

ABSTRACT

Title of dissertation: PHASE TRANSITIONS IN COMPLEX
NETWORK DYNAMICS

Shane Squires, Doctor of Philosophy, 2014

Dissertation directed by: Professor Michelle Girvan
Department of Physics

Two phase transitions in complex networks are analyzed. The first of these is a percolation transition, in which the network develops a macroscopic connected component as edges are added to it. Recent work has shown that if edges are added “competitively” to an undirected network, the onset of percolation is abrupt or “explosive.” A new variant of explosive percolation is introduced here for directed networks, whose critical behavior is explored using numerical simulations and finite-size scaling theory. This process is also characterized by a very rapid percolation transition, but it is not as sudden as in undirected networks.

The second phase transition considered here is the emergence of instability in Boolean networks, a class of dynamical systems that are widely used to model gene regulation. The dynamics, which are determined by the network topology and a set of update rules, may be either stable or unstable, meaning that small perturbations to the state of the network either die out or grow to become macroscopic. Here, this transition is analytically mapped onto a well-studied percolation problem, which can be used to predict the average steady-state distance between perturbed and

unperturbed trajectories. This map applies to specific Boolean networks with few restrictions on network topology, but can only be applied to two commonly used types of update rules.

Finally, a method is introduced for predicting the stability of Boolean networks with a much broader range of update rules. The network is assumed to have a given complex topology, subject only to a locally tree-like condition, and the update rules may be correlated with topological features of the network. While past work has addressed the separate effects of topology and update rules on stability, the present results are the first widely applicable approach to studying how these effects interact. Numerical simulations agree with the theory and show that such correlations between topology and update rules can have profound effects on the qualitative behavior of these systems.

PHASE TRANSITIONS IN COMPLEX NETWORK DYNAMICS

by

Shane Squires

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:

Professor Michelle Girvan, Chair/Advisor

Professor Edward Ott, Co-Advisor

Professor Thomas M. Antonsen

Professor Brian Hunt

Professor Héctor Corrada Bravo

© Copyright by
Shane Squires
2014

Dedication

For my loving wife, Anita

Acknowledgements

None of the work presented here would have been possible without the guidance and generosity of my advisors, Profs. Girvan and Ott. Andrew Pomerance deserves considerable thanks for a multitude of thought-provoking discussions and brainstorming sessions, as well as for having the patience to be the sounding board for most of my bad ideas. I am grateful to Prof. Antonsen, Katherine Sytwu, and Diego Alcala, for their contributions to the research described in Chapter 2. I would also like to thank all of the members of my thesis committee for their time and help.

I have also benefited considerably from being around, and learning from, a number of bright and friendly colleagues. These include Mark Herrera, Wai Lim Ku, John Platig, Anjor Kanekar, Matthew Kretschmer, Can Guven, Zhixin Lu, Ari Zitin, and Alex Gorowara, among others. In this category, Mark Kegel, Meghan Driscoll, Nick Setzer, and Meem Mahmud also merit special attention for being such excellent and steadfast friends during my time in graduate school.

I also acknowledge ONR (grant N00014-07-1-0734) and ARO (grant W911NF-12-1-0101) for funding while performing the research for this thesis.

Finally, I am deeply thankful for my wife Anita and her unconditional love and support.

Table of Contents

List of Figures	vi
List of Tables	vii
List of Abbreviations	viii
1 Background	1
1.1 Networks	2
1.2 Random and complex networks	5
1.3 Phase transitions	8
1.4 Connected components and percolation	10
1.5 Boolean networks	13
1.6 Stability of Boolean networks	16
2 Weakly explosive percolation in directed networks	19
2.1 Introduction	19
2.2 The directed competition process	22
2.3 Results	25
2.4 Discussion	36
3 Dynamical instability in Boolean networks as a percolation problem	39
3.1 Introduction	39
3.2 Mapping the annealed approximation onto site percolation	42
3.2.1 The annealed approximation	42
3.2.2 The map to site percolation for network ensembles	44
3.3 Mapping the semi-annealed approximation onto percolation	47
3.3.1 The semi-annealed approximation on locally tree-like networks	47
3.3.2 The map for specific networks with biased update rules	49
3.3.3 The map for specific networks with canalizing update rules	50
3.4 Numerical results	52
3.5 Discussion	58
3.A Finite-size effects	60
3.B Convergence analysis	64

3.C	On the uniqueness of the steady-state damage probabilities	69
4	The joint effects of topology and update rules on the stability of Boolean networks	74
4.1	Introduction	74
4.2	Generalized semi-annealing and dynamical biases	76
4.3	Stability analysis	82
4.4	Numerical results	85
4.4.1	Example 1: XOR, OR, and AND update rules	86
4.4.2	Example 2: Threshold networks	88
4.5	Discussion	93
4.A	Asynchronous updates	95
4.B	Critical slope	97
4.C	Computational considerations	98
4.D	Individual contributions to instability	99
	Glossary of Frequently Used Notation	102
	References	104

List of Figures

1.1	Order parameter of a continuous phase transition	9
1.2	Giant component of a directed network	13
1.3	Example of a Boolean update rule	15
2.1	Percolation in undirected network growth processes	21
2.2	Percolation in directed network growth processes	26
2.3	Scaling of the maximum jump size ΔS with N	31
2.4	Scaling of the variance in S with N	32
2.5	Collapse of order parameter onto a universal scaling function	34
3.1	Ensemble averages of Y , S , and T as a function of z	54
3.2	Y versus S for individual neutrally assortative networks	55
3.3	Y , S , and T for individual networks with limited update rule averaging	57
3.4	Y versus S for networks with canalizing update rules	58
3.5	Finite-size effects in Y and S as N increases	61
3.6	Convergence time of T , Y , and S as a function of z	66
3.7	Convergence time for T , Y , and S as a function of N	68
4.1	Stability in Boolean networks with XOR, OR, and AND update rules .	89
4.2	Stability of threshold Boolean networks	91

List of Tables

2.1	Critical exponents for several percolation processes	28
-----	--	----

List of Abbreviations

AP	Achlioptas process
BT(i)	Bow-tie component of i
C(i)	Component of i (undirected)
DCP	Directed competition process
DER	Directed Erdős-Rényi process
ER	Erdős-Rényi process
IN(i)	In-component of i
GBT	Giant bow-tie
GC	Giant component
GIN	Giant in-component
GOUT	Giant out-component
GSCC	Giant strongly connected component
OUT(i)	Out-component of i
SCC(i)	Strongly connected component of i

Chapter 1: Background

The work described in this thesis addresses two major topics, both of which may be described as phase transitions in networks or network dynamics. Therefore, Sections 1.1 and 1.2 begin with a brief introduction to random and complex networks. Next follows a similar introduction to phase transitions in statistical physics in Section 1.3. Since both topics are vast, the focus here will be limited to concepts, terminology, and notation which will be useful later in the text.

Subsequently, Section 1.4 introduces the first phase transition under study, percolation on networks. Percolation refers to the development of a connected component which spans a significant fraction of a network with a large number of nodes. The other major theme of this work, the dynamical stability of Boolean networks, is introduced in Sections 1.5-1.6. This concerns the general response of a network of interacting two-state elements to a small perturbation which changes the states of some elements.

Chapters 2-4 each describe original research relating to these two topics. These chapters are based upon material first released in Refs. [1], [2], and [3], respectively. In Chapter 2, I discuss a recently introduced type of non-standard percolation process called explosive percolation, and I study its properties on directed networks.

Next, in Chapter 3, I show that there is a deep connection between the two topics of this work: for some types of Boolean networks, the stability phase transition can be mapped onto a related percolation phase transition. Finally, in Chapter 4, I develop and illustrate a new technique which can predict the stability of a much broader class of Boolean networks.

1.1 Networks

A variety of social, biological, and technological systems can be represented as a collection of interacting elements, in which each element interacts with a limited number of others [4]. Such systems are naturally represented as networks. A network is a collection of N nodes and a set of edges which join pairs of nodes. Each node may be labeled by an index from 1 to N (typically denoted by i or j), and edges may be described by the nodes that they connect (for example, the edge from j to i). The total number of edges in the network will be denoted E .

Networks can be classified into a variety of types and described by a variety of features. One important distinction is between directed networks, in which each edge can only be traversed in one direction (i.e., from j to i , but not from i to j), and undirected networks, in which each edge can be traversed in both directions. A directed network may contain edges from both j to i and from i to j , but in this case the two are considered distinct edges, whereas they are considered a single edge in an undirected network. When there is an edge from j to i , j will be called an input node of i and i an output node of j . The set of edges in either directed and

undirected networks may be described by an adjacency matrix A , where $A_{ij} = 1$ if there is an edge from j to i , and $A_{ij} = 0$ if there is not. Note that I am using the convention that A_{ij} refers to the edge *to* node i *from* node j , which will be more convenient for writing sums over a node's inputs.

One of the most important characteristics of a network node is its degree, which is the number of edges to which it is connected. In an undirected network, the degree of a node i is a non-negative integer d_i . The degree is related to the adjacency matrix by $d_i = \sum_j A_{ij} = \sum_j A_{ji}$. For directed networks, however, there are two distinct concepts of the degree of i . First, there is the number of edges from other nodes j to i , called the in-degree of i and denoted $d_i^{\text{in}} = \sum_j A_{ij}$. Second, there is the number of edges from i to other nodes j , called the out-degree of i and denoted $d_i^{\text{out}} = \sum_j A_{ji}$. The degree of a node is frequently related to its importance in the structure of the network.

In fact, two of the most important properties of a network are its average degree, which will be denoted z , and its degree distribution, which will be denoted a_{kl} for directed networks or a_k for undirected networks. In directed networks, it is important to note that each edge contributes to the in-degree of one node and the out-degree of another, so the respective totals of the in- and out-degrees in the network must be equal. Therefore, the average degree z for a given directed network can be defined as

$$z = \langle d_i \rangle = \frac{1}{N} \sum_i d_i^{\text{in}} = \frac{1}{N} \sum_i d_i^{\text{out}} = \frac{E}{N}, \quad (1.1)$$

where d_i in $\langle d_i \rangle$ could refer to either d_i^{in} or d_i^{out} . In an undirected network, the

average degree is given by $z = 2E/N$, because each edge contributes to the degrees of two nodes rather than one.

The degree distribution a_{kl} is defined as the fraction of nodes i whose degrees are $d_i^{\text{in}} = k$ and $d_i^{\text{out}} = l$. That is,

$$a_{kl} = \frac{1}{N} \sum_i \delta_{k,d_i^{\text{in}}} \delta_{l,d_i^{\text{out}}}, \quad (1.2)$$

where δ is the Kronecker delta. It is important to note that for any distribution a_{kl} , the normalization and mean must obey

$$\sum_{k,l} a_{kl} = 1, \quad (1.3a)$$

$$\sum_{k,l} k a_{kl} = \sum_{k,l} l a_{kl} = z. \quad (1.3b)$$

Similar relations hold for undirected networks. For directed networks, it is also convenient to define the marginal distributions

$$a_k^{\text{in}} = \sum_l a_{kl}, \quad (1.4a)$$

$$a_l^{\text{out}} = \sum_k a_{kl}. \quad (1.4b)$$

A degree distribution a_{kl} is said to be uncorrelated if $a_{kl} = a_k^{\text{in}} a_l^{\text{out}}$.

Empirical networks have been found to exhibit a wide variety of degree distributions, which are often believed to be related to the function, development, or dynamics of the network [4]. Three important examples which will be used in subsequent chapters are homogeneous (or regular), Poisson, and power-law (or scale-free) distributions. In an undirected network, these are respectively given by the distributions $a_k = \delta_{k,K}$ for some integer K , $a_k = z^k e^{-z}/k!$, and $a_k \propto k^{-\gamma}$ for some exponent

γ . For directed networks, the joint distribution may have a take complicated form if there are degree correlations, but in many examples, the marginal distributions a_k^{in} and a_l^{out} follow the same forms.

From here forward, I will restrict discussion to sparse networks, in which the number of nodes N in the network is large ($N \gg 1$), but the average degree z remains finite (i.e., E is on the order of N , rather than N^2). The assumption of large N allows for the application of concepts and techniques from probability theory and statistical physics, which hold exactly in the thermodynamic limit, $N \rightarrow \infty$. The assumption that z is finite is both appropriate and necessary for the applications studied in the following chapters.

1.2 Random and complex networks

Random networks are a class of models for generating networks by assigning edges to randomly selected nodes. Random networks are useful for a variety of applications, most notably that they provide a null model against which empirical data for large, sparse networks can be compared. Below, two well-known approaches to generating random networks are described, focusing on the case where the network is directed.

The prototypical random network model is the Erdős-Rényi model [5, 6], in which each pair of nodes i and j is assigned an edge with probability z/N , where z is now considered a parameter of the model. Since the existence of each edge in an Erdős-Rényi network is independent, such networks may also be generated

through a growth process in which one edge is added at a time between randomly selected pairs of nodes until the average degree is z . This perspective is adopted in Chapter 2, where this growth process is compared to other recently introduced network growth processes.

This model has been widely studied due to its simplicity (there is only one parameter, and all nodes are treated identically), as well as the fact that it undergoes a percolation phase transition (described in more detail in Section 1.4). However, the degree distribution of an Erdős-Rényi network is known to approach a Poisson distribution for large N , which severely limits the range of systems which can be modeled by Erdős-Rényi networks. In order to create random networks with more general degree distributions, the Erdős-Rényi model must be generalized. The simplest such generalization is the configuration model [4], in which an arbitrary degree distribution a_{kl} can be chosen, subject to the constraints in Eq. (1.3). Initially it is imagined that each of the E edges in the network is broken into an incoming half and an outgoing half. Each node i is then randomly assigned $d_i^{\text{in}} = k$ and $d_i^{\text{out}} = l$ with probability a_{kl} , and it is imagined that node i is then attached to d_i^{in} incoming halves and d_i^{out} outgoing halves. One by one, pairs of randomly selected incoming and outgoing halves are selected and joined to create whole edges, until all halves have been matched. In some cases, matching is impossible, but these cases can be handled through suitable re-sampling.

While random networks are an invaluable theoretical tool, a considerable number of studies over the past two decades have demonstrated that a variety of empirical social, biological, and technological networks contain structural features which

are absent in random networks. Networks with such features are typically called complex networks. Numerous complex features have been studied—see Ref. [4] for a review—but some of the most commonly studied cases are clustering, motifs, community structure, and assortative mixing. Clustering is common in friendship networks, and in this context it refers to the fact that two of an individual’s friends have a significant probability of being friends with one another, whereas in large random networks this probability would be close to 0. Motifs are small sub-networks which appear frequently in a given network, and community structure is when a network contains groups of nodes which preferentially connect to other members of their own group rather than members of other groups.

Assortative mixing is a complex network feature whose effects will be studied in detail in Chapter 3. Assortative mixing is a general concept which refers to networks where edges preferentially form between nodes which have similar values of some characteristic feature [7]; this naturally subsumes community structure as well as other types of network structure. Here, the focus is on a particular type of assortative mixing by degree called edge degree correlation. In subsequent chapters, a network will be said to be assortative if, in the set of all edges from j to i , there is a positive correlation between d_j^{in} and d_i^{out} . This can be quantified using the edge degree correlation measure ρ described in Ref. [8]. In the case of networks whose degree distribution is uncorrelated, ρ can be expressed as

$$\rho = \frac{\sum_{i,j} A_{ij} d_i^{\text{out}} d_j^{\text{in}}}{E}. \quad (1.5)$$

If d_i^{out} and d_j^{in} are uncorrelated for edges from j to i , as in a random network, then

$\rho = 1$; otherwise, positive (negative) correlations between these degrees lead to $\rho > 1$ ($\rho < 1$) [8].

In general, for complex networks, the adjacency matrix A contains significant information about the structure which is absent in the degree sequence a_{kl} . To some extent, this information can be quantified using measures of features such as community structure and assortativity. However, it is reasonable to expect that empirical networks may, for a variety of reasons, contain complex features which are not known and have not been explicitly studied in the literature. This is one major motivation for the work described in Chapters 3 and 4, which relies directly on the adjacency matrix rather than any proxy such as the degree distribution.

1.3 Phase transitions

A phase transition is a macroscopic change in the behavior of an infinite system which occurs when an external tuning parameter P passes through a critical point P_c . The behavior of the system is measured through an order parameter O . Often, several tuning parameters can be used to produce a phase transition in the same system, and the choice of order parameter is not necessarily unique.

Modern treatments generally divide phase transitions into two types, discontinuous and continuous, based on the continuity of $O(P)$ at the point $P = P_c$ [9]. These are also called first-order and higher-order phase transitions for historical reasons. A typical example of a discontinuous phase transition is the boiling of water, in which P could be the temperature and O could be the volume of the water, which

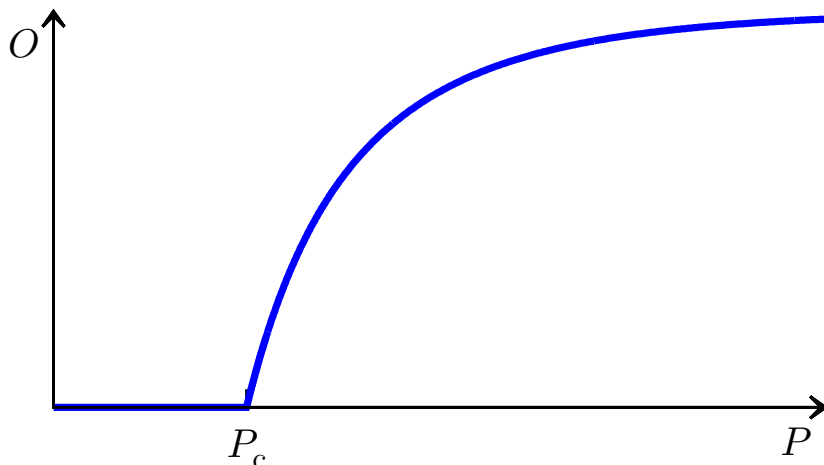


Figure 1.1: The order parameter O of a continuous phase transition as the tuning parameter P is varied (see Section 1.3). In this example, $\beta = 1$ because $O(P)$ is linear to the right of $P = P_c$. For Erdős-Rényi percolation (Section 1.4), one may use $P = z$, $P_c = 1$, and $O = S_{GC}$. For the stability phase transition in biased N - K Boolean networks (Section 1.6), one may use $P = 2b(1 - b)K$, $P_c = 1$, and $O = Y$.

changes discontinuously at the boiling temperature. In continuous phase transitions, such as the emergence of magnetization or superconductivity, the order parameter is continuous, but typically has a discontinuity in its derivative or in a higher-order derivative. A typical continuous phase transition is depicted in Fig. 1.1.

Behavior near the critical point of a continuous phase transition can be characterized by a set of critical exponents which describe the behavior of the order parameter or other variables near $P = P_c$. In Chapter 2, considerable attention is devoted to estimating the critical exponent β , which describes the scaling of $O(P)$, defined by

$$O(P) \sim (P - P_c)^\beta \tag{1.6}$$

for P near but greater than P_c . Estimating β is one way to distinguish continuous from discontinuous phase transitions, because $\beta > 0$ only in continuous phase transitions, as discussed in Chapter 2.

It is now understood that true phase transitions can only occur when the free energy becomes non-analytic, and therefore only happen in physical systems with an infinite number of degrees of freedom [9]. However, large finite systems show similar behavior to infinite systems, except that the critical point is often blurred into a “critical region,” where significant discrepancies can be observed between the finite and infinite cases. These discrepancies are known as finite size effects, and they vanish as the system size approaches infinity. Such effects will be explored quantitatively in Chapters 2 and 3.

1.4 Connected components and percolation

One elementary question which can be asked about a network node is the size of the connected component to which it belongs. In undirected networks, the concept of connectedness is unambiguous, and the network can be divided into distinct connected components. However, in directed networks, it may be possible to travel along network edges from j to i , but not from i to j . Thus, it is clear that in directed networks, there must be several distinct concepts of connected components [10]. First, there is the set of distinct nodes (including itself) which can be reached on paths along network edges beginning at i . This is called the out-component of i and is denoted $\text{OUT}(i)$. Similarly, the set of nodes from which i can be reached

is called the in-component of i and is denoted $\text{IN}(i)$. The intersection of these two sets, nodes which can both reach and be reached from i , is called the strongly connected component of i , $\text{SCC}(i)$. Finally, I define the union of $\text{IN}(i)$ and $\text{OUT}(i)$ to be the “bow-tie” $\text{BT}(i)$ (see below); note that this last definition is non-standard. For undirected networks, all four of these definitions coincide and correspond to connected components.

The percolation phase transition may be motivated by asking what the average size of the component of a randomly selected node is. Consider undirected Erdős-Rényi networks with average degree z . When $z < 1$, the network consists of a large number of isolated components, whose average size is finite. As $z \rightarrow 1$, more and more components merge, and the average size of the component of a randomly selected node grows and diverges. Finally, when $z > 1$, there is a single connected component, called the giant component (GC), which contains a macroscopic fraction of the nodes in the network [10]. That is, for large finite networks N , the size of the giant component scales as $\mathcal{O}(N)$; and for $N \rightarrow \infty$, the giant component is infinite and includes a finite fraction of the nodes in the network. This phase transition, in which a giant component is formed, is known as percolation. In this case, z is the tuning parameter for the phase transition, with $z_c = 1$, and the order parameter is given by the normalized size of the giant component,

$$S_{\text{GC}} = \frac{|\text{GC}|}{N}, \quad (1.7)$$

where $|\text{GC}|$ denotes the number of nodes in the giant component. For $N \rightarrow \infty$, $S_{\text{GC}} = 0$ when $z \leq 1$ and $S_{\text{GC}} > 0$ when $z > 1$. Moreover, for $z > 1$, $S_{\text{GC}} =$

$2(z - 1) + \mathcal{O}((z - 1)^2)$, indicating that the critical exponent β is equal to 1. This phase transition is depicted in Fig. 1.1.

These results were independently discovered by Solomonoff and Rapoport [5] and Erdős and Rényi [6] and later generalized by other authors [10, 11] to networks with arbitrary degree distributions. They may also be generalized to cases where either nodes or edges may be deleted from the network, which are known as site percolation and bond percolation, respectively. These generalizations require a longer discussion which appears in Chapter 3. The generalization to directed Erdős-Rényi networks is, however, fairly straightforward. The only necessary complication is that, as discussed above, there are several alternative concepts of connectedness in directed networks. Because of this, there are also several types of giant components, corresponding to each of the types of connected components: the giant out-component (GOUT), the giant in-component (GIN), the giant strongly connected component (GSCC), and the giant bow-tie (GBT). The relationships between these four are summarized in the well-known “bow-tie” diagram, Fig. 1.2 [12]. As order parameters, one can define S_{GIN} , S_{GOUT} , S_{GSCC} , and S_{GBT} in analogy with Eq. (1.7). It can be shown that all four giant components emerge simultaneously, so the critical point, which is also $z_c = 1$ for directed networks, is well-defined. In fact, all of the results of the preceding discussion of the giant component of an undirected network carry over in a straightforward fashion to GIN and GOUT [10]. The only notable difference is that, for the GSCC, the critical exponent β is equal to 2 rather than 1.

