theta(\sqrt{c})n$, which we close with the following:

Theorem 16. *Fix any integer $k \geq 1$, constant $c > 0$ and any online algorithm. Given a random formula with cn k-clauses,* **whp** *the algorithm accepts fewer than*

$$\left(1 - \frac{1}{2^k}\right) cn \;+\; \left(\frac{\ln 2}{-2^k \ln\left(1 - \frac{1}{2^k}\right)}\right) n \tag{2.2}$$

clauses.

This is proven in Section 6.1. The quantity in (2.2) is bounded above by $(1 - \frac{1}{2^k})cn + (\ln 2)n$ for all $k \geq 1$, and for $k = 2$ it gives an upper bound of $\frac{3}{4}cn + 0.6024n$.

Recall that Achlioptas and Peres proved that for $k \geq 2$, the threshold for satisfiability in Random k-SAT is in the range $c = 2^k \ln 2 - O(k)$. Therefore, there exists a constant λ such that a naive online algorithm, which accepts the first $(2^k \ln 2 - \lambda k)n$ clauses then sets the variables and accepts each clause after this only if it is satisfied, will accept an expected

$$\left(1 - \tfrac{1}{2^k}\right) cn + (\ln 2 - o_k(1))n$$

out of cn clauses. So, Theorem 16 shows that this naive algorithm quickly approaches optimal as k grows.

Chapter 3

THE DIFFERENTIAL EQUATIONS METHOD

This is a very powerful use of a quite simple idea. We present a general theorem given by Wormald in [40]. The idea of approximation has existed in connection with continuous processes essentially since the invention of differential equations by Newton for approximation of the motion of bodies in mechanics. The first application of this method to random graphs was by Karp and Sipser [26], where they applied it to a greedy matching algorithm. Some results for discrete processes also appeared before, for example Kurtz's theorem [29].

To execute this method, we compute the expected changes in random variables of a process per unit, then regarding the variables as continuous we write down the differential equations suggested by these expected changes. Then we use large deviation theorems to show that with high probability the value of the variables is close to the solution of the differential equations.

First we present the following Chernoff inequality in a form which will be very convenient for us to use in the context of random graphs. This is also how it appears as equation (2.6) in [24]: If $X \in \mathrm{Bi}(n,p)$, $\lambda = np$, $\varphi(x) = (1+x)\log(1+x) - x$, $x \geq -1$, and $\varphi(x) = \infty$ for $x < -1$, and $t \geq 0$, then

$$\Pr(X \leq E[X] - t) \quad \leq \quad \exp\left(-\lambda\varphi\left(-\frac{t}{\lambda}\right)\right) \quad \leq \quad \exp\left(-\frac{t^2}{2np}\right). \tag{3.1}$$

We will also make use of an Azuma-Hoeffding type inequality for supermartingales as discussed in [30, 40]: If $Y_0, Y_1, Y_2, \ldots, Y_t$ is a sequence of random variables

such that $E[Y_i|Y_1, Y_2, \ldots, Y_{i-1}] \leq Y_{i-1}$ and $|Y_i - Y_{i-1}| \leq \lambda$ for some constant λ and all $i \leq t$, then for all $\alpha > 0$,

$$\Pr(Y_t - Y_0 \geq \alpha) \ \leq \ \exp\left(-\frac{\alpha^2}{2t\lambda^2}\right) \tag{3.2}$$

The following are easily obtained from the Azuma-Hoeffding inequality: If $\{X_i\}_{i\geq 0}$ is a sequence of random variables such that all differences $X_{k+1} - X_k$ are independent and $|X_{k+1} - X_k| \leq z$ for all $k \geq 0$, then

$$\Pr(X_k - E[X_k] \geq \lambda) \quad \leq \quad \exp\left(-\frac{\lambda^2}{8kz^2}\right) \tag{3.3}$$

and

$$\Pr(E[X_k] - X_k \geq \lambda) \quad \leq \quad \exp\left(-\frac{\lambda^2}{8kz^2}\right) \tag{3.4}$$

for all $\lambda > 0$.

3.1 A General Theorem

This is from Theorem 5.1 of [40].

Definition 17. *Given a set S, we define the following two sets:*

$$S^* = \{(x_0, x_1, x_2, \ldots) \ : \ x_i \in S\}$$

$$S^+ = \{(x_0, x_1, x_2, \ldots, x_t) \ : \ x_i \in S, t \in \{1, 2, \ldots\}\}$$

We lay a fairly general setting. Suppose that we have a sequence of random processes of the following form:

- $S_1, S_2, \ldots$ are sets.
- $\Omega_1, \Omega_2, \ldots$ are probability spaces and $\Omega_n \subseteq S_n^*$.
- $y_n : S_n^+ \to \mathbb{R}$ are functions defining a sequence of random variables for each process.

- $Y_n(0), Y_n(1), \ldots$ are such that $Y_n(t)(\omega) = y_n(\omega_t)$ for $\omega \in \Omega_n$. Here, if $\omega = (x_0, x_1 \ldots,)$, then $\omega_t = (x_0, x_1, \ldots, x_t)$. We will drop the n in $Y_n(0)$.

Let $\mathcal{F}_t$ denote the σ-algebra on Ω generated by the partition in which $\alpha, \omega \in S^*$ are in the same part if and only if $\alpha_t = \omega_t$.

We scale variable values and time by a factor of n because this gives them a fixed limiting distribution. This is convenient when considering the solution of the corresponding differential equations because there is only one set of equations rather than different equations for each n.

Theorem 18. *(Wormald) If $f : \mathbb{R}^2 \to \mathbb{R}$ is a function and $D \subseteq \mathbb{R}^2$ is a bounded, open, connected set such that*

(0.) There exists a constant C_0 such that

$$|y(\omega_t)| \leq C_0 n \quad \text{for all } \omega \in \Omega, t \geq 0.$$

(i.) **(boundedness hypothesis)** *There exists a function $\beta = \beta(n) \geq 1$ such that*

$$|y(\omega_{t+1}) - y(\omega_t)| \quad \leq \quad \beta \quad \text{for all } \omega \in \Omega$$

(ii.) **(trend hypothesis)** *There exists a function $\lambda = \lambda(n) = o(1)$ such that*

$$(\tfrac{t}{n}, \tfrac{Y(t)}{n}) \in D \quad \Rightarrow \quad \left| \, E\left[Y(t+1) - Y(t) \mid \mathcal{F}_t\right] \; - \; f(\tfrac{t}{n}, \tfrac{Y(t)}{n}) \, \right| \quad \leq \quad \lambda$$

(iii.) **(Lipschitz)** *f is continuous on D and satisfies a Lipschitz condition on D.*

Then,

(a.) For $(0, z_0) \in D$, the differential equation

$$\frac{dz}{dt} = f(t, z)$$

has a unique solution $z : \mathbb{R} \to \mathbb{R}$ passing through $z(0) = z_0$ and which extends to points arbitrarily close to ∂D.

(b.) For C a sufficiently large constant, there exists a constant k such that

$$\Pr\left(\exists\ 0 \le t \le \sigma n \text{ such that } |Y(t) - nz\left(\tfrac{t}{n}\right)| \ > \ k\lambda n\right) \ = \ O\left(\frac{\beta}{\lambda}\exp\left\{-\frac{n\lambda^3}{\beta^3}\right\}\right)$$

where

$$\sigma \ = \ \sup\{x \ : \ d_\infty(z(x), \partial D) \ge C\lambda\}$$

3.2 Example - A naive number generator

Suppose that for some $c < 1$ we want to generate a random permutation $N_1, N_2, \ldots, N_{cn}$ of numbers from the set $[n] = \{1, 2, \ldots, n\}$, and we use the following naive algorithm:

1. Let $k = 0$.
2. Choose a number r uniformly at random from $[n]$.
   Repeat this step until $r \notin \{N_1, N_2, \ldots, N_k\}$.
3. Let $k = k + 1$, $N_k = r$, and goto step 2.

Clearly choosing i numbers randomly won't result in a list of i distinct elements, there will be some repetition, but we would like to know how practical this algorithm is as $n \to \infty$.

We begin with an informal discussion that motivates the differential equation. For any $i = 0, 1, 2, \ldots$, let K_i be the value of k after exactly i random numbers have been generated. We have $K_0 = 0$ and

$$E[K_{i+1} \mid K_i] \ = \ K_i + 1 - \tfrac{K_i}{n}, \tag{3.5}$$

as exactly $n - K_i$ of the n possible numbers will cause an increase in k by exactly one. Letting $t = \frac{i}{n}$, $\Delta t = \frac{1}{n}$, and $k(t) = \frac{K_i}{n}$, we rewrite (3.5) as

$$\frac{k(t + \Delta t) - k(t)}{\Delta t} \ = \ 1 - k(t),$$

assuming $K_{i+1} = E[K_{i+1}|K_i]$. As $n \to \infty$, and therefore $\Delta t \to 0$, this becomes $k'(t) = 1 - k(t)$, which with $k(0) = 0$ has solution $k(t) = 1 - \mathbf{e}^{-t}$.

Now we go through the analysis formally as an application of Theorem 18. We fit Definition 17 by defining $S_n = \{1, 2, \ldots, n\}$, Ω_n as the probability space on sequences of numbers from the set S_n, $f(t, k) = 1 - k$, $D = \mathbb{R}^2$, and $y_n(\omega_t)$ equals the number of numbers that appear in ω_t. The conditions of Theorem 18 are easily seen to hold with $C_0 = 1$, $\beta = 1$, and $\lambda = \frac{1}{n^{1/4}}$ is sufficiently large to ensure that the probability in Theorem 18 is $o(1)$. Therefore, by Theorem 18, **whp**

$$K_i \quad = \quad (1 - \mathbf{e}^{-i/n})n \pm O(n^{3/4}) \text{ for all } i.$$

So, for example, if we want a permutation of $\frac{1}{2}n$ of the numbers from $[n]$, we see that **whp**, generating μn random numbers for $\mu > \ln 2$ is sufficient, while $\mu < \ln 2$ is not.

3.3 Another example - Random Graphs

Here we look at the classical $G_{n,m}$ random graph where $m = cn$ for some constant $c < \frac{1}{2}$. For any vertex v, define $s(v)$ to be the size of the component containing v. Also, define

$$X \quad = \quad \sum_{v \in V} s(v)$$

Note that X is the sum of the *squares* of the component sizes, and it is also n times the expected component size of a randomly chosen vertex. We look at the series of random variables $X_0, X_1, X_2, \ldots$, where X_i is the value of X after exactly i random edges have been added. Certainly $X_0 = n$, as $s(v) = 1$ for all v.

Claim 19. *Assume that all components are of size $O(\log n)$. Then, we have*

$$E[X_{i+1} - X_i \mid X_i] \quad = \quad 2X_i^2 - o(1). \tag{3.6}$$

Proof: Let $e = \{u, v\}$ be the $(i+1)$-st randomly chosen edge, and let Y_u, Y_v be the sizes of their two components. If Y_u, Y_v are in different components then we have

$$X_{i+1} - X_i \quad = \quad (Y_u + Y_v)^2 - Y_u^2 - Y_v^2 \quad = \quad 2Y_uY_v, \tag{3.7}$$

because X_i is the sum of the squares of the component sizes. If they are in the same component then $X_i = X_{i+1}$ because no component sizes are changed by adding a loop.

In general, we have

$$E[X_{i+1} - X_i|X_i] \quad = \quad E[2Y_uY_v|X_i] \quad + \quad E[X_{i+1} - X_i - 2Y_uY_v|X_i]$$

by linearity of expectation. The first summand is $2X_i^2$ because X_i is the expected component size of a randomly chosen vertex.

The second summand is 0 if u, v are in different components by (3.7), however if they are in the same component then $X_{i+1} - X_i = 0$. In this case, it takes value between $-2C^2$ and 0, where $C = O(\log n)$ is the size of the largest component in the graph. By our assumption, u and v are in the same component with probability $O(\frac{\log n}{n})$, so we can bound the second summand with $-O(\log n)^2 O(\frac{\log n}{n}) = -o(1)$, this yields (3.6).

□

Again we begin with an informal discussion. Letting $t = \frac{i}{n}$, $\Delta t = \frac{1}{n}$, and $x(t) = \frac{X_i}{n}$, letting $n \to \infty$, and assuming $X_{i+1} - X_i = E[X_{i+1} - X_i|X_i]$, we can rewrite (3.6) (dropping the $o(1)$ term) as $x'(t) = 2x(t)^2$. Along with $x(0) = \frac{X_0}{n} = 1$ we have solution $x(t) = \frac{1}{1-2t}$.

To fit Definition 17, we have S_n as the set of all possible edges in $\binom{[n]}{2}$, Ω_n as the set of sequences of edges from the set of these edges, $y_n(\omega_t)$ is X of the graph

given by the edges in ω_t, $\lambda = \frac{1}{n^{1/4}}$, $\beta = \log^3 n$, and $f(t,x) = 2x^2$. The domain D is $[0, \frac{1}{2} - \epsilon] \times [0, \frac{1}{\epsilon}]$ where $\epsilon > 0$ is some small constant, because $\frac{1}{2}$ is the blow-up point of the differential equation.

All of the conditions of Theorem 18 are easy to verify except for the boundedness hypothesis. For the sake of our example, we will also assume that the maximum component is of size $O(\log n)$ when $c < \frac{1}{2}$, as the Erdős and Rényi result has shown us occurs **whp**. This also makes the boundedness hypothesis easy to verify because $O(\log^2 n)$ is an upper bound on the change, and therefore all conditions of Theorem 18 hold. Therefore, **whp** we have

$$X_i \quad = \quad \frac{1}{1 - 2i/n} n \pm O(n^{3/4}) \quad \text{for all } i.$$

The actual value of X_i isn't too important, what matters is that it is finite when $i = cn$ for $c < \frac{1}{2}$. This isn't a proof, but it's at least some intuition for why there is a major change after $c = \frac{1}{2}$. We revisit this example more formally in Chapter 4, where we use susceptibility to show when a giant component occurs as a result of execution of different algorithms for choosing random edges.

The use of this differential equation was suggested by Svante Janson. As we will see in Chapter 4, this also provides motivation for the product rule.

Chapter 4

CREATING A GIANT COMPONENT

Let A be a fixed bounded first edge algorithm. For notational convenience we work with $\mathcal{S}_A$ in a slightly different form. We introduce the symbol ℓ which represents 'all integers larger than m,' and we work with

$$\mathcal{S}_A \subseteq ([m] \cup \{\ell\})^2$$

where $(i, \ell) \in \mathcal{S}_A$ if A accepts e_{t+1} when $a_{t+1} = i$ and $b_{t+1} > m$, etc. We assume without loss of generality that $\mathcal{S}_A \neq ([m] \cup \{\ell\})^2$ (note that if $\mathcal{S}_A = ([m] \cup \{\ell\})^2$ then A always chooses e_{t+1} and we have the standard random graph). We will show (roughly) that the sum of the squares of the component sizes of $G_A(cn)$ is bounded for $c < c_A$ but goes to infinity as c approaches c_A. Furthermore, we will show that for $c < c_A$ the 'large' components do not make a significant contribution to the sum of the squares of the component sizes. Part (b) of Theorem 6 then follows from an application of Lemma 20.

Lemma 20. *Let G be a graph on n vertices and let τ be a constant such that G has $y_i n$ vertices in components of size i for $i = 1, 2, \ldots, \tau$, and $y_1 + \cdots + y_\tau = 1$. If η is a constant such that*

$$2\eta \sum_{i=1}^{\tau} i y_i \; > \; 1, \tag{4.1}$$

then the graph obtained by adding ηn random edges to G will **whp** *have a component of size $\Omega(n)$.*

Note that the summation in (4.1) is the sum of the squares of the component sizes of G. This quantity is also the expected component size of a vertex chosen at random, which is known as the susceptibility in statistical physics. The proof of Lemma 20 is given at the end of this section.

We track $m+2$ random variables over the evolution of the process using the differential equations method for random graph process (we follow the notation and use a variation of a general theorem of Wormald, who gives an excellent treatment of this method [39]). For $x = 1, \ldots, n$ let $Y_x(i)$ be the number of vertices in components of size x in $G_A(i)$. Furthermore, define

$$X(i) = \sum_{x=1}^{n} xY_x(i)$$
$$Z(i) = \sum_{x=1}^{n} x^2 Y_x(i).$$

Note that X gives the sum of the squares of the component sizes while Z gives the sum of the cubes of the component sizes. We track the random variables $Y_1, Y_2, \ldots, Y_m, X, Z$ and show that $Y_1, Y_2, \ldots, Y_m, X$ are concentrated around an expected trajectory. We will only give an upper bound the growth of Z over the course of the algorithm. The relationship between the sum of the squares of the components sizes and the sum of the cubes of the component sizes near the critical value is also a key feature of a recent result of Aldous and Pittel [4] on the emergence of the giant component in a version of the random graph in which both vertices and edges appear as the process evolves.

