Automatica 125 (2021) 109453

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Quantifying the total effect of edge interventions in discrete multistate networks☆

David Murrugarra [a,*], Elena Dimitrova [b]

[a] Department of Mathematics, University of Kentucky, Lexington, KY 40506-0027, USA
[b] Department of Mathematics, California Polytechnic State University, San Luis Obispo, CA 93407-0403, USA

ARTICLE INFO

Article history:
Received 3 December 2019
Received in revised form 12 October 2020
Accepted 8 December 2020
Available online 11 January 2021

Keywords:
Edge control
Boolean networks
Canalization
Total effect

ABSTRACT

Developing efficient computational methods to assess the impact of external interventions on the dynamics of a network model is an important problem in systems biology. This paper focuses on quantifying the global changes that result from the application of an intervention to produce a desired effect, which we define as the *total effect* of the intervention. The type of mathematical models that we will consider are discrete dynamical systems which include the widely used Boolean networks and their generalizations. The potential interventions can be represented by a set of nodes and edges that can be manipulated to produce a desired effect on the system. We use a class of regulatory rules called nested canalizing functions that frequently appear in published models and were inspired by the concept of canalization in evolutionary biology. In this paper, we provide a polynomial normal form based on the canalizing properties of regulatory functions. Using this polynomial normal form, we give a set of formulas for counting the maximum number of transitions that will change in the state space upon an edge deletion in the wiring diagram. These formulas rely on the canalizing structure of the target function since the number of changed transitions depends on the canalizing layer that includes the input to be deleted. We also present computations on random networks to compare the exact number of changes with the upper bounds provided by our formulas. Finally, we provide statistics on the sharpness of these upper bounds in random networks.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Boolean networks (BN) have been proposed as an appropriate framework for modeling the state of cells due to their simplicity and the variety of tools available for model analysis (Veliz-Cuba, Jarrah, & Laubenbacher, 2010; Zañudo & Albert, 2015). However, some complex gene interactions cannot be represented in the Boolean setting and several generalizations of the Boolean approach have been developed (Thomas & D'Ari, 1990). Multistate models, a generalization of the BN framework, where the genes can attain more than two states have been proposed as appropriate models for capturing complex gene expression patterns, such as consideration of three states (low, medium, and high). We note that while in theory it is possible to develop models where the variables can take on any number of possible states (possibly only restricted by the requirement that it needs to be a power of a prime number so that the domain can have the structure of a finite field), in practical applications this number is typically small — rarely larger than 5.

A Gene Regulatory Network (GRN) is a representation of the intricate relationships among genes, proteins, and other compounds that are responsible for the expression levels of mRNA and proteins. Boolean networks have been successfully used to model and study the properties of GRN (Albert & Othmer, 2003; Li, Long, Lu, Ouyang, & Tang, 2004). In particular, Boolean canalizing rules were introduced by S. Kauffman and collaborators (Kauffman, Peterson, Samuelsson, & Troein, 2003, 2004) and reflect the concept of canalization in evolutionary biology that Waddington pioneered in 1942 (Waddington, 1957) – that organisms evolve developmental robustness, producing an invariant phenotype even under genetic or environmental perturbations.

In this article, we study the network-wide effect of an experimental intervention that either prevents a regulation from happening or silences a node. Such intervention is modeled through edge deletion and can be achieved via therapeutic drugs that target a specific gene interaction (Campbell & Albert, 2019; Choi, Shi, Jung, Chen, & Cho, 2012). In Murrugarra and Dimitrova (2015) we introduced methods for quantifying side effects in

☆ ED was partially supported by NSF, USA Award #1937717. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Hernan Haimovich under the direction of Editor Sophie Tarbouriech.

* Corresponding author.
E-mail addresses: murrugarra@uky.edu (D. Murrugarra), edimitro@calpoly.edu (E. Dimitrova).

https://doi.org/10.1016/j.automatica.2020.109453
0005-1098/© 2020 Elsevier Ltd. All rights reserved.

Boolean networks. However, many of the more recently published discrete dynamical models include variables that take on more than two states due to the need for capturing mechanisms that are not binary in nature (Chifman et al., 2017; Espinosa-Soto, Padilla-Longoria, & Alvarez-Buylla, 2004; Remy et al., 2015; Thieffry & Thomas, 1995; Zañudo, Scaltriti and Albert, 2017). Consequently, Boolean nested and partially nested canalizing functions were generalized to multistate (Kadelka, Li, Kuipers, Adeyeye, & Laubenbacher, 2017; Murrugarra & Laubenbacher, 2011, 2012) which enables the possibility of capturing more complex interactions among the genes in the network. Such functions can be viewed as a discrete dynamical system with a stratified structure which consists of hierarchical layers of variables according to their relative influence over the system dynamics.

The ability to quantify the global changes in the dynamics of the network after an external perturbation has important applications. In the presence of external network modifications where the topology of the network changes but the attractor structure remains unchanged, it is still desirable to quantify the changes in other aspects of the dynamics such as the transient time. For instance, in evolutionary biology to simulate evolution one often evolves an ensemble of networks (by performing mutations, crossover, selection, etc.) for many generations (Wagner, 1996). At the end of the simulations, one measures the changes in the evolved networks to compare with the features of the original ensemble. In Wagner (1996), after simulated evolution, the evolved networks had similar features to the original ones such as the number of attractors and basin sizes. One feature that had changed is the transient time (Wagner, 1996). The theoretical tools presented in this paper will be useful to measure global changes even if the attractor structure is preserved after an intervention.

