On Some Fundamental Problems for Multi-Agent Systems Over Multilayer Networks

Daniel J. Rosenkrantz
University of Virginia
Charlottesville, VA, USA
drosenkrantz@gmail.com

Madhav V. Marathe
University of Virginia
Charlottesville, VA, USA
marathe@virginia.edu

Zirou Qiu
University of Virginia
Charlottesville, VA, USA
zq5au@virginia.edu

S. S. Ravi
University of Virginia
Charlottesville, VA, USA
ssravi0@gmail.com

Richard E. Stearns
University of Virginia
Charlottesville, VA, USA
thestearns2@gmail.com

ABSTRACT

Many researchers have considered multi-agent systems over single-layer networks as models for studying diffusion phenomena. Since real-world networks involve connections between agents with different semantics (e.g., family member, friend, colleague), the study of multi-agent systems over multilayer networks has assumed increased importance. Our focus is on one class of multi-agent system models over multilayer networks, namely multilayer synchronous dynamical systems (MSyDSs). We study several fundamental problems for this model. We establish properties of the phase spaces of MSyDSs and bring out interesting differences between single-layer and multilayer dynamical systems. We show that, in general, the problem of determining whether two given MSyDSs are inequivalent is **NP**-complete. This hardness result holds even when the only difference between the two systems is the local function at just one node in one layer. We also present efficient algorithms for the equivalence problem for restricted versions of MSyDSs (e.g., systems where each local function is a bounded-threshold function, and systems where the number of layers is fixed and each local function is symmetric). In addition, we investigate the expressive power of MSyDSs based on the number of layers. In particular, we examine conditions under which a system with $k \geq 2$ layers has an equivalent system with $k - 1$ or fewer layers.

KEYWORDS

Multilayer networks, Multi-agent systems, Dynamical systems, Equivalence problem, Expressive power, Complexity, Algorithms

ACM Reference Format:
Daniel J. Rosenkrantz, Madhav V. Marathe, Zirou Qiu, S. S. Ravi, and Richard E. Stearns. 2025. On Some Fundamental Problems for Multi-Agent Systems Over Multilayer Networks. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

1.1 Background and Motivation

Motivated by the spread of epidemics, information and social behavior in populations, the topic of contagion propagation over networks has been studied extensively in the literature (see e.g., [13, 16, 29, 41, 42]). When the contagion represents information about a product produced by an organization, the propagation models enable the organization to develop strategies for maximizing the spread of the information (see e.g., [21]). When the contagion is an epidemic, the propagation models enable public health officials to take appropriate actions to contain the spread of the epidemic (see e.g., [2, 23]). Such studies typically use networked dynamical systems models, where nodes represent individuals and edges represent relationships that enable a contagion to spread in the underlying network (see e.g., [1, 3, 5]). Each node has an active or inactive state which indicates whether or not the node has acquired the contagion. The **configuration** of the system at any time is the vector of state values of nodes at that time. The interaction between a node and its neighbors is based on an appropriate local function, and this interaction may change the state of the node. Throughout this paper, we will assume that nodes compute and update their states *synchronously*. The time evolution of the system is represented by a sequence of successive configurations. The global behavior of a dynamical system is captured by its **phase space**, which is a directed graph in which each node is a configuration and each directed edge indicates a single step transition from one configuration to another.

This work has generally assumed that the underlying network consists of a single layer; that is, there is only one type of relationship between entities for contagion propagation. A number of fundamental questions have been studied over the years for synchronous dynamical systems over single-layer networks. Following the literature (see e.g., [1]), we refer to these single-layer systems as SyDSs. Papers on SyDSs have addressed questions concerning reachability of configurations (i.e., does a given SyDS starting from a configuration C reach a configuration C'?), equivalence (i.e., are two given SyDSs equivalent in terms of their phase space?), existence of **fixed points** (i.e., configurations in which none of the nodes changes its state), etc.; see e.g., [3, 8, 30, 34, 36] and the references cited therein.

A number of papers have pointed out that single-layer networks are inadequate to capture complex real-world contagion phenomena. The reason is that single-layer networks model just one type of relationship among individuals. Instead, one needs to consider multilayer[1] networks (also known as multiplex networks [22, 31, 45]) that capture multi-relational structures (e.g., family member, friend, colleague, contact in a social network) among individuals to effectively model contagion propagation [17, 22, 29]. As a simple example, information that an individual obtains through a social network such as Facebook or Twitter may play a role in deciding whether the individual decides to wear a mask or get vaccinated. Thus, it is important to study dynamical systems where the underlying graph has multiple layers. In standard models of multilayer networks, there is just one set of nodes; each layer may have a different set of edges modeling different types of relationships (see, e.g., [10–12, 29, 31]). The model for a dynamical system over a multilayer network is a simple extension of the corresponding model over a single-layer network. In such a system, each node has a *single* state value but a possibly different interaction function for each layer. At each time instant, after computing the value of the interaction function of the node in each layer, a **master function** uses these values to compute the next state of that node. As mentioned earlier, we focus on the *synchronous* update scheme where all the nodes compute their next states and update their states simultaneously. For consistency with previous work [32], we refer to this model as a **multilayer synchronous dynamical system** (MSyDS). The notions of configuration and phase space for a MSyDS are the same as those for a dynamical system over a single-layer network. Algorithmic work on multilayer networks has centered around defining and computing appropriate centrality measures (see e.g., [11, 29, 31]). There has also been some work on estimating the sizes of cascades under certain contagion models (e.g., susceptible-infected-recovered or SIR model) through analysis and simulation in multilayer networks (see e.g., [12, 29] and the references cited therein). Additional discussion regarding work on multilayer systems appears in Section 1.4. For multilayer dynamical systems, there has been some work [32] on learning the local interaction functions under the probably approximately correct (PAC) model [40]. However, to our knowledge, fundamental issues such as the structure of phase spaces and the equivalence of multilayer dynamical systems have not been studied.

1.2 Contributions

Our work takes the first step towards exploring some fundamental aspects of multilayer synchronous dynamical systems (MSyDSs). A summary of our contributions is provided below.

(1) Phase Space Properties: We present MSyDSs whose local functions are threshold functions but whose phase spaces include long cycles in contrast to the phase spaces of single-layer SyDSs. For example, we show (see Section 3) that there is a MSyDS with two layers such that all its local functions are threshold functions and its phase space contains a cycle whose length is exponential in the number of nodes. We also give another construction of a MSyDS with n nodes and $2n-1$ layers such that all local functions are threshold functions and the phase space is a cycle of length 2^n. In contrast, it is known that for single-layer SyDSs where each local function is a threshold function, the maximum possible length of a phase space cycle is two [15]. Our constructions of MSyDSs with threshold local functions also show that their phase spaces may contain simple directed paths whose length is exponential in the number of nodes. (These paths are part of an exponentially long cycle.) For single-layer SyDSs with n nodes and threshold local functions, it is known that the length of any simple path in the phase space is $O(n^2)$ [4, 15].

