

TOPOLOGY AND DYNAMICS OF BOOLEAN NETWORKS WITH STRONG INHIBITION

YONGWU RONG

Department of Mathematics
The George Washington University
Washington, DC 20052, USA

CHEN ZENG

Department of Physics
The George Washington University
Washington, DC 20052, USA

CHRISTINA EVANS

Department of Mathematics
The George Washington University
Washington, DC 20052, USA

HAO CHEN AND GUANYU WANG

Department of Physics
The George Washington University
Washington, DC 20052, USA

ABSTRACT. A major challenge in systems biology is to understand interactions within biological systems. Such a system often consists of units with various levels of activities that evolve over time, mathematically represented by the dynamics of the system. The interaction between units is mathematically represented by the topology of the system. We carry out some mathematical analysis on the connections between topology and dynamics of such networks. We focus on a specific Boolean network model - the Strong Inhibition Model. This model defines a natural map from the space of all possible topologies on the network to the space of all possible dynamics on the same network. We prove this map is neither surjective nor injective. We introduce the notions of “redundant edges” and “dormant vertices” which capture the non-injectiveness of the map. Using these, we determine exactly when two different topologies yield the same dynamics and we provide an algorithm that determines all possible network solutions given a dynamics.

1. Introduction. Understanding interactions between various units of a biological system is a major challenge in systems biology. Such a system often consists of units with various levels of activity that evolve over time, mathematically represented by the dynamics of the system. The interaction between units can often be represented by the topology of a graph. The study of interconnections between topology and dynamics of such systems leads to interesting mathematical problems which could have important consequences in biology.

2000 *Mathematics Subject Classification.* Primary: 37N25; Secondary: 90B10, 92B99, 92C42.
Key words and phrases. Boolean network, dynamical system.

In this paper, we study such interconnections for a specific Boolean network model – the *Strong Inhibition Model*. In Section 2, we give definition of this model and provide relevant backgrounds. Given a system with n units, the network model defines a map from the space of all possible networks on these units (i.e. the space of all possible topologies) to the space of all possible dynamics of the state space on the same units. We show, in Section 3, that this map is neither one to one nor onto. We introduce the concepts of *dormant vertices* and *redundant edges*. They are used to completely characterize those networks that are determined by their dynamics, and to determine when different networks have the same dynamics. These ideas naturally lead to an algorithm, presented in Section 4, that solves for all networks with a given dynamics. In Section 5, we present some counting results that measure the extent of non-injectivity and non-surjectivity of the map.

2. The network model. We begin with the following Boolean network model which has been explored by several authors, e.g., [1, 2]. It is a version of the Hopfield network [3].

This network model consists of a graph $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ is the set of nodes and E is the set of (directed and colored) edges between nodes. Each unit of the system is represented by a node v_i (often abbreviated by node i) of G . At any given time, each node is either active (i.e. in state 1) or inactive (i.e. in state 0). The interactions between the nodes are denoted by directed edges that are colored red or green. A red (resp. green) edge from node j to node i (where $i \neq j$) indicates that j inhibits (stimulates) i . In addition, this model also includes self-degradation which is indicated by a red edge from node i to itself. The dynamics of the model is described by the following equation.

$$S_i(t+1) = \begin{cases} 1 & \sum_{j \neq i} a_{ji} S_j(t) > 0 \\ 0 & \sum_{j \neq i} a_{ji} S_j(t) < 0 \\ 0 & \sum_{j \neq i} a_{ji} S_j(t) = 0 \ \& \ a_{ii} = -1 \\ S_i(t) & \sum_{j \neq i} a_{ji} S_j(t) = 0 \ \& \ a_{ii} = 0, \end{cases} \quad (1)$$

where $S_i(t) \in \{0, 1\}$ denotes the state of node i at time t . The connection matrix (a_{ji}) encodes the network topology in which a_{ji} ($j \neq i$) takes the value $-\gamma$, 1, or 0, depending on whether node j inhibits (red edge), stimulates (green edge), or is not connected to, node i ; and a_{ii} takes the value -1 for self-degradation (red edge to itself) or 0 for no edge back to itself. The case of $a_{ii} = 1$ for self stimulation is not considered here. Note that self-degradation, if present on a node, say, node i , takes effect only when the total input to the node from all other nodes is zero (i.e., $\sum_{j \neq i} a_{ji} S_j(t) = 0$). The numerical value of γ ($\gamma \geq 1$) models the relative dominance of inhibition over stimulation.