There are several published control methods for Boolean networks such as Stable Motifs (Zañudo & Albert, 2015), Feedback Vertex Sets (Zañudo, Yang and Albert, 2017), Minimal Hitting Sets (Klamt, Saez-Rodriguez, Lindquist, Simeoni, & Gilles, 2006; Vera-Licona, Bonnet, Barillot, & Zinovyev, 2013), and several others (Gates & Rocha, 2016; Li, Yang, & Chu, 2015; Poret & Boissel, 2014; Qiu, Tamura, Ching, & Akutsu, 2014; Sordo Vieira, Laubenbacher, & Murrugarra, 2019; Zañudo, Scaltriti et al., 2017). While these control methods focus on finding control interventions that satisfy a control objective (e.g., to drive the system into a specific attractor), there are very few studies of the total extent of the consequences of applying a certain control action (beyond satisfying the control objective). This paper contributes methods for measuring the impact of the control actions on the dynamics of multistate networks. The type of theoretical tools presented here can help to discriminate control actions with minimal effect on the state space. That is, even if we have different control candidates that can achieve a certain objective, they might have different impact on the dynamics of the network and we might be interested in distinguishing the control action that produces the least changes in the dynamics of the network.

The rest of the paper is structured as follows. In Section 2, we introduce discrete dynamical systems and their representation as polynomial dynamical systems. In Section 3 we define the control actions for multistate networks. In Section 4 we provide a polynomial normal form for discrete functions and then we use this representation to derive a set of formulas for counting the maximum number of transitions in the state space upon edge deletions. In Section 5 we present computational results for random networks. Finally, in Section 6, we provide the conclusions of the paper.

2. Background

A discrete dynamical system can be defined as a dynamical system that is discrete in time as well as in variable states. More formally, consider a collection $x_1, \ldots, x_n$ of variables, each of which can take on values in finite sets $X_1, \ldots, X_n$. Let $X = X_1 \times \cdots \times X_n$ be their Cartesian product. A discrete dynamical system in the variables $x_1, \ldots, x_n$ is a function

$$\mathbf{F} = (f_1, \ldots, f_n) : X \to X$$

where each coordinate function f_i is a discrete function on a subset of $\{x_1, \ldots, x_n\}$ which represents how the future value of the ith variable depends on the present values of the variables. If $X_i = \{0, 1\}$, then each f_i is a Boolean rule and $\mathbf{F}$ is a Boolean network.

In this article, for the purpose of exploiting the algebraic properties of discrete functions, it is assumed that the variables $x_1, \ldots, x_n$ take on values from a finite field $\mathbb{F}$. Then using the fact that any discrete function $f_i : \mathbb{F}^n \to \mathbb{F}$ can be represented as a polynomial in $x_1, \ldots, x_n$, that is $f_i \in \mathbb{F}[x_1, \ldots, x_n]$, the discrete network can be represented as $\mathbf{F} = (f_1, \ldots, f_n) : \mathbb{F}^n \to \mathbb{F}^n$ where each $f_i \in \mathbb{F}[x_1, \ldots, x_n]$. If any of the variables $x_1, \ldots, x_n$ take on values from a set that cannot be directly identified with a finite field, then it is straightforward to embed the system $\mathbf{F} : X \to X$ into a system $\hat{\mathbf{F}} : \mathbb{F}^n \to \mathbb{F}^n$, where $X \subset \mathbb{F}^n$, while preserving the attractor structure of $\mathbf{F}$; see Veliz-Cuba et al. (2010).

Given a discrete network $\mathbf{F} = (f_1, \ldots, f_n)$, a directed graph $\mathcal{W}$ on n nodes $x_1, \ldots, x_n$ is associated to $\mathbf{F}$ as follows: there is a directed edge in $\mathcal{W}$ from x_j to x_i if x_j appears in f_i, i.e. x_j is in the *support* of f_i, written $x_j \in supp(f_i)$. In the context of a molecular network model, this graph represents the wiring diagram of the network.

The dynamics of a discrete network is given by the difference equation $x(t+1) = \mathbf{F}(x(t))$; that is, the dynamics is generated by iteration of $\mathbf{F}$. More precisely, the dynamics of $\mathbf{F}$ is represented by the state space graph S, defined as the graph with vertices in $\mathbb{F}^n$ which has an edge from $x \in \mathbb{F}^n$ to $y \in \mathbb{F}^n$ if and only if $y = \mathbf{F}(x)$. In this context, the problem of finding the states $x \in \mathbb{F}^n$ where the system will get stabilized is of particular importance. The collection of these special points of the state space are called *attractors* of a discrete network and elements of the attractors may include steady states (fixed points), where $\mathbf{F}(x) = x$, or cycles, where $\mathbf{F}^r(x) = x$ for some integer $r > 1$. Attractors in network modeling might represent cell types (Kauffman, 1969) or cellular states such as apoptosis, proliferation, or cell senescence (Huang, 1999; Shmulevich & Dougherty, 2010).

3. Methods

Network interventions can be modeled through edge and node manipulations and can be achieved via therapeutic drugs that target a specific gene interaction (Campbell & Albert, 2019; Choi et al., 2012). In Murrugarra and Dimitrova (2015) and Murrugarra, Veliz-Cuba, Aguilar, and Laubenbacher (2016) we provided definitions for these actions in Boolean networks. These definitions are usually used for encoding the control parameters with the purpose of identifying control targets as shown in Murrugarra et al. (2016). In this paper we will consider the deletion and constant expression of edges in the multistate setting.

Throughout the paper, $S_{i,j}$ will be a subset of $\mathbb{F}$ and $Q_{i,j}(u)$ will be the indicator functions of $S_{i,j}$. That is, they return 1 when u is in the set and 0 when u is not. The index i indicates the node x_i from which the edge begins and the second index j is used when necessary to mark the function under consideration.

3.1. Edge control in multistate networks

In the Boolean setting, the deletion of an edge was implemented by setting an input to zero so that the interaction of that input (represented by an edge) was being silenced. For the multistate case, the silencing of the interaction will be applied whenever the control variable is within a range of values of the possible discrete values.

Definition 1 (*Edge Deletion*)**.** Consider the edge $x_i \to x_j$ in a wiring diagram. For $u \in S_{i,j}$, the control of the edge $x_i \to x_j$ consists of manipulating the input variable x_i for f_j in the following way:

$$\mathcal{F}_j(x, u) = f_j(x_{j_1}, \ldots, (1 - Q_{i,j}(u))x_i, \ldots, x_{j_m}).$$

For each value of $u \in \mathbb{F}$ we have the following control settings:

- For $u \in S_{i,j}$,

  $$\mathcal{F}_j(x, u) = f_j(x_{j_1}, \ldots, x_i = 0, \ldots, x_{j_m}).$$

  That is, the control is active and the action represents the removal of the edge $x_i \to x_j$.
- For $u \notin S_{i,j}$,

  $$\mathcal{F}_j(x, u) = f_j(x_{j_1}, \ldots, x_i, \ldots, x_{j_m}).$$

  That is, the control is not active.

