

Random Graphs as both a Process & Model with emphasis on the Giant Component

Alex Henniges

University of Arizona
Mentor: Dr. Nicholas Ercolani

October 30, 2012

1 Introduction

In the study of random graphs, it has been well-known since the work of Erdős and Rényi that for graphs with enough edges, there is a largest component of the random graph known as the giant component. The component is the largest in the sense that it is the only component whose size, that is the number of vertices belonging to the component, is on the order of the total number of vertices of the graph. In other words, if n denotes the number of vertices of the random graph, all other components have sizes at most $\beta \log n$ for some constant β .

Previous work in this area has shown that if the number of edges of the random graph is slightly greater than $\frac{n}{2}$, then there will almost always be a giant component in the limit as the number of vertices increases. Likewise, if the number of edges is slightly less than $\frac{n}{2}$, no components have size on the order of n . When a random graph is created by randomly selecting edges one at a time, the period of time around $\frac{n}{2}$ edges can be thought of as the critical phase, and that the giant component is formed from the small components at this time.

A component with more edges than vertices is called complex. It might be expected that a component as large as the giant component is complex and perhaps that any component that would be complex would be absorbed by the giant component during creation. In fact, the paper of Janson, Knuth, Luczak, and Pittel [2] was formed on the rumor that, at $\frac{n}{2}$ edges, there is only one complex component, the giant one. While Stepanov showed that this wasn't the case, Janson, et al improved upon the results and showed that the probability that a random graph with n vertices and $\frac{n}{2}$ edges has at most one complex component is $0.9957 + O(n^{-1/3})$.

This paper explores some of the results of random graph theory and the giant component from several perspectives. First, we study the creation of random graphs via the random graph process as proposed by Erdős and Rényi and the results about complexity that arise from this method as shown in [2]. Then, we will approach random graphs from a different angle via the random graph model. Using [1] as a guide, we show how to use branching processes to prove

the existence of the giant component and then state some results about the properties of this giant component. The results throughout are supplemented by numerical data obtained through simulations that explore the strength of convergence and bounds given in the theorems.

2 Random Graph Processes

A random graph is a graph where the edges are chosen randomly. How the edges are chosen has an effect on calculations. When the edges are chosen and added to the graph one at a time, it is known as a random graph process. We consider two processes, one that allows for multigraphs and one that doesn't.

Definition 2.1. *The uniform model, or multigraph process, on n vertices is defined as follows: Randomly choose an ordered pair of vertices $\langle x, y \rangle$ and add an edge between vertices x and y . Each ordered pair is equally likely to be chosen, with probability $\frac{1}{n^2}$. Repeat until m edges are chosen.*

This process uses sampling with replacement. That is, after an edge is chosen, it is placed back as a candidate to be chosen again. Furthermore, this process allows for choosing an ordered pair of the same vertex, $\langle x, x \rangle$. A graph that has either self-loops, edges connecting a vertex to itself, or multiple edges between a pair of vertices is called a multigraph. The next process allows only for graphs.

Definition 2.2. *The permutation model, or uniform model without replacement, on n vertices is defined as follows: Let $N = \binom{n}{2}$ be the number of possible edges, and randomly choose one of the $N!$ orderings of the edge. Introduce the edges to the graph in that order.*

Both of the previous two processes involve an evolution of the graph. In other words, edges are randomly added to the graph in steps, known as the evolution, and the graph can be evaluated at each step.

For either case, we can represent a graph or multigraph by a matrix M , where an entry m_{xy} is the number of edges between vertices x and y . We note that $m_{xy} = m_{yx}$, so the matrix is symmetric. Furthermore, M represents a

graph if and only if M has only 0's and 1's and the diagonal contains only 0's.

Not all multigraphs are created equal. We can differentiate between the likelihood of multigraphs by defining a weight κ :

$$\kappa(M) = \frac{1}{\prod_{x=1}^n 2^{m_{xx}} \prod_{y=x}^n m_{xy}!}.$$

In short, multigraphs with self-loops and multiple edges are less likely to occur. This is because there are fewer possible sequences of the multigraph process that will produce the multigraph M . As there are n^{2m} sequences for multigraphs with n vertices and m edges, the probability that a random multigraph M is chosen is $\frac{2^m m! \kappa(M)}{n^{2m}}$, as there are $2^m m!$ sequences that will produce M , all with weight $\kappa(M)$.

To provide a little more of the basic theory that lends itself to the proofs of the results that follow, we consider generating functions for multigraphs. Let \mathcal{F} be a family of multigraphs. The bivariate generating function (bgf) associated to the family \mathcal{F} is

$$F(w, z) = \sum_{M \in \mathcal{F}} \kappa(M) w^{m(M)} \frac{z^{n(M)}}{n(M)!},$$

where $m(M)$ and $n(M)$ are the number of edges of M and the number of vertices of M , respectively. By collecting terms of multigraphs with the same edges and vertices, we can rewrite F as

$$F(w, z) = \sum_{m, n=0}^{\infty} a_{n, m} w^m \frac{z^n}{n!}.$$

Using this notation, it follows that the probability that a random multigraph with m edges and n vertices is in \mathcal{F} is $\frac{2^m m!}{n^{2m}} a_{n, m}$.