In the limit of $\gamma \rightarrow \infty$, the above model becomes the so-called *Strong Inhibition Model* [4] that is characterized as follows:

1. If there is a red edge from an active node to node i , then node i will become inactive.
2. If there is no red edge from an active node to node i , and there is a green edge from an active node to i , then i will become active.
3. If there is no edge from an active node to node i , and there is a red edge from i to itself, then i will be inactive.

4. If none of the above conditions holds (i.e. there is no edge from any active node to i , and there is no edge from i to itself), then i will maintain its previous activity status.

Therefore, in the Strong Inhibition Model, a single inhibition (represented by a red edge from an active node) to a node is sufficient to turn the node off, regardless of how many stimulations. This is reasonable since inhibition is known to be dominant over stimulation in biomolecular interaction. Moreover, it has been shown that the network dynamics is usually not sensitive to the choice of γ [1].

One advantage of using the Strong Inhibition Model is the use of Boolean equations describing network solutions to a given dynamics. Given a network W , let us define Boolean variables $r_{ij} = r_{ij}(W), g_{ij} = g_{ij}(W)$ by

$$r_{ij} = \begin{cases} 1 & \text{if there is a red edge from node } i \text{ to node } j \\ 0 & \text{otherwise.} \end{cases} \quad \text{where } 1 \leq i, j \leq n.$$

$$g_{ij} = \begin{cases} 1 & \text{if there is a green edge from node } i \text{ to node } j \\ 0 & \text{otherwise.} \end{cases} \quad \text{where } 1 \leq i \neq j \leq n$$

Then each transition $S_i(t) \rightarrow S_i(t + 1)$ is equivalent to one of the following four equations.

$$0 \rightarrow 0 : \quad (\wedge_{j \neq i: s_j(t)=1} \overline{g_{ji}}) \vee (\vee_{j \neq i: s_j(t)=1} r_{ji}) = 1$$

$$0 \rightarrow 1 : \quad (\vee_{j \neq i: s_j(t)=1} g_{ji}) \wedge (\wedge_{j \neq i: s_j(t)=1} \overline{r_{ji}}) = 1$$

$$1 \rightarrow 0 : \quad (\wedge_{j \neq i: s_j(t)=1} \overline{g_{ji}} \wedge r_{ii}) \vee (\vee_{j \neq i: s_j(t)=1} r_{ji}) = 1$$

$$1 \rightarrow 1 : \quad (\vee_{j \neq i: s_j(t)=1} g_{ji} \vee \overline{r_{ii}}) \wedge (\wedge_{j \neq i: s_j(t)=1} \overline{r_{ji}}) = 1$$

This allows one to take advantage of the techniques in Boolean equations and to make connection with the well-known Satisfiability problem in solving these Boolean equations.

3. Non-injectivity and non-surjectivity of map ϕ . In this section, we study the map ϕ between topology and dynamics given by the Strong Inhibition Model, and determine if the map is injective or surjective. One might expect that different topologies yield different dynamics. We show, however, that this is not the case. In other words, the map ϕ is not injective. We introduce the notions of *dormant vertice* and *redundant edges* that explains such non-injectivity and further enables us to determine precisely when two networks have the same dynamics (Corollary 1). It is easy to see that the map ϕ cannot be onto. We will present an algorithm (Section 4) that determines when a given dynamics is in the image of ϕ , and solves for the network solutions if it is in the image.

