Phase Transition Continuity Dependence on Edge Evaluation and Acceptance

Universität Leipzig

Cameron Perot

September 2019

Abstract

In 2000 Dmitris Achlioptas asked a question about whether or not there existed processes in which adding edges to a random network would produce a phase transition later than that seen in the traditional Erdős-Rényi model. Little did he know that this question would have a simple yet complicated answer. Yes, there did exist such processes and yes they were able to show transitions later than that in the Erdős-Rényi model; however, the continuity of these transitions was not immediately clear. These processes in question were designed in such a way as to minimize the size of the largest cluster, and in doing so some of them delayed the phase transition to a later point at which it appeared to transition rapidly to the state of large scale connectivity, so rapid that it was brought into question whether the transition was discontinuous. Over the years there was a lot of research into this topic, eventually concluding that algorithms which take only local information into account produce continuous transitions whereas algorithms that make use of global information can produce discontinuous transitions. In this thesis we set out to summarize how the field developed as well as produce a new model which exhibits this rapid phase transition behavior. The new model explored here is called stochastic edge acceptance (SEA), and is a local information based algorithm that uses probabilities (favoring the minimization of the largest cluster) and random numbers to determine which edges to accept. Analysis of the SEA model performed here shows that it exhibits beautiful power law scaling behavior near the critical point, allowing us to determine critical exponents and show that it does indeed produce a continuous transition.

Contents

Chapter 1

Introduction

1.1 What is Percolation Theory?

Nature is full of graph-like structures which can be anything from a random network composed of nodes and undirected edges between them to a set of nodes in a periodic arrangement (i.e. a lattice). To study such structures we need a framework in which to do so; this is where percolation theory enters the picture. Percolation theory is in the intersection of multiple other topics such as graph theory, probability theory, and statistical physics, combining them in a way to give us the tools we need to study how clusters of nodes behave and evolve within graph-like structures. Percolation can be observed in a variety of places ranging from the spread of a forest fire to the dissemination of information across a communications network. We often use percolation theory in physics to help us model how something dissipates or flows through a system, e.g. the electrical conductivity or porousness of a material. Some of the applications outside of physics include testing the durability of a transportation infrastructure or determining to what extent a product or idea might be adopted by society. Percolation theory is a broad subject with the ability to be applied to problems in many different disciplines [1] [2].

To get a better understanding of what percolation is let's discuss a few examples. Society is developing faster than ever before due to the propagation of information and goods through human developed networks. These networks allow us to create and participate in increasingly more advanced networks, thus it is critical we study and understand how these networks behave so that we can better manage and improve them. The most well known being the internet, which sustains a massive flow of information between billions of people across the globe. When something is posted online we might ask: how many people will it reach? To answer this we would start at the source node then see which nodes it could possibly be connected to. If each person has a probability p of sharing it with another person then we can construct a probabilistic model to determine what the cluster could possibly look like. If the probability is high then we would expect the size of cluster to be significant when compared to that of the whole network.

Another interesting application is that to the spread of health epidemics across a population. If a virus is spreading, then there is an underlying network that can be studied; there is the initial person(s) who contracted the virus and then the people who were contaminated from coming in contact with those infected. The obvious question is: how much of the population will it affect? Let's say in a very basic

sense that an infected person has a p percent chance of infecting those they come into contact with. If the virus is treatable and p is low, then we would expect the virus to be eradicated fairly quickly since doctors have the time and ability to treat the outbreak properly. On the other hand if p is high, say higher than some critical probability p_c, then we might expect a different scenario where the virus spreads across the population, i.e. percolates, much like the Ebola epidemic in West Africa during 2013-2016. Therefore, percolation theory plays a crucial role in helping us answer important questions when it comes to determining how something might propagate.

1.2 Phase Transitions

To understand the most important use of percolation theory we must first understand what a phase transition is. In physics we often encounter systems which have different properties depending on which phase they're in. Thus we are interested in how the system behaves around the critical point. A phase transition is usually characterized by an order parameter, which is a variable that is zero in one phase and non-zero in the other; giving us the ability to distinguish phases and identify where the system undergoes transition.

In percolation theory we define the order parameter as the ratio of the largest cluster to the size of the system, i.e. if we have N nodes total and the largest cluster C contains $|C|$ nodes, then the order parameter m is given by:

$$m := \frac{|C|}{N} \tag{1.1}$$

Of course the system is usually examined in the thermodynamic limit ($N \to \infty$), this means that m is zero unless there exists a macroscopic cluster.

There are two main types of phase transitions: first- and second-order. In a first-order phase transition there is what we call a latent heat, i.e. energy that is either absorbed or emitted during a process where the temperature is fixed, e.g. the transition of water between the solid and liquid phases. Second-order (also called continuous) phase transitions are characterized by a divergent correlation length and power law behavior of variables which leads to a set of critical exponents (more about critical exponents below). The reason they are called continuous transitions is because the order parameter is continuous at the critical point. The classic percolation models are known to undergo continuous phase transitions, making them useful for modeling such systems.

1.2.1 Critical Exponents

Around the critical point in a percolation model there exists a set of numbers called the critical exponents that contain information about how certain quantities of interest behave. For example, let's take the order parameter as a function of the occupation probability p, which is described by the critical exponent β [1]:

$$m(p) \sim (p - p_c)^{\beta} \tag{1.2}$$

The average cluster size $\langle s \rangle$ is described by γ [1]:

$$\langle s \rangle(p) \sim (p - p_c)^{-\gamma} \tag{1.3}$$

The correlation length ξ is described by ν [1]:

$$\xi(p) \sim |p - p_c|^{-\nu} \tag{1.4}$$

The neat thing about these exponents is that they are universal in the sense that they only depend on the dimension d of the system, regardless of the microscopic configuration. Some values for these are given in the table below [1]:

d	β	γ	ν
2	5/36	43/18	4/3
3	0.41	1.82	0.88

1.3 Percolation on a Lattice

We can look at percolation on a lattice from two different perspectives: site and bond percolation. With site percolation we study systems where the sites of a lattice are either occupied or unoccupied, whereas with bond percolation we study systems where the edges between sites are either active or inactive. The underlying concepts of percolation are the same, but in practice the details differ slightly.

1.3.1 Site Percolation

For this we consider a lattice of dimension d. Let's take our lattice to be hyper-cubic with side length L, giving us a total of L^d sites in the lattice. We then occupy each site of the lattice with probability p (unoccupied with probability $1-p$), independent of all other sites. We can then use indicator random variables $X_1, ..., X_{L^d}$ to represent if the sites are occupied or not, where:

$$X_i = \begin{cases} 1 & \text{if site } i \text{ occupied} \\ 0 & \text{otherwise} \end{cases}$$

The number of occupied sites in the lattice is then given by $\sum_i X_i$, which is expected to be $L^d p$. The notion of a cluster $I = \{i_1, ..., i_s\}$ of size $s = \sum_{i \in I} X_i$ is defined as a set of s occupied sites which are nearest neighbors with at least one other site in the set. If we let N_s represent the number of clusters of size s, and $n_s = N_s / L^d$, then we can also look at the probability that any given site is part of a cluster:

$$\sum_s s n_s \tag{1.5}$$

This now gives us the ability to determine the probability that the given site is part of a cluster of size s:

$$\frac{s n_s}{\sum_{s'} s' n'_s} \tag{1.6}$$

Using the above we can compute the average cluster size:

$$\langle s \rangle = \sum_s s \frac{s n_s}{\sum_{s'} s' n_s'} \tag{1.7}$$

Therefore, it is clear that we can obtain a significant amount of information by applying basic probability theory. Now that we have defined clusters and cluster sizes we can talk about the idea of a percolating cluster. This is a cluster which spans from one side of the lattice to the opposite in any of the d dimensions. It's obvious that the presence of a percolating cluster is heavily dependent on p, since for low p we wouldn't expect to see a cluster spanning across the lattice. However, this is also dependent on L, and we often seek to study the limiting case where $L \to \infty$. This leads us to the concept of a critical probability, p_c, such that when $p > p_c$ we expect to see a percolating cluster, but not when $p < p_c$.

One-Dimension We now consider the simplest case where $d = 1$, which allows us to analytically solve for some quantities of interest. Fig. 1.1 illustrates a snippet of an infinite 1D lattice where the solid black sites are active and the hollow white ones are inactive. There are a total of four clusters visible (highlighted in blue): two one-clusters, a three-cluster, and a four-cluster.

Figure 1.1: 1D Site Clusters

For comparison Fig. 1.2 illustrates a 1D lattice where all sites are active and only some of the bonds between them are active. There are a total of four clusters visible (highlighted in blue): two two-clusters, a three-cluster, and a four-cluster.

Figure 1.2: 1D Bond Clusters

The probability that any given site is part of a cluster of size s is then given by the product of the independent probabilities of the site to the left of the leftmost site in the cluster being unoccupied, s consecutive sites being occupied, and the site to the right of the rightmost site in the cluster being unoccupied, i.e.:

$$n_s = (1-p) \cdot p^s \cdot (1-p) = p^s(1-p)^2 \tag{1.8}$$

Plugging Eq. 1.8 into Eq. 1.5 we find:

$$
\begin{aligned}
\sum_s n_s s &= (1-p)^2 \sum_s s p^s \\
&= (1-p)^2 p \frac{d}{dp} \sum_s p^s \\
&= (1-p)^2 p \frac{d}{dp} \frac{p}{1-p} \\
&= (1-p)^2 p \left[\frac{1}{1-p} + \frac{p}{(1-p)^2} \right] \\
&= p
\end{aligned}
\tag{1.9}
$$

Where the second equality is obtained by linearity of the derivative and summation operators and realizing that $s p^s = p \frac{d}{dp} p^s$, the third by noticing that $\sum_s p^s$ is the power series for $\frac{p}{1-p}$, the fourth by application of the derivative operator, and the final by algebaic manipulation. Therefore, there is a p percent chance that any given site is part of a cluster, which when thought about is fairly obvious. Using the above we can also analytically determine the average cluster size (again using the trick of differentiating p^s to bring down an s):

$$
\begin{aligned}
\langle s \rangle &= \sum_s \frac{s^2 n_s}{\sum_{s'} s' n_s'} \\
&= \frac{(1-p)^2}{p} \sum_s s^2 p^s \\
&= \frac{(1-p)^2}{p} \left[p \frac{d}{dp} \right]^2 \sum_s p^s \\
&= \frac{(1-p)^2}{p} \left[p \frac{d}{dp} \right]^2 \frac{p}{1-p} \\
&= \frac{(1-p)^2}{p} p \frac{d}{dp} \left[\frac{p}{1-p} + \frac{p^2}{(1-p)^2} \right] \\
&= \frac{1+p}{1-p}
\end{aligned}
\tag{1.10}
$$

The above illustrates that we can analytically solve for some of the quantities of interest in the one-dimensional case. We're often more interested in the thermodynamic limit, i.e. when $L \to \infty$. In the thermodynamic limit, percolation on the chain only occurs when we have an infinite number of neighboring occupied sites. This means that if just one site out of the entire chain is unoccupied there is no percolating cluster which leads us to conclude that for the one-dimensional case $p_c = 1$.

Multiple-Dimensions Unfortunately, analytical solutions to percolation problems are not as straight forward in higher dimensions due to the many different arrangements and shapes that the clusters can take on. This is illustrated in Fig.
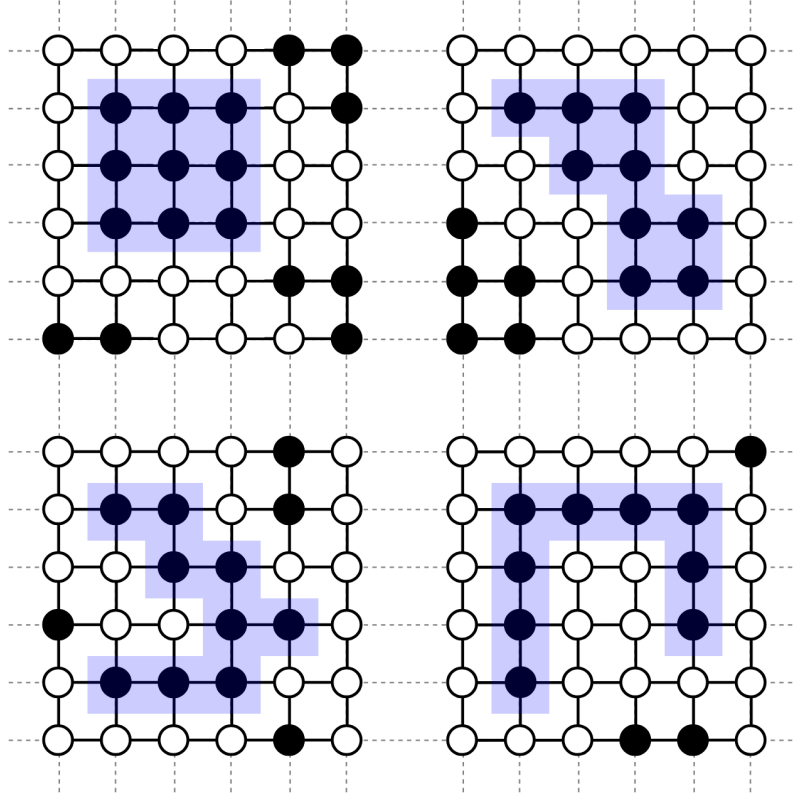
Figure 1.3: 2D Site Clusters

1.3 where we can see four possible arrangements of a nine-cluster, each with different numbers of neighboring inactive sites.

