Int J Adv Eng Sci Appl Math (2019) 11(2):153–171
https://doi.org/10.1007/s12572-018-0237-6

IIT, Madras

Graphical dynamical systems and their applications to bio-social systems

Abhijin Adiga[1] · Chris J. Kuhlman[1] · Madhav V. Marathe[1,2] · Henning S. Mortveit[1] · S. S. Ravi[1,3] · Anil Vullikanti[1,2]

Published online: 15 December 2018

Abstract In this review paper, we discuss graphical dynamical systems (GDSs) and their applications to biological and social systems (bio-social systems). Traditionally, differential equation-based models have been central in modeling bio-social systems. GDSs provide an alternate modeling framework. This framework explicitly represents individual components of the system and captures the interactions among them via a network. The purpose of this review is to enable modelers to obtain an understanding of this basic mathematical and computational framework so that it can be used to study specific bio-social applications. The work covers the range from computational theory to simulation-based analysis. We also provide some directions for future work.

✉ Madhav V. Marathe
mvm7hz@virginia.edu

Abhijin Adiga
abhijin@virginia.edu

Chris J. Kuhlman
cjk8gx@virginia.edu

Henning S. Mortveit
henning.mortveit@vt.edu

S. S. Ravi
ssravi@virginia.edu

Anil Vullikanti
vsakumar@virginia.edu

[1] Biocomplexity Institute and Initiative, University of Virginia, 1001 Research Park Boulevard, Charlottesville, VA 22911, USA

[2] Present Address: Computer Science Department, University of Virginia, Charlottesville, VA, USA

[3] University at Albany – State University of New York, Albany, NY, USA

Keywords Graphical dynamical systems · Mathematical modeling · Simulation science · Complexity theory · Biological and social systems

1 Introduction

This review article focuses on the foundations of large interacting biological and social (bio-social) systems. Such systems are comprised of a large number of interacting entities/agents. The global properties of such systems are among the outcomes of the interactions of individual agents/entities with a relatively small number of other entities. In general, the underlying interaction structure (often called a network) is heterogeneous and time varying. Examples of such systems include: (1) the mammalian immune system, (2) epidemiology of infectious diseases, (3) gene interaction networks and (4) spatial ecologies. These systems are highly nonlinear and complex; the heterogeneity in such systems exists among individual agents as well as the interactions among them. This heterogeneity is a hallmark of such systems and is often responsible for the rich dynamics they exhibit. Furthermore, the global dynamics and the interaction structure coevolve. This makes studying such systems challenging. Simple aggregate models, although good starting points, are not adequate to capture the complexity of the systems and lead to results that ignore the effects of heterogeneities. What is needed is a mathematical formalism that can capture the interaction structures and diversity of local agents and their interactions.

In this article, we describe a formal theory based on *graphical dynamical systems* to study such massive bio-social systems. Abstractly, a graphical dynamical system

(GDS) captures the following situation: There is a set of entities (e.g., genes in a gene regulatory network) represented by a set $\{1, 2, \ldots, n\}$ of nodes, where each node i has a state s_i from a set K such as $\{0, 1\}$. Here $s_i = 1$ could encode that gene i is expressed, while $s_i = 0$ could encode the non-expressed state. A set of *local transition functions* $\mathcal{F} = \{f_1, \ldots, f_n\}$ governs the local dynamics in the sense that f_i is used to determine how the state of node i evolves from time t to $t + 1$ for some suitable time scale. To specify how the system evolves globally, an update scheme $\mathcal{U}$ is used to determine how the functions $\mathcal{F}$ assemble to a map $F : K^n \longrightarrow K^n$ of the form

$$F = (F_1, F_2, \ldots, F_n). \tag{1}$$

For example, if the update scheme is to "apply the functions f_i in parallel," we have $F_i = f_i\,(1 \leq i \leq n)$ in Eq. (1). We note that the set $\mathcal{F}$ has an associated graph G called the *dependency graph* or *wiring diagram* that captures the dependence among variables: This graph has nodes $\{1, 2, \ldots, n\}$ (e.g., the genes), and there is a directed edge $\{i, j\}$ whenever a function f_i depends on the state of node j. Generally, however, the graph G is given by the system or application (e.g., the graph of the gene regulatory network).

Abstracting the above, a graphical dynamical system (GDS) is a triple

$$(G, \mathcal{F}, \mathcal{U}) \tag{2}$$

with an associated GDS map $F : K^n \longrightarrow K^n$ as in Eq. (1) assembled from $\{f_1, \ldots, f_n\}$ via the update scheme $\mathcal{U}$. A large part of GDS theory seeks to infer properties about dynamics from its defining constituents G, $\mathcal{F}$ and $\mathcal{U}$, an approach often referred to as *structure-to-function* analysis.

1.1 Agent-based models and GDS

GDSs are closely related to agent-based models (ABMs); in fact, they are a natural language for formally specifying ABMs. In other words, given the description of an ABM under the GDS framework, formal statements about the dynamics of the ABM can be readily translated into formal properties of the corresponding GDS. Also, formal properties determined from analysis of a GDS can guide or assist in verification of an ABM. As discussed in recent papers [1–4], ABMs are thought of as a modeling framework for generative sciences. We will discuss this aspect further in Sect. 5. For many classes of GDSs, computational theory can be used to demonstrate that certain analysis questions are computationally intractable; that is, unless some widely believed hypotheses in computational complexity turn out to be false, exhaustive enumeration is provably the most efficient way to solve those analysis problems. To cope with this intractability, systems are usually analyzed through computation using a *simulation model*, that is, by judiciously running an implementation of the mathematical model describing the system on a computer (see Sect. 5). In fact, the interplay between analysis and carefully executed simulations can be a powerful driver of the theory.

In Sect. 4, we present some theory that assesses the *computational cost* associated with GDS analysis questions over its phase space, such as determining the number of *fixed points*.[1]

1.2 Organization

We review the motivation behind the construction of the GDS framework in Sect. 2 and also discuss related work. Terminology and definitions are given in Sect. 3 before presenting computational theory and applications in Sect. 4. Simulation approaches are presented in Sect. 5. A brief discussion on a game theoretic view of GDSs appears in Sect. 6. We close with some directions for future work in Sect. 7.

2 GDS background and related work

2.1 Background

The framework of graphical dynamical systems was introduced by the authors in the context of *massively interacting systems*. Examples of systems motivating the GDS model include the transportation analysis system TRANSIMS [5], networked epidemiology systems EpiSims, EpiSimdemics, Epifast and Indemics [6–9], sociotechnical systems where humans interact with critical infrastructure in crisis scenarios [10, 11] and the mammalian immune system [12, 13]. See [2–4, 14] for additional examples in the context of energy systems, computational social sciences and ecological systems.

A mathematical and computational theory based on GDS aims to (1) accurately model large interacting biosocial systems such as the above, (2) express analysis questions involving policy formulation, interventions and control in a precise formal manner, (3) support analytic reasoning for model validation, verification, sensitivity analysis, uncertainty quantification, optimization, etc. and (4) facilitate mapping of these models efficiently onto high-performance computing architectures [9, 15]. GDSs provide a natural framework to abstract the essence of analysis problems and therefore develop theory across application domains.

[1] Concepts such as phase space and fixed points are defined in Sect. 3.

2.2 Related work

Several subclasses of graphical dynamical systems have been analyzed in the literature. *Cellular automata* (CA) were introduced by von Neumann in [16] as a model for computation and were later studied by many others [17–20]. Generally, a CA is defined over a regular graph such as a k-dimensional lattice.

Boolean networks were introduced by Kauffman as models for gene regulatory networks; see [21, 22]. In the original setting, this class used $\{0, 1\}$ as the state space, typically involved regular graphs, and employed a synchronous update scheme (see Sect. 3). Some of these conditions have been relaxed in later work. *Automata networks* were introduced by Goles and, in many ways, overlap with synchronous and sequential GDSs [23–26]. *Probabilistic (or random) Boolean networks* (PBNs) are constructed as Boolean networks, but the vertex functions are chosen in a stochastic manner; in this case, one may view the evolution as that of an ensemble of Boolean networks, one for each function configuration. PBNs have been studied as a modeling framework for gene regulatory networks [27–30]. Synchronous and sequential graphical dynamical systems are *finite dynamical systems* and may therefore be regarded as *polynomial dynamical systems* [31, 32]. This approach is a starting point for the use of approaches based on computational algebra to analyze the dynamics as well as to reconstruct the system from appropriate time series data.

In our work, the focus is on the discrete time/discrete state setting. We remark that there is active research on networked ordinary differential equation (ODE) models (see [33–35]); however, these have yet to scale to the levels of GDS-based simulation models. It may in fact be interesting to see which results under the GDS framework extend to the analysis of networked ODE models. Examples of Boolean network models as alternatives to ODE-based models of biological systems are given in [36, 37].

Computationally, GDSs are closely related to other well-studied models, including recurrent neural networks (including Hopfield networks), concurrent transition systems and graph automata; see [2–4, 38–40] for further discussion on this topic.

3 Graphical dynamical system terminology and definitions

3.1 Synchronous and sequential graphical dynamical systems

This section presents definitions and terminology used throughout the paper. Let $G(V, E)$ be a graph on n nodes. Here we assume that the graph is undirected; extending definitions to the directed case is straightforward, but adds bookkeeping. Each node is assigned a state from a finite domain K, frequently taken to be the two-valued Boolean domain $\mathbb{B} = \{0, 1\}$. We denote the state assigned to node i by s_i, or by s_i^t, when we want to explicitly reference the time t. A *configuration* at time t is an n-tuple $s^t = (s_1^t, s_2^t, \ldots, s_n^t) \in K^n$. To each node i, we assign a *G-local transition function* f_i. This function is local in the sense that it only depends on the state of node i and those of the neighbors of i in G. We write $n[i]$ for the sorted sequence of these nodes and denote by $s[i]$ the corresponding sub-configuration of s. We also introduce the functions $F_i : K^n \longrightarrow K^n$ defined by

$$F_i(s_1, \ldots, s_n) = (s_1, \ldots, s_{i-1}, f_i(s[i]), s_{i+1}, \ldots, s_n). \tag{3}$$

Definition 1 (*Synchronous/sequential graphical dynamical systems*) [38, 41] Let $G(V, E)$ be a graph, $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ local transition functions and $\pi = (\pi_1, \ldots, \pi_n)$ a permutation of V. The synchronous graph dynamical system (SyGDS) over graph G with local transition functions $\mathcal{F}$ is the triple $\mathcal{S} = (G, \mathcal{F}, \mathcal{U} = \text{synch})$ with associated map $F : \mathbb{B}^n \longrightarrow \mathbb{B}^n$ defined by

$$F(s_1, s_2, \ldots, s_n) = (f_1(s[1]), f_2(s[2]), \ldots, f_n(s[n])). \tag{4}$$

The triple $\mathcal{S} = (G, \mathcal{F}, \pi)$ is the *sequential graphical dynamical system* (SeGDS) over G with local transition functions $\mathcal{F}$, update sequence π and associated map $F_\pi : K^n \longrightarrow K^n$ defined by

$$F_\pi = F_{\pi_n} \circ F_{\pi_{n-1}} \circ \cdots \circ F_{\pi_1}. \tag{5}$$

A key difference between a SyGDS and an SeGDS is that for an SeGDS, all states $s_{\pi_1}^{t+1}$ though $s_{\pi_i}^{t+1}$ are used in the update of vertex states $s_{\pi_{j>i}}^t$. More general update schemes such as block synchronous updates [42] and stochastic rate-based schemes [43] have also been studied.

3.1.1 Classes of local transition functions

Here we present several classes of Boolean local transition functions referenced throughout the paper. One such class is symmetric Boolean functions [38]. A Boolean function $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ is symmetric if there exists a function $\eta : \{0, 1, \ldots, n\} \longrightarrow \mathbb{B}$ such that $f(s_1, \ldots, s_n) = \eta(\sum_i s_i)$ for all $s \in \mathbb{B}^n$. Clearly, a symmetric Boolean function can be specified by giving the $n + 1$ values for its matching map η. *Threshold functions* (e.g., [40, 44]) are symmetric Boolean functions specified by an integer $h \geq 0$. The h-threshold function has value 1 precisely when h or more of its inputs are 1; see [45, 46] for applications.