(2) Equivalence Problem for MSyDSs: Given two MSyDSs $\mathcal{S}$ and $\mathcal{S}'$ on the same set of nodes, we say that they are equivalent if their phase spaces (as directed graphs) are identical. We show (see Section 4) that the equivalence problem for MSyDSs is **NP**-hard even for highly restricted versions of the MSyDSs. Specifically, our reduction from 3SAT (see [14] for a definition) produces two MSyDSs $\mathcal{S}$ and $\mathcal{S}'$ over a set of nodes with the following properties: (i) in each layer, the graphs of $\mathcal{S}$ and $\mathcal{S}'$ are identical; (ii) the graph in each layer is a star graph with zero or more isolated nodes; (iii) all local functions are threshold functions; and (iv) the MSyDSs $\mathcal{S}$ and $\mathcal{S}'$ differ in only the threshold of one node in one layer. Thus, a minor difference between two MSyDSs is sufficient to obtain computational intractability for the equivalence problem for MSyDSs with threshold local functions. In contrast, it is known that for two single-layer SyDSs where the underlying graphs are the same and the local functions are threshold functions, the equivalence problem can be solved efficiently [3].

(3) Efficient Algorithms for the Equivalence Problem for Restricted Classes of MSyDSs: We provide efficient algorithms (see Section 5) for the equivalence problem for several special classes of MSyDSs. These special classes include the following: (i) the number of layers in the two MSyDSs is *fixed* and each local function is symmetric[2]; (ii) each local function is a threshold function, the maximum node degree in each layer is *bounded* and each master function is `OR` or each master function is `AND`. These algorithms are obtained by reducing the equivalence problem for the two systems to a form of equivalence with respect to each node and showing that the node equivalence problem can be solved efficiently.

(4) Expressive Power of MSyDSs: We initiate the study of the expressive power of MSyDSs (see Section 6) based on the number of layers. We observe that for any MSyDS over $k \geq 2$ layers, there is an equivalent single-layer SyDS with more complex local functions. However, if one also restricts the class of local functions, we show that there are MSyDSs with k layers for which there is no equivalent MSyDS with fewer layers.

We also observe that many analysis problems (e.g., existence of fixed points) that are computationally intractable for single-layer dynamical systems remain so for multilayer systems as well.

[1]In much of the literature, the word "multilayer" is not hyphenated. We follow the same practice here.

[2]Symmetric functions (which are a proper superset of threshold functions) are defined in Section 2.

1.3 Additional Remarks

Long cycles (and paths) in the phase space are indicative of the complexity of the **reachability problem** for the underlying dynamical system (i.e., given a multilayer SyDS $\mathcal{S}$ and configurations C and C', can $\mathcal{S}$ reach C' starting from C?). As mentioned in Section 1.2, in any single layer system with n nodes and threshold local functions, the length of any phase space path is $O(n^2)$, and any such path ends in a limit cycle of length at most 2. So, the reachability problem for such a system is efficiently solvable. However, when the phase space of a system contains exponentially long cycles/paths, the reachability problem may be computationally intractable; such complexity results have appeared in several references (e.g., [4, 8]). The novelty in our work is that with just two layers, one can construct multilayer systems with threshold functions whose phase spaces contain exponentially long cycles (and hence exponentially long paths). To our knowledge, this is the first work that formally establishes phase space properties of multilayer systems.

Our hardness result for the equivalence problem for multilayer systems with threshold functions holds even when there is a very minor difference between the two systems. (As will be seen from the proof in Section 4, the only difference between the two systems constructed through a reduction from the 3SAT problem is in the threshold value of just one node in one layer.) We highlight this to point out that while the equivalence problem for single layer systems with threshold functions is efficiently solvable [3], a very minor difference suffices to make the problem computationally intractable for multilayer systems.

Finally, we note that when the number of layers is *fixed*, our efficient algorithm for the MSyDS equivalence problem allows *symmetric* local functions, a proper superset of threshold functions.

Note: For space reasons, only proof sketches are included for some results. Complete proofs for those results can be found in [35].

1.4 Related Work

Networked dynamical systems provide a systematic formal framework for agent-based models (ABM) and to capture interactions among agents in a network. Wellman [43] discusses the relationships between ABM and multi-agent systems. Networked dynamical system models have been used by the multi-agent systems community to study a variety of topics, including contagion propagation, graphical games and migration due to catastrophic events (see e.g., [18, 20, 24, 26]). Researchers have also studied formal aspects of various computational problems for networked dynamical systems (see e.g., [4, 8, 34, 36]). While this work has been in the context of single-layer systems, various research issues in the context of multilayer networks are also being actively pursued by the research community (see [45] and the references cited therein). For example, a number of papers have studied structural properties of multilayer networks and the computation of various centrality measures for such networks (see [6, 27, 45] and the references cited therein). In addition, there is work on other topics on multilayer networks, including reliability issues, analysis of cascades, enabling cooperation, and applications in various domains (see e.g., [12, 17, 19, 22, 25, 33, 39]).

To our knowledge, the only reference which considers fundamental computational questions regarding multilayer dynamical systems is [32]. As mentioned earlier, this work considers learning the local functions of MSyDSs under the PAC model but does not address the research questions studied in this paper.

2 PRELIMINARIES

2.1 Multilayer Synchronous Dynamical Systems

In this section, we present the definitions associated with multilayer synchronous dynamical systems. The notation and terminology used in this section are based on the presentation in [32].

Unless otherwise mentioned, the networks considered in our paper are undirected. We begin with the definition and notation for multilayer networks. A **multilayer network** [22] with $k \geq 1$ layers is a set of graphs $M = \{G_i : 1 \leq i \leq k\}$, where $G_i = (V, E_i)$ is the graph in the i^{th} layer. Thus, all of the graphs have the same node set V with n nodes, but the edge sets in different layers may be different. The definition of a discrete dynamical system over a multilayer network is a generalization of the corresponding definition for a single-layer system which has been studied by many researchers (see e.g., [3, 4, 28]). Our focus is on networked dynamical systems over the domain $\mathbb{B}$ = {0,1}. A **multilayer synchronous dynamical system** (MSyDS) $\mathcal{S}$ over the domain $\mathbb{B}$ has the following components.

(a) A multilayer network $M = \{G_i(V, E_i) : 1 \leq i \leq k\}$ with $k \geq 1$ layers. Each node $v \in V$ has a **state** from $\mathbb{B}$.

(b) A collection $\mathcal{F} = \{f_{i,v} : 1 \leq i \leq k,\ v \in V\}$ of functions, with $f_{i,v}$ denoting the **local interaction function** for node v in layer i.

(c) A collection $\Psi = \{\psi_v : v \in V\}$ of functions, with ψ_v denoting the **master function** of node v.

Let $V = \{v_1, v_2, \ldots, v_n\}$. At any time, the **configuration** C of the system is an n-vector $(s_1, s_2, \ldots, s_n)$, where $s_i \in \mathbb{B}$ is the state of node v_i at that time, $1 \leq i \leq n$. Given the configuration at time t, the configuration of the system at time $t+1$ is computed as follows.

(1) In each layer i, each node v computes the value of its local function $f_{i,v}$. The inputs to the function $f_{i,v}$ are the state of v and those of its neighbors in G_i, and the output of $f_{i,v}$ is a value in $\mathbb{B}$. We use $W_v(i)$ to denote this output value. Since there are k layers, the local functions provide a vector $\mathbf{W}_v$ of k values $(W_v(1), W_v(2), \ldots, W_v(k))$ for each node $v \in V$.
(2) Then, for each node v, its master function ψ_v is evaluated. The input to ψ_v is $\mathbf{W}_v$ and the output of ψ_v is a value in $\mathbb{B}$. This value becomes the state of v in time step $t+1$.
(3) All the nodes carry out the above computations and update their states *synchronously.*

Let C_t and C_{t+1} denote the configurations of a MSyDS $\mathcal{S}$ at times t and $t+1$ respectively. We refer to C_{t+1} as the **successor** of C_t and C_t as the **predecessor** of C_{t+1}. Since we restrict our attention to deterministic local and master functions, each configuration of a MSyDS has a *unique* successor; however, a configuration may have zero or more predecessors [32].