We define a set of $m+2$ functions on $\mathbb{R}^{m+2}$. For $(z_1, \ldots, z_a, z_X, z_Z) \in \mathbb{R}^{m+2}$ set

$$z_\ell = 1 - z_1 - \cdots - z_a$$
$$\rho = \sum_{(x,y)\in\mathcal{S}_A} z_x z_y.$$

For $w = 1, \ldots, m$ define

$$\begin{aligned} f_w\left(z_1, \ldots, z_a, z_X, z_Z\right) = & \sum_{(x,y)\in\mathcal{S}_A : x+y=w} z_x z_y w \\ & - \sum_{(x,y)\in\mathcal{S}_A : y=w} z_x z_w w - \sum_{(x,y)\in\mathcal{S}_A : x=w} z_y z_w w \\ & + (1-\rho) w \left[\sum_{x=1}^{w-1} z_x z_{w-x} - 2 z_w\right]. \end{aligned}$$

Further define

$$\xi_w = w z_w \qquad \zeta_w = w^2 z_w \qquad \text{for } w = 1, \ldots, m$$
$$\xi_\ell = z_X - \sum_{x=1}^{m} \xi_x \qquad \zeta_\ell = z_Z - \sum_{x=1}^{m} \zeta_x$$

and

$$f_X(z_1, \ldots, z_a, z_X, z_Z) = \sum_{(x,y)\in\mathcal{S}_A} 2\xi_x\xi_y + (1-\rho) 2 z_X^2$$
$$f_Z(z_1, \ldots, z_a, z_X, z_Z) = \sum_{(x,y)\in\mathcal{S}_A} (3\xi_x\zeta_y + 3\xi_y\zeta_x) + (1-\rho) 6 z_X z_Z.$$

Note that $f_1, \ldots, f_m, f_X, f_Z$ are continuous and satisfy a Lipschitz condition on any bounded domain.

We are interested in the solution of the system of differential equations

$$\frac{dz_x}{dt} = f_x(z_1, \ldots, z_m, z_X, z_Z) \qquad x \in [m] \cup \{X, Z\}$$

with initial condition

$$z_1(0) = 1, \quad z_2(0) = \cdots = z_m(0) = 0, \quad z_X(0) = z_Z(0) = 1.$$

Note that $f_1, \ldots, f_m$ do not depend on z_X or z_Z. The solution of the system

$$\begin{gathered} \frac{dz_x}{dt} = f_x(z_1, \ldots, z_m, z_X, z_Z) = f_x(z_1, \ldots, z_m), \qquad x \in [m] \\ z_1(0) = 1, \quad z_2(0) = \cdots = z_m(0) = 0 \end{gathered} \tag{4.2}$$

can be viewed as a collection of function on the non-negative reals.

Claim 21. *The solution of (4.2) satisfies*

$$0 < z_1(t) < 1 \qquad \textit{and} \qquad z_2(t), \ldots, z_m(t), z_\ell(t) > 0$$

for $t > 0$ in some neighborhood of 0.

A proof of Claim 21 is given at the end of this section. Since

$$\begin{aligned} \frac{dz_\ell}{dt} &\geq 0 \quad \text{and} \\ (z_1, \ldots, z_m, z_\ell) \in (0,1)^{m+1} \quad &\Rightarrow \quad \frac{dz_x}{dt} \geq -4x z_x \quad \text{for } x \in [m], \end{aligned}$$

it follows from Claim 21 for any $T > 1/4$ there exists a constant $\delta > 0$ such that

$$z_1(t), z_2(t), \ldots, z_m(t), z_\ell(t) > \delta \qquad \text{for } \; t \in (1/4, T). \tag{4.3}$$

With the functions $z_1, z_2, \ldots, z_m$ in hand we can write

$$\frac{dz_X}{dt} = f_X(z_1, \ldots, z_a, z_X, z_Z) = g_1(t) + g_2(t) z_X + g_3(t) z_X^2 \tag{4.4}$$

where g_1, g_2, g_3 are bounded, smooth functions of t defined on $[0, +\infty)$. The differential equation (4.4) has a unique solution $z_X(t)$ passing through $z_X(0) = 1$. Note that this function blows up at some point (to see this, it may be easier to work with ξ_ℓ instead of z_X). We define c_A to be this blow-up point. With $z_X(t)$ in hand we can write

$$\frac{dz_Z}{dt} = f_Z(z_1, \ldots, z_a, z_X, z_Z) = g_4(t) + g_5(t) z_Z \tag{4.5}$$

where g_4 and g_5 are smooth, bounded, non-negative functions of t on intervals of the form $[0, s)$ where $s < c_A$. It follows that the unique solution $z_Z(t)$ of (4.5) passing through $z_Z(0) = 1$ exists for $0 \leq t < c_A$. In other words z_X and z_Z blow up *at the same point in time*, c_A. Finally, we note that it follows immediately from (4.3) that there exists a constant $\delta > 0$ such that

$$z_1(t), z_2(t), \ldots, z_m(t), z_\ell(t) > \delta \tag{4.6}$$

for $1/4 \leq t \leq 2c_A$.

We add a small wrinkle to the graph process $G_A(1), G_A(2), \ldots$ to produce $G'_A(1), G'_A(2), \ldots$. Set $r(n) = n^{1/13}$. If A calls for the acceptance of an edge $\{x, y\}$ at round i and x or y is in a component of size greater than $r(n)$ then the edge is **not** added to the process (and no edge is added to the graph in the round). This produces the slightly altered graph process $G'_A(1), G'_A(2), \ldots$. The random variables $Y'_1, \ldots Y'_m, X', Z'$ refer to this altered graph process (i.e. $Y'_x(i)$ is the number of vertices in $G'_A(i)$ in components of size x, etc).

Lemma 22. *Let $0 < s < c_A$ be fixed. With probability $1 - O(n^{4/13}\exp(-n^{1/13}))$ we have*

$$\begin{aligned} Y'_x(i) &= nz_x(i/n) + O\left(n^{12/13}\right) \qquad \forall x \in [m] \\ X'(i) &= nz_X(i/n) + O\left(n^{12/13}\right) \quad , \text{ and} \\ Z'(i) &\leq nz_Z(i/n) + O\left(n^{12/13}\right) \end{aligned}$$

uniformly for all $0 \leq i \leq sn$.

The proof of Lemma 22 is given below.

In order to complete the proof of Theorem 6 we must make an observation about the relationship between $G_A(i)$ and $G'_A(i)$. Note that $G'_A(i)$ is *not* simply a subgraph of $G_A(i)$: the edges we neglect in the formation of G'_A may influence future decisions. We use the symbol Δ to denote symmetric difference.

Lemma 23. *Let $0 < s < c_A$ be fixed.*

$$Pr\left[\left|E\left(G_A(sn)\right)\Delta E\left(G'_A(sn)\right)\right| = O\left(\frac{n\log n}{r^2(n)}\right)\right] = 1 - o(1).$$

Proof. For $i = 1, \ldots, sn$ let $L(i)$ be the set of vertices $x \in [n]$ such that the component of $G'_A(i)$ containing x has more than $r(n)$ vertices. Let $D(i)$ be the set of vertices $x \in [n]$ such that the component of $G_A(i)$ containing x is different than the

component of $G'_A(i)$ containing x *and the size of at least one of these components is at most* m. Let B be the set of rounds $i \leq sn$ such that

$$E(G_A(sn)) \cap \{e_i, f_i\} \neq E(G'_A(sn)) \cap \{e_i, f_i\}.$$

Let $\mathcal{A}$ be the event that there exists $i < sn$ such that $|L(i)| > \frac{Kn}{r^2(n)}$, where $K = 2z_Z(s)$. Note that it follows from Lemma 22 that the probability of $\mathcal{A}$ is exponentially small. We have

$$Pr(i+1 \in B) \leq 4\frac{|L(i)| + |D(i)|}{n}.$$

Furthermore, $|D(i+1)| \leq |D(i)| + 3m$. We have

$$\begin{aligned} E[|D(i+1)|] &\leq \sum_{k=0}^{n} Pr[D(i) = k] \left(k + \frac{12m(k + Kn/r^2(n))}{n} \right) + nPr(\mathcal{A}) \\ &= E[|D(i)|] \left(1 + \frac{12m}{n} \right) + \frac{12Km}{r^2(n)} + nPr(\mathcal{A}) \\ &\leq E[|D(i)|] \left(1 + \frac{12m}{n} \right) + \frac{24Km}{r^2(n)} \end{aligned}$$

for n sufficiently large. It follows that

$$\begin{aligned} E[|D(i)|] &\leq \frac{24Km}{r^2(n)} \sum_{j=0}^{i} \left(1 + \frac{12m}{n} \right)^j \\ &= \frac{2Kn}{r^2(n)} \left[\left(1 + \frac{12m}{n} \right)^{i+1} - 1 \right]. \end{aligned}$$

A similar calculation gives the bound $E[|B|] = O\left(n/r^2(n)\right)$. □

Let $c < c_A$. The largest component in $G'_A(cn)$ has at most $2r(n)$ vertices. It then follows from Lemma 23 that **whp** the largest component in $G_A(cn)$ is of size $O(n \log n/r(n))$. This establishes part (a) of Theorem 6.

4.1 Proof of Part (b) of Theorem 6

Let $\epsilon > 0$ and $c = c_A + \epsilon$. Let K_1 be a constant such that

$$\epsilon\delta^2 K_1 > 1. \tag{4.7}$$

(Recall that δ is defined in (4.6).) There exists $t < c_A$ and a constant K_2 such that

$$z_X(t) > K_1 + 3 \text{ and } z_Z(t) < K_2 - 1.$$

It follows from Lemma 22 that **whp** we have

$$X'(tn) > (K_1 + 2)n \text{ and } Z'(tn) < K_2 n.$$

Thus

$$\sum_{x=K_2}^{\infty} xY'_x(tn) \le \frac{1}{K_2} \sum_{x=K_2}^{\infty} x^2 Y'_x(tn) < \frac{1}{K_2} Z'(tn) < n.$$

It follows that

$$\sum_{x=1}^{K_2} xY'_x(tn) > (K_1 + 1)n.$$

Now, it follows from Lemma 23 that $o(n)$ of the components in $G'_A(tn)$ intersect edges in $E\left(G_A(tn)\right) \Delta E\left(G'_A(tn)\right)$. Therefore,

$$\sum_{x=1}^{K_2} xY_x(tn) > K_1 n.$$

We note that $\left(\frac{i}{n}, \frac{Y_1(i)}{n}, \ldots, \frac{Y_m(i)}{n}\right)$ follows $(i/n, z_1(i/n), \ldots, z_m(i/n))$ well past the critical value.

Lemma 24. *Let $0 < s < 2c_A$ be fixed. With probability*

$$1 - O\left(n^{1/4} \exp\left(\frac{-n^{1/4}}{8m^3}\right)\right)$$

we have

$$Y_x(i) = nz_x(i/n) + O\left(n^{3/4}\right)$$

for all $x \in [m]$, uniformly for all $0 \le i \le sn$.

Lemma 24 follows from a routine application of Theorem 5.1 of [39]. It follows from Lemma 24 and (4.6) that the probability that f_i is chosen by A is at least δ^2 for $i = tn, \ldots, cn$. Therefore, **whp** at least

$$\frac{3}{4}\delta^2(c - t)n > \frac{3}{4}\delta^2 \epsilon n$$

edges f_i such that $tn < i < cn$ are chosen by A. As these are purely random edges, part (b) of Theorem 6 follows from an application of Lemma 20.

It remains to prove Lemma 20, Claim 21, and Lemma 22.

Proof of Lemma 22. We apply Theorem 5.1 of [39]. Let D be the domain

$$D = [0, c_A] \times [0,1]^m \times [0, 2z_X(s)] \times [0, 2z_Z(s)].$$

The stopping time T is the smallest i for which

$$\left(\frac{i}{n}, \frac{Y_1'(i)}{n}, \ldots, \frac{Y_m'(i)}{n}, \frac{X'(i)}{n}, \frac{Z'(i)}{n}\right)$$

does not lie in the domain D.

For note keeping purposes we set

$$Y_\ell'(i) = \sum_{x=m+1}^{2r(n)} Y_x'(i) = n - \sum_{x=1}^{m} Y_x'(i)$$

$$Y_r'(i) = \sum_{x=r(n)+1}^{2r(n)} Y_x'(i)$$

$$X_\ell'(i) = \sum_{x=m+1}^{2r(n)} xY_x'(i) = X'(i) - \sum_{x=1}^{m} xY_x'(i)$$

$$X_r'(i) = \sum_{x=r(n)+1}^{2r(n)} xY_x'(i)$$

$$X_x'(i) = xY_x'(i), \quad Z_x'(i) = x^2 Y_x'(i) \qquad \text{for } \; x = 1, \ldots, m$$

$$p(i) = \sum_{(x,y)\in\mathcal{S}_A} \frac{Y_x'(i)Y_y'(i)}{n^2}.$$

Note that $p(i)$ gives the probability that the edge e_{t+1} is chosen. Note further that concentration of $X_\ell'(i)$ and $Y_\ell'(i)$ around some expected trajectory will follow from the concentration results we obtain for $Y_1'(i), Y_2'(i), \ldots, Y_a'(i), X'(i)$. We will not establish concentration of $Y_r'(i)$ or $X_r'(i)$. However, by bounding $Z'(i)$ we will have a trivial upper bound on $Y_r'(i)$ and $X_r'(i)$.

We are now ready to consider the expected changes in our random variables that result from one step of the process. We use h_i to denote the history of the process up to time i (this is just the sequence of edges $e_1, f_1; e_2, f_2; \ldots; e_i, f_i$). For $1 \leq u \leq m$ we have

$$\begin{aligned}
E[Y'_u(i+1) - Y'_u(i)|h_i] = & \sum_{(x,y)\in\mathcal{S}_A : x+y=u} \frac{Y'_x(i)Y'_y(i)u}{n^2} \\
& - \sum_{(x,y)\in\mathcal{S}_A : y=u} \frac{Y'_x(i)Y'_u(i)u}{n^2} - \sum_{(x,y)\in\mathcal{S}_A : x=u} \frac{Y'_u(i)Y'_y(i)u}{n^2} \\
& + (1-p(i))u \left[\sum_{v=1}^{u-1} \frac{Y'_v(i)Y'_{u-v}(i)}{n^2} - 2\frac{Y'_u(i)}{n} \right] \\
& - \mathbf{1}_{(u/2,u/2)\in\mathcal{S}_A} \frac{Y'_{u/2}(i)u^2/2}{n^2} + 2\mathbf{1}_{(u,u)\in\mathcal{S}_A} \frac{Y'_u(i)u^2}{n^2} \\
& + \mathbf{1}_{(u,\ell)\in\mathcal{S}_A} \frac{Y'_u(i)Y'_r(i)u}{n^2} + \mathbf{1}_{(\ell,u)\in\mathcal{S}_A} \frac{Y'_u(i)Y'_r(i)u}{n^2} \\
& + (1-p(i))u \left[-\mathbf{1}_{u \text{ even}} \frac{Y'_{u/2}(i)u/2}{n^2} \right. \\
& \qquad \left. + 2\frac{Y'_u(i)u}{n^2} + 2\frac{Y'_u(i)Y'_r(i)}{n^2} \right]
\end{aligned}$$

The first three lines of this expression give the expected change in Y'_u under the assumption that e_{i+1} and f_{i+1} neither fall within a connected component nor touch a component of size greater than $r(n)$ (the first two lines give the change when e_{i+1} is chosen while the third line gives the change when f_{i+1} is chosen). The other lines account for the these other possibilities. It follows that for $i < T$ we have

$$\begin{aligned}
\left| E[Y'_u(i+1) - Y'_u(i)|h_i] - f_u\left(\frac{Y'_1(i)}{n}, \ldots, \frac{Y'_m(i)}{n}, \frac{X'(i)}{n}, \frac{Z'(i)}{n}\right) \right| & \\
= O\left(\frac{1}{n}\right) + O\left(\frac{Y'_r(i)}{n}\right) & \qquad (4.8) \\
= O\left(\frac{1}{n}\right) + O\left(\frac{1}{r(n)^2}\right). &
\end{aligned}$$

Note that we use the fact that $i < T$ implies $Z'(i) < 2z_Z(s)n$ and therefore $Y'_r(i) = O(n/r(n)^2)$.

The expected changes for X' and Z' are more delicate. The key to the calculation for X' is the following observation: If H is an arbitrary graph with connected components $C_1, \dots C_m$, we set $X(H) = \sum_{i=1}^{m} |C_i|^2$ and we add a single random edge to H to form the graph H^+ then the expected value of $X(H^+) - X(H)$ is

$$\sum_{i \neq j} \frac{|C_i||C_j|}{n^2} 2|C_i||C_j| = 2\frac{X^2(H)}{n^2} - 2\sum_{i=1}^{m} \frac{|C_i|^4}{n^2}. \tag{4.9}$$

The idea is that $2X^2/n^2$ will be the main term and, so long as no large components have appeared, the other term is just a small error (this reasoning is attributed to Janson [36]). We have

$$\begin{aligned}
E[X'(i+1) - X'(i)|h_i] = &\sum_{(x,y)\in\mathcal{S}_A} \frac{2X'_x(i)X'_y(i)}{n^2} + (1-p(i))\frac{2X'(i)^2}{n^2} \\
&- \sum_{x:x\neq\ell,(x,x)\in\mathcal{S}_A} \frac{2x^3Y'_x(i)}{n^2} - \mathbf{1}_{(\ell,\ell)\in\mathcal{S}_A} \sum_{x=a+1}^{r(n)} \frac{2x^3Y'_x(n)}{n^2} \\
&- (1-p(i)) \sum_{x=1}^{r(n)} \frac{2x^3Y'_x(i)}{n^2} \\
&- \sum_{(x,y)\in\mathcal{S}_A:x=\ell} \frac{2X'_y(i)X'_r(i)}{n^2} - \sum_{(x,y)\in\mathcal{S}_A:y=\ell} \frac{2X'_x(i)X'_r(i)}{n^2} \\
&+ \mathbf{1}_{(\ell,\ell)\in\mathcal{S}_A} \frac{2X'_r(i)^2}{n^2} + (1-p(i))\frac{-4X'(i)X'_r(i) + X'_r(i)^2}{n^2}
\end{aligned}$$

The second and third lines account for the sum of the fourth powers of the component sizes as pointed out in (4.9). The fourth and fifth lines account for the fact that we drop edges that touch components that have more than $r(n)$ vertices. It follows that for $i < T$ we have

$$\begin{aligned}
&\left| E[X(i+1) - X(i)|h_i] - f_X\left(\frac{Y'_1(i)}{n}, \dots, \frac{Y'_m(i)}{n}, \frac{X'(i)}{n}, \frac{Z'(i)}{n}\right)\right| \\
&\qquad = O\left(\frac{r(n)}{n}\right) + O\left(\frac{X'_r(i)}{n}\right) \\
&\qquad = O\left(\frac{r(n)}{n}\right) + O\left(\frac{1}{r(n)}\right).
\end{aligned} \tag{4.10}$$

Note that we use the fact that $i < T$ implies both $Z'(i) = O(n)$ and $Y'_r(i) = O(n/r(n)^2)$.