Example 1 Consider the SyGDS whose underlying graph is shown in Fig. 1. The local transition functions at the

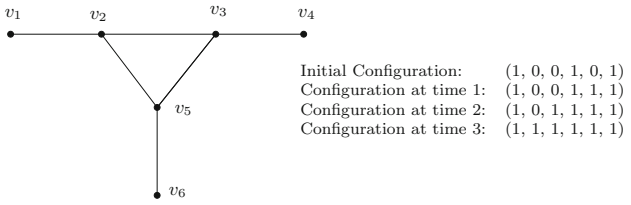

Fig. 1 SyGDS of Example 1. Each configuration has the form $(s_1, s_2, s_3, s_4, s_5, s_6)$, where s_i is the state of node v_i with $1 \leq i \leq 6$. The configuration at time 3 is a fixed point

nodes v_1, v_4, v_5, v_6 are one-threshold functions; the function at v_3 is the two-threshold function, while the function at v_2 is the three-threshold function. Assume that initially v_1, v_4 and v_6 are in state 1 and all other nodes are in state 0. During the first time step, the state of node v_5 changes to 1 since its neighbor v_6 is in state 1; the states of other nodes do not change. The configurations at subsequent time steps are shown in the figure. The system reaches the configuration (1, 1, 1, 1, 1, 1) at time step 3. Subsequently, none of the nodes changes its state. Thus, the configuration (1, 1, 1, 1, 1, 1) is a *fixed point* for this system. □

A more general class of Boolean functions is that of r-symmetric functions where $r \geq 1$ is an integer. A Boolean function f is r-symmetric function if its input arguments can be partitioned into r subsets A_1 through A_r such that f factors as $f(s) = \eta'(\sum_{i \in A_1} s_i, \ldots, \sum_{i \in A_r} s_i)$. This class generalizes threshold functions ($r = 1$) as well as bi-threshold functions ($r = 2$) [47–50], that is, functions for which the transitions from 0 to 1 and from 1 to 0 are governed by separate threshold functions with thresholds t_1 and t_0, respectively.

Let $F : K^n \longrightarrow K^n$ be a GDS map. If $F(\mathcal{C}) = \mathcal{C}'$, we say that $\mathcal{C}'$ is the successor of $\mathcal{C}$ and that $\mathcal{C}$ is a predecessor of $\mathcal{C}'$ under F. A configuration $\mathcal{C}$ is called a fixed point if the successor of $\mathcal{C}$ is $\mathcal{C}$ itself. A configuration $\mathcal{C}$ with no predecessors is called a Garden of Eden (GE) configuration. A configuration $\mathcal{C}$ for which there exists an integer $r > 0$ such that $F^r(\mathcal{C}) = \mathcal{C}$ is a periodic point, and the sequence $(\mathcal{C}, F(\mathcal{C}), F^2(\mathcal{C}), \ldots, F^{r-1}(\mathcal{C}))$ is a limit cycle. A configuration that is not a periodic point is called a transient point.

The phase space of a GDS map $F : K^n \longrightarrow K^n$, denoted by $\mathcal{P}_F$, is the directed graph defined as follows: Its node set is K^n and its edge set is $\{(\mathcal{C}, F(\mathcal{C})) \mid \mathcal{C} \in K^n\}$. If $K = \mathbb{B}$, the number of nodes in the phase space is 2^n; thus, the size of phase space is *exponential* in the size of the description of the graph G. Clearly, each node in the phase space has outdegree 1, fixed points are those nodes having a self-loop, and GE configurations are the nodes with indegree zero. A forward trajectory of F is a sequence of the form $(s, F(s), F^2(s), \ldots, F^k(s))$ for some integer $k \geq 0$.

Example 2 A small SyGDS with 3 nodes and its phase space (which has 8 nodes) are shown in Fig. 2. In this example, (0, 0, 1) is the successor of (1, 1, 0). Both (0, 0, 1) and (1, 0, 0) are predecessors of (0, 1, 1). Further, both (0, 1, 1) and (0, 1, 0) are fixed points and both (1, 0, 1) and (1, 0, 0) are Garden of Eden configurations. The sequence $(0, 0, 0) \rightarrow (1, 1, 0) \rightarrow (0, 0, 1) \rightarrow (0, 1, 1)$ is a forward trajectory.

3.2 Stochastic SyGDS

SyGDSs with stochastic local transition functions, denoted by SSyGDSs, are useful in many applications; see [51, 52]. Using states from a domain $\mathbb{D}$ and denoting the neighbors of node i by $n[i] = (i_1, \ldots, i_r)$, the local transition functions are specified in this case as a collection of probabilities

$$f_i(\theta', \theta^1_{i_1}, \ldots, \theta^1_{i_r}, \theta) = Pr\{s^t_i = \theta | s^{t-1}_i = \theta', s^{t-1}_{i_j} = \theta^1_{i_j}, 1 \leq j \leq r\}, \tag{6}$$

where for each combination of inputs, the sum of the probabilities assigned by f_i over all the values $\theta \in \mathbb{D}$ must be 1. All nodes update their states synchronously and independently of each other. Thus, for any pair of configurations $\mathcal{C}_1 = (b_1, b_2, \ldots, b_n)$ and $\mathcal{C}_2 = (b'_1, b'_2, \ldots, b'_n)$ we have

$$Pr\{\mathcal{C}_1 \rightarrow \mathcal{C}_2\} = \prod_{i=1}^{n} f_i(b_i, b_{i_1}, \ldots, b_{i_r}, b'_i). \tag{7}$$

The generalized phase space $\mathcal{P}_{\mathcal{S}}$ of an SSyGDS $\mathcal{S}$ is the directed graph where the nodes are the set of all possible configurations. There is a directed edge from $\mathcal{C}_1$ to $\mathcal{C}_2$ labeled by $p = Pr\{\mathcal{C}_1 \rightarrow \mathcal{C}_2\}$ whenever $p > 0$. A SSyGDS is thus a particular type of Markov chain on the configuration space; the size of the Markov chain is *exponential* in the size of the underlying graph G.

Example 3 A SSyGDS whose graph G has the node set $V = \{a, b, c, d\}$ is shown in the left panel of Fig. 3. For this system, the transition function f_a is the three-input stochastic OR function: When all its inputs are 0, $Pr\{f_a = 1\} = 0$; for all other input combinations, we specify that $Pr\{f_a = 1\} = 1/2$. The function f_b is a deterministic three-input OR function: When all its inputs are 0, $Pr\{f_b = 1\} = 0$; when at least one of its inputs is 1, $Pr\{f_b = 1\} = 1$. The transition function f_c is a four-input stochastic AND function where $Pr\{f_c = 1\} = 3/4$ for the input configuration (1, 1, 1, 1) and $Pr\{f_c = 1\} = 0$ in all other cases. Finally, f_d is a two-input deterministic OR function similar to f_b. From the specifications of the stochastic local transition functions, the reader can verify that for $\mathcal{C}_1 = (1, 1, 1, 1)$ and $\mathcal{C}_2 = (0, 1, 0, 0)$, Pr

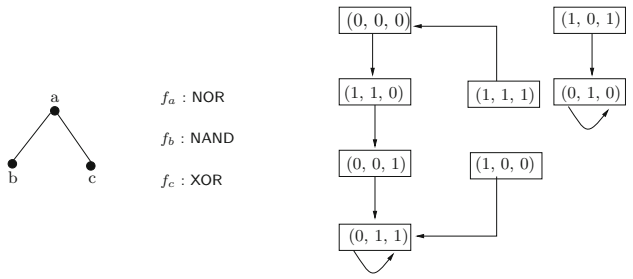

Fig. 2 A small SyGDS and its phase space (see Example 2). *Note*: The domain of state values for each node is $\{0, 1\}$. As indicated above, the Boolean functions at nodes a, b and c are NOR, NAND and XOR, respectively. (The configuration (1, 0, 1) indicates that $s_a = 1$, $s_b = 0$ and $s_c = 1$.)

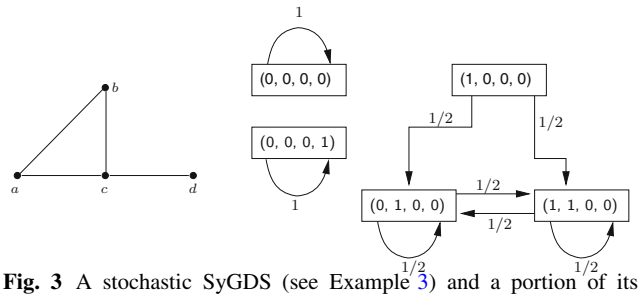

Fig. 3 A stochastic SyGDS (see Example 3) and a portion of its generalized phase space. The complete generalized phase space, which has 16 nodes and many edges, is not shown to avoid clutter

$\{\mathcal{C}_1 \to \mathcal{C}_2\} = 1/8$. A portion of the generalized phase space of the SSyGDS in the left panel of Fig. 3 is shown in the right panel of that figure.

3.2.1 Notational convention

For the rest of the paper, we use GDS to mean graphical dynamical systems without any specific constraints on update order, location functions or the type of network. We will use SeGDS and SyGDS to refer to a sequential graphical dynamical system and a synchronous graphical dynamical system, respectively. Finally, we use SSyGDS to refer to a stochastic synchronous graphical dynamical system. In some of our earlier papers, we have used SDS and SyDS to refer to sequential graphical dynamical systems and synchronous graphical dynamical systems, respectively. The current notation makes the graphical aspect more explicit.

3.2.2 Additional notes

Stochastic sequential graphical dynamical systems are not discussed in this paper. Due to space reasons, most of the results discussed here pertain to SyGDSs; in many cases, analogous results hold for SeGDSs as well. Furthermore, we assume that the underlying graph G is given. Synthesizing the network that models a bio-social system is an important area of research. Finally, again for space reasons, our focus on formal GDS results in the next section is on *decision* problems. *Counting* problems (i.e., find the number of solutions), *ambiguous* problems (i.e., given a solution, decide whether there is a different solution) and *uniqueness* problems (i.e., decide whether the instance has exactly one solution) are not discussed here; nevertheless, they are interesting and have been studied in the literature (for example, see [53–55]).

4 Application of computational and complexity theory to GDSs

4.1 Overview

We now discuss some computational problems (called analysis problems) for dynamical systems that are useful in addressing questions that arise in the study of bio-social systems. We first discuss these problems for deterministic SyGDSs and then mention the extensions to stochastic SyGDSs. Our focus is on the computational complexity results for these problems and the significance of such results.

4.2 Analysis problems under the SyGDS model

Analysis problems arise in the context of determining the *dynamic behavior* of a system, given its static description. We consider systems whose static description is specified as a SyGDS; that is, it is assumed that the underlying graph and the local transition functions are given. Since the phase space of a SyGDS captures the dynamic behavior of the corresponding system, analysis questions correspond to structural properties of the directed graph representing the phase space of the given SyGDS. In this section, we define several such questions for SyGDSs and summarize known results about those questions.

Of the many applications that can be studied using the SyGDS model, we focus on diffusion processes, which arise in diverse kinds of phenomena, such as the spread of epidemics, fads and beliefs in social networks (see, e.g., [46, 56–59]). Here, the underlying graph of the SyGDS represents a social contact network, with each node representing a person and an edge between two nodes indicating that the corresponding people came into contact with each other; thus, each edge represents an opportunity for one of the end points to influence the other. Each node takes on state values from $\{0, 1\}$. We say that a node in state 1(0) is "infected" ("uninfected"). Here, infection is a general term denoting that the node is infected due to a

disease or has adopted a new belief, etc., depending on the phenomenon being modeled. The local transition function at a node v determines whether node v gets infected when some of its neighbors are infected. For example, suppose the local transition function at v is the k-threshold function for some integer $k \geq 1$. In such a case, node v gets infected at a time step t if k or more of its neighbors were infected prior to t. Starting from an initial configuration (where some nodes are in state 1), the system goes through a sequence of successive configurations. The analysis problems discussed below capture several aspects of this form of dynamical behavior.