A Note About the Graph Model: In real-world applications, multilayer systems may have a different set of nodes and edges in each layer. When this happens, one can merge all the node sets into a single set and have different sets of edges in each layer. Hence, the commonly used model for multilayer systems (e.g., [10, 11, 17, 19, 22]) assumes one set of nodes for all the layers but (possibly) different

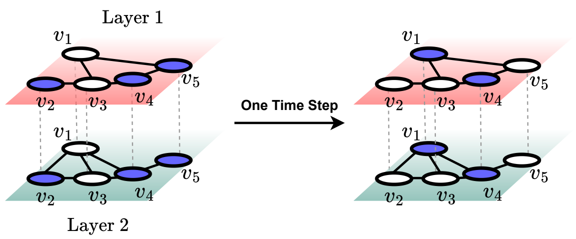

Figure 1: An example of a small MSyDS, where state-1 nodes are highlighted in blue. The master function at each node is OR. The thresholds of the nodes (same for the two layers) are: $v_1 : 2$, $v_2 : 2$, $v_3 : 3$, $v_4 : 1$ and $v_5 : 3$.

sets of edges in different layers. Our results also use this single node set model.

Graph Theoretic Terminology: Let M denote the underlying multilayer graph of a MSyDS $\mathcal{S}$. Recall that $G_i(V, E_i)$ denotes the graph in layer i of M. The **degree** of a node v in layer i is the number of neighbors of v in that layer. For a node v in layer i, the **closed neighborhood** of v in that layer includes v and each node u such that $\{u, v\} \in E_i$. To discuss the structure of phase spaces of networked dynamical systems, we will also need some terminology for directed graphs. In a directed graph, the **outdegree** of a node x is the number of outgoing edges from x. The phase space may also contain **self loops**, that is, directed edges of the form (x, x). A **simple path** in a directed graph is a directed path in which all nodes are distinct. Similarly, a **simple cycle** in a directed graph is a directed cycle in which all nodes are distinct.

Local and Master Functions: Since the domain under consideration is $\mathbb{B}$ = {0,1}, we consider several classes of Boolean functions as local and master functions. The definitions provided below are available in many standard references such as [9]. An important class of Boolean functions studied in the literature on the spread of social contagions is that of **threshold** functions (see e.g., [7, 16, 42]). For any integer $\tau \geq 0$, the τ-**threshold** function has the value 1 iff at least τ of its inputs have the value 1. We note that the 1-threshold function is the OR function. Also, the AND function with q inputs is the q-threshold function. When the local function $f_{i,v}$ of a node v is the τ-threshold function, we say that its threshold condition is satisfied if at least τ of the nodes in the closed neighborhood of v in G_i are in state 1. Symmetric Boolean functions are a superset of the class of threshold functions. The value of a **symmetric** Boolean function depends only on the number of 1's in the input. Thus, each threshold function is also a symmetric function. Likewise, the XOR function is symmetric. Each symmetric Boolean function with q inputs can be specified by a table with $q + 1$ rows, with row i specifying the value of the function when the number of 1's in the input is exactly i, $0 \leq i \leq q$. We refer to this as the **symmetry table** for the (symmetric) function. We consider threshold and symmetric Boolean functions for local and master functions.

An Example for a MSyDS: To illustrate the above definitions, we present a simple example of a 2-layer MSyDS in Figure 1. One can think of the edges in each of the two layers as representing different relationships (e.g., family member, friend) between pairs of nodes. There are 5 nodes in the system and the local function for each node in each layer is a threshold function. These threshold values are indicated in the caption of Figure 1. For simplicity, we have chosen the same threshold value for each node in the two layers. The master function at each node is OR. The initial configuration of the system is shown in the left panel of Figure 1), where nodes in state-1 are highlighted in blue. Thus, in the initial configuration, nodes v_2, v_4 and v_5 are in state-1 while nodes v_1 and v_3 are in state 0. This configuration C is represented by $(0, 1, 0, 1, 1)$.

We now explain how the system transitions from one configuration to the next. As mentioned above, let $C = (0, 1, 0, 1, 1)$ denote the current configuration. Consider node v_1 which is in state 0 in C. In Layer 1, v_1 has only neighbor (namely v_5) in state 1. In Layer 2, v_1 has two neighbors (namely v_2 and v_4) in state 1. Since the threshold of v_1 is 2 in each layer, the local functions of v_1 in layers 1 and 2 have values 0 and 1 respectively. Since the master function is OR, the next state of v_1 is 1. In a similar manner, one can verify that the next states of v_2, v_3, v_4 and v_5 are 0, 0, 1 and 0 respectively. Thus, C', the successor of C, is given by $(1, 0, 0, 1, 0)$, as shown in the right panel of Figure 1. It can also be verified that the successor of C' is C' itself; that is, once the system reaches the configuration $C' = (1, 0, 0, 1, 0)$, no more state changes occur, and the system stays in that configuration for ever. Thus, C' is a *fixed point.* □

Additional terminology regarding dynamical systems: We review some necessary terminology regarding networked dynamical systems from the literature (see e.g., [1, 4, 28]). Suppose the node set of the underlying multilayer graph of a MSyDS $\mathcal{S}$ over $\mathbb{B}$ has n nodes. Thus, the total number of distinct configurations is 2^n. The **phase space** of $\mathcal{S}$, denoted by $\mathcal{P}(\mathcal{S})$, is directed graph with a node for each possible configuration; for each pair of nodes x and y, there is a directed edge (x, y) in $\mathcal{P}(\mathcal{S})$ iff node y represents the configuration which is the successor of the configuration represented by the node x. A **transient** in the phase space is a directed path leading to a cycle. Since our MSyDSs are deterministic, each configuration has a unique successor. In other words, the outdegree of each phase space node is 1. Thus, the phase space has 2^n nodes and 2^n edges.

2.2 Problem formulations

We study several fundamental questions concerning MSyDSs. The first of these questions involves the structure of the phase spaces of MSyDSs. In particular, we study how the (directed) paths and cycles in the phase spaces of MSyDSs can be significantly different compared to those of single-layer SyDSs, even when the local functions used by the dynamical systems are threshold functions. In fact, our results show that one can construct MSyDSs on two-layer networks whose phase spaces have significantly longer cycles and paths compared to single-layer SyDSs.

The second question involves the equivalence of two MSyDSs. Let $\mathcal{S}$ and $\mathcal{S}'$ be a pair of MSyDSs with the same node set V. We say that $\mathcal{S}$ and $\mathcal{S}'$ are *equivalent* if their phase spaces (considered as directed graphs) are identical. We formulate the corresponding decision problem as the *inequivalence* problem as follows.

Inequivalence of MSyDSs:
Given: Two MSyDSs $\mathcal{S}$ and $\mathcal{S}'$ on the same set V of nodes.
Question: Are the phase spaces of $\mathcal{S}$ and $\mathcal{S}'$ *different*?