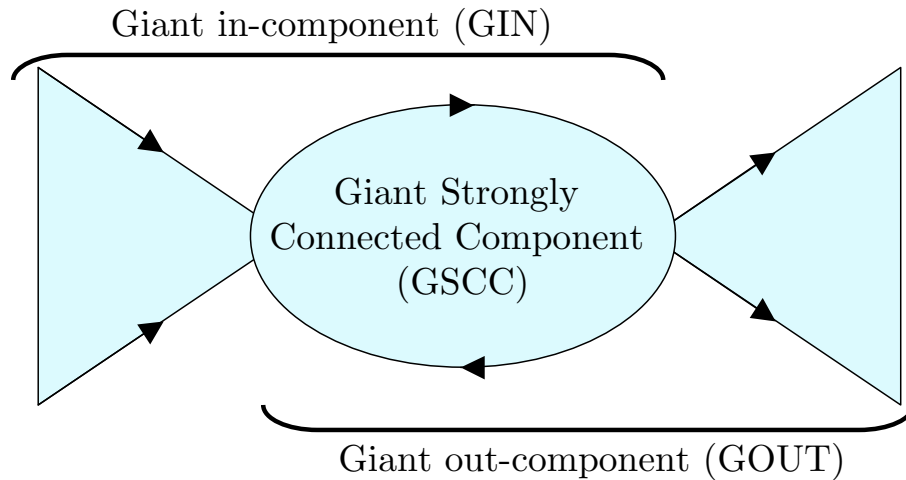


Figure 1.2: An illustration of the “bow-tie” structure of the giant component in a directed network above the percolation threshold [12]. The whole figure comprises the giant bow-tie (GBT), and the GOUT, GIN, and GSCC are marked. See text of Section 1.4 for details.

1.5 Boolean networks

Here, I introduce the second major topic of this thesis, the dynamics of Boolean networks, to be studied in Chapters 3 and 4. Boolean networks have been a prominent tool for modeling genetic regulation in cells since their introduction by in 1969 [13, 14]. Genetic regulation refers to mechanisms by which the expression of genes in a cell are controlled by other biochemical elements in the cell. The expression of a gene refers to the cellular concentration of the mRNA sequence encoded by that gene, which can, for example, be modified by proteins which encourage or inhibit transcription. These proteins, in turn, may be encoded by other genes which have regulatory functions. The network of human gene regulatory interactions is believed

to be large (on the order of thousands of elements) and have complex topology. In a Boolean network, each gene (or other relevant biochemical element) is modeled as a node in a network, and its expression level is modeled as the state of the node. Boolean modeling of gene regulatory systems relies on the approximation that each node can be modeled as having one of two expression levels, which are denoted 0 (“off”) or 1 (“on”).

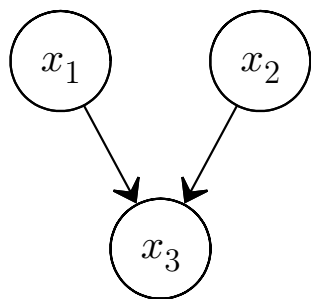
Formally, a Boolean network is a dynamical system consisting of a directed network and a fixed update rule F_i for each node i . In the network, each edge from j to i represents the existence of a regulatory interaction where j regulates i . Each update rule F_i specifies the regulatory relationship between i and its inputs. That is, F_i is a Boolean-valued function which depends only upon i ’s network inputs. Each F_i may be specified in the form of a “truth table” which lists each of the $2^{d_i^{\text{in}}}$ possible combined input states and gives the corresponding output (see Fig. 1.3).

In Boolean network models, time is typically taken in discrete steps ($t = 0, 1, 2, \dots$) and the states of nodes are updated synchronously (though asynchronous variations are also considered in Chapter 4). The dynamics are defined as follows. Consider a node i which has d_i^{in} network inputs with indices $j_1 \dots j_{d_i^{\text{in}}}$. The state of node i , $x_i(t)$, evolves in time according to

$$x_i(t+1) = F_i \left(x_{j_1}(t), \dots, x_{j_{d_i^{\text{in}}}}(t) \right). \quad (1.8)$$

A more convenient notation is to denote the ordered list of input states to node i at time t as

$$X_i(t) = \left(x_{j_1}(t), \dots, x_{j_{d_i^{\text{in}}}}(t) \right), \quad (1.9)$$



(a) A node with two inputs

x_1	x_2	F_3
0	0	0
0	1	0
1	0	0
1	1	1

(b) An update rule in truth table form

Figure 1.3: In the example network depicted in (a), node 3 has two network inputs, whose states are x_1 and x_2 . An example of one possible Boolean update rule F_3 for node 3 is shown in (b), expressed in its “truth table” form. The left-hand side of the truth table lists all possible combined input states of x_1 and x_2 , and the right-hand side lists the associated output F_3 .

which is called a combined input state for i . In this notation, Eq. (1.8) may be re-written as

$$x_i(t+1) = F_i(X_i(t)). \quad (1.10)$$

The evolution of the state vector of the network $x(t)$, starting from a specific initial condition $x(0)$, will be called an orbit or a trajectory of the network.

When modeling real gene regulatory systems, both the network edges and the update rules should be determined experimentally. However, when investigating the dynamics of the model, they can be chosen arbitrarily, and a variety of ensembles of network topologies and update rules have been studied in the literature. One recurring example will be the original system studied by Stuart Kauffman, who originally introduced Boolean networks [13, 15]. He studied random networks in

which each node has K inputs randomly selected from among the $(N - 1)$ other nodes, which are frequently called N - K networks. For update rules, he randomly assigned an output value of 0 or 1 to $F_i(X_i)$, with probability $1/2$, for each combined input state X_i . (Note that these update rules F_i are only initially assigned randomly and are fixed afterwards; they do not depend on t .) This procedure was later generalized by Derrida and Pomeau, who introduced a bias parameter b ($0 < b < 1$) and generated update rules for which each $F_i(X_i)$ was randomly assigned the value 1 with probability b and 0 with probability $(1 - b)$. Such update functions are called biased functions and are widely studied in the Boolean networks literature.

1.6 Stability of Boolean networks

One important question about any dynamical system is whether or not it is stable, i.e., whether or not small perturbations of a typical initial state tend to grow or shrink as the system evolves. While studying ensembles of N - K Boolean networks with biased update rules, Kauffman found numerically that as the in-degree K increases, there is a phase transition between a phase in which the dynamics are stable to one in which they are unstable. This unstable phase is often called chaotic, because its dynamics mimic some of the well-known properties of chaotic systems, including exponential sensitivity to initial conditions.

The standard order parameter for this stability phase transition is the average long-time normalized Hamming distance between two close initial conditions, defined as follows. Consider two trajectories, $x(t)$ and $\tilde{x}(t)$, which evolve on the same

Boolean network. Assume that $N \gg 1$ and that the initial conditions $x(0)$ and $\tilde{x}(0)$ are “close” in the sense that they differ only on a small fraction ε of nodes. Then the normalized Hamming distance between these two orbits is defined as the fraction of the nodal states that differ between them,

$$H(x(t), \tilde{x}(t)) = \frac{1}{N} \sum_{i=1}^N |x_i(t) - \tilde{x}_i(t)|. \quad (1.11)$$

Therefore, $H(x(0), \tilde{x}(0)) = \varepsilon \ll 1$. If $H(x(t), \tilde{x}(t))$ decreases to zero as the dynamics evolve under Eq. (1.10), the system is considered to be stable; if it typically grows to $\mathcal{O}(1)$, the system is considered to be unstable. More specifically, the order parameter Y is defined to be

$$Y = \langle H(x(t), \tilde{x}(t)) \rangle_{t, \text{i.c.}}, \quad (1.12)$$

where the average $\langle \cdot \rangle_{t, \text{i.c.}}$ is taken over times $t = 0$ to $t = \infty$ and over all initial conditions for which $H = \varepsilon$ at $t = 0$.

There are a variety of choices for the tuning parameter, depending on the parameters used to determine the network topology and update rules. Given the standard example of N - K networks using biased update rules with bias probability b , the quantity $2b(1-b)K$ can be used as a tuning parameter, with critical value 1. This is derived in Chapter 3, where it is shown that for this case (as well as a number of others), the stability phase transition can be mapped onto an associated percolation problem. See Fig. 1.1 for a visual depiction of the transition.

One motivation for considering the question of stability is that it may have important ramifications for biological systems which can be modeled by Boolean networks, including gene regulation and neuronal avalanches [16]. In the case of gene

regulatory networks, Kauffman introduced a long-standing hypothesis that gene regulatory networks exist near the critical border separating the stable and unstable regimes, known as the “life at the edge of chaos” hypothesis [15]. Recently, Pomerance et al. [17] introduced the additional hypothesis that orbital stability of the gene regulatory system may be causally related to cancer. Specifically, they suggested that mutations that promote instability—causing the gene regulatory network to move closer to, or across, the edge of chaos—may be a contributing factor for some types of cancers. This hypothesis is consistent with recent results which indicate that a distinguishing feature of cancer cells is extreme variation in gene expression levels [18]. Although biological aspects of this hypothesis are not explored further here, this hypothesis does motivate the treatment of possible intervention strategies for stabilizing unstable networks in Chapter 4.

Chapter 2: Weakly explosive percolation in directed networks

2.1 Introduction

In this chapter, I explore the properties of a new category of percolation processes with unusually rapid growth of the giant component, known as “explosive percolation.” These processes have recently been studied on undirected networks, but here they are generalized to directed networks, based on work published in Ref. [1]. As described in Section 1.4, a large directed or undirected network may transition from a non-percolating phase, in which every connected component is microscopic, to a percolating phase, in which there is a single “giant” component which contains a macroscopic fraction of the nodes in the network, as edges are added to the network [10]. The fraction of nodes in the giant component is the order parameter for the percolation phase transition.

The network growth process studied by Erdős and Rényi, now the prototypical example of network percolation, may be characterized as follows for the undirected case. The network initially consists of $N \gg 1$ nodes and no edges. Then, on each successive step of the growth process, a pair of nodes is selected randomly and an edge is added between them. The size of the largest connected component is recorded and the process is repeated. The percolation phase transition for networks grown in

this manner is continuous (or second-order) in the number of edges in the network. However, recent work by Achlioptas et al. demonstrated that simple modifications to this growth algorithm can induce surprisingly different behavior in the growth of the giant component [19]. In particular, they found that introducing “edge competition” during network growth results in a delayed, seemingly discontinuous (or first-order) transition, which has been called explosive percolation.

The network growth process proposed by Achlioptas et al. is designed to inhibit the formation of large connected components. At each step, *two* random candidate edges are considered, with the intention of selecting only one of them for addition to the network. If one of the edges connects two nodes in the same component, it is selected automatically because its addition would not cause any component to grow. If the addition of either edge would connect two distinct components, the product of the sizes of these two components is compared, and only the edge with the smaller product is added to the network.¹

Networks grown in this fashion percolate much later than Erdős-Rényi networks; however, when a giant component eventually forms, it grows extremely rapidly. Based on numerical simulations, Achlioptas et al. conjectured that the phase transition is discontinuous, but it has now been shown that the Achlioptas process actually produces a continuous transition [20–23]. The abrupt growth observed in numerical experiments is due to a small but positive critical exponent for the growth of the order parameter, along with strong finite-size effects which diminish only very slowly as $N \rightarrow \infty$. In spite of this, the Achlioptas process continues to attract considerable interest because, at network sizes that are typical in applica-

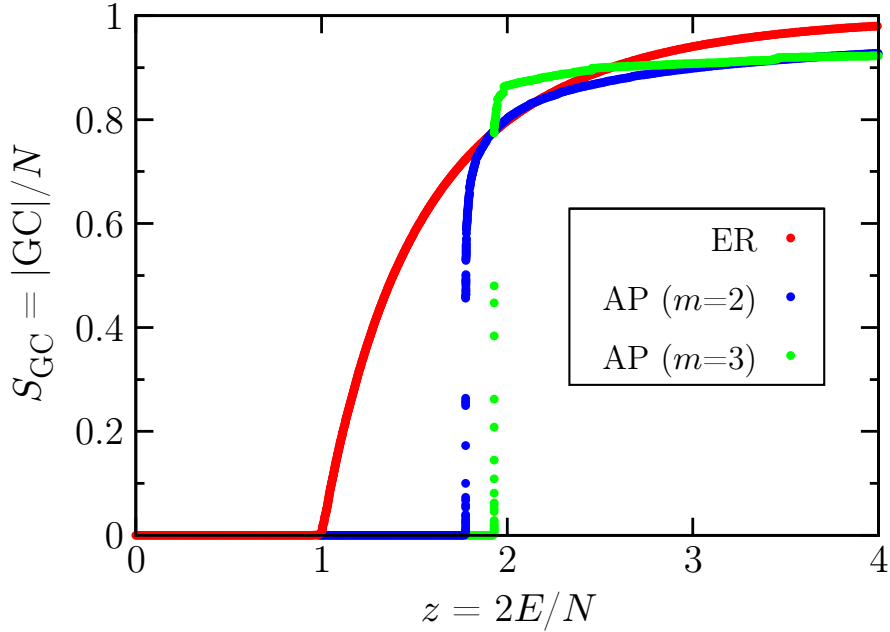


Figure 2.1: The growth of S_{GC} , the fraction of nodes in the giant component of an undirected network, for three individual networks with $N = 2^{23}$. The growth process is repeated using the Erdős-Rényi growth process (red), the Achlioptas process (blue), and a modified Achlioptas process in which three candidate edges, rather than two, are used at each network growth step (green).

tions, these finite-size effects give the percolation phase transition an “effectively” discontinuous appearance that is qualitatively different from that of traditional percolation problems (see Fig. 2.1). It has also spurred interest in other models which exhibit abrupt phase transitions, including Kuramoto [24] and Ising [25] models, as well as other modified percolation processes [26–34], many of which are believed to exhibit genuine discontinuous transitions.

As the existing literature on explosive percolation is exclusively focused on undirected networks, this chapter introduces a growth process which generalizes the

Achlioptas to directed networks. The scaling properties of this process are studied, and it is found that competitive edge percolation on directed networks shares some of the unusual qualitative features of the Achlioptas process on undirected networks (discussed further in Section 2.3). For the purposes of this chapter, these features—those which distinguish the Achlioptas process from both ordinary percolation as well as truly discontinuous models—will be referred to as “explosive,” and the behavior of the directed model introduced below as “weakly explosive.” This terminology is discussed further in Section 2.3.

2.2 The directed competition process

In order to define an Achlioptas-like process on directed networks, one first needs to define connected components in directed networks. As discussed in Section 1.4, there are multiple related definitions of the component to which a node i in a directed network belongs: (1) the in-component, $\text{IN}(i)$, the set of all nodes which have paths to i ; (2) the out-component, $\text{OUT}(i)$, the set of all nodes which can be reached from i ; (3) the strongly connected component, $\text{SCC}(i)$, the intersection of $\text{IN}(i)$ and $\text{OUT}(i)$; and (4) the full bow-tie, $\text{BT}(i)$, the union of $\text{IN}(i)$ and $\text{OUT}(i)$.² This comparison extends to the percolation transition in the directed Erdős-Rényi process, in which directed edges are successively added between randomly selected, unconnected pairs of nodes. At the critical point, a giant strongly connected component (GSCC), giant in-component (GIN), and giant out-component (GOUT) form simultaneously [10], comprising the giant bow-tie (GBT). This is illustrated in the

well-known “bow-tie diagram” (Fig. 1.2) [12].

The new directed network growth process introduced here will be called the directed competition process (DCP) to distinguish it from the Achlioptas process (AP), the Erdős-Rényi process (ER), and the directed Erdős-Rényi process (DER). It also consists of repeatedly choosing two random directed candidate edges $j_1 \rightarrow i_1$ and $j_2 \rightarrow i_2$ from the set of all distinct unoccupied edges, then selecting one for addition to the network. As in the Achlioptas process, one of the edges is automatically selected if that edge is redundant to the connectedness of the network, i.e., if there is already a path from j to i . Otherwise, the edge which has the minimum value of $|\text{IN}(j)| \cdot |\text{OUT}(i)|$ is chosen. Here, the vertical bars denote cardinality, so $|\text{IN}(j)|$ refers to the number of nodes in $\text{IN}(j)$. As in Refs. [35, 36], I also discuss generalizations of both AP and DCP in which m (rather than two) edges are chosen for consideration at each step in the growth process, and results will be shown for both $m = 2$ and $m = 3$. Note that the $m = 1$ case of AP corresponds to ER, and the $m = 1$ case of DCP corresponds to DER. Also, note that DCP is a generalization of AP, because the two processes are identical when applied to an undirected network.

The DCP edge selection rule may also be motivated by noting that it minimizes the “throughput” which is created by the addition of each edge in a way which is analogous to the Achlioptas product rule. More formally, let B_{ij} indicate whether or not there is a path from j to i , i.e., $B_{ij} = 1$ if there is such a path and $B_{ij} = 0$ if there is not. The throughput of the network can be defined as $\bar{B} = \langle B \rangle$, where the average is taken over all node pairs i and j ($i \neq j$). Well below the percolation threshold, when there are few paths from nodes in $\text{IN}(j)$ to nodes in $\text{OUT}(i)$, adding

an edge from j to i on average increases \bar{B} by approximately $|\text{IN}(j)| \cdot |\text{OUT}(i)|/N^2$. Similarly, in the Achlioptas process for an undirected network, the change in \bar{B} from the addition of a single edge to a network well below the percolation threshold is approximately $2|\text{C}(i)| \cdot |\text{C}(j)|/N^2$, where $\text{C}(i)$ and $\text{C}(j)$ are the components to which i and j belong. Thus, both rules may be construed as minimizing \bar{B} early in the network growth process. This, in turn, leads to an explosive phase transition by creating what has been termed a “powder keg” [37] of mesoscopic components which “ignites” at the critical point, when edge competition can no longer prevent them from merging.

The normalized size S_{GC} of a giant component,

$$S_{\text{GC}} = \frac{|\text{GC}|}{N}, \quad (2.1)$$

is used as the order parameter for the phase transition, where, for directed networks, GC may also be replaced with GIN, GOUT, GSCC, or GBT. Numerically, the GSCC is considered to be the largest strongly connected component in the network, the GIN and GOUT to be its in- and out-components, and the GBT to be the union of the two.³ The average degree of the network, z , will be used as a tuning parameter. For undirected networks, $z = 2E/N$, whereas for directed networks, $z = E/N$ (see Section 1.1). Note that, for undirected networks, the use of z as the tuning parameter differs slightly from the usual convention of using E/N as a tuning parameter. The use of the average degree is motivated by the observation that both undirected and directed Erdős-Rényi networks percolate at the same average degree ($z_c = 1$), so z is a natural scale for comparison between the directed and undirected cases.

Computationally, percolation simulations are more time-intensive for directed networks than undirected networks. While only $\mathcal{O}(N)$ operations are needed to simulate an entire network growth process in an undirected network [38], a naïve algorithm for competitive edge percolation in a directed network would require at least $\mathcal{O}(N^2)$ operations, because there are $\mathcal{O}(N)$ edge additions, between each of which several processes with up to $\mathcal{O}(N)$ steps must occur. These processes include checking for a path from j to i for each prospective edge $j \rightarrow i$, finding $\text{IN}(j)$ and $\text{OUT}(i)$, and decomposing the network into strongly connected components [39]. In order to improve computational performance, each part of the giant component is tracked during the network growth process, and this knowledge is used to speed up or eliminate the first two processes. For example, if j is in GIN and i is in GOUT, checking for a path from j to i is unnecessary because one must exist. Additionally, results are only reported for the giant component, not the distribution of other component sizes, to avoid the third process. The performance of this algorithm scales approximately as $\mathcal{O}(N^{1.5})$, where most of the time is spent in the critical region where more than one macroscopic or near-macroscopic component exists. This improvement enables the simulation of networks with significantly larger N than would otherwise be feasible.

2.3 Results

Plots of the order parameters versus z for single realizations of DER (red) and DCP with $m = 2$ (blue) or $m = 3$ (green) are shown in Fig. 2.2 for $N = 10^{23}$. Panels

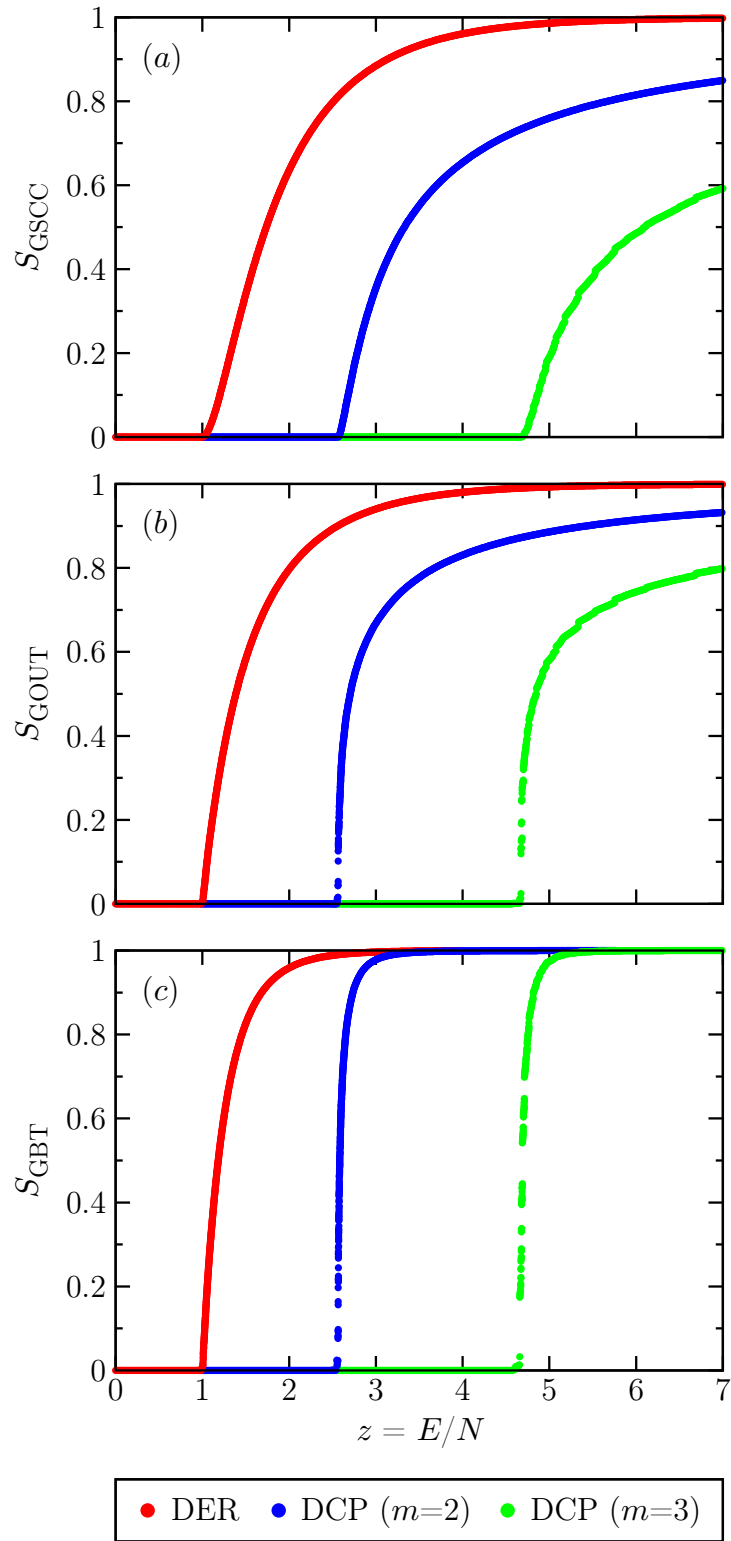


Figure 2.2: The formation of the giant component in directed networks; see text.

(a), (b), and (c) show the emergence of the giant strongly connected component, giant out-component, and giant bow-tie, respectively. When edge competition is present, the emergence of all four parts of the giant component are delayed, and the GOUT and GBT display sudden growth at the critical point which is qualitatively similar to (though less marked than) that of the Achlioptas process (Fig. 2.1). By symmetry, results for GIN are the same as those for GOUT and are not displayed.