Let $V = \{v_1, \dots, v_n\}$ be the set of nodes, and \mathcal{T} the set of all possible topologies of the networks on V , i.e. the set of all possible networks on these n nodes. Given any two vertices v_i and v_j , there could be a red edge, or a green edge, or no edge from i to j . Given any vertex v_i , there could be a red edge, or no edge from v_i to itself. It follows that the cardinality of \mathcal{T} is

$$|\mathcal{T}| = 3^{n(n-1)} 2^n$$

Let \mathcal{D} be the set of all possible dynamics on the state space on V . That is, \mathcal{D} is the set of all maps from $S(V)$ to itself, where $S(V)$, the state space, is the set of all possible states on V . The cardinality of $S(V)$ is $m = 2^n$. Therefore, the cardinality of \mathcal{D} is

$$|\mathcal{D}| = m^m = (2^n)^{2^n} = 2^{n2^n}$$

The network model is essentially a map

$$\phi : \mathcal{T} \rightarrow \mathcal{D}$$

Obviously this map cannot be onto, because the cardinality of \mathcal{D} is bigger than that of \mathcal{T} . It is natural to ask if this map is injective. A simple example shows that this is not the case.

Example 3.1. Let W_1 be the network of two nodes v_1 and v_2 with a red edge from v_2 to itself and no other edge. Let W_2 be the network of the same nodes v_1 and v_2 with a red edge from v_2 to itself and a red edge from v_1 to v_2 (see Figure 1). One can easily check that $\phi(W_1) = \phi(W_2)$.

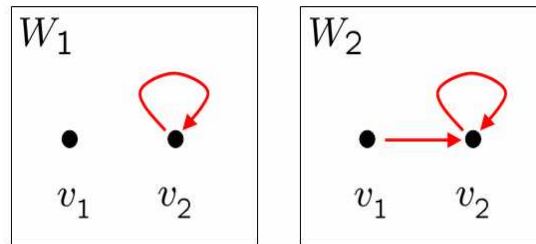


FIGURE 1. Two networks with the same dynamics

Note that, in the above example, the red edge from v_1 to v_2 in W_2 is redundant since v_2 always becomes inactive following any state even without this red edge. This leads to the definition of *dormant vertices* and *redundant edges* in Definition 3.2 and Definition 3.4 below.

Definition 3.2. (a) Let $f : S(V) \rightarrow S(V)$ be a dynamics on the set V . We say a node v is “dormant” if it becomes inactive under f following any state.
 (b) Likewise, given a network W , a node v is “dormant” if it becomes inactive under the dynamics $\phi(W)$ following any state.

Lemma 3.3. *Given a network W , a node v is dormant if and only if there is a red edge from v to itself, and there is no green edge to v .*

Proof. Suppose node v is dormant in W . Let s be the state that is 1 at v and 0 at all other nodes. Since v becomes inactive after s , there must be a red edge from v to itself. Next suppose there is a green edge from node v' to v . Let s' be the state that is 1 at v' and 0 at all other nodes, then v becomes active after s' . This contradicts to the assumption that v is dormant. Therefore there cannot be any green edge to v .

Conversely, suppose there is a red edge from v to itself, and there is no green edge to v . If its current state is inactive, then it remains inactive since there is no

green edge to turn it on. Now suppose its current state is active. If there is a red edge from active node to v , this will turn v into inactive state. If there is no such red edge, then the self-degradation turns v off. Therefore, v is dormant. \square

Definition 3.4. Given a network W , suppose there is a red edge from node v to itself and there is no green edge from any other node to v (by Lemma 3.3, this is equivalent to say that v is dormant). Then any red edge into v is called a “redundant” red edge (or simply redundant edge).

Lemma 3.5. *Adding a redundant edge does not change the dynamics of the network.*

Proof. Let W' be obtained from a network W by adding a redundant red edge from some node to node v . In both networks W and W' , it is easy to check that node v becomes inactive. For any other node, the edges of W and edges of W' to such a node agree completely, and thus their actions agree as well. Therefore, the dynamics of W and W' are in complete agreement. \square

Our theorem below describes $\phi^{-1}(f)$, the set of preimages of a given dynamics f .

