Dynamical Instability in Boolean Networks as a Percolation Problem

Shane Squires,* Edward Ott, and Michelle Girvan
University of Maryland—College Park
(Dated: May 3, 2012)

Boolean networks, widely used to model gene regulation, exhibit a phase transition between regimes in which small perturbations either die out or grow exponentially. We show and numerically verify that this phase transition in the dynamics can be mapped onto a static percolation problem which predicts the long-time average Hamming distance between perturbed and unperturbed orbits.

Boolean networks have been a prominent tool for modeling gene regulation since their introduction by S. Kauffman in 1969 [1, 2]. A Boolean network is a directed network in which each node is assigned a state, 0 or 1, which is synchronously updated at discrete time steps according to a preassigned update function that depends on the states of the node’s inputs on the previous time step. When used as a model of gene regulatory networks, each node represents a gene, and the state of the node indicates whether or not the gene is being expressed. Edges correspond to regulatory interactions, whose biochemical details are modeled by the update functions. Partly because of the prevalence of noise in biological systems, there has been a great deal of interest in the stability of Boolean networks with respect to small perturbations. In his treatment, Kauffman considered random update functions and random networks in which each of the N inputs has K input links from other randomly chosen nodes (the N-K model). Kauffman found numerically that when the in-degree K crosses a critical value, there is a phase transition between a stable phase, in which small perturbations die out, to a “chaotic” phase, in which small perturbations grow and become macroscopic.

An analytic derivation of the critical in-degree was given by Derrida and Pomeau [3] for annealed N-K networks, where by annealed we mean that the network edges and update functions are randomly redrawn between time steps. They hypothesized that for large networks the stability properties of the annealed system are similar to those of the original frozen (non-annealed) system. This hypothesis is well-supported by numerical experiments [3, 4], and we refer to it as the “annealed approximation.” Recent work related to our study [5] has vastly extended the applicability of 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 [6], motifs [7], and community structure [8]. The only necessary assumption is that the network must be locally treelike (it cannot have many short loops) [9].

Some recent papers have derived stability properties of Boolean networks without the use of an annealing procedure [10, 11]. These papers should be considered complementary to ours 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. By contrast, because our results rely on the semi-annealed approximation, they can model the dynamics of a specific network.

Here, using our semi-annealed approach, we map the problem of dynamical instability on a Boolean network onto a static network percolation problem in the N → ∞ limit. Previous authors have discussed maps between the N-K model and percolation on lattices rather than networks [12, 13], and others have discussed the percolation properties of the “frozen component” of a Boolean network [14–16]. In contrast, we show 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 defined on that network. We will illustrate this map in three different contexts. First, we consider the well-known annealed approximation and map it onto percolation in the configuration model [17]. Second, we give a similar map from the semi-annealed approximation [5] to weighted site percolation [18]. Finally, we treat a more general class of update functions by mapping to a correlated bond percolation problem.

Model: A Boolean network is a directed network of N nodes, in which each node i is assigned a state, x_i(t) = 0 or x_i(t) = 1, at each discrete time step t. The in- and out-degrees of node i will be denoted d^in_i and d^out_i, and we will refer to the set of inputs to node i as J_i. The time evolution of node i is governed by a Boolean function or “truth table” F_i, which is fixed in time and takes as arguments the states of all incoming nodes at the previous time step. That is, x_i(t) = F_i({x_j(t−1) : j ∈ J_i}).

In the literature, the truth tables F_i are usually generated randomly (e. g., [3]); for each combination of input states to a node i, the value of F_i is assigned to be 1 with probability p or 0 with probability 1 − p, where p is the “bias probability.” Below, as in [5], we will consider the more general case where each F_i is generated by a different bias p_i assigned to each node i. Later, we will also consider the case of “canalizing” functions, in which one input acts as a master switch for the truth table. That is, an input j to a 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, the value of \( F_i \) depends on the states of its other inputs.) Canalizing functions are thought to be common in real gene networks [19, 20].