The reason for considering the Inequivalence problem is that the problem belongs to the class **NP**. To see this, define an **inequivalence witness** as a configuration C over V, such that the successors of C under $\mathcal{S}$ and $\mathcal{S}'$ are different. The Inequivalence problem is in **NP** since one can guess an inequivalence witness C, and efficiently verify that the successors of C under $\mathcal{S}$ and $\mathcal{S}'$ are different. We show that the Inequivalence problem is **NP**-complete even when there is only a minor difference between the given MSyDSs. We also present efficient algorithms for restricted versions of the Inequivalence problem.

The last fundamental question that we address concerns the expressive power of MSyDSs in terms of the number of layers. Specifically, we investigate whether for a given MSyDS with k layers, one can obtain an equivalent MSyDS with fewer layers. Our results point out that while this can be done if more complex local functions are permitted, it may not be possible if there are additional restrictions on the local functions.

3 PHASE SPACE PROPERTIES OF MULTILAYER SYSTEMS

Here, we examine phase space properties of multilayer systems. We identify some interesting differences between the phase space properties of single-layer and multilayer systems. We begin with a lemma that is useful in showing that phase spaces of MSyDSs with two layers may have long cycles.

Lemma 3.1. *For every $n \geq 1$, there is a MSyDS $\mathcal{S}_n$ with two layers and n nodes, where every local function is a threshold function, and every master function is symmetric, with the following properties: the phase space of $\mathcal{S}_n$ contains exactly one cycle, the length of this cycle is $n+1$, and every transient is of length one.*

Proof. Let the nodes of $\mathcal{S}_n$ be denoted as v_i, $1 \leq i \leq n$. The graph in both layers is a complete graph. In layer 1, the threshold of each node is n. In layer 2, the threshold of each node v_i, $1 \leq i \leq n$, is $i-1$. Every master function is XOR, which is a symmetric function.

Now, consider the phase space of $\mathcal{S}_n$. For each j, $0 \leq j \leq n$, let C_j denote the configuration of $\mathcal{S}_n$ such that for each v_i, $1 \leq i \leq n$, if $i \leq j$ then $C_j[v_i] = 1$, and if $i > j$ then $C_j[v_i] = 0$. Note that the successor configuration of each C_j is $C_{j+1 \bmod (n+1)}$. Thus, the $n+1$ configurations C_j, $0 \leq j \leq n$, form a phase space cycle of length $n+1$. Moreover, for any configuration C, let $|C|$ denote the number of 1's in C, i.e., the Hamming weight of C. Let $j' = |C|+1 \bmod (n+1)$. Then, the successor configuration of C is $C_{j'}$, a configuration in the above phase space cycle. □

The next result shows that the cycles in the phase spaces of MSyDSs with just two layers and threshold local functions can be exponentially large in the number of nodes of the system. We note that such cycles are much larger than the ones possible in single-layer SyDSs with threshold local functions.

Theorem 3.2. *For MSyDSs with two layers, threshold local functions, and XOR master functions, a phase space cycle can be exponentially large in the number of nodes.*

Proof. For an integer $q \geq 1$, let $p_1, p_2, \ldots, p_q$ be the first q primes, where we assume that 2 is the first prime. Let Σ_q be the two-layer MSyDS obtained as the union of the q two-layer MSyDSs $\mathcal{S}_{p_1-1}, \mathcal{S}_{p_2-1}, \ldots, \mathcal{S}_{p_q-1}$, each of which is constructed as described in the proof of Lemma 3.1. More precisely, the node set of Σ_q is the union of the node sets of the q constituent MSyDSs, the edge set in each layer is the union of the edge sets on that layer of the constituent MSyDSs, and each node has the same master function as in its constituent MSyDS. Note that neither layer of Σ_q has an edge between nodes occurring in two distinct constituent MSyDSs. Since each constituent MsyDS $\mathcal{S}_{p_j-1}$, $1 \leq j \leq q$, has a phase space cycle of length p_j, Σ_q has a phase space cycle of length $\Pi_{j=1}^{q} p_j$. In this construction, the number of nodes in the MSyDS is $\sum_{j=1}^{q} p_j - q$. It is known that $\sum_{j=1}^{q} p_j$ is asymptotically $\Theta(q^2 \log q)$ [38]. As explained above, the length of a cycle in the phase space is $\prod_{j=1}^{q} p_j$, which is asymptotically $\Omega(e^{q \log q})$ [44]. Thus, the length of this cycle is exponential in the number of nodes in the system. □

The next result shows that there are MSyDSs where all the 2^n nodes in the phase space form a single simple cycle.

Theorem 3.3. *For every $n \geq 2$, there is a MSyDS $\mathcal{S}_n$ with n nodes, for which every local function is a threshold function, and every master function is symmetric, whose phase space consists of a single cycle of length 2^n.*

Proof sketch: Let the nodes V of $\mathcal{S}_n$ be denoted as v_i, $1 \leq i \leq n$. For any configuration C of V, let $\widehat{C}$ denote the integer encoded by C, with $C[v_1]$ viewed as the low order bit. Let C' denote the successor configuration of C under $\mathcal{S}_n$. $\mathcal{S}_n$ will be constructed to have the property that $\widehat{C'}$ will equal $\widehat{C} + 1 \bmod 2^n$. This property ensures that the entire phase space of $\mathcal{S}_n$ consists of a single cycle of length 2^n.

For any node j, $1 \leq j \leq n$, let V_j denote the set of nodes $\{v_1, v_2, \ldots, v_j\}$. Note that V_j corresponds to bit j and the lower bits of $\widehat{C}$.

MSyDS $\mathcal{S}_n$ is constructed as follows. There are $2n-1$ layers. The layer 1 graph contains no edges. For each j, $2 \leq j \leq n$, the two layers j and $n+j-1$ contain the same graph. This graph contains the $j-1$ edges $\{v_j, v_i\}$, $1 \leq i < j$, i.e., an edge between v_j and each node in V_{j-1}.

In layer 1, every node has threshold 1. In layer j, $2 \leq j \leq n$, node v_j has threshold $j-1$, and every other local function is the constant function 0, i.e. a threshold function with threshold equal to the node degree plus two. In layer $n+j-1$, $2 \leq j \leq n$, node v_j has threshold j, and every other local function is the constant function 0.

The master function for node v_1 is NOR. For each node v_j, $2 \leq j \leq n$, the master function is the symmetric function that is 1 iff either exactly one or exactly two of its $2n-1$ inputs equal 1. It can be shown that for any configuration C of V, the successor configuration C' has the required property that $\widehat{C'} = \widehat{C} + 1 \bmod 2^n$. For details see [35]. □

4 COMPLEXITY OF EQUIVALENCE

In this section, we show that the Inequivalence problem is **NP**-Complete, even when there is just a minor difference between given pair of MSyDSs $\mathcal{S}$ and $\mathcal{S}'$.