For Z', we note that if H is an arbitrary graph with connected components $C_1, \ldots C_m$, we define $Z(H)$ to be $\sum_{i=1}^{m} |C_i|^3$ and we add a single random edge to H to form the graph H^+, then the expected value of $Z(H^+) - Z(H)$ is

$$\sum_{i \neq j} \frac{|C_i||C_j|}{n^2} \left(3|C_i|^2|C_j| + 3|C_i||C_j|^2\right) = 6\frac{X(H)Z(H)}{n^2} - 6\sum_{i=1}^{m} \frac{|C_i|^5}{n^2}$$
$$\leq 6\frac{X(H)Z(H)}{n^2}.$$

It follows that we have

$$E[Z'(i+1) - Z'(i)|h_i] \leq \sum_{(x,y) \in \mathcal{S}_A} \frac{3X'_x(i)Z'_y(i) + 3X'_y(i)Z'_x(i)}{n^2}$$
$$+ (1 - p(t))\frac{6X'(t)Z'(t)}{n^2}.$$

Note that we do not have to take into account the edges that touch vertices in components having $r(n)$ or more vertices, as doing so would only result in a stronger upper bound. Also note that this function is *increasing* in $Z'(i)$. We have

$$E[Z'(i+1) - Z'(i)|h_i] \leq f_Z\left(\frac{Y'_1(i)}{n}, \ldots, \frac{Y'_m(i)}{n}, \frac{X'(i)}{n}, \frac{Z'(i)}{n}\right). \tag{4.11}$$

We now apply Theorem 5.1 of [39] (conforming to the notation there as much as possible). We set

$$\beta(n) = 8r(n)^3 = 8n^{3/13}, \quad \lambda(n) = \frac{1}{r(n)} = \frac{1}{n^{1/13}} \quad \text{and} \quad \gamma(n) = 0.$$

Note that the boundedness hypothesis follows from the fact that we do not add edges that touch the component that have $r(n)$ or more edges. The trend hypothesis for variable $Y'_1, \ldots, Y'_m, X'$ follows from (4.8) and (4.10). Note that we have only a one-sided trend hypothesis for Z'. A minor alteration of the proof of Theorem 5.1 in [39] can account for this because

(a) z_Z does not appear in any of the functions other than f_Z, and

(b) f_Z is increasing in z_Z.

It follows from (a) that a one-sided bound on Z' does not influence the bounds for any other variable. It follows from (b) that having only a one-sided bound on Z' suffices to establish a one-sided bound on Z' in future iterations.

Finally, we note that in our application of Theorem 5.1 of [39] we violate one of the stated conditions: There is not a constant C_0 such that $|y_l(h_i)| < C_0 n$ for all h_i. However, this condition is not necessary as we have $\gamma(n) = 0$.

□

Proof of Lemma 20. Let $G = ([n], E)$ and $E' \subseteq \binom{[n]}{2}$ be a set of ηn edges chosen uniformly at random. Let $C_1, \ldots, C_\kappa$ be the connected components of G. Let H be the graph with vertex set $\{v_{C_1}, \ldots, v_{C_\kappa}\}$ and an edge between vertices v_{C_i} and v_{C_j} if there is an edge in E' between components C_i and C_j. We have

$$Pr\left(deg(v_C) = i\right) = \binom{2\eta n}{i} \left(1 - \frac{|C|}{n}\right)^{2\eta n - i} \left(\frac{|C|}{n}\right)^i.$$

It follows that **whp** H has

$$d_n(i) := \sum_{k=1}^{\tau} \frac{a_k n}{k} \frac{(2\eta k)^i}{i!} e^{-2\eta k} + o\left(n^{2/3}\right) \tag{4.12}$$

vertices of degree i for each $i \leq O(\log n / \log\log n)$ and no vertices of higher degree.

If we condition on the degree sequence, then we can view H as a graph chosen uniformly at random from the collection of all graphs having the given degree sequence. When (4.12) holds we apply a theorem of Molloy and Reed on the giant component in graphs with a fixed degree sequence. Following the notation of [33], we set

$$\lambda_i = \sum_{k=1}^{\tau} \frac{a_k}{k} \frac{(2\eta k)^i}{i!} e^{-2\eta k}$$

and $\mathcal{D} = d_n(1), d_n(2), \ldots$. If (4.12) holds then we have

$$\begin{aligned} Q(\mathcal{D}) &= \sum_{i\geq 1} i(i-2)\lambda_i \\ &= \sum_{i\geq 1} i(i-2)\left[\sum_{k=1}^{\tau} \frac{a_k}{k}\frac{(2\eta k)^i}{i!}e^{-2\eta k}\right] \\ &= 2\eta\left[\sum_{k=1}^{\tau}(2\eta k - 1)a_k\right] \\ &> 0. \end{aligned}$$

It follows from Theorem 1(a) of [33] that **whp** H has a component of size $\Omega(n)$. Therefore, $G + E'$ has a component of size $\Omega(n)$. □

Proof of Claim 21. It follows from a simple induction argument that

$$z_k^{(i)}(0) = 0 \quad \text{for} \quad i = 0, \ldots, k-2 \quad \text{and} \quad 2 \leq k \leq m. \tag{4.13}$$

Define $w_k = 1 - z_1 - z_2 - \cdots - z_k$ for $k = 1, \ldots, m$. Note that $z_\ell = w_m$. Again by induction we have

$$w_k^{(i)}(0) = 0 \quad \text{for} \quad i = 0, \ldots, k-1 \quad \text{and} \quad 1 \leq k \leq m. \tag{4.14}$$

It follows from (4.13) and (4.14) that for all $\delta > 0$ there exists $\epsilon > 0$ such that

$$|z_k(t)| < \frac{\delta}{(k-2)!}t^{k-2} \quad \text{for} \quad 0 \leq t \leq \epsilon \quad \text{and} \quad 2 \leq k \leq m \tag{4.15}$$

$$|w_k(t)| < \frac{\delta}{(k-1)!}t^{k-1} \quad \text{for} \quad 0 \leq t \leq \epsilon \quad \text{and} \quad 1 \leq k \leq m \tag{4.16}$$

Let m' be the smallest integer greater or equal to 2 such that there does not exist $(x, y) \in \mathcal{S}_A$ such that $x + y = m'$. Another inductive argument gives (choosing ϵ sufficiently small)

$$z_k(t) = \Theta(t^{k-1}) \quad \text{for} \quad 0 \leq t \leq \epsilon \quad \text{and} \quad 1 \leq k < m' \tag{4.17}$$

$$w_k(t) = \Theta(t^k) \quad \text{for} \quad 0 \leq t \leq \epsilon \quad \text{and} \quad 1 \leq k < m' - 1. \tag{4.18}$$

Note that this gives the full Claim if $m' \geq m+2$.

Suppose that $m' \leq m+1$. A careful analysis of ρ (using (4.15), (4.17) and (4.18)) gives

$$1-\rho = \Theta\left(t^{m'-2}\right) \quad \text{for} \quad 0 \leq t \leq \epsilon.$$

It then follows that

$$z_k(t) > 0 \quad \text{for } m' \leq k \leq m \quad \text{and } 0 < t \leq \epsilon.$$

Finally,

$$z_\ell(t) > 0 \quad \text{for } 0 < t < \epsilon.$$

□

4.2 Online Lower Bound

Let c be a constant and let $(e_1, f_1), (e_2, f_2), \ldots, (e_{cn}, f_{cn})$ be a sequence of ordered pairs of edges on vertex set $[n]$ chosen independently and uniformly at random. Define

$$f(d) \;=\; \frac{3+8d-8de^{-4d}-14de^{-12d}+14de^{-16d}}{20-8e^{-4d}-14e^{-12d}+14e^{-16d}}. \tag{4.19}$$

Note that $f(d)$ has a local maximum when $d \approx 0.019974$, where $f(d) \approx .2545$.

We will prove Theorem 8 by showing that for any $c > 0$, if there exists $d \in (0, c)$ such that $c < f(d)$ (i.e. if $c \leq .2545\ldots$) then any online algorithm for the sequential choice of one edge from each presented pair (e_i, f_i), $i = 1, 2, \ldots, cn$, **whp** will create a graph whose components are all of size $O(\log n)$. To do this, we will use the fact that online algorithms usually cannot make "good" choices early in the process. For small i, there is a good chance that all four vertices from (e_i, f_i) are appearing for the first time. In this case any online algorithm must make an essentially arbitrary choice, which could be a costly mistake if, for example, f_i is an isolated edge in the graph consisting of all $2cn$ edges but e_i is a bridge in the giant component.

Fix some constant $d \in (0, c)$. We will determine its value later. Divide the process into two parts, the first dn pairs and the last $(c-d)n$ pairs. Given the pairs $(e_1, f_1), \ldots, (e_{cn}, f_{cn})$, we create an auxiliary graph G on vertex set $[n]$ using the following steps:

(i.) For all $i < dn$, if all four vertices in (e_i, f_i) are occurring for the first time in the process, then randomly eliminate one of the edges with probability $\frac{1}{2}$.

(ii.) Take **all** remaining edges, including the last $2(c-d)n$ edges, and let this be the edge set of G.

In other words, G will contain every edge except for some that were chosen with probability $\frac{1}{2}$. It follows from symmetry that for any fixed algorithm A, the probability that A produces a giant is bounded above by the probability that G has a giant. Indeed, if we condition on the first dn rounds of the process then there is a permutation of $[n]$ that maps the graph produced by A in the first dn rounds to a subgraph of the graph consisting of the edges in G that come from $(e_1, f_1), \ldots, (e_{dn}, f_{dn})$.

In order to analyze G we consider a slightly different probability space. Let $a_1, b_1, c_1, d_1; a_2, b_2, c_2, d_2; \ldots$ be a sequence of vertices chosen uniformly and independently at random from $[n]$ with replacement (e.g. we allow $u_i = v_i$). Setting $e_i = \{a_i, b_i\}$ and $f_i = \{c_i, d_i\}$, we get a sequence of pairs of random edges. Of course, this model allows loops and multiple edges, but since the expected number of each over the cn rounds of our process is bounded by an absolute constant, **whp** the total number of loops and multiple edges is at most $\log(n)$. Thus, we can accommodate these flaws in the process by adding an extra $\log(n)$ rounds (which have no impact on the rest of argument).

The following is a variation of the branching process proof of the classical giant component results for $G_{n,p}$ which is given in [24, p. 109]. We generate an upper bound on the size of the component of G containing a fixed vertex v with the following process.

We begin by observing the locations of all appearances of v in the sequence $a_1, b_1, c_1, d_1; a_2, b_2, c_2, d_2; \ldots$. Note that we have not yet observed any whole edge and, conditioning on these locations, the rest of the random vertices in the sequence are uniform in $[n] \setminus \{v\}$. The edge-partners of each occurrence of v are now **active positions** (we still haven't looked at the vertices in these positions). An active position in round i such that $i \leq dn$ is called an **early** active position. An active position in round i such that $i > dn$ is called a **late** active position.

Now, if the first appearance of v is in round i and $i \leq dn$ then we reveal the other vertices in round i. Let x be the edge-partner of v in this round and let y and z be the other two vertices that appear in this round. We check to see if there is appearance of vertices x, y or z *before* round i. In particular, we determine the first occurrence of a vertex from the set $\{x, y, z\}$. If x, y and z do not appear before round i then we delete this active position with probability $\frac{1}{2}$. If x is the first vertex from the set $\{x, y, z\}$ to appear and its first appearance occurs before round i then the position remains active and we call it a **heavy** active position. If y or z is the first vertex from the set $\{x, y, z\}$ to appear then the position remains an early active position. The vertices x, y and z are now **sussed** and the vertex v is **saturated**. Note that so-far this process has simply determined the degree of v and checked to see if the edge containing the first occurrence of v is deleted in the formation of G while revealing as little information about the random graph as possible.

Now suppose that after some number of steps in this process we have given sets of saturated and sussed vertices, and sets of early, heavy, and late active positions. The saturated vertices are vertices in the component containing v whose complete neighborhood has already been determined, and the active positions correspond to vertices in the component containing v for which we have not yet observed the full neighborhood. We now consider the vertex, say vertex u, in an arbitrary active position π. We reveal all other occurrences of u in the sequence

$a_1, b_1, c_1, d_1; a_2, b_2, c_2, d_2; \ldots$ and the edge-partners of all occurrences of u (other than the edge-partner of π which is an already saturated vertex) become active positions (early or late depending on the round in which they occur). If π is an early or heavy active position then we may have already checked to see if the first occurrence of u is in a deleted edge and we cannot make that check again. If π is a late active position and the first occurrence of u is in a round i such that $i \leq dn$, then we suss out the other vertices in round i. If all vertices appearing in round i are making their first appearance, then we delete the active position in round i with probability $\frac{1}{2}$. If, on the other hand, the edge-partner of the first occurrence of u appears before any of the other vertices that appear in round i then this position becomes a heavy active position. This step is repeated until the set of active positions is exhausted.

Before turning to a thoroughly rigorous analysis of this process, it will be helpful to note that it can be loosely modelled with a branching process in which there are three types of offspring: types 1,2 and 3 corresponding to late, early and heavy active positions, respectively. Note that the probability that a late active position generates an early active position that is deleted can (roughly speaking) be bounded from below by

$$\tfrac{1}{2}\left(1-\left(1-\tfrac{1}{n}\right)^{4dn}\right)\left(1-\tfrac{3}{n}\right)^{4dn} \approx \frac{1-e^{-4d}}{2e^{12d}} =: p.$$

Furthermore, the probability that a late active position generates a heavy active position is (roughly speaking) at most

$$\tfrac{1}{3}\left(1-\left(1-\tfrac{1}{n}\right)^{4dn}\right)\left(1-\left(1-\tfrac{3}{n}\right)^{4dn}\right) \approx \tfrac{1}{3}(1-e^{-4d})(1-e^{-12d}) =: q.$$

The i, j position in the matrix

$$A = \begin{bmatrix} 4(c-d) & 4d-p & q \\ 4(c-d) & 4d & 0 \\ 4(c-d) & 4d+1 & 0 \end{bmatrix}$$

gives the expected number of offspring of type j of an animal of type i in this multi-type branching process. The largest eigenvalue of A is less than 1 if and only if $c < f(d)$, where $f(d)$ is defined in (4.19). Since this branching process dies out with probability one (see [6, page 186]) if the largest eigenvalue of the matrix A is less than one, we expect $c < f(d)$ to imply that **whp** the component of G containing v is small.

We now make a rigorous argument. Let $\epsilon > 0$ be a constant such that

$$4c + 4(c-d)(2q-p) + 8\epsilon < 1,$$

note that such a constant exists as the condition we impose on c in the statement of the theorem is equivalent to $c < (\frac{1}{4} + d(2q-p))/(1+2q-p))$. Also, set

$$m := \frac{4(4c+\epsilon)}{\epsilon^2}\ \log n.$$

We consider the first m steps in the process. We begin by noting that it may happen that when we reveal the vertex in some position we will find that it has already been sussed (note that the vertices in some of the early active positions and all of the heavy active positions have already been sussed but this fact is built-in to the process). This 'bad' event occurs with probability at most $\frac{3m}{n-m}$ at each position. Since the number of positions that we inspect is at most $4m$, the probability that the bad event occurs more than 1 time in the process is at most $\binom{4m}{2}(\frac{3m}{n-m})^2 = o(\frac{1}{n})$. Furthermore, when this bad event occurs it introduces at most 2 'extra' active positions, which will have no affect on the rest of the argument. We henceforth assume that no vertex that is revealed has been previously sussed.

It may also happen that the first appearance of some vertex that is revealed in a late active position is in a round that also contains a previously viewed vertex (either saturated or sussed). Of course, this event will change the probability of both the deletion of this active position and the probability that this position becomes a heavy active position. The probability of this event is at most $\frac{3}{n-4m}$

times the number of rounds that hold previously viewed vertices. Since the probability that there exists a vertex that appears more than $2\log n$ times in the sequence $a_1, b_1, c_2, d_1; a_2, b_2, c_2, d_2; \ldots$ is $o(\frac{1}{n})$, the probability of this second type of 'bad' event in any step of the process is at most $12m\frac{\log n}{n-4m}$. Again, we see that we may assume that this event occurs at most once during the process, with no consequence for the rest of the proof.