Now, note that if a multigraph M consists of components M_1, \dots, M_k that are pair-wise disconnected, then based on the definition of κ , it should be clear that $\kappa(M) = \kappa(M_1) \cdot \dots \cdot \kappa(M_k)$. Thus, if F_1, \dots, F_k are bgf's of multigraph families $\mathcal{F}_1, \dots, \mathcal{F}_k$, the product $F_1(w, z) \cdot \dots \cdot F_k(w, z)$ is a bgf for the ordered

tuples (M_1, \dots, M_k) where $M_i \in \mathcal{F}_i$. This suggests that we should focus our attention on the bgf's for connected multigraphs. Then, a general multigraph can be decomposed into its connected components:

Let $C(w, z)$ be the bgf for the family of all connected multigraphs, and let $G(w, z)$ be the bgf for the family of all multigraphs. Then by the above,

$$\begin{aligned} e^{C(w, z)} &= \sum_{k \geq 0} \frac{C(w, z)^k}{k!} \\ &= G(w, z), \end{aligned}$$

as $C(w, z)^k/k!$ represents the bgf for the family of all multigraphs having k connected components. When restricting to the case of only random graphs, we denote by $\hat{C}(w, z)$ and $\hat{G}(w, z)$ the bgf's for connected and all graphs, respectively.

2.1 Complexity of the Graph

Definition 2.3. *The excess of a connected multigraph, $r(M)$, is*

$$r(M) = m(M) - n(M).$$

The minimum possible excess is -1 , the excess of a tree. A connected multigraph or component with excess 0 is said to be a unicycle, or unicyclic. Similarly, a component of excess 1 is bicyclic and so on. Any connected multigraph or component with excess $r(M) > 0$ is called complex.

Now, let $C_r(w, z)$ be the bgf for the family of all connected multigraphs of excess r . Similarly, let $\hat{C}_r(w, z)$ be the bgf for all connected graphs of excess r . At times, we ignore the number of edges by setting $w = 1$ and consider the univariate generating function $C_r(z)$. We can then write $C(w, z)$ as

$$\begin{aligned} C(w, z) &= \sum_{r=-1}^{\infty} C_r(w, z) \\ &= \sum_{r=-1}^{\infty} w^r C_r(wz). \end{aligned}$$

We will now focus on a main result of [2]. This theorem states the likelihood of a multigraph to have complex components near the critical phase.

Theorem 2.4. *The probability that a random graph or multigraph with n vertices and $\frac{1}{2}n + O(n^{1/3})$ edges has exactly r_1 bicyclic components, r_2 tricyclic components, ..., $r_q(q+1)$ -cyclic components, and no components of higher cyclic order, is*

$$\left(\frac{4}{3}\right)^r \sqrt{\frac{2}{3}} \frac{c_1^{r_1}}{r_1!} \frac{c_2^{r_2}}{r_2!} \cdot \dots \cdot \frac{c_q^{r_q}}{r_q!} \frac{r!}{(2r)!} + O(n^{-1/3}),$$

where $r = r_1 + 2r_2 + \dots + qr_q$ and for constants c_j .

The constants c_j are obtained through work with the generating functions, but we will provide the first few here, noting that a more in-depth discussion is provided in [2]:

$$c_1 = \frac{5}{24}, c_2 = \frac{5}{16}, c_3 = \frac{1105}{1152}, c_4 = \frac{565}{128}, c_5 = \frac{82825}{3072}$$

For notation, we represent a multigraph with r_1 bicycles, r_2 tricycles, ..., and $r_q(q+1)$ -cycles as $[r_1, r_2, \dots, r_q]$ and we will call this the multigraph's complex type.

From the theorem, one can conclude that as n goes to infinity, the probability that a multigraph has no complex components in the critical phase is about 81.6%. Moreover, by summing over the probabilities that $r_i = 1$ and all other $r_j = 0$ yields that a multigraph has at most one complex component during the critical phase with probability $0.9957 + O(n^{-1/3})$. Hence, a complex component found during the critical phase is likely to be the giant component, but most of the time there are no complex components yet.

In exploring this theorem numerically, we will focus on two things. First, what is the behavior of the convergence as n increases? And second, how tight is the bound $O(n^{-1/3})$? To study these, we created 10,000 random multigraphs for each of 100, 1000, 5000, 10000, and 20000 vertices. Then we counted the number of occurrences of five of the most common complex types, $[0], [1], [2], [0, 1], [0, 0, 1]$. The results, along with the expected probability (ignoring the bound) given by the theorem, can be found in Table 1.

