Boolean Network Models of Collective Dynamics of Open and Closed Large-Scale Multi-Agent Systems

Predrag T. Tošić[1,*] and Carlos Ordonez[2]

[1] School of EECS and Department of Mathematics & Statistics, Washington State University, Pullman, Washington, USA
[2] Department of Computer Science, University of Houston, Houston, Texas, USA
[*]Contact author. Email: ptosic@eecs.wsu.edu

Abstract. This work discusses theoretical models of decentralized large-scale cyber-physical and other types of *multi-agent systems* (MAS). Arguably, various types of Boolean Networks are among the simplest such models enabling rigorous mathematical and computational analysis of the emerging behavior of such systems and their collective dynamics. This paper investigates determining possible asymptotic dynamics of several classes of *Boolean Networks* (BNs) such as *Discrete Hopfield Networks*, *Sequential and Synchronous Dynamical Systems*, and (finite, Boolean-valued) *Cellular Automata*. Viewing BNs as an abstraction for a broad variety of decentralized cyber-physical, computational, biological, social and socio-technical systems, similarities and differences between open and closed such systems are rigorously analyzed. Specifically, this paper addresses the problem of enumerating all possible dynamical evolutions of large-scale decentralized cyber-physical, cyber-secure and holonic systems abstracted as Boolean Networks. We establish that, in general, the problem of enumerating possible dynamics is provably computationally hard for both "open" and "closed" variants of BNs, even when all of the following restrictions simultaneously hold: i) the local behaviors (node update rules) are very simple, monotone Boolean-valued functions; ii) the network topology is sparse; and iii) either there is no external environment impact on the system, or the model of the environment is of a rather simple, deterministic nature. Our results provide lower bounds on the complexity of possible behaviors of "real-world" large-scale cyber-physical, socio-technical, social and other distributed systems and infrastructures, with some far-reaching implications insofar as (un)predictability of such systems' collective dynamics.

Keywords: Cyber-Physical Systems; Multi-Agent Systems; Boolean Networks; Cellular & Network Automata; Network Dynamics; Systems Science

1 Introduction

Network Science and *Agent-Based Modeling* (ABM) have provided useful abstractions, research methodology, as well as mathematical and computational tools for investigating fundamental behavioral properties of a broad variety of physically and logically decentralized, "networked" systems and infrastructures in engineering, physics, biological sciences and social sciences (see, e.g., [1, 7, 16, 19, 25, 26]). In particular, investigating possible dynamics of discrete, agent-based models has been of a considerable research interest among those studying, designing and/or analyzing various distributed computing infrastructures, as well as various cyber-physical and cyber-secure systems. Both analytical and computational (i.e., simulation-based) studies have been undertaken, often providing valuable insights that would be much harder, or even impossible, to obtain using the "traditional" mathematical and computational methods based on solving appropriate systems of ordinary or partial differential equations analytically whenever possible, or via numerical simulation otherwise.

One important distinction that is often made about such complex distributed systems is, whether a given system is *open* or *closed.* In an open system, there is in general influence of an "environment" external to the explicitly modeled, designed and/or controlled "agents"; and the system designer, who is in charge of controlling or "programing" the behavior of the agents, in general does not exercise control over that environment, how the environment may impact the agents, or indeed how will that environment respond to the agents' actions. In contrast, in a closed system, there is no (relevant) "environment" or other uncontrollable and/or unpredictable sources of impact on the agents in the system, outside of those agents themselves and their own behaviors.

Everything else being equal, in general it is easier to design, analyze, and predict or control behavior of agents in a closed environment than in an open environment. Needless to say, however, most if not all "real-world" cyber-physical, socio-technical, biological and physical systems are in reality open, in the sense that it is rather rare that the system designer or the organization deploying a particular engineered multi-agent system has the full control of all relevant entities, interactions within and influences on that system. However, depending on what properties of the system behavior one is interested in, as well as whether and to what extent the actual external factors impact those properties of interest, when modeling cyber-physical, socio-technical and other decentralized multi-agent systems, assuming the system to be (approximately) closed in order to simplify analysis and/or design may still be justifiable.

Distributed computing and distributed AI researchers have extensively studied interactions, emerging behavior, coordination, resource and task sharing, and other important problems formulated in both open and closed system settings. In some circumstances, it may be appropriate to model a particular multi-agent or cyber-physical system as a closed system – for example, when it's justifiable to assume that no "external" aspects of the environment would have any relevant impact on the agents, their resources, decisions, goals or tasks. On the other hand, designing protocols, algorithms, and other techniques for autonomous software or robotic agents in open environments is, as a rule of thumb, both more realistic and more challenging, esp. when possible impact of the environment on agents, their actions and their goals is complex, nondeterministic and/or only partially observable (see, e.g., [14, 17]).

Communicating Finite State Machines (CFSMs) and *Boolean* (or other discrete-valued) *Networks* are among the most popular mathematical formalisms for a broad range of biological, physical, computational, cyber-physical, socio-technical and other decentralized systems and architectures [2, 3, 15-18, 20, 21]. These models allow for crisp formalizations of many properties of such systems' collective and emerging behavior, especially with regards to long-term or asymptotic dynamics [16, 17, 22, 23]. One typical example is formalizing the fairness, liveness, deadlock avoidance and similar properties of interest when it comes to modeling and verification of distributed computing infrastructures, in terms of the fundamental configuration space properties of a formal discrete dynamical system based on CFSMs or Boolean Networks.