4.2.1 Configuration reachability

This form of analysis problem concerns the *reachability* of certain configurations. Formally, given a SyGDS $\mathcal{S}$ and two configurations $\mathcal{C}$ and $\mathcal{C}'$, the goal of the reachability problem is to determine whether $\mathcal{S}$ starting from $\mathcal{C}$ can reach $\mathcal{C}'$. One can also formulate a timed version of the reachability problem where an additional time parameter t is specified as input to the problem, and the goal is to decide whether $\mathcal{C}'$ can be reached in t or fewer time steps. In the context of diffusion processes, one application of the reachability problem is in deciding whether an undesirable situation (such as a large infected population) can occur within t time steps, given the current conditions.

Many papers in the literature have addressed the reachability problem for discrete dynamical systems. Barrett et al. [38, 60] showed that, in general, the problem is computationally intractable (technically, PSPACE-complete[2]) even when all nodes have the same symmetric local transition function. When each local transition function is a threshold function, it was shown in [38] that the reachability problem can be solved efficiently. However, when both threshold functions and negative threshold functions[3] are permitted as local transition functions, the reachability problem becomes PSPACE-complete [40]. It is also known that the reachability problem for SyGDSs with bi-threshold functions can be solved efficiently when all the threshold values are 1 [48]. Kuhlman et al. [50] show that the reachability problem remains efficiently solvable for SyGDSs with bi-threshold local transition functions when the underlying graph is directed and acyclic. For some classes of local transition functions, upper and lower bounds on the number of time units to reach fixed points were established in [62].

[2] For definitions concerning complexity classes, we refer the reader to [61].

[3] A negative threshold function is the negation of a threshold function. For example, a negative three-threshold function has the value 1 if and only if two or fewer of its inputs have the value 1.

4.2.2 Predecessor existence

The reachability problem mentioned above addresses how a contagion process progresses through a network over time. Another problem addresses the question of finding a configuration that can occur immediately before the current configuration. A formulation of this predecessor existence problem is the following: Given a SyGDS $\mathcal{S}$ and a configuration $\mathcal{C}$, find a configuration $\mathcal{C}'$ such that $\mathcal{C}'$ is a (one-step) predecessor of $\mathcal{C}$ (if one exists). A generalization of this one-step predecessor problem is the k-step predecessor problem, where the goal is to find a configuration $\mathcal{C}'$ such that the SyGDS reaches $\mathcal{C}$ from $\mathcal{C}'$ in exactly k time steps, for some $k \geq 2$. In diffusion processes, solutions to k-predecessor problems for appropriate values of k help in understanding how the process may have spread through a population.

Barrett et al. [54] present a comprehensive study of the predecessor existence problem and its variants (e.g., determining whether a configuration has a unique predecessor, counting the number of predecessors) for discrete dynamical systems. They consider the problem for various classes of graphs and local transition functions. For example, they show that the predecessor existence problem is NP-complete even when the underlying graph of the SyGDS is a grid and each local transition function is symmetric. Further, they show that when the underlying graph is treewidth bounded[4] and each local transition function is r-symmetric for some fixed integer r, the predecessor existence problem and its counting version can be solved efficiently. They also present an extension of this algorithm to the case where predecessor configurations must satisfy additional constraints (e.g., a configuration in which at least α nodes are in state 1, for a given integer α). Kuhlman et al. [50] show that the predecessor existence problem remains NP-complete for SyGDSs with bi-threshold functions even when the maximum node degree of the underlying graph is 3; they also show that the problem can be solved efficiently when each node has a degree of at most 2.

4.2.3 Fixed-point existence

Recall that a fixed point of a SyGDS is a configuration which is its own successor. When a SyGDS is used to model a diffusion process, fixed points represent situations in which no additional infections can occur. This motivates the problem of determining whether a given SyGDS has a fixed point. Formally, given a SyGDS $\mathcal{S}$, the goal of the fixed-point existence problem is to determine whether $\mathcal{S}$ has a fixed point. Note that given a SyGDS $\mathcal{S}$ and a

[4] For definitions related to treewidth, we refer the reader to [63].

configuration $\mathcal{C}$, determining if $\mathcal{C}$ is a fixed point is trivial. The fixed-point existence problem has been studied by several groups of researchers [64–66]. Barrett et al. [64] showed that the problem is NP-complete in general, but efficiently solvable for several classes of local transition functions (e.g., monotone functions[5]). Other researchers (e.g., [65, 66]) have established complexity results for the problem of counting the number of fixed points for various classes of local functions.

4.2.4 *Analysis problems for more general phase space properties*

As mentioned earlier, each fixed point of a SyGDS $\mathcal{S}$ is a self-loop (i.e., a directed cycle with one edge) in the phase space of $\mathcal{S}$. In some biological applications [67], it is useful to determine whether the phase space contains longer cycles. An approach for studying the complexity aspects of this and more general properties of SyGDSs (e.g., the phase space has two node-disjoint simple paths each with seven nodes) has been proposed in [53]. This approach encodes phase space properties as appropriate graph predicates and develops a theoretic framework for analyzing these predicates. A number of known complexity results and efficient algorithms for testing phase space properties of SyGDSs follow as special cases from this general framework.

4.2.5 *Dynamical systems with nested canalyzing local functions*

Two special classes of Boolean functions, called canalyzing and nested canalyzing functions, were proposed and studied in [21, 68, 69] to model stability in gene regulatory networks. The complexity of analysis problems for SyGDSs in which each local function is a nested canalyzing function (NCF) has been studied in [39]. This reference shows that the reachability problem for SyGDSs with NCF local functions remains PSPACE-complete, while predecessor existence and fixed-point existence remain NP-complete. Thus, restricting the local functions to NCFs does not make these analysis problems computationally easier for SyGDSs.

For the reader's convenience, definitions of the analysis problems discussed above and some references which mention applications of the problems to bio-social systems are given in Table 1. For simplicity, we define only the basic versions of the problems in the table. The references cited in this section discuss several variants of the basic analysis problems.

[5] A Boolean function is monotone if does not change from 1 to 0 when one more of the inputs is changed from 0 to 1. For example, every k-threshold function (for any integer $k \geq 0$) is monotone.

4.3 Analysis problems under the SSyGDS model

Algorithmic aspects of analysis problems under the SSyGDS model (i.e., the SyGDS model with stochastic local transition functions) have not received as much attention in the literature as the deterministic SyGDS model. We will now summarize the known results for analysis problems under the SSyGDS model.

4.3.1 *Reachability and predecessor existence*

The work reported in [51, 52] is motivated by applications in epidemiology, and the focus is on problems related to reachability and predecessor existence for SSyGDSs. Since transitions in SSyGDS are stochastic, the reachability (decision) problem is reformulated as follows: Given a SSyGDS$\mathcal{S}$, two configurations $\mathcal{C}$ and $\mathcal{C}'$, and a probability value p, determine whether $\mathcal{S}$ starting from $\mathcal{C}$ reaches $\mathcal{C}'$ with a probability of at least p. Likewise, the definition of the predecessor existence problem is as follows: given a SSyGDS $\mathcal{S}$, a configuration $\mathcal{C}$ and a probability value p, is there a configuration $\mathcal{C}'$ such that the one-step transition probability from $\mathcal{C}'$ to $\mathcal{C}$ is at least p?

When $\mathcal{C}$ represents the initial onset of a disease and $\mathcal{C}'$ represents an undesirable situation (e.g., with a large percentage of infected population), the goal of the reachability problem is to determine whether an undesirable state is likely to occur in the absence of any effort to contain the disease. As mentioned earlier, the predecessor existence and its k-step generalization are helpful in understanding the progression of an epidemic.

It is shown in [52] that the reachability problem for SSyGDSs is hard for the complexity class RSPACE(n), which is the class of problems that can be solved by a probabilistic Turing machine which uses space bounded by a linear function of the input size. As pointed out in [52], under standard assumptions in complexity theory, reachability problems for SSyGDSs are likely to be computationally more difficult than those for (deterministic) SyGDSs. It is also observed that the NP-hardness of the predecessor problem for SSyGDSs follows from the corresponding result for SyGDSs. However, the problem is shown to be efficiently solvable when the following conditions hold: (1) the underlying graph is treewidth bounded, (2) the local stochastic functions are r-symmetric for some fixed integer r and (3) the number of distinct probability values used in all the local transition functions is bounded. Barrett et al. [52] also address other problems related to reachability. They point out that the problem of determining the most likely successor of a given configuration of a SSyGDS can be solved efficiently, while computing the two-step transition probability from a configuration $\mathcal{C}_1$

Table 1 A list of analysis problems for GDS that arise in practice while studying bio-social systems

Problem	Definition	References
Reachability	Does a given SyGDS starting from $\mathcal{C}$ reach $\mathcal{C}'$?	[38, 39, 70, 71]
Predecessor existence	Does the configuration $\mathcal{C}$ of a given SyGDS have a predecessor? How many predecessors does $\mathcal{C}$ have? Does $\mathcal{C}$ have two distinct predecessors?	[54, 72]
Fixed-point existence	Does a given SyGDS have a fixed point? How many fixed points does a given SyGDS have?	[64]
Other properties	Ex: Does the phase space of a given SyGDS have a cycle of length ≥ 3?	[53, 67]

to a configuration $\mathcal{C}_2$ is computationally intractable (technically, #P-complete).

5 Scalable simulations of bio-social systems based on the GDS formalism

In this section, we first relate the GDS framework to simulation. Then we provide a motivation for simulation in terms of the analysis problems of Sect. 4. Thereafter, we present three illustrative examples of simulation that use the GDS framework. We provide additional examples of simulation in the bio-social realm and note parallel developments in discrete dynamical system theory and simulation. We conclude with a brief mention of other modeling approaches.

5.1 Aspects of GDS-based simulation

Simulation is defined here as the process of executing software to mimic or replicate the behavior of a system. Our focus is on systems of interacting agents or entities, where entities can be any combination of humans, animals, insects, plants and inanimate objects: any object that can act or be acted on. We represent this system of agents as a (time-varying) graph, where nodes represent agents and edges represent pairwise interactions. We note, by comparison, that there can be many entities (e.g., humans) in a system, but they do not have to interact (see, e.g., [73]).

Here, we confine ourselves to the sub-topic of simulation that uses implementations of algorithms that are based (at least in part) on the theoretic foundations presented in Sects. 1 and 3. The central steps in building a simulation system, then, are as follows: (1) formally specify the system components, namely the agents and their interactions; (2) using the GDS framework, specify mathematically the local functions (i.e., behaviors) of the agents in the system; (3) formalize the dynamics (e.g., behaviors and interactions) using algorithms; and (4) build and verify the software system that implements the algorithms used to simulate the system under study. Many agent-based models (ABMs) and agent-based modeling and simulation (ABMS) systems can be represented within the GDS framework; see, e.g., [74, 75]. Several reviews of ABMs and software modeling systems have appeared in the literature [74–77].

5.2 Simulations as approximate representations of dynamics of systems

There is a close connection between each of the GDS formalisms of Sect. 3 and the analysis problems of Sect. 4, and simulation. First, the ideal outcome from evaluating a GDS is that the complete dynamical behavior of the system is computed. From Sect. 3, this means computing the phase space of the dynamical system. But recall that the phase space is exponential in the number of nodes of the graph (i.e., the population). This means that for dynamical systems of hundreds, thousands or billions of nodes (i.e., interacting entities), computing the phase space is not possible given the current state of computing technology. Second, the great majority of the analysis problems in Sect. 4 are computationally intractable. Thus, solving these problems for large GDSs is also not currently possible. This is why there are many works on specializing computationally intractable problems along one or more dimensions to make them computationally efficient (e.g., [38, 48, 53, 54, 62, 64, 65]) and why there are many works that devise heuristics to solve intractable dynamical system problems (e.g., [46, 78–81]).