Averages	100	1000	5000	10000	20000	Expected
[0]	0.9550	0.8885	0.8563	0.8556	0.8434	0.8165
[1]	0.0372	0.0811	0.0949	0.0968	0.1032	0.1134
[2]	0.0000	0.0007	0.0011	0.0016	0.0017	0.0026
[0, 1]	0.0064	0.0215	0.0304	0.0279	0.0315	0.0378
[0, 0, 1]	0.0012	0.0059	0.0106	0.0103	0.0121	0.0154

Table 1: The occurrence of complex types for varying number of vertices.

From the table, it appears that multigraphs of complex type [0] appear more often for small n and the probability approaches the expected value from above. On the otherhand, all other complex types have probabilities that approach the expected value from below.

To study the bound provided in Theorem 2.4, let $E_n([r_1, \dots, r_q])$ denote the actual probability that a random multigraph with n vertices and $\frac{n}{2}$ edges has complex type $[r_1, \dots, r_q]$. Let $E_\infty([r_1, \dots, r_q])$ be the expected probability (ignoring the bound) for complex type $[r_1, \dots, r_q]$ as given by the formula in the theorem. Then if the logarithm plot of $|E_n([r_1, \dots, r_q]) - E_\infty([r_1, \dots, r_q])|$ against $\log n$ is a line with slope m , it follows that

$$|E_n([r_1, \dots, r_q]) - E_\infty([r_1, \dots, r_q])| \approx n^m.$$

Therefore, the question becomes whether or not the log plots are lines and if the slopes are $-\frac{1}{3}$. Using the table above, we approximate the E_n and obtain the following scatter plot (with best-fit line) and slopes as given in Table 2.

Type	Slopes
[0]	-0.3014
[1]	-0.3604
[2]	-0.2014
[0,1]	-0.2967
[0,0,1]	-0.2688

Table 2: The slopes of the best-fit lines.

While nothing too definitive can be drawn from the data, the bound of $n^{-1/3}$

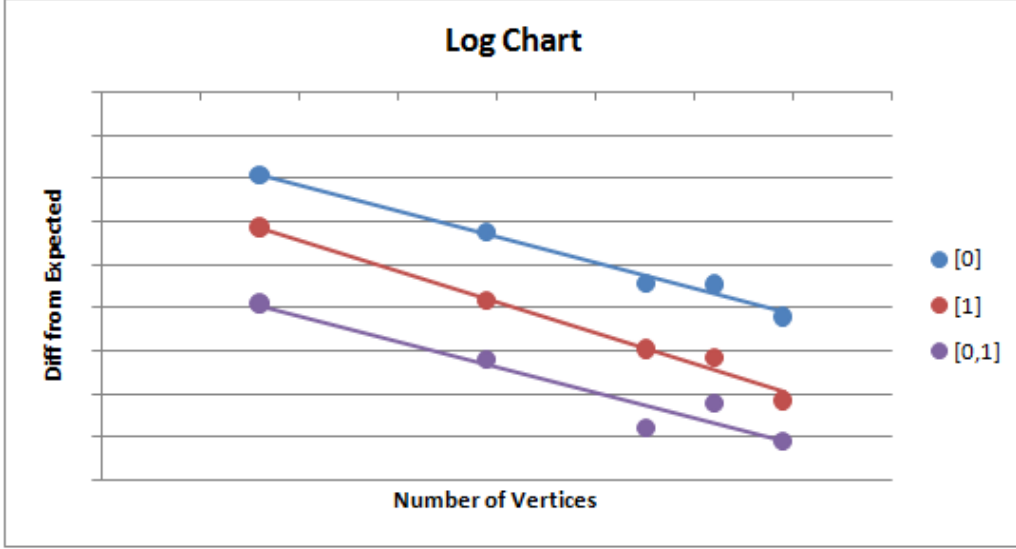


Figure 1: The scatter plots for types $[0]$, $[1]$, and $[0, 1]$

seems to be fairly accurate.

3 Random Graph Models

In the random graph process, each edge added was a step in the process and the state of the graph changed from step to step. Also, this meant that the number of edges of the graph was deterministic. In the random graph model, there is only ever one state for a single instance of the random graph. An edge between vertices i and j is assigned a Bernoulli random variable X_{ij} with parameter p . That is, there will be an edge between vertices i and j with probability p . The same is true for each possible edge and so all edges occur at one time.