Let X be the number of late active positions that are introduced in the first m steps of the process. Since the number of late active positions introduced in any one of these steps is dominated by $\mathrm{Bi}\left(4(c-d)n, \frac{1}{n-m}\right)$, X is dominated by $\mathrm{Bi}\left(4(c-d)mn, \frac{1}{n-m}\right)$. It follows from the Chernoff bound ([24], p. 26), that

$$\Pr\Big(X \le (4(c-d)+\epsilon)m\Big) \;=\; 1 - o(\tfrac{1}{n}). \tag{4.20}$$

Note that we do not necessarily saturate all of the late active positions during the first m steps of the process: if there are many active positions it may be the case that some of the late active positions are left over after the m^{th} step. Let X' be the number of late active positions that are actually saturated during the first m steps.

Let Z be the number of heavy active positions that are generated during this process. The probability that a late active position generates a heavy active position is at most

$$\tfrac{1}{3}\left(1-\left(1-\tfrac{1}{n-4m}\right)^{4dn}\right)\left(1-\left(1-\tfrac{3}{n-4m}\right)^{4dn}\right) = q + O\left(\tfrac{m}{n}\right)$$

So, conditioning on (4.20), Z is dominated by $\mathrm{Bi}\Big((4(c-d)+\epsilon)m, q + O\left(\frac{m}{n}\right)\Big)$. It follows from the Chernoff bound that

$$\Pr\Big(Z < \left(4(c-d)q + 2\epsilon\right)m\Big) \;=\; 1 - o\left(\tfrac{1}{n}\right). \tag{4.21}$$

It remains to bound the number of early active positions that are generated in the first m steps of the process. The probability that a late active position generates an early active position that is deleted is at least

$$\tfrac{1}{2}\left(1-\left(1-\tfrac{1}{n}\right)^{4dn-8dm\log n}\right)\left(1-\tfrac{3}{n-4m}\right)^{4dn} = p - O\left(\tfrac{m\log n}{n}\right)$$

Let W be the number of deleted early positions in the first m steps of the process. W dominates $\mathrm{Bi}\Big(X', p - O(\frac{m \log n}{n})\Big)$, and hence

$$\Pr\Big(W \geq X'(p-\epsilon)\Big) \;=\; 1 - o\left(\tfrac{1}{n}\right). \tag{4.22}$$

Let Y be the sum of m i.i.d. $\mathrm{Bi}\left(4dn, \frac{1}{n-4m}\right)$ random variables. We have

$$\Pr\Big(Y \leq (4d+\epsilon)m\Big) \;=\; 1 - o(\tfrac{1}{n}) \tag{4.23}$$

The number of early active positions that are generated (and not deleted) in this process is dominated by $Y + Z - W$.

Now, in the event the component containing v has more than m vertices, the process does not terminate until after the m^{th} step and we have

$$m \leq X' + Z + (Y + Z - W). \tag{4.24}$$

It follows from (4.20), (4.21), (4.22) and (4.23) that, with probability $1 - o(\frac{1}{n})$, the right hand side of (4.24) is at most

$$\begin{aligned} X'(1-p+\epsilon) + 2m(4(c-d)q) + 4dm + 5\epsilon m & \\ &\leq 4(c-d)(1-p)m + 2m(4(c-d)q) + 4dm + 8\epsilon m \\ &< m. \end{aligned}$$

Thus, the probability that the component containing v has more than m vertices is $o(\frac{1}{n})$, and the probability that G contains a component having more than m vertices is $o(1)$, and we have proved Theorem 8.

4.3 Offline Phase Transition

Fix $c > \frac{1}{4}$ and suppose $(e_1, f_1), \ldots, (e_{cn}, f_{cn})$ are pairs of random edges. In this section we prove that there exists a set E of edges such that $|E \cap \{e_i, f_i\}| \leq 1$ for $i = 1, \ldots, cn$ and the graph $([n], E)$ has a component of size $\Omega(n)$.

Let G be the graph with vertex set $[n]$ and all $2cn$ edges. For any tree T within graph G, we will say that T **survives** if there is no $i \in \{1, 2, \ldots, cn\}$ such

that $\{e_i, f_i\} \subseteq E(T)$. In other words, T survives if no two edges in T were paired together. Clearly, if T is a tree in G which survives, then it is possible to make choices so all of T ends up in the final graph.

If T is a tree in G with t vertices, let $\phi(t)$ be the probability that it survives. A straightforward calculation shows

$$\phi(t) \;=\; \prod_{i=1}^{t-2} \frac{2cn-2i}{2cn-i} \;=\; \prod_{j=1}^{\frac{t}{2}-1} \frac{2cn-t-2j}{2cn+1-2j} \;=\; \prod_{j=1}^{\frac{t}{2}-1} \left(1 - \frac{t+1}{2cn+1-2j}\right).$$

Bounding using the first and last terms in the product and using $\mathbf{e}^{-2x} \leq 1-x \leq \mathbf{e}^{-x}$ for small x leads to

$$\exp\left(-\tfrac{t^2}{2cn-t}\right) \;\leq\; \phi(t) \;\leq\; \exp\left(-\tfrac{t^2}{4cn} + O(\tfrac{t}{n})\right). \tag{4.25}$$

With high probability, there exists a surviving tree with $\lfloor n^{1/3} \rfloor$ vertices because $\phi(\lfloor n^{1/3} \rfloor) \to 1$ and **whp** G has a component of size $\Omega(n) \geq \lfloor n^{1/3} \rfloor$.

If T is a surviving tree, we will say that T is **maximal** if there is no edge e such that $T \cup \{e\}$ is a surviving tree. So, if T is maximal then for every edge $\{u, v\} \in E(G)$ such that $u \in T$ and $v \notin T$, necessarily $\{u, v\}$ is paired with some edge in T, otherwise we could add it to T and create a larger surviving tree. Define

$$\mathcal{T} \;=\; \{t \;:\; \log^2 n \;\leq\; t \;\leq (c - \tfrac{1}{4})^2 n \;\} \tag{4.26}$$

We will prove:

$$\Pr(\exists \text{ a maximal surviving tree with size } t \in \mathcal{T}) \to 0. \tag{4.27}$$

Since $\lfloor n^{1/3} \rfloor \in \mathcal{T}$, (4.27) establishes the existence of a surviving tree of size at least $(c - \frac{1}{4})^2 n$ **whp**.

For the remainder of this section, note that any given edge appears in G with probability $\frac{2cn}{\binom{n}{2}} = \frac{4c}{n-1}$. Since the appearance of fixed edges makes others less likely, we can say

$$\Pr(\text{a set of } j \text{ edges appears}) \leq (\tfrac{4c}{n-1})^j.$$

Lemma 25. *Let $P = \Pr(T \text{ maximal} | T \text{ survives})$, where T is a tree on K_n with t vertices. We have*

$$P \;\leq\; t\exp\left(-\tfrac{4c}{n-1}t(n-t)\left[1-\tfrac{t}{c(n-1)}-\tfrac{t}{2cn-t}\right]+O\big(\tfrac{t}{n}\big)\right). \tag{4.28}$$

Proof. To find the probability that T is maximal, note that there are $t(n-t)$ edges "leaving" T. Every one of these edges must either be left out of G or be paired with an edge from T. The probability that an edge is paired with something in T is bounded above by $\frac{t}{2cn-t}$. In the following sum, j is the exact number of the $t(n-t)$ edges "leaving" T that appear in G:

$$P \;\leq\; \sum_{j=0}^{t-1}\tbinom{t(n-t)}{j}(\tfrac{4c}{n-1})^j(\tfrac{t}{2cn-t})^j(1-\rho)^{t(n-t)-j},$$

where ρ is a lower bound on the probability that a fixed edge appears in G, conditioned on all edges that have been observed. The probability ρ will take its smallest possible value when $2(t-1)$ edges have already been observed to exist in G while possibly zero edges have been left out. We can say

$$\rho \;\geq\; \frac{2cn-2(t-1)}{\binom{n}{2}-2(t-1)} \;\geq\; \frac{4c}{n-1}-\frac{4t}{(n-1)^2}.$$

Now we will find the maximum possible value of P. We avoid the sum by finding the maximum and multiplying by t.

$$P \;\leq\; t\exp\left(-t(n-t)\left(\tfrac{4c}{n-1}-\tfrac{4t}{(n-1)^2}\right)\right) \cdot \max_{0\leq j<t}\left(t(n-t)\tfrac{4c}{n-1}\exp(\tfrac{4c}{n-1}-\tfrac{4t}{(n-1)^2})\tfrac{t}{2cn-t}\tfrac{\mathbf{e}}{j}\right)^j.$$

The function $(\lambda\mathbf{e}/j)^j$ is maximized at $j=\lambda$ with maximum value $\exp(\lambda)$. After this and substituting $\exp(\frac{4c}{n-1}-\frac{4t}{(n-1)^2}) = 1+O(\frac{1}{n})$, then factoring out $\frac{4c}{n-1}$, we get (4.28). □

Let E_t be the expected number of maximal surviving trees with t vertices. In order to prove (4.27), it is sufficient to show $nE_t \to 0$ for any $t \in \mathcal{T}$. First let

us find the probability p_T that a given tree T with t vertices becomes a maximal surviving tree.

$$p_T \;=\; \Pr(T \subseteq G)\Pr(T \text{ survives}|T \subseteq G)\Pr(T \text{ maximal}|T \text{ survives})$$

It is easy to see that $\Pr(T \subseteq G) \le (\frac{4c}{n-1})^{t-1} \le \frac{n}{4c}(\frac{4c}{n-1})^t$. We have an upper bound for $\Pr(T \text{ survives}|T \subseteq G)$ from (4.25). Combining these with (4.28),

$$p_T \;\le\; \tfrac{n}{4c}(\tfrac{4c}{n-1})^t\; t\exp\left(-\tfrac{4c}{n-1}t(n-t)\left[1-\tfrac{t}{c(n-1)}-\tfrac{t}{2cn-t}\right]-\tfrac{t^2}{4cn}+O\big(\tfrac{t}{n}\big)\big)\right).$$

Since there are $\binom{n}{t}t^{t-2} \le \frac{1}{t^2}(n\mathbf{e})^t$ trees on K_n with t vertices, we have $E_t \le \frac{1}{t^2}(n\mathbf{e})^t p_T$, so

$$nE_t \;\le\; \tfrac{n^2}{4ct}(4c\mathbf{e})^t\left\{\exp\left(-\tfrac{4c}{n-1}(n-t)t\left[1-\tfrac{t}{c(n-1)}-\tfrac{t}{2cn-t}\right]-\tfrac{t^2}{4cn}+o(t)\right)\right\},$$

leading to

$$\begin{aligned} nE_t \;\le\; &\exp\left(2\log n + t\log(4c\mathbf{e})\right) \\ &\cdot\exp\left(-\tfrac{4c}{n-1}(n-t)t\left[1-\tfrac{t}{c(n-1)}-\tfrac{t}{2cn-t}\right]-\tfrac{t^2}{4cn}+o(t)\right). \end{aligned}$$

We have $2\log n = o(t)$. Take a factor of t out of the exponent, then let $t = \alpha n$.

$$nE_t \;\le\; \exp\left(\log(4c\mathbf{e}) - 4c(1-\alpha)\left[1-\tfrac{\alpha}{c}-\tfrac{\alpha}{2c-\alpha}\right]-\tfrac{\alpha}{4c}+o(1)\right)^t.$$

The inside is increasing over all values of $\alpha \ge 0$ provided $c > \frac{1}{4}$, and it is negative whenever $\alpha < (c-\frac{1}{4})^2$. Therefore, $nE_t \to 0$ whenever $t \in \mathcal{T}$. We have proved Theorem 9.

4.4 Two Algorithms

We begin with the bounded first-edge algorithm A_1. Using the notation of Section 2 we have $m = 1$ and $\mathcal{S}_{A_1} = \{(\ell,\ell)\}$. In words, A_1 chooses edge e_i if it

includes no isolated vertices and otherwise chooses edge f_i. For this algorithm we have

$$\begin{aligned} \frac{dz_1}{dt} &= -2z_1(2z_1 - z_1^2) \\ \frac{dz_X}{dt} &= 2(z_X - z_1)^2 + 2z_X^2(2z_1 - z_1^2) \end{aligned}$$

with initial conditions $z_1(0) = 1$, $z_X(0) = 1$. In order to get an upper bound on the critical value c_{A_1}, we approximate the solution by implementing Euler's method, being sure to underestimate z_X. (The program for our approximation, written in C++, is available at `http://www.math.cmu.edu/~tbohman`.)

Let $\tilde{z}_1$ and $\tilde{z}_X$ denote our approximations. We use the standard Euler's method for $\tilde{z}_1$. For $\tilde{z}_X$ we set

$$\begin{aligned} \tilde{z}_X(t+h) &= \tilde{z}_X(t) + h\phi(\tilde{z}_X, \tilde{z}_1) - 2\epsilon, \quad \text{where} \\ \phi(\tilde{z}_x, \tilde{z}_1) &= 2\max\left(\tilde{z}_X - \tilde{z}_1 - 2\tfrac{\epsilon}{h}, 0\right)^2 + 2\tilde{z}_X^2(2\tilde{z}_1 - \tilde{z}_1^2 - \tfrac{2\epsilon}{h}), \end{aligned}$$

h is the step size, and ϵ accounts for the computational rounding errors. For our approximation we take $h = 10^{-7}$, and $\epsilon = 10^{-12}$ suffices. We claim that we have the following

(a.) $\tilde{z}_X(t) < z_X(t)$

(b.) $\tilde{z}_X(0.3847) > 10^4$, and

(c.) $z_1(t) > 1/2$ for $t \in [0, 0.385]$.

It then follows from Lemmas 20 and 22 that $c_{A_1} < 0.3847 + 0.0001 < 0.385$. In order to establish these claims we first note that

(d.) $e(t) := |z_1(t) - \tilde{z}_1(t)| \leq \frac{2\epsilon}{h}$ for $t \in [0, 0.4]$.

To see this, let $f_1(z_1) = -2z_1(2z_1 - z_1^2)$. Since $|z_1''(t)| \leq 4.2$ and $|f_1(z_1(t)) - f_1(\tilde{z}_1(t))| \leq 3e(t)$ for all $t \in [0, 0.4]$, we have

$$e(t+h) \leq (1+3h)e(t) + 2.1h^2 + \epsilon.$$

This leads to (d.). We note that (d.) together with observation of $\tilde{z}_1$ suffices to establish (c.). Of course, (b.) follows from the observation of $\tilde{z}_X$ alone. It remains to prove (a.). To this end, we first note that

(e.) $z''_X(t) > 0$ for all $t > 0$.

We have

$$z''_X \;=\; 2(z_X - z_1)(z'_X - z'_1) + 2z_X z'_X z_1(2 - z_1) + 2z_X^2(1 - z_1)z'_1,$$

and therefore we may use $-2 \leq z'_1 \leq 0$, $z_X \geq 1$, and $z_1 \geq \frac{1}{2}$ to get

$$z''_X \;\geq\; 2(1 - z_1)[z'_X + \tfrac{1}{2}z_X z'_X - 2z_X^2] \;\geq\; \tfrac{3}{2}z'_X - 2z_X^2.$$

It is easy to see from the original differential equation that $z'_X \geq \frac{3}{2}z_X^2$ whenever $z_1 \geq \frac{1}{2}$, thus $z''_X \geq \frac{1}{4}z_X^2 > 0$. Finally, we note that

(f.) $\tilde{z}_X(t) \leq z_X(t)$ implies $\phi(\tilde{z}_X(t), \tilde{z}_1(t)) < z'_X(t)$.

Claim (a.) now follows from $z_X(0) = \tilde{z}_X(0)$, (e.) and (f.).

We now turn to the algorithm of Conjecture 3 and apply our machinery to determine the existence and location of a phase transition. Let the bounded first-edge algorithm A_2 defined by $m = 1$ and $\mathcal{S}_{A_2} = \{(1, 1)\}$. In words, this algorithm chooses edge e_i if it is an isolated edge and otherwise chooses f_i. For this algorithm we have

$$\frac{dz_1}{dt} = -2z_1(t)^2 - 2z_1(t)(1 - z_1(t)^2)$$
$$\frac{dz_X}{dt} = 2z_1(t)^2 + 2z_X^2(t)(1 - z_1(t)^2)$$

Using the methods above to approximate the solution of this system of differential equations we have $z_X(.5882) > 10^4$. This implies $c_{A_2} < .589$.

We close this section by noting that simulations suggest that there the Achlioptas processes the chooses the edge from the pair (e_i, f_i) that results in the largest

increase in the sum of the squares of the component sizes creates a giant component in as few as $0.34n$ rounds. However, this algorithm is not a not a first-edge algorithm: it depends on all four vertices in $e_i \cup f_i$. For such an algorithm it is a considerable challenge to bound the errors in the numerical simulations.

4.5 A simple Achlioptas process.

In this section we give a simple Achlioptas process that succeeds **whp** in creating giant for any $c > \frac{3\sqrt{6}}{16}$.

Begin by fixing a set $S \in \binom{[n]}{\alpha n}$ for some $\alpha \in (0,1)$. During each round, choose only edges which are in $\binom{S}{2}$. If two such edges are presented, choose one at random. If no such edges are presented, choose neither.

