Phase Transitions in Possible Dynamics of Cellular and Graph Automata Models of Sparsely Interconnected Multi-Agent Systems

Predrag T. Tošić
School of EECS, Washington State University
Pullman, Washington, USA
pedja.tosic@gmail.com

ABSTRACT

We are interested in *Communicating Finite State Machine* (CFSM) based models of large-scale multi-agent systems and their emerging behavior. CFSM-based models are suitable for studying large ensembles of simple reactive agents, and collective dynamics of such ensembles. In this paper, we focus on the asymptotic dynamics of a class of the classical (finite) *Cellular Automata* (CA) and more general *Network or Graph Automata* (GA). We restrict our attention to CA and GA with Boolean-valued *linear threshold functions* as the node update rules, inspired by well-known models of biological neurons. The linear threshold update rules on which we focus the most are the Boolean-valued functions that do not allow for negative weights. We fully characterize the configuration spaces of such simple threshold CA, with a focus on the Majority update rule that results in the most interesting dynamics among the CA in this class. In particular, we show the combinatorics behind determining the total number of fixed point configurations for simple threshold CA. Even when the combinatorics is non-trivial, such as in the case of Majority update rule, the counting problems of interest are computationally tractable. We then discuss a stark contrast with respect to intractability of counting for the related classes of Boolean graph automata with the same restrictions on the node update rules. The GA with proven complex dynamics have only slightly more complex structures when it comes to (i) the underlying interconnection topologies ("cellular spaces") and (ii) the diversity of the node update rules, i.e., whether all the nodes use the same update rule (as in the classical CA), or just two different rules from the given class are allowed.

Keywords

Emerging behavior in MAS, mathematical models of MAS, cellular automata, graph automata, asymptotic dynamics, configuration space properties, complexity of counting

1. INTRODUCTION

We study theoretical models of large-scale *multi-agent systems* (MAS) and their collective dynamics. In that setting, we have undertaken a qualitative and quantitative analytical study of *emerging behavior* and, in particular, *asymptotic dynamics* of certain types of network-based dynamical systems. Researchers across a range of disciplines, from theoretical biology to artificial life to statistical physics to cyber-physical/cyber-secure systems to the "traditional" (large-scale) multi-agent systems have studied dynamical properties of various models of discrete complex networks in general, and models based on *communicating finite state machines* (CFSMs), in particular. Prominent among such network and CFSM models are (finite) cellular automata (CA) and several of their graph or network automata extensions. Boolean-valued CA, and some of their generalizations, are the models on which we focus in this paper.

We investigate various *configuration space properties* of cellular and graph automata, as well as the computational (in)tractability of determining those properties. We are particularly interested in the *counting problems*: how many "fixed point", temporal cycle, unreachable ("garden of Eden") or other types of configurations of interest a discrete dynamical system such as a CA may have, and how hard are the computational problems of enumerating those various types of configurations. The motivation for this undertaking is, that such results provide lower-bounds on the complexity of *"real-world"* collective dynamics of large ensembles of simple interactive agents. In the context of interesting enumeration problems about dynamics of various cellular and graph automata, it has been demonstrated that both exact and approximate counting of various types of configurations in two prominent classes of Boolean network automata, called Sequential and Synchronous Dynamical Systems (SDSs and SyDSs, respectively), are in general, demonstrably computationally intractable. Similar general intractability holds for Discrete Hopfield Networks (DHNs) [43]. Interestingly (and less expectedly), the computational intractability holds even when the structures of the underlying graphs as well as the node update rules are severely restricted [46, 39, 49]. Moreover, computational intractability of counting stable of fixed-point configurations has been established to hold even when the underlying graphs are required to be *uniformly sparse* [50].

A broader objective of our research on configuration space properties of cellular and network automata is to apply these insights to behavior analysis and deeper understanding of interesting phenomena in the contexts of large-scale distributed cyber-physical and computational infrastructures, distributed AI and multi-agent systems, as well as biologi-

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

cal networks (including, but not limited to, those arising in computational neuroscience). In particular, several classes of Boolean (or other discrete-valued) *network* or *graph automata* have been proposed as a suitable abstraction of the classical networked distributed systems, various multi-agent systems and *ad hoc* networks, as well as a theoretical model for the (agent-based) computer simulation of a broad variety of computational, physical, biological, cyber-physical, and socio-technical decentralized systems (see, e.g., [3, 30, 41]. There has been a considerable research effort over the past 15-20 years on analytical and simulation-based investigation of such Boolean cellular, network and graph automata; see, e.g., [3, 2, 7, 6, 45, 46, 49]. The general approach has been to study mathematical and computational *configuration space properties* of such network or graph automata: what are the possible *global behaviors* of the entire system, given the simple local interactions of its components.