The parameter p will instead be written λ/n , where n is the number of vertices, as the λ will have increasing importance. As a first example, note that the number of edges in the random graph is given by the random variable $\sum_{i=1}^n \sum_{j=i+1}^n X_{ij}$, which is a Binomial distribution, $\text{Bin}(\frac{n(n-1)}{2}, \lambda/n)$. Thus, the expected number of edges is given by $\lambda \cdot \frac{n-1}{2}$. As we will often consider

large n , this can be thought of more simply as $\lambda \cdot \frac{n}{2}$. Thus, the value for λ determines the average number of edges as a fraction of the number of vertices. Using the random graph process as a guide, this also suggests what values of λ to consider. Since the critical phase for the random graph process occurs at $n/2$ edges, it would be reasonable to study the random graph model for $\lambda = 1$. Also, small changes in λ have a considerable effect on the number of edges, so increases in λ during tests do not need to be very large.

The advantages to the random graph model are numerous. For one, calculations of probabilities can be easier as there are not multiple states of the graph to contend with. Also, the random graph model can be approximated by a branching process, as discussed in the next section, and also a random walk. Lastly, in situations that can be modeled by a random graph, a deterministic number of edges may not be preferred.

On the other hand, the random graph process is also useful. For one, it is the more natural way to think of creating a random graph (certainly, to program the random graph model, one must add the edges one at a time). It also allows for more flexibility in creating random graphs. For example, in a popularity model, where edges are more likely to be added to vertices with larger degrees. In a case like this, the edges need to be added one at a time, with the probabilities changing as the state changes. As usual, the best course of action is to find the relationship between the two and choose the best one for each situation.

3.1 Branching Processes

The real power of the random graph model is that results can be proved using a branching process to mimic the model. Nowhere is this more evident than in proving the existence of the giant component. The basic idea is that every component of the random graph can be thought of as a branching process with any of its vertices as the root. Each vertex of the branching process has a nonnegative integer-valued random variable that represents the number of children of the vertex. Those processes that reach extinction are small tree components, whereas all of the branching processes that survive are merged into a unique giant component. When the random variables in the branching

process are Poisson distributions in particular, this theory directly leads to results on the size of the giant component as well as other properties of the random graph, such as the complexity of the components.

Definition 3.1. Let ζ_i^t be identical, independently-distributed non-negative integer-valued random variables for $i, t \geq 0$. Let $Z_0 = 1$ and $Z_t = \zeta_1^t + \dots + \zeta_{Z_{t-1}}^t$ if $Z_{t-1} > 0$, and 0 otherwise. Then the Z_t 's represent a branching process, called a Galton-Watson process, where the ζ_i^t represent the offspring distribution. Let $p_k = P(\zeta_i^t = k)$.

Let μ be the expectation of the offspring distribution. Intuitively, if $\mu < 1$, then on average a member of the branching process produces less than 1 offspring and it can be expected that the branching process will eventually die. On the other hand, if $\mu > 1$, then it could be that the branching process will continue indefinitely. Indeed, [1] shows that Z_t/μ^t is a martingale and as a consequence if $\mu < 1$, then $Z_t = 0$ for all t sufficiently large.

In the case of $\mu > 1$, we have the following result:

Theorem 3.2. If $\mu > 1$, then $P(Z_t = 0 \text{ for some } t) = \rho$, where $\rho < 1$. In other words, the branching process will continue indefinitely with probability $1 - \rho$.

The proof states how to calculate the *extinction probability*, ρ , so we will give an outline below:

Proof. Let $g(\theta) = \sum_{k=0}^{\infty} p_k \theta^k$, $\theta \in [0, 1]$ be the probability generating function for the offspring distribution ($p_k = P(\zeta_i^t = k)$). We will show that this generating function has a unique fixed point on the interval $[0, 1)$, which we will call ρ , and that this fixed point is the extinction probability. Of course, $g(1) = 1$ is the other fixed point of the function.

Now, $g'(\theta)$ and $g''(\theta)$ both converge and are positive for $\theta < 1$, so g is both increasing and concave up. $g'(1) = \sum_{k=1}^{\infty} k p_k = \mu > 1$, so it follows that the graph of $g(\theta)$ must lie below the line of fixed points, $f(\theta) = \theta$. If 0 is not a fixed point of g , so $g(0) > 0$, then the graph of g lies above f near 0. Hence,

the two graphs must intersect at exactly one point, ρ .

The function g has another purpose: If we let $\theta_t = P(Z_t = 0)$, then $\theta_t = g(\theta_{t-1})$. This can be reasoned from observing that all the direct children of the root are branching processes that must become extinct in $t - 1$ steps, then partitioning on the number of children of the root.

Now, the sequence of θ_t 's is increasing since g is increasing, so $\theta_t > \theta_{t-1}$. Moreover, $\theta_0 = 0$ since Z_0 is identically 1. So the sequence is bounded by ρ . Therefore, $\lim_{t \rightarrow \infty} \theta_t$ exists, which we temporarily call θ_∞ . Taking the limit as t approaches infinity on both sides of $\theta_t = g(\theta_{t-1})$ shows that θ_∞ is a fixed point at most ρ , so $\theta_\infty = \rho$.