In each round, the probability that we choose an edge from $\binom{S}{2}$ is $2\alpha^2 - \alpha^4$. Furthermore, in each round, any edge in $\binom{S}{2}$ is equally likely. Therefore, **whp** we will take $(2\alpha^2 - \alpha^4 + o(1))cn$ edges at random from $\binom{S}{2}$. So **whp** we have a giant component whenever

$$(2\alpha^2 - \alpha^4 + o(1))cn \quad > \quad \tfrac{1}{2}\alpha n. \tag{4.29}$$

To optimize (4.29), divide by α and note that the left side is minimized when $\alpha^2 = \frac{2}{3}$. Therefore, we create a component of size $\Omega(n)$ **whp** whenever $c > \frac{3\sqrt{6}}{16} = 0.459\ldots$.

Chapter 5

OFFLINE SATISFIABILITY

First we restate the theorems.

Theorem 11. If G is a graph with $2n$ vertices, less than $(1-\epsilon)n$ edges for some $\epsilon > 0$, and $\Delta(G) = o(\frac{n^{1/10}}{\log n})$, then $S(G)$ is satisfiable **whp**.

Theorem 13. If G is a graph with $2n$ vertices and $\Delta(G) = o(n^{1/8})$, and there is some $\epsilon > 0$ and function $\tau \leq c \log n$ for some constant $c < \frac{3\epsilon}{16}$ such that

$$\sum_{0}^{\tau} i d_i = (1+\epsilon)2n, \tag{5.1}$$

then $S(G)$ is not satisfiable **whp**.

5.1 When d_0 is small

The proof of Theorem 13 will use the following. If G has few isolated vertices, then it is not satisfiable provided at least some ratio of the vertices have degree 2 or more.

Theorem 26. *If G is a $2n$ vertex graph such that*

$$\sum_{i \geq 2} d_i \quad >> \quad n^{7/8}\Delta^{1/2} + n^{1/2} d_0^{1/2} \tag{5.2}$$

then $S(G)$ is not satisfiable **whp**.

Note that in this case $\Delta = o(n^{1/4})$ and $d_0 = o(n)$ are implied since $\sum_{i \geq 0} d_i = 2n$.

Proof of Theorem 26 Suppose that G is any graph with $2n$ vertices. Begin by iteratively removing any edges which join two vertices of degree at least 3. Note

that this doesn't change n or (5.2), and when finished it will allow us to say that G satisfies

(a.) Every edge in G is incident with at least one vertex of degree 1 or 2.

Now define functions $\alpha(n)$ and $\mu(n)$ which satisfy the following:

(b.) $\Delta(G) \leq \frac{n^{1/4}}{\alpha^2(n)}$.

(c.) $d_0 \leq \frac{n}{\alpha^2(n)}$.

(d.) Either (b.) or (c.) is satisfied with equality.

(e.) $d_1 = 2n(1 - \mu(n))$.

(f.) $\alpha(n)\mu(n) \to \infty$ as $n \to \infty$.

Existence of $\alpha(n)$ is clear from the conditions of Theorem 26, and (e.) defines $\mu(n)$. To show that (5.2) also implies (f.), note that

$$\alpha\mu \quad \geq \quad \frac{\alpha}{2n}\sum_{i\geq 2} d_i \quad >> \quad \alpha n^{-1/8}\Delta^{1/2} + \alpha n^{-1/2} d_0^{1/2} \quad \geq \quad 1,$$

with the last inequality coming from (d.).

First, we will pick any non-isolated vertex v_0 from G. Start by setting v_0 false, we are going to prove that **whp**, this will lead to a contradiction. To do this, we are going to expose the matching of G one edge at a time and simultaneously keep track of the following three sets:

- T is the set of "active" true vertices, vertices which must be true but are not yet matched. Our contradiction will be a matching edge within T. Initially $T = N(v_0)$ since v_0 is false, and $T \neq \emptyset$ when we start because v_0 is a non-isolated vertex.

- U is the set of all unmatched vertices which are considered "free" because at least one of their neighbors was set, or because they are isolated. Initially U will be the set of all isolated vertices along with $N(N(v_0))$.

- V is the set of all other unmatched vertices not in $T \cup U$. Initially $V = \mathbf{X} \setminus T \setminus U$.

Notation 27. *For any vertex v, we will write $N_2(v) = N(N(v)) - v$.*

So, $T \cup U \cup V$ is the set of currently unmatched vertices. As long as $T \neq \emptyset$ we are going to select $v \in T$ and match it with a randomly chosen unmatched vertex $\overline{v}$. Then $N(\overline{v})$ must be true so it goes to T, and $N_2(\overline{v})$ will be declared "free". This is the precise algorithm we will follow.

1. Start with $i = 0$ and initial sets T_0, U_0, V_0 described above.

2. While $T_i \neq \emptyset$ and $i \leq \alpha(n)\sqrt{n}$:
   Pick any vertex $v_i \in T_i$ and match it with a random vertex $\overline{v}_i \in T_i \cup U_i \cup V_i - v_i$.
   Then update T, U, V as follows:

   - If $\overline{v}_i \in T_i$ then STOP, we have our contradiction.
   - If $\overline{v}_i \in U_i$ then $T_{i+1} = T_i - v_i$, $U_{i+1} = U_i - \overline{v}_i \cup N(\overline{v}_i)$.
   - If $\overline{v}_i \in V_i$ then $T_{i+1} = T_i \cup N(\overline{v}_i) - v_i$, $U_{i+1} = U_i \cup N_2(\overline{v}_i)$, $V_{i+1} = V_i \setminus N_2(\overline{v}_i) \setminus N(\overline{v}_i) - \overline{v}_i$.
   - $i = i + 1$.

3. STOP (Note that either $T_i = \emptyset$ or $i \geq \alpha(n)\sqrt{n}$.)

Note that in this algorithm the graph we work with at step i is the graph induced by $U_i \cup V_i$.

We note some bounds on $|U_i|$ and $|V_i|$ in the course of the algorithm. For any vertex u, we have $N_2(u) \leq 2\Delta$ from (a.). Therefore, $|U_{i+1}| - |U_i| \leq 2\Delta$ for all i, and since $i \leq \alpha(n)\sqrt{n}$ through our process, we have

$$|U_i| \;\leq\; 2i\Delta + |U_0| \;\leq\; \frac{2n^{3/4}}{\alpha(n)} + \frac{n}{\alpha(n)^2} \;=\; O\left(\frac{n}{\alpha^2(n)}\right),$$

Similarly, $|V_i| - |V_{i+1}| \leq 3\Delta$ for all i and $|V_0| \geq 2n - 2\Delta$, therefore for all i

$$|V_i| \;\geq\; 2n - 3(i+1)\Delta \;\geq\; 2n - o(n^{3/4}).$$

Now we look at $|T_i|$. We have $|T_{i+1}| < |T_i|$ only if $\overline{v}_i \in U$, and in this case $|T_{i+1}| = |T_i| - 1$. We have

$$\Pr\left(|T_{i+1}| < |T_i|\right) \;\leq\; \frac{|U_i|}{|T_i \cup V_i \cup U_i| - 1} \;\leq\; \frac{O\left(\frac{n}{\alpha^2(n)}\right)}{2n - o(n^{3/4})} \;=\; O\left(\frac{1}{\alpha^2(n)}\right).$$

Now, if $\overline{v}_i \in V_i$ then $|T_{i+1}| - |T_i| = |N(\overline{v}_i)| - 1$, therefore we increase $|T_i|$ if $\deg(\overline{v}_i) > 1$. Define

$$p_L \;=\; \max_i \Pr(\deg(\overline{v}_i) = 1 \mid \overline{v}_i \in V_i).$$

The number degree 1 vertices in V never increases through the process because any vertex which loses an edge is immediately "free", therefore if a vertex of degree 1 is created it would move from V to U. Thus, we have

$$p_L \;\leq\; \frac{d_1}{\min_i |V_i|} \;\leq\; \frac{2n(1-\mu(n))}{2n - o(n^{3/4})} \;=\; 1 - \mu(n) + o(n^{-1/4}).$$

So,

$$\Pr(|T_{i+1}| - |T_i| \geq 1) \;\geq\; (1-p_L)\frac{|V_i|}{|T_i \cup V_i \cup U_i|} \;\geq\; \left[\mu(n) - o(n^{-1/4})\right]\frac{2n - o(n^{3/4})}{2n}$$
$$\geq\; \mu(n) - o(n^{-1/4}).$$

Lemma 28. *With high probability*

(i.) $|T_i| \neq 0$ *for all* $i \leq \alpha(n)\sqrt{n}$.

(ii.) If $j = \lfloor \sqrt{n}\alpha(n) \rfloor$ *then* $|T_j| \geq \frac{\mu(n)}{2} j$.

We prove this below, for now assume it is true. So, **whp** our algorithm will end either with $\overline{v}_i \in T_i$ or with $i > \alpha(n)\sqrt{n}$, not with $T_i = \emptyset$. If it ends with $\overline{v}_i \in T_i$ we are done, if not then Lemma 28 implies that **whp** we will finish with $|T| \geq \frac{\mu(n)\alpha(n)}{2}\sqrt{n}$. In this case it is extremely likely that a matching edge will occur within T, the probability of no such edge can be bounded above by

$$\prod_{i=1}^{|T|}\left(1 - \frac{|T|-i}{2n}\right) \leq \exp\left(-\frac{1}{2n}\sum_{i=1}^{|T|-1}|T|-i\right) = \exp\left(-\Omega\left(\frac{|T|^2}{n}\right)\right)$$
$$\leq \exp\left(-\Omega(\mu(n)^2\alpha(n)^2)\right) = o(1).$$

Thus, from Lemma 28 we can say that **whp** we will have a matching edge within T, therefore we have our contradiction.

We have

$$\begin{aligned}\Pr(S(G) \text{ satisfiable}) \leq\ & \Pr(\exists \text{ satisfying assignment with } v_0 \text{ false}) \\ +\ & \Pr(\exists \text{ satisfying assignment with } \overline{v}_0 \text{ false}),\end{aligned}$$

therefore

$$\begin{aligned}\Pr(S(G) \text{ satisfiable}) \leq\ & \Pr(\exists \text{ satisfying assignment with } v_0 \text{ false}) \\ +\ & \Pr(\overline{v}_0 \text{ is isolated}) \\ +\ & \Pr(\exists \text{ satis. assignment with } \overline{v}_0 \text{ false and not isol.}).\end{aligned}$$

The first and third summands on the right-hand side are $o(1)$ because of our contradiction, and the second is $\frac{1}{n}o(n) = o(1)$ because there are only $o(n)$ isolated vertices in G. Thus, $\Pr(S(G) \text{ is satisfiable}) = o(1)$.

□

Proof of Lemma 28 We first note that $\{|T_i|\}_{i\geq 0}$ can be thought of a series of random variables whose differences aren't quite independent, but clearly there is a series $\{X_i\}_{i\geq 0}$ of random variables such that $X_{i+1} - X_i$ are independent for all $i \geq 0$, and the following are all true:

1. $\{|T_i|\}_{i\geq 0}$ majorizes $\{X_i\}_{i\geq 0}$, i.e. $X_i \leq |T_i|$ for all $i \geq 0$.

2. $X_0 = |T_0| \geq 1$ because we chose a non-isolated vertex to start.

3. $\Delta \geq X_{i+1} - X_i \geq -1$ for all $i \geq 0$.

4. $\Pr(X_{i+1} < X_i) = O\left(\frac{1}{\alpha^2(n)}\right)$.

5. $\Pr(X_{i+1} \geq X_i + 1) = (1 - o(1))\mu(n)$.

Let P_1 be the probability that $X_i = 0$ for some $i \leq \alpha(n)\sqrt{n}$. Furthermore, define $p_< = \Pr(X_1 < X_0)$ and $p_> = \Pr(X_1 > X_0)$. A simple recursion gives us

$$P_1 \;\leq\; p_< + (1 - p_< - p_>)P_1 + p_> P_1^2,$$

which leads to

$$0 \;\leq\; (p_< - p_> P_1)(1 - P_1).$$

Certainly $P_1 < 1$, therefore

$$P_1 \;\leq\; \frac{p_<}{p_>} \;\leq\; \frac{O\left(\frac{1}{\alpha^2(n)}\right)}{(1 - o(1))\mu(n)} \;=\; O\left(\frac{1}{\alpha(n)^2\mu(n)}\right) \;=\; o(1).$$

Now, define P_2 as the probability that (i.) is true and (ii.) is false. Since

$$E\left[X_{i+1} - X_i\right] \;\geq\; (1 - o(1))\mu(n) - O\left(\frac{1}{\alpha(n)^2}\right) \;=\; (1 - o(1))\mu(n)$$

for all $i \leq j$, we have $E[X_j] \geq (1 - o(1))\mu(n)j$. So,

$$P_2 \;\leq\; \Pr\left(E[X_j] - |X_j| \geq \tfrac{\mu(n)}{3}j\right) \;=\; \Pr\left(|X_j| - E[X_j] \leq -\tfrac{\mu(n)}{3}j\right).$$

Condition (3.) above allows us to use (3.4):

$$P_2 \;\leq\; \exp\left(-\frac{\mu(n)^2 j}{72\Delta^2}\right) \;=\; \exp\left(-\Omega(\alpha(n)^5\mu(n)^2)\right) \;=\; o(1).$$

Since $P_1 + P_2 = o(1)$, we know that (i.) and (ii.) are true **whp**. $\square$

5.2 Proof of Theorem 13

Suppose that G is a graph with $2n$ vertices and $\Delta(G) = o(n^{1/8})$. Also, assume there is some $\epsilon > 0$ and some function $\tau \leq c \log n$ for some constant $c < \frac{3\epsilon}{16}$ such that

$$\sum_{i=0}^{\tau} id_i = (1+\epsilon)2n.$$

Notation 29. *For any number x, we will write $x^+ = x + o(1)$.*

Let δ be some small positive function satisfying

$$\exp\left(-\tau[\tfrac{2^+}{\epsilon} + \phi]\right) > \ \delta \ > \ n^{-3/8+\phi}$$

for some $\phi > 0$, a fixed constant, we know such a δ exists because of our assumption on τ.

If v is an isolated vertex in G, then any optimal assignment algorithm can set v to be false and $\overline{v}$ to be true. This defines a procedure which is commonly called **pure literal elimination**. We are going to do pure literal elimination on G and show that **whp** it leads to a graph which is not satisfiable **whp** by Theorem 26.

Notation 30. *We will write d_i as a function of s, since it will change throughout the process.*

(a.) Set $s = 0$.

(b.) While $d_0(s) > 0$ and $s < (1-\delta)n$:

Step s: Choose any isolated vertex v, and then randomly choose its match $\overline{v}$ from all other vertices. Make v false and $\overline{v}$ true, then delete both vertices from the graph, along with any edges incident with $\overline{v}$.

Increment s by 1.

First, we will show that the ratio between the number of edges and the number of vertices is likely not to decrease too much. Define

$$D_s^T := \sum_{i=0}^{T} id_i(s)$$

for any integer $T \leq \tau$, and

$$V_s := \sum_{i \geq 0} d_i(s) - 1.$$

Note that at any time $V_s = 2n - 2s - 1$. This will be the size of the "pool" of vertices that we have to choose from for $\overline{v}$.

Furthermore, define s_1 to be the step when the above process stops.

Notation 31. *Let $\overline{d}(s) = \{d_0(s), d_1(s), d_2(s), \dots\}$ be the entire degree sequence at step s.*

Lemma 32. *For any $i \geq 0$ and $s < s_1$, we have*

$$V_s E[d_i(s+1) - d_i(s) \mid \overline{d}(s)] = (i+1)\left(d_{i+1}(s) - d_i(s)\right) - V_s \mathbf{1}_{i=0}.$$

Lemma 33. *With high probability, for all $s < (1-\delta)n$ and $s < s_1$, we have*

$$\sum_{i=2}^{\tau} d_i(s) \quad \geq \quad \frac{\epsilon V_s}{1^+\tau}. \tag{5.3}$$

Let s_2 denote the first step s in which (5.3) does not hold, if such a step exists, and let $\overline{s} = \min\{s_1, s_2\}$ (If s_2 does not exist then $\overline{s} = s_1$.). We will continue our process beyond $s = \overline{s}$ for the sake of defining a martingale, but the graph (and hence the degree sequence) will not change after this point.

Lemma 34. *With high probability, $\overline{s} < (1-\delta)n$.*

Lemmas 33 and 34 show that **whp**, either we will stop because (5.3) does not hold or there is some number $\overline{s} < (1-\delta)n$ such that $\overline{s}$ steps of pure literal elimination will lead to a graph with $V_{\overline{s}} \geq 2\delta n$ vertices, $d_0 = 0$, $\Delta = o(n^{1/8})$, and $\sum_{i \geq 2} d_i \geq \Omega(\frac{V_{\overline{s}}}{\tau})$. Theorem 26 shows this is not satisfiable **whp** whenever n is sufficiently large with respect to $\delta = \delta(T, \epsilon)$ and $\delta \geq n^{-1/2}$. It remains only to prove the Lemmas.