We focus in the present paper on the problem of determining *how many* asymptotically stable configurations such cellular and network automata have, and *how hard* is the computational problem of *counting* those configurations. We establish that counting FP configurations in classical *Simple Threshold Cellular Automata* is actually computationally tractable, and outline combinatorial analysis for this problem. In contrast, the corresponding counting problems for seemingly only slightly more complex class of network automata such as *Sequential* and *Synchronous Dynamical Systems* have been shown computationally intractable (e.g., [49, 50, 46]). In other words, for the infrastructures that can be modeled as networks of interacting *finite state machines* (FSMs), how complex their behaviors can get critically depends on how exactly different simple components interact with one another, and whether the network is homogeneous across the nodes, or different nodes are allowed to behave even slightly differently from each other. The broader picture of our results is that two interesting phase transitions in the complexity of dynamics of Boolean Networks can be identified. One, for the simple threshold Cellular Automata (corresponding to homogeneous ensembles of reactive agents), the number of possible dynamics is small for all agent update rules *other than* the Majority function; and in the latter case, there are exponentially many possible asymptotic dynamics. Two, predicting asymptotic dynamics of such homogeneous systems is always computationally feasible. However, we have then established that, when even the smallest degree of heterogeneity is added to individual agent behaviors and inter-agent interaction patterns, in several such settings the worst-case complexity of predicting asymptotic dynamics becomes intractable for nontrivial ensemble sizes. The main conclusion is that even a very modest differences with respect to MAS (non)uniformity can lead to a major phase transition from imple collective dynamics that are "easy in principle" to fully characterize, to very complex behaviors that are provably computationally intractable to predict.

The rest of this paper is organized as follows. In the next section, we elaborate on the motivation behind the problems we address. Section 3 is devoted to the necessary preliminaries on Boolean networks, cellular automata and graph automata studied in this paper, and and their configuration space properties of interest. Subsequently, in Section 4 we briefly review some classical work on computational problems about CA and related models, as well as the most relevant recent prior art. Our main contributions are presented after the prior arts review. Finally, we summarize our results and draw some broader conclusions in Section 6.

2. DYNAMICS OF CELLULAR AUTOMATA & BOOLEAN NETWORKS

Most of biological, social and socio-technical systems are inherently decentralized and distributed. An example are human and other advanced animals' brains. Fully decentralized large-scale systems are growing in their number and importance among various engineering and other man-designed infrastructures, as well. In particular, various computational and communication systems and networks are getting increasingly distributed both logically and physically. The sophistication and complexity of most such systems does not stem from the sophistication of their individual components, since functioning of those components is typically well-understood. Rather, the challenges of effective analysis of and forecasting about the behavior of such systems are primarily due to nontrivial interactions among the individual components at the system level.

In order to understand the global behavioral properties of these and many other computational, physical, socio-economic, and socio-technical distributed infrastructures, and to be able to at least sometimes and at least approximately *predict* their long-term dynamic behavior patterns, it seems natural to apply the methodology, tools and paradigms from the study of *discrete dynamical systems*. From a computational perspective, the standard questions posed in the distributed computing context, such as those related to various *liveness, fairness* and *safety* properties, the problems of *reaching distributed consensus* among the computational agents, and the like, can be appropriately formally phrased in terms of the basic *configuration space properties* of the corresponding formal dynamical system (see, e.g., [16, 40]).

To be able to predict the long-term behavior of various decentralized systems and infrastructures, one may want to, first, abstract those infrastructures and translate them into formal dynamical systems, and, second, answer a kind of questions like the ones above within the formal framework of those dynamical systems. The computational hardness of these idealized configuration space problems would then provide *lower bounds* on analyzing the dynamics and emergent behavior of the actual computational and communication networks as well as other kinds of distributed infrastructures, and on how (un)predictable their long-term behavior can be expected to be. That is, formal *computational intractability* of an idealized configuration space problem defined for an appropriate class of cellular, graph or network automata viewed as *discrete time, discrete space* dynamical systems would imply that, in general, the long-term behavior of the corresponding actual multi-agent system or other distributed infrastructure cannot be reliably predicted. For such systems, under the usual assumptions in computer science, such as that $\mathbf{P} \neq \mathbf{NP}$ and $\mathbf{P} \neq \#\mathbf{P}$ [14], there is no short-cut to a step-by-step system execution – or, from a modeling and simulation standpoint, to a step-by-step computer simulation [7, 40].