Both of these issues are addressed, at least in part, by computing forward trajectories, and this is where simulations play a vital role. Rather than generating the entire phase space (i.e., all system transitions), a single simulation starts with a prescribed state of the system (that is part of the "initial conditions") and runs the dynamics of the system forward for a prescribed number of steps or for some specified duration. This sequence of computed successive system configurations—a forward trajectory—is part of the phase space. By running multiple simulations with different initial conditions (or the same initial conditions to evaluate the effects of stochasticity), one seeks to generate a portion of the phase space that is relevant to the particular problem of interest.

5.3 Three illustrative examples

We provide three examples of simulation in biological and social systems. The first example explores dynamics on a gene regulatory network in a plant [26, 82]. While the simulation and analysis are very close to the mathematical setup in Sect. 3, there is a direct connection to experimental observations: The model provides the first explanation of the importance of a particular hormone for plant development. The second example is a simulation system for inflammatory and regulatory immune pathways governed by cell interactions in the human gut [13, 83, 84]—this utilizes a much more complex set of local functions than does the first example. Our final example is a stochastic model of epidemic spread which is used extensively in public health analysis.

These examples have been chosen to demonstrate the range in utility of a GDS-based simulation approach for systems of various sizes. In the first example, there are 12 genes (agents) in the model. The second example has 10^7 cells (agents), a six orders of magnitude increase in problem size. The third example can have 10^9 agents, e.g., when used to model epidemic spread in the continental US.

5.3.1 Example 1: gene regulatory networks

Boolean networks were first used to study regulatory networks in [21, 85]. Since that time, a host of works on regulatory networks has been published, evolving from networks represented as regular graphs to current considerations of wirings in actual networks, to various forms of local transition functions, to different update schemes [22, 28, 82, 86, 87]. Here we highlight the work of [82].

In [82], the regulatory system is modeled as a GDS, described as follows. The regulatory network $G(V, E)$ represents a set V of genes and a set E of interactions. See Fig. 4. A gene is either activated (state 1) or inactivated (state 0), so that the state set—the possible states of a node/gene—is $K = \{0, 1\}$. Each of the G-local transition functions f_i is a generalized threshold function given by

$$f_i(x) = H\left(\left(\sum_{j\in n[i]}(w_{j,i}x_j)\right) - k_i\right), \tag{8}$$

where H is the Heaviside function $H : \mathbb{R} \longrightarrow \mathbb{R}$ given by $H(x) = 1$ if $x > 0$ and $H(x) = 0$ if $x \leq 0$. In Eq. (8), k_i is the threshold of node i, and $w_{j,i}$ is the weight assigned to the directed edge from j to i, the latter capturing the influence of node j on i. The edge weights and vertex thresholds may be positive or negative. A negative edge weight $w_{j,i}$ means that j tries to inhibit the transition of i.

There are multiple dimensions to this work. First, observing that the states of several nodes become constant (i.e., 0) after a few time steps, Demongeot et al. [82] show that the original GDS can be transformed into a system $\mathcal{F}'$ for which the matrix of weights W is symmetric. For the symmetric case, they apply theorems guaranteeing that all limit cycles must be either fixed points or two-cycle. Second, they highlight six fixed points and seven two-cycle as attractors that are significant in terms of the genes of a particular plant species. Third, robustness is characterized numerically by attractors (i.e., limit cycles) and the attractor basins. (Note that the basin of an attractor is the set of system states that are forward asymptotic to the attractor.) Viewing phase space as a directed graph, the attractor basin is the set of states in the weakly connected component of the attractor; the attractor states are those in the strongly connected component. Clearly, phase space is partitioned into attractor basins. We denote the set of attractors by $\mathcal{A} = \{A_1, \ldots, A_m\}$ and the corresponding set of basins by $\mathcal{B} = \{B_1, \ldots, B_m\}$.

Robustness is assessed in terms of perturbations of the vertex states in regulatory networks (that is, changing a gene's state from 0 to 1 or from 1 to 0). At issue is whether a change of a single vertex's state will cause the newly generated state to move to a different attractor basin. That is, for a configuration x in basin B, will the perturbation x' by a single vertex state of x belong to $B' \neq B$? Depending on the system, the states of two or more vertices may have to be perturbed in order for a system state to map to a different attractor basin. Then, a question is to determine, for a particular system state and target attractor basin, the minimum number of vertices whose state must be changed to transform the system state into a new system state in the target attractor basin. This number is called the Hamming distance. An overarching theme of these questions is the number of perturbations required to alter the long-term dynamics (i.e., limit cycles) of a system. The robustness of a system increases as the number of perturbations required

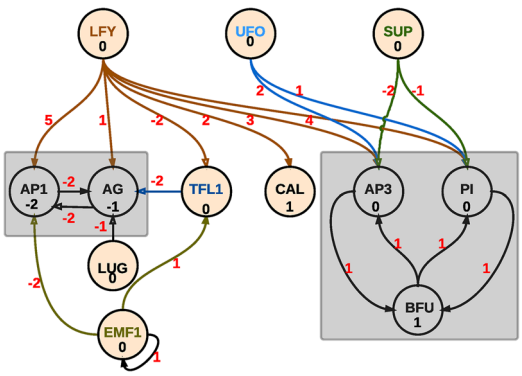

Fig. 4 Gene regulatory network of Example 1, reproduced from Figure 3 of [82]. The two strongly connected components are highlighted in gray. Genes are shown as circles. A gene's threshold is depicted within its circle. Edge weights for the GDS are shown in red (color figure online)

to change the attractor basin increases. These types of issues are explored in [82], where not only the number of gene state changes is studied, but also the particular genes whose states are changed are studied in the context of a particular plant species. Other studies of robustness include [22, 88, 89].

In [82], the authors also consider stability properties with respect to the update mode. One may choose to apply the vertex functions in Eq. (8) synchronously, sequentially, block sequentially or possibly in some random order. Understanding attractor structure as a function of update mode gives insight into model stability and quite possibly the underlying gene regulatory network.

Here we give an illustration of how one may assess this type of stability under all possible sequential update orders. In this case, an order $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ of the nodes/genes specifies the order in which the functions f_i are applied. Using different orders, one would possibly expect different attractor structures. For the graph G in Fig. 4 with $n = 12$ vertices, there are $12! = 479,001,600$ different update orders, and potentially this many different attractor structures. Using the theory of κ-equivalence from [90, 91], one may show that this number is in fact a lot smaller. For this, we will say that two systems F_π and F'_π are cycle equivalent if their multi-sets of limit cycle sizes coincide. (For a GDS map F with one three-cycle, two two-cycle and three fixed points, its multi-set of limit cycle sizes is $\{1(3), 2(2), 3(1)\}$.) The measure $\kappa(G)$ can be evaluated as $T_G(1,0)$ where T_G is the Tutte polynomial of G [92]. Applying this to the network in Fig. 4 we obtain $\kappa(G) = 210$. In other words, without looking at the form of the functions $\mathcal{F}$, we can immediately conclude that one may at most observe 210 distinct attractor structures when varying the update order. However, and as shown in [82], the long-term behavior of the system is effectively modeled by the subgraph contained in the gray boxes of Fig. 4, and the fact that the state of the gene labeled as UFO is fixed at its initial value. Thus, for each choice of $x_{\mathrm{UFO}} \in \{0, 1\}$, the limit cycle structure of the gene regulatory network is governed by the disjoint sub-networks indicated in the gray boxes. For this graph, one can show that the κ-value is 1. In other words, for each choice of x_{UFO} at $t = 0$, there is only one possible limit cycle structure, and *one may observe at most two distinct limit cycle structures under sequential update modes.* Moreover, from the analysis in [82], we know that fixed points and two-cycle are the only possibilities for limit cycles in these two cases.

5.3.2 Example 2: mammalian immune networks

Graphical dynamics systems and associated agent-based models have recently been used to model numerous processes that underlie host–pathogen interactions and the immune system. A pathogen here can be thought of as any infectious agent that can lead to the illness of a host (pathogenesis). The mammalian intestinal tract constitutes the largest component of the immune system. The mammalian gut comprises non-hemopoietic cells (epithelia, Paneth cells, goblet cells), hemopoietic cells (macrophages, dendritic cells, T cells) and a large community of microbes (now called the microbiota). Enteric pathogenesis in the gastrointestinal (GI) tract is often caused by ingestion of microbes in food and water. Upon microbe entry, immune cells in the GI tract mount an immuno-inflammatory response that eliminates the microbe, but may also cause tissue damage. This collateral damage is often the basis of disease pathogenesis.

As the GI tract is constantly exposed to foreign antigens, most of which are innocuous, this inherent inflammatory response must be regulated so that the system does not remain in a constant state of tissue-damaging hyper-inflammation. Immune regulation is carried out by the *regulatory* (or anti-inflammatory) immune response triggered by factors such as host tissue damage or commensal gut microflora. The prevailing understanding of the gut mucosa is one in which immune cells are in a dynamic balance (homeostasis) between regulatory and inflammatory responses, with regulatory phenotypes generally predominating [93, 94].

An understanding of the role of individual components of these immune pathways is needed to elucidate the mechanisms that lead to microbial persistence and severity of symptoms. This in turn can lead to improved treatment and prevention against pathogenic strains of *Escherichia coli*, *Clostridium difficile* and *Helicobacter pylori* for example.

Over the last two decades, several ABMs have been developed to study various immune processes. This includes: Agent-based Artificial IS (AbAIS) [95], CAFISS [96], ImmSim [97], ImmSim3 [98], C-ImmSim [99], ParImm [99], ImmunoGrid [100], Rhapsody [101, 102], SIS [103], SIMMUNE [104], NFSim [105], BIS [106] and the work reported in [107]. The GDS framework and its extensions provide a natural underlying mathematical abstraction for these ABMs. Comprehensive surveys of ABMs for immune processes can be found in [108–110]. See [12, 83] for a detailed account of these ABMs and their relative strengths.

As part of ongoing work, we have developed *ENteric Immunity SImulator* (ENISI), a scalable high-performance agent-based modeling environment to study the inflammatory and regulatory immune pathways initiated by microbe-immune cell interactions in the gut [12, 13]. ENISI is an interaction-based model where individual cells are modeled, along with their movement though different

tissues, and the probabilistic outcomes of cell–cell interaction.

The ENISI modeling environment is unique in its scope and approach. The modeling environment is designed to specifically represent regulatory mechanisms of both adaptive and innate immunity, multi-location migration of cells, and cross talk between antigen presenting cells and T cells. This is done by explicitly representing each participating cell of the immune pathway. This facilitates mapping of model parameter specifications and predictions to laboratory techniques which manipulate specific cell populations. In other words, the model is mechanistic and represents cells individually.

ENISI has the ability to simulate 10^7 or more individual cells. With ENISI, mucosal immunologists can test and generate hypotheses for enteric disease pathology and propose interventions through experimental infection of an in silico gut. This information can then be used to better understand immunological mechanisms and to generate novel treatment strategies that can be tested in the laboratory using mouse and pig models of infection as well as human clinical experimentation.

The ENISI modeling environment has already been illustrated by developing (1) a in silico model and dynamic simulation of H. pylori and (2) a simulation of dysentery resulting from B. hyodysenteriae infection so as to identify aspects of the host immune pathways that lead to continued inflammation-induced tissue damage even after pathogen elimination.

GDSs (and their extension to time-varying systems) provide a natural mathematical abstraction of the immune system as well as the computer simulation. The cells and the microbial community can be thought of as nodes. The network is dynamic in such a setting; vertices as well as edges come and go during the course of the system evolution. The cell/bacterial movement, their birth/death and their phenotypic change form the basis of the time-varying system. Cells and bacteria interact with each other via signaling pathways; see [12, 83] for further details.