Without the ability to answer our questions through analytical methods we must resort to numerical methods and simulations to study the system. This is where computer simulations and finite-size scaling analysis come in handy. Due to the limitations of computational power we can only simulate systems up to certain sizes; however, we would still like to gather some information about the infinite size system. We can simulate systems of varying sizes and see how the quantities of interest scale with the system size, giving us the ability to extrapolate to the case of the infinite system.

Example: Ising Model The Ising model is the most well-known model of ferromagnetism, undergoing a second-order phase transition (for $d \geq 2$) from the ordered, ferromagnetic state at low temperatures to the disordered, paramagnetic state at high temperatures. This model has played a significant role in the development and understanding of statistical physics. The basic idea is that given a lattice of size L^d, at each site i of the lattice there is a spin σ_i oriented in either the up or down direction. The order parameter characterizing the transition is the magnetization $m = \frac{1}{L^d} \sum_i \sigma_i$. A (simplified) Hamiltonian of the system in the configuration $\{\sigma\}$ is then given by:

$$H(\{\sigma\}) = -J \sum_{\langle ij \rangle} \sigma_{ij} - h \sum_i \sigma_i \qquad (1.11)$$

6

Where the sum is over all neighboring spin pairs, J is the coupling constant between spins, and h is a mean field per spin approximation. Using this combined with statistical and computational physics methods we are able to simulate how this system behaves as the size increases. One useful tool when modeling such a system is the use of periodic boundary conditions, which help reduce the finite-size effects of smaller lattice sizes. In Fig. 1.4 we can see a percolating cluster of up (black) spins at the critical temperature $T_c = 2/\log(1 + \sqrt{2})$ on a 2D square lattice with side-length $L = 128$. Fig. 1.4 was generated using the Wolff single cluster algorithm with $5 \cdot 10^4$ update sweeps over the lattice.
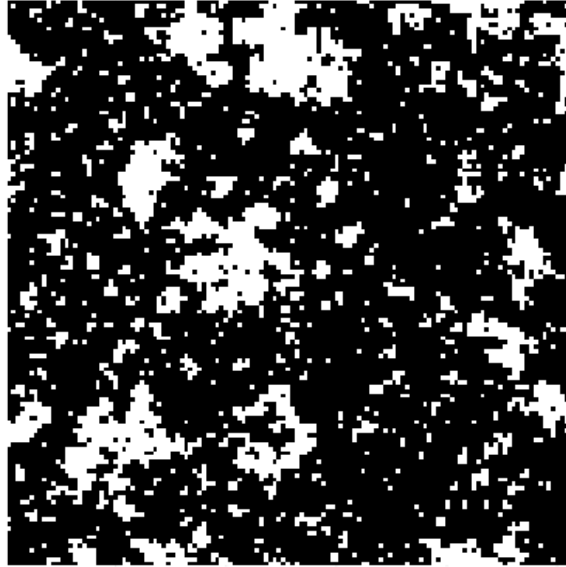

Figure 1.4: 2D Ising Model Percolating Cluster

1.4 Percolation on a Random Network

Percolation can also be observed on networks, so we will now discuss some of the basic ideas behind network percolation as a segue to the next section. In the late 1950s and early 1960s Paul Erdős and Alfréd Rényi published several papers on random networks, leading to the creation of what is known today as the Erdős-Rényi (ER) model [3] [4].

To get an idea of how it works we imagine a graph starting with a set of N disconnected nodes at $t = 0$, giving us $\binom{N}{2}$ possible connections between them. Then at each step in the evolution process two nodes are chosen at random and connected by an edge, thus after t steps there are a total t edges present in the network.

A cluster on a random network is a set of nodes connected by edges, either directly or indirectly. Being that some clusters in the graph will of course merge together over time, there exists an attractive potential of sorts between clusters corresponding to the likelihood that the clusters will merge together to form a larger one. To demonstrate this let's take three clusters C_1, C_2, and C_3 of sizes $|C_1|, |C_2|$, and $|C_3|$, with $|C_1| > |C_2|$ leaving $|C_3|$ arbitrary. Then we let Φ_1 (Φ_2) be the potential between the first (second) cluster and the third. In a network where each

connection is equally probable, the probability that the first or second cluster merges with the third is directly proportional to the size of the cluster because more nodes means more possible connections. Therefore, there is a higher probability that the first cluster rather than the second merges with the third and we conclude that $\Phi_1 > \Phi_2$.

Percolation occurs in a network when a giant (macroscopically large) cluster appears. The order parameter here is defined as the largest cluster size divided by the network size, i.e. $|C|/N$. Let N be the number of nodes, t be the number of edges present in the graph, and take $r = t/N$. If $r > 0.5$ then there will exist a percolating cluster within the graph, but not if $r < 0.5$ [4]. This tells us that $r_c = 0.5$ is the critical point where the system transitions from the disconnected state to the state of large scale connectivity. Letting $|C|$ be the size of the largest cluster, it can be shown that for $r < 0.5$ the largest cluster size scales as $C \sim \log N$, and for $r > 0.5$ $C \sim N$, and more specifically if $r \gtrsim 0.5$ one finds $C \approx (4r - 2)N$ [4]. This tells us that the ER model exhibits a continuous phase transition when the number of edges exceeds half the number of nodes in the graph. Looking at Fig. 1.5 we can see that the order parameter is zero up until $r = 0.5$ when it transitions to the state of large scale connectivity.
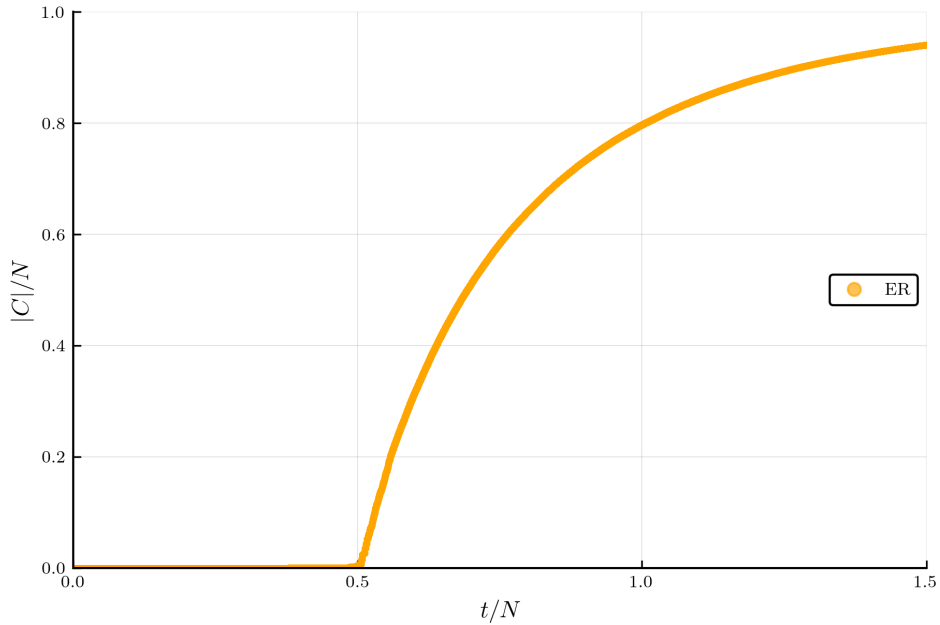

Figure 1.5: Erdős-Rényi model order parameter as a function of the number of edges divided by the system size. It transitions phases right at $r = 0.5$ as has been proven [4]. $N = 10^6$.

Chapter 2

An "Explosive" Transition

In this paper we are going to take a deeper look into a specific type of percolation referred to as explosive percolation, where the underlying concept is that the onset of percolation is delayed until a certain point where it then occurs at an accelerated rate. This happens when the underlying evolution process works in such a way that the largest cluster size $|C|$ is controlled. We can think of a graph where multiple clusters might evolve separately without merging. Collectively they take up a large portion of the graph but there isn't a percolating cluster yet due to the lack of connections between the them. If at some point these clusters do begin to connect then the graph transitions to the percolating state. For certain evolution processes it has been heavily debated whether or not the phase transition is continuous or not, so the aim of the next section is to summarize what we know to this point regarding the nature of the transition for said processes.

2.1   A Selective Process

This all started in the year 2000 when Dimitris Achlioptas raised an interesting question about how a graph of N nodes would evolve under certain conditions [5]. First we suppose that at each step t in the evolution process two edges e_t and e'_t are evaluated, but only one of them is chosen to be added to the graph. Which of the proposed edges is chosen is based only on the information contained in $e_1, e'_1, ...e_{t-1}, e'_{t-1}$. Does there exist an algorithm for adding edges to the graph such that with high probability a giant component does not appear until $t/N > 0.5$? This process of evaluating $q \geq 2$ edges at each step is now referred to as an q-edge Achlioptas process. In the next few sections we will see how a small change to the way edges are added can have a big impact on the evolution of the system.

2.1.1   The First Look

In 2001 Tom Bohman and Alan Frieze were the first to design and analyze an Achlioptas process in their paper "Avoiding a Giant Component" [5], where they set out to answer Achlioptas' question. This paper laid out the framework for the Bohman-Frieze (BF) model and showed that there does exist a process in which the appearance of a giant cluster is delayed; so the answer to Achlioptas' question is yes! This can be seen in Fig. 2.1 where the order parameter for the BF model remains
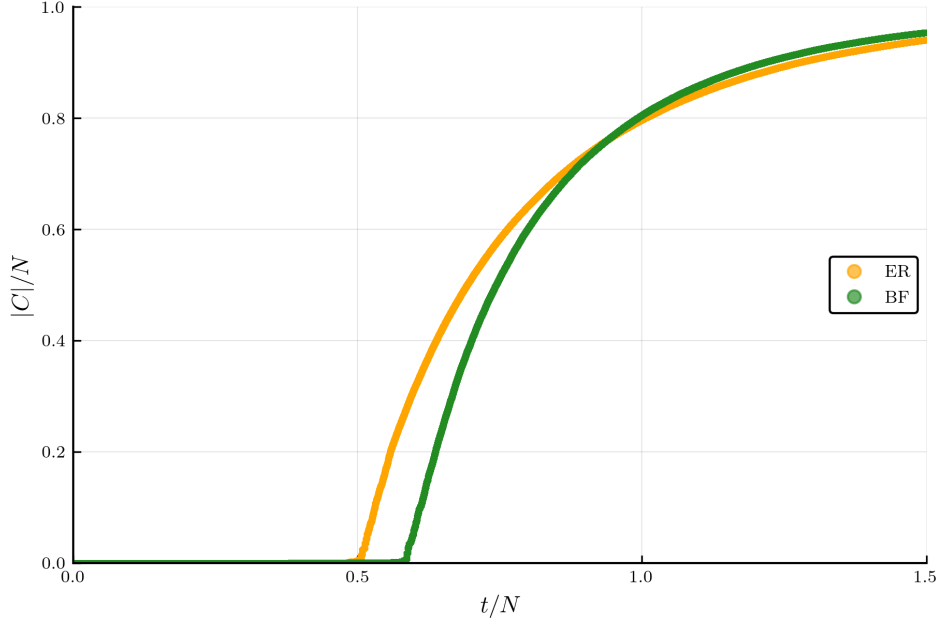
Figure 2.1: Bohman-Frieze model order parameter compared to that of the ER model. It is clear that the transition is delayed to a point beyond 0.5. $N = 10^6$.

smaller than for the ER model, but a little after $r = 0.5$ it begins to rise at a faster rate than in the ER model, eventually overtaking the ER model.

The algorithm they came up with randomly/uniformly samples and evaluates two edges e_t and e'_t at each step t, adding e_t if it connects two isolated nodes and discarding e'_t, otherwise adding e'_t and discarding e_t. This is known as a bounded-size rule which has since been generalized to a K bounded-size rule where e_t is added if it connects clusters of size smaller than K, otherwise e'_t is added. What this process does is equally discriminate against clusters of size greater than or equal to K. It has been hypothesized that all bounded-size rules exhibit continuous phase transitions [6].

A form of the algorithm can be written in pseudocode as follows:

- Let T be the total number of edges to add to the graph.