Quantitative comparisons between DCP and AP can be made by examining several scaling exponents which characterize the features of explosive percolation [21–23, 40, 41]. In fact, the Achlioptas process is striking precisely because these exponents are small, but it is continuous because they are nonzero. These critical exponents are listed in Table 2.1 for all processes studied here.

The first quantitative measure studied here is the critical exponent β , defined by

$$\langle S \rangle \sim (z - z_c)^\beta \tag{2.2}$$

as $z \rightarrow z_c$ from above, for networks in the thermodynamic limit $N \rightarrow \infty$. The average $\langle \cdot \rangle$ is taken over the ensemble of grown networks. Clearly, $\beta > 0$ indicates a continuous transition, and it has been observed that $0 < \beta \ll 1$ for AP [21, 22].

The next reported exponent is κ , defined by

$$\langle \max(\Delta S) \rangle \sim N^{-\kappa}, \tag{2.3}$$

where $\max(\Delta S)$ is the largest jump in S upon the addition of a single edge during a network growth process. In a discontinuous phase transition, the maximum jump would approach a nonzero constant as $N \rightarrow \infty$, corresponding to $\kappa = 0$, but κ has

Growth Rule		z_c	ϕ	Cpt.	β	κ	ξ
$m = 1$	ER	1	$1/3$	GC	1	0.328(7)	0.30(9)
	DER	1	$1/3$	GOUT	1	0.329(3)	0.31(9)
				GSCC	2	0.64(7)	0.50(5)
$m = 2$	AP	1.7769(8)	0.5(0)	GC	0.0861(5)	0.0645(5)	—
	DCP	2.565(9)	0.44(1)	GOUT	0.34(5)	0.14(1)	0.12(9)
				GSCC	1.2(9)	0.55(8)	0.53(3)
$m = 3$	AP	1.92(9)	0.50(1)	GC	0.03(0)	0.020(7)	—
	DCP	4.86(1)	0.42(7)	GOUT	0.30(0)	0.10(5)	0.09(3)
				GSCC	1.(4)	0.40(7)	0.4(9)

Table 2.1: Critical exponents for each process (see text). For ER and DER, z_c , ϕ , and β are well-known exact results (see, for example, [6] and [7]). For AP with $m = 2$, values of z_c , ϕ , and β are reproduced from [21] and κ from [40]; refer to [21] for additional comments about the interpretation of ϕ . All other exponents listed above are derived from numerical simulations, as described below. Due to symmetry, results for GIN are identical to those for GOUT, and results for GBT are not listed because, in most cases, they are similar to those for GOUT.

also been observed to be small and positive for AP [23, 40].

Finally, this work introduces a third scaling exponent ξ , defined by

$$\max_z (\text{Var}[S]) \sim N^{-\xi} \quad (2.4)$$

for sufficiently large N . This is motivated by the observation in Ref. [21] that, for the Achlioptas process, the maximum variance of S_{GC} initially increases as N grows, then begins to decrease very slowly only when N is extremely large. This is related to other unusual finite-size effects in AP; see Ref. [21] for a thorough discussion. In a continuous transition, one expects that $\text{Var}[S] \rightarrow 0$ for all z in the thermodynamic limit, so $\xi > 0$. Moreover, a small value of ξ indicates that for finite N , there may be large changes in S near the critical point.

These observations suggest the use of the following descriptions for the critical behavior of percolation models for large but finite networks. However, I emphasize that these categories are merely useful heuristics for describing qualitative behavior, rather than precise definitions.

- Discontinuous: Cases in which $\beta = 0$, $\kappa = 0$, and $\xi = 0$.
- Explosive: Cases in which $0 < \beta \ll 1$, $0 < \kappa \ll 1$, and $0 < \xi \ll 1$.
- Weakly explosive: Intermediate cases which cannot be clearly designated as either “explosive” or “ordinary.”
- Ordinary: Cases in which β is on the order of 1 and κ and ξ are not small.

(In practice, a natural standard for comparison is ER, in which both κ and ξ are approximately $1/3$.)

In order to avoid confusion, I note that this terminology is not directly related to the language of [23], which distinguishes between “strongly” and “weakly” discontinuous transitions.⁴

Numerical estimation of both κ and ξ may be obtained by a straightforward fit of simulation data to the scaling laws Eqs. (2.3) and (2.4). This is illustrated in Figs. 2.3 and 2.4. In Figure 2.3, the scaling for (a) the GSCC (diamonds), (b) the GOUT (squares), and (c) the GBT (triangles) are compared in each panel to the results for undirected networks (circles). Figure 2.4 shows the scaling of the largest variance in (a) S_{GC} (circles) and (b) S_{GOUT} (squares) with N . In both cases, ER and DER (red) are compared to AP and DCP with $m = 2$ (blue) and $m = 3$ (green). Both figures use a simulation data set in which each point is averaged over many network growth trials (50 to 10,000, depending on m and N). Error bars (one standard deviation of the mean) are smaller than the point size except where shown. Solid lines are power-law fits, whose slopes are given as κ and ξ , respectively, in Table 2.1. On the other hand, the dashed lines in Fig. 2.4 merely connect the data points to guide the eye of the reader. No attempt is made to fit the maximum variance of S_{GC} to Eq. (2.4) because of the unusual finite-size behavior of AP noted above, in which the variance first increases and then slowly decreases.

Estimating the critical exponent β , as well as the critical point z_c , is more difficult than estimating κ and ξ . However, this may be accomplished by analyzing the finite-size scaling properties of the system. Finite-size effects in a continuous phase transition are important in a “critical region” in which the correlation length of the system, which diverges at the critical point, is comparable to or greater than

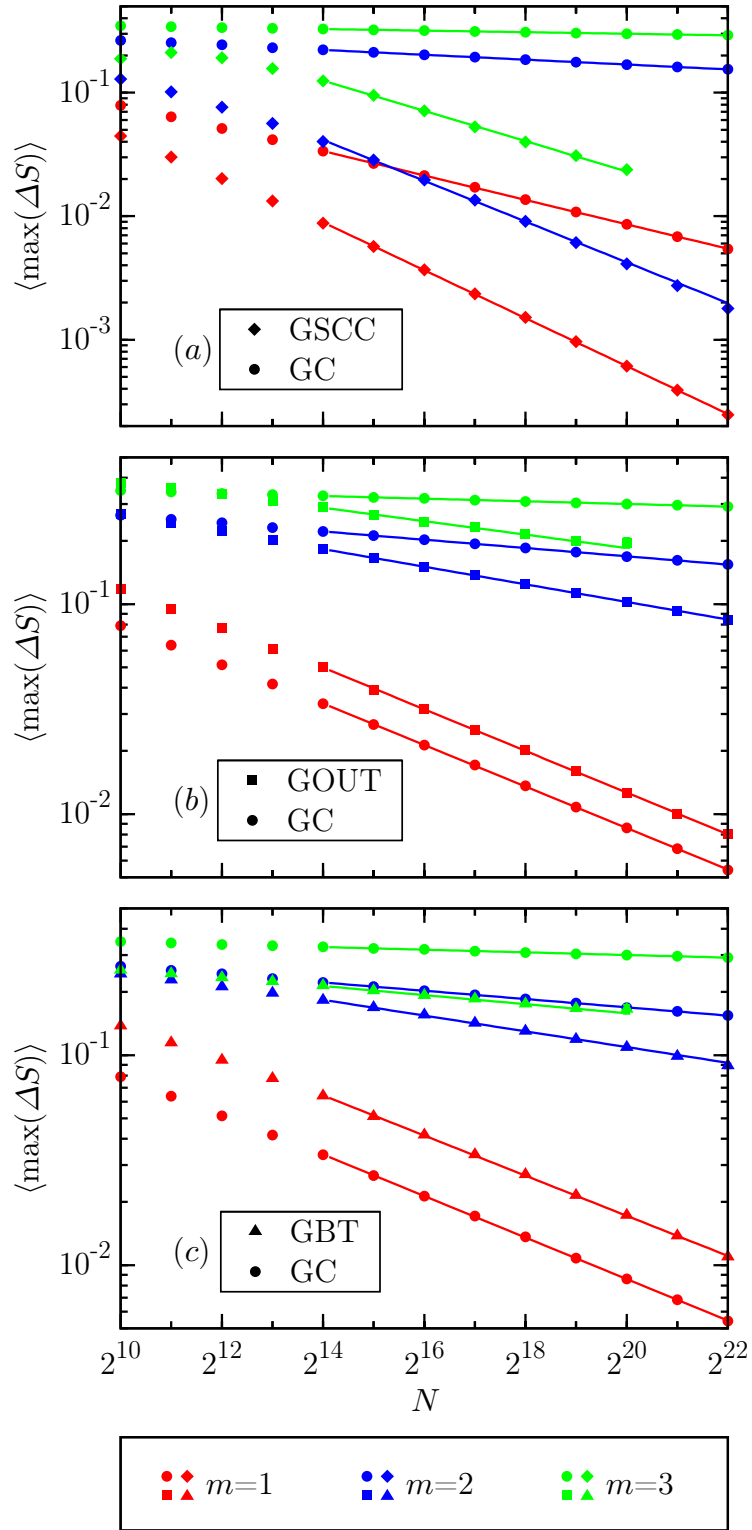


Figure 2.3: Scaling of the maximum jump size ΔS with N ; see text.

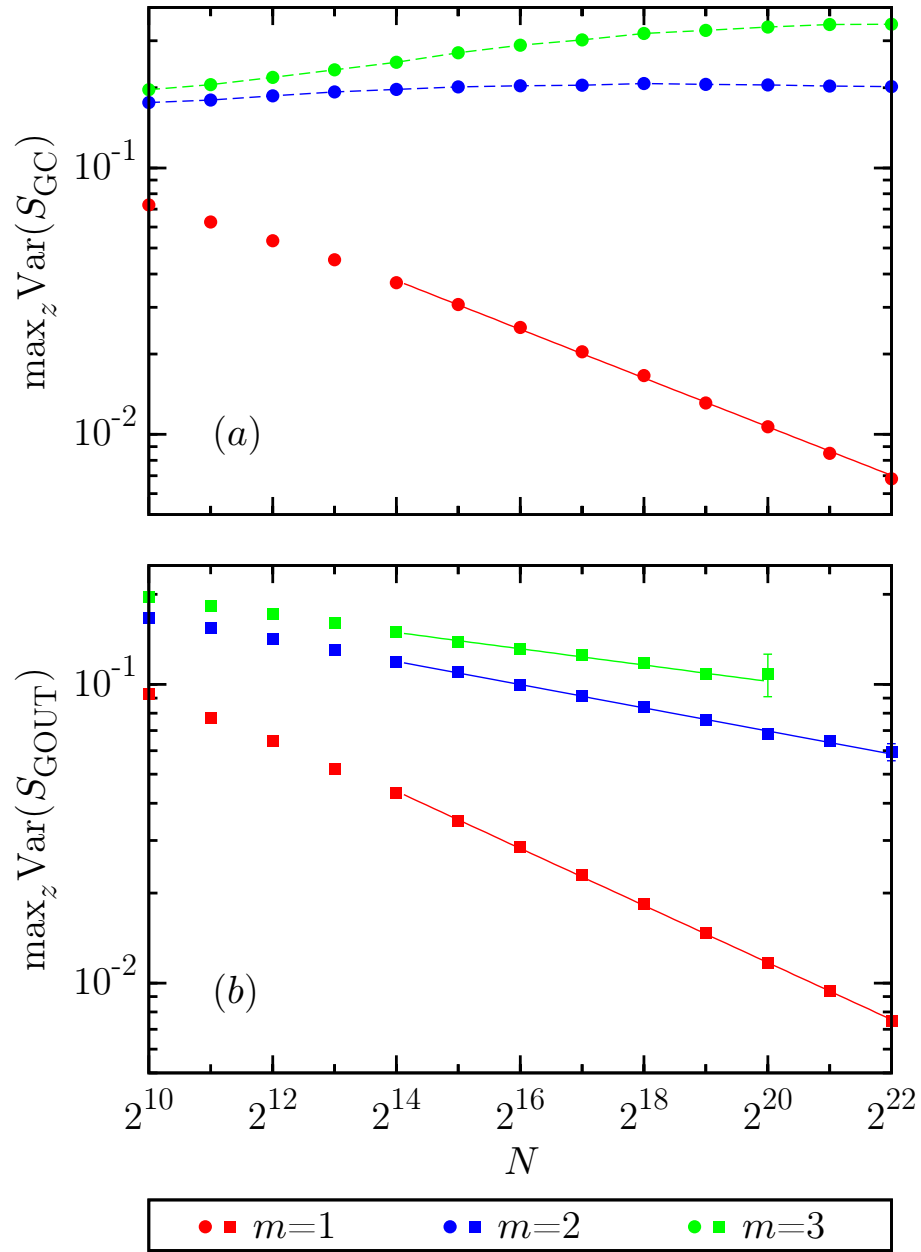


Figure 2.4: Scaling of the largest variance in (a) S_{GC} (circles) and (b) S_{GOUT} (squares) as a function of N . See text for details.

the system size. In the present case, the critical region scales asymptotically as $N^{-\phi}$, so the order parameter S is hypothesized to obey the finite-size scaling relation

$$\langle S \rangle = (z - z_c)^\beta g(N^\phi(z - z_c)), \quad (2.5)$$

where g is a universal scaling function [21, 42]. The function $g(w)$ has the properties that $\lim_{w \rightarrow -\infty} g(w) = 0$ and $\lim_{w \rightarrow \infty} g(w)$ exists and is positive, so that as $N \rightarrow \infty$, Eq. (2.2) holds for z slightly over z_c , and $\langle S \rangle = 0$ for $z < z_c$. The scaling function $g(w)$ also has a pole at $w = 0$, where $g(w) \sim w^{-\beta}$, which is necessary so that, at finite N , $\langle S \rangle$ may be nonzero at $z = z_c$, while $\langle S \rangle$ is nonetheless analytic for all z .

Equation (2.5) may be written in the equivalent form

$$\langle S \rangle = N^{-\beta\phi} h(N^\phi(z - z_c)), \quad (2.6)$$

by defining another universal scaling function $h(w) = w^\beta g(w)$ to remove the singularity at $w = 0$ [41, 42]. Equation (2.6) may be interpreted by saying that plots of $\langle S \rangle$ versus $w = N^\phi(z - z_c)$ for various large values of N will all collapse, when appropriately scaled, onto $h(w)$, when w is near 0 (i.e., $z \approx z_c$). The parameters β , ϕ , and z_c are chosen to optimize this data collapse; see Fig. 2.5. Specifically, β , ϕ , and z_c are chosen to minimize the function

$$V(\beta, \phi, z_c) = \frac{1}{\Delta w} \int_{-\Delta w}^{\Delta w} \text{Var}_N [N^{\beta\phi} \langle S(w, N) \rangle] dw, \quad (2.7)$$

which is zero when the data collapse perfectly onto a single curve. For further details, see Ref. [42]. Since there is no straightforward way to estimate the range of validity of Eq. (2.6), which also depends on N , Δw is treated as an external

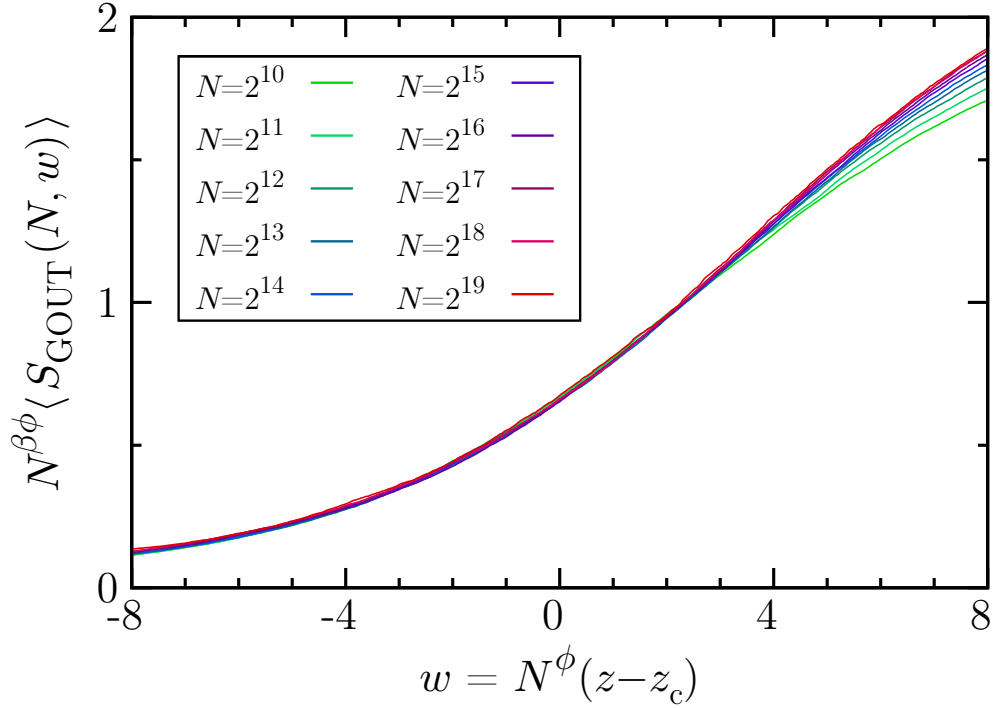


Figure 2.5: Collapse of $\langle S_{\text{GOUT}} \rangle$ for DCP ($m = 2$) using various values of N onto the universal scaling function $h(w)$, according to Eq. (2.6). From the bottom curve to the top curve, N increases from 2^{10} (green) to $N = 2^{19}$ (red). The first eight curves are averaged over 10,000 network growth processes, and the last two are averaged over 5,000 and 2,500 respectively. For these values of N , the collapse is excellent up to $w \approx 4$. Similar collapses are used to fit the values of z_c , ϕ , and β reported in Table 2.1.

parameter in Eq. (2.7) and Δw is chosen to give the smallest minimum in V . For all values of m and all components in Table 2.1, Δw was between 1 and 3.

The results in Table 2.1 summarize the important features of DCP and how they relate to both DER (the analogous non-explosive case) and AP (the analogous undirected case). For the GSCC, β and κ are lower in DCP than in DER, but are not small enough to lead to interesting behavior; therefore, I will focus on GOUT from here forward. It is clear that β and κ are significantly smaller in DCP than in DER, but not nearly as small as in AP. This provides quantitative support for the characterization of DCP as weakly explosive, in contrast to both the explosive behavior of AP and the ordinary behavior of DER. It is clear that DCP belongs somewhere between these two previously-studied regimes.

Several other features of Table 2.1 are worth noting. For example, in the Achlioptas process, β and κ change quite significantly when m is changed from 2 to 3, but the corresponding changes for DCP are comparatively small. This suggests again that the amount of edge competition has a more pronounced effect on the critical behavior of undirected networks than directed networks. However, the opposite is true of the critical point z_c , which, for successive values of m , increases by a much greater factor for directed networks than for undirected networks. If one views the purpose of edge competition as *delaying* the formation of a giant component rather than producing an explosive transition, then this goal is better achieved by DCP than by AP.

Finally, in Fig. 2.4, it can be seen that DCP lacks some of the unusual scaling behavior observed for AP in Ref. [21]. Although the values of ξ for the giant

out-component in DCP are smaller than those for DER, again indicating weakly explosive behavior, it is nonetheless clear that they are positive. In AP, a much more detailed analysis is required to show that $\text{Var}[S]$ eventually approaches 0 for all z as $N \rightarrow \infty$ [21], so a value for ξ in AP is not reported here.⁵ However, the qualitative differences between AP and DCP are quite clear.

2.4 Discussion

This chapter shows that an extension of the Achlioptas process to directed networks exhibits critical behavior which is, in many respects, partway between classical percolation and explosive percolation, and has been termed weakly explosive percolation here. This has several interesting ramifications for future research on controlling or modifying percolation phase transitions. One fundamental open question is how general the phenomenon of explosive percolation is, and whether the explosiveness of a percolation process can be predicted in a relatively straightforward way. From the perspective of classical percolation, the primary distinguishing features of the Achlioptas network growth process are that it is irreversible [22] and uses nonlocal information [21]; however, there are clearly such processes which are not explosive (see, for example, [43]). The strong explosiveness of the Achlioptas process may be contingent on several factors, and the present work suggests that the use of undirected networks is one of these factors.

Another avenue for further research is the possibility of tailoring percolation transitions with particular features. For example, different growth rules may create

different complex network structures. In Figure 2.2, nearly all network nodes have joined the giant bow-tie soon after the critical point, but this is not true of the giant in- or out-components until z is quite large.⁶ While it is beyond the scope of this chapter to investigate this feature, it suggests that there is additional interesting structure in networks grown through the directed competition process which cannot exist in undirected networks. More importantly, it may be possible to control the critical point and the critical behavior of the giant component by using a mix of directed and undirected edges in the network growth process. Because the Achlioptas process produces a more explosive transition, but the directed competition process delays the onset of criticality for longer, this may produce some degree of control for both features. Along with the above results, this suggests that further study of competitive percolation processes on directed networks will widen the known repertoire of percolation behavior in fascinating ways.

Notes for Chapter 2

¹Achlioptas et al. present another rule in which the sum, rather than the product, of the component sizes are used. Since they found similar results for the two cases, I refer only to the product rule.

²Note that the full bow-tie of i is not equivalent to the weakly connected component of i , which is the component to which i would belong if all edges in the network were undirected. This difference can be illustrated by a sample directed network of three nodes in which the edges are $1 \rightarrow 2$ and $3 \rightarrow 2$; node 3 is in the weakly connected component of node 1 but not in $BT(1)$. All percolation properties of weakly connected components on directed networks are equivalent to those of components on undirected networks, so they are not studied here.

³One consequence of this definition is that, while the size of the giant component in undirected percolation must increase monotonically as edges are added, this is not always true here. Because GIN, GOUT, and GBT are defined in terms of the GSCC, their sizes may decrease if, for example, a strongly connected component with a relatively large GIN grows to overtake the GSCC and becomes the new GSCC. These events are relatively rare but do occur.

⁴In Ref. [23], “strongly discontinuous” transitions have a jump discontinuity ($\beta = 0, \kappa = 0$) and “weakly discontinuous” transitions are pointwise continuous but contain supralinear growth ($0 < \beta < 1, 0 < \kappa < 1$). Here, the categorization between explosive and weakly explosive are used instead, because the emphasis is on behavior which depends on how small the critical exponents are, rather than whether or not they are nonzero.

⁵The exponent ξ is related to the exponent η_+ considered in Ref. [21]. Comments in that work suggest that ξ should have a value of approximately 0.04 for the $m = 2$ case, but an explicit value is not reported there.

⁶In DER, one can show that $S_{\text{GBT}} = S_{\text{GOUT}}(2 - S_{\text{GOUT}})$ [10], but the behavior of S_{GOUT} and S_{GBT} in Fig. 2.2 is quite different. Compared to DER, nodes in DCP are much more likely to be in either GIN or GOUT but not in both (that is, not in the GSCC) when z is not too far above z_c .

Chapter 3: Dynamical instability in Boolean networks as a percolation problem

3.1 Introduction

In this chapter, I discuss how the order parameter for the phase transition in the dynamical stability of a Boolean network can be mapped onto the order parameter for a related percolation phase transition [2]. One primary advantage of this technique is that it is valid for, and can be used to make predictions for, specific networks with complex topology. In contrast, previously existing techniques have mostly focused on the average behavior of ensembles.