Consider two trajectories, \( \mathbf{x}(t) \) and \( \tilde{\mathbf{x}}(t) \), which evolve on the same Boolean network. The initial conditions \( \mathbf{x}(0) \) and \( \tilde{\mathbf{x}}(0) \) differ only on a small randomly chosen fraction \( \varepsilon \) of nodes. We will say that a node \( i \) is “damaged” at time \( t \) if \( x_i(t) \neq \tilde{x}_i(t) \), and our goal is to predict the extent of the damage at long times. Let \( y_i \) be the fraction of time that node \( i \) is damaged, i.e.,

\[
y_i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} |x_i(t) - \tilde{x}_i(t)|.
\]

The normalized long-time average Hamming distance \( Y = \langle y_i \rangle, \ 0 \leq Y \leq 1 \), is used as the order parameter for the stability phase transition. Here, the average \( \langle \cdot \rangle \) is taken over all nodes \( i \), then over all initial conditions which differ on a fraction \( \varepsilon \) of the nodes.

Analytic Results: First we treat the annealed approximation for random configuration-model networks [21]. We assume that the truth tables are randomly generated by a probability bias which depends only on degree. Let \( P_{jk} \) be the probability that a node has \( j \) inputs and \( k \) outputs, and let the bias of such a node be \( p_{jk} \). We define the sensitivity of such a node to be \( q_{jk} = 2p_{jk}(1 - p_{jk}) \) [22]. Using our definition, the sensitivity \( q_{jk} \in [0, 1] \) can be interpreted as the probability that a node with \( j \) inputs and \( k \) outputs will become damaged at time \( t \) if at least one of its inputs is damaged at time \( t - 1 \).

In the annealed approximation, \( Y \) can be predicted analytically using a method derived in [3] and [23], which can be explained as follows. Let \( z \) denote the average degree of the network, i.e., \( z = \sum_{j,k} jP_{jk} = \sum_{j,k} kP_{jk} \), and let \( E \) denote the probability that a randomly selected edge originates from a damaged node. A randomly selected edge originates from a node with \( j \) inputs and \( k \) outputs with probability \( \frac{kP_{jk}}{z} \), and such a node will become damaged with probability \( q_{jk} \) if it has at least one damaged input, which occurs with probability \( 1 - (1 - E)^j \). Therefore, \( E \) and \( Y \) satisfy the conditions

\[
E = \sum_{j,k} \frac{kP_{jk}}{z} q_{jk} \left[ 1 - (1 - E)^j \right],
\]

\[
Y = \sum_{j,k} P_{jk} q_{jk} \left[ 1 - (1 - E)^j \right].
\]

In the stable regime, these equations only have the trivial solution \( E = 0 \) and \( Y = 0 \), but there will be a nonzero solution in the unstable regime [23].

We will now show that Eq. (2) can be mapped onto the generating function formalism for treating weighted site percolation in directed configuration-model networks, which was developed in [17] and [24]. 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, above which there is a macroscopic connected component or “giant component.” This giant component contains a core of mutually path-connected nodes called the giant strongly connected component (GSCC); this, along with all the nodes which can be reached from it, is called the giant out-component (GOUT). In our map, we will identify the probability that a node is not deleted with the sensitivity, writing \( q_{i,j,k} \) for the probability that a node with \( j \) inputs and \( k \) outputs is undeleted. With this identification, we will go on to show that \( Y \) maps onto the expected fraction of nodes in GOUT, which we denote \( S \).

In [17, 24], it is shown that \( S \) can be found through the following procedure. First, the generating functions for the in-degrees of nodes and edges are defined as

\[
F_0(w) = \sum_{j,k} P_{jk} q_{jk} w^j,
\]

\[
F_1(w) = \sum_{j,k} \frac{kP_{jk}}{z} q_{jk} w^j.
\]

Next, \( u \) is defined as the probability that the in-component of a random edge is finite, and it is shown through diagrammatic expansion that \( u \) and \( S \) satisfy

\[
u = 1 - F_1(1) + F_1(u),
\]

\[
S = F_0(1) - F_0(u).
\]

We note that the substitutions \( E = 1 - u \) and \( Y = S \) map Eq. (2) onto Eqs. (3-4). Therefore, the phase transition to dynamic instability in this ensemble of random Boolean networks is equivalent to the static percolation phase transition on the same ensemble.