Lastly, the probability that the branching process becomes extinct is precisely the probability that Z_t is zero for some t , which is $\lim_{t \rightarrow \infty} \theta_t = \rho$. \square

As a loaded example, suppose that the offspring distribution is given by $\text{Poisson}(\lambda)$, so that $\mu = \lambda > 1$ and $p_k = e^{-\lambda} \frac{\lambda^k}{k!}$. Then

$$\begin{aligned} g(\theta) &= \sum_{k=0}^{\infty} e^{-\lambda} \frac{\lambda^k \theta^k}{k!} \\ &= e^{-\lambda} e^{\lambda \theta} \\ &= e^{\lambda(\theta-1)}. \end{aligned}$$

Thus, the extinction probability is the value ρ that makes $\rho = e^{\lambda(\rho-1)}$.

3.1.1 Connection to Random Graphs

Recall that an edge of a random graph appears with probability λ/n , represented by a Bernoulli random variable X_{ij} where n is the number of vertices. Therefore, the degree of a vertex of the graph is a binomial random variable, $X_i = \sum_{j \neq i} X_{ij} = \text{Bin}(n-1, \lambda/n)$. Note that for large n , the expected degree of any vertex is about λ .

The first connection to a branching process is that we can think of the degree of a vertex to be the number of children in a branching process with the designated vertex as the root. Therefore, λ is analogous to μ . Indeed, when $\lambda = 1$, the expected number of edges in the random graph is about $n/2$, the critical phase in a random graph process. Moreover, as n approaches ∞ , the random variable $\text{Bin}(n-1, \lambda/n)$ approaches a Poisson random variable with parameter λ in probability. This is why an offspring distribution represented as a Poisson distribution with parameter μ best mimics a random graph model.

3.2 Properties of the Giant Component

3.2.1 Size of the Giant Component

Suppose that a vertex from a random graph with $\lambda > 1$ is chosen and consider the component that this vertex belongs to as a branching process. If the process becomes extinct, this component is likely not very large and not very complex. On the other hand, if the branching process survives, then the component would have an infinite number of vertices. Of course, the random graph has only a finite number of vertices, n , so we could instead construe survival as the component merging with any other survived branching processes, and all of these forming the giant component. If this notion were to be believed, then since a branching process survives with probability $1 - \rho$, where ρ is the extinction probability, it would be expected that the giant component would have size $n(1 - \rho)$ as we can begin a branching process at any vertex. Indeed, intuition proves to be true in the following theorem:

Theorem 3.3. *Suppose $\lambda > 1$. There is a constant β so that with probability approaching 1, there is only one component of the random graph with more than $\beta \log n$ vertices, namely the giant component. This component has size about $n(1 - \rho)$, where ρ is the extinction probability for the $\text{Poisson}(\lambda)$ branching process.*

This prediction on the size of the giant component can be quite accurate, even for small n . The accuracy appears to depend on two factors: the size of λ and the size of n . The more important factor is λ . As λ increases, the error decreases. As an example, with $\lambda = 1.1$ and $n = 1000$, the predicted size of the largest component is about 176, and the tested average was about

168, an error of about 4.5%. On the otherhand, with $\lambda = 1.5$, the predicted size is about 583 and the tested average was around 577, an error of about 1%. At $\lambda = 5$, the error is about 0.002%.

On the other hand, as λ approaches the critical phase of 1, the actual average can be quite different from the predicted size. For example, at $\lambda = 1.01$ and $n = 1000$, the expected size is only about 20, but in reality the average was greater than 97 vertices for the giant component. Another interesting occurrence is that for λ near 1, the tested average was greater than the expected size, but as λ increases, the tested average becomes less than the predicted size. The question then becomes: for a fixed n , is there a value of λ near 1 in which the predicted size is as accurate as when λ is much larger than 1? For $n = 1000$, this value seems to be near $\lambda = 1.1$.

The accuracy of the theorem improves as n increases for λ near 1 (in our tests, $\lambda < 2$), but in the cases of large λ , there can be more error. Of course, at such high λ the error is not very large to begin with, so more runs to improve the accuracy of the average test size could make a difference.

The constant β in the theorem is more specifically $1/\alpha$ where $\alpha = \lambda - 1 - \log(\lambda)$. As stated in [1], this bound is derived from several estimates and is not a precise indicator of the size of the second largest component for small n , but is the size asymptotically. In fact, the size of the second largest component ranged from only about 7 percent of the bound to 33 percent on average in numerical tests, but appears that it may approach the bound as λ and n increase (see Table 6).

Knowing that the giant component does exist, one could consider the size of the largest component to be a random variable. As it turns out, this random variable, or at least a slight modification, satisfies a central limit theorem.