3. PRELIMINARIES AND DEFINITIONS

We now formally introduce cellular automata and their configuration space properties. We focus on a restricted, yet interesting class of CA, namely, the *simple threshold cellular automata* [44, 45, 48]. In that context, we study (i) characterizing what *fixed point* (FP) configurations of such CA structurally look like, and (ii) determining *how many* FPs simple threshold CA may have. We note, that for the CA and Hopfield Nets whose nodes updates sequentially one at a time, and according to any linear threshold function, the resulting dynamics *always* eventually ends at a fixed point, for any initial configuration. For the same class of CA where the nodes update perfectly synchronously in parallel, the asymptotic dynamics ends either at a FP, or, in certain cases, it may end in a temporal cycle; for the parallel linear threshold CA *with memory*, only temporal 2-cycles are possible, there are (for sufficiently large values of the number of nodes n) only a few such cycles per CA, and those cycles can be computationally efficiently identified given the number of nodes, and the update rule type and radius [45, 48]. Therefore, enumerating exactly all the fixed points of a linear threshold CA gives us either the exact number of *all possible dynamical evolutions of the underlying system* (in scenarios for which it is known, there are no temporal cycles), or else a close approximation to it (in those scenarios where some 2-cycles may be present).

DEFINITION 1. *A* Cellular Space Γ *is an ordered pair* (G, Q)*, where* G *is an undirected regular graph (in general, finite or infinite), with each node labeled with a distinct integer, and* Q *is a finite set of states that has at least two elements, one of which being the special* quiescent state 0. *A* Cellular Automaton *(CA) is an ordered triple* (Γ, N, M) *where* Γ *is the CA's* cellular space*,* N *is its the* fundamental neighborhood*, and* M *is a finite state machine such that the input alphabet of* M *is* $Q^{|N|}$*, and the local transition function (update rule) for each node is of the form* $\delta : Q^{|N|+1} \rightarrow Q$ *for* CA with memory*, and* $\delta : Q^{|N|} \rightarrow Q$ *for* memoryless CA.

In this paper, we restrict our attention to CA on finite cellular spaces *with memory* (and likewise with the GA generalizations of such CA). The local transition rule δ specifies how each node updates, based on its current value and that of its neighbors in N. By composing local transition rules for all nodes together, we obtain *the global map* on the set of (global) configurations of a cellular automaton. Our main results in this paper are formulated in the context of the simple threshold CA defined over particularly simple underlying cellular spaces, namely, the *one-dimensional* (1-D) CA defined over finite rings (and, in infinite cases, one- or two-way infinite lines). We note, that these results can be generalized to other, higher-dimensional Cartesian cellular space (but the combinatorics details do get trickier with an increase in cellular space's dimensionality [42]).

DEFINITION 2. *A* graph automaton *or* network automaton *generalizes finite CA in one or both of the following two respects:(i) the underlying graph can be any simple undirected graph (i.e., it need not be regular); and (ii) different nodes in the graph can be different finite-state machines, i.e., may use different local update rules.*

Throughout this paper, all update rules will be (restricted types of) Boolean-valued functions, that is, in all of our CA and GA models, each node can be in one of two different states, 0 or 1; and each such node makes state transitions, as a function of the states of some of its neighboring nodes, according to a fixed, deterministic *local update rule*.

DEFINITION 3. *A* 1-D cellular automaton *of radius* $r \geq 1$ *is a CA defined over a one-dimensional string of nodes, such that each node's next state depends on the current states nodes away (and, in case of the* CA with memory*, on the current state of that node itself).*

DEFINITION 4. *A* Boolean-valued linear threshold function *of* n *inputs,* $x_1, ..., x_n$*, is any function of the form*

$$f(x_1, ..., x_n) = \begin{cases} 1, & \textit{if } \sum_i w_i \cdot x_i \geq \theta \\ 0, & \textit{otherwise} \end{cases} \quad (1)$$

where θ *is an appropriate* threshold constant*, and* w_i *are real-valued* weights.