The rest of this paper is organized as follows. In Section 2 we introduce open and closed distributed multi-agent and cyber-physical systems, as well as how they can be formalized in terms of *Boolean Networks* (BNs). We also outline some MAS applications that could benefit from our theoretical analysis of differences in possible emerging behaviors between open and closed systems. In Section 3 we formally define the key configuration space properties of BNs that capture the most important aspects of asymptotic dynamics of those networks. We then focus on the computational complexity of characterizing network dynamics, i.e., answering fundamental questions about those Boolean Networks' configuration spaces. We characterize those problems in both open and closed system settings, and summarize key results for a broad range of sparse BNs with simple local interactions. To the best of our knowledge, no prior work has addressed such comparative analysis of open vs. closed discrete dynamical systems in a formal Boolean Network setting. Last but not least, we summarize the key insights and outline some directions for future research.

2 On Collective Dynamics of Open and Closed Systems

Our goals in this paper are to, first, mathematically formalize open and closed decentralized information processing systems, and, second, establish some key properties of such systems from a unified standpoint of dynamical system theory and computational complexity / (un)predictability. We would like our framework to be sufficiently abstract yet versatile, so that it provides meaningful insights on a broad variety of decentralized information processing systems and distributed infrastructures, ranging from the classical distributed computing environments and cyber-physical systems to social networks and socio-technical systems to biological systems.

In case of the main results in this paper, these insights will for the most part be in the form of *lower bounds* on complexity of an agent ensemble's collective dynamics. By a unified dynamical systems and computational complexity viewpoint, we mean that we want to understand the dynamics of these complex networks (as our abstraction of the real-world cyber-physical, socio-technical, biological and other systems), and in particular to address the computational complexity (that is, relative hardness or easiness) of characterizing that dynamics.

Our choices of mathematical abstractions for the open and closed distributed multi-agent systems are driven by our interests in agent-based modeling and distributed AI on one hand, and network science, on the other – and esp. the cross-fertilization between these two research areas. In particular, the networks we study are characterized by the following properties: time is discrete (and there is an implicit assumption of the existence of a *global clock*, an important premise discussed in detail, e.g., in [16, 19]). Likewise, the states of the individual agents are discrete. For simplicity, we will assume each agent is *binary-valued*, i.e., it can, at any discrete time step t, be in one of two possible states: 0 or 1. The graph or network structure captures, which (pairs of) agents can potentially directly influence each other. Under these ontological commitments, *Boolean Networks* of interconnected *Finite State Machines* are a natural modeling framework [2, 16, 20]. (Of course, other modeling choices are possible – for example, those in which either time, or states of agents, or both, are continuous as opposed to discrete. Closer to our adopted modeling framework, individual agents or nodes are often allowed to be more general finite state machines, so that, in particular, each node may have more than two states. We prefer the Boolean, i.e., binary-valued model for two reasons: (i) it's the simplest non-trivial model w.r.t. the states of individual nodes or agents, and (ii) binary-valued nodes allow for direct comparisons with the rich existing literature on discrete Hopfield networks, cellular automata and other types of binary-valued *Network Automata*.)

How do we differentiate between open and closed multi-agent systems in this modeling framework provided by *Communicating Finite State Machines* (CFSMs) in general, and *Boolean Networks* as a restricted class of CFSMs, in particular? A closed system can be captured by a Boolean Network in which each node is an agent, whose individual behavior is defined, controlled, or at the very least, well-understood by us. In contrast, in an open system abstracted as a CFSM or a Boolean Network, some nodes correspond to agents, whereas other nodes capture the "environment" that influences the agents, and that we do not have control over. The novelty in the present paper is the explicit differentiation between "open" and "closed" CFSMs and BNA, and how the openness (that is, the presence of an "uncontrollable" environment external to the network of agents) impacts some of these fundamental properties of discrete network's dynamics and their configuration spaces.

An alternative approach, studied by researchers working at the intersection of multi-agent systems and statistical physics, is based on a network of agents operating in an external potential field akin to external electric or magnetic or other force fields studied in physics. We will not discuss this latter class of models; for us, every relevant aspect of the world is a node (finite state machine) in the network, but while we have control over the behavior of certain nodes (namely, those representing our "agents"), we do not have control over others (capturing various relevant aspects of "the environment"). We note that the explicit distinction between open and closed multi-agent systems, and some implications of that distinction, has been studied by the Distributed AI researchers (see, e.g., [1]), although, as far as we are aware, not within our formal modeling framework based on Boolean Networks.

2.1 Some Examples of Open and Closed Cyber-Physical Multi-Agent Systems

To provide some real-world "grounding" of the theoretical models of MAS studied in the rest of the paper, we outline some decentralized cyber-physical and/or multi-agent applications in which one can readily differentiate between open and closed systems.

Team Robotics. One popular example is robotic soccer: two teams of robots, each belonging to a different designer, playing soccer against each other. Within a single team, we have a purely cooperative multi-agent system engaging in distributed coordination (see, e.g., [25, 27]). However, from a *Distributed AI* standpoint, the coordination of robots within the same team is more complex than that in purely collaborative, distributed problem-solving settings, as it is done in presence of an adversary – namely, the other team of robots. Importantly, however, the entire "environment" in this application is made of the agents from the two teams of robots (plus the ball, goal posts, and other relevant aspects of the environment that, however, in general do not act deliberately or unpredictably); in particular, there is no "external control" nor an unknown external "environment node" that may unpredictably influence the multi-agent interactions among the robots within a team, or indeed between the two robotic teams. Hence, the robotic soccer is an example of a closed cyber-physical system.