Theorem 4.1. *The MSyDS Inequivalence problem is **NP**-complete, even when (i) all local functions are threshold functions, (ii) the two multilayer networks are identical, (iii) the graph in each layer is a star graph plus a set (possibly empty) of isolated nodes, (iv) all master functions are* OR*, and (v) the two given MSyDSs differ only in the value of the threshold of one node in one layer.*

Proof sketch: As mentioned in Section 2.2, the MSyDS Inequivalence problem is in **NP**. For **NP**-hardness, a reduction from the 3SAT problem [14] proceeds as follows. Let f be the given CNF formula over a set X of Boolean variables. Let n be the number of variables in X and m be the number of clauses in f. The constructed MSyDSs $\mathcal{S}$ and $\mathcal{S}'$ have the same node set V, consisting of $2n+1$ nodes, as follows. For each variable $x_i \in X$, V contains the two nodes y_i and z_i. Intuitively, node y_i corresponds to the literal x_i, and node z_i corresponds to the literal $\overline{x_i}$. We refer to these $2n$ nodes as **literal nodes**. There is also one additional node, w, which we refer to as the **center node**.

The constructed pair of multilayer graphs each contains node set V and $n+m+1$ layers. We refer to the set of layers as $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3$. Γ_1 consists of a single graph G_1. Γ_2 consists of n graphs, which we refer to as G_2^i, $1 \le i \le n$. Γ_3 consists of m graphs, which we refer to as G_3^j, $1 \le j \le m$.

Graph G_1 contains $2n$ edges, and is a star graph with node w in the center, and the other $2n$ nodes as leaves. Each graph G_2^i contains the two edges $\{w, y_i\}$ and $\{w, z_i\}$. Each graph G_3^j contains an edge for each literal in clause c_j. The endpoints of a given edge are w and the literal node for the *complement* of the literal occurring in the clause. For instance, if clause c_j is $x_3 \vee x_6 \vee \overline{x_9}$, then G_3^j contains the three edges $\{w, z_3\}$, $\{w, z_6\}$ and $\{w, y_9\}$.

The threshold of every node other than node w is 0 in every graph. The threshold of node w in each graph G_2^i is 2. The threshold of node w in each graph G_3^j is the number of literals in clause j. The threshold of node w in graph G_1 is n in $\mathcal{S}$, and is $n+1$ in $\mathcal{S}'$.

The master function at each node is OR.

This completes the construction of $\mathcal{S}$ and $\mathcal{S}'$. Note that the only difference between $\mathcal{S}$ and $\mathcal{S}'$ is the value of the threshold of node w in G_1. It can be shown that f is satisfiable iff $\mathcal{S}$ and $\mathcal{S}'$ are inequivalent. For details, see [35]. ◻

Remark. The equivalence problem for single-layer systems with threshold local functions is known to be efficiently solvable [3]. Thus, from the complexity perspective, the above theorem points out a significant difference between single-layer and multilayer systems with threshold local functions.

5 EFFICIENT ALGORITHMS FOR EQUIVALENCE FOR SPECIAL CLASSES OF MSYDSS

We first introduce several key concepts which help us derive efficient solutions for the equivalence problem for several restricted classes of MSyDSs. Let $\mathcal{S}$ and $\mathcal{S}'$ be a pair of MSyDSs with the same node set V. For a given node $v \in V$, we say that $\mathcal{S}$ and $\mathcal{S}'$ are *v-equivalent* if for every configuration C of V, the value of v in the successor of C under $\mathcal{S}$ and $\mathcal{S}'$ are the same. Note that $\mathcal{S}$ and $\mathcal{S}'$ are equivalent iff for every node $v \in V$, $\mathcal{S}$ and $\mathcal{S}'$ are v-equivalent. Thus, an efficient algorithm for node-equivalence would lead to an efficient algorithm for MSyDS equivalence.

For a given threshold function, we use the term *negative threshold* to mean the minimum number of 0's that make the function equal 0. Thus, for a given node v of a MySDS, if the local function for v on a given layer is a threshold function with threshold t, and the node has degree δ in that layer, the negative threshold for that local function is $\delta - t + 2$.

5.1 Fixed Number of Layers

Suppose that for a node v of a given pair of MSyDSs $\mathcal{S}_0$ and $\mathcal{S}_1$ with a common node set V, we want to determine the v-equivalence of $\mathcal{S}_0$ and $\mathcal{S}_1$. We could, for each of the $2^{|V|}$ configurations C of V, compare the value of v in the successor configuration of C under $\mathcal{S}_0$ and under $\mathcal{S}_1$. However, since the number of configurations is exponential in the number of nodes, this straightforward approach would take time exponential in the number of nodes.

In contrast, we show that when there is a bound on the number of layers of the given MSyDSs, and each local function is symmetric, v-equivalence can be determined in time polynomial in the number of nodes, where the degree of the polynomial depends on the bound on the number of layers. Our algorithm for doing this is based on constructing a partition Π_v of the node set V, such that Π_v has the following two key properties.

(1) The number of blocks in partition Π_v is a function of the number of layers in the given MSyDSs, and is independent of the number of nodes.
(2) For any configuration C of V, the value of node v in the successor configuration of C for each of the given MSyDSs is determined by how many of the nodes in each of the blocks of Π_v equal 1 in C.

For any configuration C of V, let the *v-profile* of C be a count of how many of the nodes in each of the blocks of Π_v are equal to 1 in C. Since the number of blocks of Π_v is polynomial, the number of possible profiles is polynomial, and so the set of all possible v-profiles can be explored in polynomial time. The approach of exploring all v-profiles yields a polynomial time algorithm, as described by the following result.

Theorem 5.1. *If the number of layers is bounded by a fixed value for both dynamical systems, the MSyDS equivalence problem can be solved in polynomial time when all local functions are symmetric functions, even when each master function is arbitrary.*

Proof. Let K be the fixed bound on the number of layers. Let $\mathcal{S}_0$ and $\mathcal{S}_1$ denote the given pair of MSyDSs, with common node set V. Let k_0 and k_1 be the number of layers in $\mathcal{S}_0$ and $\mathcal{S}_1$, respectively. Let $k' = k_0 + k_1$. We represent a layer of $\mathcal{S}_0$ or $\mathcal{S}_1$ as a pair (a, ℓ), where $a \in \mathcal{B} = \{0, 1\}$ specifies one of the two given MSyDSs, and ℓ is a layer number of $\mathcal{S}_a$, i.e., $1 \le \ell \le k_a$. We refer to such a pair as an *anchored-layer*.

Let L denote the set of k' anchored layers. Let $\mathcal{L}$ denote the power set of L, so that $|\mathcal{L}| = 2^{k'}$. We define a *profile* to be a vector with an element for each member of $\mathcal{L}$, where each element value is an integer in the range 0 through $|V|$. Thus, a profile has an element for each subset of L.

Now, consider a given node $v \in V$. Let $\xi_v : V \to \mathcal{L}$ denote the function that maps each node u into the set of anchored layers (a, ℓ) such that u and v are generalized neighbors[3] in layer ℓ of $\mathcal{S}_a$. Note that ξ_v induces a partition of V, with each block of the partition consisting of those nodes that ξ_v maps into the exact same set of anchored layers. We refer to this partition as Π_v. Thus, a pair of nodes u and w are in the same block of Π_v iff $\xi_v(u) = \xi_v(w)$, i.e., for each layer of $\mathcal{S}_0$ and each layer of $\mathcal{S}_1$, u and w are either both generalized neighbors of v in that layer or neither is a generalized neighbor of v.

Based on partition Π_v, for each set $\lambda \in \mathcal{L}$, we let $V_{v,\lambda}$ denote the set of nodes in the block of the partition associated with λ, i.e., the set of nodes u such that $\xi_v(u) = \lambda$. Thus, $V_{v,\lambda}$ is the set of nodes u such that v and u are generalized neighbors in every anchored layer in λ, and in no other anchored layer.