We will also consider the constant expression of edges, which we define as follows.

Definition 2 (*Constant Expression*)**.** Consider the edge $x_i \to x_j$ in a wiring diagram and $a \in \mathbb{F}$. For $u \in S_{i,j}$, the control of the edge $x_i \to x_j$ consists of manipulating the input variable x_i for f_j in the following way:

$$\mathcal{F}_j(x, u) = f_j(x_{j_1}, \ldots, (1 - Q_{i,j}(u))x_i + aQ_{i,j}(u), \ldots, x_{j_m}).$$

For each value of $u \in \mathbb{F}$ we have the following settings:

- For $u \in S_{i,j}$,

  $$\mathcal{F}_j(x, u) = f_j(x_{j_1}, \ldots, x_i = a, \ldots, x_{j_m}).$$

  That is, the control is active and the action represents the constant expression (to a) of the edge $x_i \to x_j$.
- For $u \notin S_{i,j}$,

  $$\mathcal{F}_j(x, u) = f_j(x_{j_1}, \ldots, x_i, \ldots, x_{j_m}).$$

  That is, the control is not active.

Remark 3. Node control of x_i can be defined as setting the function f_i to a constant $a \in \mathbb{F}$.

4. Results

In this section we present a definition of k-canalizing functions for the multistate case and then we characterize this functions in terms of layers of canalizations. Subsequently, we use this canalizing layers representation to derive an upper bound for the number of changes in the state space of a discrete system upon an edge deletion in the wiring diagram.

4.1. Multistate k-canalizing functions

In the following definition, we assume that σ is a permutation on $\{1, \ldots, n\}$.

Definition 4. The function $f : \mathbb{F}^n \to \mathbb{F}$ is a *k-canalizing function* in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(k)}$ with *canalizing input sets* $S_1, \ldots, S_k \subset \mathbb{F}$ and *canalizing output values* $b_1, \ldots, b_k \in \mathbb{F}$ if it can be represented in the form

$$f(x_1, \ldots, x_n) = \begin{cases} b_1, & \text{if } x_{\sigma(1)} \in S_1, \\ b_2, & \text{if } x_{\sigma(1)} \notin S_1, x_{\sigma(2)} \in S_2, \\ \vdots & \\ b_k, & \text{if } x_{\sigma(1)} \notin S_1, \ldots, x_{\sigma(k)} \in S_k, \\ g \neq b_k, & \text{if } x_{\sigma(1)} \notin S_1, \ldots, x_{\sigma(k)} \notin S_k, \end{cases} \tag{1}$$

where $g = g(x_{\sigma(k+1)}, \ldots, x_{\sigma(n)})$ is a multistate function on $n - k$ variables. When g is not a canalizing function, the integer k is the canalizing *depth* of f. If g is not a constant function, then g is called the *core function* of f and is denoted by P_C.

Remark 5. Note that in Definition 4 we require that the function g be unique when all the canalizing variables are not in their corresponding canalizing input sets. As a result, a function could be canalizing but not 1-canalizing, see Example 6.

Example 6. Let $\mathbb{F} = \{0, 1, 2\}$ and $n = 2$. Consider the function

$$f(x_1, x_2) = 1 + 2x_1^2 + 2x_2 + 2x_1^2x_2 + 2x_2^2.$$

For this function x_2 is canalizing (with $S_1 = \{2\}$) because $f(x_1, 2) = 1$. However, f is not a 1-canalizing function because $f(x_1, 0) = 1 + 2x_1^2 \neq 2 + x_1^2 = f(x_1, 1)$. Thus, even though x_2 is canalizing for f, the function f has no layers of canalization. Thus, $P_C = f$.

4.2. Layers of canalization in multistate networks

In Theorem 7 we provide a polynomial normal description of discrete functions. Basically, this theorem gives a partition of the inputs of the function into canalizing and non-canalizing variables and, within the canalizing ones, we categorize the input variables into layers of canalization. This theorem is a generalization of a theorem in He and Macauley (2016) from Boolean to the multistate case.

Let $S \subset \mathbb{F}$ be a subset of $\mathbb{F}$ and $\widetilde{Q}_S(u)$ be the indicator function of the complement of S. That is,

$$\widetilde{Q}_S(x) = \begin{cases} 1 & \text{if } x \notin S, \\ 0 & \text{if } x \in S. \end{cases}$$

Theorem 7. *Every multistate function can be uniquely written as*

$$f(x_1, \ldots, x_n) = M_1(M_2(\ldots(M_{r-1}(M_rP_C + B_r) + B_{r-1})\ldots) + B_2) + B_1, \tag{2}$$

where $M_i = \prod_{j=1}^{k_i} \widetilde{Q}_{S_{i,j}}$, $d = k_1 + \cdots + k_r$ *is the canalizing depth,* P_C *is a polynomial that has no canalizing variables,* $B_1, B_2, \ldots, B_r \in \mathbb{F}$*, and* $B_r \neq 0$*. Each variable* x_i *appears in exactly one of the* $M_1, M_2, \ldots, M_r, P_C$*.*

Proof. If $f(x_1, \ldots, x_n)$ is non-canalizing, then $P_C = f$. If $f(x_1, \ldots, x_n)$ is canalizing, then we proceed by induction. For $n = 1$, if f is canalizing in x_i but not 1-canalizing in x_i, then we set $P_C = f$. If f is 1-canalizing in x_i, then it can be written as $f = \widetilde{Q}_{S_1}(x_i) + B_1$ for some set $S_1 \subset \mathbb{F}$. Then f has the form of Eq. (2) by setting $M_1 = \widetilde{Q}_{S_1}(x_i)$ and $P_C = 1$. For $n = 2$, if $f(x_i, x_j)$ is not 1-canalizing on any of its variables, then we set $P_C = f$. If f is 1-canalizing on x_i, then f can be written as $f(x_i, x_j) = M_1(x_i)g(x_j) + B_1$ for some $g(x_j)$. Then f has the form of Eq. (2) by setting $P_C = g$.