Proof of Lemma 32 First, fix any $i \geq 1$. To make notation easier, let S_i, S_{i+1} be the set of all vertices of degree $i, i+1$, respectively, and let w_i, w_{i+1} be arbitrary vertices in their respective sets. We have

$$E[d_i(s+1) - d_i(s) \mid \overline{d}(s)] = E[|S_i(s+1) \setminus S_i(s)| \mid \overline{d}(s)] - E[|S_i(s) \setminus S_i(s+1)| \mid \overline{d}(s)].$$

Choose an arbitrary $w_i \in S_i(s)$. We have

$$\Pr(w_i \in S_i(s) \setminus S_i(s+1)) \quad = \quad \Pr(\overline{v} = w_i \text{ or } \overline{v} \in N(w_i)) \quad = \quad \frac{i+1}{V_s}$$

Thus,

$$E[S_i(s) \setminus S_i(s+1) \mid \overline{d}(s)] \quad = \quad |S_i| \left(\frac{i+1}{V_s}\right) \quad = \quad d_i \left(\frac{i+1}{V_s}\right)$$

Now, the only way a vertex is in $S_i(s+1) \setminus S_i(s)$ is if it had degree $i+1$ and it lost a neighbor. Therefore,

$$\Pr(w_{i+1} \in S_i(s+1) \setminus S_i(s)) \quad = \quad \Pr(\overline{v} \in N(w_{i+1})) \quad = \quad \frac{|N(w_{i+1})|}{V_s} \quad = \quad \frac{i+1}{V_s}$$

Thus,

$$E[S_i(s+1) \setminus S_i(s) \mid \overline{d}(s)] \quad = \quad |S_{i+1}| \left(\frac{i+1}{V_s}\right) \quad = \quad d_{i+1} \left(\frac{i+1}{V_s}\right)$$

When $i = 0$, the only difference is that $E[S_0(s+1) \setminus S_0(s)]$ is one less because pure literal elimination randomly matches a degree 0 vertex.

□

Proof of Lemma 33 We will examine the series of variables $\{\frac{D_i^\tau}{V_i}\}_{i \geq 0}$. First to bound the expected change. Lemma 32 gives

$$V_s E[D_{s+1}^\tau - D_s^\tau \mid \overline{d}(s)] \quad = \quad \sum_{i=1}^{\tau} i(i+1) \left(d_{i+1}(s) - d_i(s)\right),$$

for all s, therefore

$$V_s E[D_{s+1}^\tau - D_s^\tau \mid \overline{d}(s)] \quad = \quad -2 \sum_{i=1}^{\tau} i d_i(s) + \tau(\tau+1) d_{\tau+1}(s) \quad \geq \quad -2 D_s^\tau$$

because $d_{\tau+1} \geq 0$. Since V_s is known and $V_{s+1} = V_s - 2$, we use this to get

$$\begin{aligned} E\left[\frac{D^\tau_{s+1}}{V_{s+1}} - \frac{D^\tau_s}{V_s} \mid \overline{d}(s)\right] &= \frac{E[D^\tau_{s+1}V_s - D^\tau_s(V_s-2) \mid \overline{d}(s)]}{V_s(V_s-2)} \\ &= \frac{V_s E[D^\tau_{s+1} - D^\tau_s \mid \overline{d}(s)] + 2D^\tau_s}{V_s(V_s-2)} \geq 0, \end{aligned}$$

for all s during our process. So, for all s we have

$$E\left[\frac{D^\tau_s}{V_s}\right] \geq \frac{D^\tau_0}{V_0} = 1+\epsilon. \tag{5.4}$$

Now to bound the actual difference. Each step deletes at most one non-isolated vertex, therefore $|D^\tau_{s+1} - D^\tau_s| \leq 2\Delta$. Furthermore, $D^\tau_s \leq V_s\Delta$.

$$\left|\frac{D^\tau_{s+1}}{V_{s+1}} - \frac{D^\tau_s}{V_s}\right| \leq \frac{\left|D^\tau_{s+1} - D^\tau_s\right|}{V_s-2} + \frac{2D^\tau_s}{V_s(V_s-2)} \leq \frac{2\Delta + \frac{D^\tau_s}{V_s}}{V_s-2} \leq \frac{3\Delta}{V_s-2}$$

for all s.

Define $\beta(n) = n^{-\phi/2}$ to be a small positive function, and fix any $s < (1-\delta)n$. We use the above with (5.4), (3.2), $s < n$, and $V_s \geq 2\delta n$ to get

$$\begin{aligned} \Pr\left(\frac{D^\tau_s}{V_s} \leq (1+\epsilon) - \beta(n)\right) &\leq \exp\left(-\frac{1}{2s}\left[\frac{\beta(n)V_s}{3\Delta}\right]^2\right) \leq \exp\left(-\frac{1}{2n}\left[\frac{n^{-\phi/2}2\delta n}{3n^{1/8}}\right]^2\right) \\ &= \exp\left(-\tfrac{2}{9}n^{3/4-\phi}\delta^2\right) < \exp\left(-\tfrac{2}{9}n^{\phi}\right). \end{aligned}$$

So, the probability that this is true for any $s < (1-\delta)n$ can be bounded from above by $n\exp(-\frac{2}{9}n^\phi) = o(1)$. Therefore, **whp** we have

$$0 \leq D^\tau_s - (1+\epsilon)V_s + \beta(n)V_s = D^\tau_s - V_s - (1-o(1))\epsilon V_s.$$

whp for all $s < (1-\delta)n$, this yields

$$\tau\sum_{i=2}^{\tau} d_i(s) \geq \sum_{i=2}^{\tau}(i-1)d_i(s) \geq D^\tau_s - V_s \geq (1-o(1))\epsilon V_s = \frac{\epsilon V_s}{1^+}.$$

□

Proof of Lemma 34 We will examine the random variables $\{\frac{d_0(i)}{V_i}\}_{i\geq 0}$. First, we bound the difference for all s. Here we use $V_{s+1} = V_s - 2$ and $|d_0(s+1) - d_0(s)| \leq \Delta$.

$$\left|\frac{d_0(s+1)}{V_{s+1}} - \frac{d_0(s)}{V_s}\right| \quad = \quad \left|\frac{d_0(s+1) - d_0(s)}{V_s - 2} + \frac{2d_0(s)}{V_s(V_s-2)}\right| \quad = \quad O\left(\frac{\Delta}{V_s}\right).$$

Now we look at the expected change. First using Lemma 32:

$$E\left[\frac{d_0(s+1)}{V_{s+1}} - \frac{d_0(s)}{V_s} \mid \overline{d}(s)\right] \quad = \quad \frac{d_1(s) + d_0(s) - V_s}{V_s(V_s-2)} \quad \leq \quad \frac{-\sum_{i=2}^{\tau} d_i(s)}{V_s(V_s-2)}.$$

Now we can use Lemma 33 and $V_s = 2(n-s)-1$ (for $s < \overline{s}$) to say that **whp** either $\overline{s} < s$ or

$$E\left[\frac{d_0(s+1)}{V_{s+1}} - \frac{d_0(s)}{V_s} \mid \overline{d}(s)\right] \quad \leq \quad -\frac{\epsilon}{2^+\tau(n-s)}$$

holds for all $s < (1-\delta)n$. Although the differences in $\{\frac{d_0(i)}{V_i}\}_{i\geq 0}$ are not independent, and the process stops if $d_0(s) = 0$, the "2^+" function clearly leaves room for a series of random variables $\{X_i\}_{i\geq 0}$ such that the differences $X_{i+1} - X_i$ are independent for all $i \geq 0$, and the following are true **whp** for all $s \in [0, (1-\delta)n]$:

1. Either $d_0(s) = 0$ or $s_2 < \overline{s} < s_1$ or $X_s \geq \frac{d_0(s)}{V_s}$.
2. $X_0 = \frac{d_0(0)}{V_0} \leq 1$.
3. $|X_{s+1} - X_s| = O(\frac{\Delta}{V_s})$.
4. $E[X_{i+1} - X_i] \leq -\frac{\epsilon}{2^+\tau(n-s)}$.

So,

$$E\left[X_s\right] \quad \leq \quad X_0 - \frac{\epsilon}{2^+\tau}\sum_{r=1}^{s}\frac{1}{n-r} \quad \leq \quad 1 + \frac{\epsilon}{2^+\tau}\log\left(1 - \frac{s}{n}\right)$$

for all s provided n is sufficiently large. So, if $\delta < \exp(-\frac{2^+\tau}{\epsilon} - \phi\tau)$, we will have $\overline{s} < (1-\delta)n$ satisfying $E[X_s] < -\frac{\epsilon\phi}{2^+}$, a constant. This allows us to use (3.3), so for any function $\alpha(n) \to \infty$,

$$\Pr(X_{\overline{s}} > 0) \quad \leq \quad \exp\left[-\Omega\left(\frac{V_{\overline{s}}}{\Delta\sqrt{\overline{s}}}\right)^2\right] \quad \leq \quad \exp\left[-\Omega\left(\frac{\delta n}{n^{1/8}\sqrt{n}}\right)^2\right]$$

$$\leq \quad \exp\left(-\Omega(n^{2\phi})\right) \quad = \quad o(1).$$

Therefore, **whp** we have $s_1 < (1-\delta)n$ or $s_2 < (1-\delta)n$. In either case we have $\bar{s} < (1-\delta)n$.

5.3 Proof of Theorem 11

Suppose that G is any graph with $2n$ vertices and $\epsilon > 0$ satisfies the following:

1. G has less than $(1-\epsilon)n$ edges.
2. $\Delta(G) \leq \frac{n^{1/10}}{\alpha(n)}$ where $\alpha(n) = \frac{5}{\epsilon^2}\log n$.

To make notation easier, we will define

$$i_* := \left\lfloor \Delta^2 \alpha(n) \right\rfloor$$

First we choose any vertex $v_0 \in G$ and set it false, a set T will give rise to a process similar to section 5.1. However, now that the expected degree is less than 1, we will show that **whp** there will be no contradiction, we will most likely finish with $T = \emptyset$ instead of an edge within T or $i > i_*$.

Here is the exact procedure we will follow. Since we only need an upper bound on $|T|$, there is no need to keep track of a set U like in section 5.1.

1. Choose any vertex v_0, set $i = 0$, $T_0 = N(v_0)$, $V = \mathbf{X} \setminus N(v_0) \setminus v_0$.
2. While $T_i \neq \emptyset$ and $i \leq i_*$:

   Pick any vertex $v_i \in T_i$ and match it with a random vertex $\overline{v}_i \in T_i \cup V_i - v_i$.

   - If $\overline{v}_i \in T_i$ then STOP, we have a contradiction.
   - If $\overline{v}_i \in V_i$ then $T_{i+1} = T_i \cup N(\overline{v}_i) - v_i$, $V_{i+1} = V_i \setminus N(\overline{v}_i) - \overline{v}_i$.
   - $i = i + 1$.
3. STOP, either $T_i = \emptyset$ or $i \geq i_*$.

The only thing that can raise the expected degree of $\overline{v}_i$ above $1-\epsilon$ is deleting isolated vertices, as deletion of any other vertices will also delete edges. However, we have

$$|V_i| \;>\; 2n-(i_*+1)\Delta-(\Delta+1) \quad=\quad 2n-O(i_*\Delta) \quad\leq\quad 2n-o(n).$$

Since we start with at least ϵn isolated vertices and won't lose more than $o(n)$ of them, we know that the increase in expected degree must be small, namely

$$E[|N(\overline{v}_i)|] \leq 1-\epsilon+o(1)$$

for all $i \geq 0$. So, we bound $E[|T_i|]$ with the following:

$$E[|T_i| \mid T_{i-1}] \quad=\quad |T_{i-1}|-1+[1-\epsilon+o(1)] \quad=\quad |T_{i-1}|-\epsilon+o(1). \tag{5.5}$$

Much like the proof of Lemma 28, we take the random variables $\{|T_i|\}_{i\geq 0}$, and note that the $o(1)$ term in (5.5) clearly leads to a series random variables $\{X_i\}_{i\geq 0}$ such that for all i in our process we have $X_i \geq |T_i|$, $|X_{i+1}-X_i| \leq \Delta$, and all differences $X_{i+1}-X_i$ are independent. Furthermore, the X_i variables can "continue" even after $T_i = \emptyset$ and our process stops, so we have

$$E[X_i] \leq -\epsilon i+\Delta+o(i) \text{ for all } i \leq i_*. \tag{5.6}$$

For any vertex $v \in V(G)$, we have defined a process which begins by setting v false and continues keeping track of set T (as defined in the proof of Theorem 26) until either $T=\emptyset$, $\overline{v}_i \in T$, or $i = i_*$. Let E_v be the event that this process does not end with $T=\emptyset$, and define Z_v to be the set of all vertices which appear in the corresponding T at any time.

Lemma 35. *For any $v \in V(G)$, $\Pr(E_v) = O(n^{-3/5})$.*

Lemma 36. *If u is fixed and $\overline{u}$ is chosen randomly from $V(G)$, then*

$$\Pr(E_u \wedge E_{\overline{u}}) \quad=\quad o(\tfrac{1}{n}).$$

Lemmas 35 and 36 are proven below.

Consider an instance not in the union

$$\bigcup_u E_u \wedge E_{\overline{u}}.$$

By Lemma 36, the probability of such an instance is $1 - o(\frac{1}{n})O(n) = 1 - o(1)$ by the union bound. Therefore this deterministic entity has a satisfying assignment. (We can iteratively choose a pair of vertices $u, \overline{u}$ and set one of them false because this instance is not in $E_u \wedge E_{\overline{u}}$.) So, we are done once we prove Lemmas 35 and 36.

Proof of Lemma 36 Assume Z_u is fixed. When we choose $\overline{v}$ (the partner of v) we need $\overline{v} \notin Z_u$ and $N(\overline{v}) \cap Z_u = \emptyset$. The probability of a problem is bounded above by

$$\frac{(\Delta+1)|Z_u|}{n} \quad = \quad O\left(\frac{\Delta^2 i_*}{n}\right).$$

for any randomly chosen $\overline{u} \in V(G)$, whether E_u is true or not. We make i_* choices in the formation of $Z_{\overline{u}}$, so the probability of a problem is bounded from above by

$$O\left(\frac{\Delta^2 i_*^2}{n}\right) \quad = \quad O\left(\frac{\Delta^6 \alpha(n)^2}{n}\right) \quad = \quad O\left(\frac{n^{-2/5}}{\alpha(n)^4}\right) \quad = \quad o(n^{-2/5})$$

Therefore,

$$\Pr(Z_u \cap Z_{\overline{u}} \neq \emptyset | E_u) = o(n^{-2/5}) \tag{5.7}$$

Define $A = A_{u,\overline{u}}$ to be the event that $Z_u \cap Z_{\overline{u}} = \emptyset$. We have

$$\begin{aligned}\Pr(E_u \wedge E_{\overline{u}}) \quad &= \quad \Pr(E_u)\left[\Pr(E_{\overline{u}}|A, E_u)\Pr(A|E_u) + \Pr(E_{\overline{u}}|\overline{A}, E_u)\Pr(\overline{A}|E_u))\right] \\ &\leq \quad \Pr(E_u)\left[\Pr(E_{\overline{u}}|A, E_u) + \Pr(\overline{A}|E_u)\right].\end{aligned}$$

For the second term, note that being given A, E_u ensures that the process starting at $\overline{u}$ avoids Z_u at all times. Therefore, the exact same proof of Lemma 35 with $G - Z_u$ in place of G tells us that $\Pr(E_{\overline{u}}|A, E_u) = O(n^{-3/5})$. So, using Lemma 35 and (5.7) we see that

$$\Pr(E_u \wedge E_{\overline{u}}) \quad \leq \quad O(n^{-3/5})\left[O(n^{-3/5}) + o(n^{-2/5})\right] \quad = \quad o(\tfrac{1}{n}).$$

□

Proof of Lemma 35 We will prove that all of the following are true with probability $1 - O(n^{-3/5})$:

(a.) $|T_i| = 0$ for some $i \leq i_*$.

(b.) $|T_i| \leq 2i_*\alpha(n)$ for all $i \leq i_*$.

(c.) No edges will occur within T.

We have $i_* >> \Delta$, so (5.6) tells us that $E[X_{i_*}] \leq -\frac{\epsilon}{1^+} i_*$. We use (3.3) with this and the fact that $|X_{i+1} - X_i| \leq \Delta$ for all $i \geq 0$:

$$\begin{aligned} \Pr(\text{(a.) false}) &\leq \Pr(X_{i_*} > 0) \leq \exp\left(-\frac{\epsilon^2 i_*}{8^+\Delta^2}\right) \leq \exp\left(-\frac{\epsilon^2\alpha(n)}{8^+}\right) \\ &= \exp\left(-\frac{\epsilon^2}{8^+}\frac{5}{\epsilon^2}\log n\right) = n^{-5/8^+} \leq n^{-3/5}. \end{aligned}$$

For (b.), it is easy to see that

$$X_i > 2i_*\alpha(n) \Rightarrow X_i - E[X_i] \geq i_*\alpha(n),$$

because $E[X_i] \leq \Delta + o(1) << i_*\alpha(n)$. So, by (3.3):

$$\Pr\left(X_i > 2i_*\alpha(n)\right) \leq \exp\left(-\frac{(i_*\alpha(n))^2}{8\Delta^2 i}\right) \leq \exp\left(-\tfrac{1}{8}\alpha(n)^3\right) = o(\tfrac{1}{n})$$

for all $i \leq i_*$, therefore the probability of this happening for any $i \leq i_*$ is actually $o(n^{-4/5})$. Finally, if (b.) is true then we have for all $i \leq i_*$

$$\Pr(\overline{v}_i \in T_i) = \frac{|T_i|}{|T_i| + |V_i| - 1} = \frac{|T_i|}{2n - o(n)} < \frac{X_i}{n} < \frac{2i_*\alpha(n)}{n}.$$

Therefore, the probability that (b.) is true and (c.) is false is bounded by

$$\sum_{i=0}^{i_*} \Pr(\overline{v}_i \in T_i) \leq \left(\frac{2i_*\alpha(n)}{n}\right) i_* = O\left(\frac{\Delta^4\alpha(n)^3}{n}\right) = O\left(\frac{n^{-3/5}}{\alpha(n)}\right).$$

□

5.4 Why The Maximum Degree Condition is Needed

If the maximum degree is large, then the satisfiability depends much more on where the large degree vertices are matched and less on the actual graph. One example of this is a graph G which is the union of $K_{\alpha\sqrt{n}}$ and $2n - \alpha\sqrt{n}$ isolated vertices. Note that $S(G)$ is **not** satisfiable if and only if two or more of the matching edges end up within the complete graph $K_{\alpha\sqrt{n}}$. So,

$\Pr(S(G)$ **is** satisfiable)

$$= \prod_{i=0}^{\alpha\sqrt{n}-1} \frac{2n - \alpha\sqrt{n} - i}{2n - 1 - 2i} + \binom{\alpha\sqrt{n}}{2} \frac{1}{2n-1} \prod_{i=2}^{\alpha\sqrt{n}-1} \frac{2n - \alpha\sqrt{n} - (i-2)}{2n - 1 - 2i}$$

$$\approx \left(1 + \frac{\alpha^2}{4}\right) \prod_{i=2}^{\alpha\sqrt{n}-1} \frac{2n - \alpha\sqrt{n} - (i-2)}{2n - 1 - 2i}.$$

By taking the logarithm of the product and using $\log(1-x) \approx -x$ for $x \approx 0$, we can approximate the value of the product, and we arrive at the following:

$$\Pr(S(G) \text{ is satisfiable}) \approx \left(1 + \frac{\alpha^2}{4}\right) \exp\left(-\frac{\alpha^2}{4}\right)$$

Thus, G has about $\alpha^2 n$ edges, but the probability of satisfiability of $S(G)$ does not have a threshold, it is a smooth function of α.