We say that a given profile θ is a *v-profile* if for every $\lambda \in \mathcal{L}$, $\theta[\lambda] \leq |V_{v,\lambda}|$. Let Θ_v denote the set of all v-profiles.

We now specify how a given v-profile θ specifies a value for node v for each of the two given MSyDSs $\mathcal{S}_0$ and $\mathcal{S}_1$. Let function $\rho_v : \Theta_v \times \mathcal{B} \to \mathcal{B}$ be defined as follows: Consider a given $(\theta, a) \in \Theta_v \times \mathcal{B}$. For each anchored layer (a, ℓ) of $\mathcal{S}_a$, let

$$w_v((a,\ell)) = \sum_{\lambda \in \mathcal{L} \mid (a,\ell) \in \lambda} \theta[\lambda]$$

and let $W_v((a, \ell))$ be the value of the symmetric local function $f_{\ell,v}$ of $\mathcal{S}_a$ when exactly $w_v((a, \ell))$ of its inputs equal 1. This defines a k_a-vector $\mathbf{W}_{v,a}$ of Boolean values, where $\mathbf{W}_v(\ell) = w_v((a, \ell))$. Then, $\rho_v(\theta, a)$ is defined to be the value of the master function $\psi_{v,a}$ of $\mathcal{S}_a$ when $\mathbf{W}_{v,a}$ is the input to $\psi_{v,a}$.

Let μ_v denote the function that maps each configuration C over V into a v-profile, as follows. For each configuration C over V, $\mu_v(C)$ is the v-profile such that for each $\lambda \in \mathcal{L}$, $\mu_v(\lambda)$ is the number of nodes in $V_{v,\lambda}$ that have value 1 in C. Note that μ is an onto function, i.e., for every v-profile θ, there is at least one configuration C of V such that $\mu_v(C) = \theta$. Such a configuration can be constructed by setting, for each $\lambda \subseteq L$, exactly $\theta(\lambda)$ members of $V_{v,\lambda}$ to value 1, and the other members of $V_{v,\lambda}$ to value 0.

Note that for each MSyDS $\mathcal{S}_a$ and any configuration C of V, the value of node v in the successor configuration of C under $\mathcal{S}_a$ equals the value of $\rho_v(\mu_v(C), a)$. Thus, we can determine the v-equivalence of $\mathcal{S}_0$ and $\mathcal{S}_1$ by comparing the values of $\rho(\theta, 0)$ and $\rho(\theta, 1)$ for every v-profile θ.

While the number of configurations is exponential in the number of nodes, the number of elements in any profile is bounded by a function of K, and the value of each component of any v-profile is at most the number of nodes. Since K is fixed, the number of profiles is polynomial in the number of nodes. Thus, this approach yields a polynomial time algorithm, as follows.

The algorithm. We now describe a polynomial time algorithm for equivalence, utilizing the above concepts. Given $\mathcal{S}_0$ and $\mathcal{S}_1$, the algorithm first constructs $\mathcal{L}$. Then, for each node v, the algorithm determines whether $\mathcal{S}_0$ and $\mathcal{S}_1$ are v-equivalent, as follows. The algorithm constructs partition Π_v, and then constructs Θ_v, the set of all v-profiles. Next, for each $\theta \in \Theta_v$, the algorithm computes $\rho(\theta, 0)$ and $\rho(\theta, 1)$. Note that $\mathcal{S}_0$ and $\mathcal{S}_1$ are v-equivalent iff for every $\theta \in \Theta_v$, $\rho(\theta, 0) = \rho(\theta, 1)$.

Finally, $\mathcal{S}_0$ and $\mathcal{S}_1$ are equivalent iff for every node v and every $\theta \in \Theta_v$, $\rho(\theta, 0) = \rho(\theta, 1)$. □

[3]We use the term "generalized neighbors" to mean that u and v are in each other's closed neighborhood.

5.2 Bounded Threshold

We need a couple of lemmas to get an efficient algorithm for this special case.

Lemma 5.2. *Let $\mathcal{S}_0$ and $\mathcal{S}_1$ be a pair of MSyDSs with common node set V, such that every local function is a threshold function, and all master functions are* OR. *Let τ denote the largest threshold value occurring in $\mathcal{S}_0$ and $\mathcal{S}_1$. Then $\mathcal{S}_0$ and $\mathcal{S}_1$ are inequivalent iff there is an inequivalence witness with at most τ 1's.*

Proof sketch: Trivially, if there is an inequivalence witness with at most τ 1's, then $\mathcal{S}_0$ and $\mathcal{S}_1$ are inequivalent. For the converse, it can be shown that whenever there is an inequivalence witness, there is one with at most τ 1's. The details of this proof appear in [35]. □

We now note that a dual lemma holds, with regard to maximum negative threshold value. The proof is the dual to the proof of Lemma 5.2.

Lemma 5.3. *Let $\mathcal{S}_0$ and $\mathcal{S}_1$ be a pair of MSyDSs with common node set V, such that every local function is a threshold function, and all master functions are* AND. *Let ν denote the largest negative threshold value occurring in $\mathcal{S}_0$ and $\mathcal{S}_1$. Then $\mathcal{S}_0$ and $\mathcal{S}_1$ are inequivalent iff there is an inequivalence witness with at most ν 0's.*

We are now ready to prove the main result of this subsection.

Theorem 5.4. *(a) If all local functions are threshold functions, the maximum threshold value is bounded by a fixed value, and all master functions are* OR, *the MSyDS equivalence problem can be solved in polynomial time.*

(b) If all local functions are threshold functions, the maximum negative threshold value is bounded by a fixed value, and all master functions are AND, *the MSyDS equivalence problem can be solved in polynomial time.*

Proof. (a) Let τ_{max} denote the fixed bound on the maximum threshold value. From Lemma 5.2, the two given MSyDSs are inequivalent iff there is an inequivalence witness with at most τ_{max} 1's. In polynomial time, all configurations with at most τ_{max} 1's can be generated, and the successor configurations under the two given MSyDSs can be found.

(b) Let ν_{max} denote the fixed bound on the maximum negative threshold value. From Lemma 5.3, the two given MSyDSs are inequivalent iff there is an inequivalence witness with at most ν_{max} 0's. In polynomial time, all configurations with at most ν_{max} 0's can be generated, and the successor configurations under the two given MSyDSs can be found. □

The corollary below follows immediately.

Corollary 5.5. *If the maximum node degree of any graph in any layer is bounded by a fixed value, the MSyDS equivalence problem can be solved in polynomial time when all local functions are threshold functions, and either every master function is* OR *or every master function is* AND.

6 EXPRESSIVE POWER OF MSYDSS

In this section, we consider the expressive power of some classes of MSyDSs, based on the number of network layers.

Proposition 6.1. *Every MSyDS is equivalent to a single-layer SyDS.*

Proof. Given MSyDS $\mathcal{S}$, there is an equivalent SyDS $\mathcal{S}'$ whose underlying graph is a complete graph. The local function of a given node v in $\mathcal{S}'$ is the master function of $\mathcal{S}$ for v, applied to the set of local functions for v on the layers of $\mathcal{S}$. □

Example: We use the 2-layer system in Figure 1 to illustrate the construction outlined in the proof of Proposition 6.1. The underlying graph of an equivalent single-layer system is a complete graph on 5 nodes. Let $\text{Thr}(V', q)$, where V' is a subset of nodes and q is a non-negative integer, denote the Boolean function which has the value 1 when at least q of the nodes in V' have the value 1. Using this notation, the local function associated with node v_1 in the single-layer system is:

$$\text{Thr}(\{v_1, v_3, v_5\}, 2) \texttt{ OR } \text{Thr}(\{v_1, v_2, v_3, v_4\}, 2)$$

The local functions for the other nodes of the system in Figure 1 can be constructed in a similar manner. □

The next result shows that there is a hierarchy of expressive power, based on the number of layers.