Now assume that Eq. (2) is true for any canalizing function that is essential in at most $n-1$ variables (that is, for all functions that depend in at most $n-1$ variables). Let f be a function that is essential in n variables. If f is not 1-canalizing on any of its variables, then we set $P_C = f$. If f is 1-canalizing in $x_{i_1}, \ldots, x_{i_{k_1}}$, then $f = M_1 g + B_1$, where M_1 is the product of indicator functions of the complements of sets $S_{i_1}, \ldots, S_{i_{k_1}} \subset \mathbb{F}$ and g has $n-k_1$ variables. If g has no canalizing variables, then f has the form of Eq. (2) with $P_C = g$. If g is canalizing, then by the inductive hypothesis g can be written as

$$g = M_2(\ldots(M_{r-1}(M_r P_C + B_r) + B_{r-1})\ldots) + B_2.$$

Thus, f has the form of Eq. (2).

Remark 8. For a multistate nested canalizing function, the formula in Eq. (2) reduces to

$$f(x_1, \ldots, x_n) = M_1(M_2(\ldots(M_{r-1}(B_{r+1}M_r + B_r) + B_{r-1})\ldots) + B_2) + B_1, \tag{3}$$

as was shown in Kadelka et al. (2017).

In the following example we describe a 2-canalizing function with noncanalizing variables.

Example 9. Let $\mathbb{F} = \{0, 1, 2\}$ and $n = 4$. Consider the function

$$f(x_1, x_2, x_3, x_4) = 1 + x_1^2 + x_1^2x_2 + 2x_1^2x_2^2 + x_1^2x_2x_3 + 2x_1^2x_2^2x_3 + x_1^2x_2x_4 + 2x_1^2x_2^2x_4.$$

The function f can be written as in Eq. (2) as

$$f = (M_1(M_2(P_C + 1) + 1) + 1),$$

where $M_1 = \widetilde{Q}_{S_1}(x_1) = x_1^2$, $S_1 = \{0\}$, $M_2 = \widetilde{Q}_{S_2}(x_2) = x_2 + 2x_2^2$, $S_2 = \{0, 1\}$, and $P_C = x_3 + x_4$. Thus f has two layers and two noncanalizing variables. Note that f can also be written as in Eq. (1) as

$$f(x_1, x_2, x_3, x_4) = \begin{cases} 1, & \text{if } x_1 \in S_1 = \{0\}, \\ 2, & \text{if } x_1 \notin S_1, x_2 \in S_2 = \{0, 1\}, \\ P_C, & \text{if } x_1 \notin S_1, x_2 \notin S_2. \end{cases}$$

4.3. Upper bounds

Using the polynomial normal form of multistate functions in Theorem 7, we derive a set of formulas for counting the maximum number of transitions that will change in the state space upon an edge deletion in the wiring diagram. The formulas presented here are generalizations from the Boolean case to the multistate setting of the formulas we presented in Murrugarra and Dimitrova (2015).

For the next theorem, we are going to assume that the functions of the discrete network $\mathbf{F} = (f_1, \ldots, f_n) : \mathbb{F}^n \to \mathbb{F}^n$ are written in the format of Theorem 7. That is, for $t = 1, \ldots, n$ the coordinate function f_t has the following form,

$$f_t(x_1, \ldots, x_n) = M_1^t(M_2^t(\ldots(M_{r-1}^t(M_r^t P_C + B_r) + B_{r-1})\ldots) + B_2) + B_1, \tag{4}$$

where $M_i^t = \prod_{j=1}^{k_i} \widetilde{Q}_{S_{j,t}}$, $d = k_1 + \cdots + k_r$ is the canalizing depth, P_C is a polynomial with no canalizing variables, $B_1, B_2, \ldots, B_r \in \mathbb{F}$, and $B_r \neq 0$. Each variable x_i appears in exactly one of $M_1^t, M_2^t, \ldots, M_r^t, P_C$.

Remark 10. Note that the function f_t has r layers and there are k_i variables in each layer for $i = 1, \ldots, r$.

In the following theorem, we assume that the canalizing input sets are all of the same size for all the variables. In Theorem 13 we study the general case where the canalizing input sets of the variables can be different.

Theorem 11. *Let $\mathbf{F} = (f_1, \ldots, f_n) : \mathbb{F}^n \to \mathbb{F}^n$ be a multistate network where f_t is a k-canalizing function written as in Eq. (4) with $k_1, \ldots, k_r$ the numbers of variables in layers $1, \ldots, r$, respectively. Let x_s be in the ℓth layer, where $\ell \leq r$ and r is the number of layers. Suppose that all canalizing variables have the same canalizing input set S and that $0 \in S$. Then, the maximum number of transitions in the state space that will change upon deletion of $x_s \to x_t$ is given by*

$$p^{n-k_1-\cdots-k_\ell}\,(p - |S|)^{k_1+\cdots+k_\ell}. \tag{5}$$

Proof. Let $m = k_1 + \cdots + k_\ell$. The number of input vectors where the other canalizing variables (not x_s) of f_t do not take on their canalizing input is $(p - |S|)^{m-1}$. For these input vectors, if x_s was already set to 0 or to any other of its canalizing values in S, then the output of f_t will not change as a result of deleting $x_s \to x_t$. Finally, since we have $n - m$ non-canalizing variables, the total number of input vectors for which the output of f_t can possibly change is $(p - |S|)^{m-1}(p - |S|)p^{n-m} = (p - |S|)^m p^{n-m}$.

Remark 12. Note that from Eq. (5) that the number of variables in each layer affects the number of changes and that there are potentially more changes when the deletion happens in a more dominant layer, see Examples 17–18.