Our second result is a more general derivation of the same correspondence, using the framework of Pomerance et al. [5]. This framework applies to a specific locally treelike network in which each node \( i \) can have its own arbitrarily chosen bias \( p_i \), with an associated sensitivity \( q_i = 2p_i(1 - p_i) \). Once again, we will identify the sensitivity \( q_i \) with a site nondeletion probability and map the Hamming distance \( Y \), onto the size of GOUT, \( S \). We begin by writing an analogue of Eq. (2) for a specific node in a semi-annealed, locally treelike Boolean network,

\[
y_i = q_i \left[ 1 - \prod_{j \in J_i} (1 - y_j) \right].
\]

This is the long-time limit of a damage-spreading equation derived by Pomerance et al. [5], who noted that \( i \) will become damaged with probability \( q_i \) if at least one of its inputs is damaged. The assumption that the network is locally treelike is necessary because all the probabilities in the product are treated as being independent.

Restrepo et al. [18] found a similar condition for percolation on locally treelike directed networks in which the
probability that each node is not deleted is $q_i$. They define $\eta_i$ as the fraction of site-deleted networks for which node $i$ is not in GOUT, and they show that $\eta$ satisfies

$$\eta_i = 1 - q_i + q_i \prod_{j \in J_i} \eta_j, \quad (6)$$

because a node is not in GOUT when it is either deleted or when it has no inputs from GOUT. Under the map $y_i = 1 - \eta_i$, Eq. (5) maps onto Eq. (6). Because $Y = \langle y_i \rangle$ and $S = \langle 1 - \eta_i \rangle$, this map also yields $Y = S$. For $S$, the average $\langle \cdot \rangle$ is first taken over all nodes $i$, then over all node deletion trials.

We now introduce a third case, in which we consider Boolean networks with canalizing functions. For each of our previous results, we first identified a system of equations describing the long-time damage probabilities $y_i$ and then found a percolation problem described by the same equations. This method can be extended to canalizing functions, but because the truth table elements in a canalizing function are not generated independently, we will need to consider a new type of percolation problem which we call correlated bond percolation. Instead of typical bond percolation, in which each bond is occupied or deleted independently, we will consider joint probabilities where the deletion of two bonds may be correlated if they are both inputs to the same node.

Here we describe a correlated bond percolation problem that corresponds to a Boolean network whose truth tables 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 truth table 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 $p_i$. To begin, we imagine that the system is equally likely to be in any of its states. As we will show, it is then formally possible to obtain equations describing damage spreading in closed form. Based on our numerical results, we conjecture that these equations can be used to predict damage spreading in a large class of Boolean networks with frozen truth tables.

Working under the supposition that all system states are equally probable, we now derive an expression for $y_i$. Let $r_i$ denote the “activity” of $c_i$ on $i$ [22], defined 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, and the second case is equivalent to Eq. (5). Therefore, we find

$$y_i = r_i y_c + \frac{1}{2} q_i (1 - y_c) \left[ 1 - \prod_{j \in J_i} (1 - y_j) \right], \quad (7)$$

where $J_i' = J_i - \{c_i\}$ and $q_i$ is the sensitivity of the half of the truth table where $x_{c_i}$ is not in its canalizing state. It can be shown that this is equivalent to

$$\eta_i = 1 - r_i + \left( r_i - \frac{1}{2} q_i \right) \eta_c + \frac{1}{2} q_i \prod_{j \in J_i} \eta_j, \quad (8)$$

where $\eta_i = 1 - y_i$. 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. It can be shown that $q_i \leq 2r_i$ so that the second probability is nonnegative. Also, note that it is straightforward to describe the case where only some of the nodes have a canalizing input by using Eqs. (7-8) for those nodes and Eqs. (5-6) for the others.

Numerical Results: We begin with the map described by Eqs. (5-6), since it is more general than Eqs. (2-4). We wish to compare the long-time average Hamming distance $Y$ to the size of the giant out-component $S$ for particular networks. At the same time, we can compare both $Y$ and $S$ to the theoretical prediction given by the solution to Eq. (5), which we denote $T$.