- Let $A_t = \{e_1, e_2, ..., e_t\}$ be the set of accepted edges at step t.

- Let e_t^1 and e_t^2 be the two nodes which edge e_t would connect.

- Let $C(e_t^i)$ be the cluster which e_t^i belongs to.

10

Algorithm 1 Bohman-Frieze

1: **procedure** BF($T, K = 2$)
2:     $A \leftarrow \emptyset$
3:     $t \leftarrow 1$
4:     **while** $t \leq T$ **do**
5:         **if** $|C(e_t^1)| < K$ and $|C(e_t^2)| < K$ **then**
6:             $A \leftarrow A \cup \{e_t\}$
7:         **else**
8:             $A \leftarrow A \cup \{e_t'\}$
9:         **end if**
10:       $t \leftarrow t + 1$
11:     **end while**
12: **end procedure**

A few years after their original paper, Bohman and Frieze published another paper with Nicholas Wormald titled "Avoidance of a giant component in half the edge set of a random graph" [7]. In this paper they designed a new algorithm (BFW) for adding edges to a graph that runs in phases beginning at $k = 2$. Starting with N isolated nodes, edges are randomly/uniformly sampled and evaluated one at a time. If the edge being evaluated would create a cluster of size less than or equal to k then the edge is accepted, otherwise the edge is rejected. If edge is rejected and the ratio of accepted edges is less than the function $g(k) = 1/2 + \sqrt{1/(2k)}$, then the algorithm moves to the next phase.

A form of the algorithm can be written in pseudocode as follows:

- Let T be the total number of edges to add to the graph.

- Let $A_i = \{e_1, e_2, ..., e_i\}$ be the set of accepted edges at step i.

- Let $C(A)$ be the largest cluster in A.

Algorithm 2 Bohman-Frieze-Wormald

1: **procedure** BFW(T)
2:     $A \leftarrow \emptyset$
3:     $k \leftarrow 2$
4:     $t \leftarrow 1$
5:     $i \leftarrow 1$
6:     **while** $t \leq T$ **do**
7:         **if** $|C(A \cup \{e_i\})| \leq k$ **then**
8:             $A \leftarrow A \cup \{e_i\}$
9:             $t \leftarrow t + 1$
10:           $i \leftarrow i + 1$
11:         **else if** $|A|/i < g(k)$ **then**
12:           $k \leftarrow k + 1$
13:         **else**
14:           $i \leftarrow i + 1$
15:         **end if**
16:     **end while**
17: **end procedure**

2.2 A Discontinuous Transition?

The first mention of explosive percolation appeared in 2009 in the paper "Explosive Percolation in Random Networks" [8] by Dimitris Achlioptas, Raissa M. D'Souza, and Joel Spencer, which will henceforth be referred to as Achlioptas et al. In this paper they laid out a method of evaluating candidate edges called the product rule (PR). At each step t in the evolution process two edges are selected randomly/uniformly and evaluated based on the product of the cluster sizes corresponding to the clusters which the nodes that the edges would connect belong to. More specifically, if edge e_t connects clusters $C(e_t^1)$ and $C(e_t^2)$ and edge e_t' connects clusters $C(e_t'^1)$ and $C(e_t'^2)$, then e_t is accepted if $|C(e_t^1)| \cdot |C(e_t^2)| < |C(e_t'^1)| \cdot |C(e_t'^2)|$, otherwise e_t' is accepted.

A form of the algorithm can be written in pseudocode as follows:

- Let T be the total number of edges to add to the graph.

- Let $A_t = \{e_1, e_2, ..., e_t\}$ be the set of accepted edges at step t.

- Let e_t^1 and e_t^2 be the two nodes which edge e_t would connect.

- Let $C(e_t^i)$ be the cluster which e_t^i belongs to.

Algorithm 3 Product Rule

1: **procedure** PR(T)
2:     $A \leftarrow \emptyset$
3:     $t \leftarrow 1$
4:     **while** $t \leq T$ **do**
5:         **if** $|C(e_t^1)| \cdot |C(e_t^2)| < |C(e_t'^1)| \cdot |C(e_t'^2)|$ **then**
6:             $A \leftarrow A \cup \{e_t\}$
7:             $t \leftarrow t + 1$
8:         **else**
9:             $A \leftarrow A \cup \{e_t'\}$
10:             $t \leftarrow t + 1$
11:         **end if**
12:     **end while**
13: **end procedure**

This is illustrated in Fig. 2.2 where the first proposed edge e_t would connect the blue cluster of size 5 to the red cluster of size 3 and the second proposed edge e_t' would connect the orange cluster of size 2 to the green cluster of size 4. The product corresponding to e_t is $5 \cdot 3 = 15$ whereas the product corresponding to e_t' is $2 \cdot 4 = 8$, so e_t is rejected and e_t' accepted. If we were looking at Fig. 2.2 from within the framework of the BF model with $K = 1$, then e_t would be rejected and e_t' accepted even though both the orange and green clusters are of size greater than K.
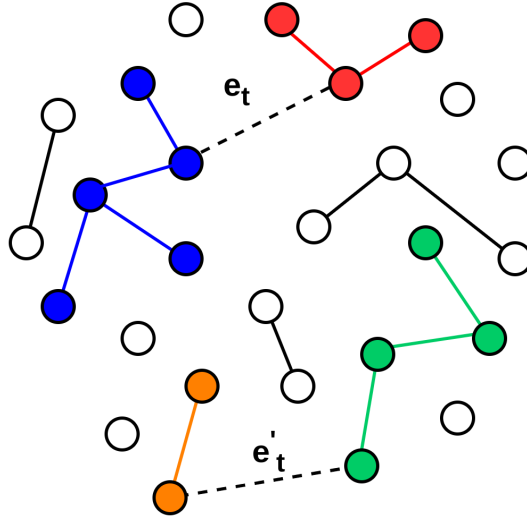
Figure 2.2: Edge Selection

Fig. 2.3 shows the order parameters for the ER, BF, and PR models, making it clear just how drastically different the PR model transition is than in the other models. As can be seen the order parameter remains near zero for much longer until the critical point $r_c \approx 0.888...$, which is much later than the critical point in the ER model of $r_c = 0.5$. Not long after the critical point is reached the order parameter appears to jump up discontinuously, which was a surprising find.
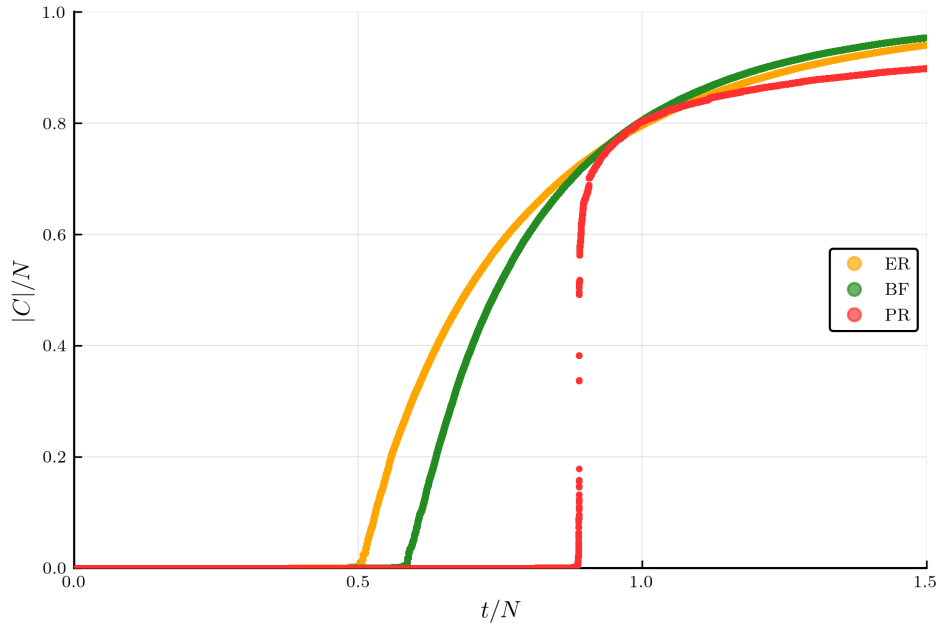

Figure 2.3: Product rule model order parameter plotted with the ER and BF models for comparison. The transition is much more delayed than that of BF, and has an almost vertical slope to the order parameter at the critical point. $N = 10^6$.

Achlioptas et al. analyzed the scaling behavior of their model, defining the quantity $\Delta := t_1 - t_0$, where t_0 is the last step where $C < N^{1/2}$ and t_1 is the first step where $C > 0.5N$. Δ is linear in N for continuous transitions (i.e. an extensive quantity); however, their analysis indicated otherwise that $\Delta < 2N^{2/3}$

and $\Delta/N^{2/3} \to 1$ in their simulations for sizes up to $6.4 \cdot 10^7$. To arrive at these results they took an ensemble average of 50 different independent and identically distributed observations and fit a curve to determine the relation between Δ and N. Their findings are illustrated in Fig. 2.4.

Fig. 2. (**A**) The ratio Δ/n for ER and BF for increasing system sizes. (**B**) The ratio $\Delta/n^{2/3}$ for PR for increasing system sizes. (**C**) Convergence to $r_c = 0.888\ldots$ from above and below (the two curves fitted independently). (**D**) A linear scaling relation is obeyed in the range $\gamma \in [0.2, 0.6]$, shown here for $A = 0.5$. Color shows convergence with increasing system size n to the relation $\gamma + 1.2\beta = 1.3$. Our numerical experiments establish this scaling relation for $A \in [0.1, 0.6]$ and we expect that in larger system sizes this range of A would broaden, particularly the lower end.
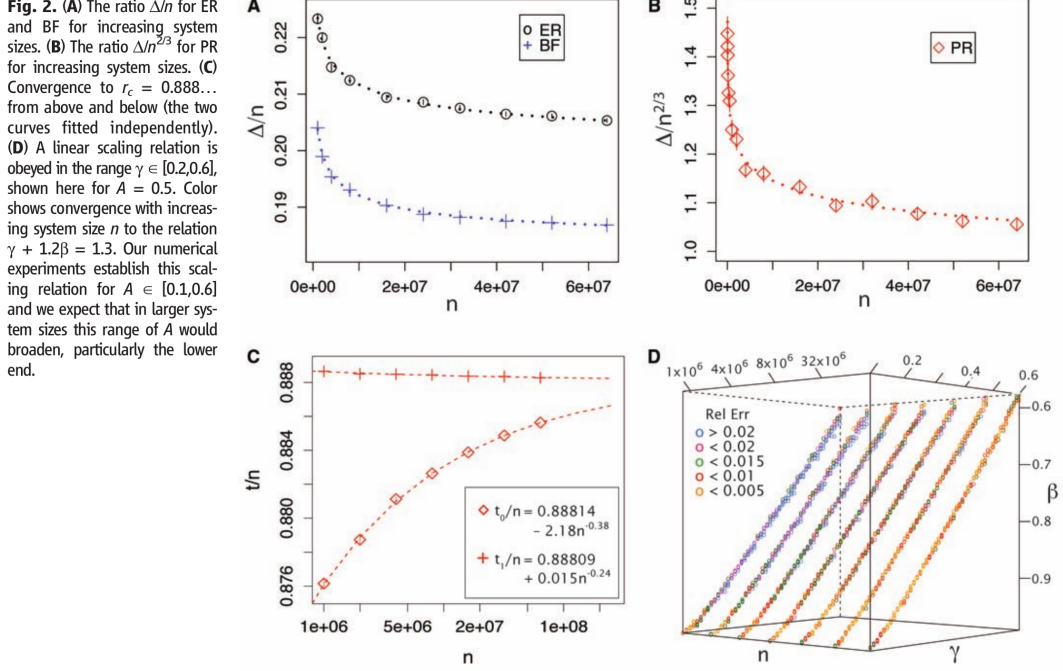
Figure 2.4: Achlioptas et al. simulation results. Source: [8].

Achlioptas et al. ended their paper by saying: "We have demonstrated that small changes in edge formation have the ability to fundamentally alter the nature of percolation transitions. Our findings call for the comprehensive study of this phenomenon, and of its potential use in bringing phase transitions under control." Indeed their message seemed to resonate, leading to a wave of papers from around the world exploring the subject [9–65].

2.2.1 No, It's Actually Continuous!

In 2010 Rui da Costa, Sergey Dorogovtsev, Alexander Goltsev, and Jose Fernando Mendes published [16] wherein they dispute that the explosive percolation transition is actually continuous. They used an edge selection rule which at each step chooses two sets of q nodes, determines for each set the node belonging to the smallest cluster, then activates the edge between those two thus favoring the merging of smaller clusters and minimizing the largest cluster in the process. They were able to show power law behavior of the largest cluster size for $m = 2$:

$$|C| \propto \delta^\beta \tag{2.1}$$

where $\delta = |r - r_c|$, with $t_c = 0.923207508(2)$ and $\beta = 0.0555(1)$ which is close to $1/18 = 0.055\overline{5}$.