Theorem 13. *Let $\mathbf{F} = (f_1, \ldots, f_n) : \mathbb{F}^n \to \mathbb{F}^n$ be a multistate network where f_t is a k-canalizing function written as in Eq. (4) with $k_1, \ldots, k_r$ the numbers of variables in layers $1, \ldots, r$, respectively. Let x_s be in the ℓth layer, $\ell \leq r$ and r is the number of layers. The maximum number of transitions in the state space that will change upon deletion of $x_s \to x_t$ is given by*

$$p^{n-k_1-\cdots-k_\ell} \cdot \left(\prod_{i=1}^{\ell-1}\prod_{j=1}^{k_i}(p - |S_{j,t}|)\right) \left(\prod_{\substack{j=1 \\ j\neq s}}^{k_\ell}(p - |S_{j,t}|)\right)(p - R), \tag{6}$$

where

$$R = \begin{cases} |S_{s,t}| & \text{if } 0 \in S_{s,t} \\ p - |S_{s,t}| & \text{if } 0 \notin S_{s,t}. \end{cases}$$

Proof. The strategy is to first count the number of inputs that do not contain values from the canalizing sets of the variables in the first $\ell - 1$ layers (that do not contain x_s). Thus, the term in the first line of Eq. (7) counts the number of non-canalizing inputs in the previous layers to the layer containing x_s; the term inside the second set of parentheses of Eq. (7) counts the number of non-canalizing inputs of the variables (except of x_s) in the layer containing x_s; the last term in Eq. (7) counts the number of non-canalizing inputs of x_s. For the last term, notice that deleting $x_s \to x_t$ results in setting $x_s = 0$ in f_t. If 0 is in the canalizing set of x_s, $S_{s,t}$, then the rest of the values in $S_{s,t}$ will yield the same output as 0. Since $|S_{s,t}|/p$ of the input values in the transition table of f_t contain a canalizing value for x_s, it is the remaining $\frac{p-|S_{s,t}|}{p}$ of the table that can potentially change as a result of the edge deletion. On the other hand, if $0 \notin S_{s,t}$, then it is the inputs not in $S_{s,t}$ that have the potential to change the output as a result of deleting $x_s \to x_t$ which constitutes $1/p$ of the transition table,

with $\frac{p-1}{p}$ of the table that can potentially change as a result of the edge deletion. Thus, to obtain Eq. (6) we multiply the following expressions:

$$\begin{aligned}&\frac{p^n}{p^{k_1+\cdots+k_{\ell-1}}}\cdot\left(\prod_{i=1}^{\ell-1}\prod_{j=1}^{k_i}(p-|S_{j,t}|)\right)\\&\left(\frac{1}{p^{k_\ell-1}}\prod_{\substack{j=1\\j\neq s}}^{k_\ell}(p-|S_{j,t}|)\right)\frac{1}{p}(p-R)=\\&p^{n-k_1-\cdots-k_\ell}\cdot\left(\prod_{i=1}^{\ell-1}\prod_{j=1}^{k_i}(p-|S_{j,t}|)\right)\\&\left(\prod_{\substack{j=1\\j\neq s}}^{k_\ell}(p-|S_{j,t}|)\right)(p-R),\end{aligned}\tag{7}$$

Remark 14.

(1) The bound in Eq. (6) is sharp, see Examples 17–18.
(2) When $p = 2$, the formula in Eq. (6) reduces to $2^{n-k_1-k_2-\cdots-k_r}$.
(3) If instead of edge deletion, we consider constant expression to $a \in \mathbb{F}$ (see Definition 2) of $x_s \to x_t$, then the formula in Eq. (6) remains the same except for R which becomes

$$R = \begin{cases} |S_{s,t}| & \text{if } a \in S_{s,t} \\ p - |S_{s,t}| & \text{if } a \notin S_{s,t}. \end{cases}$$

Proposition 15. *Let $\mathbf{F} = (f_1, \ldots, f_n) : \mathbb{F}^n \to \mathbb{F}^n$ be a multistate network where f_t is written as in Eq. (4). Let $x_s \in supp(P_C)$. The maximum number of transitions in the state space that will change upon deletion of $x_s \to x_t$ is*

$$p^{n-k_1-\cdots-k_r-1}\left(\prod_{i=1}^{r}\prod_{j=1}^{k_i}(p-|S_{j,t}|)\right)(p-1).\tag{8}$$

Remark 16.

(1) This upper bound is sharp.
(2) When $p = 2$, the expression reduces to 2^{n-d-1}, where d is the canalizing depth.
(3) If f has no canalizing variables, then Eq. (8) reduces to $p^{n-1}(p-1)$.

5. Applications

To provide further insights into the results presented above and to illustrate the use of the formulas here we present numerical results for random networks where we compare the exact number of changes to the upper bounds provided by the formulas.

For the next examples, we generated random networks with scale-free structure using the Barabasi–Albert algorithm (Barabasi & Albert, 1999). We note that the Barabasi–Albert algorithm produces undirected edges but for our examples we need directed edges. Thus, for each undirected edge between x_i and x_j, we converted the edge $x_i - x_j$ into either $x_i \to x_j$ or $x_j \to x_i$ at random.

Example 17 (*Boolean Case*)**.** In order to calculate the exact number of changes we use random networks with 10 nodes. For each network, we selected the node with the maximum in-degree and generated a random partition of its inputs to assign the canalizing layers. The functions of the other nodes were generated at random.

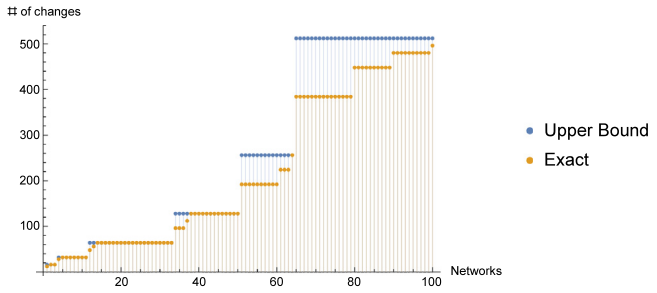

Fig. 1. Statistics for the number of changes in the first layer of scale-free Boolean networks. The x-axis shows the 100 networks that were randomly generated and the y-axis shows the number of changes corresponding to a network in the x-axis. In Fig. 2 we plot the differences between upper bounds and the exact number of changes for these networks.