This can be contrasted with, for example, ensembles of autonomous unmanned aerial, underwater or ground vehicles used in a surveillance, search-and-rescue or other type of military or law-enforcement deployment [14, 16, 24]. The physical environments in which such autonomous vehicles operate are, in general, complex and unpredictable; in particular, they often contain other goal-driven agents, possibly including adversarial ones. Unlike with robotic soccer, the behavior of the adversarial and other deliberative and goal-oriented agents is typically not a priori known; likewise, possible impact of external agents and other aspects of the environment on "our" autonomous vehicle agents also in general is not known to the designer of unmanned autonomous vehicles or the organization deploying those vehicles. The natural modeling framework for such ensembles of autonomous unmanned vehicles, therefore, is that of open (cyber-physical, multi-agent) systems.

Traffic Systems. If one considers just the "core infrastructure" such as the signaling system (where it is known, for example, where the traffic lights and other components of a city's traffic signaling system are located, and how all those components function), that would be an example of a closed cyber-physical system. However, when modeling and simulating such traffic systems, the "overall" traffic system is usually considered to also include vehicles, pedestrians and possibly other "agents" whose behaviors, in general, are not a priori known. While one may have a model of possible behaviors, and/or constraints on possible speeds of motion and other relevant aspects of those agents' behaviors, in general, these vehicle and pedestrian agents are more complex and less predictable than say the traffic lights alone. So, in many traffic modeling contexts, the broader traffic system should be considered to be an open system. This is particularly significant when the new agents such as vehicles may unpredictably enter and/or leave the modeled system, thus also making such traffic system open also in the usual distributed computing sense. (We note that the first author's original exposure to the world of agent-based modeling and multi-agent systems, back in the early 2000's, was precisely in the context of defining and analyzing mathematical and computational models for fairly large-scale -- ranging from tens to hundreds of thousands of agents -- urban/metropolitan area traffic simulations [2].)

Epidemics Propagation. Consider spreading of an epidemic, for example, a flu virus, in a community. Accurate models of epidemics propagation are important in public health domain, as well as for designing the best response when faced with outbreak of massive epidemics or a biological warfare attack. Over the past 20+ years, the classical, continuous mathematics based models (specifically, those based on solving differential equations) have been increasingly being replaced by discrete, agent-based models. For a recent work on agent-based modeling of epidemics, that also provides a good survey of the state of the art and methodology, see, e.g., [28].

Many agent-based models of epidemics propagation assume closed systems (for example, a town or city with a "fixed" population). More realistically, however, on practically any scale larger than that of an individual family, the population(s) that may be affected by the epidemics should be viewed as open systems, since new individuals may enter into the population, some of the existing members may leave it, etc.

Depending on the particular aspects of epidemics propagation one wishes to model, however, it may still be justifiable to "treat" the affected population as a closed multi-agent system (with an understanding that, at non-trivial scales, this strictly speaking is hardly ever the case). Therefore, when modeling propagation of epidemics (or opinions, political or social influences, etc.) in a population, sometimes it may be suitable to treat the population in question as a closed system, whereas in other scenarios it is of essence to explicitly take into account the intrinsically open nature of population dynamics problems in most practical scenarios and for all but the smallest of scales (cf. in terms of population sizes).

3 Preliminaries and Definitions

We formulate and then characterize some fundamental properties of *asymptotic dynamics* of open and closed distributed multi-agent systems in the formal setting of *Boolean Networks* and, as their prominent special case, *Discrete Hopfield Networks* [8, 9]. We define those two classes of discrete dynamical systems next.

Definition 1: A *Boolean Network* (also called *Boolean Network Automaton*, or BNA) is a directed or undirected graph so that each node in the graph has a state, 0 or 1; and each node periodically updates its state, as a function of the current states of (some or all of) its neighboring nodes (possibly, but not necessarily, including itself). A BNA dynamically evolves in discrete time steps. If the node v_i has k neighbors denoted $v_{i1}, \ldots, v_{ik}$ (where this list may or may not include v_i itself), then the next state of v_i is determined by evaluating a Boolean-valued function $f_i(v_{i1}, \ldots v_{ik})$ of k Boolean variables; f_i is called the *local update function* or *transition rule* (for the node v_i).

Several comments are in order. First, in general, different nodes v_i may use different local update functions f_i. This applies to *Discrete Hopfield Nets* [4, 5, 8, 9], as well as many other classes of Boolean Networks including those originally introduced by S. Kauffman in the context of systems biology [6, 10], and also several related models proposed in the context of modeling large-scale distributed computing and other decentralized cyber-physical infrastructures [2, 3, 11, 18]. Classical *Cellular Automata* (CA) can then be viewed as a special case of BNA, where all the nodes use the same local update rule f_i [20]. (We note that the underlying graphs in BNA are almost always assumed to be finite, whereas Cellular Automata have been extensively studied in both finite and infinite settings.)