They then show that if the cluster size distribution is power law distributed at the critical point then the transition must be continuous. Their paper concludes by

saying the reason it is difficult to distinguish this transition from a discontinuous one is due to the small value of β, and that the irreversible nature of the process makes it is an attractive subject to further develop.

2.2.2 Well, Maybe Not...

In their paper "Explosive Percolation with Multiple Giant Components" [29], Wei Chen and Raissa M. D'Souza studied a modified version of the BFW model mentioned above. They took the parameter α which is the asymptotic fraction of accepted edges and demonstrated that changing α determines the number of stable giant clusters. The smaller the value of α the longer the transition will be delayed and the more explosive it will be. This is shown in Fig. 2.5
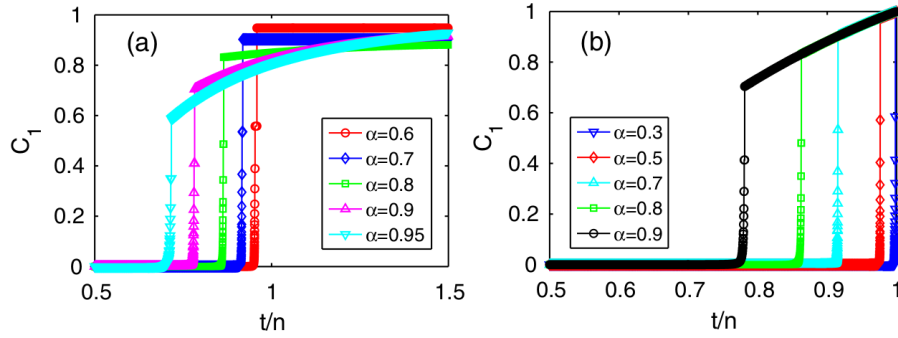

FIG. 4 (color online). The results of varying α on (a) the unrestricted BFW process in the regime where only one giant component emerges, and (b) the restricted BFW process. Both (a) and (b) show increasing delay and larger $\Delta C_{\max}$ for smaller α.

Figure 2.5: Phase transition dependence on α. Source: [29].

Their analysis shows that it is possible to have multiple stable giant components emerge at the same critical point. Furthermore they finish their paper by stating that the BFW model is strongly discontinuous.

2.2.3 Continuity With Unusual Finite-Size Scaling Behavior

In 2011 Grassberger et al. published "Explosive Percolation is Continuous, but with Unusual Finite Size Behavior" [24] in which they studied the phase transitions of four different Achlioptas processes. They specifically looked at the distribution $P_{r,N}(m)$ of the order parameter $m = |C|/N$, and at the critical point it scales as:

$$P_{r=r_c,N}(m) \sim N^\eta f(mN^\eta) \tag{2.2}$$

In the Ising model (continuous transition) the function $f(z)$ has two peaks and as $N \to \infty$ the peaks move closer together to form a single peak, whereas in a first order transition $f(z)$ is double peaked and the peaks move further apart as $N \to \infty$. The finite size scaling of $\langle m \rangle$ is given by:

$$\langle m \rangle \sim (r - r_c)^\beta g((r - r_c)N^\Theta) \tag{2.3}$$

15

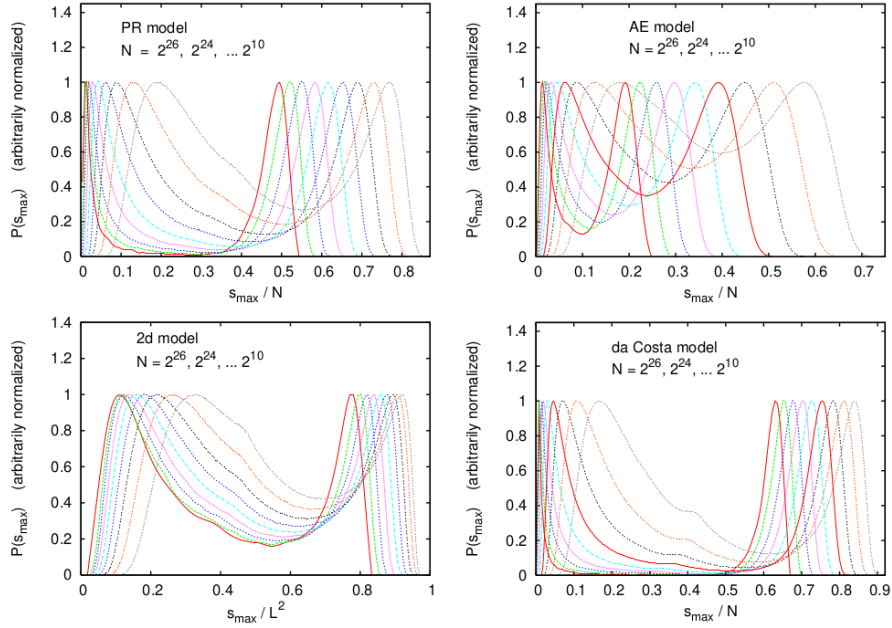$g(z)$ is analytic for all $z \in \mathbb{R}$.

FIG. 1. (Color online) Distributions of the order parameter $s_{\max}/N$ for four EP models. They are shown at the effective critical point, defined such that both peaks have the same height. Normalization is such that their height is 1. For the largest systems, curves were approximated by cubic splines to make them smooth.

Figure 2.6: Scaling of $P_{r=r_c,N}(m)$. Source: [24].

Their results shown in Fig. 2.6 showed that $P_{r=r_c,N}(m)$ was indeed double peaked; however, they say that this does not prove it is continuous, rather it reflects how abruptly the system transitions. They then go on to show (through the use of FSS of $\langle m \rangle$) that the transitions of the studied four models are indeed continuous, yet are each in a different universality class and all exhibit unusual finite size scaling behavior.

2.2.4 Large Scale Simulations Show It's Continuous

The 2011 paper "Continuity of the Explosive Percolation Transition" [25] by Hyun Keun Lee, Beom Jun Kim, and Hyunggyu Park analyzed the finite-size scaling of the product rule in depth, using data from simulations of system sizes up to $2^{37} \approx 1.4 \cdot 10^{11}$. They use a fairly straightforward method for determining the continuity of the transition:

1. Set $t_0(N) < t_c$ and $t_1(N) > t_c$ with the expectation that as $N \to \infty$, $t_0 \to t_c$ and $t_1 \to t_c$.

2. Find an upper bound for the change in the largest cluster size $\Delta |C|$.

3. Show that $\Delta |C|$ scales sublinearly with N.

16

Near the end they arrive at:

$$\Delta m = \frac{\Delta |C|}{N} \lesssim N^{-0.022} \tag{2.4}$$

Which means that as $N \to \infty$, $\Delta m \to 0$ and they conclude that the transition is indeed continuous.

2.2.5 Proof of Continuity Emerges

Oliver Riordan and Lutz Warnke published "Achlioptas Process Phase Transitions Are Continuous" [26] in 2012, wherein they lay out several theorems and proofs to show that all Achlioptas processes result in a continuous phase transition. Due to the long and detailed nature of the proofs the reader is referred to their work for further information.

2.2.6 "Solution of the explosive percolation quest"

In 2014 and 2015 Rui da Costa, Sergey Dorogovtsev, Alexander Goltsev, and Jose Fernando Mendes published three more papers [50, 53, 54] where they performed a significant amount of research on the explosive percolation transition of the q-edge model laid out in their previous paper (see above 2.2.1).

In "Critical exponents of the explosive percolation transition" [53] they study extensively the critical exponents and how they scale as q changes, wherein they observe that as q increases β quickly declines, with $\beta(q = 4) \approx \beta(q = 2)/20$. They talk about how models that only use local information, i.e. they do not require knowledge of all of the clusters within the graph but only those corresponding to the candidate edges, lead to a continuous transition, but models that take global information into account might lead to discontinuities.

In "Solution of the explosive percolation quest: Scaling functions and critical exponents" [50] and "Solution of the explosive percolation quest. II. Infinite-order transition produced by the initial distributions of clusters" [54] they develop a thorough theory of finite-size scaling that explains the continuity of the transition. Their theory is quite extensive and worth the read.

2.2.7 A Look At the Developments

After seeing a number of papers stating that the transition is actually continuous, it looked like the debate had been settled. q-edge Achlioptas processes are always continuous if they only take local information into account, but there exist some processes that take global information into account that lead to a discontinuous transition. A lot of different rules for selecting, evaluating, and adding edges were created since Dimitris Achlioptas raised the question. In their 2015 paper "Explosive Percolation: Novel critical and supercritical phenomena" [56], Raissa M. D'Souza and Jan Nagler summarized a lot of the developments and findings in the field up to that point. They discussed similarities and differences between some of the well known explosive percolation models and tied it all together nicely with some real world applications and future directions for the field. Overall it is an excellent paper to read to get acquainted with how the field developed from 2009 to 2015.

Chapter 3

A New Model

3.1 Stochastic Edge Acceptance

After reading a lot about other models for evolving graphs I decided to come up with a new model for adding edges with an extra element of randomness. The model is called Stochastic Edge Acceptance or SEA for short. In most models the set of edges selected for evaluation are random but, but choosing which edge to add from that set is deterministic, like in the product rule where the one with the smallest product of cluster sizes is always accepted.

   The underlying idea is to have a process which favors the minimization of the largest cluster through considering multiple edges at each step in the evolution algorithm, making it an Achlioptas process. This is a local rule, i.e. it only considers local information such as the size of the clusters to which the nodes corresponding to the considered edges belong. I wanted to create a simple probability using only the sizes of the clusters corresponding to the considered edges. Using just the cluster size information it is easy to create a set of probabilities that decrease as the cluster size increases. The way it is designed actually allows us to generalize this algorithm to evaluate q edges at once with corresponding probabilities p_i, $i \in \{1, 2, ..., q\}$:

$$p_i = \frac{\frac{1}{|C_i|}}{\sum_j \frac{1}{|C_j|}} \tag{3.1}$$

A form of the algorithm for $q = 2$ can be written in pseudocode as follows:

- Let T be the total number of edges to add to the graph.

- Let $A_t = \{e_1, e_2, ..., e_t\}$ be the set of accepted edges at step t.

- Let e_t^1 and e_t^2 be the two nodes which edge e_t would connect.

- Let $C(e_t^i)$ be the cluster which e_t^i belongs to.

- Let $C(e_t) = C(e_t^1) \cup C(e_t^2)$ be the cluster which would be formed by activating e_t.

- Let $R \in [0, 1)$ be a randomly generated number.

Algorithm 4 Stochastic Edge Acceptance

```
 1: procedure SEA(T)
 2:     A ← ∅
 3:     t ← 1
 4:     while t ≤ T do
 5:         if C(e_t^1) = C(e_t^2) then
 6:             A ← A ∪ {e_t}
 7:             t ← t + 1
 8:         else if C(e_t'^1) = C(e_t'^2) then
 9:             A ← A ∪ {e_t'}
10:             t ← t + 1
11:         else if p > R then
12:             A ← A ∪ {e_t}
13:             t ← t + 1
14:         else
15:             A ← A ∪ {e_t'}
16:             t ← t + 1
17:         end if
18:     end while
19: end procedure
```

The q-edge algorithm is similar except it would use a cumulative probability $P_i = \sum_{j=1}^{i} p_j$ (much like in the q-state Potts model) and loop over $i \in \{1, 2, ..., q\}$ until $P_i > R$, and then once that is satisfied accept the edge and move to the next step.

3.2   Simulation and Analysis

All data in this section was generated using `GraphEvolve.jl` (see Appendix A for more information). Random networks of sizes 2^k, $k \in \{15, 16, ..., 24\}$ were studied, and for each system size 1000 simulations were run with different random number generator seed values. It would be ideal to run simulations for larger system sizes to get a more representative sample, and is something which can be investigated further in the future. All code and the associated generated data used for this is publicly available on the corresponding GitHub page `https://github.com/cameronperot/explosive-percolation`.