A state chart-like formalism was used to model each cell type, including each cell's individualized state transitions, the other cell types that are required in interactions that cause the state transitions, and transition probabilities. The cell types are: epithelial cell, inflammatory bacteria, tolerogenic bacteria, commensal bacteria, sampling dendritic cell, dendritic cell, macrophage, conventional CD4+ T cell and natural T-regulatory cell (nTreg). As an example, the state chart-like formal specification for tolerogenic bacteria cell type is provided in Fig. 5. In this figure, ovals represent states of the cell type, the solid arrow represents a transition that occurs in the next time step after transition to the state represented by the arrow's tail, and dashed arrows represent single-cell interaction-initiated state transitions where the edge labels characterize the initiator of the transition.

Of particular note here is that the cells and their interactions can be represented by a time-varying graph G. Thus, descriptions such as the one in Fig. 5 can be converted to local transition functions that capture the interactions of different cells i. Consequently, an expression can be written for the local function of each node of the form $s_i^{(t+1)} = f_i(s_{j_1}^t, s_{j_2}^t, \ldots, s_{j_k}^t)$, where $s_i^{(t+1)}$ is the next state of node (cell, agent) i, and $j_\ell \in J$ are the k elements of the closed neighborhood J of node i. (This form of local function was described in Sect. 3.) In ENISI, the synchronous update method is used in simulations.

5.3.3 *Example 3: networked epidemiology*

Controlling epidemics caused by infectious diseases remains an important societal challenge, despite significant medical advances. Understanding and forecasting the dynamics of the disease, inferring the source and developing interventions to control the spread are some of the fundamental questions studied by epidemiologists and public health policy planners. Since there are limited data available for any outbreak, especially in the early stages, mathematical and computational models play a key role in studying epidemic spread. These are typically considered at the level of an individual, with a simple model capturing the spread of the disease from one individual to those coming in contact with him/her. Many researchers have used ABMs to model the spread of epidemics; see, e.g., [7, 9, 111–114]. Details incorporated in different models vary widely. We refer the reader to [56, 115] for a discussion on different models of epidemic spread.

One of the most commonly used SSyGDS models is the SIR model of epidemic spread. In the basic SIR model, each node is in one of the following states: Susceptible ($\mathbb{S}$), Infected ($\mathbb{I}$) or Recovered/Removed ($\mathbb{R}$) states, i.e., $K = \{\mathbb{S}, \mathbb{I}, \mathbb{R}\}$. A node j in state $\mathbb{I}$ spreads the infection to each neighbor i in state $\mathbb{S}$ with probability p, independent

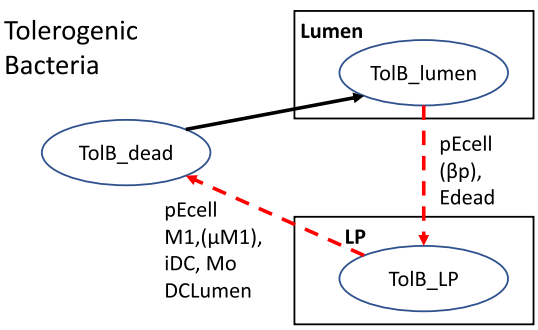

Fig. 5 One of nine state chart-like formal specifications used to represent cell types (cells are agents) in ABMs of immunological processes in the gut. Adapted from [12]

of the other neighbors of i. A node in state $\mathbb{I}$ recovers after a prescribed duration t_{inf} in the infectious state, i.e., switches to the R state. Formally, for the $\mathbb{S}$ to $\mathbb{I}$ transition, the local function for node i is given by:

$$f_i(\mathbb{S}, \mathbb{I}) = 1 - (1 - p)^k,$$

where k is the number of infected neighbors of i, and $f_i(\mathbb{I}, \mathbb{R}) = 1$ after $t_{\text{inf}} = 1$ units of infectivity. Most often, the update method $\mathcal{U}$ is synchronous. Many other variants of this basic model are used commonly, e.g., SI, SIS and SIRS, where the letters indicate the state transitions. For example, in the SI model, an infected node stays infected, whereas in the SIS model an infected node switches to state $\mathbb{S}$ after a period of time.

If the network $G(V, E)$ does not change over time and there are no new externally induced infections, the GDSs corresponding to all these models have fixed points in which there are no infected nodes. Computing the values of parameters related to epidemic analysis can be mapped to computations that use properties of the phase space of the corresponding GDS (similar to the analysis questions discussed in Sect. 4.2). Examples include the following: (1) the average outbreak size is equal to the average number of nodes in state $\mathbb{R}$ in the fixed points reachable from a given initial configuration and (2) the (average) duration of an epidemic is the (average) length of a transient to a fixed point with no infected nodes. An important topic in epidemiology is *forecasting*: Here, the objective is to determine the future dynamics, given the current configuration. An example of a forecasting problem is the following: Given the current configuration, will the system reach a fixed point with a large number of infected nodes? The dynamics of epidemic spread are very complex in heterogeneous real-world networks and cannot be computed analytically. This makes efficient simulations necessary. Designing highly scalable epidemic simulation tools which can handle SIR models at a national or global scale (with billions of agents) is a fundamental research problem [15, 116].

5.3.4 Additional examples of bio-social systems

Beyond the applications described above, a number of other biological and social systems can be fruitfully analyzed using computer simulations that implement ABMs based on GDSs. These include: depression [117], incarceration [118], segregation [119], invasive species [120, 121], drug trials and predictions of drug effects [76], reproductive growth of bacteria [122], forest management and fire protection [123], cell biology [124], computer viruses/malware [125], finance (works on housing, mortgages, banking systems and credit risk appear in [126]), disasters [127, 128] and ecology [129, 130]. See [70, 71, 74, 131, 132] for additional applications.

Social science modeling continues to make ever greater use of ABM. Reviews of social modeling include [133–135]. In particular, Bianchi and Squazzoni [135] break down ABM work by different topics such as cooperation and social norms, reciprocity, reputation, punishment, conventions, trust, social influence, culture and social inequality.

These and other references illustrate a development in ABM that parallels the one in dynamical systems. That is, as explained in Sect. 2.2, dynamical system formulations have progressed from cellular automata, to Boolean networks, and to GDS, among others. The interaction structures have evolved from lattice (regular) patterns, to random patterns, to structured patterns of interactions based on data. The local functions have also grown in sophistication within each of these dynamical system constructs, over time. We observe the same types of evolution in ABM (in social systems). Early ABM works of segregation [119] and user choices [136], for example, are cellular automata formulations wherein agents are located in a grid (or lattice) and interact with their geo-spatial nearest neighbors, and possibly change state, based on some rule (i.e., local function). Later, in [137], a grid system is still employed to measure civil disobedience, but the local functions are more complicated. Current modeling efforts, in the three examples above, and in the other references cited in this subsection, illustrate increasing complexity in terms of population representation, environment, state update scheme and agent local functions. Clearly, this evolution is not without exceptions, but there appears to be a general trend of increasing complexity in discrete dynamical systems and ABM, along similar dimensions.

5.3.5 Other modeling approaches

We discussed ABM in the context of GDSs above. In contrast, analytical approaches have also been developed (see, e.g., [138, 139]). Intermediate solutions in terms of granularity—between analytical solutions and fine-grained ABMs—include ODE and compartmental models [140–146]. Various kinds of modeling of biological systems are presented in [147]. Multi-scale aspects of biological modeling are discussed in [148].

6 Decentralized dynamics: game theoretic models

The GDS model, as defined in Sect. 3, works with a fixed local function at each node. Furthermore, the analysis problems discussed earlier consider GDSs as computing

devices; thus, we are interested in estimating the computational resources needed to determine phase space properties. In many other settings, GDSs can be viewed as graphical models of games. From this point of view, we can think of each node as an agent that interacts with its neighbors to optimize individual utility. The local function at a node v can then be viewed as a strategy adopted by the player represented by v. We illustrate this in the specific context of vaccination decisions for epidemic control. Recall the SI/SIS/SIR models of epidemic spread discussed in Sect. 5. An individual v can protect oneself by getting vaccinated (which can be modeled by the function $f_v = 0$); however, this involves a certain cost C_v, which might be the cost of the vaccine, or the side effects. On the other hand, if enough nodes in the network are vaccinated, the probability that v gets infected would be low, in which case node v has no incentive to get vaccinated. This is a natural game theoretic setting and has been studied extensively using noncooperative game models.

Vaccination games were first introduced by Bauch and Earn [149], who model vaccination decisions as a strategic noncooperative game. This was first extended to a network setting by Aspnes et al. [150] for the SI model, and we use their terminology to describe the basic formulation. We assume that the infection starts from a random source $s \in V$ (which may be vaccinated, in which case the disease dies out immediately). We assume the vaccine has 100% efficacy, so that if a node is vaccinated, it is protected from the infection (and also does not participate in the spread of the infection). Therefore, the vaccination by node v can be modeled by removing v from G. We also restrict ourselves to *pure* decisions, where the nodes only have two choices: (1) get vaccinated (denoted by $a_v = 1$), in which case it incurs a fixed cost C_v; (2) not get vaccinated (denoted by $a_v = 0$): In this case, the node incurs zero cost if it does not get infected, but incurs cost L_v if it gets infected. The expected cost of infection is $Pr[v \text{ gets infected}] \cdot L_v$. Note that the probability v gets infected depends on the strategies of all the nodes. We denote $\mathbf{a} = (a_v : v \in V)$ as the strategy vector, and $G_{\mathbf{a}}$ as the graph obtained by removing the subset $S = \{v : a_v = 1\}$. Putting these together, the expected cost for node v, given a strategy vector $\mathbf{a}$, can be expressed as

$$cost_v(\mathbf{a}) = a_v C_v + (1 - a_v) L_v Pr[v \text{ gets infected in } G_{\mathbf{a}}] .$$

The main solution concept in noncooperative game theory is a *Nash equilibrium* (NE): A strategy $\mathbf{a}$ is said to be a NE if no node v can decrease its cost by unilaterally switching its strategy a_v. See [56, 151] for an introduction to game theory. Aspnes et al. [150] characterize the structure of Nash equilibria (NE) in epidemic games on networks and the complexity of computing them. They show that a pure NE always exists (in which $a_v \in \{0, 1\}$), and can be computed using an iterative *best response* method: Each node v updates its strategy a_v if that results in lower cost. Let $cost(\mathbf{a}) = \sum_v cost_v(\mathbf{a})$; a strategy vector $\mathbf{a}$ that minimizes the cost is referred to as a social optimum. A quantity of interest in noncooperative games is the Price of Anarchy (PoA), which is the maximum ratio of the cost of a NE to the cost of a social optimum. Aspnes et al. show that for a game involving n players, the PoA can be $\Theta(n)$ in general, which implies that decentralized decisions can lead to highly sub-optimal outcomes for the system as a whole.

7 Final remarks

We discussed various models of graphical dynamical systems and pointed out how they are useful in studying several problems that arise in the context of bio-social systems. Due to space reasons, our focus was on a few basic analysis problems. Many other kinds of phase space analysis problems arise in various bio-social systems in the literature; their mathematical and computational study is an important topic of research. We also considered only a few classes of local functions. Many other classes of functions are likely to be useful in modeling various aspects of bio-social systems. We close by discussing the limitations of the GDS framework, highlighting some open questions and providing a brief discussion on other research topics which have been studied in the GDS context.

7.1 Limitations

Despite being very useful as a modeling framework, the GDS approach has some limitations. As stated in Sect. 5.2, one limitation is that the size of the phase space of a GDS is exponential in the number of nodes, and hence, methods such as simulation must be used for systems of large size to approximate the dynamics. Also, under standard assumptions in computational complexity [152], many analysis problems cannot be solved efficiently. However, this difficulty also opens up a research direction, namely the development of efficient heuristics for the analysis problems. For biological and social systems, determining the exact local functions can be difficult and may require significant numbers of data points. However, this difficulty is not unique to the GDS formalism; producing accurate models of agent behavior is difficult in many contexts.