Theorem 3.6. *Let $f \in \mathcal{D}$ be a dynamics. If $f \in \text{Im } \phi$, then there exists a unique network W_0 that satisfies*

- (1) $\phi(W_0) = f$, and
- (2) W_0 contains no redundant edge.

Moreover, any other network W satisfying $\phi(W) = f$ can be obtained from W_0 by adding some redundant edges.

Proof. Let $f = \phi(W)$. We delete all possible redundant red edges from W to obtain W_0 . By Lemma 3.5, $\phi(W_0) = \phi(W) = f$. The only thing we need to show is there is a unique such W_0 . We prove this by showing that $r_{ij}(W_0)$ and $g_{ij}(W_0)$ (abbreviated by r_{ij} and g_{ij} respectively) are all determined by f . This will be done by solving r_{ij}, g_{ij} explicitly in term of f .

First we define some specific states. Let p_i be the state that is 1 on node i and 0 on all other nodes. Let p_{ij} be the state that is 1 on nodes i and j , and 0 on all other nodes, where $i \neq j$. Let p_{ijk} be the state that is 1 on nodes i, j and k , and 0 on all other nodes, where i, j, k are mutually distinct nodes.

We now solve for r_{ii} for each i . Consider $f(p_i) = (f_1(p_i), \dots, f_n(p_i))$ Its i^{th} coordinate $f_i(p_i)$ is the activity level of node i after the state p_i . Since node i is the only active node in p_i , $f_i(p_i) = 0$ if and only if there is a self-degradation on node i , i.e. $r_{ii} = 1$. Thus $r_{ii} = \overline{f_i(p_i)}$.

Next, we solve for g_{ij} for each $1 \leq i \neq j \leq n$. consider $f_j(p_i)$, i.e. the activity level of node j after the state p_i . Since node i is the only active node in p_i , $f_j(p_i) = 1$ if and only if there is a green edge from node i to node j , i.e. $g_{ij} = 1$. Therefore $g_{ij} = f_j(p_i)$.

Finally, we show r_{ij} is determined by f for all $i \neq j$. We consider two cases.

Case 1. There is no red edge from node j to itself, i.e. $r_{jj} = 0$.

Consider $f_j(p_{ij})$. This is the activity level of node j after the state p_{ij} . Node j was active in p_{ij} . Thus it becomes inactive if and only if there is a red edge from node i to node j . That is to say, $f_j(p_{ij}) = 0$ if and only if $r_{ij} = 1$. Thus $r_{ij} = \overline{f_j(p_{ij})}$.

Case 2. There is a red edge from node j to itself, i.e. $r_{jj} = 1$. We consider two subcases.

Case 2a. There is a green edge to j , say $g_{kj} = 1$.

Case 2a (i) If $k = i$, then we automatically have $r_{ij} = 0$ since a green edge automatically rules out a red edge from i to j .

Case 2a (ii) If $k \neq i$, we claim $r_{ij} = \overline{f_j(p_{ijk})}$. The proof is as follows.

If $r_{ij} = 0$, the green edge from k to j turns j on, thus $f_j(p_{ijk}) = 1$. If $r_{ij} = 1$, the red edge from i to j turns j off despite of the green edge from k to j , thus $f_j(p_{ijk}) = 0$ (here we use the strong inhibition assumption in the model). Thus $\overline{f_j(p_{ijk})} = 0$. We have proved $f_j(p_{ijk}) = 0$ if and only $r_{ij} = 1$. Therefore $r_{ij} = \overline{f_j(p_{ijk})}$ must hold.

Case 2b. There is no green edge to j . In this case, any red edge to node j is redundant. Since we assume there is no redundant red edge, we have $r_{ij} = 0$.

In all cases, we expressed r_{ii} , g_{ij} and r_{ij} uniquely in terms of f . It follows that W_0 is determined by $\phi(W_0) = f$. \square

Remark 1. (a) In Case 2a (ii) of the above proof, one may wonder if a different choice of k would lead to a different value of r_{ij} . It makes no difference, since $\overline{f_j(p_{ijk})} = r_{ij} = \overline{f_j(p_{ijk'})}$ for any two choices k and k' . This is essentially due to the assumption that the dynamics f comes from a network.