As discussed in Section 1.5, in a Boolean network, each node is assigned a state, 0 or 1, which is synchronously updated at discrete time steps according to a pre-assigned update function which depends on the states of that node's inputs on the previous time step. That is, each node i is assigned a state, $x_i(t) = 0$ or $x_i(t) = 1$, at each discrete time step t , according to

$$x_i(t+1) = F_i(X_i(t)), \quad (3.1)$$

where F_i is a Boolean function which depends on the combined state of the d_i^{in}

inputs to node i ,

$$X_i(t) = \left(x_{j_1}(t), \dots, x_{j_{d_i^{\text{in}}}}(t) \right). \quad (3.2)$$

One common choice of the functions F_i are biased functions, in which each output is independently and randomly selected to be $F_i(X_i) = 1$ with some bias probability, b . As discussed in Section 1.6, it is well-known that such networks undergo a phase transition from stable to unstable dynamics. The order parameter for this phase transition is Y , which is defined in Eq. (1.12) as the average normalized Hamming distance between pairs of initially close trajectories $x(t)$ and $\tilde{x}(t)$.

A derivation of the critical in-degree was given by Derrida and Pomeau for N - K networks using an “annealed approximation” [44], which is discussed in more detail in Section 3.2.1. An “annealed” system is one in which the network edges and update functions are randomly re-drawn between time steps, as opposed to the original “quenched” system under study, in which the network and update rules are fixed. Derrida and Pomeau hypothesized that for large networks the stability properties of the annealed system are similar to those of a typical quenched system. This hypothesis, called the “annealed approximation,” is well-supported by numerical experiments [44, 45].

Recent work in Ref. [17] has extended this approach by using a partial randomization, in which only the update functions (but not the network topology) are randomly generated at each time step. In contrast with the annealed approximation, this “semi-annealed” approximation describes the dynamics on a fixed network which may have nontrivial topological features such as edge assortativity [46], mo-

tifs [47], and community structure [48]. The only necessary assumption is that the network must satisfy a locally tree-like condition discussed in Section 3.3.1.

Some recent papers have derived stability properties of Boolean networks without annealing [49, 50]. These papers are complementary to this work in the following sense. Although rigorous, their results only apply to the ensemble average of random networks with restrictions on their network topology and/or update functions. In contrast, because these results rely on the semi-annealed approximation, they can model the dynamics of a specific network.

Here, using a semi-annealed approach, the *dynamical* problem of stability on a Boolean network is mapped onto the *static* problem of network percolation in the $N \rightarrow \infty$ limit. Previous authors have discussed the percolation properties of the “frozen component” of N - K networks [51–53], and others have used percolation to discuss the stability of N - K lattices [54, 55]. In contrast, this chapter shows that a dynamic quantity, the long-time average Hamming distance between two initially close trajectories on a Boolean network, can be mapped onto the size of the giant out-component in a percolation problem. This map will be illustrated in three different contexts. First, the well-known annealed approximation is mapped onto site percolation in the configuration model for networks with biased update rules in Section 3.2. Second, the semi-annealed approximation is similarly mapped onto weighted site percolation for a given complex network in Sections 3.3.1–3.3.2. Finally, another biologically interesting class of update rules—canalizing functions—is treated by mapping to a correlated bond percolation problem in Section 3.3.3.

3.2 Mapping the annealed approximation onto site percolation

3.2.1 The annealed approximation

Derrida and Pomeau’s original statement of the annealed approximation was derived in the context of N - K networks with biased update rules [44]. In that context, an annealed system is one in which between *every* time step of the dynamics, each node is given K inputs selected randomly from among the $(N - 1)$ other nodes in the network, along with a new biased update rule F_i using the bias probability b . Note that, although the inputs and update rules are selected randomly, the same sequence of inputs and update rules are used for both of the trajectories $x(t)$ and $\tilde{x}(t)$ which are considered in the stability problem. The annealed approximation is Derrida and Pomeau’s conjecture that, in the large N limit, the stability boundaries for the quenched and annealed situations are approximately the same.

The stability of this case can be evaluated in the following way. First, define a node i to be “damaged” at time t if $x_i(t) \neq \tilde{x}_i(t)$, i.e., if its state differs on the original and perturbed systems. (Note that the order parameter Y may be interpreted as the average amount of damage in the steady state of the system.) Next, consider the probability of damage spreading to a node i from one of its inputs j , that is, $\Pr[x_i(t+1) \neq \tilde{x}_i(t+1) | x_j(t) \neq \tilde{x}_j(t)]$. This quantity is typically called the sensitivity associated with the bias b and is denoted q [56]. Since each output value of F_i is determined randomly and independently, q is just the probability that $F_i = 0$ for one set of inputs and $F_i = 1$ for another. This can occur in two ways, each with

probability $b(1 - b)$, so

$$q = 2b(1 - b). \quad (3.3)$$

Because each node has K randomly selected inputs, one expects that, the number of damaged nodes created by a single damaged node in the network is qK . When this quantity is greater than 1, the number of damaged nodes in the network initially grows, indicating instability, so the critical boundary separating the stable and unstable regimes is determined by $qK = 1$.

In this framework, the value of the order parameter Y can be predicted analytically in the following way [44]. A node will become damaged with probability q if it at least one of its K inputs is damaged. Therefore, the average amount of damage, Y , reaches a steady state when

$$Y = q (1 - (1 - Y)^K), \quad (3.4)$$

which is straightforward to solve numerically. Nonzero solutions for Y occur only when $qK > 1$, as expected.

Since its introduction, the annealing approach has been generalized in a number of ways to apply to systems with a distribution of in-degrees [57–61], joint in-degree/out-degree distributions [62, 63], canalizing update rules [56, 64–67], and threshold update rules [68–71]. This work focuses on the treatment of biased update rules in networks specified by joint in-degree/out-degree distributions, where a_{kl} is the probability that a node i has in-degree $d_i^{\text{in}} = k$ and out-degree $d_i^{\text{out}} = l$. For simplicity, it is assumed that the bias of the update rules depends only on degree, so the bias of such a node is b_{kl} , with an associated sensitivity $q_{kl} = 2b_{kl}(1 - b_{kl})$.

An annealed system with a specified degree distribution means that each node is assigned a degree, which is fixed, but its edges are re-drawn randomly from the set of all configuration-model networks (see Section 1.2) between each time step.

In this scenario, Y can still be predicted analytically using a similar technique as above, first derived in Ref. [62]. Let ψ denote the probability that a randomly selected edge originates from a damaged node; this definition is necessary here, but not in the case of N - K networks, because of the heterogeneity in the degree distribution. A randomly selected edge originates from a node with k inputs and l outputs with probability $\frac{la_{kl}}{z}$, where z is the average degree. A node will become damaged with probability q_{kl} if it has at least one damaged input, which occurs with probability $1 - (1 - \psi)^k$. Therefore,

$$\psi = \sum_{k,l} \frac{la_{kl}}{z} q_{kl} \left[1 - (1 - \psi)^k \right], \quad (3.5a)$$

$$Y = \sum_{k,l} a_{kl} q_{kl} \left[1 - (1 - \psi)^k \right]. \quad (3.5b)$$

In the stable regime, these equations only have the trivial solution $\psi = 0$ and $Y = 0$, but there will be a nonzero solution in the unstable regime [62].

3.2.2 The map to site percolation for network ensembles

I now show that Eq. (3.5) can be mapped onto the generating function formalism for treating site percolation in directed configuration-model networks, as developed in Refs. [10] and [72]. In this model, each node is deleted with some probability which depends only on its degree. The resulting ensemble of site-deleted networks exhibits a percolation phase transition similar to that of directed Erdős-

Rényi networks, as described in Section 1.4. Above the percolation threshold, a giant strongly connected component (GSCC) of mutually path-connected nodes forms; this, along with all nodes that can be reached from it, is called the giant out-component (GOUT). In this map, the probability that a node is *not* deleted will be identified with the sensitivity. That is, q_{kl} will stand for both the sensitivity and the non-deletion probability for a node with k inputs and l outputs. With this identification, I will show that Y maps onto the expected fraction of nodes in GOUT, S_{GOUT} , which in this chapter is simply denoted S .

It is shown in Refs. [10] and [72] that S can be found as follows. The first step is to define modified generating functions for the in-degrees of nodes and edges,

$$G_0(w) = \sum_{k,l} a_{kl} q_{kl} w^k, \quad (3.6a)$$

$$G_1(w) = \sum_{k,l} \frac{l a_{kl}}{z} q_{kl} w^k. \quad (3.6b)$$

As above, the coefficient $\frac{l a_{kl}}{z}$ on terms in $G_1(w)$ is the probability that a randomly selected edge originates from a node with in-degree k and out-degree l . Note that while many generating functions are normalized so that $G(1) = 1$, in this case, because of the q_{kl} term, $G_0(1) < 1$ and should be interpreted as the fraction of nodes which are undeleted; similarly, $G_1(1) < 1$ and should be interpreted as the fraction of edges which originate at undeleted nodes.

A variety of percolation-related features may be derived using G_0 and G_1 , including the full distribution of component sizes [10, 72]. Below, only the size of GOUT is derived. Let Γ be the probability that a randomly selected edge is not in GOUT (i.e., its parent node is not in GOUT). A self-consistent equation for

Γ may be obtained by noting that there are two cases in which an edge will not be in GOUT. The first case is if it comes from a deleted node, which occurs with probability $1 - G_1(1)$. The second is if its parent is an undeleted node with k inputs (with probability $\sum_l \frac{la_{kl}}{z} q_{kl}$), all of which are not in the giant out-component (with probability Γ^k). This results in Eq. (3.7). Similarly, a node is in GOUT if it is not deleted and it is not true that all of its input edges are not in GOUT. Therefore,

$$\Gamma = 1 - G_1(1) + G_1(\Gamma), \quad (3.7)$$

$$S = G_0(1) - G_0(\Gamma). \quad (3.8)$$

Once the mathematical and conceptual parallels between the derivations of Eqs. (3.5) and (3.6-3.7) have been noted, it is straightforward to show that there is a map between these two cases, which I give below. The spread of damage from node to node in Boolean networks is similar to the transmission of the state of “being in GOUT” from node to node in percolation. However, since damage only spreads with a given probability q_{kl} , the appropriate percolation process involves site deletion, which prevents a node from being in GOUT. As long as the site non-deletion probability is identified with the network sensitivity, then the substitutions

$$\psi = 1 - \Gamma \quad (3.9)$$

$$Y = S \quad (3.10)$$

map Eq. (3.5) onto Eqs. (3.6-3.7). Therefore, the phase transition between dynamical stability and instability in this ensemble of random Boolean networks is equivalent to the static percolation phase transition on a site-deleted network ensemble.

3.3 Mapping the semi-annealed approximation onto percolation

3.3.1 The semi-annealed approximation on locally tree-like networks

This section summarizes a more powerful stability analysis technique known as semi-annealing, which was introduced and analyzed in Ref. [17]. For now, the discussion is restricted to biased update rules, but these are generalized so that the bias may be arbitrarily selected for each node, i.e., a different bias b_i assigned to each node i , with a corresponding sensitivity $q_i = 2b_i(1 - b_i)$. In the semi-annealed approximation, the network topology is considered to be fixed, but the update rules F_i are randomly re-drawn between time steps.

In order to make the analysis of the semi-annealed system analytically tractable, it is convenient to assume that any correlations between the states of two network inputs j_1 and j_2 to the same node i are typically close to zero. This assumption is supported by numerical simulations in cases where the network structure is locally tree-like. Here, locally tree-like means that if two nodes j and i are connected by a short directed path, it is very unlikely that there will exist a second such path.¹ Almost all commonly studied Boolean networks are locally tree-like, including, for example, all sparse random networks in the configuration model with large N [10]. In locally tree-like networks, the states of two inputs to a node are approximately uncorrelated, because correlations arise when two nodes are both influenced by a common ancestor. Analyses based on this approximation have been found to yield accurate results, even in cases where the network contains significant clustering

[73, 74] (including for Boolean networks [17]).

Reference [17] derived a damage-spreading condition for specific, complex, locally tree-like Boolean networks with semi-annealed, biased update rules. This derivation may be summarized as follows. First, $y_i(t)$ is defined to be the probability that a node i is damaged at time t ,

$$y_i(t) = \langle |x_i(t) - \tilde{x}_i(t)| \rangle_{\text{i.c., dyn.}}, \quad (3.11)$$

where the average $\langle \cdot \rangle_{\text{i.c., dyn.}}$ is over all possible initial conditions with initial Hamming distance ε (see Section 1.6) and all stochastic realizations of the semi-annealed dynamics. The time evolution of $y_i(t)$ can be derived by a more sophisticated version of the argument leading to Eq. (3.5). The conditional probability that a node i will become damaged at a time $t + 1$, given that it has at least one damaged input at time t , is q_i . Moreover, the probability that node i has a damaged input at time t can be expressed using the values $y_j(t)$, assuming that the damage probabilities are independent, as described above. Multiplying these gives $y_i(t)$. This can be expressed most simply by introducing the notation \mathcal{J}_i for the set of inputs to node i , giving

$$y_i(t + 1) = q_i \left[1 - \prod_{j \in \mathcal{J}_i} (1 - y_j(t)) \right]. \quad (3.12)$$

The probability of any node being damaged in the initial conditions is ε , so the time-evolution of $y(t)$ can be found by solving Eq. (3.12) with initial conditions $y_i(0) = \varepsilon$.

Appendix 3.C argues that, in cases of interest, Eq. (3.12) approaches a single unique stable fixed point as $t \rightarrow \infty$, representing the expected damage probabilities

in the steady state of the system. Here these steady-state values are denoted by dropping the time-dependence of y_i . By Eq. (3.12), these must satisfy the consistency conditions

$$y_i = q_i \left[1 - \prod_{j \in \mathcal{J}_i} (1 - y_j) \right]. \quad (3.13)$$

This is the analogue of Eq. (3.5) for a specific node in a complex network. Equation 3.13 defines a system of N equations and N unknowns which can, in practice, be solved for y_i simply by iterating Eq. (3.12) until convergence is achieved. Since

$$Y = \langle y_i \rangle \quad (3.14)$$

in the semi-annealed approximation, this technique can be used to solve for Y in specific, complex locally tree-like Boolean networks.

3.3.2 The map for specific networks with biased update rules

The value of Y in the semi-annealed approximation can be mapped to the size of GOUT in a site percolation problem. In particular, I show that it can be mapped onto the results obtained in Ref. [73] for weighted site percolation in locally tree-like directed networks. The sensitivity q_i will once again be identified with a site non-deletion probability. Independently deleting each node i with probability $(1 - q_i)$ generates an ensemble of site-deleted networks. In Ref. [73], η_i is defined as the fraction of networks in this ensemble for which node i is *not* in GOUT. It is shown that η_i must satisfy the equations

$$\eta_i = 1 - q_i + q_i \prod_{j \in \mathcal{J}_i} \eta_j, \quad (3.15)$$

because a node is not in GOUT when it is either deleted (with probability $1 - q_i$), or when it is not deleted but has no inputs from GOUT (the final term above). The expected size of GOUT, S , is then given by

$$S = \langle 1 - \eta_i \rangle_i \quad (3.16)$$

Equations (3.13-3.14) may be mapped onto Eqs. (3.15-3.16) through the simple substitution

$$\eta_i = 1 - y_i \quad (3.17)$$

$$Y = S \quad (3.18)$$

This map subsumes the map described in Eq. (3.9), and has the additional advantage of being applicable to networks with complex topology.

3.3.3 The map for specific networks with canalizing update rules

Finally, I consider the case of Boolean networks with “canalizing” update rules, in which one input acts as a master switch for the update rule. That is, an input j to node i is canalizing if there is a state of x_j which completely determines the value of F_i independent of the other inputs to i . When x_j is not equal to its canalizing value, F_i depends on the states of its other inputs. Canalizing functions are thought to be common in real gene networks [64, 75].

The methods used above can be extended to canalizing functions, but because the update rule elements in a canalizing function are not generated independently, it is necessary to consider a new type of percolation problem, which I will call

correlated bond percolation. Unlike typical bond percolation, which treats the case in which each bond is considered individually for deletion, the deletion of two bonds may be correlated if they are inputs to the same node. Here follows the derivation of a correlated bond percolation problem that corresponds to a Boolean network whose update rules each have one canalizing input but are otherwise generated randomly. That is, for each node i , there is a canalizing input c_i , and all the rows of the update rule on which x_{c_i} assumes its canalizing value have the same constant output; but the outputs of the other rows are randomly generated with a probability bias b_i .

In the derivation below, I assume that the system is approximately equally likely to be in any of its states. This assumption will be revisited in Chapter 4, which correctly accounts for the fact that the system is more likely to be in some states than others. For the biased and canalizing update rules studied in this chapter, however, this is a reasonable first approximation. As shown below, it is then formally possible to obtain equations describing damage spreading in closed form for canalizing update rules. Based on the numerical results, I conjecture that these equations can be used to predict damage spreading in a large class of Boolean networks with frozen update rules.

An expression for y_i for canalizing update rules may be derived as follows. Let r_i denote the “activity” of c_i on i [56], defined here as the fraction of states in which i will become damaged if c_i becomes damaged. If c_i is not damaged, it may be in either the canalizing or non-canalizing state, each with probability $\frac{1}{2}$. In the first case it is impossible for i to become damaged, while the second case is equivalent

to Eq. (3.13). Therefore,

$$y_i = r_i y_{c_i} + \frac{1}{2} q_i (1 - y_{c_i}) \left[1 - \prod_{j \in \mathcal{J}'_i} (1 - y_j) \right], \quad (3.19)$$

where $\mathcal{J}'_i = \mathcal{J}_i - \{c_i\}$ and q_i is the sensitivity of the update rule when x_{c_i} is not in its canalizing state. As above, it can be shown that this is equivalent to

$$\eta_i = 1 - r_i + \left(r_i - \frac{1}{2} q_i \right) \eta_{c_i} + \frac{1}{2} q_i \prod_{j \in \mathcal{J}_i} \eta_j, \quad (3.20)$$

with the substitution $\eta_i = 1 - y_i$. This corresponds to a correlated bond percolation problem in which one of the following three things may occur. With probability $1 - r_i$, all edges to i are deleted; with probability $r_i - \frac{1}{2} q_i$, all of i 's edges are deleted except for the edge from c_i ; and otherwise no input edges are deleted.² Note that it is straightforward to describe the case where only some of the nodes have a canalizing input by using Eqs. (3.19-3.20) for those nodes and Eqs. (3.13-3.15) for the others.

3.4 Numerical results

Figures 3.1 and 3.2 demonstrate the results of the map described in Section 3.3.2. (Numerical results for the map of Section 3.2.2 are not shown, since it averages the map in Section 3.3.2 over a random network ensemble.) The long-time average Hamming distance Y is compared to the size of the giant out-component S for both ensembles of random networks as well as particular complex networks. Both Y and S are also compared to the theoretical prediction given by the solution to Eq. (3.13), which is denoted T .

Numerical results are found as follows. First, a configuration-model network with $N = 10^5$ nodes is generated. Networks used in the data in the figures have no

degree correlations, $a_{kl} = a_k^{\text{in}} a_l^{\text{out}}$, where the in-degree distribution a_k^{in} is a Poisson distribution and the out-degree distribution a_l^{out} follows a truncated power law with a cutoff of 50. If desired, interesting topological features such as assortativity are then enhanced using the same algorithms as in Ref. [17]. For assortativity, this entails randomly selecting pairs of edges, $j_1 \rightarrow i_1$ and $j_2 \rightarrow i_2$, and exchanging their outputs (i.e., replacing them with the edges $j_1 \rightarrow i_2$ and $j_2 \rightarrow i_1$) if doing so would change the assortativity in the desired direction. I have also tested networks with other degree distributions and other biologically motivated features, such as feedforward motifs, and found similar results. Next, each node is assigned a bias b_i . These may be distributed randomly, or, to encourage (impede) instability on the network, they may be distributed so that the nodal average $\langle q_i d_i^{\text{in}} d_i^{\text{out}} \rangle$ is maximized (minimized); see Ref. [17] for more details. For the data in the figures, the biases b_i were distributed randomly so that the sensitivities q_i form a uniform distribution on the interval [.3, .5], but other choices also give similar results.

In the Boolean network dynamics, the initial conditions for x are chosen randomly, and a randomly selected fraction $\varepsilon = .01$ of the nodes are damaged for the initial conditions of \tilde{x} . To find Y , the system is time-evolved and $|x_i(t) - \tilde{x}_i(t)|$ is averaged between $t = 900$ and $t = 1000$, also averaging over 100 initial conditions. The theoretical prediction is found by iterating Eq. (3.12) until it converges to a solution y^* , then taking $T = \langle y_i^* \rangle$. Finding S is less straightforward, because a typical percolation problem is only guaranteed to have a single, well-defined giant out-component in the $N \rightarrow \infty$ limit. For reasons discussed in Appendix 3.A, the following procedure is used. Each node i is deleted with probability $1 - q_i$ and any

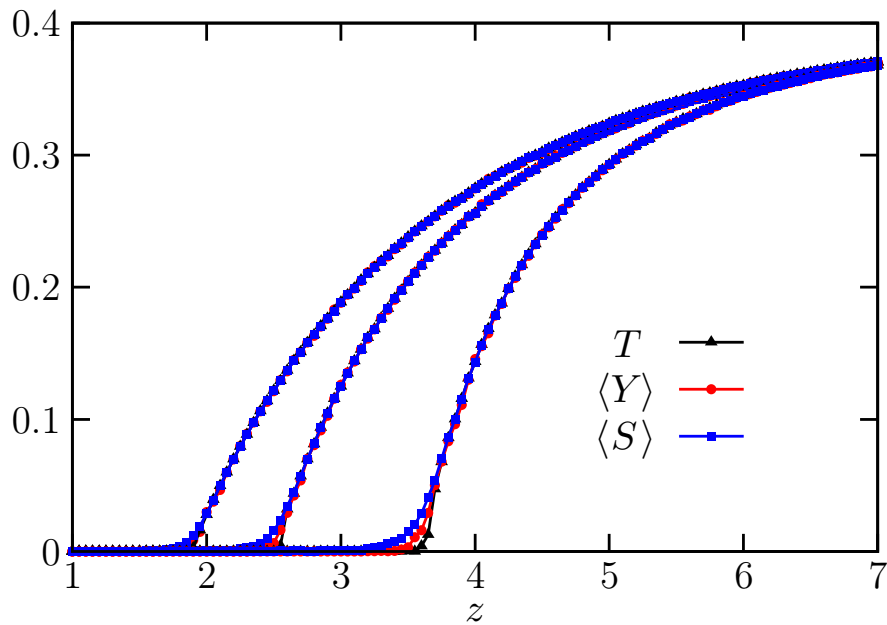


Figure 3.1: The ensemble averages of Y (red), S (blue), and T (black) versus the average degree z , for three families of networks. The three families of networks are assortative (left), neutral (middle), and disassortative (right). Each point is averaged over 20 networks. The mean assortativity of networks in the three families are $\langle \rho \rangle = 1.26, 1.00,$ and 0.76 , respectively, where ρ is defined as in Eq. (1.5). Both Y and S agree well with T , except for finite-size effects near the critical point.