7.2 Some open questions and research directions

For many analysis problems which are known to be computationally intractable in general, it is of interest to study whether special cases can be solved efficiently. Some

results along this direction were mentioned in Sect. 4, such as for the reachability, predecessor existence, the fixed-point existence problems and analysis of more general phase space properties. As discussed, many of these problems are computationally very hard, in general, but can be solved efficiently in some cases if the graphs have bounded treewidth. The applicability of these results to real bio-social systems is of considerable interest. In particular, an important open direction is to determine whether such problems can be solved efficiently for SyGDS instances whose underlying networks arise in real bio-social systems. Such networks have many special properties, e.g., high clustering and low diameter (see, e.g., [56]); exploiting these properties to develop efficient algorithms for the various analysis problems is an important research topic. A first step in this direction might be to develop better characterizations of phase space properties and their sensitivity in SyGDS instances on such networks. Another important direction is to develop "approximate" notions of these problems, as a way to address the computational hardness. For instance, in the predecessor existence problem, one might ask whether there is a configuration $\mathcal{C}'$ such that its successor is "close" to a given configuration $\mathcal{C}$. Such notions are likely to make more bio-social instances amenable to analysis; they can also help in incorporating uncertainty.

We also discussed the use of simulations to understand the behavior of bio-social systems modeled as graph dynamical systems. This topic also offers many interesting research issues. For example, Adiga et al. [153] discuss over a dozen future research directions for the modeling of epidemics in tropical climates; each of them can be cast as a specific analysis problem involving one or more components (e.g., network, local functions) of a GDS. Exploiting the features of practical bio-social systems so that simulations can be carried out efficiently on commonly available computing platforms is an additional direction for future work.

7.3 Other research topics

There are many additional topics beyond those covered here—this includes inference, control and optimization problems. For example, in epidemiology and more broadly the spread of contagions over networks, one is often interested in accentuating [78, 154–156] or thwarting [81, 143, 157–159] the spread of contagions. Maximizing the spread of information (e.g., through judicious seeding) is an example of the former, while efficient allocation of pharmaceutical interventions (e.g., antivirals or vaccines) is an example of the latter. A very important open topic is to design effective interventions (e.g., determining whom to vaccinate), given a limited set of resources. These are challenging optimization problems, even when the intervention is implemented in a centralized manner. In practice, game theoretic issues need to be taken into consideration, as discussed in Sect. 6.

The GDS formalism allows researchers to use established mathematical and computational machinery in addressing these issues for bio-social systems. Inference problems cut across all the areas just mentioned. As an example, one may want to determine a system's GDS representation (assuming that one exists) through observations of the system's transitions (e.g., a time series) [160]. Other examples arise in epidemiology, wherein one wants to determine the index node (also called "patient zero"), given some observations regarding the state of the epidemic after a certain time.

Acknowledgements We thank members of the Network Dynamics and Simulation Science Laboratory (NDSSL) for their comments and input. Specifically, we thank Chris Barrett, Christian Reidys, Daniel Rosenkrantz and Richard Stearns for their collaboration on several papers discussed in this article. We thank the computer systems administrators and managers at the Biocomplexity Institute of Virginia Tech for their help in this and many other works: Dominik Borkowski, William Miles Gentry, Jeremy Johnson, William Marmagas, Douglas McMaster, Kevin Shinpaugh and Robert Wills. This work has been partially supported by DTRA CNIMS (Contract HDTRA1-11-D-0016-0001), NSF BIG DATA Grant IIS-1633028, NSF DIBBS Grant ACI-1443054 and NSF EAGER Grant CMMI-1745207. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US Government.

References

1. Epstein, J.M.: Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press, Princeton (2007)
2. Barrett, C., Bisset, K., Eubank, S., Marathe, M., Kumar, V., Mortveit, H.: Modeling and simulation of large biological, information and socio-technical systems: an interaction based approach. In: Laubenbacher, R. (ed.) Modeling and Simulation of Biological Networks, pp. 101–147. American Mathematical Society (2007)
3. Barrett, C., Eubank, S., Marathe, M.: Modeling and simulation of large biological, information and socio-technical systems: an interaction based approach. In: Goldin, D., Smolka, S.A., Wegner, P. (eds.) Interactive Computation, pp. 353–392. Springer, Berlin (2006)
4. Kuhlman, C.J., Kumar, V.A., Marathe, M.V., Mortveit, H.S., Swarup, S., Tuli, G., Ravi, S., Rosenkrantz, D.J.: A general-purpose graph dynamical system modeling framework. In: Simulation Conference (WSC), Proceedings of the 2011 Winter, pp. 296–308. IEEE (2011)

5. Barrett, C., Beckman, R., Berkbigler, K., Bisset, K., Bush, B., Campbell, K., Eubank, S., Henson, K., Hurford, J., Kubicek, D., Marathe, M., Romero, P., Smith, J., Smith, L., Speckman, P., Stretz, P., Thayer, G., Eeckhout, E., Williams, M.: TRANSIMS: transportation analysis simulation system. Technical Report LA-UR-00-1725, LANL (2001)
6. Eubank, S.: Scalable, efficient epidemiological simulation. In: Proceedings of Symposium on Applied Computing, pp. 139–145 (2002)
7. Eubank, S., Guclu, H., Kumar, V.S.A., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N.: Modelling disease outbreaks in realistic urban social networks. Nature **429**, 180–184 (2004)
8. Bisset, K.R., Chen, J., Deodhar, S., Feng, X., Ma, Y., Marathe, M.V.: Indemics: an interactive high-performance computing framework for data intensive epidemic modeling. ACM Trans. Model. Comput. Simul. (TOMACS) **24**(1), 4:1–4:32 (2014). **(Special Issue on Simulation in Complex Service Systems)**
9. Barrett, C.L., Bisset, K.R., Eubank, S.G., Feng, X., Marathe, M.V.: Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, pp. 37:1–37:12 (2008)
10. Parikh, N., Swarup, S., Stretz, P.E., Rivers, C.M., Lewis, B.L., Marathe, M.V., Eubank, S.G., Barrett, C.L., Lum, K., Chungbaek, Y.: Modeling human behavior in the aftermath of a hypothetical improvised nuclear detonation. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 949–956. Saint Paul, MN, USA (2013)
11. Barrett, C., Bisset, K., Chandan, S., Chen, J., Chungbaek, Y., Eubank, S., Evrenosoğlu, Y., Lewis, B., Lum, K., Marathe, A., Marathe, M., Mortveit, H., Parikh, N., Phadke, A., Reed, J., Rivers, C., Saha, S., Stretz, P., Swarup, S., Thorp, J., Vullikanti, A., Xie, D.: Planning and response in the aftermath of a large crisis: an agent-based informatics framework. In: Pasupathy, R., Kim, S.H., Tolk, A., Hill, R., Kuhl, M.E. (eds.) Proceedings of the 2013 Winter Simulation Conference, pp. 1515–1526 (2013)
12. Bisset, K., Alam, M., Bassaganya-Riera, J., Carbo, A., Eubank, S., Hontecillas, R., Hoops, S., Mei, Y., Wendelsdorf, K., Xie, D., Yeom, J., Marathe, M.: High-performance interaction-based simulation of gut immunopathologies with enteric immunity simulator (ENISI). In: Proceedings of International Parallel and Distributed Processing Symposium (IPDPS), pp. 48–59 (2012)
13. Alam, M., Deng, X., Philipson, C., Bassaganya-Riera, J., Bisset, K., Carbo, A., Eubank, S., Hontecillas, R., Hoops, S., Mei, Y., Abedi, V., Marathe, M.: Sensitivity analysis of an ENteric Immunity SImulator (ENISI)-based model of immune responses to *Helicobacter pylori* infection. PLoS ONE **10**(9), e0136139 (2015)
14. Barrett, C., Eubank, S., Kumar, V.S.A., Marathe, M.: Understanding large-scale social and infrastructure networks: a simulation-based approach. SIAM News **37**(4), 1–5 (2004)
15. Yeom, J.S., Bhatele, A., Bisset, K.R., Bohm, E., Gupta, A., Kale, L.V., Marathe, M., Nikolopoulos, D.S., Schulz, M., Wesolowski, L.: Overcoming the scalability challenges of epidemic simulations on Blue Waters. In: Proceedings of the IEEE 28th International Parallel and Distributed Processing Symposium, pp. 755–764 (2014)
16. von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966). **(Edited and completed by Arthur W. Burks)**
17. Kari, J.: Theory of cellular automata: a survey. Theor. Comput. Sci. **334**, 3–33 (2005)
18. Ilachinski, A.: Cellular Automata: A Discrete Universe. World Scientific Publishing Company, Cambridge (2001)
19. Delorme, M., Mazoyer, J. (eds.): Cellular Automata—A Parallel Model, Mathematics and Its Applications, vol. 460. Kluwer Academic Publishers, Alphen aan den Rijn (1999)
20. Wolfram, S.: Theory and Applications of Cellular Automata, Advanced Series on Complex Systems, vol. 1. World Scientific Publishing Company, Singapore (1986)
21. Kauffman, S.: Metabolic stability and epigenesis in randomly constructed genetic nets. J. Theor. Biol. **22**(3), 437–467 (1969)
22. Ribeiro, A.S., Kauffman, S.A.: Noisy attractors and ergodic sets in models of gene regulatory networks. J. Theor. Biol. **247**(4), 743–755 (2007)
23. Goles, E., Martinez, S.: Neural and Automata Networks: Dynamical Behaviour and Applications. Kluwer Academic Publishers, Alphen aan den Rijn (1990)
24. Goles-Chacc, E., Fogelman-Soulie, F., Pellegrin, D.: Decreasing energy functions as a tool for studying threshold networks. Discrete Appl. Math. **12**, 261–277 (1985)
25. Goles, E., Olivos, J.: Periodic behavior in generalized threshold functions. Discrete Math. **30**, 187–189 (1980)
26. Ruz, G.A., Goles, E.: Reconstruction and update robustness of the mammalian cell cycle network. In: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pp. 397–403 (2012)
27. Gershenson, C.: Introduction to random Boolean networks (2004). arXiv:nlin.AO/040806v3-12Aug2004. Accessed Aug 2005
28. Shmulevich, I., Dougherty, E.R., Zhang, W.: From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. Proc. IEEE **90**(11), 1778–1792 (2002)
29. Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W.: Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics **18**(2), 261–274 (2002)
30. Shmulevich, I., Kauffman, S.A.: Activities and sensitivities in Boolean network models. Phys. Rev. Lett. **93**(4), 048701:1–048701:4 (2004)
31. Jarrah, A.S., Laubenbacher, R.: On the Algebraic Geometry of Polynomial Dynamical Systems, The IMA Volumes in Mathematics and Its Applications, vol. 149, pp. 109–123. Springer, New York (2009)
32. Jarrah, A.S., Raposa, B., Laubenbacher, R.: Nested canalyzing, unate cascade, and polynomial functions. Physica D **233**, 167–174 (2007)
33. Kaneko, K.: Pattern dynamics in spatiotemporal chaos. Physica D **34**, 1–41 (1989)
34. Golubitsky, M., Pivato, M., Stewart, I.: Interior symmetry and local bifurcations in coupled cell networks. Dyn. Syst. **19**(4), 389–407 (2004)
35. Nishikawa, T., Sun, J., Motter, A.E.: Sensitive dependence of optimal network dynamics on network structure. Phys. Rev. X **7**, 041044:1–041044:21 (2017)
36. Davidich, M., Bornholdt, S.: The transition from differential equations to Boolean networks: a case study in simplifying a regulatory network model. J. Theor. Biol. **255**(3), 269–277 (2008)
37. Veliz-Cuba, A., Stigler, B.: Boolean models can explain bistability in the lac Operon. J. Comput. Biol. **18**, 783–794 (2011)
38. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Complexity of reachability problems for finite discrete dynamical systems. J. Comput. Syst. Sci. **72**(8), 1317–1345 (2006)
39. Rosenkrantz, D.J., Marathe, M.V., Ravi, S.S., Stearns, R.E.: Testing phase space properties of synchronous dynamical systems with nested canalyzing local functions. In: Proceedings of 17th International Conference on Autonomous Agents and