(b) On the other hand, if f is any dynamics that may or may not come from a network, different choices of k may lead to different values of r_{ij} . More precisely, we follow the proof of Theorem 3.6 to obtain solutions $r_{ii} = \overline{f_i(p_i)}$, $g_{ij} = f_j(p_i)$, and $r_{ij} = \overline{f_j(p_{ijk})}$. Solutions r_{ii} and g_{ij} are uniquely determined by f , but solutions for r_{ij} depends on choices of k .

As a consequence of Theorem 3.6, we have

Corollary 1. *Two networks on the same set of nodes yield the same dynamics if and only if they agree after deleting all redundant edges.*

Proof. Let W and W' be two networks on the same set of nodes.

If $\phi(W) = \phi(W')$. We apply Theorem 3.6 to $f = \phi(W) = \phi(W')$ to get W_0 , which is obtained from W , as well as from W' , by deleting redundant edges. Thus W and W' agree after deleting redundant edges. Conversely, if W and W' yields the same network W_0 after deleting all redundant edges, Lemma 3.5 implies $\phi(W) = \phi(W_0) = \phi(W')$. \square

In particular, if we do not allow self degradation, then the map ϕ is injective.

Corollary 2. *Let $\mathcal{T}_0 \subset \mathcal{T}$ be those networks in \mathcal{T} that does not have red edges from a vertex to itself. Then $\phi|_{\mathcal{T}_0} : \mathcal{T}_0 \rightarrow \mathcal{D}$ is injective.*

4. Solving networks for a given dynamics - algorithm and examples. An interesting problem is to determine the network(s) with given (full or partial) information on dynamics. Here we consider the situation in which the full information is given. Using ideas in the previous section, we obtain an algorithm that solves for the network solutions with a given dynamics.

The other situation, in which only partial information is given, is actually biologically more relevant. Often, such partial information is expressed as dynamics on a sequence of states, called a *biological pathway* or *biological process*. The problem of solving network solutions is the so-called *Reverse Engineering Problem*. In a forth coming paper, we will provide a fast algorithm for the Reverse Engineering Problem.

Let f be a given dynamics on the state space $S(V)$. Let p_i, p_{ij} , and p_{ijk} be the states defined in the proof of Theorem 3.6. Our algorithm is described as follows.

Algorithm (Network Solution).

Input. A dynamics $f : S(V) \rightarrow S(V)$.

Output. All networks W with $\phi(W) = f$.

Step 1. Let $r_{ii} = \overline{f_i(p_i)}$ for $i = 1, \dots, n$.

Step 2. Let $g_{ij} = f_j(p_i)$ for $1 \leq i, j \leq n$.

Step 3. For each $i \neq j$, define r_{ij} as follows.

If $r_{jj} = 0$, define $r_{ij} = \overline{f_j(p_{ij})}$.

If $r_{jj} = 1$, and if there is a $k \neq i$ with $g_{kj} = 1$, pick such a k and define

$r_{ij} = \overline{f_j(p_{ijk})}$;

if there is no such k , define $r_{ij} = 0$.

Step 4. Let W be the network defined by r_{ij} and g_{ij} .

Check if $\phi(W) = f$ on every state in $S(V)$.

Step 5. If $\phi(W) \neq f$, the network solution does not exist.

If $\phi(W) = f$, W is a network solution, the one given in Theorem 3.6. All other solutions are obtained by adding some redundant edges to W .

Remark 2. (a) One may ask what the running time for this algorithm is. Let n be the number of nodes. Then the running time for Steps 1-3 is a polynomial function of n , and the running time for Steps 4-5 is an exponential function for n . On the other hand, the input size is often an exponential function of n if we express f explicitly on each of the 2^n states in $S(V)$. Thus the running time is no more than a polynomial of the input size.