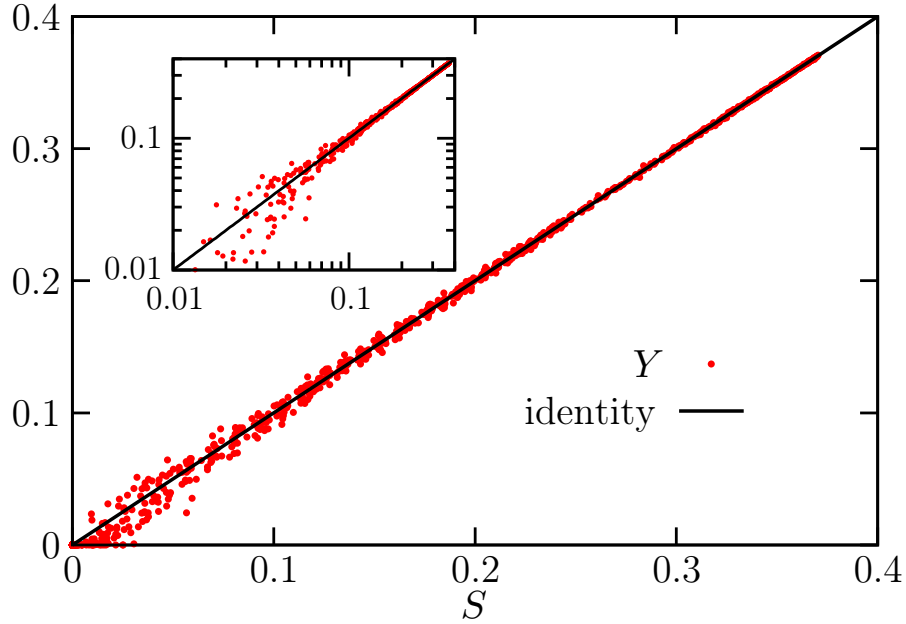


Figure 3.2: Y versus S for individual neutrally assortative networks on linear and (inset) log-log axes.

strongly connected components (SCCs) in the resulting network are found, where an SCC is defined to be a mutually path-connected set of nodes containing at least one loop. The size of GOUT, S , is defined to be the fraction of nodes which can be reached from at least one of these SCCs, averaged over the ensemble of deletion trials. For each network, 10^3 deletion trials are averaged. The numerical uncertainty in the measured values of T , Y , and S for each point in each figure is smaller than the point size; see Appendix 3.B for details.

Figure 3.1 illustrates the relationship between Y , S , and T for networks generated in this way, showing that Y and S have the same average values on the ensemble of random networks with given average degree z , up to finite-size effects near the transition. However, in Fig. 3.2, it can be seen that the prediction $Y = S$

sometimes fails for individual networks, especially near the phase transition. These deviations are primarily caused by the quenched disorder in the update rules, which may cause orbits to fall onto attractors which visit only a small fraction of the state space. For finite systems, this leads to deviations from Eq. (3.12), which is one motivation for the work in Chapter 4.

In Figure 3.3, each point has been averaged over this quenched disorder by choosing update rules from an ensemble of closely related frozen update rules (but not networks) as follows. Before time-evolving each new pair of initial conditions, a set of exchanges are performed on the update rules. For each edge $j \rightarrow i$, with probability $\frac{1}{2}$, $x_j = 0$ and $x_j = 1$ are exchanged in the update rule for i . Note that there are two major differences between this and the semi-annealed approximation. In the latter, the update rules are changed *during* the dynamics, whereas here they are only changed before each new dynamical trial. Second, whereas the semi-annealed approximation treats all inputs interchangeably, this procedure preserves input-specific information (such as whether an input is canalizing). In Figure 3.3, this procedure yields excellent agreement between Y , S , and T for individual networks well above the transition. Near the transition and below it, finite-size effects still cause S (and, to a lesser extent, Y) to deviate slightly from the prediction T . These effects are discussed further in Appendix 3.A.

Figure 3.4 shows the results of the same numerical experiment but for the case in which each node has one canalizing input. Here, Y , S , and T agree for individual networks if the map between Eqs. (3.19) and (3.20) is used, but the map between Eqs. (3.13) and (3.15) fails for this case. This indicates that significant

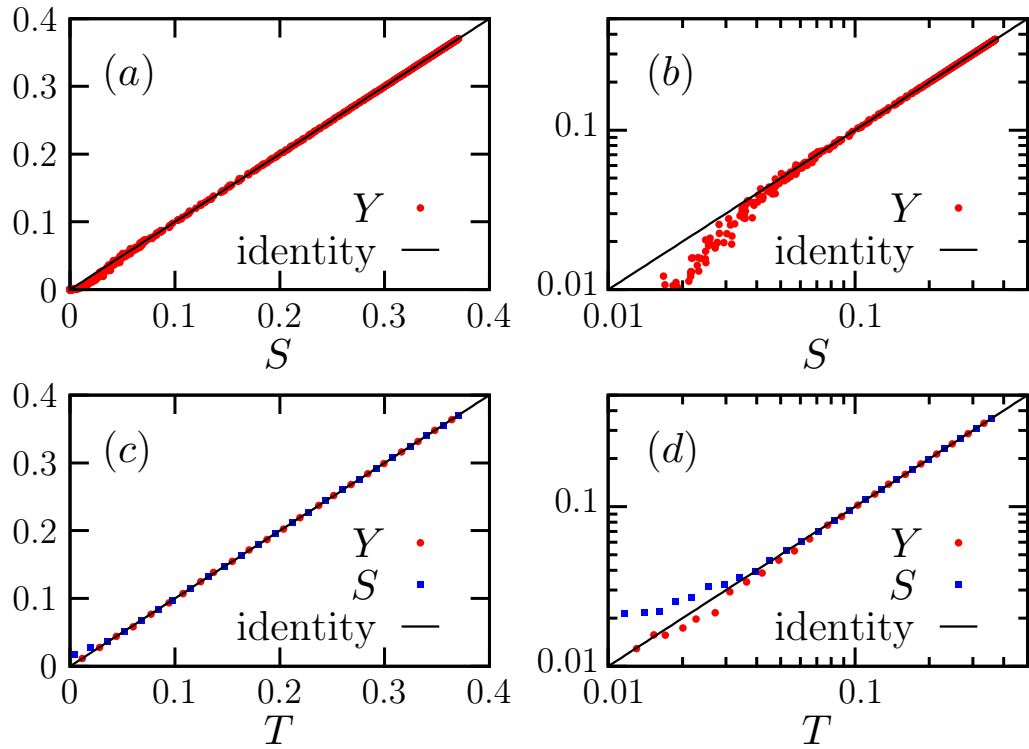


Figure 3.3: (a) Linear and (b) log-log scatterplots of Y versus S for data generated in the same way as that of Fig. 3.2, except that averaging over the quenched disorder in the update rules is applied, as described in the text. (c) Linear and (d) log-log scatterplots of Y (red) and S (blue) versus T for the same data, sampling alternate points for visibility.

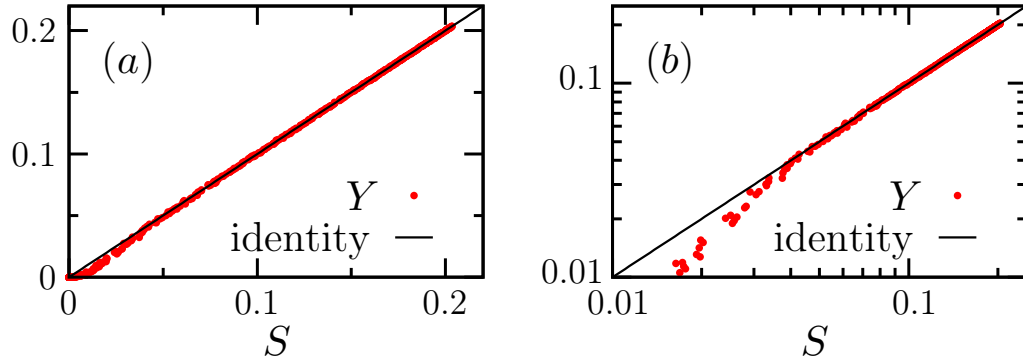


Figure 3.4: Y versus S on (a) linear and (b) log-log axes, for networks in which each node has one canalizing input, using the map described in Section 3.3.3.

input-specific information about the dynamics is retained when averaging over the quenched disorder in the update rules.

3.5 Discussion

The stability of a Boolean network can be understood in terms of a related percolation problem on that network, in which being in the giant out-component represents damage in the steady state, and node or edge deletion enables the treatment of cases in which a node with a damaged input does not become damaged. This relationship may be helpful in understanding the stability of systems modeled by Boolean networks, such as gene regulatory networks and neural networks. Two previously-studied cases (the annealed and semi-annealed approximations) map onto known results for percolation, and a case of biological interest (canalizing update rules) maps onto a novel percolation problem.

These maps are valid for the typical cases in the literature (large, locally tree-

like networks with biased or canalizing update rules), but have the advantage of applying to specific networks rather than ensembles of random networks. Numerical experiments show excellent agreement with the analysis when averaged over a family of quenched update rules. In Chapter 4, I address the fact that, contrary to the approximation used in this chapter, the system does not actually visit all states equally, which has important ramifications for systems with some types of update rules. It is possible that there is an improved map between stability and correlated bond percolation which correctly takes into account the results of Chapter 4. If so, it would be able to address systems with a much broader range of update rules than the maps presented in this chapter. At the same time, it would probably eliminate the need for the quenched averaging used above. However, it is unclear whether such a map exists, and this question is beyond the scope of the present work.

Appendices

In the following appendices, I discuss the role of finite-size effects in the results above and how they influence the choice of definition of S used in the main body of the chapter. I also discuss the tests used to confirm that the numerical simulations in this chapter are converged, and I give a brief argument that there is typically a unique stable solution to Eq. (3.13).

3.A Finite-size effects

In this section, I expand on two points from the above chapter. First, I explain the deviations of S and Y from T when T is small (see Fig. 3.3). Figure 3.5 shows numerically that these deviations are due to finite-size effects; it is identical to Fig. 3.3 except that it includes results for $N = 10^3$ and $N = 10^4$ as well, showing that the deviations of Y and S from the theory decrease as N increases. The origin of these finite-size effects is discussed at length below. Second, I explain the choice of algorithm for the numerical calculation of S , which is motivated by another finite-size effect for percolation.

Two significant finite-size effects for percolation must be considered:

- S1. A finite network near the percolation phase transition may not contain a single well-defined giant component, which forces a choice between multiple definitions for S .

- S2. The theory is derived using a locally tree-like approximation, but finite networks typically contain some simple loops that contribute to S but not T . This is the most evident finite-size effect in Fig. 3.3, where it causes $S > T$ when T is small.

Random networks in the limit $N \rightarrow \infty$ have a single macroscopic GSCC, whose out-component is the GOUT, but the above results are derived in the context of specific, finite-sized locally tree-like networks. Such networks, including finite random networks, may contain several strongly connected components with significant

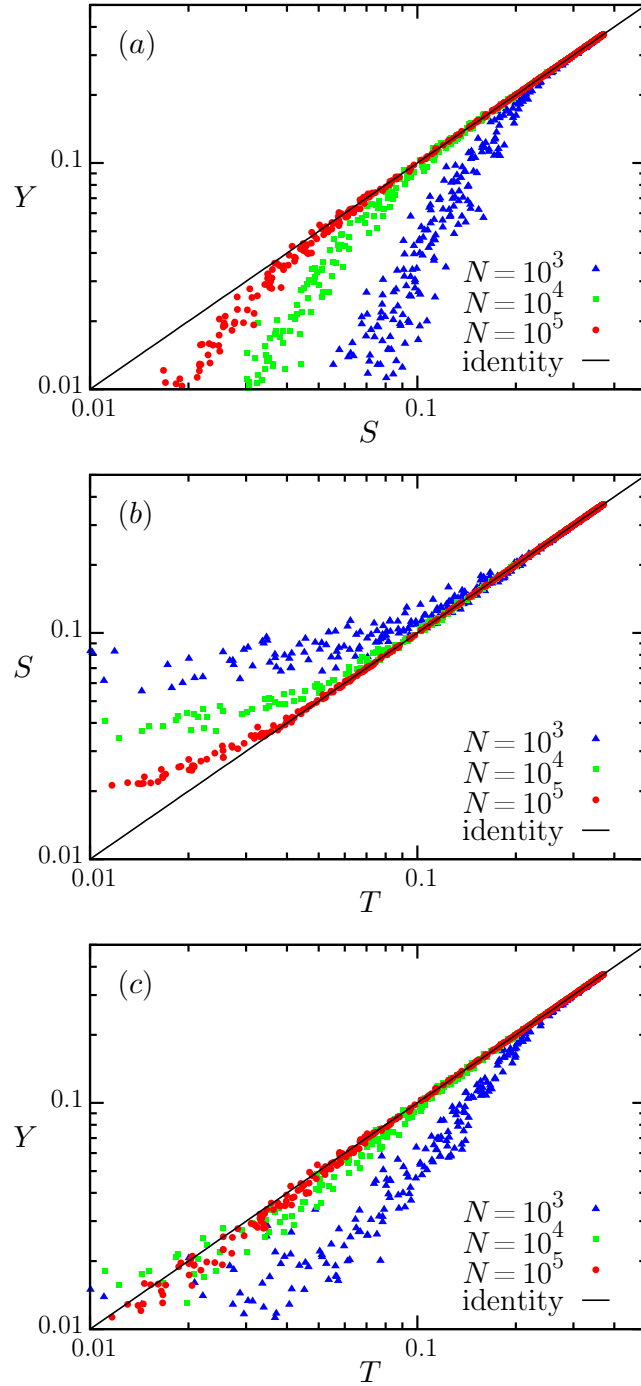


Figure 3.5: Log-log scatterplots of (a) Y versus S , (b) Y versus T , and (c) S versus T for increasing N . The data approach $Y = S = T$ as N grows. The data for $N = 10^3$ and $N = 10^4$ are generated as in Fig. 3.3, but more initial conditions and node deletion trials are used to ensure convergence of Y and S .

out-components, particularly near the percolation phase transition. Above the transition, many of these SCCs join into a single giant component which scales as $\mathcal{O}(N)$, but the effect of multiple SCCs may be significant for finite N (effect S1). In this scenario, two obvious choices are to either choose the largest SCC as the GSCC or to include all SCCs as part of the GSCC. Either choice leads to finite-size effects, and both reduce to the standard one in the infinite limit for random networks, when the only SCC which contributes to S is the GSCC. Based on the observation that in a finite network, more than one SCC can in principle contribute to T , I use the convention that all SCCs and their out-components should be included in S . (This also gives the most conservative estimate of the accuracy of the theory, because in networks without strong community structure, the former choice leads to finite-size effects which cancel to some extent.)

With this choice, the most significant finite-size effect in Fig. 3.3 is that $S > T$ for small T . This is due to the definition of S combined with the fact that the derivation of T relies on the assumption that the network is locally tree-like, which is only approximately true for typical finite networks near the percolation transition. In particular, finite configuration-model networks contain some short loops, which contribute to S but are assumed not present in T (effect S2). The source of this discrepancy may be illustrated with an example: Consider an SCC which is a simple loop between two nodes, for which $q_1 = q_2 = \frac{1}{2}$. Equation (3.15) reduces to $\eta_1 = \frac{1}{2}(1 + \eta_2)$ and $\eta_2 = \frac{1}{2}(1 + \eta_1)$. The solution is $\eta_1 = \eta_2 = 1$, contributing nothing to T . However, the probability that both nodes are undeleted is $\frac{1}{4}$, and when this occurs, there will be a nonzero contribution to S from this SCC. Near the percolation

transition, many simple loops form, and numerical results suggest that they typically form between high-degree nodes with large out-components. The total contribution of such loops may be substantial for near-critical networks.

There are also two finite-size effects in the estimate of Y :

- Y1. When the number of damaged nodes is small, a “gambler’s ruin” effect may cause $Y < T$ for some networks with small T .
- Y2. Finite networks contain some simple loops on which damage can persist indefinitely, contributing to Y but not T . This is analogous to effect S2.

Consider the number of damaged nodes in a Boolean network with semi-annealed dynamics. This number is discrete and evolves stochastically in time, undergoing significant fluctuations early in the dynamics when the number of damaged nodes is initially small. Sometimes the number of damaged nodes fluctuates to zero even though it would have been expected to grow macroscopic in the absence of fluctuations. If this occurs, the two orbits x and \tilde{x} have become identical, and the damage will remain at zero for the rest of the dynamics (effect Y1). This is a variant of the classic “gambler’s ruin” problem from probability theory, in which a gambler wishes to become arbitrarily rich by betting with an unfair coin, but may be “ruined” by losing all of his money at some point along the way, at which point he is forced to stop gambling. The gambler’s ruin effect in Boolean networks has previously been found to occur when the number of initially damaged nodes εN is small (see Supplementary Information 2 in Ref. [17]). Specifically, it occurs when very few damaged nodes belong to the in-components of SCCs on which damage

can persist. For the networks used in the numerical results above, the size of these in-components is similar to S (the size of the out-components of SCCs), which is modeled by T . This suggests that gambler's ruin will be significant when εNT is small, which is consistent with Fig. 3.5(c).

Below the stability phase transition in a finite network, it is possible for damage to persist indefinitely on short loops which do not contribute to T (effect Y2). This occurs because short loops are not locally tree-like and there is quenched disorder in the update rules, violating the assumptions used to derive Eq. (3.13). Therefore, this leads to deviations from the semi-annealed approximation for small SCCs. For networks close to the stability phase transition, these loops can spread damage to comparatively large out-components in a manner that is analogous to effect S2 described above, and the same example can be used to illustrate both cases. Therefore, numerical results show $Y > T$ for some networks with very small T ($T < .01$). However, the effect is quite small, and it is quickly overtaken by effect Y1 when $T > .01$, so it cannot be seen in the figures above.

3.B Convergence analysis

Several numerical tests confirm that the numerical uncertainties in the values of T , Y , and S reported in the main text are significantly smaller than the corresponding point sizes in Figs. 3.1-3.4. The theoretical prediction T is determined by iterating Eq. (3.12) from the main text until $\langle |y_i^*(t) - y_i^*(t-1)| \rangle < 10^{-7}$. (This level of accuracy for T is not strictly necessary, but it adds virtually nothing to the cal-

ulation time.) The convergence time here is denoted t_T . Figures 3.6 and 3.7 show t_T as a function of z and N . There is a critical slowing down in the convergence time near the critical point $z = 2.5$, but t_T has no obvious dependence on N .

For the average Hamming distance Y and the expected giant out-component S , the uncertainties ΔY and ΔS are interpreted as the standard deviations of the sample means used to calculate Y and S . Recall that Y is determined by choosing two similar initial conditions $x(t)$ and $\tilde{x}(t)$, time-evolving each of them, averaging $|x_i(t) - \tilde{x}_i(t)|$ over i and t , and averaging over many initial conditions. The dominant source of uncertainty in this process is averaging over initial conditions, which contributes significantly to ΔY near the critical point. In the simulations in the main text, using 100 initial conditions per network is sufficient to reduce the uncertainty to below $3.5 \cdot 10^{-4}$ for every network when $N = 10^5$.

A second source of uncertainty in Y is the uncertainty in the measurement of the Hamming distance for each initial condition. This uncertainty is estimated here using an additional simulation. Many systems are time-evolved in parallel, where each system consists of two trajectories starting from close initial conditions. The Hamming distance in each *system* i is observed, which is denoted $Y^i(t)$. Starting at $t = 100$, convergence is tested every 10 time steps according to the following procedure. The average Hamming distance $\bar{Y}(t)$ is recorded for the last 100 time steps, where the average is taken over the initial conditions. The Pearson coefficient of $\bar{Y}(t)$ and t is calculated for these last 100 time steps. When \bar{Y} has reached its equilibrium behavior, it will resemble a random variable fluctuating around some mean value, so the Pearson coefficient will be approximately 0. Here, $\bar{Y}(t)$ is consid-

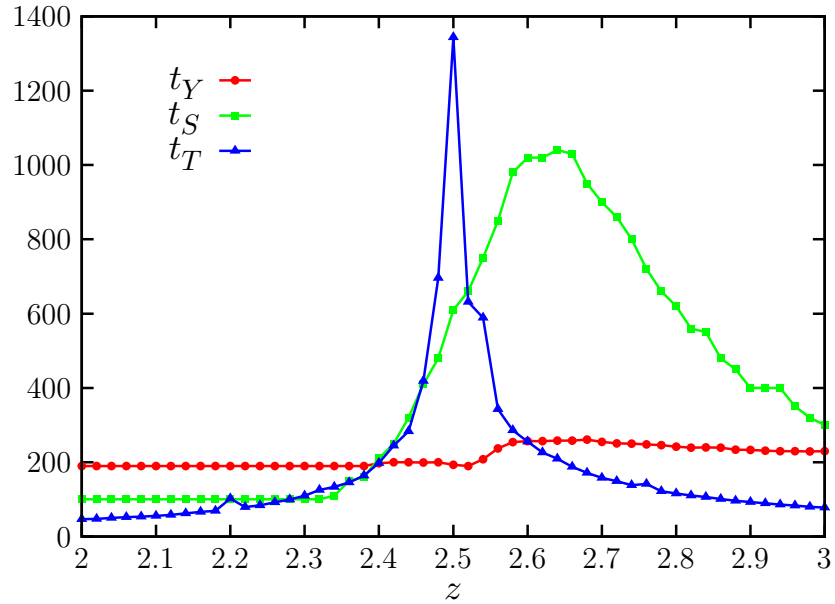


Figure 3.6: Average convergence times for T , Y , and S on networks with $N = 10^5$, as a function of average degree z . The convergence times for each point are averaged over 10 networks, each of which is generated with the same parameters as the calculations in Fig. 3.2 of the main text. The method of determining convergence is described in the text. It appears that T undergoes critical slowing down near the critical point $z = 2.5$ while Y and S do not, although S slows down significantly in the region slightly above the critical point.

ered to have converged when the absolute value of the Pearson coefficient is smaller than $1 \cdot 10^{-7}$ for at least 10 consecutive convergence checks. Note that because this method is conservative in its estimate, the earliest possible convergence time is $t_Y = 190$, even if the actual equilibrium behavior has been reached earlier.

Once the system has reached its equilibrium behavior, it is time-evolved for 200 more time steps and a measurement is taken over the last 100. To estimate the uncertainty of the measurement, each $Y^i(t)$ is regarded as a random variable fluctuating in time with a mean μ^i and a variance $(\sigma^i)^2$, which is estimated using a sample mean and unbiased sample variance. The standard deviation of the mean Y is then given by $\frac{1}{100} \sqrt{\sum_i (\sigma^i)^2}$. This quantity is always less than $1 \cdot 10^{-4}$ for the parameters used in the simulations in the main text. Therefore, for each simulation, the combined uncertainty ΔY is smaller than $5 \cdot 10^{-4}$, which corresponds to the smallest point size in Figs. 3.2-3.4. That is, the most restrictive condition is that $\Delta Y/Y < .05$ for the log-log plots, where ΔY is the uncertainty in Y . For $Y = .01$, this corresponds to $\Delta Y < 5 \cdot 10^{-4}$.