Multiagent Systems (AAMAS), Stockholm, Sweden, pp. 1585–1594 (2018)
40. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Reachability problems for sequential dynamical systems with threshold functions. Theor. Comput. Sci. **295**(1–3), 41–64 (2003)
41. Mortveit, H.S., Reidys, C.M.: An Introduction to Sequential Dynamical Systems. Springer, Berlin (2007)
42. Abdelhamid, S.H.E., Kuhlman, C.J., Marathe, M.V., Mortveit, H.S., Ravi, S.S.: GDSCalc: a web-based application for discrete graph dynamical systems. PLoS ONE **10**(8), 24 (2015)
43. El Samad, H., Khammash, M., Petzold, L., Gillespie, D.: Stochastic modeling of gene regulatory networks. Int. J. Robust Nonlinear Control **15**, 691–711 (2005)
44. Eubank, S., Kumar, V.S.A., Marathe, M.V., Srinivasan, A., Wang, N.: Structure of social contact networks and their impact on epidemics. In: Abello, J.M., Cormode, G. (eds.) Discrete Methods in Epidemiology, vol. 70, pp. 179–200. American Mathematical Society, Providence, RI (2006)
45. Granovetter, M.: Threshold models of collective behavior. Am. J. Sociol. **83**(6), 1420–1443 (1978)
46. Kuhlman, C.J., Kumar, V.S.A., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J.: Inhibiting diffusion of complex contagions in social networks: theoretical and experimental results. Data Min. Knowl. Discov. **29**(2), 423–465 (2015)
47. Kuhlman, C.J., Kumar, V.S.A., Marathe, M.V., Swarup, S., Tuli, G., Ravi, S.S., Rosenkrantz, D.J.: Inhibiting the diffusion of contagions in bi-threshold systems: Analytical and experimental results. In: Complex Adaptive Systems: Energy, Information, and Intelligence, Papers from the 2011 AAAI Fall Symposium, Arlington, Virginia, pp. 91–100 (2011)
48. Kuhlman, C.J.: High performance computational social science modeling of networked populations. Ph.D. thesis, Computer Science Department, Virginia Tech, Blacksburg, VA, USA (2013)
49. Kuhlman, C.J., Kumar, V.S.A., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Swarup, S., Tuli, G.: A bi-threshold model of complex contagion and its application to the spread of smoking behavior. In: Proceedings of SNA-KDD Workshop, pp. 18.1–18.10 (2011)
50. Kuhlman, C.J., Marathe, M.V., Kumar, V.S.A., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Analysis problems for special classes of bi-threshold dynamical systems. In: Proceedings of Workshop on Multiagent Interaction Networks (MAIN 2013), held in conjunction with AAMAS, pp. 26–33 (2013)
51. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., Thakur, M.: Computational aspects of analyzing social network dynamics. In: IJCAI 2007, Proceedings of 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 Jan 2007, pp. 2268–2273 (2007)
52. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Modeling and analyzing social network dynamics using stochastic discrete graphical dynamical systems. Theor. Comput. Sci. **412**(30), 3932–3946 (2011)
53. Rosenkrantz, D.J., Marathe, M.V., Hunt III, H.B., Ravi, S.S., Stearns, R.E.: Analysis problems for graphical dynamical systems: a unified approach through graph predicates. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, 4–8 May 2015, pp. 1501–1509 (2015)
54. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., Thakur, M.: Predecessor existence problems for finite discrete dynamical systems. Theor. Comput. Sci. **386**(1–2), 3–37 (2007)
55. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Predecessor and permutation existence problems for sequential dynamical systems. In: DMCS, pp. 69–80 (2003)
56. Easley, D., Kleinberg, J.: Networks, Crowds and Markets: Reasoning About a Highly Connected World. Cambridge University Press, New York, NY (2010)
57. Dodds, P., Watts, D.: A generalized model of social and biological contagion. J. Theor. Biol. **232**(4), 587–604 (2005)
58. Centola, D., Macy, M.: Complex contagions and the weakness of long ties. Am. J. Sociol. **113**(3), 702–734 (2007)
59. Kleinberg, J.: Cascading behavior in networks: algorithmic and economic issues. In: Nissan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds.) Algorithmic Game Theory, pp. 613–632. Cambridge University Press, New York, NY (2007)
60. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Analysis problems for sequential dynamical systems and communicating state machines. In: Proceedings of MFCS, pp. 159–172 (2001)
61. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. W. H. Freeman and Co., San Francisco, CA (1979)
62. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: On special classes of sequential dynamical systems. Ann. Comb. **7**, 381–408 (2003)
63. Bodlaender, H.: Treewidth: algorithmic techniques and results. In: Proceedings of 22nd Symposium on Mathematical Foundations of Computer Science, pp. 29–36 (1997)
64. Barrett, C.L., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., Tosic, P.T.: Gardens of Eden and fixed points in sequential dynamical systems. In: Proceedings of International Conference on Discrete Models Combinatorics, Computation and Geometry (DM-CCG), pp. 95–110 (2001)
65. Kosub, S., Homan, C.M.: Dichotomy results for fixed point counting in Boolean dynamical systems. In: Proceedings of ICTCS, pp. 163–174 (2007)
66. Tosic, P.T.: On the complexity of enumerating possible dynamics of sparsely connected Boolean network automata with simple update rules. In: Automata 2010—16th International Workshop on CA and DCS, pp. 125–144 (2010)
67. Akutsu, T., Kosub, S., Melkman, A., Tamura, T.: Finding a periodic attractor of a Boolean network. IEEE/ACM Trans. Comput. Biol. Bioinform. **9**(5), 1410–1421 (2012)
68. Kauffman, S.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York, NY (1993)
69. Kauffman, S., Peterson, C., Samuelsson, B., Troein, C.: Random Boolean network models and the yeast transcriptional network. Proc. Natl. Acad. Sci. (PNAS) **100**(25), 14796–14799 (2003)
70. Bornholdt, S.: Boolean network models of cellular regulation: prospects and limitations. J. R. Soc. Interface **5**(Suppl 1), S85–S94 (2008)
71. Wang, R.S., Saadatpour, A., Albert, R.: Boolean modeling in systems biology: an overview of methodology and applications. Phys. Biol. **9**(5), 055001 (2012)
72. Laschov, D., Margaliot, M., Even, G.: Observability of Boolean networks: a graph-theoretic approach. Automatica **49**(8), 2351–2362 (2013)
73. Nguyen, C., Schlesinger, K.J., Carlson, J.M.: Data-driven models for individual and group decision making. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 852–859 (2017)
74. Macal, C., North, M.: Introductory tutorial: agent-based modeling and simulation. In: Proceedings of the 2014 Winter Simulation Conference, pp. 6–20 (2014)

75. Weimer, C.W., Miller, J.O., Hill, R.R.: Agent-based modeling: an introduction and primer. In: Proceedings of the 2016 Winter Simulation Conference, pp. 65–79 (2016)
76. An, G., Mi, Q., Dutta-Moscato, J., Vodovotz, Y.: Agent-based models in translational systems biology. Wiley Interdiscip. Rev. Syst. Biol. Med. **1**, 159–171 (2009)
77. Nikolai, C., Madey, G.: Tools of the trade: a survey of various agent based modeling platforms. J. Artif. Soc. Soc. Simul. **12**, 1–37 (2009)
78. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
79. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208 (2009)
80. Nguyen, N.P., Yan, G., Thai, M.T., Eidenbenz, S.: Containment of misinformation spread in online social networks. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 213–222 (2012)
81. Kuhlman, C.J., Tuli, G., Swarup, S., Marathe, M.V., Ravi, S.S.: Blocking simple and complex contagion by edge removal. In: 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 Dec 2013, pp. 399–408 (2013)
82. Demongeot, J., Goles, E., Morvan, M., Noual, M., Sene, S.: Attraction basins as gauges of robustness against boundary conditions in biological complex systems. PLoS ONE **5**, e11793-1–e11793-18 (2010)
83. Wendelsdorf, K., Alam, M., Bassaganya-Riera, J., Bisset, K., Eubank, S., Hontecillas, R., Hoops, S., Marathe, M.: Enteric immunity simulator: a tool for in silico study of gastroenteric infections. IEEE Trans. NanoBiosci. **11**(3), 273–288 (2012)
84. Wendelsdorf, K., Bassaganya-Riera, J., Bisset, K., Eubank, S., Hontecillas, R., Marathe, M.: Enteric immunity simulator: a tool for in silico study of gut immunopathologies. In: Proceedings of the IEEE International Conference Bioinformatics and Biomedicine, pp. 462–469 (2011)
85. Kauffman, S.: Homeostasis and differentiation in random genetic control networks. Nature **224**, 177–178 (1969)
86. Thomas, R.: Boolean formalisation of genetic control circuits. J. Theor. Biol. **42**(3), 563–585 (1973)
87. Goles, E., Salinas, L.: Comparison between parallel and serial dynamics of Boolean networks. Theor. Comput. Sci. **396**, 247–253 (2008)
88. Serra, R., Villani, M., Barbieri, A., Kauffman, S., Colacci, A.: On the dynamics of random Boolean networks subject to noise: attractors, ergodic sets and cell types. J. Theor. Biol. **265**, 185–193 (2010)
89. Luo, J.X., Turner, M.S.: Evolving sensitivity balances Boolean networks. PLoS ONE **7**, e36010-1–e36010-8 (2012)
90. Macauley, M., Mortveit, H.S.: On enumeration of conjugacy classes of Coxeter elements. Proc. Am. Math. Soc. **136**(12), 4157–4165 (2008)
91. Macauley, M., Mortveit, H.: Cycle equivalence of graph dynamical systems. Nonlinearity **22**(2), 421–436 (2009)
92. Welsh, D.: The Tutte polynomial. Random Struct. Algorithms **15**, 210–228 (1999)
93. Gordon, S., Taylor, P.R.: Monocyte and macrophage heterogeneity. Nat. Rev. Immunol. **5**(12), 953–64 (2005)
94. Iwasaki, A.: Mucosal dendritic cells. Ann. Rev. Immunol. **25**, 381–418 (2007)
95. Grilo, A., Caetano, A., Rosa, A.: Agent based artificial immune system. In: Proceedings of GECCO-01, vol. LBP pp. 145–151 (2001)
96. Tay, J.C., Jhavar, A.: CAFISS: A complex adaptive framework for immune system simulation. In: Proceedings of the 2005 ACM Symposium on Applied Computing, SAC '05, pp. 158–164. ACM, New York, NY (2005)
97. Celada, F., Seiden, P.E.: A computer model of cellular interactions in the immune system. Immunol. Today **13**(2), 56–62 (1992)
98. Castiglione, F., Duca, K., Jarrah, A., Laubenbacher, R., Hochberg, D., Thorley-Lawson, D.: Simulating Epstein-Barr virus infection with C-ImmSim. Bioinformatics **23**, 1371–1377 (2007)
99. Bernaschi, M., Castiglione, F.: Design and implementation of an immune system simulator. Comput. Biol. Med. **31**(5), 303–31 (2001)
100. Emerson, A., Rossi, E.: Immunogrid—the virtual human immune system project. Stud. Health Technol. Inform. **126**, 87–92 (2007)
101. Efroni, S., Harel, D., Cohen, I.: Reactive animation: realistic modeling of complex dynamic systems. IEEE Comput. **38**(1), 38–47 (2005)
102. Swerdlin, N., Cohen, I.R., Harel, D.: The lymph node B cell immune response: dynamic analysis in-silico. Proc. IEEE **96**(8), 1421–1443 (2008)
103. Mata, J., Cohn, M.: Cellular automata-based modeling program: synthetic immune system. Immunol. Rev. **216**(1), 198–212 (2007)
104. Meier-Schellersheim, M., Mack, G.: SIMMUNE, a tool for simulating and analyzing immune system behavior. CoRR **cs.MA/9903017** (1999)
105. Sneddon, M.W., Faeder, J.R., Emonet, T.: Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. Nat. Methods **8**(2), 177–83 (2011)
106. Folcik, V.A., An, G.C., Orosz, C.G.: The basic immune simulator: an agent-based model to study the interactions between innate and adaptive immunity. Theor. Biol. Med. Model. **4**, 39 (2007)
107. Sutterlin, T., Huber, S., Dickhaus, H., Grabe, N.: Modeling multi-cellular behavior in epidermal tissue homeostasis via finite state machines in multi-agent systems. Bioinformatics **25**(16), 2057–2063 (2009)
108. Bauer, A.L., Beauchemin, C.A.A., Perelson, A.S.: Agent-based modeling of host-pathogen systems: the successes and challenges. Inf. Sci. **179**(10), 1379–1389 (2009)
109. Fachada, N., Lopes, V.V., Rosa, A.: Agent based modelling and simulation of the immune system: a review. Technical report, Systems and Robotics Institute, Instituto Superior Tecnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal (2000)
110. Pappalardo, F., Zhang, P., Halling-Brown, M., Basford, K., Scalia, A., Shepherd, A.J., Moss, D., Motta, S., Brusic, V.: Computational simulations of the immune system for personalized medicine: state of the art and challenges. Curr. Pharmacogenom. Personal. Med. **6**(4), 260–271 (2008)
111. Ferguson, N., Cummings, D., Cauchemez, S., Fraser, C., Riley, S., Meeyai, A., Iamsirithaworn, S., Burke, D.: Strategies for containing an emerging Influenza pandemic in Southeast Asia. Nature **437**, 209–214 (2005)
112. Barrett, C., Bissett, K., Chen, J., Feng, X., Kumar, V.S.A., Marathe, M.: Epifast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems. In: International Conference on Supercomputing (ICS), pp. 430–439 (2009)
113. Kumar, S., Piper, K., Galloway, D.D., Hadler, J.L., Grefenstette, J.J.: Is population structure sufficient to generate area-level inequalities in influenza rates? An examination using agent-based models. BMC Public Health **15**, 947 (2015)