(b) Step 3 depends on the choice of k . If f does come from a network, then the choice of k does not matter in the end.

Our algorithm also helps to relate the Strong Inhibition Model, which is logic based, to work in other setting. For example, given a finite system in which the activity level of each node lies in a finite field, then every dynamics is a *polynomial dynamical system*, i.e. a map for which each coordinate map can be expressed as a polynomial map. This viewpoint has led to fruitful research by Laubenbacher and his collaborators (see, e.g. [5] [6]). This is indeed the case for Boolean networks, as the activity levels lie in F_2 , the field with two elements.

The following is such a polynomial dynamical system (Example 1.2 in [6]). We use our algorithm to determine if it comes from a network.

Example 4.1. Let $f : F_2^3 \rightarrow F_2^3$ be given by $f(x_1, x_2, x_3) = (x_2x_3, x_1 + x_3, x_1x_2)$. Figure 2 shows the dynamics.

First, we determine all r_{ii} using $r_{ii} = f_i(p_i)$. We have

$r_{11} = f_1(100) = (010)_1 = 0$, here $(010)_1$ denotes the first coordinate of (010) .

$r_{22} = f_2(010) = (000)_2 = 0$, and

$r_{33} = f_3(001) = (010)_3 = 0$.

Thus, none of the three nodes admits a red edge to itself.

Next, we determine g_{ij} using $g_{ij} = f_j(p_i)$. We have

$f(p_1) = f(100) = (010)$ which implies $g_{12} = 1, g_{13} = 0$.

$f(p_2) = f(010) = (000)$ which implies $g_{21} = 0, g_{23} = 0$.

$f(p_3) = f(001) = (010)$ which implies $g_{31} = 0, g_{32} = 1$.

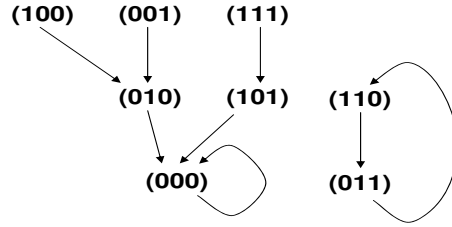


FIGURE 2. An example of dynamics.

Finally, for r_{ij} , we apply the formula $r_{ij} = \overline{f_j(p_{ij})}$ (note that all $r_{jj} = 0$). We have

$$\begin{aligned} r_{12} &= \overline{f_2(p_{12})} = \overline{f_2(110)} = \overline{(011)_2} = \overline{1} = 0 \\ r_{13} &= \overline{f_3(p_{13})} = \overline{f_3(101)} = \overline{(000)_3} = \overline{0} = 1 \\ r_{21} &= \overline{f_1(p_{21})} = \overline{f_1(110)} = \overline{(011)_1} = \overline{0} = 1 \\ r_{23} &= \overline{f_3(p_{23})} = \overline{f_3(011)} = \overline{(110)_3} = \overline{0} = 1 \\ r_{31} &= \overline{f_1(p_{31})} = \overline{f_1(101)} = \overline{(000)_1} = \overline{0} = 1 \\ r_{32} &= \overline{f_2(p_{32})} = \overline{f_2(011)} = \overline{(110)_2} = \overline{1} = 0 \end{aligned}$$

This determines a network W_0 which is the only possible network whose dynamics might agree with f . A straight forward computation shows, however, that $\phi(W_0)(001) = (011)$. This does not agree with f since $f(001) = (010)$. It follows that there is no network whose dynamics is equal to f .

Example 4.2. As another example, consider $f : F_2^3 \rightarrow F_2^3$ defined by $f(x_1, x_2, x_3) = (0, x_2 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3, x_1 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3)$.