Our algorithm is as follows. First we create a configuration-model network with $N = 10^5$ nodes. The data in the figures were obtained using networks with Poisson-distributed in-degrees and scale-free out-degrees; we have also tested other degree distributions and found...
similar results. If desired, we then enhance interesting topological features such as assortativity or feedforward loops using the same algorithms as in [5]. Next, we assign each node a bias $p_i$. These may be distributed randomly, or, if we wish to encourage (impede) instability on the network, we distribute them so that the nodal average $\langle q_i d_{ii}^{in} d_{ii}^{out} \rangle$ is maximized (minimized) [5]. For the data in the figures, the biases $p_i$ were distributed randomly so that the sensitivities $q_i$ form a uniform distribution on the interval $[1,5]$. We choose random initial conditions for $x$, and a randomly selected fraction $\varepsilon = .01$ of the nodes are flipped for the initial conditions of $x$.

To find $Y$, we time-evolve the system and average $|x_i(t) - \tilde{x}_i(t)|$ between $t = 900$ and $t = 1000$, averaging over 100 initial conditions. The theoretical prediction is found by iterating Eq. (5) until it converges to a solution $\tilde{y}_i$, then taking $T = \langle \tilde{y}_i \rangle$. However, it is less straightforward to calculate $S$, because a typical percolation problem is only guaranteed to have a single, well-defined giant out-component in the $N \to \infty$ limit. For reasons which are discussed in the Supplemental Material, we choose the following procedure. We delete each node $i$ with probability $1 - q_i$, and find any strongly connected components (SCCs) in the resulting network, where we define an SCC to be a mutually path-connected set of nodes containing at least one loop. We define $S$ to be the fraction of nodes which can be reached from at least one SCC, averaged over the ensemble of deletion trials. Numerically, we average $10^5$ deletion trials per network.

Figure 1 illustrates the relationship between $Y$, $S$, and $T$ for networks generated in this way. In Fig. 1(a), we see that $Y$ and $S$ have the same average values on the ensemble of random networks with given average degree $z$. However, in Fig. 1(b), we see that the prediction $Y = S$ sometimes fails for individual networks, especially near the phase transition. The deviations in Fig. 1(b) are primarily caused by the quenched disorder in the truth tables, which may cause orbits to fall onto attractors which visit only a small fraction of the state space (and so may deviate from the semi-annealed approximation).

In Fig. 2, we have averaged over this quenched disorder by choosing the truth tables from an ensemble of closely related frozen truth tables (but not networks) according to the following procedure. Before we time-evolve each new set of initial conditions, we perform a set of exchanges on the truth tables. For each edge $j \to i$, with probability $\frac{1}{2}$, we exchange $x_j = 0$ and $x_j = 1$ on the truth table for $i$. We note that there are two major differences between this and the semi-annealed approximation. In the latter, the truth tables are changed during the dynamics, whereas here they are only changed each time we start a 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 Figs. 2 and 3, we see that 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 the Supplemental Material.

In Fig. 4, we perform the same numerical experiment for the case in which each node has one canalizing input. We find that $Y$, $S$, and $T$ agree for individual networks when we use the map between Eqs. (7) and (8), but the map between Eqs. (5) and (6) completely fails for this case. This supports our argument that we retain significant input-specific information about the dynamics when we average over the quenched disorder in the truth tables.

**Discussion:** We have presented evidence that the stability of a Boolean network can be understood in terms of a related percolation problem on that network, where nodes or edges are deleted with probabilities determined from the truth tables. Our results relate two previously-studied cases (the annealed and semi-annealed approximations) to known results for percolation on networks. We have also shown that a case of biological interest, canalizing truth tables, can similarly be mapped onto a percolation problem. Numerical experiments show excellent agreement with our analytical results when averaged over an ensemble of frozen truth tables. Since percolation is a static property of the network and the deletion probabilities, this suggests that some of the attention directed toward the stability phase transition in Boolean networks can be directed toward the associated percolation properties of the underlying network.

**Acknowledgements:** This work was funded by ONR grant N000140710734 and ARO grant W911NF1210101.
The locally treelike condition is discussed in more detail for Boolean networks in [5] and for site percolation in [18]. Configuration-model random networks with finite average degree are locally treelike as $N \to \infty$ [17].

[9] The locally treelike condition is discussed in more detail for Boolean networks in [5] and for site percolation in [18]. Configuration-model random networks with finite average degree are locally treelike as $N \to \infty$ [17].