Fig. 3.1 illustrates the order parameter of the SEA model in comparison with the ER, BF, and PR models. The SEA model experiences a transition later than the ER and BF models, but before the PR model. The order parameter shows a sharp increase at the point of phase transition with an almost vertical slope similar to that of the PR.
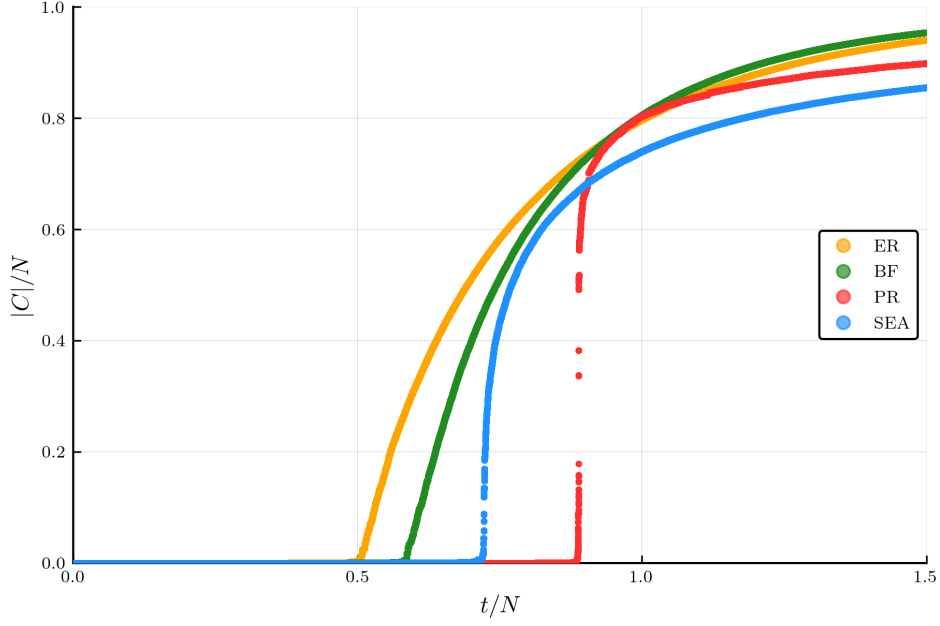
Figure 3.1: Comparison of the stochastic edge acceptance order parameter to that of well-known models ER, BF, and PR. $N = 10^6$.
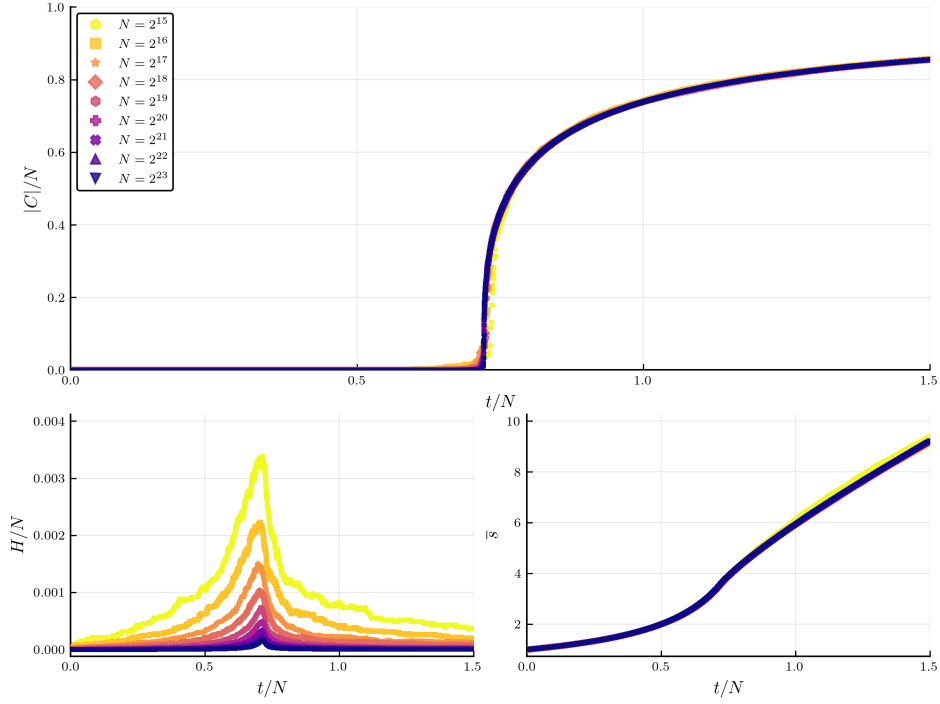

Figure 3.2: Observable dependence on system size. H is the cluster heterogeneity and $\overline{s}$ is the average cluster size. As the system size increases the peak of the heterogeneity per node H/N decreases and moves towards the right. The average cluster size lines up for all system sizes.

Now that we've seen it exhibits interesting behavior at the point of phase transition we would like to learn more such as where does it transition, and most importantly, is it continuous? In order to answer these questions we will use the framework laid out in [25]. First we need to find two points, one before the transition (t_0) and

one after the transition (t_1). We set t_0 as the step at which the cluster heterogeneity peaks, i.e. when the number of unique cluster sizes is at its highest. We set t_1 as the step after the order parameter experiences its largest increase. Let $r_i = t_i/N$ represent the relative number of edges in the system at step t_i (because at step t_i there are t_i edges present in the graph), and $m_i = m(r_i)$ be the order parameter at step t_i. We will define two quantities which we would like to study the behavior of:

$$\Delta r = r_1 - r_0$$
$$\Delta m = m_1 - m_0 \tag{3.2}$$

Fig. 3.3 shows the scaling behavior of r_0 and r_1 as a function of the system size N. As the system size increases the two curves appear to be converging to the exact point of phase transition. Using the ensemble averaged data for the system size $N = 2^{24}$ we find the critical point to be:
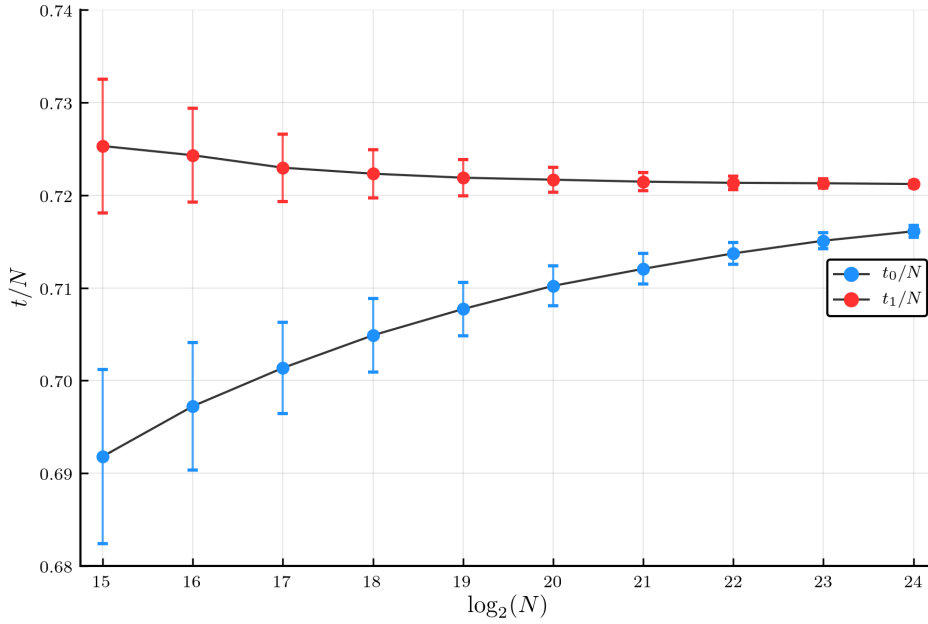
$$r_c \approx 0.719(3) \tag{3.3}$$

Figure 3.3: The curves for r_0 and r_1 appear to be converging to the point of phase transition. Error bars represent 1 standard deviation.

When looking at the scaling behavior of certain quantities it often helps to view them on $\log - \log$ plots to identify any power law scaling behavior. Fig. 3.4 shows Δr as a function of the system size, and as we can see it exhibits power law scaling behavior of the form:

$$\Delta r \sim N^{-\delta} \tag{3.4}$$

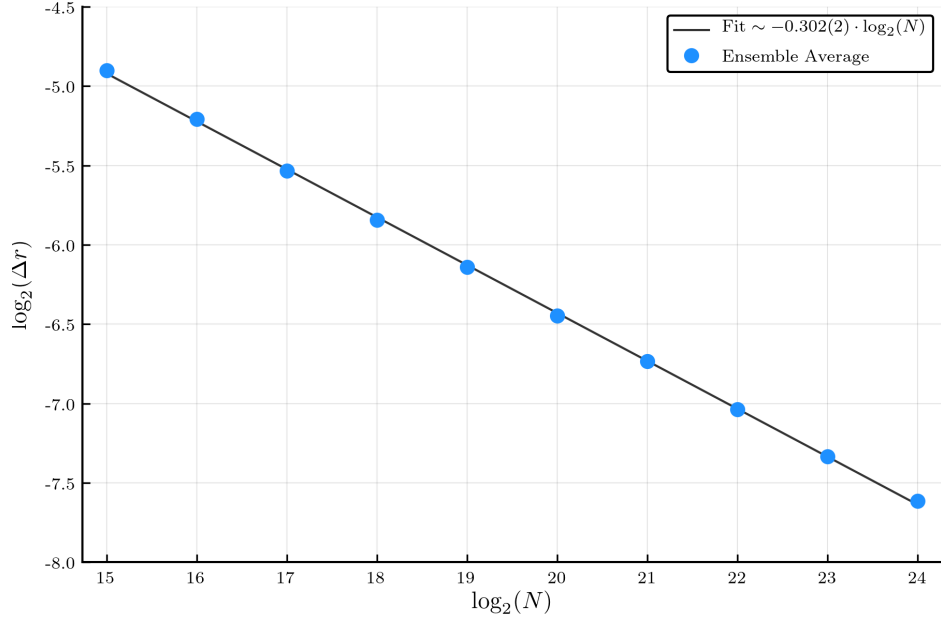with the fitted value:

$$\delta \approx 0.302(2) \tag{3.5}$$

Figure 3.4: Δr exhibits power law scaling in N.

There is a lot of information contained in the cluster size distributions at t_0 and t_1. We first start by analyzing the distribution at t_0, shown in Fig. 3.5.
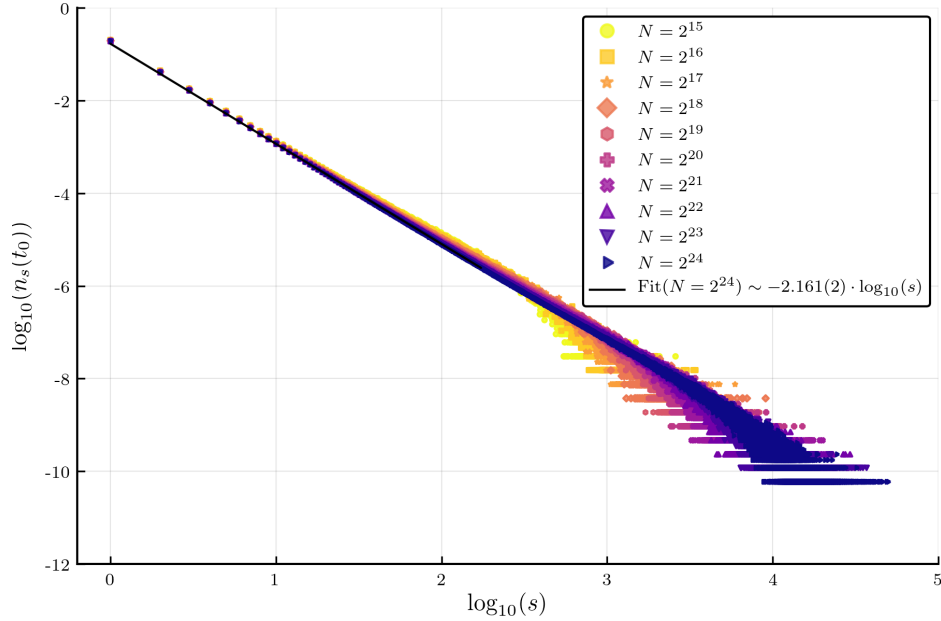

Figure 3.5: Scaling behavior of the cluster size distribution at t_0.