The individual node updates can be done either synchronously in parallel, or sequentially, one at a time (and if so, either according to the fixed update ordering, or in a random order). While other communication models are worth considering [19, 21], the above three possibilities have been studied the most. In this paper, we will focus entirely on the parallel, perfectly synchronous node updates. This means, the next state of the node v_i is determined according to

$$v_i^{t+1} \leftarrow f_i(v^t_{i1}, \ldots, v^t_{ik}) \quad (1)$$

The tuple of all f_i's put together, $F = (f_{1, \ldots,} f_n)$, denotes the *global map* that acts on (global) configurations of a BNA. When all f_i are the same, the notation in the literature is often abused so that no differentiation is made between the local transition function, acting on a state of a single node, and the global map F, acting on entire configurations of a cellular or network automaton (that is, on all the nodes). In classical CA, all nodes update according to the same local update rule. In most other Boolean Network models, different nodes in general update according to different rules.

Definition 2: A *Discrete Hopfield Network* (DHN) is made of n binary-valued nodes. Associated to each pair of nodes (v_i, v_j) is (in general, real-valued) their weight, w_{ij}. The weight matrix of a DHN is defined as $W = [w_{ij}]_{i,j=1..n}$. Each node also has a fixed real-valued threshold, h_i. A node v_i updates its state x_i from time step t to step t + 1 according to a (binary-valued) *linear threshold function* of the form

$$x^{t+1}_i \leftarrow \text{sgn}(\sum w_{ij} \cdot x^t_j - h_i) \qquad (2)$$

where the summation is over j = 1, …, n; the term h_i is the *threshold* that the weighted sum needs to reach or exceed in order for the node's state to update to +1; to break ties, we define sgn(0) = +1.

The default notation in most of the literature on Hopfield networks is that the binary states of an individual node are {-1, +1}. In this paper, however, we adopt the Boolean values {0, 1} for the states of our nodes, in order to be able to discuss DHNs and our results about them in the broader context of Boolean Networks (see, e.g., [6, 10, 19]) without the need for cumbersome "translations". Furthermore, in most of the existing literature on DHNs (e.g., [5, 8, 9, 12, 13]), two additional assumptions are usually made, namely, that (i) the diagonal elements of weight matrix W are all zeros: $w_{ii} = 0$; and (ii) the weight matrix is symmetric, $w_{ij} = w_{ji}$ for all pairs of nodes i, j. We will adopt (ii) throughout (we note, this does not affect the main results and insights from them discussed in the next section). As for (i), we will consider two possibilities on the nodes' "memory" (of their own current state, as a part of the local transition rule): either $w_{ii} = 0$ for all nodes v_i, or else $w_{ii} = 1$ for all v_i. The main results in the next section hold under either the *memoryless* ($w_{ii} = 0$) or *with memory* ($w_{ii} = 1$) assumption. Moreover, in the memory case, our results can be readily extended to more general weighing w_{ii} of how is a node's state at time t+1 affected by its own state at time t; these variations will be discussed in an expanded, journal version but are left out of this paper due to space constraints.

We study a variety of Boolean Network models and their asymptotic dynamics. Several of our main results are formulated in the DHN context [22]; some of our prior work was formulated in the context of two other types of Network or Graph Automata (possibly but not necessarily Boolean), called *Sequential and Synchronous Dynamical Systems* [2, 15]. Hopfield Networks were originally inspired by biology and especially computational neuroscience (in particular, they were introduced as a model of *associative memory* [8]). Subsequently, in addition to theoretical models in computational biology and neuroscience, Hopfield Networks (both discrete and continuous) were used for as a connectionist, self-organizing map model for "learning" and "searching for a solution", i.e., as a powerful tool for various search and optimization problems in computer science, operations research and beyond [9].

We note that some of the earliest Boolean Network models were also originally introduced in the context of theoretical and systems biology, albeit not specifically neuroscience. Indeed, the very name *Boolean Networks* comes from the seminal work in theoretical biology by S. Kauffman [10]. In contrast, Sequential and Synchronous Dynamical Systems (SDS and SyDS, resp.) were specifically introduced in the context of agent-based simulation of complex cyber-physical, socio-technical and engineering systems [3, 11, 15, 16, 17]. For clarity and space constraints reasons, we will not formally introduce S(y)DS models here, but rather refer the reader to relevant references. We emphasize that our results in this paper apply to all of the above models (DHNs, SDSs and SyDSs), and indeed most other discrete-time Boolean (or other finite-domain) Networks or BNA found in the existing literature.

Since BNA and DHN are deterministic discrete-time dynamical systems, for any given current configuration C^t at time t, there is a unique next-step configuration C^{t+1}. We can therefore define the BNA or DHN *configuration or phase spaces*, and also various types of *global configurations* (i.e., tuples capturing the states of all nodes in a network) of interest:

Definition 3: A (global) configuration of a cellular or network automaton or a discrete Hopfield Network is a vector $(x_{1,\ \ldots,}\ x_n)\ \varepsilon\ \{0,1\}^n$, where x_i denotes the state of the i^{th} node. A global configuration can also be thought of as a function γ: V → {0,1}, where V denotes the set of nodes in the underlying graph of a CA, BNA or DHN.

Definition 4: A *fixed point* (FP) is a configuration such that, once a BNA or DHN reaches that configuration, it stays there forever. A *cycle configuration* (CC) is a global state that, once reached, will be revisited infinitely often with a fixed, finite temporal period of 2 or greater. A *transient configuration* (TC) is a global configuration that, once reached, is never going to be revisited again.

Definition 5: Given two configurations C and C' of a CA, BNA or DHN, if F(C) = C' then C' is the *successor* of C (and C is a *predecessor* of C'). That is, configuration C' is reached from configuration C by a single application of the global map.