DEFINITION 5. *Given two arbitrary Boolean vectors,* $X = \langle x_1, x_2, \ldots, x_n \rangle$ *and* $Y = \langle y_1, y_2, \ldots, y_n \rangle$*, define a binary relation "*$\preceq$*" as follows:* $X \preceq Y$ *if* $x_i \leq y_i$ *for all* i*,* $1 \leq i \leq n$*. An n-input Boolean function* f *is* monotone *if and only if* $X \preceq Y$ *implies that* $f(X) \leq f(Y)$*.*

DEFINITION 6. *An n-input Boolean function* g *is called* symmetric *if it only depends on* how many *of its inputs are equal to 1, but not which particular ones. In other words, in case of symmetric functions, the ordering in which the inputs are specified does not matter.*

Examples of Boolean-valued threshold functions that are both monotone and symmetric include the familiar *AND*, *OR* and *Majority* functions. Linear threshold functions that are also simultaneously both monotone and symmetric are called *simple threshold functions*.

DEFINITION 7. *A Threshold Cellular Automaton is a (parallel or sequential) cellular automaton such that its node update rule* δ *is a* Boolean-valued linear threshold function. *A simple threshold (S)CA is a CA whose local update rule* δ *is a* monotone symmetric *Boolean (threshold) function.*

We refer to the cellular automata whose nodes update according to Boolean threshold functions that are both symmetric and monotone as to *simple threshold* CA. Of a particular interest among simple threshold CA are those with the MAJORITY function as their local update rule, $\delta = MAJ$. In the 1-D case, a node of a Boolean-valued CA with memory that updates according to the MAJ rule will evaluate to 1 if and only if at least $r+1$ out of its $2r+1$ inputs are currently in the state 1.

We next define the basic configuration space properties of cellular and network automata. In those definitions, *finite* CA with n nodes are assumed; extending the definitions to infinite CA is straight-forward. $\mathcal{D}$ denotes the set of possible states (that is, values) that a node of a cellular or network automaton can be in. We assume $\mathcal{D} = \{0, 1\}$ (i.e.,, the Boolean domain) in the rest of the paper.

DEFINITION 8. *A* configuration *of a CA (or other kind of discrete dynamical system) is a vector* $(b_1, b_2, \ldots, b_n) \in \mathcal{D}^n$*. The function computed by a cellular or network automaton* $\mathcal{S}$*, denoted by* $F_{\mathcal{S}}$*, specifies for each configuration* $\mathcal{C}_1$*, the next configuration* $\mathcal{C}_2$ *reached by* $\mathcal{S}$ *after carrying out the updates of all the nodes' states:* $F_{\mathcal{S}}(\mathcal{C}_1) = \mathcal{C}_2$*. Thus, the function* $F_{\mathcal{S}} : \mathcal{D}^n \rightarrow \mathcal{D}^n$ *is a* global map *on the set of configurations; this global map defines the dynamics of a cellular or network automaton.*

DEFINITION 9. *The* configuration space *(also called* phase space*)* $\mathcal{P}_{\mathcal{S}}$ *of a cellular or network automaton* $\mathcal{S}$ *is a directed graph whose vertices are configurations and whose directed edges capture* transitions *from a configuration to its successor configuration. That is, there is a vertex in* $\mathcal{P}_{\mathcal{S}}$ *for each global configuration of* $\mathcal{S}$*; and there is a directed edge from a vertex representing configuration* $\mathcal{C}'$ *to that representing configuration* $\mathcal{C}$ *iff* $F_{\mathcal{S}}(\mathcal{C}') = \mathcal{C}$.

Since the classes of CA we study in this paper, as well as the related (Boolean) Network Automata, S(y)DS and (Discrete) Hopfield Network models (see Bibliography), are all deterministic, each configuration in the phase space of any of those cellular or network automata necessarily has an out-degree of 1 and, hence, a *unique* successor configuration.

DEFINITION 10. *Given two configurations* $\mathcal{C}'$ *and* $\mathcal{C}$ *of a cellular or network automaton* $\mathcal{S}$*, configuration* $\mathcal{C}'$ *is a* PREDECESSOR *of* $\mathcal{C}$ *iff* $F_{\mathcal{S}}(\mathcal{C}') = \mathcal{C}$*, that is, if* $\mathcal{S}$ *moves from* $\mathcal{C}'$ *to* $\mathcal{C}$ *in one global transition step. Similarly,* $\mathcal{C}'$ *is an* ANCESTOR *of* $\mathcal{C}$ *if there is an integer* $t \geq 1$ *such that* ${F_{\mathcal{S}}}^{t}(\mathcal{C}') = \mathcal{C}$*, i.e., if* $\mathcal{S}$ *evolves from* $\mathcal{C}'$ *to* $\mathcal{C}$ *in one or more transitions.*

In particular, a predecessor of a given configuration is a special case of an ancestor.