When plotted on a $\log_{10} - \log_{10}$ basis we see a linear relationship up to a certain cutoff cluster size s_f where $n_s(t_0)$ then drops off exponentially. Large statistical errors are visible for higher values of s, but the overall message of the plot is clear. This linear behavior tells us this is a power law in N and using the ansatz:

$$n_s(t_0) \sim s^{-\tau} \tag{3.6}$$

Performing a fit on the linear portion of the data for the system size $N = 2^{24}$ gives us the value of the critical exponent:

$$\tau_0 \approx 2.161(2) \tag{3.7}$$

Looking at the data for the cluster size distribution at t_1 (Fig. 3.6) we see a similar power law scaling relation, except with a slightly different exponent (also obtained from fitting data for the system size $N = 2^{24}$):

$$\tau_1 \approx 2.213(1) \tag{3.8}$$

I believe the discrepancy in the exponents here is due to the size of the simulations, and that if simulated with larger system sizes (such as up to 2^{37} seen in [25]) that the values would converge towards the true value of τ. In Fig. 3.6 we see $\log_{10}(n_s(t_1))$ versus $\log_{10}(s/N)$ (rather than just $\log(s)$ as in Fig. 3.5), where we again observe this linear scaling up to a point where it then dips and breaks the linear trend (again large statistical errors present past the cutoff point).
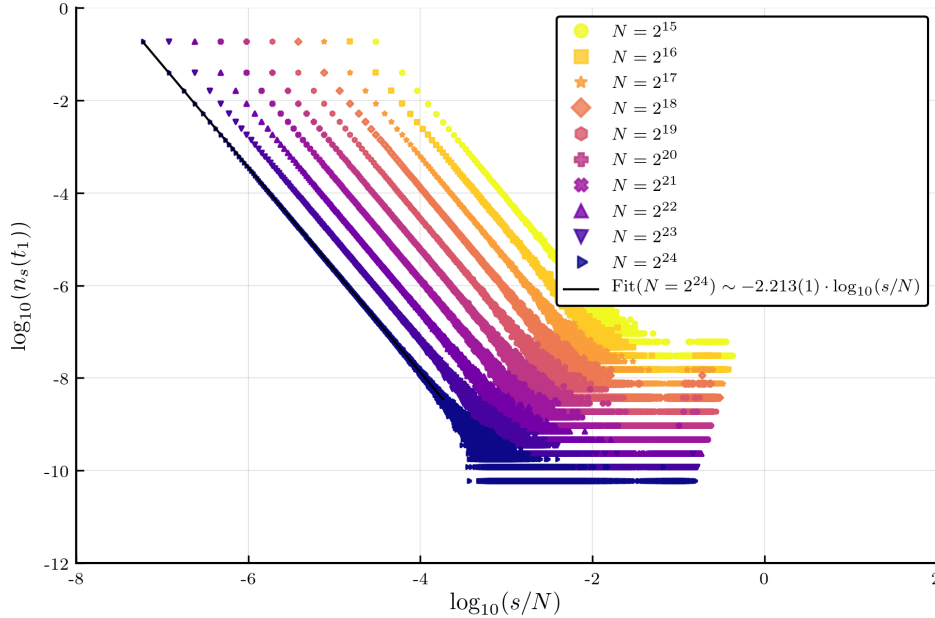

Figure 3.6: Scaling behavior of the cluster size distribution at t_1.

In [25] they determine s_f by taking the ansatz $\sum_{s \geq s_f} \sim 1/N_{C,0}$ where $N_{C,0}$ is the total number of clusters at t_0. Then through the use of several relations such as the number of clusters scaling linearly with the system size they arrive at:

$$s_f(N) \sim N^{1/\tau} \tag{3.9}$$

This can be verified by assuming a single characteristic cluster size and taking the finite-size scaling form of n_s [25]:

$$n_s(t_0) = s^{-\tau} f(s/s_f) = s^{-\tau} f(sN^{-1/\tau}) \tag{3.10}$$

Applying this same ideology to the stochastic edge acceptance method works quite well as seen in Fig. 3.7, illustrating that the data for all of the system sizes lines up well and exhibits this hump behavior also seen in [25]. This verifies that

23

fitted value of τ_1 is a good estimation of τ. The data analysis up to this point leads me to believe that the discrepancies in the fitted values of τ could be due to t_1 being closer to the critical point than t_0, and as N increases we see t_0 move closer to t_1 (Fig. 3.3), thus if we ran simulations with larger values of N then we should be able to obtain a more accurate reading of τ.
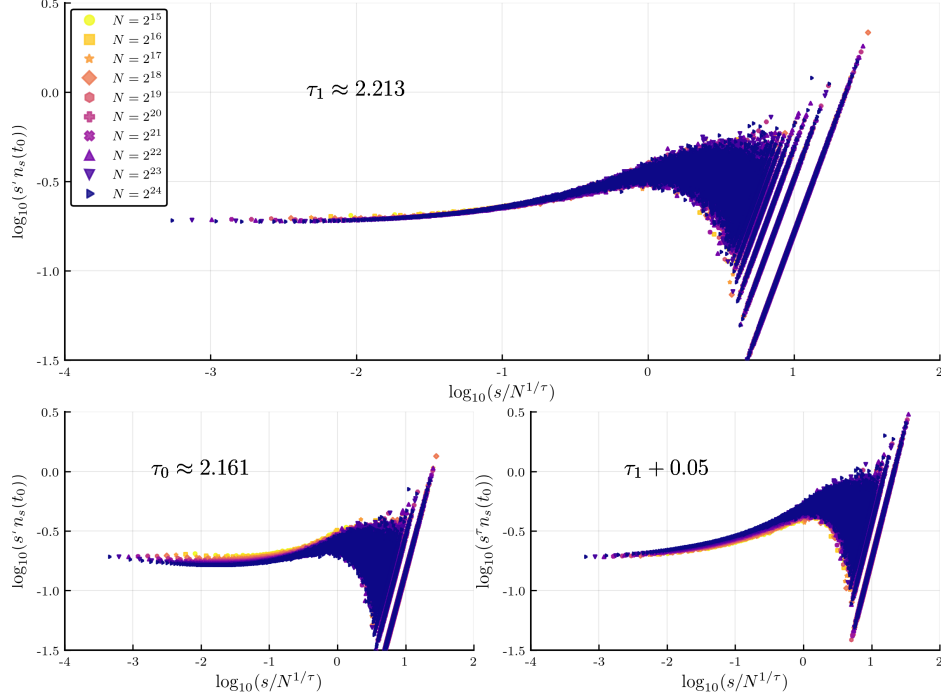

Figure 3.7: $n_s(t_0)$ finite-size scaling shows a collapse of the data for the critical exponent value $\tau_1 \approx 2.213$ obtained from fitting the linear portion of the distribution at t_1. The lower left plot shows how the collapse looks for the value τ_0 and the lower right plot illustrates a slightly higher value of τ, both of which do not line up as well as when using τ_1.

Now that we have extracted some useful information, namely the values of the critical exponents, we want to show that as the system size tends to infinity, the difference in the order parameter, Δm, tends to zero (i.e. $\lim_{N\to\infty} \Delta m = 0$). In order to show this we want to place an upper bound on Δm. In [25] they laid out a clever method for setting an upper bound by starting with the cluster size distribution at t_0 and taking an ideal process to add edges in such a way as to maximize the largest cluster size from t_0 to t_1, thus any other process will produce a Δm that is bounded above by the Δm produced in the ideal process. This ideal process would seek to merge all clusters larger than some size s_δ, which is determined by $\Delta t = t_1 - t_0$ and the cluster size distribution at t_0, i.e. $\Delta t = N_{C,0} \sum_{s>s_\delta} n_s(t_0)$, where $N_{C,0}$ is the total number of clusters at step t_0. What they arrived at is that [25]:

$$\Delta m \lesssim s_\delta^{2-\tau} \sim N^{-\delta(\tau-2)/(\tau-1)} \tag{3.11}$$

Plugging in our obtained values of δ and τ we find that:

$$\Delta m \lesssim N^{-0.053} \tag{3.12}$$

which implies that the transition of the stochastic edge acceptance model is continuous like other local information based models. Fig. 3.8 shows $\log_2(\Delta m)$ versus $\log_2(N)$ and fitting a line we observe a decay exponent of $0.152(1)$ and verify that this is indeed below the upper bound we arrived at.
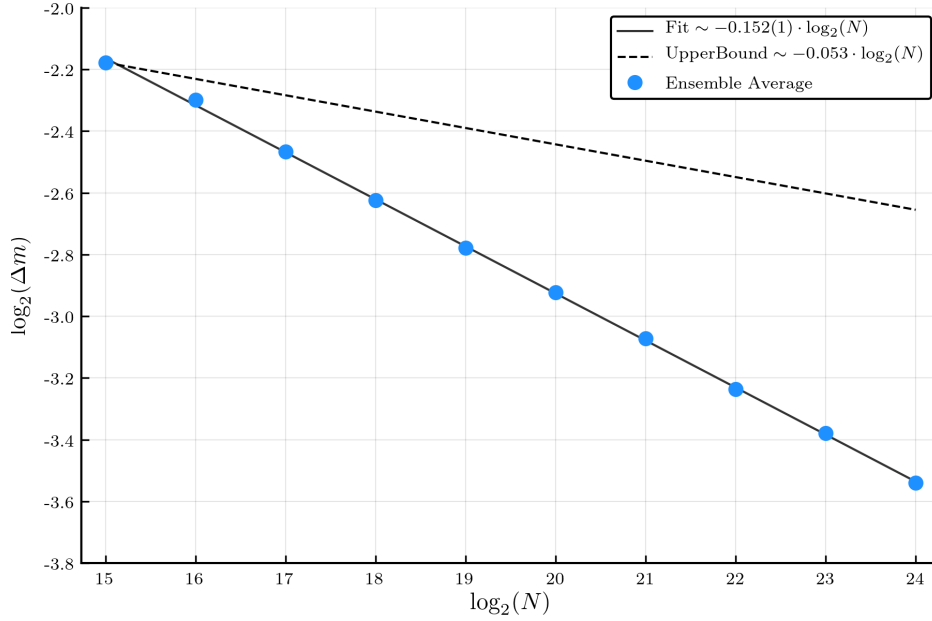

Figure 3.8: Plotting the change in the order parameter versus the system size on a $\log_2 - \log_2$ scale shows that it is well below the upper bound we determined.

3.3   Possible Future Directions

There are many different avenues we could go down in the future to further the analysis presented here and in the referenced sources. The most obvious are studying q-edge models for $q \geq 3$ and seeing how the system behaves as q changes, as well as looking at systems other than random networks, such as a lattice. As a sneak peak, the Figs. below illustrate how the order parameter changes in these scenarios.

Fig. 3.9 shows the order parameter for systems of size $N = 10^6$ and $q \in \{2, 3, ..., 7, 8\}$. As we can see, increasing the number of edges evaluated delays the transition even longer, which makes sense from a practical standpoint since more candidate edges means we have more opportunities to minimize the size of the largest cluster.

Fig. 3.10 illustrates what happens when we change the graph type to a 2D square lattice allowing only edges between nearest neighbors. We see that doing so shifts order parameter to a later point, which makes sense due to the fact that on a two dimensional lattice each point only has four neighbors thus only four possible edges it can be connected with.
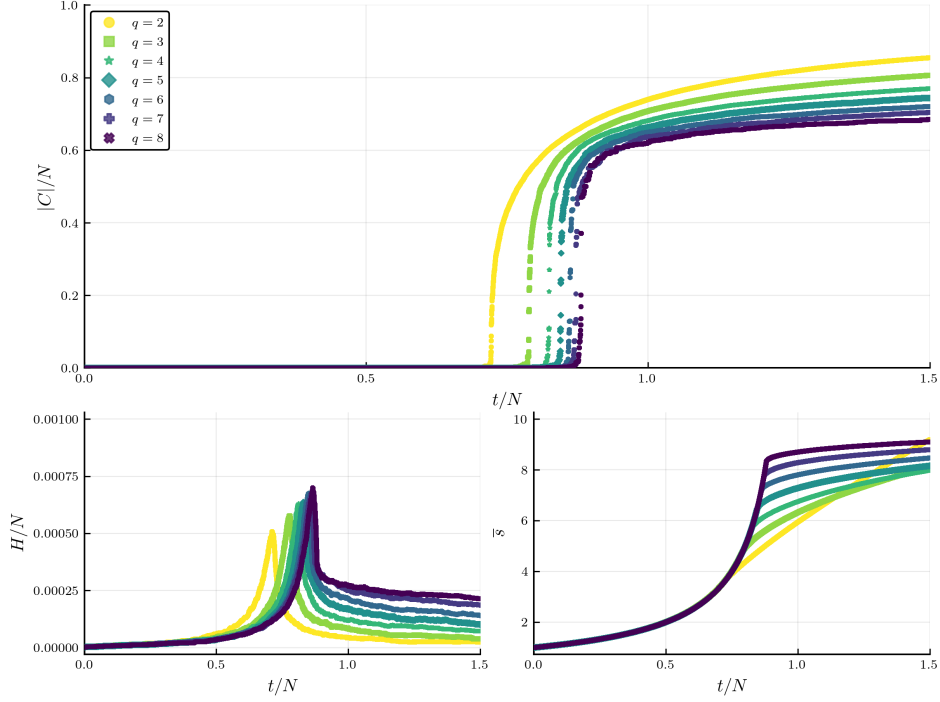
Figure 3.9: Stochastic edge acceptance model dependence on q. As q increases we see the transition is delayed even longer. The heterogeneity per node exhibits sharper peaks, and the average cluster size shows interesting behavior. $N = 10^6$.
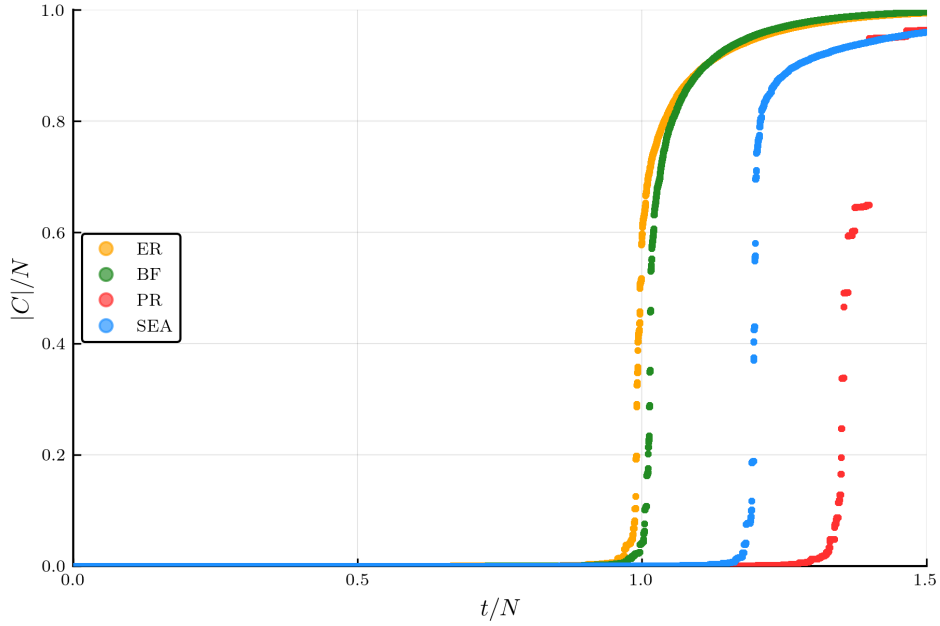
Figure 3.10: Stochastic edge acceptance model order parameter when run on a 2D square lattice. $L = 10^3$.