If the dynamics of a BNA or DHN is deterministic (which we shall assume throughout this paper), then each configuration has a unique successor. However, a configuration may have 0, 1 or more predecessors. By the "pigeonhole principle", it then follows that the global dynamics of a (deterministic) BNA or DHN is invertible iff each configuration has exactly one predecessor.

Definition 6: A configuration with no predecessors is called *Garden of Eden*. Lastly, configuration A is an *ancestor* of configuration C, if starting from A, the dynamics reaches configuration C after finitely many time steps (equivalently, if there exists $t \geq 1$ such that $F^t(A) = C$).

In particular, a predecessor is a special type of an ancestor. Further, "fixed points" are the only type of configurations such that each is its own predecessor. Similarly, each cycle configuration is its own ancestor. In contrast, due to determinism, a TC can never be its own ancestor.

A BNA is called *dense* if the underlying graph on which it is defined is dense. Similarly, a DHN is dense if its weight matrix W is dense, i.e., if W contains many non-zero entries [22, 23]. The natural interpretation of a zero weight $w_{ij} = 0$ in a DHN is that the corresponding nodes i and j do not directly affect each other. (That is, change of state of the i^{th} node does not immediately affect the state of the j^{th} node, and vice versa; of course, they can still indirectly affect each other, via connected paths in the underlying graph whose edge weight products are nonzero). In contrast, we say that a BNA is *sparse* if the underlying network topology (that is, the graph structure) is sparse -- which, for us, will mean that $|E| = O(|V|)$. That is, for our purposes, sparseness means O(1) neighbors per node (alternatively, only O(1) non-zero weights per row of the weight matrix W), on average; that is, the total number of edges in the underlying graph (equivalently, the total number of non-zero entries in W) is of the order $O(n)$ where n is the number of nodes.

Further, we call a Boolean Network or a Hopfield Network *uniformly sparse* if every node is required to have only O(1) neighbors (that is, every row/column in W has only O(1) non-zero entries). So, for example, a star or a wheel graph on n nodes would be sparse (the average node degree being O(1) in each case), but neither of those types of networks would be uniformly sparse, as the center of a star or a wheel has $\Theta(n)$ neighbors.

What is the relationship between configuration space properties of a formal BNA, DHN or CA model, and behaviors of real-world cyber-physical and multi-agent systems? Consider, for instance, Gardens of Eden (GEs): these configurations can only occur as initial states of the system. Hence, if one can show that all undesirable or dangerous configurations of, for example, a real-world cyber-secure system are GEs, then, as long as one can ensure that the system does not start in one of those dangerous states, it is safe to assert that the system will never reach any of those "bad" states. Similarly, knowing that all configurations, or more commonly in practice all members of an appropriately defined subset of global configurations satisfying certain pre-specified properties, are actually all recurrent states, would imply that certain fairness and liveness properties for that (sub)set of configurations must hold.

Reachability properties (whether certain types of configurations are reachable from a subset of initial configurations of interest) have been connected to *fairness* in the distributed computing sense. What is also often of interest, is the speed of convergence of a system to its stationary behavior (be it of a temporal cycle or fixed point variety); that speed or rate of convergence can be formally related to the depth of "the basin of attraction" of a fixed point of temporal cycle in question, thus establishing a formal connection between the system dynamics and the distributed computing perspectives. Likewise, enumerating exactly or approximately all "initial states" leading to a given FP or temporal cycle captures the overall size of the basin of attraction.

Last but not least, being able to *enumerate* the total number of (non-trivial) temporal cycles and stable ("fixed point") configurations of a deterministically behaving system is in essence equivalent to knowing in how many ways that system can evolve, for all possible initial configurations. It has been known since the 1990s that certain types of DHNs and CA cannot have non-trivial temporal cycles, implying that the only recurrent states are fixed points. For such systems, enumerating the FPs is therefore synonymous with determining the total number of possible asymptotic behaviors.

4 Configuration Space Properties and Asymptotic Dynamics of Open and Closed Cyber-Physical and Holonic MAS

We now summarize some of the key insights about several fundamental problems about Discrete Hopfield Networks and other types of Boolean Network Automata. Most of these results describe the worst-case computational complexity of determining key configuration space properties of various classes of such networks. Examples of such configuration space properties include: i) determining the existence of fundamental types of configurations such as stable or fixed-point states (FPs), Cycle Configurations (CCs) or Gardens of Eden (GEs); ii) determining the exact or approximate number of fundamental types of configurations such as FPs, CCs or GEs; iii) answering questions about reachability of FPs or of a particular configuration, from a specific initial state or a set of initial states; iv) answering questions about whether a given DHN's or other BNA's dynamics is invertible. These and related questions about CFSM and BNA dynamics have been studied (by one of the authors as well as other researchers) since at least 2001; we summarize and interpret some of the main results, applied to *closed* cyber-physical and multi-agent systems, in the next subsection.

4.1 Dynamics and Configuration Spaces of Boolean Networks Modeling Closed Distributed Multi-Agent Systems

All models of BNA, DHNs and CFSMs we have been studying are deterministic, implying that, regardless of the details of the local update rules, the "underlying topology" (that is, the graph structure), and the particular choice of starting configuration, asymptotically the system will either eventually reach a fixed point or a temporal cycle of length 2 or greater. But can we in general tell which of these two ultimate outcomes will take place? It turns out, differentiating between these scenarios is in general computationally intractable:

Theorem 1: Determining whether an arbitrary Boolean Network or Boolean-valued CFSM, starting from an arbitrary initial configuration, will eventually evolve to a FP or a non-trivial temporal cycle is in general **PSPACE-**complete.