In Figure 3.6, it can be seen that the convergence time for Y , t_Y , is much less than 900, the transient time used for the calculations in the main text. The convergence of Y does not appear to undergo critical slowing down. The relatively short transient period can be understood intuitively by considering that the average path length in a random network with N nodes grows only logarithmically in N [10]. Information passes quickly through the network, which typically falls quickly into either a short periodic orbit or a “chaotic” orbit with seemingly ergodic dynamics. In either case, the convergence to the equilibrium behavior is generally quite fast. Note

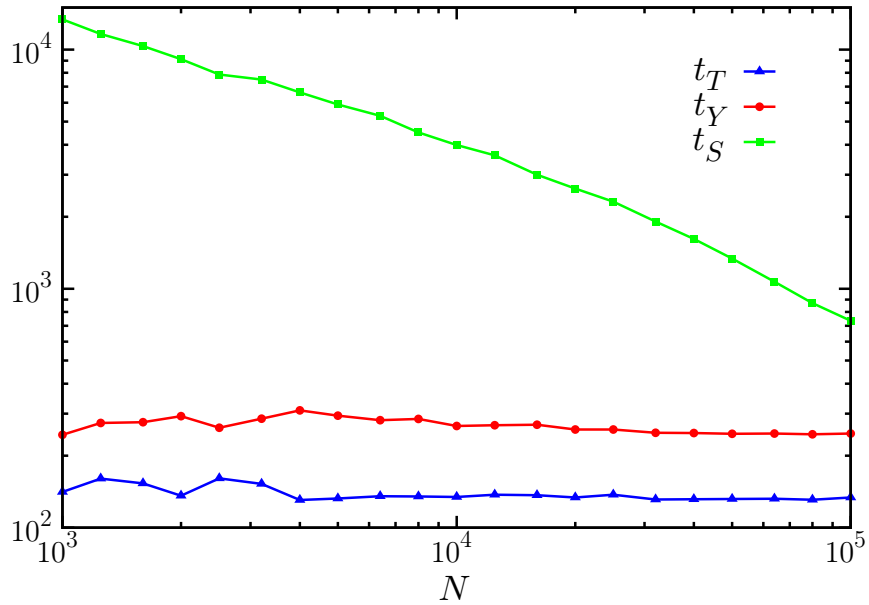


Figure 3.7: Average convergence times for T , Y , and S as a function of the network size N when $z = 2.75$. Each point is averaged over 20 network realizations.

that although t_Y does not undergo critical slowing down, the variance in μ^i between different initial conditions is largest near the critical point. It is possible that there is a critical slowing down effect in the number of initial conditions required for the results to converge to a specified level of relative accuracy (i.e., $\frac{\Delta Y}{Y} < \text{const.}$), but this possibility is not explored here. Also, note from Fig. 3.7 that t_Y has no obvious dependence on N , although it is possible that the fact that the minimum possible value of t_Y is 190 has obscured a small N -dependence.

Finally, consider the convergence of S . In the main text, S is defined as the mean size of GOUT over 1000 node deletion trials. The uncertainty ΔS is the standard deviation of the sample mean for these trials. In the simulations in the main text, ΔS is less than the point size $5 \cdot 10^{-4}$ for all data points. Figures 3.6

and 3.7 show the number of node deletion trials which are actually necessary for convergence, which is denote t_S , by checking every 100 trials whether $\Delta S < 5 \cdot 10^{-4}$. Interestingly, critical slowing down is not observed for S (at least when finding an absolute rather than relative level of accuracy). Instead, the largest fluctuations in S occur in the region slightly above the critical point, due to the finite-size effects discussed in the previous section. These fluctuations decrease as N grows, leading to the N -dependence of t_S seen in Fig. 3.7.

Finally, the uncertainties in Fig. 3.1, in which T , Y , and S are averaged over 20 networks, can be estimated in the following way. Let Y_k denote the measured value of Y for the k^{th} network and ΔY_k its uncertainty. Then the uncertainty in $\langle Y \rangle$ is given by $\Delta \langle Y \rangle = \sqrt{\frac{1}{20} \text{Var}_k[Y_k] + \frac{1}{20^2} \sum_k (\Delta Y_k)^2}$. This incorporates both the uncertainty from incomplete sampling of the ensemble of networks as well as the uncertainty associated with each measurement. Analogous expressions are used for $\Delta \langle T \rangle$ and $\Delta \langle S \rangle$. For each point in Fig. 3.1, $\Delta \langle T \rangle$, $\Delta \langle Y \rangle$, and $\Delta \langle S \rangle$ are all significantly less than the point size, $5 \cdot 10^{-3}$.

3.C On the uniqueness of the steady-state damage probabilities

In the main text, I made the assertion that in cases of interest, the update equation for damage in semi-annealed, locally treelike networks, Eq. (3.13), has a unique stable solution, to which Eq. (3.12) will converge. While this claim is not an important focus of this chapter, it is an interesting side topic worthy of exploration. The assertion cannot be proven in general, because there are straightforward

counterexamples, such as the case where the “network” consists of two completely disconnected pieces, each of which may independently have its own steady-state damage probabilities. However, I show in this section that, with suitable restrictions, the assertion is correct.

In particular, the discussion is restricted to a strongly connected network with a primitive adjacency matrix, i.e., there is some integer n_0 such that, for all $n \geq n_0$, $(A^n)_{ij} > 0$ for all i and j . The sensitivities are also restricted so that $q_i > 0$ for all nodes i ; otherwise, network edges to node i do not actually influence the dynamics. The primitivity of A allows for the application of Frobenius-Perron theorem, a powerful result that applies to such matrices [76]. It states, among other things, that the eigenvalue of A with largest magnitude, as well as all the entries of its associated eigenvectors, will be real and positive.

The restrictions imposed here are suitable for any network which does not have multiple strongly connected components, which is an excellent approximation for networks above the percolation threshold. In this case, according to the argument here, the damage probabilities in the giant strongly connected component are fixed, which determines damage probabilities in the rest of the network, barring the possibility that small strongly connected components have a significant effect. Networks below the percolation threshold, on the other hand, are nearly always dynamically stable (i.e., the only stable fixed point is $y = 0$). The main case where the conclusion does not apply, then, is when, perhaps due to strong community structure, there are many distinct strongly connected components which are all important.

Begin by defining the right-hand side of Eq. (3.13) to be the i^{th} component

M_i of a map $M : [0, 1]^N \rightarrow [0, 1]^N$,

$$M_i(y) = q_i \left[1 - \prod_{j \in \mathcal{J}_i} (1 - y_j) \right]. \quad (3.21)$$

This map has a trivial fixed point, $M(0) = 0$, but it may also have a nonzero fixed point, denoted y^* . Below, the first step is to show that if a nonzero fixed point y^* exists, then it must be unique and positive. Subsequently, it is shown that a nonzero fixed point only exists if the fixed point at zero is unstable, in which case the positive fixed point is stable and the only attractor of iterating M ; otherwise, the fixed point at zero is the only attractor of iterating M .

Note that $\partial_j M_i$ (i.e., the partial derivative of M_i with respect to y_j) is always positive when there is an edge from j to i and is zero otherwise. Similarly, the second derivatives $\partial_j \partial_k M_i$ is always negative when there are edges from j and k to i and $j \neq k$. This suggests that there is a sense in which M is an increasing, concave function, for which there are useful fixed-point theorems. However, rather than considering M directly, it is more convenient to consider the n^{th} iterate of this map, where $n \geq n_0$ as defined above.³ This map, which will be denoted $\hat{M}(y) = M^n(y)$, can be shown to be *strictly* increasing as follows. The Jacobian matrix $D\hat{M}$ of \hat{M} can be related to that of M through the chain rule,

$$D\hat{M}(y) = DM(M^{n-1}(y))DM(M^{n-2}(y)) \dots DM(y). \quad (3.22)$$

The sign of each factor of DM_{ij} is the same as A_{ij} , so the sign of each element in the product is the same as that of A^n , which is strictly positive. Since every element of $D\hat{M}$ is strictly positive, \hat{M} is strictly increasing. By a similar argument, the sign of the second derivative $\partial_j \partial_k \hat{M}_i$ is never positive, so \hat{M} is concave.

Since \hat{M} is strictly increasing, any nonzero fixed point y^* must be strictly positive in each element, $y_i^* > 0$ for all i . Moreover, it has been shown that a strictly increasing, concave map can have at most one positive fixed point [77], so any positive fixed point y^* of \hat{M} must be unique; see Theorem 1 and Remark 1 in Ref. [77]. If \hat{M} does have a positive fixed point y^* , this must also be a fixed point of M (rather than, say, lying on periodic orbit of M), because the argument above can be applied for any $n \geq n_0$.

It remains to be shown that there are two cases, one in which $y = 0$ is stable and attracting, but there is no positive fixed point, and one in which $y = 0$ is unstable, but $y = y^*$ is stable and attracting. The first part follows from the linear stability analysis of this system, which is discussed more fully in Ref. [17]. To linear order, Eq. (3.12) is $y(t + 1) = Qy(t)$, where $Q_{ij} = DM(0)_{ij} = q_i A_{ij}$. When the largest eigenvalue of Q , λ_Q , is less than 1, it is easy to show that the zero fixed point will be stable and attracting. Since M is concave, $M_i(y) \leq M_i(0) + (DM(0)y)_i = (Qy)_i$ for all i and y , so $|M(y)| \leq |Qy| \leq \lambda_Q |y| < |y|$ for all y .⁴

When $\lambda_Q > 1$, another theorem can be used to guarantee that iteration will lead to convergence (and also that y^* is in the unit hypercube). Using Theorem 3 from Ref. [77], the only additional necessary conditions are that there are two positive vectors y and y' such that $M_i(y) > y_i$ and $M_i(y') < y'_i$ for all i . (See also Remarks 1 and 3 from Ref. [77].) The first requirement can be satisfied by using $y = \epsilon v_Q$, where v_Q is the right eigenvector associated to λ_Q , for ϵ sufficiently small. Choosing $y'_i = 1$ for all i satisfies the second requirement, since $M_i(y') = q_i < 1 = y'_i$. Therefore, when $\lambda_Q > 1$, a positive fixed point exists and is stable and attracting.

It is also instructive to note that the stability of any positive fixed point can be verified for this map using the following argument, independent of most of the above discussion. Since M is concave, a first-order Taylor series at $y = y^*$ can be used to show that $(DM(y^*)y^*)_i < M_i(y^*)$, which is equal to y_i^* by definition. (This can also be shown algebraically using the Weierstrass inequality.) Since $DM(y^*)$ is a primitive non-negative matrix, the Frobenius-Perron theorem applies, and so its largest eigenvalue λ_* is positive, and the associated left eigenvector u^* is also positive. Multiplying by u_i^* and summing over i gives $\lambda_*(u^*)^T y^* = (u^*)^T DM(y^*)y^* < (u^*)^T y^*$. Dividing through by $(u^*)^T y^*$, which must be positive, shows that $\lambda_* < 1$.

Notes for Chapter 3

¹A more thorough definition is that for locally tree-like networks, short paths between j and i should account for a small fraction of the total number of short paths originating at j or terminating at i [73]. For synchronously updated Boolean networks, an even weaker locally tree-like condition is sufficient for the derivation at hand, that paths from j to i are a small fraction of paths *of the same length* beginning at j or ending at i . However, since asynchronously updated networks are also treated in Chapter 4, this point is not emphasized further. Networks with motifs that violate each of these conditions are studied in Refs. [17].

²It can be shown that $r_i \geq \frac{1}{2}q_i$, so the second probability is non-negative.

³Perhaps the most straightforward approach would be to apply Theorem 1 of Ref. [77] directly to M . Unfortunately, this approach seems to be rigorous only in cases where each node has an in-degree of at least two.

⁴When $\lambda_Q = 1$, y^* is marginally stable. Convergence, which undergoes critical slowing down, depends in this case on the fact that M is *strictly* concave along the right eigenvector for λ_Q .

Chapter 4: The joint effects of topology and update rules on the stability of Boolean networks

4.1 Introduction

In this chapter, I present and numerically verify a general method for studying the stability of large, locally tree-like directed Boolean networks [3]. This method can be used to study a variety of Boolean network systems for which, up to now, no generally effective method has been available. Like the technique used in Chapter 3, it can be applied to *specific* complex networks with given topology, but it is much more general in the sense that it can also be applied to systems with essentially arbitrary update rules. Because of its generality, this technique may be useful for modeling systems whose topology and update rules are determined experimentally.

For example, one question which may be important in gene regulatory systems is how silencing (or “knocking down”) a particular gene will affect the expression of other genes, as well as the stability of the dynamics of the entire system. The present analysis provides a framework for addressing this kind of question, in contrast to previous work based on the annealed approximation, which typically obtained analytical results on the stability of ensembles of networks with given degree

distributions. The framework below can also be used in a more general sense to study how topology, update rules, and correlations between the two affect stability. This is demonstrated with two examples illustrating that the joint effects of network topology and update rules can have profound effects on Boolean network dynamics, which cannot be captured by previous theories.

As described in Section 1.5, in a Boolean network, each node i is assigned a binary state $x_i(t) = 0$ or 1 at each discrete time t using an update rule F_i which depends on the states of its network inputs on the previous time step, i.e.,

$$x_i(t+1) = F_i(X_i(t)), \quad (4.1)$$

where the combined input state $X_i(t)$ is an ordered list of $x_j(t)$ for all j which are network inputs to i . A Boolean network is considered to be stable or unstable, respectively, if two initially close orbits $x(t)$ and $\tilde{x}(t)$ evolving under Eq. (4.1) tend to converge or diverge with time, as discussed in Sections 1.6 and 4.4. A node i is called “damaged” at time t if $x_i(t) \neq \tilde{x}_i(t)$. The main theoretical result of this chapter is a criterion for stability in Section 4.3 that accounts for the joint effects of network topology (represented by an adjacency matrix A) and node dynamics (i.e., the functions F_i). For now, only synchronously updated Boolean networks are considered, but Appendix 4.A demonstrates that this stability criterion is unchanged if nodes are updated asynchronously.

The chapter is organized as follows. Section 4.2 reviews previous work on the annealed and semi-annealed approximations (also discussed in Sections 3.2.1 and 3.3.1), then introduces a new method of semi-annealing which can be applied to

arbitrary update rules. This semi-annealing procedure is then used to derive a set of “dynamical biases” which describe the steady-state dynamics of the system. Section 4.3 uses these dynamical biases to analyze the stability of the system, giving the main result of the chapter. Agreement between theory and numerical simulations is illustrated for two examples in Section 4.4. The first of these examples is primarily pedagogical, but the second is application-oriented, using threshold update rules. These take the form

$$x_i(t+1) = U\left(\sum_j W_{ij}x_j(t) - \theta_i\right), \quad (4.2)$$

where U denotes the unit step function, θ_i is a threshold value, and W_{ij} is a signed weight whose magnitude reflects the strength of the influence of node j on node i and whose sign indicates whether node j “activates” or “represses” node i (i.e., promotes x_i to be 1 or 0).

4.2 Generalized semi-annealing and dynamical biases

The stability of Boolean networks has, for three decades, been analyzed primarily in terms of Derrida and Pomeau’s annealed approximation [44]. This states that the stability of a typical “quenched” or “frozen” realization of a random network with randomly selected update rules is typically similar to that of an “annealed” system in which the network and update rules are randomized between every time step. Because the annealed system can be treated analytically, the annealed approximation has provided powerful techniques for studying ensembles of random networks. However, its major limitation is that it cannot be applied to specific networks with

complex topological features.

This limitation was addressed in Ref. [17], which developed a “semi-annealing” technique that directly uses the adjacency matrix of the system under examination when addressing the question of stability. In a semi-annealed system, the network topology is *fixed* and may be complex, and only the update rules are selected randomly at each time step. The semi-annealed approximation is the hypothesis that the stability of the semi-annealed system, which can be computed analytically, is a good approximation for the stability of a typical system with quenched update rules on the same network. Besides allowing for the consideration of complex network topologies, this approach also makes it possible to consider the contributions of individual nodes to the system dynamics.

The semi-annealed approximation has been used to analyze the stability of complex locally tree-like networks with biased update rules, [2, 17], and has also been applied to canalizing update rules [2, 78]. In each case, the stability of the system is determined by the magnitude of the largest eigenvalue of a modified adjacency matrix, which allows for the consideration of quite general topological features. In fact, Refs. [17, 78] considered systems with a variety of complex features, such as degree assortativity, community structure, motifs, and correlation of the in- and out-degrees d_i^{in} and d_i^{out} with node bias b_i . Many of these features dramatically affect the stability of the system in ways that cannot be predicted using fully annealed techniques. Numerical results strongly support the validity of the semi-annealed analysis [2, 17, 78].

Thus, semi-annealed techniques have vastly extended the range of network

topologies whose stability can be analyzed. However, they have only been applied to two limited classes of update rules, and a clear next step is the development of a semi-annealed analysis which can be applied with fewer constraints on the update rules. The principal difficulty in this endeavor is that it requires detailed knowledge of the steady-state behavior of the system. Here, this problem is addressed by introducing a set of “dynamical biases,” p_i , which describes the steady-state behavior of the semi-annealed system and can be determined from a set of consistency conditions.

In the generalized semi-annealing procedure introduced here, each node i is assigned an ensemble of update rules, \mathcal{T}_i , from which a specific update rule is randomly drawn at each time step t . This choice is made independently at each network node i , and the probability of drawing a particular update rule f from \mathcal{T}_i is denoted by $\Pr[F_i = f]$. In typical applications of Boolean networks, each quenched update rule F_i are selected from a random ensemble, and in these cases one should use the same ensemble for \mathcal{T}_i . Otherwise, the selection of the update rule ensemble \mathcal{T}_i may depend on the case of interest. This is illustrated with examples in Section 4.4.

The dynamics of this model are somewhat similar to those of probabilistic Boolean networks [79]. They may be described by the probabilities $\bar{F}_i(X_i)$ that the state of node i , given inputs X_i , will be 1 on the next time step. This is just the average of $F_i(X_i)$ over the ensemble \mathcal{T}_i ,

$$\bar{F}_i(X_i) = \sum_{f \in \mathcal{T}_i} \Pr[F_i = f] f(X_i), \quad (4.3)$$

because $f(X_i) = 0$ or 1 . It is important to note that $\bar{F}_i(X_i)$ is solely determined from \mathcal{T}_i , i.e., independent of the update rule assignments at other nodes. Thus,

computation of $\bar{F}_i(X_i)$ is straightforward.

Another important quantity, which is also purely local and therefore can also be calculated directly from \mathcal{T}_i , is the probability that a node i will become damaged if its input states on the previous time step are given by X_i and \tilde{X}_i for the original and perturbed orbits. This quantity is denoted by $D_i(X_i, \tilde{X}_i)$, and it may be calculated using

$$D_i(X_i, \tilde{X}_i) = \Pr \left[x_i(t+1) \neq \tilde{x}_i(t+1) \mid X_i, \tilde{X}_i \right] \quad (4.4a)$$

$$= \sum_{f \in \mathcal{T}_i} \Pr[F_i = f] \cdot \left| f(X_i) - f(\tilde{X}_i) \right|, \quad (4.4b)$$

because, once again, $f(X_i) = 0$ or 1 .

This formalism allows for the treatment of a wide variety of update rules, including but not limited to previously studied cases. As an example, consider the case of biased update rules with node-dependent biases b_i ; other examples will be given in Section 4.4. For biased rules, the ensemble \mathcal{T}_i must contain all possible update rules with d_i^{in} inputs, since any such rule could be generated when $0 < b_i < 1$. The probabilities $\Pr[F_i = f]$ follow a binomial distribution $\binom{d_i^{\text{in}}}{n} b_i^n (1-b_i)^{d_i^{\text{in}}-n}$, where n is the number of combined input states which result in $f = 1$. It is easy to show that with this distribution, $\bar{F}_i(X_i) = b_i$ for all X_i , as well as the well-known result that the ‘‘sensitivity’’ of a biased function is $D_i(X_i, \tilde{X}_i) = 2b_i(1-b_i)$ for all $\tilde{X}_i \neq X_i$ [17, 44, 56]. However, here, as in most cases, it is not actually necessary to specify the ensemble \mathcal{T}_i . In fact, the analysis below only makes use of the functions \bar{F}_i and D_i , and in many cases, including the example in Section 4.4.2, it is more convenient to specify them directly than to calculate the probabilities $\Pr[F_i = f]$.

Note that this approach can also be used to describe a quenched system if only one function is in each \mathcal{T}_i . However, I will primarily focus on cases in which there is a non-trivial degree of semi-annealing, which typically results in the existence of a single ergodic attractor for the system, simplifying the analysis. (A sufficient, but certainly not necessary, condition for this to occur is if $0 < \bar{F}_i(X_i) < 1$ for all i and X_i .) The semi-annealed approximation asserts that the stability of this attractor in the semi-annealed system is the same as that of typical attractors in quenched systems. Further comments on applying these results to quenched dynamics will follow below.

The next step is to define the dynamical biases p_i and derive a set of consistency equations which can be used to determine them. Given an orbit on the attractor of the semi-annealed system, p_i is defined to be the fraction of time that the state $x_i(t)$ of node i is 1, or the probability that $x_i(t) = 1$ at a randomly chosen time t . A global average of this quantity has been considered in fully annealed network ensembles (e.g., [66, 67, 69]).

The consistency conditions which determine p_i can be derived as follows. First, note that p_i is determined by the set of probabilities $\Pr[X_i]$ of i receiving each possible combined input state X_i , using

$$p_i = \sum_{X_i} \Pr[X_i] \Pr[x_i = 1 | X_i] \quad (4.5)$$

$$= \sum_{X_i} \Pr[X_i] \bar{F}_i(X_i), \quad (4.6)$$

where \bar{F}_i is as defined in Eq. (4.3). Assuming that the network topology is locally tree-like, the states of the inputs to node i can be treated as statistically independent

(see the discussion of semi-annealing in locally tree-like networks in Section 3.3.1). Therefore, the probabilities $\Pr[X_i]$ are determined by the biases of i 's inputs. Letting \mathcal{J}_i denote the set of indices of nodes that are inputs to i ,

$$\Pr[X_i] = \prod_{j \in \mathcal{J}_i} [x_j p_j + (1 - x_j)(1 - p_j)], \quad (4.7)$$

where the fact that each $x_j = 0$ or 1 has been used. That is, the probability of a state X_i in which an input $x_j = 0$ is weighted by a factor of $(1 - p_j)$, and the probability of a state in which $x_j = 1$ is weighted by a factor of p_j .

Inserting Eq. (4.7) into Eq. (4.5) yields a set of N equations for the N dynamical biases p_i . In typical applications, these equations can be solved by choosing a random initial value for each p_i and iterating Eqs. (4.5-4.7) until they converge. Moreover, in numerical simulations such as those considered in Section 4.4, there is a unique solution which is stable under iteration, and this represents the attractor of the semi-annealed system. This will be assumed in the analysis below, but note that the analysis can be extended in a natural way to cases where this does not occur. For example, some systems (including completely quenched systems) may have more than one attractor, in which case the attractors can be found by iterating multiple initial guesses for p_i , and the stability of each attractor can be evaluated individually.¹

4.3 Stability analysis

Now, consider the stability of the semi-annealed system. As in Section 3.3.1, let $y_i(t)$ be the probability that i is damaged at time t , i.e.,

$$y_i(t) = \Pr [x_i(t) \neq \tilde{x}_i(t)]. \quad (4.8)$$

Marginalizing over X_i and \tilde{X}_i in Eq. (4.8) and inserting Eq. (4.4a),

$$y_i(t+1) = \sum_{X_i} \sum_{\tilde{X}_i} \Pr [X_i(t), \tilde{X}_i(t)] D_i (X_i, \tilde{X}_i). \quad (4.9)$$

In the question of stability, it is assumed that the two orbits under consideration, $x(t)$ and $\tilde{x}(t)$, are close to each other in the sense of Hamming distance for times t close to the perturbation time t_0 . In this case, terms of $\mathcal{O}(y^2)$ on the right hand side of Eq. (4.9) can be dropped, which corresponds to ignoring the possibility that $X_i(t)$ and $\tilde{X}_i(t)$ differ for two or more inputs. Moreover, if $\tilde{X}_i(t)$ and $X_i(t)$ are the same, $D_i = 0$ via Eq. (4.4b), so nothing is contributed to the sum in Eq. (4.9). Therefore, the only values of \tilde{X}_i which contribute significantly to the sum are ones in which $\tilde{X}_i(t)$ and $X_i(t)$ differ for exactly one node j . Hence, $X_i^j(t)$ is defined to be a vector which is the same as $X_i(t)$ except that the state of input node j is damaged [$\tilde{x}_j(t) = 1 - x_j(t)$]. Using this notation, Eq. (4.9) can be rewritten as

$$y_i(t+1) = \sum_{j \in \mathcal{J}_i} \sum_{X_i} \Pr [X_i(t), X_i^j(t)] D_i (X_i, X_i^j). \quad (4.10)$$

Furthermore, because the network is locally tree-like, the inputs to node i can

be treated as uncorrelated, yielding

$$\begin{aligned} \Pr [X_i(t), X_i^j(t)] &= \Pr [X_i] \Pr [x_i(t) \neq \tilde{x}_j(t)] \\ &= \Pr [X_i] y_j(t). \end{aligned} \tag{4.11}$$

When substituted into Eq. (4.10), this leads to

$$y_i(t+1) = \sum_{j \in \mathcal{J}_i} y_j(t) \sum_{X_i} \Pr [X_i] D_i (X_i, X_i^j). \tag{4.12}$$

Since the second sum is time-independent, this can be written as

$$y_i(t+1) = \sum_j R_{ij} y_j(t) + \mathcal{O}(y^2), \tag{4.13a}$$

$$R_{ij} \equiv \sum_{X_i} \Pr [X_i] D_i (X_i, X_i^j), \tag{4.13b}$$

where $R_{ij} = 0$ when there is no edge from j to i . (The second-order terms in this expansion are discussed further in Appendix 4.B, which derives an expression for the critical slope of the stability phase transition.) R_{ij} may be interpreted as the probability that damage will spread from node j to node i and here it is named the “effective activity” of j on i in analogy with the terminology of Ref. [56].