DEFINITION 11. *A configuration* $\mathcal{C}$ *of a cellular or network automaton* $\mathcal{S}$ *is a* FIXED POINT (FP) *configuration iff* $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}$*, i.e., if the transition out of* $\mathcal{C}$ *is back to* $\mathcal{C}$ *itself.*

Thus, a FP is a configuration that is its own predecessor. The fixed point configurations are also often referred to as *stable configurations* (esp. in the Hopfield networks literature); we use the two terms interchangeably.

DEFINITION 12. *A configuration* $\mathcal{C}$ *of a CA or a Network Automaton is a* CYCLE CONFIGURATION *(CC) if there is an integer* $t \geq 2$ *such that (i)* ${F_{\mathcal{S}}}^{t}(\mathcal{C}) = \mathcal{C}$*; and (ii)* ${F_{\mathcal{S}}}^{q}(\mathcal{C}) \neq \mathcal{C}$*, for any integer* q*,* $0 < q < t$*. Integer* t *above is called the* period *or* length *of the temporal cycle. A configuration* $\mathcal{C}$ *is a* TRANSIENT CONFIGURATION *(TC) if* $\mathcal{C}$ *is neither a fixed point nor a cycle configuration.*

So, $\mathcal{C}$ is a *cycle configuration* if it is reachable from itself in two or more transitions; that is, it is its own ancestor, but not a predecessor. In contrast, a transient configuration can never be its own ancestor.

4. RELATED WORK

Cellular automata (CA) were originally introduced as a mathematical model of the behavior of biological systems capable of self-reproduction [31]. Subsequently, variants of CA have been extensively studied in a great variety of application domains – in particular, for modeling complex physical, biological and social systems and their collective dynamics (e.g., [15, 17, 18, 42, 54, 55, 56]). Various computational aspects of CA, including computational complexity of determining various properties of interest, have been studied by a number of researchers; see, e.g., [20, 8, 17, 18, 35, 36, 45, 42]. Computational complexity problems about determining various properties of CA dynamics (for finite cellular spaces) have been addressed in e.g. [10, 20, 28, 34, 35], and more recently in e.g. [46, 41, 49, 50].

The first **NP**-complete problems for finite CA are shown in [20]; these problems are of a general REACHABILITY flavor, i.e., they address the properties of the FORWARD DYNAMICS of CA. [26] studies the reverse dynamics, more specifically, the reversibility and surjectivity problems about CA. CA BACKWARD DYNAMICS, such as the problem of an arbitrary configuration's PREDECESSOR EXISTENCE, and computational complexity of that and other related problems is addressed in [35]. In [10], Durand solves the injectivity problem for arbitrary 2-D CA but restricted to the *finite subconfigurations* only; that work contains one of the first results on **coNP**-completeness of a natural problem about CA. Further, Durand addresses the REVERSIBILITY PROBLEM in the same, two-dimensional CA setting in [11].

Insofar as the most relevant prior arts on the GA generalizations of finite CA, many such models have been proposed over the past 30-40 years, including discrete Hopfield Nets, Sequential and Synchronous Dynamical Systems, one-way automata, and others. The computational complexity of answering questions about the existence [7], the number [46, 49] or the reachability [6] of *fixed points* of an appropriate class of graph automata can be argued to provide important insights into the *collective dynamics* of multi-agent systems, as well as of other complex physical, biological, and socio-technical networks that are abstracted via those formal network automata [7, 30, 40].

Among various proposed models of graph or network automata of interest, we mention Sequential and Synchronous Dynamical Systems (SDS and SyDS, respectively). The SDS and SyDS models were introduced as a formal mathematical framework for the theory of (agent-based) simulations [27, 40]. These models have been previously applied to modeling, for instance, large-scale traffic systems, biological warfare and terror attacks, and agent-based models of epidemics propagation. S(y)DSs are closely related to the *graph automata* (GA) models studied in [28, 32] and the *one-way cellular automata* studied in [34]. Computational complexity of determining various configuration space properties of S(y)DSs is investigated in, e.g., [7, 6, 46, 41, 50].