5.5 Concerning Conjecture 14

Here we present evidence which leads us to believe that Conjecture 14 should be true.

5.5.1 Two Examples

Here we present two vastly different graphs G_1, G_2 with $(1+\epsilon)n$ edges but which violate (5.1), and both $S(G_1)$ and $S(G_2)$ are not satisfiable **whp**.

Graph G_1: Fix $\log n << \alpha(n) \leq n^{\phi}$. Let G_α be any $\alpha(n)$-regular graph with $2(1+\epsilon)\frac{n}{\alpha}$ vertices. Let G_1 be G_α plus $2n - |V_{G_\alpha}|$ isolated vertices. We give the

following "informal" argument to show that $S(G_1)$ is satisfiable **whp**:
Match all vertices which start out isolated, those which are matched may be "deleted" because they are no longer relevant. We will be left with an induced subgraph of G_α, say G'_α, where $v \in V(G_\alpha)$ exists in G'_α with probability $\frac{|V(G_\alpha)|}{2n-1} \approx \frac{1+\epsilon}{\alpha}$. Also, $e \in E(G_\alpha)$ makes it to G'_α only if both of its vertices survive, which happens with probability close to $\left(\frac{1+\epsilon}{\alpha}\right)^2$. So, **whp** G'_α has about $2(1+\epsilon)^2\frac{n}{\alpha^2}$ vertices and **whp**

$$\frac{|E(G'_\alpha)|}{|V(G'_\alpha)|} \approx \frac{\left(\frac{1+\epsilon}{\alpha}\right)^2 |E_{G_\alpha}|}{\left(\frac{1+\epsilon}{\alpha}\right)|V_{G_\alpha}|} = \left(\frac{1+\epsilon}{\alpha}\right)\frac{\alpha}{2} = \frac{1+\epsilon}{2}.$$

Also, we can most likely say a lot more about the degrees of the vertices. It is extremely unlikely that G_α has many high-degree vertices, in fact **whp** G'_α satisfies (5.1) with τ equal to some sufficiently large constant, therefore G_1 is not satisfiable **whp** by Theorem 13.

Graph G_2: Again fix $\log n << \alpha(n) \leq n^\phi$, and assume that $\phi < \frac{1}{4}$. Take $(1+\epsilon)\frac{n}{\alpha}$ disjoint stars, each with α leaves, then add $(1-\epsilon)n - (1+\epsilon)\frac{n}{\alpha}$ isolated vertices to make G_2. We can use a procedure similar to that of Section 5.1, starting at any non-isolated vertex and stopping if $i \geq \sqrt{n}$. With stars we know exactly what we are working with, for any $\overline{v}_i$ we have a clearly defined $N(\overline{v}_i)$, $N_2(\overline{v}_i)$, and we know that declaring $N_2(v_i)$ "free" doesn't assume anything, leaves whose parent is deleted are indeed isolated. It is easy to see that for all $i \leq \alpha^3$ (since each step involves moving at most $\alpha+1$ vertices) we have

$$|U_i| \leq (1-\epsilon)\,n + o(n) \text{ and } |V_i| \geq (1+\epsilon)\,n - o(n).$$

If $\overline{v}_i \in T_i$ for any i we are done. Otherwise, $|T_i|$ behaves as follows:

$$|T_{i+1}| - |T_i| = \begin{cases} -1 & \text{prob. } \frac{1-\epsilon}{2} - o(1) \\ +\alpha - 1 & \text{prob. } \frac{1}{\alpha}(\frac{1+\epsilon}{2} - o(1)) \\ 0 & \text{otherwise} \end{cases}$$

The first of the three cases above corresponds to when $\overline{v}_i \in U_i$, so the only change to T is v_i is removed. The second case corresponds to when $\overline{v}_i \in V_i$ and $\overline{v}_i$ is a star center, therefore v_i gets removed from T and α leaves get added. The third case is when $\overline{v}_i \in V_i$ is a leaf, therefore v_i is removed from T but the center of the respective star is added.

Regardless of our what our non-isolated starting vertex is, for some constant c we have $\Pr(|T_{\lfloor\sqrt{n}\log n\rfloor}| >> \sqrt{n}) \geq c$ because on every step the expected change in $|T|$ is a positive constant. Since **whp** $|T| >> \sqrt{n}$ forces an edge within T, **whp** we have unsatisfiability.

5.5.2 Starting With Bounded Degree

Suppose we run the pure literal algorithm on a graph with a bounded degree sequence whose degree sum exceeds its number of vertices. Here we show that as step s approaches n in the pure literal algorithm, the degrees fall exponentially. It seems likely that this should continue even if the degree is not bounded. If this is the case, then Conjecture 14 is true because we can begin by running the pure literal algorithm, then creating a graph which will meet the conditions Theorem 13.

During the pure literal algorithm, we started with $s = 0$ and we increased s until it was something close to n. If we let $t = \frac{s}{n}$ and $v_i(t) = \frac{1}{n}d_i(s)$ for all i, then we can look at this as a function of t, as t goes from 0 to 1. If the maximum degree starting out is a constant T, then we can use Lemma 32 along with methods discussed in Chapter 3 to create a system of differential equations, which **whp** is accurate within $O(n^{-1/2})$. Here is what the system looks like for $T = 4$, the pattern should be clear.

$$2(1-t)\begin{bmatrix} v_1'(t) \\ v_2'(t) \\ v_3'(t) \\ v_4'(t) \end{bmatrix} = \begin{bmatrix} -2 & 2 & 0 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 0 & -4 & 4 \\ 0 & 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix}. \tag{5.8}$$

This can be solved using the diagonalization $M\Lambda M^{-1}$ of the square matrix. In this case $\Lambda_{ii} = -(i+1)$ for all i, and M is an upper triangular matrix defined by

$$M_{i,j} \quad = \quad \begin{cases} (-1)^{i+j}\binom{j}{i} & i \leq j \\ 0 & \text{otherwise} \end{cases}.$$

As it turns out, $(M^{-1})_{ij} = |M_{ij}|$ for all i, j. The solution to this system is

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix} \quad = \quad M\text{diag}\left[M^{-1}\overline{d}(0)\right] \begin{bmatrix} (1-t) \\ (1-t)^{3/2} \\ (1-t)^2 \\ (1-t)^{5/2} \end{bmatrix}, \tag{5.9}$$

where $\text{diag}(w)$ for any vector w is the diagonal matrix W where $W_{ii} = w_i$ for all i. (Again the pattern should be clear for any T, not just $T = 4$.)

To see that this is indeed the solution, let $\mu(t)$ be the last vector on the right-hand side of (5.9). Note that $2(1-t)\mu'(t) = \Lambda\mu(t)$, where Λ is defined above. Using this, it is easy to see that (5.8) is satisfied.

So, if the largest degree is bounded to start, then so are the binomial coefficients, thus we have

$$v_i(t) \quad = \quad \Theta\left((1-t)^{(i+1)/2}\right)$$

for all i. This implies that for any N, there exists $\tau > 0$ such that by time $1-\tau$, **whp** we have

$$\frac{d_i(\tau)}{d_{i+1}(\tau)} \quad > \quad N.$$

Although it seems much more difficult to prove, we believe that this nice distribution will continue even if the starting degree is larger than a constant T. If this is the case, then Conjecture 14 is true.

Chapter 6

ONLINE SATISFIABILITY

Here we will prove the following:

Theorem 16. *Fix any integer $k \geq 1$, constant $c > 0$ and any online algorithm. Given a random formula with cn k-clauses,* **whp** *the algorithm accepts fewer than*

$$\left(1 - \frac{1}{2^k}\right) cn \; + \; \left(\frac{\ln 2}{-2^k \ln\left(1 - \frac{1}{2^k}\right)}\right) n$$

clauses.

Notation 37. *We will use J_k to denote an arbitrary k-tuple $(j_1, \ldots, j_k)$.*

To generate k-clauses, first we will select J_k randomly from $\{1, 2, \ldots, n\}$, with replacement, then we will choose one of the 2^k corresponding clauses, each with probability $\frac{1}{2^k}$. For example, if $k = 3$ and we choose $j_1 = 9, j_2 = 1, j_3 = 5$, then we would select one of the following 8 clauses:

$$\{x_9, x_1, x_5\}, \{\overline{x}_9, x_1, x_5\}, \{x_9, \overline{x}_1, x_5\}, \{x_9, x_1, \overline{x}_5\},$$

$$\{\overline{x}_9, \overline{x}_1, x_5\}, \{\overline{x}_9, x_1, \overline{x}_5\}, \{x_9, \overline{x}_1, \overline{x}_5\}, \{\overline{x}_9, \overline{x}_1, \overline{x}_5\},$$

each with probability $\frac{1}{8}$. We will also assume that $j_1, \ldots, j_k$ are all distinct, because as $n \to \infty$ the expected number of clauses where this is not the case is $o(n)$. Therefore, rejecting all of these clauses would not change Theorem 16.

A simple Argument: We first present this short proof which shows that **whp**, no more than $(1 - \frac{1}{2^k})cn + 2^{k+1}n$ out of cn random k-clauses can be accepted. Although

it is weaker than Theorem 16, it is still a huge improvement over the previously best-known bound of $(1-\frac{1}{2^k})cn + \Theta(\sqrt{c})n$, and a similar idea is used in Section 6.1 in our proof of Theorem 16.

Suppose an online algorithm accepts more than

$$(1-\tfrac{1}{2^k})cn + 2^{k+1}n \tag{6.1}$$

out of cn random k-clauses. Note that (6.1) requires c to be large (i.e. beyond the phase transition).

Notation 38. *After an online algorithm has seen exactly i clauses, let $\mathcal{A}_i$ be the set of accepted clauses, and define S_i to be the set of valid assignments.*

$|S_0| = 2^n$ because every assignment is valid before any clauses are accepted. Furthermore, $S_i \subseteq S_{i-1}$ for every i, with equality if clause i is rejected because this does not change the set of valid assignments. For a valid assignment to exist, $|S_i| \geq 1$ is necessary. Call a clause **bad** if accepting it would make

$$|S_i| \leq (1-\tfrac{1}{2^k})|S_{i-1}|.$$

For every i and every $j_1, \ldots, j_k$, at least one of the 2^k clauses is bad because the 2^k corresponding possibilities for $S_{i-1} \setminus S_i$ partition S_{i-1}. Therefore, for all i,

$$\Pr(\text{clause } i \text{ is bad}) \geq \tfrac{1}{2^k},$$

so from (3.3) with $z = 1$, $k = cn$, and $\sqrt{n} << \lambda << n^{0.51}$ we see that **whp** at least $\frac{1}{2^k}cn - o(n^{.51})$ out of cn clauses will be bad. Assuming (6.1) is so forces an overlap of size $(2^{k+1} - o(1))n$ between the sets of bad and accepted clauses, so

$$|S_{cn}| \;\leq\; |S_0|\left(1-\tfrac{1}{2^k}\right)^{(2^{k+1}-o(1))n} \;\leq\; 2^n\left[\mathbf{e}^{-2+o(1)}\right]^n \;<\; 1$$

whp, a contradiction.

6.1 Constant Improvement

Suppose a positive integer k and constant $c > 0$ are both fixed, and fix some online algorithm. Define $A_i = |\mathcal{A}_i|$ for all $i = 0, 1, 2, \ldots, cn$, so A_i is the number of clauses that the algorithm accepts out of the first i given. We will show that if all accepted clauses are satisfied, then **whp**

$$A_{cn} - \left(1 - \tfrac{1}{2^k}\right) cn \quad \leq \quad \left(\frac{\ln 2}{-2^k \ln\left(1 - \frac{1}{2^k}\right)}\right) n. \tag{6.2}$$

As discussed previously, clauses will be generated by first choosing the k-tuple $J_k = (j_1, j_2, \ldots, j_k)$ randomly from $\{1, 2, \ldots, n\}^k$, then by choosing one of the 2^k corresponding clauses, each with probability $\frac{1}{2^k}$.

Define a clause as a **bad clause** if choosing it would make $|S_{i+1}| \leq \mathbf{e}^{-2}|S_i|$, and define J_k as a **bad k-tuple** if one of the corresponding 2^k clauses is bad. (Note that no k-tuple can have more than one bad clause.) Furthermore, for $i = 1, 2, \ldots, cn$, define F_i as an indicator variable which takes value 1 if both of the following are true:

- Clause i corresponds to some bad k-tuple J_k.
- If clause i is the bad clause corresponding to J_k, then it will be accepted.

6.1.1 Proof Sketch

Any algorithm can "stop thinking" at any point by picking an assignment of the variables and taking only clauses that are satisfied by that assignment (which occurs with probability $1 - \frac{1}{2^k}$), so we would like to look at

$$B_i = A_i - (1 - \tfrac{1}{2^k})i,$$

the number of clauses taken "beyond" the number that come easily. For any J_k, there are two possibilities, either the algorithm will take all 2^k of its corresponding clauses, or it will turn down at least one. In the latter case, we will assume nothing

other than $|S_i| \leq |S_{i-1}|$, however in this case we also have that the expected change in B_i is at most 0 because the probability of rejection is at least $\frac{1}{2^k}$.

The other case is more interesting. Here we know for sure that $B_i = B_{i-1} + \frac{1}{2^k}$. However, we will show that the expected value of $\ln|S_i| - \ln|S_{i-1}|$ in this case is $\ln\left(1 - \frac{1}{2^k}\right)$ or less. So, each time the algorithm tries to increase B_i, it pays for it with a decrease in $\ln|S_i|$. In particular, we can define the following random variable:

$$Y_j := \ln|S_j| - 2^k \ln\left(1 - \tfrac{1}{2^k}\right) B_j.$$

Notation 39. *$\mathcal{F}_i$ is the filtration defined by the first i clauses that appear.*

Here $E[Y_i - Y_{i-1}|\mathcal{F}_{i-1}] \leq 0$ for all i, and therefore $Y_{cn} \leq Y_0 + o(n)$ *should* be true. Since $\ln|S_{cn}| \geq 0$ is necessary, this would give

$$-2^k \ln\left(1 - \tfrac{1}{2^k}\right) B_{cn} \leq Y_{cn} \leq Y_0 + o(n) \approx n \ln 2.$$

From here (6.2) is immediate. The only flaw in this argument comes because $Y_i - Y_{i-1}$ is unbounded from below, otherwise this would be a proof.

To get around this, we note that if $Y_i << Y_{i-1}$ then a bad clause was taken. So, we will introduce random variable $X_i = Y_i + \ln(1 - \frac{1}{2^k})(F_1 + \cdots + F_i)$. The more the algorithm "tries" to increase B_i, the more likely it is that $F_i = 1$, and in this case it pays a penalty which guarantees $X_i \leq X_{i-1} - 2$. If it makes no such attempt, then we still have $E[X_i - X_{i-1}|\mathcal{F}_{i-1}] \leq 0$, and we also have a bound on $|X_i - X_{i-1}|$. Putting the two cases together gives a probable lower bound on $X_0 - X_{cn}$ which is proportional to $F_1 + \cdots + F_{cn}$. We then show that **whp** this sum is either very small or proportional to B_{cn}, which allows us to bound B_{cn}.