Theorem 6.2. *(1) For every $k \geq 2$, there is a MSyDS $\mathcal{S}_k$ that contains k layers, for which every local function is a threshold function, and every master function is* OR*, with the following properties: For each $\mathcal{S}_k$, there is no equivalent MSyDS with fewer layers, for which every local function is symmetric, and every master function is* OR*. (2) However, there is an equivalent MSyDS with two layers, for which every local function is a threshold function, and every master function is* AND.

Proof sketch: For Part (1), let $\mathcal{S}_k$ be the following k layer MSyDS. Node set V contains the $k + 1$ nodes: $\{a, b_1, b_2, \ldots, b_k\}$. Layer i, $1 \leq i \leq k$, contains only one edge: $\{a, b_i\}$. In every layer, the local function for node a is a threshold function with threshold 2, and every other local function is the constant function 0, i.e., a threshold function with threshold value equal to two plus the node degree in that layer. Every master function is OR.

It can be shown by contradiction that there does not exist a MSyDS $\mathcal{S}'$ with fewer than k layers that is equivalent to $\mathcal{S}_k$, such that every local function of $\mathcal{S}'$ is symmetric, and every master function is OR. The details of this proof appear in [35].

To prove Part (2) of the theorem, let $\mathcal{S}''$ be the following two layer MSyDS with node set V. Layer 1 contains no edges, and layer 2 contains the k edges $\{a, b_i\}$, $1 \leq i \leq k$. In layer 1, the local function for node a is a threshold function with threshold 1. In layer 2, the local function for node a is a threshold function with threshold 2. In both layers, the local function for every node other than a is the constant function 0. Every master function is AND. It can be verified that $\mathcal{S}_k$ and $\mathcal{S}''$ are equivalent. □

7 ADDITIONAL REMARKS AND FUTURE RESEARCH DIRECTIONS

We have studied some fundamental questions for MSyDSs, such as the structure of their phase spaces, the complexity of equivalence problems and their expressive power. Researchers have studied several other fundamental questions for single layer dynamical systems. Examples of such questions are **fixed point existence** (i.e., does the given single-layer system have a fixed point?) and **predecessor existence** (i.e., given a single-layer system and a configuration C, is there a configuration C' such that there is a one-step transition from C' to C?); see [4, 30, 37] and the references cited therein. Many of these problems are known to be **NP**-complete for single-layer systems. One can easily extend these hardness results to MSyDSs using the following result.

Proposition 7.1. *Let $\mathcal{S}$ be a given single-layer SyDS over the domain $\mathbb{B}$ = {0,1}. For any $k \geq 2$, a k-layer MsyDS $\mathcal{S}'$ over $\mathbb{B}$ whose phase space is identical to that of $\mathcal{S}$ can be constructed.*

Proof. Let $G(V, E)$ denote the underlying graph of $\mathcal{S}$. The MSyDS $\mathcal{S}'$ uses the same set of nodes V. For $1 \leq i \leq k$, the graph $G_i(V, E_i)$ in each layer i is the same as G. Further, in each layer, the local function for a node $v \in V$ is the same as the local function of v in $\mathcal{S}$. Finally, the master function for each node v of $\mathcal{S}'$ is the OR function. It is straightforward to verify that for each configuration C, the successor of C is the same in both $\mathcal{S}$ and $\mathcal{S}'$. Thus, the phase spaces of $\mathcal{S}$ and $\mathcal{S}'$ are identical. □

From Proposition 7.1, it is seen that hardness results for single-layer systems yield similar results for multilayer systems as well.

We close by pointing out some directions for future research concerning computational problems for multi-layer dynamical systems. First, it will be useful to examine whether other problems that are efficiently solvable for single-layer systems remain so for multilayer systems. For example, it is known that for single-layer SyDSs where the treewidth of the underlying graph is bounded and the local functions have certain properties, the existence of certain subgraphs of the phase space can be checked efficiently [34]. It is of interest to study whether such algorithms can be extended to the multilayer case. It is also of interest to investigate whether results for single-layer SyDSs with stochastic local functions [5] can be extended to MSyDSs. Our work on the expressive power of MSyDSs has considered systems where the local functions are threshold or symmetric functions. An interesting direction is to extend the results on expressive power to multilayer systems with other classes of Boolean local functions.

ACKNOWLEDGMENTS

We thank the reviewers of AAMAS 2025 and the editors for their valuable suggestions. This work was supported by University of Virginia Strategic Investment Fund award number SIF160, Virginia Department of Health grant VDH-21-501-0135-1, DTRA Grant HDTRA1-24-R-0028, and NSF Grants CCF-1918656 (Expeditions) and OAC-1916805 (CINES). Opinions, findings, and conclusions are those of the authors and do not necessarily reflect the view of the funding entities.