Fig. 3.11 illustrates the case of allowing only nearest neighbor edges on a 3D cubic lattice. The transition is delayed longer than that in a random network but is before that on a 2D lattice, which again makes sense because now we have six neighbors with which we can activate edges between.
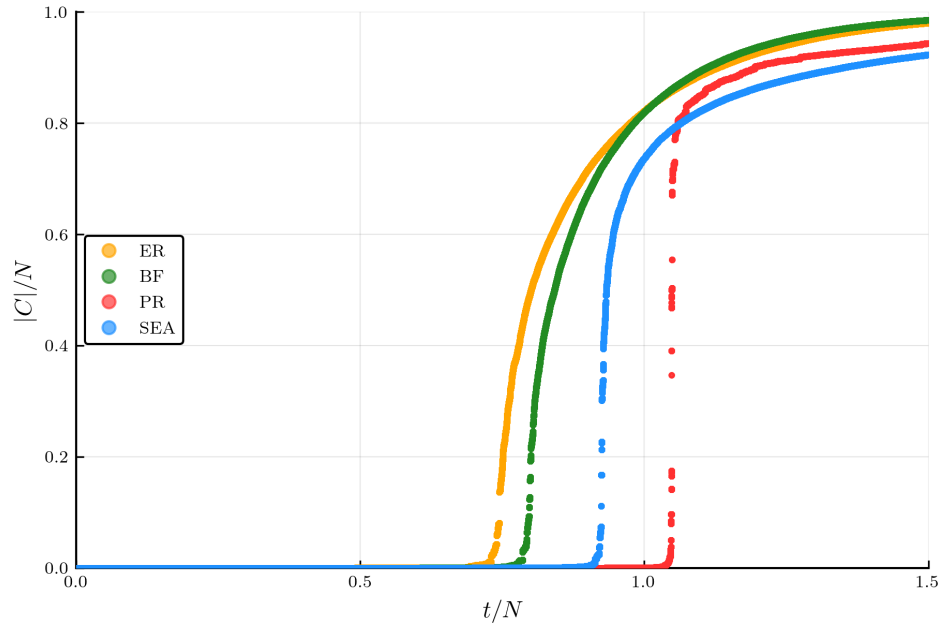
Figure 3.11: Stochastic edge acceptance model order parameter when run on a 3D cubic lattice. $L = 10^2$.

Chapter 4

Conclusion

When Dmitris Achlioptas raised the question in 2000, he probably did not foresee the cascade of follow up questions it would lead to. As we have seen, small changes to the way that edges are added to the graph can drastically alter the nature of the phase transition, leading up to what some people believe to be a discontinuous transition. There was a lengthy debate over the course of several years discussing the continuity of these transitions, which as of now appears to have been resolved. The deciding factor in whether or not these transitions are continuous is what information they take into account; algorithms that take only local information into account produce continuous transitions whereas some algorithms that take global information into account can product discontinuous transitions.

Since it was first discussed the topic of explosive percolation has been heavily researched and there have been many different models created and analyzed. The product rule seems to be the most famous of them all, and is the rule which was initially believed to be discontinuous, but was later proved continuous after using more advanced analysis techniques combined with larger scale simulations. The da Costa rule is another that was studied in depth, later leading to a full fledged theory on how to study such systems. Even though these local information based models were proven continuous, they still show unusual finite-size scaling behavior and each are in their own universality class, which makes them interesting to study.

In 3 we analyzed a new model (stochastic edge acceptance) using the framework laid out in [25], where we studied the scaling behavior of the cluster size distribution near the critical point to determine critical exponents which we used to place an upper bound on the order parameter, thus showing continuity. The SEA model was merely a toy model to show proof of concept and to see how the phase transition would look like with a little extra randomness implemented. Overall I think that the stochastic edge acceptance model was a success, as the data obtained from the simulations fit well with the framework within which we were studying it.

I believe that through this thesis I have gained a much deeper understanding of percolation theory and computational physics as a whole. I very much enjoyed writing the associated library (GraphEvolve.jl) which I used to run my simulations, and it was extremely satisfying when the output data showed nice power law scaling relations near the critical point. As of now I am happy with what I was able to accomplish here and the future directions listed at the end of Ch. 3 are there and ready to be studied if I choose to dive further into this topic.

I must say thanks to Prof. Dr. Wolfhard Janke, Dr. Stefan Schnabel, M.Sc.

Henrik Christiansen, and Alan Sammarone for the interesting and stimulating conversations regarding the topic of explosive percolation.

Appendix A

Appendix

Full documentation for `GraphEvolve.jl` can be found at percolation.fun.

A.1 Designing the Library

I chose Julia as the language to write the library in for several reasons, the largest being that I believe Julia is the perfect language for running large scale distributed computational physics/mathematics simulations. The benefit of being able to use it as an interactive language while at the same time getting speeds of a compiled language can not be understated since the immediate feedback and ability keep a kernel running with data in RAM allows for easy debugging and data exploration. Julia code revolves around its type system which works nicely with its use of multiple dispatch, and at the heart of the type system is defining custom types and creating hierarchies with which functions can take specific types. The type hierarchy of Julia's default types is illustrated in Fig. A.1.
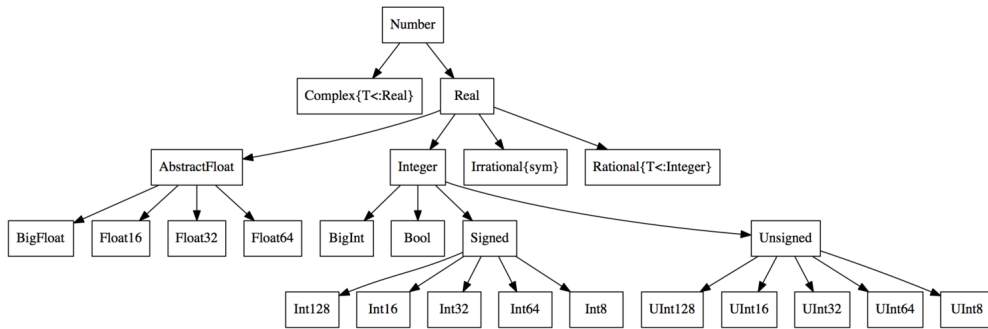

Figure A.1: Julia Type Hierarchy

The type system is actually opt-in, meaning that one need not type their variables, but typing can help with performance as well as gaining a better understanding of how the code works.

I started off with creating custom types to represent networks/lattices, the type hierarchy of them is illustrated below in Fig. A.2.
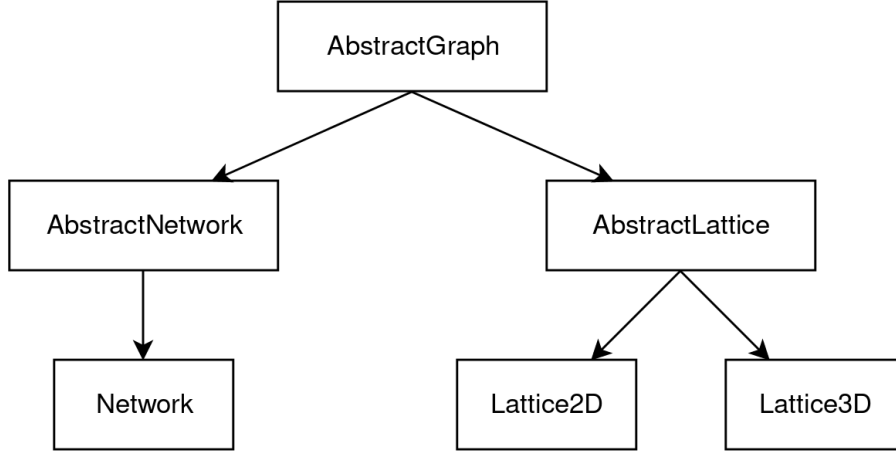
Figure A.2: Custom Type Hierarchy

As of now there are three different types of graphs: `Network`, `Lattice2D`, and `Lattice3D`. The `Network` type represents an undirected random network where edges can be present between any two unique nodes. The `Lattice2D` type represents a two-dimensional square lattice where edges can be present only between nearest neighbors, similarly where `Lattice3D` represents a three-dimensional cubic lattice. The following is a brief conceptual overview of how the system functions, for more detailed descriptions and explanations of methods or variables please refer to the documentation.

Each of these types have a similar structure in the way they store information about the underlying graph they represent. To instantiate a type one must pass it the system size and optionally a seed value for the random number generator (which is used in the evolution for randomly selecting edges), which in a `Network` means passing the number of nodes `n` and in a `LatticedD` passing the side length `L` ($n = L^d$). The types also have a variable called `t` which refers to the step in the evolution process and represents the number of edges present in the graph. The edges present in the graph are kept track in the `edges` variable, which is a set of two-tuples of integers representing an edge between node `i` and `j` as `(i, j)`. As of now the edges in all types are undirected i.e. `(i, j) = (j, i)`, but only `(i, j)` is stored to minimize unnecessary RAM usage.

Keeping track of the clusters as they evolve is the meat of these simulations, therefore it is necessary to speak about how this is done. `cluster_ids` is a d-dimensional array `(d = 1 (Network), 2 (Lattice2D), 3 (Lattice3D))` which contains the information relating to which cluster a node is in. `clusters` is a dictionary where the keys are the cluster IDs and the values are sets of nodes which are in the given cluster. `cluster_sizes` is a dictionary where the the keys are the cluster sizes (s) and the values are the cluster counts (n_s). The associated observables are stored in a custom type called `Observables` under the name `observables`.

Now that we understand how the clusters are tracked let's talk about how this is done in reality. Let `t` be the current step in the evolution process where edge `(i, j)` has been chosen to be added to the graph. Let C_i represent the cluster to which node `i` is a member of, i.e. $i \in C_i$. We then look and determine which cluster is smaller and merge the smaller cluster into the larger cluster in-place (this small

trick saves a significant amount of computational time), e.g. if $|C_\mathtt{j}| < |C_\mathtt{i}|$ then all of the nodes in $C_\mathtt{j}$ will have their cluster ID updated to that of $C_\mathtt{i}$ and the set of nodes representing $C_\mathtt{i}$ will be updated to include the nodes in $C_\mathtt{j}$, and the entry for $C_\mathtt{j}$ will then be deleted from `clusters` to reduce the memory footprint.

After the clusters are merged the associated observables are updated, which currently includes the largest cluster size, the average cluster size, the cluster heterogeneity (number of unique cluster sizes), and some post-simulation analysis quantities used for identifying where the phase transitions and if it is continuous.

A.1.1   Making Use of Multiple Dispatch

Multiple dispatch allows us to define multiple functions with the same name, yet different code depending on the input type. This is extremely useful for dealing with differences between networks and lattices such as when selecting an edge, because for example in a `Network` we can select any inactive edge between nodes, whereas in a `LatticedD` we can only select an inactive edge between nearest neighbors (this can be seen implemented in `edge_methods.jl`).