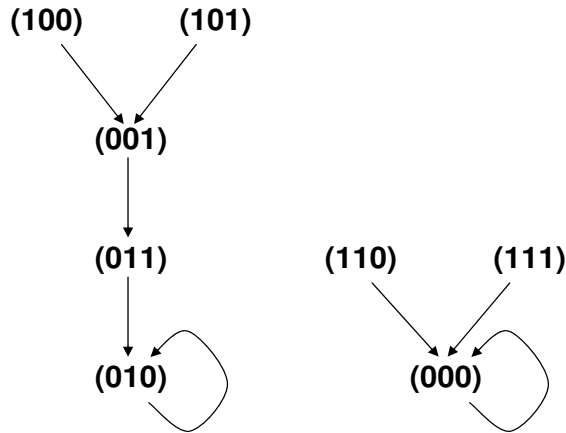


FIGURE 3. An example of dynamics.

Figure 3 shows the dynamics. We have

$$\begin{aligned} f(110) &= f(111) = f(000) = (000), \\ f(101) &= f(100) = (001), f(001) = (011), f(011) = f(010) = (010). \end{aligned}$$

A calculation as before shows the following results, shown in Figure 4. We have $r_{11} = 1; r_{12} = r_{23} = g_{13} = g_{32} = 1$, shown in solid red and green edges (for easier visualization, the green edges are marked with diamond tails); r_{21} and r_{31} are the redundant edges, shown in dotted red edges; and all other edges must not be present. Thus there are four network solutions.

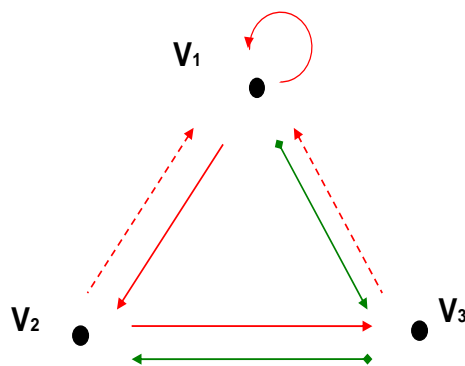


FIGURE 4. Network solution for Example 4.3.

5. Subspaces of network topology and network dynamics. In this section, we study the extent of non-injectivity and non-surjectivity of the map ϕ . First, we determine precisely when a network is uniquely determined by its dynamics (Theorem 5.1 (1)). This is expressed nicely in terms of dormant vertices. Then, we move our attention to the dynamics side, focusing on dynamics that have network solutions. We show that the set of such dynamics is in one-to-one correspondence with a subset of networks described in terms of redundant edges (Theorem 5.1 (2)).

The results in Theorem 5.1 allow us to give a precise counting for the number of networks that are determined by their dynamics, as well as the number of dynamics that have network solutions (Theorem 5.2).

Theorem 5.1. (1). A network W is uniquely determined by its dynamics if and only if it has no dormant vertices. (2). There is a one-to-one correspondence between the set of dynamics that have network solutions and the set of networks that have no redundant edges.

Proof. To prove (1), note that W is uniquely determined by $\phi(W)$ if and only one cannot add a redundant edge to W or delete a redundant edge to W , and the later is equivalent to the non-existence of dormant vertices. For (2), the map sending W_0 to $\phi(W_0)$ is a one-to-one correspondence by Theorem 3.6. \square

Now, recall that the number of interaction topologies for a node is $2 \times 3^{n-1}$ where 2 comes from the two choices of either a self-degradation red edge or no edge to the node itself and 3^{n-1} comes from the three choices of a green edge, a red edge, or no edge to the node from each of the other $(n - 1)$ nodes. Since for a node to be dormant, it must contain a self-degradation red edge to itself (i.e., one choice) whereas each of the other $(n - 1)$ nodes must not have a green edge to it (i.e., two choices of either a red edge or no edge), the number of interaction topologies for the node to be *dormant* is $1 \times 2^{n-1} = 2^{n-1}$. Therefore, the number

of interaction topologies for the node to be *non-dormant* is $(2 \times 3^{n-1} - 2^{n-1})$. Moreover, the number of interaction topologies for a node to contain no *redundant edge* is $(2 \times 3^{n-1} - 2^{n-1} + 1)$ where the last term of value 1 indicates the unique topology in which the node is dormant (i.e. the one with one red edge to itself and no other edge to the node). We thus arrive at the following conclusions about the network topology space \mathcal{T} and network dynamics space \mathcal{D} .