More generally, most non-trivial *Reachability* problems for the sufficiently general classes of BNA (such as the aforementioned SDSs and SyDSs, and other similar classes of Boolean-valued, as well as more general, CFSMs), for sufficiently general local update rules and underlying graphs, are in the worst-case **PSPACE**-complete. The original motivation, formulations and proofs of these reachability results, in the context of (Boolean-valued) Sequential and Synchronous Dynamical Systems as two special subclasses of BNA, can be found in [3] and references there.

Corollary 1: Determining the "ultimate destiny" of a deterministic closed discrete dynamical multi-agent system in which each agent is a 2-state FSM (in terms of differentiating whether that destiny will be stability or oscillation with a fixed periodicity) is **PSPACE**-complete.

Moreover, even answering the basic questions about the existence of FPs and other fundamental types of configurations is, in the worst case, computationally intractable, although these existence problems lie much lower in the computational complexity hierarchy than the related reachability problems:

Theorem 2: Given the description of an arbitrary BNA or CFSM, determining whether it has any FP configurations is **NP-**complete. Similarly, in general, determining if a BNA or CFSM has any non-trivial temporal cycles is **NP**-hard.

Analogous results hold about the fundamental problems about the BNA *inverse dynamics* – that is, about hardness of characterizing dynamical behavior when the direction of time is reversed, as summarized in the next theorem:

Theorem 3: Given the description of an arbitrary BNA or Boolean-valued CFSM, determining whether it has any TC or GE configurations is **NP-**complete. Given such a BNA or CFSM and an arbitrary configuration, determining whether that configuration is a GE is **coNP**-complete.

For proofs of the original formulations of results summarized in **Theorems 2-3**, see [2] and references therein. Note that in the first part of **Theorem 3**, the problems about (arbitrary) *Transient Configurations* (TCs) on one hand, and *Gardens of Eden* (GEs) on the other, are fundamentally equivalent: a BNA or CFSM or DHN has a TC if and only if it has a GE [2]. On the other hand, validating whether a configuration is a GE is readily seen to be in class **coNP** (since the complementary problem, viz. whether a given configuration has a predecessor, is clearly in **NP**), whereas determining whether a given configuration is an a TC (but not necessarily GE) is less obvious; however, this problem is certainly **coNP**-hard in the worst-case.

There are, however, important subclasses of both local update rules and underlying graphs, for which the fundamental problems about FPs, CCs, TCs and GEs are actually computationally tractable. In particular, if all nodes of a BNA or a CFSM update according to *monotone* Boolean-valued update rules, then the existence of fixed points is guaranteed:

Theorem 4: If every local update rule in a BNA or CFSM is a *monotone* Boolean-valued function, then the problem of FP existence is computationally easy: such a BNA or CFSM is guaranteed to have at least one FP.

Corollary 2: Discrete Hopfield Networks all of whose edge weights are non-negative are guaranteed to have FPs.

Computational problems about BNA and CFSM configuration spaces that we have investigated in greatest detail pertain to the computational complexity of counting. That counting all FPs or all CCs or all GEs of an arbitrary Boolean Network would turn out **#P**-hard, is to be expected. What is more interesting, however, is that this hardness of counting remains to hold even for severely restricted classed of BNs and CFSMs, with restrictions applying simultaneously to both the graph structures and the local update rules. In particular, we have the following results:

Theorem 5: Exactly enumerating all FPs of a Boolean Network Automaton (such as Boolean-valued SDSs, SyDSs and DHNs) is **#P**-complete, even when all local node update rules are symmetric Boolean functions, and the underlying graph is sparse (as in, sparse on average, or even uniformly).

Theorem 6: Exactly enumerating all FPs of a BNA (such as SDSs, SyDSs and DHNs) is **#P**-complete, even when all local node update rules are monotone functions, and the underlying graph is sparse (on average or uniformly).

For details on various types of BNA with symmetric and/or monotone update rules, and various classes of either sparse-on-average or uniformly sparse underlying network topologies, we refer the reader to [15, 18, 20]. Among the sparse-on-average graphs, we particularly focus on the star and wheel graphs, cf. because of their implications to open systems, that is, BNA embedded in and interacting with an external environment.

Theorem 7: The following enumeration problems are all **#P**-complete, even when the underlying graphs of a BNA or DHN are restricted to star-graphs (or wheel-graphs), and all local update rules are monotone Boolean functions:

- Determining the exact number of all FPs;
- Determining the exact number of all TCs;
- Determining the exact number of only those TCs that are Gardens-of-Eden;
- Determining the total number of predecessors of an arbitrary configuration.

Detailed discussion and full formal proof of **Theorem 7** can be found in [18]. Those results have been further refined and strengthened in [20].