Theorem 3.4. *Suppose $\lambda > 1$ and let $|\mathcal{C}^{(1)}|$ be the size of the largest component. Then the random variable*

$$\frac{|\mathcal{C}^{(1)}| - n(1 - \rho)}{\sqrt{n}}$$

converges to a normal distribution with mean 0 and variance $(\rho - \rho^2)/(1 - \lambda\rho)$.

This result lends itself to numerical tests really well. There are a number of convergence-related questions that can be asked, such as the effect of λ , n , and the number of experiments run. The full results of our tests can be found in section 4.3. While keeping everything else constant, increasing the number of vertices does cause the random variable to approach a normal distribution and also improves the variance to the expected value. However, the variance is not close to the expected even at 10000 vertices (an amount that is usually suitably large for other tests) for small λ . It does appear that λ close to 1 are more volatile, whereas as λ increases, the variance converges.

3.3 Complexity of Components

Another advantage of the random graph model over the random graph process is that we can predict the occurrence of tree and unicyclic components using straight-forward combinatorics. The main goal is to determine the expected value for the number of trees or unicyclic components of size k appearing in the graph and also the expected total number of trees and unicyclic graphs. The important step is knowing how many different trees or unicycles there are of a given size.

Let's begin with trees. The probability of a specific tree with k -labeled vertices appearing in the random graph is

$$\binom{n}{k} (\lambda/n)^{k-1} (1 - \lambda/n)^{k(n-k) + \binom{k}{2} - (k-1)}$$

as the chosen k vertices must connect to each other with $k - 1$ edges, but cannot be connected to any other vertex in the graph, nor can there be any other edges in the component. Now, it is well-known that there are k^{k-2} different trees with k -labeled vertices, and so the expected number of trees of size k is

$$k^{k-2} \binom{n}{k} (\lambda/n)^{k-1} (1 - \lambda/n)^{k(n-k) + \binom{k}{2} - (k-1)}$$

For unicyclic components we can do the same thing. There are k edges to the component, and using the result by Rényi that the number of unicyclic

graphs of size k is

$$v_k = \frac{(k-1)!}{2} \sum_{j=0}^{k-3} \frac{k^j}{j!},$$

the expected number of unicyclic components of size k is

$$v_k \binom{n}{k} (\lambda/n)^k (1 - \lambda/n)^{k(n-k) + \binom{k}{2} - k}.$$

Note that as $n \rightarrow \infty$, this expected value converges to $\frac{v_k}{k!} (\lambda e^{-\lambda})^k$. So as the number of vertices increases, the number of unicyclic components of size k approaches a finite value, unlike the case for trees.

Moreover, if we wished to know the expected total number of unicyclic components, v_k is about $(\pi/8)^{1/2} k^{k-1/2}$ for large k , and so we can sum the expected values over all k if $\lambda \neq 1$. The result is that for $\lambda > 1$, the expected number of unicyclic components in the random graph is

$$\frac{-\ln(1 - \lambda\rho) - \lambda\rho - (\lambda\rho)^2}{2}.$$

This result is intriguing, as [2] states that the generating function for unicyclic components is

$$\hat{V}(z) = \frac{-\ln(1 - T(z)) - T(z) - \frac{1}{2}T(z)^2}{2},$$

where $T(z)$ is the generating function for rooted labeled trees. A relationship, if it exists, between these two could prove useful when dealing with more complex components.

4 Numerical Results

4.1 Size of the Giant Component

The following table (Table 3) gives the expected and actual average size of the giant component for various λ with a fixed number of vertices at $n = 1000$. 1000 runs per entry was used to determine the average size.

λ	Extinction Prob. (ρ)	Expected Size ($n(1 - \rho)$)	Simulated Size	Error
1.01	0.980264	19.736	97.594	79.78%
1.05	0.906298	93.702	132.909	29.50%
1.1	0.823866	176.134	167.949	4.87%
1.2	0.686300	313.700	292.851	7.12%
1.3	0.577030	422.970	406.582	4.03%
1.4	0.488989	511.011	502.986	1.60%
1.5	0.417188	582.812	576.684	1.06%
2.0	0.203188	796.812	794.984	0.23%
3.0	0.059520	940.480	940.351	0.01%
5.0	0.006977	993.023	993.003	0.00%

Table 3: 1000 vertices, 1000 runs

The next two tables (Tables 4 and 5) shows a smaller set of λ values, but now with 5000 and 10000 vertices. Notice that error term appears to improve as the number of vertices increases.

λ	Extinction Prob. (ρ)	Expected Size ($n(1-\rho)$)	Simulated Size	Error
1.2	0.686300	1568.5	1526	2.79%
1.5	0.417188	2914.06	2907	0.24%
2.0	0.203188	3984.06	3982	0.05%

Table 4: 5000 vertices, 1000 runs

λ	Extinction Prob. (ρ)	Expected Size ($n(1-\rho)$)	Simulated Size	Error
1.2	0.686300	3137	3106	1.00%
1.5	0.417188	5828.12	5822	0.11%
2.0	0.203188	7968.12	7970	0.02%

Table 5: 10000 vertices, 1000 runs

The following table (Table 6) gives the average size of the second largest component for a varying number of vertices and λ . For comparison the bound provided in [1] is given as well as the ratio to this expected bound.