114. Fox, S.J., Miller, J.C., Meyers, L.A.: Seasonality in risk of pandemic influenza emergence. PLoS Comput. Biol. **13**(10), e1005749-1–e1005749-23 (2017)
115. Marathe, M., Vullikanti, A.: Computational epidemiology. Commun. ACM **56**(7), 88–96 (2013)
116. Bhatele, A., Yeom, J.S., Jain, N., Kuhlman, C.J., Livnat, Y., Bisset, K.R., Kale, L.V., Marathe, M.V.: Massively parallel simulations of spread of infectious diseases over realistic social networks. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 689–694 (2017)
117. Abdelhamid, S., Kuhlman, C.J., Marathe, M.V., Ravi, S.S., Reid, K.: Agent-based modeling and simulation of depression and its impact on students' success and academic retention. In: American Society for Engineering Education (ASEE) (2016)
118. Lum, K., Swarup, S., Eubank, S., Hawdon, J.: The contagious nature of imprisonment: an agent-based model to explain racial disparities in incarceration rates. J. R. Soc. Interface **11**(98), 12 (2014)
119. Schelling, T.C.: Dynamic models of segregation. J. Math. Sociol. **1**, 143–186 (1971)
120. Ameden, H.A., Boxall, P.C., Cash, S.B., Vickers, D.A.: An agent-based model of border enforcement for invasive species management. Can. J. Agric. Econ. **57**, 481–496 (2009)
121. Tonnang, H.E., Herve, B.D., Biber-Freudenberger, L., Salifu, D., Subramanian, S., Ngowi, V.B., Guimapi, R.Y., et al.: Advances in crop insect modelling method—towards a whole system approach. Ecol. Model. **354**, 88–103 (2017)
122. Merkey, B.V., Lardon, L.A., Seoane, J.M., Kreft, J.U., Smets, B.F.: Growth dependence of conjugation explains limited plasmid invasion in biofilms: an individual-based modelling study. Environ. Microbiol. **13**, 2435–2452 (2011)
123. Spies, T.A., White, E., Ager, A., Kline, J.D., Bolte, J.P., Platt, E.K., Olsen, K.A., et al.: Using an agent-based model to examine forest management outcomes in a fire-prone landscape in Oregon, USA. Ecol. Soc. **22**, 20 (2017)
124. Thorne, B.C., Bailey, A.M., Peirce, S.M.: Combining experiments with multi-cell agent-based modeling to study biological tissue patterning. Brief. Bioinform. **8**(4), 245–257 (2007)
125. Channakeshava, K., Bisset, K., Kumar, V.A., Marathe, M., Yardi, S.: High performance scalable and expressive modeling environment to study mobile malware in large dynamic networks. In: 25th IEEE International Parallel & Distributed Processing Symposium (IPDPS), pp. 770–781 (2011)
126. Bookstaber, R.: Using agent-based models for analyzing threats to financial stability. Technical Report Working Paper No. 0003, U.S. Dept. of Treasury (2012)
127. Paul, M., Dredze, M.: A model for mining public health topics from Twitter. Health **11**, 16–6 (2012)
128. Zhang, B., Chan, W.K.V., Ukkusuri, S.V.: Agent-based modeling for household level hurricane evacuation. In: Proceedings of the 2009 Winter Simulation Conference, pp. 2778–2784 (2009)
129. Grimm, V.: Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? Ecol. Model. **115**, 129–148 (1999)
130. Zhu, Y., Xie, K., Ozbay, K., Yang, H.: Hurricane evacuation modeling using behavior models and scenario-driven agent-based simulations. Procedia Comput. Sci. **130**, 836–843 (2018)
131. Shmulevich, I., Lähdesmäki, H., Dougherty, E.R., Astola, J., Zhang, W.: The role of certain post classes in Boolean network models of genetic networks. Proc. Natl. Acad. Sci. **100**(19), 10734–10739 (2003)
132. Heckbert, S., Baynes, T., Reeson, A.: Agent-based modeling in ecological economics. Ann. N. Y. Acad. Sci. **1185**, 39–53 (2010)
133. Squazzoni, F.: The impact of agent-based models in the social sciences after 15 years of incursions. Hist. Econ. Ideas **18**, 197–233 (2010)
134. Bruch, E., Atwell, J.: Agent-based models in empirical social research. Sociol. Methods Res. **44**, 186–221 (2013)
135. Bianchi, F., Squazzoni, F.: Agent-based models in sociology. WIREs Comput. Stat. **7**, 284–306 (2015)
136. Axelrod, R.: The dissemination of culture. J. Confl. Resolut. **41**, 203–226 (1997)
137. Epstein, J.M.: Modeling civil violence: an agent-based computational approach. Proc. Natl. Acad. Sci. (PNAS) **99**, 7243–7250 (2002)
138. Hethcote, H.: The mathematics of infectious diseases. SIAM Rev. **42**(4), 599–653 (2000)
139. Newman, M.: The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
140. Longini, I.M., Nizam, A., Xu, S., Ungchusak, K., Hanshaoworakul, W., Cummings, D.A., Halloran, E.M.: Containing pandemic influenza at the source. Science **309**(5737), 1083–1087 (2005)
141. Sander, B., Nizam, A., Garrison, L.P., Postma, M.J., Halloran, M.E., Ira, M., Longini, J.: Economic evaluation of influenza pandemic mitigation strategies in the United States using a stochastic microsimulation transmission model. Value Health **12**, 226–233 (2009)
142. Yang, Y., Sugimoto, J., Halloran, M., Basta, N., Chao, D., Matrajt, L., Potter, G., Kenah, E., Longini, I.M.: The transmissibility and control of pandemic influenza a (H1N1) virus. Science **326**(5953), 729–733 (2009)
143. Halloran, M., Ferguson, N., Eubank, S., Longini, I., Cummings, D., Lewis, B., Xu, S., Fraser, C., Vullikanti, A., Germann, T., Wagener, D., Beckman, R., Kadau, K., Barrett, C., Macken, C., Burke, D., Cooley, P.: Modeling targeted layered containment of an influenza pandemic in the United States. Proc. Natl. Acad. Sci. (PNAS) **105**(12), 4639–4644 (2008)
144. Pandey, A., Atkins, K.E., Medlock, J., Wenzel, N., Townsend, J.P., Childs, J.E., Nyenswah, T.G., Ndeffo-Mbah, M.L., Galvani, A.P.: Strategies for containing Ebola in West Africa. Science **346**(6212), 991–995 (2014)
145. Rivers, C., Lofgren, E., Marathe, M., Eubank, S., Lewis, B.: Modeling the impact of interventions on an epidemic of Ebola in Sierra Leone and Liberia. PLoS Curr. (2014). https://doi.org/10.1371/currents.outbreaks.4d41fe5d6c05e9df30ddce33c66d084c
146. Venkatramanan, S., Chen, J., Gupta, S., Lewis, B.L., Marathe, M., Mortveit, H.S., Vullikanti, A.: Spatio-temporal optimization of seasonal vaccination using a metapopulation model of Influenza. In: 2017 IEEE International Conference on Healthcare Informatics, ICHI, pp. 134–143 (2017)
147. Ji, Z., Yan, K., Li, W., Hu, H., Zhu, X.: Mathematical and computational modeling in complex biological systems. Hindawi BioMed Res. Int. **2017**, 1–16 (2017)
148. Walpole, J., Papin, J.A., Peirce, S.M.: Mathematical and computational modeling in complex biological systems. Ann. Rev. Biomed. Eng. **15**, 137–154 (2013)
149. Bauch, C., Earn, D.: Vaccination and the theory of games. Proc. Natl. Acad. Sci. (PNAS) **101**(36), 13391–13394 (2004)
150. Aspnes, J., Rustagi, N., Saia, J.: Worm versus alert: who wins in a battle for control of a large-scale network? In: Proceedings of Principles of Distributed Systems, 11th International Conference, OPODIS 2007, pp. 443–456 (2007)
151. Narahari, Y.: Game Theory and Mechanism Design, IISc Lecture Notes, vol. 4. World Scientific, Singapore (2014)
152. Papadimitriou, C.H.: Computational Complexity. Pearson Publishing, Reading, MA (1993)

153. Adiga, A., Chu, S., Eubank, S., Kuhlman, C.J., Lewis, B., Marathe, A., Marathe, M., Nordberg, E.K., Swarup, S., Vullikanti, A., Wilson, M.L.: Disparities in spread and control of Influenza in slums of Delhi: findings from an agent-based modelling study. BMJ Open **8**(1), 12 (2018)
154. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD '01: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66 (2001)
155. Domingos, P., Richardson, M.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining, pp. 61–70 (2002)
156. Kitsak, M., Gallos, L., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H., Makse, H.: Identifying influential spreaders in complex networks. Nat. Phys. **6**, 888–893 (2010)
157. Halloran, H., Longini Jr., I., Nizam, A., Yang, Y.: Possible containment of bio-terrorist smallpox. Science **298**, 1428–1432 (2002)
158. Barrett, C., Chen, J., Eubank, S., Kumar, V., Lewis, B., Marathe, A., Marathe, M.: Role of vulnerable and critical nodes in controlling epidemics in social networks. In: Proceedings of Epidemics (2008)
159. Barrett, C., Bisset, K., Leidig, J., Marathe, A., Marathe, M.: Economic and social impact of influenza mitigation strategies by demographic class. Epidemics **3**(1), 19–31 (2011)
160. Adiga, A., Kuhlman, C., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E.: Inferring local transition functions of discrete dynamical systems from observations of system behavior. Theor. Comput. Sci. **679**, 126–144 (2017)