6.1.2 Proof of Theorem 16

Let $G_i = F_1 + \cdots + F_i$ and define

$$X_i \quad := \quad \ln|S_i| - 2^k \ln\left(1 - \tfrac{1}{2^k}\right)\left[B_i - \tfrac{1}{2^k} G_i\right].$$

The following Lemmas are proven below:

Lemma 40. *Let J_k be a k-tuple such that $E[A_i - A_{i-1}|J_k, \mathcal{F}_{i-1}] = 1$. Then,*

$$E\left[\ln|S_i| - \ln|S_{i-1}| \mid J_k, \mathcal{F}_{i-1}\right] \quad \leq \quad \ln\left(1 - \tfrac{1}{2^k}\right).$$

Lemma 41. *For all $i = 1, 2, \ldots, cn$:*

1. *If $F_i = 1$ then $X_i \leq X_{i-1}$, and if clause i is bad (which occurs with probability at least $\frac{1}{2^k}$), then*

$$X_i \leq X_{i-1} - 2. \tag{6.3}$$

2. *If $F_i = 0$ then $|X_i - X_{i-1}| \leq 3.4$ and $E[X_i|F_i = 0, \mathcal{F}_{i-1}] \leq X_{i-1}$.*

Lemma 42. *Let $\lambda = 2 + 2c^{0.51}$. With high probability,*

$$X_{cn} \leq X_0 - \tfrac{2}{2^k}G_{cn} + \lambda n^{0.51}. \tag{6.4}$$

Because $X_0 = \ln|S_0| = n\ln 2$ and $\ln|S_{cn}| \geq 0$ is necessary for a valid assignment to exist, Lemma 42 shows that **whp**

$$-2^k \ln\left(1 - \tfrac{1}{2^k}\right)\left[B_{cn} - \tfrac{1}{2^k}G_{cn}\right] \quad \leq \quad n\ln 2 - \tfrac{2}{2^k}G_{cn} + o(n).$$

Since $\left[\ln\left(1 - \frac{1}{2^k}\right) + \frac{2}{2^k}\right] G_{cn} \geq 0$ for all $k \geq 1$, **whp**

$$-2^k \ln\left(1 - \tfrac{1}{2^k}\right) B_{cn} \quad \leq \quad n\ln 2 + o(n).$$

Dividing this by $-2^k \ln\left(1 - \frac{1}{2^k}\right)$ shows that (6.2) holds **whp**. It remains only to prove the Lemmas.

Proof of Lemma 40: Since $E[A_i - A_{i-1}|J_k, \mathcal{F}_{i-1}] = 1$, any of the 2^k possible clauses corresponding to J_k will be accepted by the algorithm. Therefore, they correspond to 2^k different possibilities for $S_{i-1} \setminus S_i$, and these 2^k sets partition S_{i-1}. So, for some constants $\lambda_1, \ldots, \lambda_{2^k} \geq 0$ such that $\Sigma\lambda_i = 1$, we have

$$E\left[\ln|S_i| - \ln|S_{i-1}| \mid J_k, \mathcal{F}_{i-1}\right] \quad = \quad \frac{1}{2^k}\sum_{l=1}^{2^k} \ln(1 - \lambda_l)$$

because each of the 2^k clauses occur with probability $\frac{1}{2^k}$. By Jensen's inequality, this is bounded from above by

$$\ln\left(\frac{1}{2^k}\sum_{i=1}^{2^k}(1-\lambda_i)\right) \quad = \quad \ln\left(\frac{1}{2^k}\left[2^k-\sum_{i=1}^{2^k}\lambda_i\right]\right) \quad = \quad \ln\left(\frac{2^k-1}{2^k}\right).$$

□

Proof of Lemma 41: We have

$$X_i - X_{i-1} \quad = \quad (\ln|S_i| - \ln|S_{i-1}|) - 2^k \ln\left(1-\tfrac{1}{2^k}\right)\left[A_i - A_{i-1} - 1 + \tfrac{1}{2^k} - \tfrac{1}{2^k}F_i\right]. \tag{6.5}$$

First assume $F_i = 1$. Putting this and $A_i - A_{i-1} \leq 1$ into (6.5) gives

$$X_i - X_{i-1} \quad \leq \quad \ln|S_i| - \ln|S_{i-1}| \quad \leq \quad 0,$$

with the last inequality true because $S_i \subseteq S_{i-1}$. Since $F_i = 1$, clause i corresponded to a bad k-tuple, therefore the algorithm chose a bad clause with probability $\frac{1}{2^k}$, and this choice makes $\ln|S_i| - \ln|S_{i-1}| \leq \ln(e^{-2}) = -2$.

Now assume $F_i = 0$, in this case we know that the algorithm did not accept a bad clause, so $0 \geq \ln|S_i| - \ln|S_{i-1}| > -2$. This with $0 \leq A_i - A_{i-1} \leq 1$ and (6.5) show

$$|X_i - X_{i-1}| \quad \leq \quad 2 - 2^k \ln\left(1-\tfrac{1}{2^k}\right) \quad \leq \quad 3.4$$

for all $k \geq 1$.

For an arbitrary k-tuple J_k there are 2^k possible clauses, so either $E[A_i - A_{i-1}|J_k, \mathcal{F}_{i-1}] = 1$ or $E[A_i - A_{i-1}|J_k, \mathcal{F}_{i-1}] \leq 1 - \frac{1}{2^k}$. In the latter case, $E[X_i - X_{i-1}|J_k, \mathcal{F}_{i-1}] \leq 0$ follows from (6.5) and $|S_i| \leq |S_{i-1}|$, and in the former case it follows from (6.5) and Lemma 40.

Since $E[X_i - X_{i-1}|J_k, \mathcal{F}_{i-1}] \leq 0$ for an any k-tuple J_k, we have $E[X_i - X_{i-1}|\mathcal{F}_{i-1}] \leq 0$. □

Proof of Lemma 42: We have

$$X_{cn} - X_0 \quad = \quad \sum_{\{0<i\leq cn : F_i = 0\}} (X_i - X_{i-1}) \; + \sum_{\{0<i\leq cn : F_i = 1\}} (X_i - X_{i-1}).$$

Let us write these two sums as Σ_0, Σ_1, respectively.

Sum Σ_0 adds at most cn random numbers, and from Lemma 41 we know each number has absolute value less than 3.4 and expected value non-positive. Thus, we can use (3.2):

$$\Pr\left(\Sigma_0 \geq n^{0.51}\right) \quad \leq \quad \exp\left(-\frac{(n^{0.51})^2}{2(cn)(3.4)^2}\right) \quad \leq \quad \exp\left(-\tfrac{1}{24c}n^{0.02}\right) \quad = \quad o(1).$$

Let Z be the number of times that the algorithm chooses a bad clause. Equation (6.3) from Lemma 41 shows $\Sigma_1 \leq -2Z$. So, **whp**

$$X_{cn} \quad \leq \quad X_0 + n^{0.51} - 2Z.$$

If $G_{cn} \leq n^{0.51}$ then $Z \geq 0$ shows (6.4). Otherwise, we can use Lemma 41 and (3.3):

$$\Pr\left(Z \leq \tfrac{1}{2^k}G_{cn} - G_{cn}^{0.51}\right) \quad \leq \quad \exp\left(-\frac{(G_{cn}^{0.51})^2}{8G_{cn}(1)^2}\right) \quad = \quad \exp\left(-\tfrac{1}{8}G_{cn}^{0.02}\right) \quad = \quad o(1),$$

so **whp**

$$X_{cn} \quad \leq \quad X_0 + n^{0.51} - 2\left(\tfrac{1}{2^k}G_{cn} - G_{cn}^{0.51}\right) \quad \leq \quad X_0 - \tfrac{2}{2^k}G_{cn} + n^{0.51} + 2G_{cn}^{0.51},$$

then $G_{cn} \leq cn$ shows (6.4). □

6.2 More about *Online-Lazy*

Define D_i as the number of variables that are set after the algorithm sees i clauses. Therefore, if ℓ is a randomly chosen literal after i clauses then $\Pr(\ell \text{ true}) = \Pr(\ell \text{ false}) = \frac{D_i}{2n}$. So, we have $D_0 = 0$ and

$$E[D_{i+1} - D_i] \quad = \quad \left(1 - \tfrac{D_i}{2n}\right)^k - \left(\tfrac{D_i}{2n}\right)^k,$$

this is the probability that none of the k literals are true minus the probability that all of them are false, because in this case we accept the clause and set one of the variables. Now, define B_i as we did in Section 6.1 as the number of clauses accepted minus $(1 - \frac{1}{2^k})i$. We have $B_0 = 0$ and

$$E[B_{i+1} - B_i] \quad = \quad \tfrac{1}{2^k} - \left(\tfrac{D_i}{2n}\right)^k$$

since a clause is accepted unless every literal chosen is false. Now, if we let $t = \frac{i}{n}$, $\Delta t = \frac{1}{n}$, and $d(t) = \frac{1}{n}D_i$, we can rewrite the first equation as

$$E[\tfrac{d(t+\Delta t)-D(t)}{\Delta t}] \quad = \quad \left(1 - \tfrac{1}{2}d(t)\right)^k - \left(\tfrac{1}{2}d(t)\right)^k.$$

As was discussed in Chapter 3, we can rewrite this as

$$d'(t) \quad = \quad \left(1 - \tfrac{1}{2}d(t)\right)^k - \left(\tfrac{1}{2}d(t)\right)^k,$$

and for any fixed k the solution $d(t)$ is such that $|d(t) - \frac{1}{n}D_{tn}| = O(\sqrt{n})$ **whp** for all t. Similarly, we can rewrite the other equation as

$$b'(t) \quad = \quad \tfrac{1}{2^k} - \left(\tfrac{1}{2}d(t)\right)^k,$$

and $b(t)$ is the function that interests us because $a_k = \lim_{t\to\infty} b(t)$.

So, for example when $k = 2$, solving this system with initial conditions $d(0) = 0, b(0) = 0$ gives $d(t) = 1 - e^{-t}$, $b(t) = \frac{1}{8}e^{-2t} - \frac{1}{2}e^{-t} + \frac{3}{8}$, and $a_2 = \frac{3}{8}$. The values for any other k are obtained in a similar manner, although for $k > 4$ we must solve the equations numerically.

6.3 Future Work

Perhaps the most natural question is to find the largest z_k for which there exists an online algorithm that accepts $(1 - \frac{1}{2^k})cn + z_k n$ out of cn clauses for any $k > 1$. (The $k = 1$ case is trivial, a greedy algorithm is easily seen to be optimal.) For $k = 2$, we analyzed an algorithm which appears to improve the 0.375 from *Online-Lazy* to about 0.453, and Theorem 16 gives an upper bound of 0.6024. The optimal algorithm and the exact value of the constant z_k are still unknown for any $k > 1$, although here we did show that as $k \to \infty$, $z_k \to \ln 2$.

When $k = 2$ there seems to be a very different situation when c is close to 1, i.e. if there are $(1 + \epsilon)n$ clauses for some small $\epsilon > 0$. The case of random 2-SAT when ϵ is $o(1)$ has been analyzed [38, 10]. When $\epsilon > 0$ is a constant, the

offline version must reject an expected $\Omega(\frac{\epsilon^3}{-\ln \epsilon} n)$ clauses [17], but whether the online version is closer to this or closer to $\Omega(\epsilon n)$ remains a mystery.

There is a natural $2n$-vertex graph which corresponds to any set of 2-clauses, and **whp** a giant component of size $\Omega(n)$ appears in this graph exactly when the probability of satisfiability goes to 0 [19, 15]. However, as was shown in [28], this is a coincidence. Whether or not an optimal algorithm uses the giant component in the online case remains to be seen.

BIBLIOGRAPHY

[1] Dimitris Achlioptas, Assaf Naor, and Yuval Peres, *On the maximum satisfiability of random formulas*, Proceedings of 44th Symposium on Foundations of Computer Science (FOCS 2003), IEEE Computer Society, October 2003, pp. 362–370.

[2] Dimitris Achlioptas and Yuval Peres, *The Fraction of Satisfiable Clauses in a Random Formula*, To appear in the Association for Computing Machinery.

[3] Dimitris Achlioptas and Yuval Peres, *The threshold for random k-SAT is* $2^k(\ln 2 + o(1))$, Proceedings of the 35th Annual ACM Symposium on Theory of Computing, 2003.

[4] David Aldous and Boris G. Pittel, On a random graph with immigrating vertices: Emergence of the giant component, *Random Structures and Algorithms* **17** (2000) 79-102.

[5] Noga Alon, Joel H. Spencer, and Paul Erdős, *The Probabilistic Method*, John Wiley and Sons, Inc., 1992.

[6] Krishna B. Athreya and Peter E. Ney, *Branching Processes*, Springer-Verlag (1972).

[7] Andrew Beveridge, Tom Bohman, Alan Frieze, and Oleg Pikhurko, *Product Rule Wins a Competitive Game*, preprint.

[8] Béla Bollobás, *Random Graphs*, Second Edition, Academic Press (2001).

[9] Béla Bollobás, *Modern graph theory*, Springer, New York, 1998.

[10] Béla Bollobás, Christian Borgs, Jennifer Chayes, Jeong Han Kim, and David Bruce Wilson, *The scaling window of the 2-SAT transition*, Random Structures and Algorithms **18** (2001), no.3., pp. 201–256.

[11] Tom Bohman and Alan Frieze, *Avoiding a Giant Component*, Random Structures and Algorithms, **19** (2001), 75–85.

[12] Tom Bohman, Alan Frieze, and Nicholas C. Wormald, *Avoidance of a giant component in half the edge set of a random graph*, **25** (2004) no. 4., pp. 432–449.

[13] Tom Bohman and Jeong Han Kim, *A Phase Transition for Avoiding a Giant Component*, Random Structures and Algorithms, **28** (2006) pp. 195–214.

[14] Tom Bohman and D. Kravitz, *Creating a Giant Component*, To appear in Combinatorics, Probability and Computing.

[15] Vašek Chvátal and Bruce Reed, *Mick gets some (the odds are on his side)*, 33rd Annual Symposium on Foundations of Computer Science, (Pittsburgh, PA, 1992), IEEE Computer Society Press, Los Alamitos, CA, 1992, pp. 620–627.

[16] Colin Cooper, Alan Frieze, and Gregory Sorkin, *Random 2-SAT with Prescribed Literal Degrees*, preprint.

[17] Don Coppersmith, David Gamarnik, Mohammad Hajiaghayi, and Gregory Sorkin, *Random MAX SAT, random MAX CUT, and their phase transitions*, Random Structures and Algorithms, **24** (2004), no. 4., pp. 502–545.

[18] Oliver Dubois, Yacine Boufkhad, and Jacques Mandler. *Typical random 3-SAT formulae and the satisfiability threshold*. In Proceedings of the 11th Annual Symposium on Discrete Algorithms, pp. 126–127, 2000.

[19] Paul Erdős and Alfréd Rényi, *On the Evolution of Random Graphs*, Publications of the Mathematical Institute of the Hungarian Academy of Sciences, **5**, pp. 17–61.

[20] Wenceslas Fernandez de la Vega, *Random 2-SAT: results and problems*, Theoretical Computer Science **265** (2001), no. 1-2, pp. 131–146.

[21] Abraham Flaxman, David Gamarnik, and Gregory Sorkin, *Embracing the giant component*, Proceedings of the 6th Conference of Latin American Theoretical Informatics (2004) 69–79.

[22] Ehud Friedgut, *Necessary and sufficient conditions for sharp thresholds of graph properties, and the k-SAT problem*, Journal of the American Mathematical Society, **12** (1999), pp. 1017–1054.

[23] Andreas Goerdt, *A threshold for unsatisfiability*, Journal of Computer and System Sciences **53** (1996), no. 3, pp. 469–486.

[24] Svante Janson, Thomasz Łuczak, and Andrzej Ruciński, *Random Graphs*, John Wiley & Sons, Inc. (2000).

[25] Alexis C. Kaporis, Lefteris M. Kirousis, and Efthimios Lalas, *The Probabilistic Analysis of a greedy satisfiability algorithm*, Fifth International Symposium on the Theory and Application of Satisfiability Testing, 2002, pp. 362–376.

[26] Richard M. Karp. and Michael Sipser, *Maximum matchings in sparse random graphs*, Proceedings of the Twenty-Second Annual IEEE Symposium on Foundations of Computer Science, (1981), pp. 364–375.

[27] D. Kravitz, *On an Online Random k-SAT model*, Submitted to Random Structures and Algorithms.

[28] D. Kravitz, *RANDOM 2-SAT Does not depend on a giant*, Submitted to Random Structures and Algorithms.

[29] Thomas G. Kurtz, *Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes*, Journal of Applied Probability, **7** (1970) pp. 49–58.

[30] Colin McDiarmid, "Concentration", In *Probabilistic Methods for Algorithmic Discrete Mathematics*, (Springer, 1998), pp. 195–248.

[31] Thomasz Łuczak, Boris G. Pittel, and John C. Wierman, *The structure of a random graph at the point of the phase transition*, Transactions of the American Mathematical Society, **341** (1994) pp. 721–748.

[32] Michael Molloy, *When does the giant component bring unsatisfiability?* Combinatorica (to appear).

[33] Michael Molloy and Bruce Reed, *A critical point for random graphs with a given degree sequence*, Random Structures and Algorithms, **6** (1995) pp. 161–180.

[34] Boris G. Pittel, *A random graph with a subcritical number of edges*, Transactions of the American Mathematical Society, **309** (1988) pp. 51–75.

[35] Gregory Sorkin. personal communication.

[36] Joel H. Spencer, *Percolating Thoughts*, email message of January 31, 2001.

[37] Joel H. Spencer and Nicholas C. Wormald, *Birth control for giants*, preprint.

[38] Yann C. Verhoeven, *Random 2-SAT and unsatisfiability*, Information Processing Letters, **72** (1999) no. 3-4, pp. 119–123.

[39] Nicholas C. Wormald, *The Differential Equation Method For Random Graph Processes and Greedy Algorithms*, in Lectures on Approximation and Randomized Algorithms, (M. Karoński and H.J. Prömel, editors) (1999) pp. 73–155.

[40] Nicholas C. Wormald, *Differential Equations for random processes and random graphs*, The Annals of Applied Probability **5** (1995), no. 4, pp. 1217–1235.