The average of the normalized Hamming distance over all possible perturbations and realizations of the semi-annealed dynamics is $\langle H(x(t), \tilde{x}(t)) \rangle = \frac{1}{N} \sum_i y_i(t)$, so the stability of the system is determined by whether or not the elements of $y(t)$ grow with time. It can be seen by writing Eq. (4.13b) in matrix form,

$$y(t+1) = R y(t), \tag{4.14}$$

that the growth of y is determined by whether or not R has any eigenvalues λ whose magnitude is greater than 1. If so, and the initial perturbation has a nonzero component along the left eigenvector associated with one of these eigenvalues, the expected

Hamming distance will initially grow as λ^t . Therefore, the largest-magnitude eigenvalue of R , denoted λ_R , predicts the stability of the system. It can be shown that this eigenvalue is always real and positive. With additional assumptions that are applicable to commonly studied cases, it can also be shown that this eigenvalue has multiplicity one and is well-separated in magnitude from other eigenvalues, which implies that it can be computed efficiently through power iteration of R on an initial random vector.² Therefore, the stability of the system can be classified using

$$\left. \begin{array}{ll} \lambda_R < 1 & \text{stable} \\ \lambda_R = 1 & \text{critical} \\ \lambda_R > 1 & \text{unstable} \end{array} \right\}. \quad (4.15)$$

One major advantage of the approach introduced here is that, from a computational perspective, evaluating p , R , and λ_R is typically much faster than finding the average Hamming distance through simulations. This is discussed further in Appendix 4.C, along with other computational aspects of the above solution. Another potential advantage of Eq. (4.15) is that it may facilitate qualitative understanding of the effects of certain network or update rule features on stability. For example, previous work has shown that introducing assortativity to a network tends to increase the largest eigenvalue of certain types of modified adjacency matrices, and this effect can be estimated analytically [8, 80]. This suggests that similar results may be derived for λ_R in future work.

It is also possible that empirical gene regulatory networks will contain important topological or update rule features which may not correspond to commonly studied network metrics. The analysis presented here nonetheless applies to these

cases. Moreover, such features may be highlighted by any cases where the direct evaluation of λ_R gives a significantly different stability prediction than annealed or approximation-based techniques. In these cases, research aimed at understanding the discrepancy may lead to the discovery or study of new network features.

Finally, in contrast to fully annealed techniques, in which nodes are distinguished only by their degree signature, semi-annealing also allows for the consideration of individual nodes and their contribution to instability in the full system. This enables the treatment of questions such as the optimal method of stabilizing or de-stabilizing a network through some limited set of possible external interventions. One interesting case is where the intervention strategy is to permanently silence or activate a small set of nodes (i.e., forcing $x_i(t) = 0$ or $x_i(t) = 1$ for some node i). Another case is when damage can actively be corrected during the dynamics for some nodes. In both cases, the problem of selecting nodes for intervention can be addressed using this framework. This and other related questions are discussed further in Appendix 4.D.

4.4 Numerical results

This section uses the framework above to analyze two cases that illustrate the effects of correlations between local topological features and update rules. Each example uses a single network with $N = 10^5$ nodes using the configuration model [7]. The in-degrees are Poisson-distributed with a mean of 4 and the out-degrees are scale-free with exponent $\gamma \approx 2.2$. In Figs. 4.1 and 4.2, the average Hamming

distance $\langle H \rangle$ and λ_R are plotted against a tuning parameter for each model.

Each Hamming distance H is calculated using the following procedure. First, an initial condition $x(0)$ for an orbit $x(t)$ is randomly selected ($x_i(0) = 0$ or $x_i(0) = 1$ with equal probability for each i). Next, $x(t)$ is time-evolved until a time $t_0 = 100$, which is large enough that transient behavior has ceased. A perturbed initial condition $\tilde{x}(t_0)$ is created by randomly choosing a small fraction $\varepsilon = 0.01$ of the components of $x(0)$ and damaging their states, i.e., if node i is chosen to be damaged, then $\tilde{x}_i(0) = 1 - x_i(0)$. The long-time behavior of H is measured by averaging $H(x(t), \tilde{x}(t))$ from a later time $t_1 = 400$ to $t_2 = 500$. Finally, this procedure is repeated for many initial conditions $x(0)$ and average H over initial conditions.

When calculating H , the update rules are *quenched*, whereas the theoretical prediction λ_R is calculated using the semi-annealed theory described above. The average $\langle H \rangle$ is taken over 10 initial conditions for each quenched set of update rules. The figures show $\langle H \rangle$ for both a single quenched system as well as an average over 50 sets of quenched update rules.

4.4.1 Example 1: XOR, OR, and AND update rules

This example illustrates the effect of correlations by assigning a node either a highly “sensitive” update rule (XOR) or a less sensitive update rule (OR or AND), based on the node’s in-degree. That is, the update rule at each node i is randomly drawn from three classes: (1) XOR, whose output is one (zero) if i has an odd (even) number of inputs that are one; (2) OR, whose output is one if and only if at least one

input is one; or (3) AND, whose output is one if and only if all d_i^{in} inputs are one. Following [56], XOR is described as highly sensitive because *any* single damaged input will cause its output to become damaged, so $R_{ij} = 1$ whenever there is an edge from j to i . On the other hand, if node i has OR or AND as its update rule, damaging node j will damage node i only if every node other than j is zero or one, respectively. Thus in cases (2) and (3), R_{ij} depends on the node biases p_j which are obtained by solving Eqs. (4.5-4.7). For OR, $R_{ij} = \prod_k (1 - p_k)$, and for AND, $R_{ij} = \prod_k p_k$, where the products are taken over all inputs k which are not equal to j .

Figure 4.1 shows results for a network with these three classes of update rules. The update rules of a fraction α of nodes are assigned to be XOR, and the remaining nodes are evenly split between OR and AND rules. For correlations, three cases are considered here: (1) XOR is assigned to the αN nodes with the largest in-degree (red squares in Fig. 4.1); (2) XOR is randomly assigned to αN nodes irrespective of their degrees (blue circles); or (3) XOR is assigned to the αN nodes with smallest in-degree (green triangles). In all three cases, the remainder are randomly assigned OR or AND. In numerical simulations, all update rule assignments are quenched. In order to find appropriate semi-annealing probabilities for the theoretical prediction, it is important to note that the initial assignment of XOR is deterministic in cases (1) and (3), but OR and AND are assigned randomly. Therefore, in the theory, the network and the identity of the XOR nodes are treated as fixed, and the identities of the OR and AND nodes are annealed, assigning a probability of $1/2$ to choosing either OR or AND on each time step. Other annealing choices are also possible,

but this procedure is chosen because it straightforwardly resembles the quenched assignment of update rules above.

In Fig. 4.1(a), filled markers are averaged over 50 quenched realizations of the update rules, while hollow markers show a single quenched realization. The values of α at which the three cases become unstable are rather different, thus demonstrating that stability is strongly affected by correlation between the local topological property of nodal in-degree and the sensitive XOR update rule. This agrees with the intuitive expectation that assigning update rules which result in high values of R_{ij} to topologically important nodes will increase instability. In panel (b), the stability criterion $\lambda_R < 1$ correctly accounts for this behavior. Re-plotting $\langle H \rangle$ against λ_R in panel (c), it is clear that in each case the network becomes unstable at $\lambda_R \approx 1$, as predicted by the theory. This is also pointedly illustrated by the vertical arrows in panel (a) marking the values of α at which $\lambda_R = 1$.

4.4.2 Example 2: Threshold networks

The second numerical example is the case of networks with threshold rules, i.e. Eq. (4.2). Such threshold rules may be re-cast as Boolean functions F_i by enumerating all possible X_i and calculating whether the weighted sum of inputs exceeds the threshold θ_i . Conversely, threshold rules are appropriate for Boolean network applications in which each edge has a fixed “activating” or “repressing” character. The special case of random networks with $\theta_i = 0$ and $W_{ij} = \pm 1$ has been considered previously using the annealed approximation [68–71, 81].

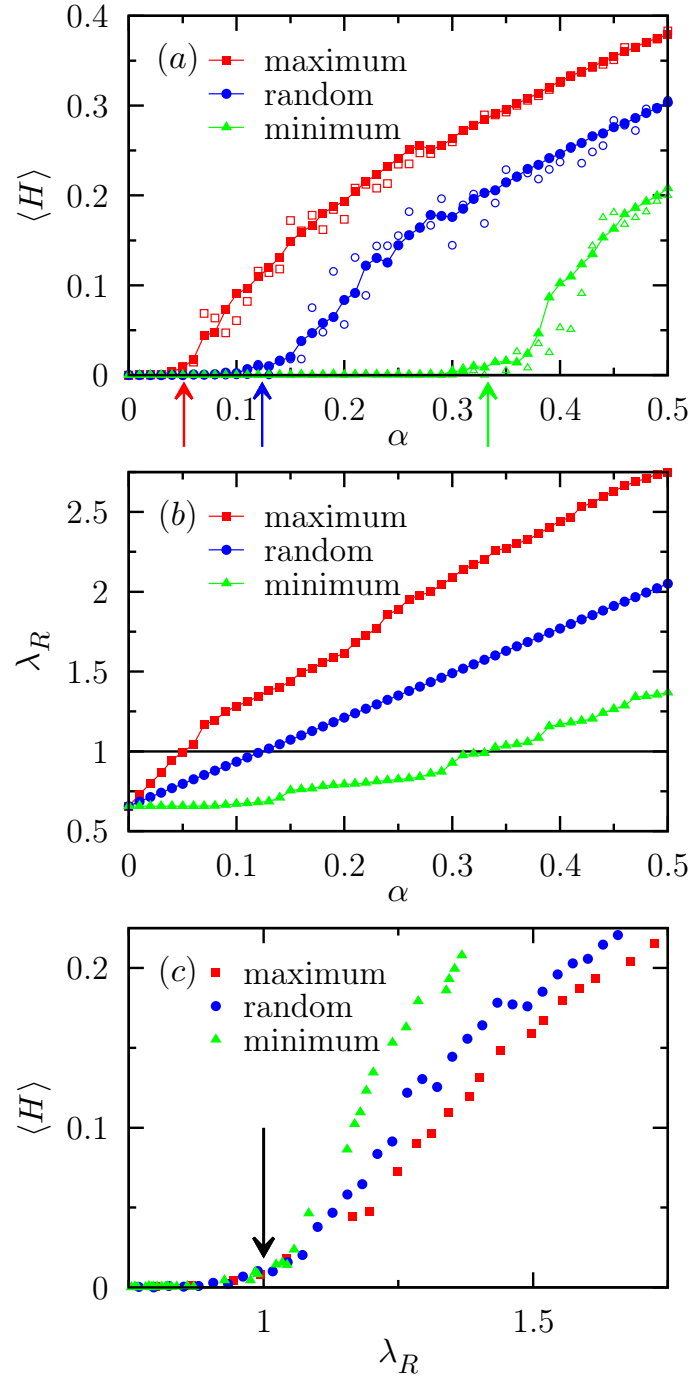


Figure 4.1: $\langle H \rangle$ and λ_R for a network with XOR, OR, and AND update rules. The tuning parameter α is the fraction of nodes using XOR rules, and different markers represent different correlations between these nodes and degree. Arrows mark points where $\lambda_R = 1$ and correctly predict transitions. See text of Section 4.4.1 for details.

Results for threshold networks are shown in Fig. 4.2 and are generated as follows. The weight W_{ij} for each edge $j \rightarrow i$ is assigned by first choosing a sign, positive or negative, each with probability $1/2$, to indicate whether the edge will be activating or repressing. Then, the weight is drawn from a normal distribution with mean ± 1 (according to the chosen sign) and standard deviation $1/4$. This procedure is used for the “uncorrelated” results in Fig. 4.2 (blue circles).

Correlations between update rules and network topology are introduced in two additional cases. Specifically, the weights of an edge W_{ij} may be correlated (red squares) or anticorrelated (green triangles) to the product of a node’s in-degree and out-degree, $d_i^{\text{in}} d_i^{\text{out}}$. It has been shown that nodes with high degree product play a crucial role in the stability of Boolean networks [17, 63, 80]. Here, the correlated and anticorrelated cases are generated by exchanging weights between pairs of edges in the original (uncorrelated) assignment of weights in the following procedure. First, two random edges $j_1 \rightarrow i_1$ and $j_2 \rightarrow i_2$ are selected from the network. Next, the edge for which i has a higher degree product is identified. Finally, in the correlated (anticorrelated) case, the values of the two weights are exchanged if doing so would increase (decrease) the weight of the edge with the higher degree product. This procedure is repeated $E/2$ times, where E is the number of edges in the network, so that each edge is expected to be considered for one exchange.

Figure 4.2 treats the case in which the thresholds of different nodes are similar, but not necessarily equal. In the theory, this is modeled by annealing the thresholds θ_i over a normal distribution with a mean $\bar{\theta}$ and standard deviation $\sigma_\theta = 1/10$. By

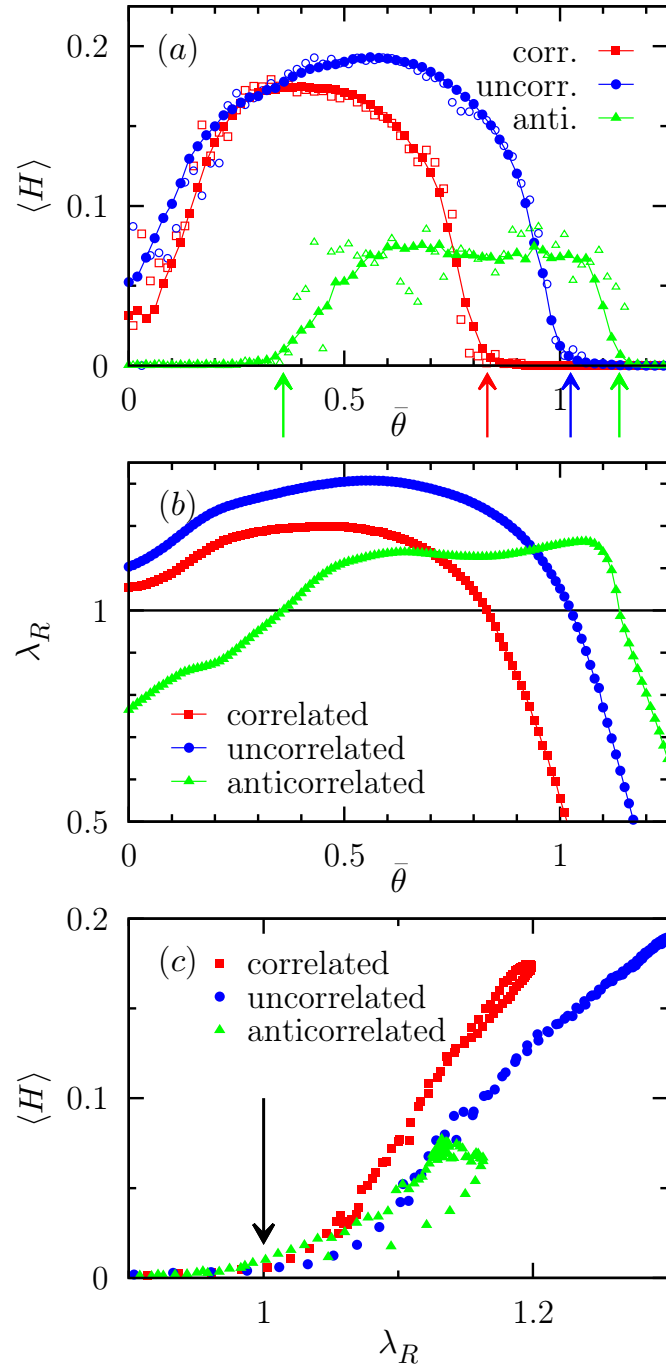


Figure 4.2: $\langle H \rangle$ and λ_R for a threshold network. The average threshold $\bar{\theta}$ is used as a tuning parameter in (a) and (b). Arrows mark points where $\lambda_R = 1$ and correctly predict transitions. See text of Section 4.4.2 for details.

Eqs. (4.2), (4.3), and (4.4b),

$$\bar{F}_i(X_i) = \Phi \left[\frac{1}{\sigma_\theta} \left(\sum_j W_{ij} x_j - \bar{\theta} \right) \right], \quad (4.16a)$$

$$D_i(X_i, X_i^j) = |\bar{F}_i(X_i) - \bar{F}_i(X_i^j)|, \quad (4.16b)$$

where $\Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^x \exp(-t^2/2) dt$. Equation (4.16) can be used to find p_i , R_{ij} , and λ_R . Here, as in many cases, it is not necessary to enumerate the ensemble of update rules or find the associated probabilities, because \bar{F}_i and D_i can be calculated directly. When finding the average normalized Hamming distance $\langle H \rangle$ in the numerical simulations, θ_i is treated as quenched by writing $\theta_i = \bar{\theta} + \delta\theta_i$, where $\delta\theta_i$ is drawn from a normal distribution with mean 0 and standard deviation σ_θ .

Figure 4.2 shows results for both a single quenched set of $\delta\theta_i$ (hollow markers) as well as an average over 50 quenched sets of $\delta\theta_i$ (filled markers). Typical single quenched realizations show similar behavior to the average, in agreement with the semi-annealing hypothesis. More striking is the qualitative difference between the anticorrelated case and the correlated and uncorrelated cases. At low thresholds, the anticorrelated network is stable, whereas both the correlated and uncorrelated case are unstable. As the threshold is increased, the anticorrelated network becomes unstable before becoming stable again at large thresholds. This agrees with panel (b), where in all three cases, λ_R initially increases with increasing threshold, but the anticorrelated case is the only one where $\lambda_R < 1$ initially. Finally, in panel (c), $\langle H \rangle$ is plotted against λ_R using the same data. It is clear that in all three cases the stability transition clearly occurs at $\lambda_R = 1$, confirming the analysis.

The behavior of the anticorrelated case again illustrates the importance of the

fact that the stability criterion Eq. (4.15) can accommodate correlations between network topology and update rules, in contrast to fully annealed techniques. Note that all three cases in Fig. 4.2(a) share the same network topology and thresholds, as well as the distribution of weights. Therefore, any theory which does not take into account the *joint* effects of network topology and update rules would be unable to predict the marked difference in the behavior of these three cases.

4.5 Discussion

This chapter presents a general framework for addressing the question of orbit stability of large, locally tree-like Boolean networks, given arbitrary network topology and update rules. There are four steps in this process: (1) identify the update rule ensemble \mathcal{T}_i (or the functions \bar{F}_i and D_i) for each node i ; (2) calculate the dynamical biases p_i of the each node i by iterating Eqs. (4.5-4.7); (3) calculate the effective activities R_{ij} using Eq. (4.13b); and (4) find the largest eigenvalue of R , λ_R , which determines the stability of the system through Eq. (4.15). As illustrated above, the first step requires a judicious selection of which aspects of the update rules should remain quenched, but is typically straightforward thereafter. The second step—and, in general, the use of dynamical biases in the analysis—is essential for all but the simplest classes of update rules, because the probability of damage spreading from one of a node’s inputs depends on the states of the other inputs.

For this reason, the analysis presented here is the first which can be applied to specific complex Boolean networks with general update rules, taking into account

the steady-state dynamics of the system. As examples of the application of the general stability criterion, both a pedagogical case and the case of threshold networks were presented. Whereas research into the stability of Boolean networks has primarily focused on either topology or update rules alone, these results confirm that correlations between the two can have profound qualitative effects on the dynamical properties of a Boolean network. The dynamical biases p_i and the stability indicator λ_R offer systematic measures for evaluating these effects. As discussed in Section 4.3, this approach also makes it possible to study the effects of interventions which modify either the topology or update rules of a given Boolean network, because specific networks with quite general update rules can be treated.

Further consequences and development of these results remain to be explored. Because the activity matrix R depends on both the dynamical biases p and the damage probability functions D_i , introducing any feature to the network or update rules can affect λ_R in several (sometimes conflicting) ways, as in the case of threshold networks. Further exploration of other features is likely to yield other interesting examples. Another interesting direction of work is to develop techniques similar to those used in Refs. [8, 17, 80], which would allow p_i and λ_R to be estimated analytically in particular cases without solving Eqs. (4.5-4.7) numerically. This would allow greater intuitive understanding of the effects of various characteristics of the network or update rules, as well as the interaction between these characteristics. I hope that the results here will draw more attention to these issues, and, more broadly, to the fascinating interplay between topology and update rules in the dynamics of Boolean networks.

Appendices

In the following appendices, I discuss four extensions and implications of the above work. Section 4.A shows that the stability criterion, Eq. (4.15), can be applied to asynchronously updated Boolean networks as well. Section 4.B derives the critical slope of the stability phase transition using the above framework, and Section 4.C discusses the computational scaling of evaluating λ_R . Finally, Section 4.D discusses the contributions of individual nodes to instability and explains how the framework introduced here can be used to evaluate the effects of modifications to a Boolean network.

4.A Asynchronous updates

Asynchronous updates may arise in discrete state systems for several reasons. For example, links may have nonuniform delays, δ_{ij} , that model delays arising from, for example, the chemical kinetics of gene regulation. In this case, the dynamics would be described by a modified version of Eq. (4.1) in which the state of node i at time t depends on the states of its inputs j at times $(t - \delta_{ij})$. Another alternative is a model in which nodes are individually chosen to be updated in a stochastically determined order. Here, it is shown that the stability condition given in the main text applies not only to the case of synchronous nodal updates, but to asynchronous models as well, including both of these examples.

In particular, consider updates which occur at times $\tau_1 < \tau_2 < \dots < \tau_t < \dots$,

where the update intervals, $(\tau_{t+1} - \tau_t)$, are arbitrary, incommensurate and do not influence the analysis. Since the update times are incommensurate, approximate the deterministic choice of node to update at each time, indexed by integer t , with a stochastic process where node i (and only node i) is chosen to be updated by Eq. (4.1) with probability ω_i . This is also appropriate, of course, for systems that are inherently stochastic.