4.2 Dynamics of Boolean Networks Modeling Open Multi-Agent Systems

To the extent that cyber-physical systems, multi-agent systems and other decentralized infrastructures can be adequately modeled by these BN and CFSM models, all results in the previous subsection pertain to *closed* such systems or infrastructures: the ones whose behavior isn't affected by anything other than the individual behaviors of agents themselves (i.e., individual nodes' local update rules) and the interaction patterns (i.e., how are these agents interconnected with each other). In contrast, an open system is one in which there's an environment, external to the agents, that in general may also impact the agents' behaviors. From a control theory standpoint, this openness of the system, i.e., a potential impact of an external environment on the agents and their individual and collective dynamics, can be modeled by adding to a BN or CFSM an additional, "environment node" that (in general) is connected to, and therefore may influence, the behavior of all (individual) agent nodes. Assuming "everything else [being] equal", all hardness results about the closed dynamical multi-agent systems in the previous subsection imply similar hardness results for open multi-agent systems. In particular, we have the following results:

Theorem 8: Determining whether an open deterministic multi-agent system will eventually reach stability or non-trivial oscillatory behavior is, in general, **PSPACE**-hard. Moreover, this problem is **PSPACE**-complete if the external control (the "environment node") is behaving according to a deterministic, Boolean-valued function.

Theorem 9: Computational problems of deciding whether an open deterministic multi-agent system's dynamics has any FPs, any unreachable (GE) configurations, or any transient configurations, are in general **NP**-hard. If the environment's behavior is known and can be represented as a deterministic Boolean-valued function, these problems are in the class, **NP** and are therefore **NP**-complete in the worst-case.

Arguably the most interesting consequences of previous complexity hardness results "translated" from closed to open discrete dynamical systems, are those in the context of counting. That is, complexity of counting FPs and other types of configurations in closed discrete dynamical systems, as established in [17, 18, 20], have direct implications for the open distributed multi-agent systems, even those have very restricted agent interaction patterns, such as when the underlying graph is a simple path or ring (i.e., cycle in the graph-theoretic sense). Concretely, the following results hold:

Theorem 10: The following counting problems are all **#P**-complete, even when the underlying topologies of a discrete dynamical system are restricted to simple paths and rings, all agents behave according to monotone Boolean update rules, and additionally, the environment is known to dynamically evolve according to a monotone Boolean function (that, in general, may depend on current states of all individual agents):

- Determining the exact number of all FPs;
- Determining the exact number of all TCs;
- Determining the exact number of only those TCs that are Gardens-of-Eden;
- Determining the total number of predecessors of an arbitrary configuration.

These results are immediate consequences of **Theorem 7**, when the center of the wheel graph corresponds to the external control (that is, the environment), whereas the peripheral nodes correspond to the agents interconnected with each other into a ring (or path), and so that each agent locally updates according to a monotone Boolean-valued function. The only allowable inputs to each agent's local update rule are the states of the neighboring nodes, possibly the current state of the node in question itself, and the "environment node".

5 Conclusions

In summary, it follows from our results that characterizing most non-trivial properties about possible dynamics of open distributed multi-agent systems is computationally intractable, even for the simplest interaction patterns among the agents, as well as very simple, deterministic local behaviors of individual agents. While intractability holds for many closed systems, as well, in case of the open systems it appears hard to find a non-trivial such system so that its dynamics is tractable, even under considerable restrictions on how can the "environment" influence the agents. The only such systems for which tractability of dynamics has be proven to hold are the CA with restricted types of update rules (such as the simple threshold functions); importantly, in classical (finite) CA, all nodes update according to the same rule. It turns out that adding even a rather modest amount of heterogeneity to local agent interactions (such as, considering BNA whose nodes use two different update rules from the same restricted class of Boolean-valued functions), essentially immediately results in systems whose asymptotic dynamics are in general computationally intractable (see [23] for details).

Moreover, our recent research, and in particular results on the computational complexity of counting fixed points of Discrete Hopfield Networks and related Boolean Network models (see, e.g., [22]), immediately imply that determining the exact or even approximate number of possible asymptotic behaviors of complex networks abstracting various multi-agent, holonic and cyber-physical systems, is in general intractable even when the individual agent behaviors and their interactions are severely restricted – including the allowable models of the environment and its impact on the agents. We suspect most of other fundamental questions about dynamics of open distributed systems are also intractable in the worst-case, likely including some interesting scenarios where those questions are tractable for closed systems.

In particular, we have some evidence to believe that the restricted classes of underlying network topologies studied in [18] provide a good candidate starting point for identifying some of the scenarios that are likely to exhibit a stark contrast in the behavioral complexity between closed and open systems. The networks studied in that paper include star and wheel graphs, as well as other types of network topologies that are sparse-on-average, and based on bipartite and/or planar graphs. Validating this intuition for the non-trivial classes of Boolean Network models whose dynamics in the closed system case are actually tractable (for some examples of closed systems whose practically all interesting aspects of dynamics can be characterized computationally efficiently, see e.g. [19, 21]), yet becomes unpredictable in the open system setting even under very restricted models of the "external environment", is the subject of our ongoing and future work.