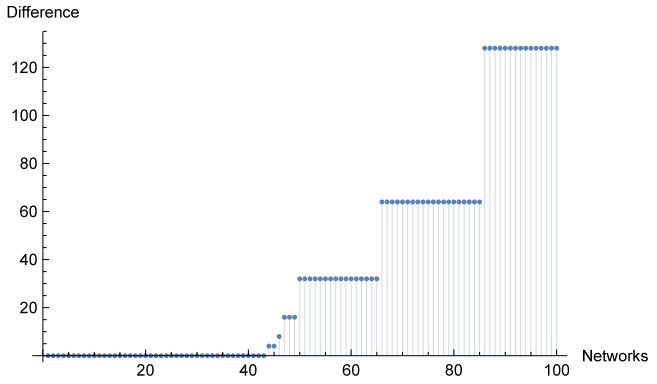

Fig. 2. Statistics for the differences between the upper bounds and the exact number of changes for the networks in Fig. 1. In about 40% of the networks the upper bounds match the exact number of changes.

In Fig. 1 we show statistics for the number of changes in the first layer. The average maximum in-degree for the networks in Fig. 1 is 4.14 ($std = 1.07$). The average number of variables in the first layer is 2.61 ($std = 1.5$). The average number of changes in the first layer is 221.28 ($std = 168.363$) and the average upper bound is 259.04 ($std = 201.562$).

In Fig. 2 we present statistics of the number of changes as well as the difference between the exact number of changes and the upper bound provided by the formulas. For these networks, in about 40% of the cases the upper bounds match the exact number of changes.

In Fig. 3 we show statistics for the number of changes in the second layer. The average maximum in-degree for the networks in Fig. 3 is 4.8 ($std = 1.07$). The average number of variables in the first layer is 1.67 ($std = 0.93$). The average number of variables in the second layer is 2.13 ($std = 0.75$). The average number of changes in the second layer is 86.0 ($std = 62.7$) and the average upper bound is 100.64 ($std = 78.88$).

In Fig. 4 we present statistics of the number of changes as well as the difference between the exact number of changes and the upper bound provided by the formulas. For these networks, in about 50% of the cases the upper bounds match the exact number of changes.

From Figs. 2 and 4, we see that the bounds are slightly more accurate when the edge intervention happens in a less dominant layer.

Example 18 (*Multistate Case*)**.** Here we also use random networks with scale-free structure with $p = 3$ and $n = 10$ nodes. For each network, we selected the node with the maximum in-degree and

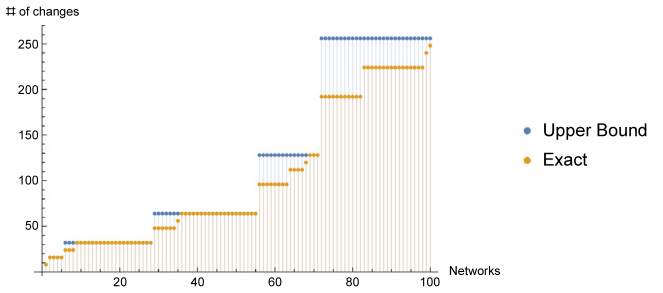

Fig. 3. Statistics for the number of changes in the second layer of scale-free Boolean networks. The x-axis shows the 100 networks that were randomly generated and the y-axis shows the number of changes corresponding to a network in the x-axis. In Fig. 4 we plot the differences between upper bounds and the exact number of changes for these networks.

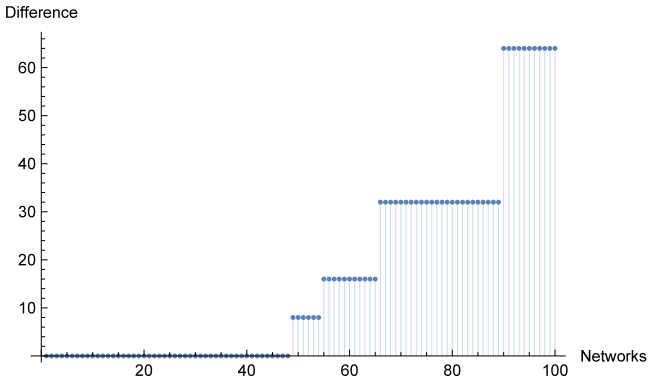

Fig. 4. Statistics for the differences between the upper bounds and the exact number of changes for the networks in Fig. 3. In about 50% of the networks the upper bounds match the exact number of changes.

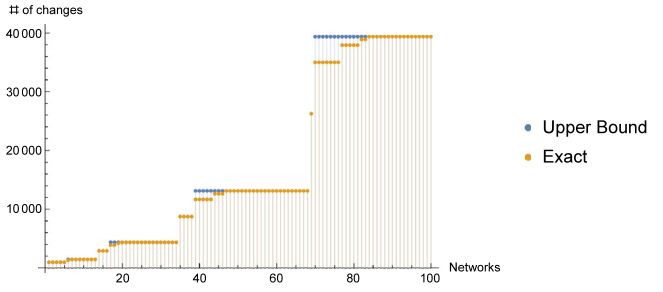

Fig. 5. Statistics for the number of changes in the first layer of scale-free multistate networks. The x-axis shows the 100 networks that were randomly generated and the y-axis shows the number of changes corresponding to a network in the x-axis. In Fig. 6 we plot the differences between upper bounds and the exact number of changes for these networks.

generated a random partition of its inputs to assign the canalizing layers. The functions of the other nodes were generated at random.

In Fig. 5 we show statistics for the number of changes in the first layer. The average maximum in-degree for the networks in Fig. 5 is 4.05 ($std = 0.88$). The average number of variables in the first layer is 2.28 ($std = 1.16$). The average number of changes in the first layer is 17303.2 ($std = 14739.4$) and the average upper bound is 17792.5 ($std = 15220.6$).

In Fig. 6 we present statistics of the difference between the upper bounds provided by the formulas and the exact number of changes. For these networks, in about 75% of the cases the upper bounds match the exact number of changes.