Counting or *enumeration* problems about various types of configurations of cellular automata, Hopfield nets and other discrete dynamical systems have been extensively studied in the literature, as well. However, overall, counting problems have received considerably less attention than their, say, existence or classification counterparts. Moreover, most of the prior art on counting different types of stable configurations and/or other structures of interest in discrete dynamical systems are of experimental nature. Namely, most of those results are (often very loose) numerical estimates based on statistical sampling and extensive computer simulations, rather than analytically proven exact or approximate enumerations; some representative examples can be found in [2, 22, 24, 25]. Among relatively rare interesting theoretical studies on (in)tractability of counting problems about CA and more general complex network models, we single out the work by Floreen and Orponen on the computational complexity of counting various phase space structures, including but not limited to the FPs, in the context of DHNs [12, 13, 33].

5. COUNTING FIXED POINTS IN SIMPLE THRESHOLD CA AND GA

We now turn to the central problem: enumerating all fixed points in simple threshold CA. The results are virtually identical for parallel and sequential CA, FPs being *invariant* with respect to the node update ordering, so we won't bother making this distinction in the sequel. For simplicity and clarity, we shall assume the CA *with memory* throughout. We first characterize the FP configurations structurally, and then establish the basic results on how many FPs simple threshold CA have, when defined on 1-D cellular spaces. Similar characterizations can be readily provided for higher-dimensional Cartesian cellular spaces; the details of combinatorial argument in higher dimensions get more cumbersome, but from a fundamental computational complexity standpoint, this problem, #FP, remains tractable regardless of the dimensionality. (As for non-Cartesian cellular spaces, as far as we know, the problem of counting FPs is still open.) Among all simple threshold rules, that is, all k-threshold Boolean-valued functions on $2r+1$ inputs where $k \in \{0, 1, ..., 2r+2\}$, arguably the most interesting one is the MAJORITY (MAJ) rule, for which $k = r + 1$ [44]. The MAJ rule is the only among k-threshold rules that treats 0s and 1s equally. Hence, among other interesting properties, we show that MAJ CA have much more numerous fixed points than the CA with any other simple k-threshold node update rule, for $k \in \{0, 1, ...r, r + 2, ..., 2r + 2\}$.

LEMMA 1. *Any fixed point configuration C of a 1-D MAJ CA belongs to one of the following three types of configurations:*

(i) global configurations made of all 0s or all 1s; or

(ii) spatially periodic *configurations with the spatial period not exceeding $O(r)$ (where the rule radius $r \geq 1$ is a fixed positive integer); or*

(iii) configurations C is made of some positive number of stable blocks *of sufficiently many consecutive 0s and sufficiently many consecutive 1s; i.e., there exist positive integers $l_1, l_2, l_3, ...$ such that C is either of the type $C = 0^{r+l_1}1^{r+l_2}0^{r+l_3}...$ or of the type $C = 1^{r+l_1}0^{r+l_2}1^{r+l_3}....$*

Claim (i) of the Lemma is obvious. Claim (ii) immediately follows from the results originally established in [8], whereas proof of claim (iii) can be found in [17]. Notice that the configurations of Type (i) are just a special case of Type (iii) configurations: in Type (i) configurations, there is *exactly one stable block* of consecutive nodes in the same state (i.e, either all nodes are in state 0 or all are in state 1). We remark that the OR CA (where Boolean OR is viewed as "1-threshold" function) and the AND CA ("(2r+1)-threshold") are only capable of fixed points of Type (i), whereas other k-threshold rules ($k \neq r + 1$) are, in general, capable of both types (i) and (ii). For example, spatially periodic configuration on $3m$ nodes given by $(001)^m$ is stable for the 3-threshold function with $r = 3$ (that is, a node updates its state to 1 iff at least 3 out of 7 of its inputs are currently in state 1) and assuming *circular boundary conditions*. More on the spatially periodic FP configurations can be found in[45, 48]. The nontrivial non-spatially-periodic FP configurations of Type (iii), with an arbitrary number of stable blocks (the only restrictions being imposed by the rule radius r and the total number of nodes n in the underlying cellular space), however, are unique to the MAJ rule among the simple threshold 1-D CA [45, 48]. That is, $k = r + 1$ is the only one among simple k-threshold rules that allows for FPs of Type (iii) above.