To analyze this case, the approximate update equation, Eq. (4.14), must be adjusted. Since node i is chosen independently of the values of the nodes, the joint probability at time step t that node i is chosen for update and that node i differs between the two initial conditions after the update is given approximately by $\omega_i \sum R_{ij} y_j(t)$. If node i is not chosen for update at this time step, y_i does not change. Putting this together yields, for small t and small initial perturbations, $y_i(t+1) \approx \omega_i \sum R_{ij} y_j(t) + (1 - \omega_i) y_i(t)$, which can be written in matrix form as

$$y(t+1) = \Omega(R - I)y(t) + y(t), \quad (4.17)$$

where Ω is a diagonal matrix with ω_i in each row, I is the identity matrix, and R is the activity matrix. In order to see that Eq. (4.15) also applies in this case, note that, at criticality, $y(t+1) = y(t)$, so that Eq. (4.17) reduces to $\Omega(R - I)y(t) = 0$. This has a solution for $y \neq 0$ only if $\lambda_R = 1$. Note, however, that in this case, for $\lambda_R > 1$, the growth rate of the Hamming distance will be at a rate of the order of $1/N$ smaller than the rate of the synchronously updated networks, because N time steps of asynchronous update correspond to one time step of synchronous update.

4.B Critical slope

The second-order terms in the expansion in Eq. (4.13a) may be used to derive the critical slope of H near $\lambda_R = 1$. A sketch of this derivation is included because it may be useful for near-critical approximations or for designing networks with extreme behavior near the critical point. Begin by considering input combinations in which two distinct inputs j and k are damaged, denoted $\tilde{X} = X_i^{j,k}$ in analogy with the definition of X_i^j above. Following similar steps as those that led to Eq. (4.13a) and retaining all terms of $\mathcal{O}(y^2)$, gives

$$y_i = \sum_j R_{ij} y_j + \sum_{j,k} R_{ijk} y_j y_k, \quad (4.18)$$

$$R_{ijk} \equiv \frac{1}{2} \sum_{X_i} \Pr[X_i] D_i [X_i, X_i^{j,k}] - R_{ij},$$

where R_{ij} is defined as in the main text, and it is assumed that y has reached its steady state behavior. Note that when $j = k$, $R_{ijk} = 0$.

Now the critical slope may be derived through a perturbation expansion near the critical point, $y_i = \varepsilon H y_i^1 + \varepsilon^2 y_i^2$ and $\lambda_R = 1 + \varepsilon \lambda_R^1$, where superscripts for y_i and λ_R refer to the level of the perturbation expansion. From Eq. (4.14), y^1 must be equal to the right Frobenius-Perron eigenvector of R , denoted v here. It is convenient to normalize v so that $\sum_i v_i = 1$. Inserting the second-order expansion and simplifying yields

$$y_i^2 = H \lambda_R^1 v_i + \sum_j R_{ij} y_j^2 + H^2 \sum_{j,k} R_{ijk} v_j v_k \quad (4.19)$$

This expression may be simplified by using the left Frobenius-Perron eigenvector, denoted u . Multiplying through by u_i and summing over i , the left-hand side and

the second term on the right-hand side of Eq. (4.19) cancel to leading order in ε .

With the remaining terms, the critical slope $m_c = H/\lambda_R^1$ is given by

$$m_c = -\frac{\sum_i u_i v_i}{\sum_{i,j,k} R_{ijk} u_i v_j v_k}. \quad (4.20)$$

This result may be used numerically to find the critical slope in particular cases. It may also be used to approximate the critical slope analytically, when good approximations for u and v are known, as in Refs. [8, 17, 80].

4.C Computational considerations

Because the theory presented here uses dynamical biases and the locally tree-like approximation, it is applicable to very large values of N for sparse networks. This offers a tremendous computational improvement over previous theoretical treatments of similar systems; for example, the analysis of probabilistic Boolean networks in Ref. [79] relies upon a state transition matrix of size $2^N \times 2^N$, which is intractable for networks with more than a few dozen nodes. Numerical results also suggest that finding λ_R is much more efficient in typical cases than direct computation of the Hamming distance, which requires $\mathcal{O}(n_{ic} t_3 E)$ operations, where n_{ic} is the number of initial conditions used, t_3 is as defined in Section 4.4, and E is the number of edges. This is because n_{ic} and t_3 must be relatively large in order to obtain good estimates for $\langle H \rangle$.

The most computationally intensive step in the analysis is iterating Eqs. (4.5-4.7), which requires on the order of $2^{d_i^{\text{in}}}$ steps for each i (see below). Calculating R_{ij} for each edge also requires $2^{d_i^{\text{in}}}$ steps, but only needs to be performed once. The

eigenvalue λ_R may be found through power iteration, which requires only $\mathcal{O}(E)$ operations per iteration (using sparse matrix-vector multiplication) and also converges quickly.

Iterating Eqs. (4.5-4.7) is fast for large- N networks as long as most nodes have $d_i^{\text{in}} < 15$, because few iterations are needed for convergence. In cases where the average degree is large, direct evaluation of p_i becomes intractable, but in many cases additional simplifying assumptions can be used for such systems. For example, for threshold networks with high in-degree, rather than enumerating all $2^{d_i^{\text{in}}}$ possible input states, one could use the central limit theorem to approximate the distribution of $\sum_j W_{ij}x_j$. In the case of moderate values of d_i^{in} , it is also worth noting that, while a naïve approach would use $d_i^{\text{in}}2^{d_i^{\text{in}}}$ steps to calculate the set of $\text{Pr}[X_i]$ for node i , this can be achieved somewhat more efficiently through the following process when $d_i^{\text{in}} > 2$. Begin with a list containing the values $(1 - p_j)$ and p_j for some input j . Then for each additional input k , copy the list, multiply one copy by $(1 - p_k)$ and the other by p_k , then join the two copies. This way, all $\text{Pr}[X_i]$ are generated in approximately $2^{d_i^{\text{in}}+1}$ operations.

4.D Individual contributions to instability

A great deal of information about the effects of network interventions (as discussed at the end of Section 4.3) can be derived from the left- and right-eigenvectors associated to λ_R , which may be obtained through power iteration along with λ_R . Here, these eigenvectors are denoted u and v , respectively. For example, to first or-

der, it is only possible to cause $Y > 0$ by initially damaging at least one node with $u_i \neq 0$, and only nodes with $v_i \neq 0$ will have $y_i(t) > 0$ when t is large. These results are intuitive in light of the relationship between the stability of Boolean networks and percolation explored in Section 3.3.2 and Ref. [2].

It is also possible to use u and v to estimate the contribution of an individual node to damage spreading in the network, using the concept of dynamical importance introduced in Ref. [82]. Let $\Delta\lambda_R$ denote the change in λ_R that would result if a node i were unable to spread damage to its outputs. Then $\Delta\lambda_R/\lambda_R \approx u_i v_i / u^T v$. A similar result for edge removal is also available [82]. These results are useful for understanding the flow of damage through a specific network, as well as for stabilization in systems where there is a mechanism for correcting the states of damaged nodes.

Another approach to intervening in a network, more applicable to gene regulatory applications such as those discussed in Section 4.1, is to silence nodes or edges. In the theory, silencing a node would correspond to forcing $p_k = 0$ for some k , and silencing an edge would correspond to considering only the values of X_i in which $x_j = 0$ in the sums in Eqs. (4.5-4.7) and (4.13b). The change in p after such an intervention describes its effect on the behavior of the rest of the system (i.e., for gene regulation, the change in the expression levels of other genes), and the resulting change in λ_R describes the overall effect on the stability of the system.

Given the goal of stabilizing an unstable network through a small intervention, one could compare a number of possible interventions and choose the one which results in the smallest value of λ_R . When doing so, it is best to repeat the analysis of

Section 1.6 for each modified system, rather than applying a perturbative approximation, because the change in p which results from silencing a node can affect R and λ_R in complicated ways. Even so, a large number of such interventions can be compared in a computationally efficient manner by storing the values of p and v for the original system. These can then be used as initial conditions for the iterative processes used to find the dynamical biases and λ_R , decreasing the convergence time for both processes.

Notes for Chapter 4

¹Attractors which correspond to periodic or chaotic orbits in the map defined by Eqs. (4.5-4.7) are not considered here, although they have been shown to exist in unusual cases [67, 83]. Such cases are unlikely to correspond to biologically realistic dynamics.

²These statements follow directly from the Frobenius-Perron theorem when R is a primitive matrix. They are also nearly always true for networks which are above the percolation threshold without strong community structure. In this case, it is safe to assume that the network has only a single macroscopic strongly connected component, whose associated Frobenius-Perron eigenvalue dominates all other eigenvalues. The effects of community structure on the spectra of adjacency matrices are discussed in detail in Ref. [84].

Glossary of Frequently Used Notation

a_{kl}	fraction of nodes with in-degree k and out-degree l
A	adjacency matrix of network
b_i	update rule bias of i
c_i	index of canalizing input to i
d_i	degree of node i
D_i	probability of i becoming damaged, given X_i and \tilde{X}_i
E	number of edges in a network
f	an arbitrary update rule
F_i	update rule for node i in a Boolean network
\bar{F}_i	average of F_i over \mathcal{T}_i
g	universal scaling function for S
G	generating function
h	another universal scaling function for S
H	Hamming distance
\mathcal{J}_i	the set of network inputs to i
K	degree of homogeneous network
m	number of candidate edges in a network growth process
M	map for the time-evolution of y
N	number of nodes in a network
q_i	sensitivity of i , or the probability of non-deletion in percolation
p_i	dynamical bias of i
R	matrix describing damage spreading in a Boolean network
S	order parameter for percolation
t	time
T	theoretical prediction for Y and S in stability-percolation map
\mathcal{T}_i	update rule ensemble for i in generalized semi-annealing
u	left eigenvector
v	right eigenvector
w	complex number
W	matrix of weights in a threshold network
x	state of a Boolean network
\tilde{x}	state of a perturbed orbit in a Boolean network
X_i	combined state of input nodes to i in orbit x
\tilde{X}_i	combined state of input nodes to i in orbit \tilde{x}
X_i^j	equal to X_i except that the j^{th} component is damaged
y_i	the probability that i is damaged
Y	order parameter for stability of a Boolean network
z	average degree

(continued on next page)

Glossary of Frequently Used Notation (continued)

α	the fraction of nodes in a Boolean network with XOR update rules
β	critical exponent for the scaling of the order parameter
ε	fraction of nodes damaged in a perturbation
η_i	the probability that i is not in GOUT
κ	exponent for the scaling of the maximum ΔS with N
λ_R	the Frobenius-Perron eigenvalue for R
ξ	exponent for the scaling of the maximum variance of S with N
ρ	assortativity of a network
ϕ	critical exponent for the scaling of the critical region
θ_i	threshold of i in a threshold network

Bibliography

- [1] S. Squires, K. Sytwu, D. Alcalá, T. M. Antonsen, E. Ott, and M. Girvan, “Weakly explosive percolation in directed networks,” *Physical Review E* **87**, 052127 (2013).
- [2] S. Squires, E. Ott, and M. Girvan, “Dynamical instability in Boolean networks as a percolation problem,” *Physical Review Letters* **109**, 085701 (2012).
- [3] S. Squires, A. Pomerance, M. Girvan, and E. Ott, “Stability of Boolean networks: The joint effects of topology and update rules,” arXiv:1310.1338 (2013).
- [4] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Review* **45**, 167 (2003).
- [5] R. Solomonoff and A. Rapoport, “Connectivity of random nets,” *The Bulletin of Mathematical Biophysics* **13**, 107 (1951).
- [6] P. Erdős and A. Rényi, “On the evolution of random graphs,” *Magyar Tud. Akad. Mat. Kutató Int. Közl* **5**, 17 (1960).
- [7] M. E. J. Newman, “Mixing patterns in networks,” *Physical Review E* **67**, 026126 (2003).
- [8] J. G. Restrepo, E. Ott, and B. R. Hunt, “Approximating the largest eigenvalue of network adjacency matrices,” *Physical Review E* **76**, 056119 (2007).
- [9] J. M. Yeomans, *Statistical Mechanics of Phase Transitions* (Oxford University Press, 1992).
- [10] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Random graphs with arbitrary degree distributions and their applications,” *Physical Review E* **64**, 026118 (2001).
- [11] M. Molloy and B. Reed, “A critical point for random graphs with a given degree sequence,” *Random Structures & Algorithms* **6**, 161 (1995).

- [12] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph structure in the web,” *Computer Networks* **33**, 309 (2000).
- [13] S. A. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets,” *Journal of Theoretical Biology* **22**, 437 (1969).
- [14] H. de Jong, “Modeling and simulation of genetic regulatory systems: a literature review.” *Journal of Computational Biology* **9**, 67 (2002).
- [15] S. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution* (Oxford Univ. Press, 1993).
- [16] J. M. Beggs and D. Plenz, “Neuronal avalanches in neocortical circuits,” *Journal of Neuroscience* **23**, 11167 (2003).
- [17] A. Pomerance, E. Ott, M. Girvan, and W. Losert, “The effect of network topology on the stability of discrete state models of genetic control,” *Proceedings of the National Academy of Sciences* **106**, 8209 (2009).
- [18] H. C. Bravo, V. Pihur, M. McCall, R. A. Irizarry, and J. T. Leek, “Gene expression anti-profiles as a basis for accurate universal cancer signatures,” *BMC Bioinformatics* **13**, 272 (2012).
- [19] D. Achlioptas, R. M. D’Souza, and J. Spencer, “Explosive percolation in random networks,” *Science* **323**, 1453 (2009).
- [20] O. Riordan and L. Warnke, “Explosive percolation is continuous,” *Science* **333**, 322 (2011).
- [21] P. Grassberger, C. Christensen, G. Bizhani, S. W. Son, and M. Paczuski, “Explosive percolation is continuous, but with unusual finite size behavior,” *Physical Review Letters* **106**, 225701 (2011).
- [22] R. A. da Costa, S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, “Explosive percolation transition is actually continuous,” *Physical Review Letters* **105**, 255701 (2010).
- [23] J. Nagler, A. Levina, and M. Timme, “Impact of single links in competitive percolation,” *Nature Physics* **7**, 265 (2011).
- [24] J. Gómez-Gardeñes, S. Gómez, A. Arenas, and Y. Moreno, “Explosive synchronization transitions in scale-free networks,” *Physical Review Letters* **106**, 128701 (2011).
- [25] S. Angst, S. R. Dahmen, H. Hinrichsen, A. Hucht, and M. P. Magiera, “Explosive Ising,” *Journal of Statistical Mechanics* **2012**, L06002 (2012).
- [26] R. M. Ziff, “Explosive growth in biased dynamic percolation on two-dimensional regular lattice networks,” *Physical Review Letters* **103**, 045701 (2009).

- [27] T. Bohman, A. Frieze, and N. C. Wormald, “Avoidance of a giant component in half the edge set of a random graph,” *Random Structures & Algorithms* **25**, 432 (2004).
- [28] W. Chen and R. M. DSouza, “Explosive percolation with multiple giant components,” *Physical Review Letters* **106**, 115701 (2011).
- [29] N. A. M. Araújo and H. J. Herrmann, “Explosive percolation via control of the largest cluster,” *Physical Review Letters* **105**, 035701 (2010).
- [30] K. J. Schrenk, N. A. M. Araújo, and H. J. Herrmann, “Gaussian model of explosive percolation in three and higher dimensions,” *Physical Review E* **84**, 041136 (2011).
- [31] S. Boettcher, V. Singh, and R. M. Ziff, “Ordinary percolation with discontinuous transitions,” *Nature Communications* **3**, 787 (2012).
- [32] S. D. S. Reis, A. A. Moreira, and J. S. Andrade, “Nonlocal product rules for percolation,” *Physical Review E* **85**, 041112 (2012).
- [33] J. Nagler, T. Tiessen, and H. W. Gutch, “Continuous percolation with discontinuities,” *Physical Review X* **2**, 031009 (2012).
- [34] Y. S. Cho, S. Hwang, H. J. Herrmann, and B. Kahng, “Avoiding a spanning cluster in percolation models,” *Science* **339**, 1185 (2013).
- [35] J. S. Andrade, H. J. Herrmann, A. A. Moreira, and C. L. N. Oliveira, “Transport on exploding percolation clusters,” *Physical Review E* **83**, 031133 (2011).
- [36] N. A. M. Araújo, J. S. Andrade, R. M. Ziff, and H. J. Herrmann, “Tricritical point in explosive percolation,” *Physical Review Letters* **106**, 095703 (2011).
- [37] E. J. Friedman and A. S. Landsberg, “Construction and analysis of random networks with explosive percolation,” *Physical Review Letters* **103**, 255701 (2009).
- [38] M. E. J. Newman and R. M. Ziff, “Efficient monte carlo algorithm and high-precision results for percolation,” *Physical Review Letters* **85**, 4104 (2000).
- [39] R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM Journal on Computing* **1**, 146 (1972).
- [40] S. Manna, “About the fastest growth of the order parameter in models of percolation,” *Physica A: Statistical Mechanics and its Applications* **391**, 2833 (2012).
- [41] F. Radicchi and S. Fortunato, “Explosive percolation: A numerical analysis,” *Physical Review E* **81**, 036110 (2010).
- [42] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford Univ. Press, 1999).

- [43] O. Riordan and L. Warnke, “Achlioptas processes are not always self-averaging,” *Physical Review E* **86**, 011129 (2012).
- [44] B. Derrida and Y. Pomeau, “Random networks of automata: A simple annealed approximation,” *Europhysics Letters* **1**, 45 (1986).
- [45] U. Bastolla and G. Parisi, “Closing probabilities in the Kauffman model: An annealed computation,” *Physica D* **98**, 1 (1996).
- [46] S. Maslov and K. Sneppen, “Specificity and stability in topology of protein networks,” *Science* **296**, 910 (2002).
- [47] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network motifs: Simple building blocks of complex networks,” *Science* **298**, 824 (2002).
- [48] Q. Cui, Y. Ma, M. Jaramillo, H. Bari, A. Awan, S. Yang, S. Zhang, L. Liu, M. Lu, M. O’Connor-McCourt, E. O. Purisima, and E. Wang, “A map of human cancer signaling,” *Molecular Systems Biology* **3** (2007).
- [49] A. Mozeika and D. Saad, “Dynamics of Boolean networks: An exact solution,” *Physical Review Letters* **106**, 214101 (2011).
- [50] C. Seshadhri, Y. Vorobeychik, J. R. Mayo, R. C. Armstrong, and J. R. Ruthruff, “Influence and dynamic behavior in random Boolean networks,” *Physical Review Letters* **107**, 108701 (2011).
- [51] H. Flyvbjerg, “An order parameter for networks of automata,” *Journal of Physics A* **21**, L955 (1988).
- [52] T. Mihaljev and B. Drossel, “Scaling in a general class of critical random Boolean networks,” *Physical Review E* **74**, 046101 (2006).
- [53] B. Samuelsson and J. E. S. Socolar, “Exhaustive percolation on random networks,” *Physical Review E* **74**, 036113 (2006).
- [54] A. Hansen, “A connection between the percolation transition and the onset of chaos in the kauffman model,” *Journal of Physics A* **21**, 2481 (1988).
- [55] S. P. Obukhov and D. Stauffer, “Upper critical dimension of Kauffman cellular automata,” *Journal of Physics A* **22**, 1715 (1989).
- [56] I. Shmulevich and S. A. Kauffman, “Activities and sensitivities in Boolean network models,” *Physical Review Letters* **93**, 048701 (2004).
- [57] R. V. Solé and B. Luque, “Phase transitions and antichaos in generalized Kauffman networks,” *Physics Letters A* **196**, 331 (1994).
- [58] B. Luque and R. V. Solé, “Phase transitions in random networks: Simple analytic determination of critical points,” *Physical Review E* **55**, 257 (1997).

- [59] J. J. Fox and C. C. Hill, “From topology to dynamics in biochemical networks,” *Chaos* **11**, 809 (2001).
- [60] M. Aldana and P. Cluzel, “A natural class of robust networks,” *Proceedings of the National Academy of Sciences* **100**, 8710 (2003).
- [61] M. Aldana, “Boolean dynamics of networks with scale-free topology,” *Physica D* **185**, 45 (2003).
- [62] D.-S. Lee and H. Rieger, “Comparative study of the transcriptional regulatory networks of *E. coli* and yeast: Structural characteristics leading to marginal dynamic stability,” *Journal of Theoretical Biology* **248**, 618 (2007).
- [63] D.-S. Lee and H. Rieger, “Broad edge of chaos in strongly heterogeneous Boolean networks,” *Journal of Physics A* **41**, 415001 (2008).
- [64] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, “Random Boolean network models and the yeast transcriptional network,” *Proceedings of the National Academy of Sciences* **100**, 14796 (2003).
- [65] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, “Genetic networks with canalizing Boolean rules are always stable,” *Proceedings of the National Academy of Sciences* **101**, 17102 (2004).
- [66] A. A. Moreira and L. A. N. Amaral, “Canalizing Kauffman networks: Nonergodicity and its effect on their critical behavior,” *Physical Review Letters* **94**, 218702 (2005).
- [67] J. Kesseli, P. Rämö, and O. Yli-Harja, “Iterated maps for annealed Boolean networks,” *Physical Review E* **74**, 046104 (2006).
- [68] T. Rohlf and S. Bornholdt, “Criticality in random threshold networks: annealed approximation and beyond,” *Physica A: Statistical Mechanics and its Applications* **310**, 245 (2002).
- [69] F. Greil and B. Drossel, “Kauffman networks with threshold functions,” *The European Physical Journal B* **57**, 109 (2007).
- [70] A. Szejka, T. Mihaljev, and B. Drossel, “The phase diagram of random threshold networks,” *New Journal of Physics* **10**, 063009 (2008).
- [71] J. G. T. Zañudo, M. Aldana, and G. Martínez-Mekler, “Boolean threshold networks: Virtues and limitations for biological modeling,” in *Information Processing and Biological Systems*, Intelligent Systems Reference Library No. 11, edited by S. Niiranen and A. Ribeiro (Springer Berlin Heidelberg, 2011) pp. 113–151.
- [72] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Network robustness and fragility: Percolation on random graphs,” *Physical Review Letters* **85**, 5468 (2000).

- [73] J. G. Restrepo, E. Ott, and B. R. Hunt, “Weighted percolation on directed networks,” *Physical Review Letters* **100**, 058701 (2008).
- [74] S. Melnik, A. Hackett, M. A. Porter, P. J. Mucha, and J. P. Gleeson, “The unreasonable effectiveness of tree-based theory for networks with clustering,” *Physical Review E* **83**, 036112 (2011).
- [75] S. E. Harris, B. K. Sawhill, A. Wuensche, and S. Kauffman, “A model of transcriptional regulatory networks based on biases in the observed regulation rules,” *Complexity* **7**, 23 (2002).
- [76] F. G. Frobenius, *Über Matrizen aus nicht negativen Elementen* (Königliche Akademie der Wissenschaften, 1912).
- [77] J. Kennan, “Uniqueness of positive fixed points for increasing concave functions on \mathbb{R}^n : An elementary result,” *Review of Economic Dynamics* **4**, 893 (2001).
- [78] A. Pomerance, M. Girvan, and E. Ott, “Stability of Boolean networks with generalized canalizing rules,” *Physical Review E* **85**, 046106 (2012).
- [79] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks,” *Bioinformatics* **18**, 261 (2002).
- [80] E. Ott and A. Pomerance, “Approximating the largest eigenvalue of the modified adjacency matrix of networks with heterogeneous node biases,” *Physical Review E* **79**, 056111 (2009).
- [81] K. E. Kurten, “Correspondence between neural threshold networks and Kauffman Boolean cellular automata,” *Journal of Physics A: Mathematical and General* **21**, L615 (1988).
- [82] J. G. Restrepo, E. Ott, and B. R. Hunt, “Characterizing the dynamical importance of network nodes and links,” *Physical Review Letters* **97**, 094102 (2006).
- [83] M. Andrecut and M. K. Ali, “Chaos in a simple Boolean network,” *International Journal of Modern Physics B* **15**, 1723 (2001).
- [84] S. Chauhan, M. Girvan, and E. Ott, “Spectral properties of networks with community structure,” *Physical Review E* **80**, 056114 (2009).