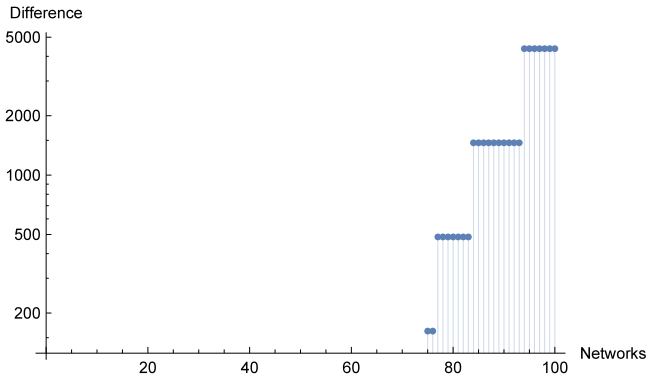

Fig. 6. Statistics for the differences between the upper bounds and the exact number of changes for the networks in Fig. 5. In about 75% of the networks the upper bounds match the exact number of changes. The vertical axis is in logarithmic scale.

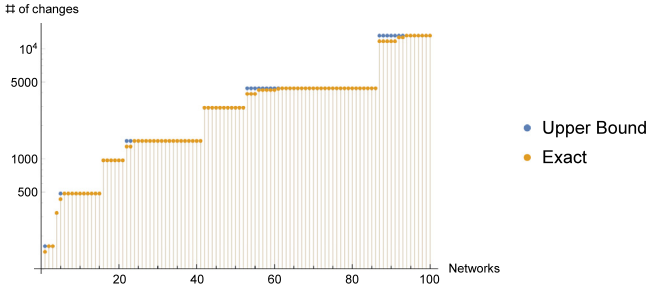

Fig. 7. Statistics for the number of changes in the second layer of scale-free multistate networks with $p = 3$ and $n = 10$ nodes. The x-axis shows the 100 networks that were randomly generated and the y-axis shows the number of changes corresponding to a network in the x-axis. The vertical axis is in logarithmic scale. In Fig. 8 we plot the differences between upper bounds and the exact number of changes for these networks.

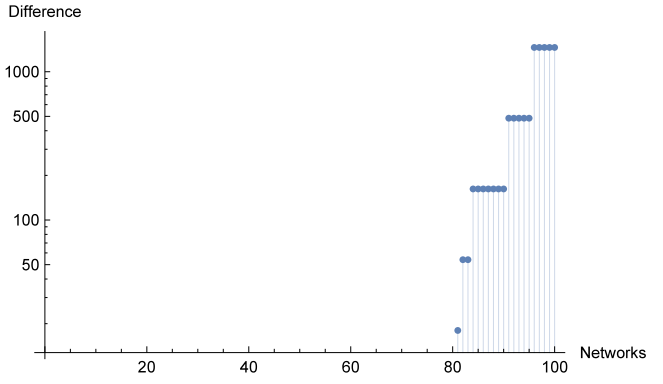

Fig. 8. Statistics for the differences between the upper bounds and the exact number of changes for the networks in Fig. 7. In about 80% of the networks the upper bounds match the exact number of changes. The vertical axis is in logarithmic scale.

In Fig. 7 we show statistics for the number of changes in the second layer. The average maximum in-degree for the networks in Fig. 7 is 4.84 ($std = 0.94$). The average number of variables in the first layer is 1.7 ($std = 0.86$). The average number of variables in the second layer is 1.93 ($std = 0.97$). The average number of changes in the second layer is 3946.68 ($std = 3789.71$) and the average upper bound is 4056.48 ($std = 3969.7$).

In Fig. 8 we present statistics of the difference between the upper bound provided by the formulas and the exact number of

changes. For these networks, in about 80% of the cases the upper bounds match the exact number of changes.

From Figs. 6 and 8, we see that the bounds are slightly more accurate when the edge intervention happens in a less dominant layer.

6. Conclusions

In this paper we present practical methods for quantifying the global changes that result from an application of an external intervention in the network, which we called the total effect of the intervention. We emphasized that, while there are several methods for identifying control targets in discrete networks, there have been very few studies focusing on the total extent of the consequences of applying a certain control action (beyond satisfying the control objective). This paper contributes methods for measuring the number of changed transitions in the state space upon the application of an edge control in multistate networks. The approach is based on a polynomial normal form description of discrete functions that provides a way of categorizing the inputs of the function and therefore of quantifying their impact on the dynamics of the network. The main computational cost of our approach is in obtaining the canalizing layers format of the functions which we used to derive our formulas. Once the functions are written in the layers format, it is straightforward to apply the formulas to calculate the upper bound. We note that obtaining the layers format could be a formidable task with exponential complexity on the number of inputs in the worst case but for the type of networks we are studying (i.e. biological networks) we believe that our approach is still practical. We applied our methods to randomly generated multistate models and verified that in many cases the upper bounds provided by our formulas were accurate. We also observed that the upper bounds tend to be more accurate when the edge interventions happen in a less dominant layer.

Acknowledgments

We would like to thank Qijun He and Matthew Macauley for useful discussions during the preliminary stage of this work. We also would like thank the anonymous reviewers for many suggestions that improved this article.

References

Albert, Réka, & Othmer, Hans G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of Theoretical Biology*, *223*(1), 1–18.

Barabasi, & Albert (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512.

Campbell, Colin, & Albert, Réka (2019). Edgetic perturbations to eliminate fixed-point attractors in boolean regulatory networks. *Chaos*, *29*(2), Article 023130.

Chifman, Julia, Arat, Seda, Deng, Zhiyong, Lemler, Erica, Pino, James C, Harris, Leonard A, et al. (2017). Activated oncogenic pathway modifies iron network in breast epithelial cells: A dynamic modeling perspective. *PLoS Computational Biology*, *13*(2), Article e1005352.

Choi, Minsoo, Shi, Jue, Jung, Sung Hoon, Chen, Xi, & Cho, Kwang-Hyun (2012). Attractor landscape analysis reveals feedback loops in the p53 network that control the cellular response to DNA damage. *Science Signaling*, *5*(251), ra83.