	λ	Bound	Average	Ratio
500 Vertices	1.2	351.536	26.816	7.63%
	1.4	97.825	16.254	16.62%
	1.6	47.806	9.609	20.10%
	1.8	29.285	6.752	23.06%
	2	20.253	5.191	25.63%
1000 Vertices	1.2	390.745	41.228	10.55%
	1.4	108.736	18.121	16.67%
	1.6	53.138	11.265	21.20%
	1.8	32.551	8.153	25.05%
	2	22.512	6.145	27.30%
5000 Vertices	1.2	481.784	63.395	13.16%
	1.4	134.070	28.098	20.96%
	1.6	65.519	16.26	24.82%
	1.8	40.135	11.597	28.89%
	2	27.757	8.815	31.76%
10000 Vertices	1.2	520.993	76.297	14.64%
	1.4	144.981	32.271	22.26%
	1.6	70.851	19.324	27.27%
	1.8	43.401	13.442	30.97%
	2	30.015	10.149	33.81%

Table 6: The size of the second largest component. The bound is given in [1].

4.2 Complexity Counts

Table 7 gives the average number of occurrences of trees of size k in the random graph, varying the number of vertices, the probability λ , and the value k . As can be seen, the numbers are extremely close to the expected value given in section 3.3. As the expected occurrence of unicycles for even small values of k is extremely small, the results are omitted here.

Number of Vertices	λ	Size of tree	Number of trees	Expected
500 Vertices	0.5	1	303.58	303.27
		5	2.57	2.67
		10	0.18	0.18
	1	1	180.62	183.94
		5	3.45	3.51
		10	0.66	0.63
	1.5	1	112.27	111.57
		5	1.59	1.46
		10	0.22	0.16
1000 Vertices	0.5	1	607.5	606.53
		5	5.53	5.34
		10	0.43	0.36
	1	1	368.48	367.88
		5	7.17	7.02
		10	1.48	1.25
	1.5	1	224.43	223.13
		5	2.98	2.92
		10	0.29	0.32
5000 Vertices	0.5	1	3034.6	3032.65
		5	26.93	26.72
		10	2.04	1.81
	1	1	1831.02	1839.4
		5	34.86	35.09
		10	6.2	6.26
	1.5	1	1123.12	1115.65
		5	14.76	14.58
		10	1.7	1.62

Table 7: The number of trees of size k .

4.3 Central Limit Theorem

The following histograms (Figures 2 to 5) show the results of comparing the size of the largest component to a normal distribution. λ is fixed at 1.5. The histograms vary by the number of vertices. For comparison, the expected variance given in the theorem is about 0.650.

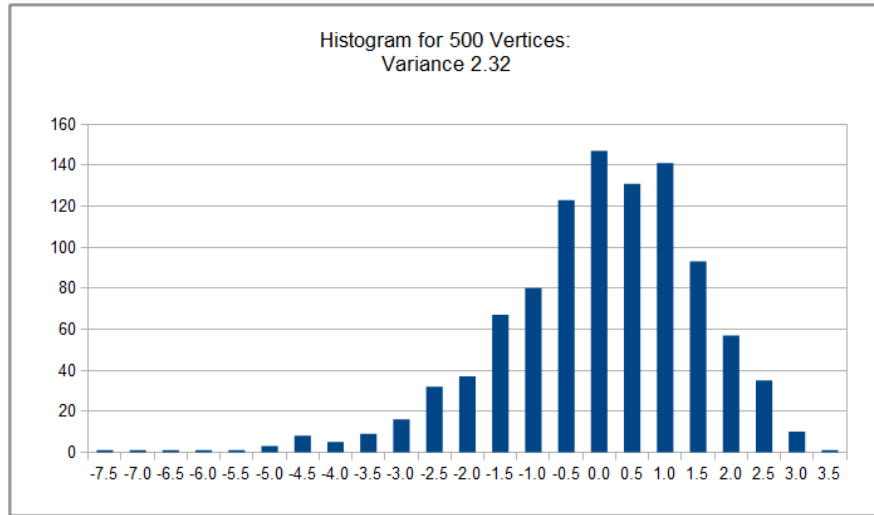


Figure 2: The histogram for 500 vertices.