REFERENCES

- [1] Abhijin Adiga, Chris J Kuhlman, Madhav V Marathe, Henning S Mortveit, S. S. Ravi, and Anil Vullikanti. 2019. Graphical dynamical systems and their applications to bio-social systems. *International Journal of Advances in Engineering Sciences and Applied Mathematics* 11 (2019), 153–171.
- [2] Aniruddha Adiga, Bryan Lewis, Simon Levin, Madhav V Marathe, H Vincent Poor, S. S. Ravi, Daniel J Rosenkrantz, Richard E Stearns, Srinivasan Venkatramanan, Anil Vullikanti, et al. 2022. AI techniques for forecasting epidemic dynamics: Theory and practice. In *Artificial Intelligence in COVID-19*. Springer, New York, NY, 193–228.
- [3] Abhijin Adiga, Samarth Swarup, Anil Vullikanti, Christopher L. Barrett, Stephen G. Eubank, Chris J. Kuhlman, Madhav V. Marathe, Henning S. Mortveit, S. S. Ravi, Daniel J. Rosenkrantz, and Richard Edwin Stearns. 2019. Validating Agent-Based Models of Large Networked Systems. In *2019 Winter Simulation Conference, WSC 2019, National Harbor, MD, USA, December 8-11, 2019*. IEEE, New York, NY, 2807–2818.
- [4] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. 2006. Complexity of reachability problems for finite discrete dynamical systems. *J. Comput. System Sci.* 72, 8 (2006), 1317–1345.
- [5] Christopher L. Barrett, Harry B. Hunt III, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, and Richard Edwin Stearns. 2011. Modeling and analyzing social network dynamics using stochastic discrete graphical dynamical systems. *Theor. Comput. Sci.* 412, 30 (2011), 3932–3946.
- [6] Tanmoy Chakraborty and Ramasuri Narayanam. 2016. Cross-layer betweenness centrality in multiplex networks with applications. In *2016 IEEE 32nd international conference on data engineering (ICDE)*. IEEE, New York, NY, 397–408.
- [7] Wei Chen, Xiaoming Sun, Jialin Zhang, and Zhijie Zhang. 2021. Network inference and influence maximization from samples. In *Proc. International Conference on Machine Learning*. PMLR, Online, 1707–1716.
- [8] D. Chistikov, G. Lisowski, M. Paterson, and P. Turrini. 2020. Convergence of Opinion Diffusion is PSPACE-Complete. In *Proc. AAAI*. AAAI Press, Online publisher, 7103–7110.
- [9] Y. Crama and P. Hammer. 2011. *Boolean Functions: Theory, Algorithms, and Applications*. Cambridge University Press, New York, NY.
- [10] Manlio De Domenico, Clara Granell, Mason A Porter, and Alex Arenas. 2016. The physics of spreading processes in multilayer networks. *Nature Physics* 12, 10 (2016), 901–906.
- [11] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. 2013. Mathematical formulation of multilayer networks. *Physical Review X* 3, 4 (2013), 041022.
- [12] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *Proceedings of the National Academy of Sciences (PNAS)* 113, 3 (2016), 554–559.
- [13] D. Easley and J. Kleinberg. 2010. *Networks, Crowds and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY.
- [14] M. R. Garey and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., San Francisco.
- [15] E. Goles and S. Martinez. 2013. *Neural and automata networks: dynamical behavior and applications, volume 58*. Springer Science & Business Media, New York, NY.
- [16] Mark Granovetter. 1978. Threshold models of collective behavior. *Amer. J. Sociology* 83, 6 (1978), 1420–1443.
- [17] Zaynab Hammoud and Frank Kramer. 2020. Multilayer networks: Aspects, implementations, and application in biomedicine. *Big Data Analytics* 5, 1 (2020), 1–18.
- [18] Yichuan Jiang and JC Jiang. 2014. Diffusion in social networks: A multiagent perspective. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 2 (2014), 198–213.
- [19] Yichuan Jiang, Yifeng Zhou, and Yunpeng Li. 2015. Reliable task allocation with load balancing in multiplex networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 10, 1 (2015), 1–32.
- [20] Michael Kearns. 2007. Graphical Games. In *Algorithmic Game Theory*, Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani (Eds.). Cambridge University Press, New York, NY.
- [21] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, 137–146.
- [22] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. 2014. Multilayer networks. *Journal of Complex Networks* 2, 3 (2014), 203–271.
- [23] Thomas Lefévre, Héléne Colineaux, Claire Morgand, and Laurent Tournois andand Cyrille Delpierre. 2022. AI, Epidemiology and Public Health in the COVID Pandemic. In *Artificial Intelligence in COVID-19*. Springer, New York, NY, 279–338.
- [24] Weihua Li, Quan Bai, Minjie Zhang, and Tung Doan Nguyen. 2018. Modelling multiple influences diffusion in on-line social networks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, Online, 1053–1061.
- [25] Zhaofeng Li, Fuhan Yan, and Yichuan Jiang. 2015. Cross-layers cascade in multiplex networks. *Autonomous Agents and Multi-Agent Systems* 29 (2015), 1186–1215.
- [26] Zakaria Mehrab, Logan Stundal, Samarth Swarup, Srinivasan Venaktramanan, Bryan Lewis, Henning Mortveit, Christopher Barrett, Abhishek Pandey, Chad Wells, Alison Galvani, et al. 2024. Network Agency: An Agent-based Model of Forced Migration from Ukraine. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, Online, 1372–1380.
- [27] Ruchi Mittal and MP S Bhatia. 2017. Mining top-k structural holes in multiplex networks. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, New York, NY, 1–6.
- [28] H. Mortveit and C. Reidys. 2007. *An Introduction to Sequential Dynamical Systems*. Springer Science & Business Media, New York, NY.
- [29] Mark Newman. 2018. *Networks*. Oxford University Press, New York, NY.
- [30] M. Ogihara and K. Uchizawa. 2017. Computational complexity studies of synchronous Boolean finite dynamical systems on directed graphs. *Inf. Comput.* 256 (2017), 226–236.
- [31] Mason A. Porter. 2018. What is a Multilayer Network? *Notices of the AMS* 65, 11 (2018), 1419–1423.
- [32] Zirou Qiu, Abhijin Adiga, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, Richard Edwin Stearns, and Anil Kumar S. Vullikanti. 2024. Efficient PAC Learnability of Dynamical Systems Over Multilayer Networks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, Online, 1–25.
- [33] Zirou Qiu, Baltazar Espinoza, Vitor V Vasconcelos, Chen Chen, Sara M Constantino, Stefani A Crabtree, Luojun Yang, Anil Vullikanti, Jiangzhuo Chen, Jörgen Weibull, et al. 2022. Understanding the coevolution of mask wearing and epidemics: A network perspective. *Proceedings of the National Academy of Sciences* 119, 26 (2022), e2123355119.
- [34] Daniel J. Rosenkrantz, Madhav V. Marathe, Harry B. Hunt III, S. S. Ravi, and Richard Edwin Stearns. 2015. 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, May 4-8, 2015*, Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind (Eds.). ACM, New York, NY, 1501–1509.
- [35] Daniel J. Rosenkrantz, Madhav V. Marathe, Zirou Qiu, S. S. Ravi, and Richard E. Stearns. 2025. *On Some Fundamental Problems for Multi-Agent Systems Over Multilayer Networks*. Technical Report BI-2025-5. Biocomplexity Institute, University of Virginia, Charlottesville, VA, USA.
- [36] D. J. Rosenkrantz, M. V. Marathe, S. S. Ravi, and R. E. Stearns. 2018. Testing phase space properties of synchronous dynamical systems with nested canalyzing local functions. In *AAMAS 2018*. IFAAMAS, Online, 1585–1594.
- [37] Daniel J Rosenkrantz, Madhav V Marathe, S. S. Ravi, and Richard E Stearns. 2024. Synchronous dynamical systems on directed acyclic graphs: Complexity and algorithms. *ACM Transactions on Computation Theory* 16, 2 (2024), 1–34.
- [38] N. K. Sinha. 2015. On the Asymptotic Expansion of the Sum of the First n primes. ArXiv Report 1011.1667v2.
- [39] Viktor Stojkoski, Zoran Utkovski, Elisabeth André, and Ljupco Kocarev. 2018. Multiplex Network Structure Enhances the Role of Generalized Reciprocity in Promoting Cooperation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, USA / ACM, 2082–2084.
- [40] Leslie G Valiant. 1984. A theory of the learnable. *Commun. ACM* 27, 11 (1984), 1134–1142.
- [41] Stanley Wasserman, Katherine Faust, et al. 1994. *Social network analysis: Methods and applications*. Cambridge University Press, New York, NY.
- [42] Duncan J Watts. 2002. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences (PNAS)* 99, 9 (2002), 5766–5771.
- [43] Michael P. Wellman. 2016. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems* 30, 6 (Nov. 2016), 1175–1189.
- [44] Wikipedia Article 2024. Primorials. https://en.wikipedia.org/wiki/Primorial. Access date: Oct. 12, 2024.
- [45] Huiyuan Zhang, Huiling Zhang, and My T Thai. 2016. A survey on multilayer networks and the applications. In *Big Data in Complex and Social Networks*, Matthias Dehmer, Frank Emmert-Streib, Stefan Pickl, and Andreas Holzinger (Eds.). Chapman and Hall/CRC, Boca Raton, FL, 193–222.