References

1. S. Bandini, S. Manzoni & C. Simone. "Heterogeneous Agents Situated in Heterogeneous Spaces", *Applied Artificial Intelligence: An Int'l Journal,* vol. 16, issues 9-10, pp. 831-852, Taylor & Francis, 2002
2. C. Barrett, et al. "Gardens of Eden and Fixed Points in SequentialDynamical Systems", *Discrete Mathematics & Theoretical Comp. Science* (DMTCS), vol. AA, pp. 95-110, 2001
3. C. Barrett, et al. "Reachability problems for sequential dynamical systems with threshold functions", *Theoretical Computer Science*, vol. 295, issues 1–3, pp. 41–64, 2003
4. N. Davey, L. Calcraft, R. Adams. "High capacity, small world associative memory models", *Connection Science*, vol. 18 (3), pp. 247-264, 2006
5. P. Floreen, P. Orponen. "On the Computational Complexity of Analyzing Hopfield Nets", *Complex Systems*, vol. 3, pp. 577–587, 1989
6. A. Graudenzi et al. "Dynamical properties of a Boolean model of a gene regulatory network with memory", *J. Comp. Biology*, vol. 18 (10), 2011
7. D. Helbing. "*Social Self-Organization*", Understanding Complex Systems series, Springer-Verlag, Berlin, 2012
8. J. Hopfield. "Neural networks and physical systems with emergent\collective computational abilities", *Proc. Nat'l Academy Sciences*, vol. 79, pp. 2554-2558, 1982
9. J. Hopfield, D. Tank. "Neural computation of decisions in optimization problems", *Biological Cybernetics*, vol. 52, pp. 141-152, 1985
10. S. A. Kauffman. "Emergent properties in random complex automata", *Physica D: Nonlin. Phenomena*, vol. 10, issues 1–2, pp. 145–156, 1984
11. H. Mortveit, C. Reidys. "Discrete sequential dynamical systems", *Discrete Mathematics*, vol. 226 (1–3), pp. 281–295, 2001
12. P. Orponen. "Computational Complexity of Neural Networks: A Survey", *Nordic Journal of Computing*, 1996
13. J. Sima, P. Orponen. "General-Purpose Computation with Neural Networks: A Survey of Complexity Theoretic Results", *Neural Computation*, vol. 15 (12), pp. 2727-2778, 2003
14. P. Tosic et al. "Modeling a System of UAVs on a Mission", invited session on agent-based computing, *Proc. 7th World Multiconference on Systemics, Cybernetics, and Informatics* (SCI'03), pp. 508-514, 2003
15. P. Tosic, G. Agha. "On Computational Complexity of Counting Fixed Points in Symmetric Boolean Graph Automata ", Proc. 4th Int'l Con. Unconventional Computing (UC'05), Seville, Spain, LNCS vol. 3699, pp. 191-205, Springer, 2005

16. P. Tosic. "Cellular Automata for Distributed Computing: Models of Agent Interaction and Their Implications", Proc. Int'l Conf. Systems, Man & Cybernetics (SMC'05), pp. 3204 – 3209, IEEE, 2005
17. P. Tosic. "On Modeling and Analyzing Sparsely Networked Large-Scale Multi-Agent Systems with Cellular and Graph Automata", Comput. Science ICCS-2006: 6th Int'l Conf. (Proc. Part III); V. Alexandrov et al. (eds.); LNCS vol. 3993, pp. 272-280, Springer, 2006
18. P. Tosic. "On the Complexity of Counting Fixed Points and Gardens of Eden in Sequential & Synchronous Dyn. Systems", *Int'l J. Foundations Comp. Sci.* (IJFCS), vol. 17 (5), pp. 1179-1203, World Scientific, 2006
19. P. Tosic. "Cellular Automata Communication Models: Comparative Analysis of Parallel, Sequential and Asynchronous CA with Simple Threshold Update Rules". *Int'l Journal of Natural Computing Research* (IJNCR), vol. 1 (3), pp. 66-84, 2010
20. P. Tosic. "On the complexity of enumerating possible dynamics of sparsely connected Bool. network automata with simple update rules", *Discete Mathematics and Theoretical Computer Science* (DMTCS), pp. 125-144, 2010
21. P. Tosic. "Modeling Large-Scale Multi-Agent Systems with Sequential and Genuinely Asynchronous Cellular Automata", *Acta Physica Polonica B* (Proc. Supplement), vol. 4, No. 2, pp. 217-236, Polish Academy of Sciences, 2011
22. P. Tosic**.** "On Simple Models of Associative Memory: Network Density Is Not Required for Provably Complex Behavior", Proc. Brain Informatics and Health, *Lecture Notes in Computer Science* (LNCS) series, vol. 9919, pp. 61-71, Springer, 2016
23. P. Tosic. "On Phase Transitions in Dynamics of Cellular and Graph Automata Models of Sparsely Interconnected Multi-Agent Systems", ACM Proc. *Autonomous Agents & Multi-Agent Systems* (AAMAS'17), Sao Paulo, Brazil, May 2017
24. P. Tosic, G. Agha. "Understanding and Modeling Agent Autonomy in Dynamic Multi-Agent, Multi-Task Environments", Proc. First European Workshop on Multi-Agent Systems (EUMAS'03), Oxford, England, UK, 2003
25. P. Tosic, C. Ordonez. "Distributed Protocols for Multi-Agent Coalition Formation: A Negotiation Perspective", Proc. Active Media Technologies (AMT'12), pp. 93-102, Springer *Lecture Notes Comp. Science* (LNCS) vol. 7669, R. Huang et al. (eds.), 2012
26. J. Garcia, C. Ordonez, P. Tosic. "Efficiently repairing and measuring replica consistency in distributed databases", *Distr. and Parallel Databases* vol. 31(3), pp. 377-411, 2013
27. S. Vig, J. A. Adams. "Issues in multi-robot coalition formation", Proc. Multi-Robot Systems: From Swarms to Intelligent Automata, vol. 3, 2005
28. M. Zhang. "Large-scale Agent-based Social Simulation - A study on epidemic prediction and control", PhD Dissertation, TU Delft, The Netherlands, 2016