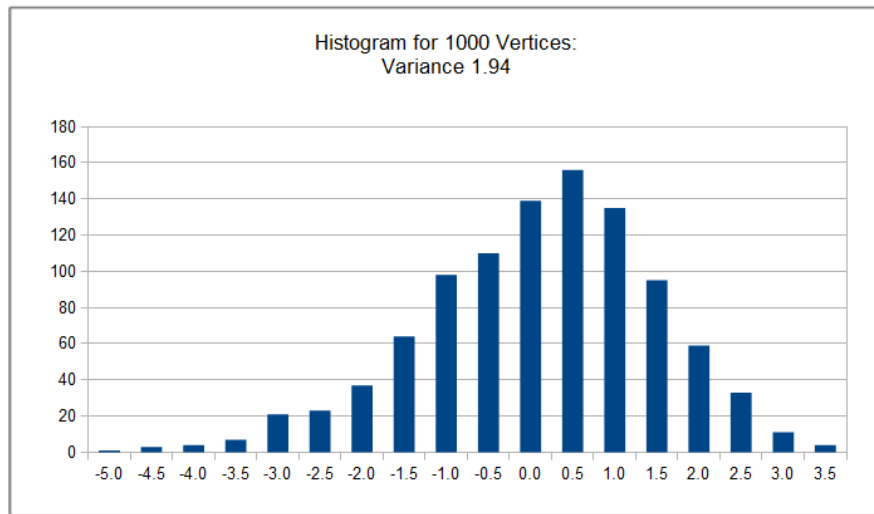


Figure 3: The histogram for 1000 vertices.

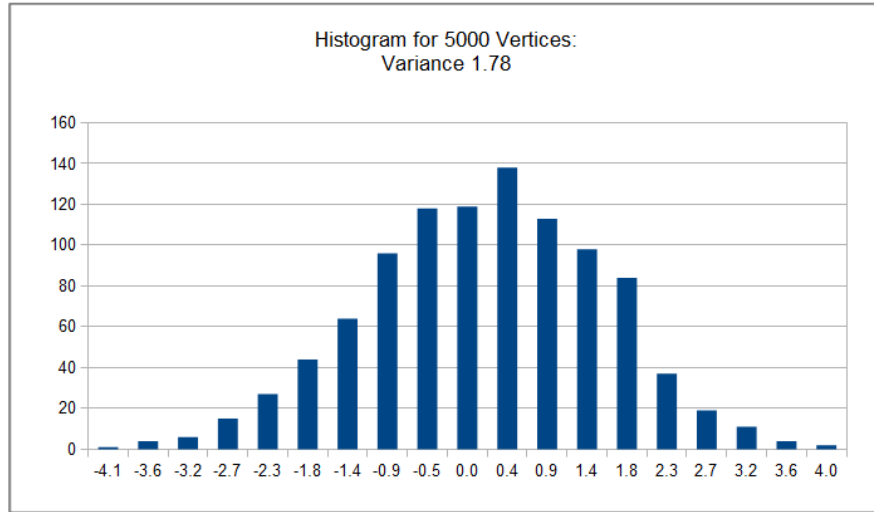


Figure 4: The histogram for 5000 vertices.

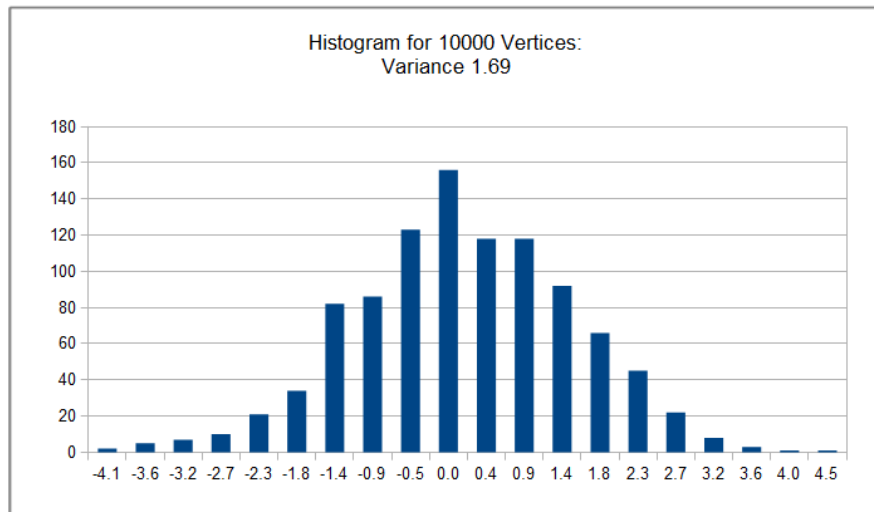


Figure 5: The histogram for 10000 vertices.

References

- [1] Rick Durrett *Random Graph Dynamics*. Cambridge, 2007
- [2] Svante Janson, Donald Knuth, Tomasz Luczak, Boris Pittel, *The Birth of the Giant Component*. Oct 1993.
- [3] Svante Janson, Tomasz Luczak, Andrzej Rucinski, *Random Graphs*.
- [4] Martin Aigner, *A Course in Enumeration*. Springer, 2007.