Another benefit of this coupled with the type system is that when we have a function that is generic enough to act on both `Network`s and `LatticedD`s then we can set the input type for that function to `AbstractGraph` or if it's general enough to act on all `LatticedD`s but not `Network`s then we can set the input type to `AbstractLattice`. This allows for consistency, e.g. we can call `product_rule!(Network(Int(1e6)))` and similarly `product_rule!(Lattice2D(Int(1e3)))` and the code will decide based on the type which functions to use.

A.1.2   Making Use of Parallelization

Currently when a single simulation is run it is run on a single core. For the data used in this thesis I use a script which has a parallel for loop which takes advantage of all threads that are enabled to run simulations in parallel with different seed values. Recently the Julia developers published an article talking about "composable multi-threaded parallelism", which is currently in the early stages, but looks promising and could make future parallelization very easy. Once stable it can be implemented in `GraphEvolve.jl` to run the updating of the observables in a separate thread, i.e. if we would like to gather information about an observable that takes too much time to run on the single thread we could spawn a new thread to compute this in parallel with the rest of the simulation.

A.1.3   Running a Simulation

Running a simulation is simple, one must first download the `GraphEvolve.jl` library, which can be found on GitHub under the MIT license. Then, to run a simulation on a `Network` with 10^6 nodes using the product rule to add $1.5 \cdot 10^6$ edges, and view the largest cluster size time series data we would run:

```
julia> using GraphEvolve

julia> using Plots; pyplot(fmt="png"); using LaTeXStrings
```

```julia
julia> PyPlot.matplotlib.rc("mathtext", fontset="cm");

julia> PyPlot.matplotlib.rc("text", usetex=true);

julia> PyPlot.matplotlib.rc("font", family="serif", size=12);

julia> g = Network(10^6);

julia> stochastic_edge_acceptance!(g, Int(1.5*10^6));

julia> x = collect(0:g.t) ./ g.N;

julia> y = g.observables.largest_cluster_size ./ g.N;

julia> plot_ = plot(dpi=300);

julia> scatter!(x, y,
          legend=false,
          marker=(2, :dodgerblue, :circle, 0.9, stroke(0)),
          xaxis=(latexstring("t/N"), (0, 1.5), 0:0.5:1.5),
          yaxis=(latexstring("|C|/N"), (0, 1), 0:0.2:1)
       );

julia> savefig(plot_, "/tmp/order_param.png")
```

Bibliography

[1] D. Stauffer and A. Aharony. *Introduction to Percolation Theory.* 2nd ed. Philadelphia, Pennsylvania: Taylor and Francis, 2003.

[2] M. Sahimi. *Applications of Percolation Theory.* 1st ed. Philadelphia, Pennsylvania: Taylor and Francis, 1994.

[3] P. Erdős and A. Rényi. "On random graphs, I." In: *Publ. Math. Debrecen* 6 (1959), pp. 290–297.

[4] P. Erdős and A. Rényi. "On the evolution of random graphs". In: *Publ. Math. Inst. Hungar. Acad. Sci.* 5 (1960), pp. 17–61.

[5] T. Bohman and A. Frieze. "Avoiding a giant component". In: *Random Structures Algorithms* 19 (2001), pp. 75–85.

[6] J. Spencer and N. Wormald. "Birth control for giants". In: *Combinatorica* 27 (2007), pp. 587–628.

[7] T. Bohman, A. Frieze, and N. C. Wormald. "Avoidance of a giant component in half the edge set of a random graph". In: *Random Struct. Algorithms* 25 (2004), pp. 432–449.

[8] D. Achlioptas, R. M. D'Souza, and J. Spencer. "Explosive Percolation in random networks". In: *Science* 323 (2009), pp. 1453–1455.

[9] R. M. Ziff. "Explosive growth in biased dynamic percolation on two-dimensional regular lattice networks". In: *Phys. Rev. Lett.* 103.045701 (2009).

[10] Y. S. Cho et al. "Percolation transitions in scale-free networks under the Achlioptas Process". In: *Phys. Rev. Lett.* 103.135702 (2009).

[11] F. Radicchi and S. Fortunato. "Explosive Percolation in scale-free networks". In: *Phys. Rev. Lett.* 103.168701 (2009).

[12] E. J. Friedman and A. S. Landsberg. "Construction and analysis of random networks with Explosive Percolation". In: *Phys. rev. Lett.* 103.255701 (2009).

[13] R. M. Ziff. "Scaling behavior of Explosive Percolation on the square lattice". In: *Phys. Rev. E* 82.051105 (2010).

[14] F. Radicchi and S. Fortunato. "Explosive Percolation: a numerical analysis". In: *Phys. Rev. E* 81.036110 (2010).

[15] R. M. D'Souza and M Mitzenmacher. "Local cluster aggregation models of Explosive Percolation". In: *Phys. Rev. Lett.* 104.195702 (2010).

[16] R. A. da Costa et al. "Explosive Percolation transition is actually continuous". In: *Phys. Rev. Lett.* 105.255701 (2010).

[17] H. D. Rozenfeld, L. K. Gallos, and H. A. Makse. "Explosive Percolation in the human protein homology network". In: *The European Physical Journal B* 75 (2010), pp. 305–310.

[18] N. A. M. Araújo and H. J. Herrmann. "Explosive Percolation via control of the largest cluster". In: *Phys. Rev. Lett.* 105.035701 (2010).

[19] A. A. Moreira et al. "Hamiltonian approach for Explosive Percolation". In: *Phys. Rev. E* 81.040101 (2010).

[20] Y. S. Cho, B. Kahng, and D. Kim. "Cluster aggregation model for discontinuous percolation transition". In: *Phys. Rev. E* 81.030103 (2010).

[21] Y. S. Cho and B. Kahng. "Suppression effect on Explosive Percolation". In: *Phys. Rev. Lett.* 107.275703 (2011).

[22] J. Nagler, A. Levina, and M. Timme. "Impact of single links in competitive percolation". In: *Nature Physics* 7 (2011), pp. 265–270.

[23] S. S. Manna and A. Chatterjee. "A new route to Explosive Percolation". In: *Physica A: Statistical Mechanics and its Applications* 390 (2011), pp. 177–182.

[24] P. Grassberger et al. "Explosive Percolation is continuous, but with unusual finite size behavior". In: *Phys. Rev. Lett.* 106.225701 (2011).

[25] H. K. Lee, B. J. Kim, and H. Park. "Continuity of the Explosive Percolation transition". In: *Phys. Rev. E* 84.020101 (2011).

[26] O. Riordan and L. Warnke. "Explosive Percolation is continuous". In: *Science* 333 (2011), pp. 322–324.

[27] H. Hooyberghs and B. Van Schaeybroeck. "Criterion for explosive percolation transitions on complex networks". In: *Phys. Rev. E* 83.032101 (2011).

[28] J. Nagler, T. Tiessen, and H. W. Gutch. "Continuous percolation with discontinuities". In: *Phys. Rev. X* 2.031009 (2011).

[29] W. Chen and R. M. D'Souza. "Explosive Percolation with multiple giant components". In: *Phys. Rev. Lett.* 106.115701 (2011).

[30] K. Panagiotou et al. "Explosive Percolation in Erdős-Rényi-like random graph processes". In: *Electronic Notes in Discrete Mathematics* 38 (2011), pp. 699–704.

[31] R. K. Pan et al. "Using Explosive Percolation in analysis of real-world networks". In: *Phys. Rev. E* 83.046112 (2011).

[32] Y. S. Cho and B. Kahng. "Discontinuous percolation transitions in real physical systems". In: *Phys. Rev. E* 84.050102 (2011).

[33] J. Gómez-Gardeñes et al. "Explosive synchronization transitions in scale-free networks". In: *Phys. Rev. Lett.* 106.128701 (2011).

[34] L. Tian and D.-N. Shi. "The nature of Explosive Percolation phase transition". In: *Phys. Lett. A* 376 (2012), pp. 286–289.

[35] O. Riordan and L. Warnke. "Achlioptas Process phase transitons are continuous". In: *The Annals of Applied Probability* 22 (2012), pp. 1450–1464.

[36] O. Riordan and L. Warnke. "Achlioptas Processes are not always self-averaging". In: *Phys. Rev. E* 86.011129 (2012).

[37] S. Boettcher, V. Singh, and R. M Ziff. "Ordinary percolation with discontinuous transitions". In: *Nature Communications* 3.787 (2012).

[38] W. Chen, Z. Zheng, and R. M. D'Souza. "Deriving an underlying mechanism for discontinuous percolation". In: *Europhysics Letters* 100.66006 (2012).

[39] S. Angst et al. "Explosive Ising". In: *J. Stat. Mech.: Theory and Experiment* 2012.L06002 (2012).

[40] G. Bizhani, M. Paczuski, and P. Grassberger. "Discontinuous percolation transitions in epidemic processes, surface depinning in random media, and Hamiltonian random graphs". In: *Phys. Rev. E* 86.011128 (2012).

[41] Y. S. Cho et al. "Avoiding a spanning cluster in percolation models". In: *Science* 339 (2013), pp. 1185–1187.

[42] M. Schröder, S. H. Ebrahimnazhad Rahbari, and J. Nagler. "Crackling noise in fractional percolation". In: *Nature Communications* 4.2222 (2013).

[43] W. Chen et al. "Phase transitions in supercritical Explosive Percolation". In: *Phys. Rev. E* 87.052130 (2013).

[44] W. Chen et al. "Unstable supercritical discontinuous percolation transitions". In: *Phys. Rev. E* 88.042152 (2013).

[45] S. Squires et al. "Weakly explosive percolation in directed networks". In: *Phys. Rev. E* 87.052127 (2013).

[46] S. Do Yi et al. "Percolation properties of growing networks under an achlioptas process". In: *Europhysics Letters* 103.26004 (2013).

[47] W. Chen et al. "Microtransition cascades to percolation". In: *Phys. Rev. Lett.* 112.155701 (2014).

[48] N. Bastas et al. "Explosive Percolation: Unusual transitions of a simple model". In: *Physica A: Statistical Mechanics and its Applications* 407 (2014), pp. 54–65.

[49] Y. S. Cho and B. Kahng. "Types of discontinuous percolation transitions in cluster merging processes". In: *arXiv* 1404.4470 (2014).

[50] R. A. da Costa et al. "Solution of the Explosive Percolation quest: Scaling functions and critical exponents". In: *Phys. Rev. E* 90.022145 (2014).

[51] O. Riordan and L. Warnke. "The evolution of subcritical Achlioptas Processes". In: *Random Structures and Algorithms* 47 (2014), pp. 174–203.

[52] S. Guan et al. "Explosive synchronization in adaptive and multilayer networks". In: *Phys. Rev. Lett.* 114.038701 (2015).

[53] R. A. da Costa et al. "Critical exponents of the explosive percolation transition". In: *Phys. Rev. E* 89.042148 (2014).

[54] R. A. da Costa et al. "Solution of the explosive percolation quest. II. Infinite-order transition produced by the initial distributions of clusters". In: *Phys. Rev. E* 91.032140 (2015).

[55] R. A. da Costa et al. "Inverting the Achlioptas rule for explosive percolation". In: *Phys. Rev. E* 91.042130 (2015).

[56] Raissa M. D'Souza and Jan Nagler. "Anomalous critical and supercritical phenomena in explosive percolation". In: *Nature Physics* 11 (2015), pp. 531–538.

[57] Satoru Hayasaka. "Explosive percolation in thresholded networks". In: *Physica A: Statistical Mechanics and its Applications* 451 (2016), pp. 1–9.

[58] Pau Clusella et al. "Immunization and Targeted Destruction of Networks using Explosive Percolation". In: *Phys. Rev. Lett.* 117.208301 (2016).

[59] S. Boccaletti et al. "Explosive transitions in complex networks' structure and dynamics: Percolation and synchronization". In: *Physics Reports* 660 (2016), pp. 1–94.

[60] İ. Yalçınkaya and Z. Gedik. "Coherent transport over an explosive percolation lattice". In: *Journal of Physics A: Mathematical and Theoretical* 50.15 (2017).

[61] M. M. Rahman and M. K. Hassan. "Explosive percolation on a scale-free multifractal weighted planar stochastic lattice". In: *Phys. Rev. E* 95.042133 (2017).

[62] Alex Waagen, Raissa M. D'Souza, and Tsai-Ching Lu. "Explosive percolation on directed networks due to monotonic flow of activity". In: *Phys. Rev. E* 96.012317 (2017).

[63] Yong Zhu and Xiaosong Chen. "Finite size scaling theory for percolation phase transition". In: *arXiv* 1710.02957 (2017).

[64] M. M. H. Sabbir and M. K. Hassan. "Product-sum universality and Rushbrooke inequality in explosive percolation". In: *Phys. Rev. E* 97.050102 (2018).

[65] A. J. Trevelyan, G. Tsekenis, and E. I. Corwin. "Degree product rule tempers explosive percolation in the absence of global information". In: *Phys. Rev. E* 97.020301 (2018).