Theorem 5.2. (1) Let $\mathcal{T}_{\text{unique}}$ be the set of those networks W such that W is uniquely determined by $\phi(W)$ (i.e., W contains no dormant node), then

$$|\mathcal{T}_{\text{unique}}| = (2 \times 3^{n-1} - 2^{n-1})^n$$

(2) Let $\mathcal{D}_{\text{feasible}}$ be the set of those dynamics f so that $\phi(W) = f$ has a solution. That is $\mathcal{D}_{\text{feasible}} = \text{Im } \phi$. Then

$$|\mathcal{D}_{\text{feasible}}| = (2 \times 3^{n-1} - 2^{n-1} + 1)^n$$

Proof. The proof follows directly from Theorem 5.1 and the above discussion. \square

Recall that

$$|\mathcal{T}| = 3^{n(n-1)}2^n \text{ and } |\mathcal{D}| = 2^{n2^n}$$

Therefore, Theorem 5.2 (1) shows that almost all networks are uniquely determined by their dynamics:

$$\frac{|\mathcal{T}_{\text{unique}}|}{|\mathcal{T}|} = \frac{(2 \times 3^{n-1} - 2^{n-1})^n}{(2 \times 3^{n-1})^n} = \left[1 - \frac{1}{2} \left(\frac{2}{3}\right)^{n-1}\right]^n \rightarrow 1 \text{ as } n \rightarrow \infty$$

Theorem 5.2 (2) provides the percentage of the dynamics that are feasible:

$$\frac{|\mathcal{D}_{\text{feasible}}|}{|\mathcal{D}|} = \frac{(2 \times 3^{n-1} - 2^{n-1} + 1)^n}{2^{n2^n}}$$

Not surprisingly, this percentage approaches 0 as $n \rightarrow \infty$. Of course, this also follows from the fact that $|\mathcal{T}| \ll |\mathcal{D}|$.

It will be interesting to see if our techniques could be applied to the situation when only partial information on dynamics is given. This will be investigated in a future project.

Acknowledgments. The project is partially supported by the NSF CDI grant CMMI-0941228, NSF individual grants DMS-0513918 (YR), DMR-0313129 (CZ), and the Columbian College Facilitating Fund at The George Washington University (YR and CZ). Our collaboration also benefited from the George Washington University Mathematical Application Seminar, which was supported by the GWU University Seminar Fund.

REFERENCES

- [1] F. Li, T. Long, Y. Lu, Q. Ouyang and C. Tang, *The yeast cell-cycle network is robustly designed*, Proc. Natl. Acad. Sci. U.S.A., **101** (2004), 4781–4786.
- [2] N. Tan and Q. Ouyang, *Design of a network with state stability*, J. Theor. Biol., **240** (2006), 592–598.
- [3] J. Hopfield, *Neural networks and physical systems with emergent collective computational properties*, Proc. National Academy of Sciences of the USA, **79** (1982), 2554–2558.

- [4] G. Wang, C. Du, H. Chen, R. Simha, Y. Rong, Y. Xiao, and C. Zeng, *Process-Based Network Decomposition Reveals Backbone Motif Structure*, Proc. National Academy of Sciences, **107** (2010), 10478–10483.
- [5] R. Laubenbacher and B. Stigler, *A computational algebra approach to the reverse engineering of gene regulatory networks*, Journal of Theoretical Biology, **229** (2004), 523–537.
- [6] A. Salam Jarrah, R. Laubenbacher and A. Veliz-Cuba, *The dynamics of conjunctive and disjunctive Boolean networks*, preprint (2008). [arXiv:0805.0275](https://arxiv.org/abs/0805.0275).

Received May 2009; revised October 2009.

E-mail address: rong@gwu.edu

E-mail address: chenz@gwu.edu

E-mail address: evans.christina.s@gmail.com

E-mail address: lazychen@gmail.com

E-mail address: gwang@gwu.edu