Espinosa-Soto, Carlos, Padilla-Longoria, Pablo, & Alvarez-Buylla, Elena R (2004). A gene regulatory network model for cell-fate determination during arabidopsis thaliana flower development that is robust and recovers experimental gene expression profiles. *Plant Cell*, *16*(11), 2923–2939.

Gates, Alexander J., & Rocha, Luis M. (2016). Control of complex networks requires both structure and dynamics. *Scientific Reports*, *6*, 24456.

He, Qijun, & Macauley, Matthew (2016). Stratification and enumeration of boolean functions by canalizing depth. *Physica D: Nonlinear Phenomena*, *314*, 1–8.

Huang, S. (1999). Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Media (Berlin)*, *77*(6), 469–480.

Kadelka, Claus, Li, Yuan, Kuipers, Jack, Adeyeye, John O, & Laubenbacher, Reinhard (2017). Multistate nested canalizing functions and their networks. *Theoretical Computer Science*, *675*, 1–14.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*(3), 437–467.

Kauffman, Stuart, Peterson, Carsten, Samuelsson, Björn, & Troein, Carl (2003). Random boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences*, *100*(25), 14796–14799.

Kauffman, Stuart, Peterson, Carsten, Samuelsson, Björn, & Troein, Carl (2004). Genetic networks with canalyzing boolean rules are always stable. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(49), 17102–17107.

Klamt, Steffen, Saez-Rodriguez, Julio, Lindquist, Jonathan A, Simeoni, Luca, & Gilles, Ernst D (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, *7*, 56.

Li, Fangting, Long, Tao, Lu, Ying, Ouyang, Qi, & Tang, Chao (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(14), 4781–4786.

Li, Rui, Yang, Meng, & Chu, Tianguang (2015). Controllability and observability of boolean networks arising from biology. *Chaos*, *25*(2), Article 023104.

Murrugarra, David, & Dimitrova, Elena S. (2015). Molecular network control through boolean canalization. *EURASIP Journal of Bioinformatics Systems and Biology*, *2015*(1), 9.

Murrugarra, David, & Laubenbacher, Reinhard (2011). Regulatory patterns in molecular interaction networks. *Journal of Theoretical Biology*, *288*, 66–72.

Murrugarra, David, & Laubenbacher, Reinhard (2012). The number of multistate nested canalyzing functions. *Physica D: Nonlinear Phenomena*, *241*(10), 929–938.

Murrugarra, David, Veliz-Cuba, Alan, Aguilar, Boris, & Laubenbacher, Reinhard (2016). Identification of control targets in boolean molecular network models via computational algebra. *BMC Systems Biology*, *10*(1), 94.

Poret, Arnaud, & Boissel, Jean-Pierre (2014). An in silico target identification using boolean network attractors: Avoiding pathological phenotypes. *Comptes Rendus Biologies*, *337*(12), 661—678.

Qiu, Yushan, Tamura, Takeyuki, Ching, Wai-Ki, & Akutsu, Tatsuya (2014). On control of singleton attractors in multiple boolean networks: integer programming-based method. *BMC Systems Biology*, *8 Suppl 1*, S7.

Remy, E., Rebouissou, S., Chaouiya, C., Zinovyev, A., Radvanyi, F., & Calzone, L. (2015). A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis. *Cancer Research*, *75*(19), 4042–4052.

Shmulevich, Ilya, & Dougherty, Edward R. (2010). *Probabilistic boolean networks - The modeling and control of gene regulatory networks*. SIAM.

Sordo Vieira, Luis, Laubenbacher, Reinhard C, & Murrugarra, David (2019). Control of intracellular molecular networks using algebraic methods. *Bulletin Mathematique Biology*, *82*(1), 2.

Thieffry, D., & Thomas, R. (1995). Dynamical behaviour of biological regulatory networks–II. Immunity control in bacteriophage lambda. *Bulletin Mathematique Biology*, *57*(2), 277–297.

Thomas, René, & D'Ari, Richard (1990). *Biological feedback*. Boca Raton: CRC Press.

Veliz-Cuba, Alan, Jarrah, Abdul Salam, & Laubenbacher, Reinhard (2010). Polynomial algebra of discrete models in systems biology. *Bioinformatics*, *26*(13), 1637–1643.

Vera-Licona, Paola, Bonnet, Eric, Barillot, Emmanuel, & Zinovyev, Andrei (2013). OCSANA: optimal combinations of interventions from network analysis. *Bioinformatics*, *29*(12), 1571–1573.

Waddington, C. H. (1957). *The strategy of the genes: a discussion of some aspects of theoretical biology*. London: Allen & Unwin.

Wagner, Andreas (1996). Does evolutionary plasticity evolve?. *Evolution*, *50*(3), 1008–1023.

Zañudo, Jorge G. T., & Albert, Réka (2015). Cell fate reprogramming by control of intracellular network dynamics. *PLoS Computational Biology*, *11*(4), Article e1004193.

Zañudo, Jorge Gómez Tejeda, Scaltriti, Maurizio, & Albert, Réka (2017). A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer. *Cancer Convergence*, *1*(1), 5.

Zañudo, Jorge Gomez Tejeda, Yang, Gang, & Albert, Réka (2017). Structure-based control of complex networks with nonlinear dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(28), 7234–7239.

David Murrugarra is an Associate Professor in the Department of Mathematics at the University of Kentucky. He received his Ph.D. from Virginia Tech in 2012. He was a postdoc at Georgia Tech before moving to Kentucky in 2014. His research program focuses on the development of computational tools for modeling, analysis, and control of regulatory networks. His research interests also include the computational prediction of RNA secondary structure using machine learning techniques.

Elena Dimitrova is an Associate Professor in the Department of Mathematics at the California Polytechnic State University (Cal Poly), San Luis Obispo and a Senior Researcher at the Southeast Center for Mathematics and Biology. She received her Ph.D. from Virginia Tech in 2006 and was a faculty member at Clemson University until 2019 when she moved to Cal Poly. Her research interests include algebraic design of experiments and polynomial dynamical systems for modeling and control of cellular and molecular processes.