How many FPs do different simple threshold CA have? For all the rules other than MAJ, this number of FPs is small and can be determined by examining a constant number of "neighborhood" (sub)configurations of at most $O(r)$ nodes, where we recall that $r \geq 1$ is an arbitrary, but *fixed* positive integer. For example, for AND and OR CA, the only two FPs are the configuration made of all 0s and the one made of all 1s. For $(k, r) = (3, 3)$ CA, in addition to these two, the only other FPs are the ones of type (ii); more specifically, the only such FP configurations are of the general form[1] $(001)^m$ where m is a positive integer, or $(001)^{\aleph_0}$ in the infinite at-least-3-out-of-7 CA case.

In case of the MAJ update rule on 1-D CA, we establish the following fundamental result:

THEOREM 1. **(The Main Theorem)** *Let a MAJ parallel or sequential CA be given on a* finite *1-D cellular space, and let n be the total number of nodes. Then this MAJ CA has a number of fixed points that is* exponential *in n.*

The estimation of the total number of Type (iii) FPs in such MAJ CA is based on a conceptually relatively straight-forward, combinatorially cumbersome counting argument (see *Appendix*). Only a few more FPs are then to be added to obtain the total FP count – those that are of Type (i) (exactly two of them) and of Type (ii) (still only a few, but the exact number of Type (ii) FPs in general depends on rule radius r and the total number of nodes n).

Proof idea: We outline the combinatorial methodology and the main ideas behind determining the *exact* number of FPs of MAJ 1-D CA. Details can be found in *Appendix*.

First, we note that establishing the exact number of FPs of types (i) and (ii) in Lemma 1, given the values of parameters n and r, is fairly straight-forward. Therefore, the challenge is to determine the number of FPs of Type (iii). This number can be obtained by the summation over the number of FPs made of exactly l stable blocks, for $l \geq 2$. To establish an exponential lower bound on #FP of Type (iii), we show that, for l sufficiently large, there are *exponentially many* fixed point configurations made of *exactly l* stable blocks alone. Mathematical tools used to establish this lower bound are *generating functions* and *Stirling's formula*; see *Appendix* for details. Consequently, it then follows that the total number of FPs, across all possible values of the number of stable blocks l, also must be exponential as a function of the number of nodes n.

Proof details: The claim of the Theorem is that $|\#FP|$ is *exponential* in n. To establish that result, it suffices to establish an exponential lower bound for $|\#FP|$. Assume a 1-D Boolean Majority CA has a fixed rule radius $r \geq 1$ and n nodes. Since r is fixed, without loss of generality, assume $n \gg r$. For notational convenience, let's assume circular boundary conditions and that the number of nodes n is even. (The analysis for other types of boundary conditions and n arbitrary is similar.)

Let's consider all FPs that are made of *exactly l* stable blocks, where $l \geq 1$ and each block is of length at least $r+1$. Recall that by a stable block we mean consecutive nodes that are either all in state 0 or all in state 1, and so that none of them can change their state ever in the future, given

[1]Note: the exact form of the type (ii) FP depends on the details of the boundary conditions; e.g., 1001001 is a FP for appropriate fixed boundary conditions but not for the periodic boundary conditions, and vice versa in case of configuration 001001.

the node update rule ($\delta = MAJ$ in this case) and a fixed rule radius $r \geq 1$. We want to determine the exact number of FPs made of exactly l blocks. For $l = 1$, the answer is *two*: these are the two configurations of Type (i) above, 0^n and 1^n. For $l = 2$, there are exactly $n \cdot (n - (2r + 1))$ different FPs, each made of exactly two stable blocks. Let's consider the number of FPs with a single block of exactly $r + 1$ consecutive zeros. We outline how is this number obtained. Due to the *wrap-around effect* of the assumed circular boundary conditions, there are exactly n such FPs (as the beginning of that block of consecutive zeros can be at any of the nodes $v_1, v_2, ..., v_n$). Similarly, under the same assumptions, there are exactly n FP configurations made of a stable block of exactly $r + 2$ zeros and $n - r - 2$ ones, etc.

Let's generalize the above argument to all possible sizes of the two stable blocks (one block of consecutive 0s and one block of consecutive 1s). Let $bs(0)$ denote the actual number of consecutive zeros in a FP. Clearly, $bs(0)$ ranges from $r+1$ (the minimum possible number of consecutive 0's so that this block of zeros is stable) to $n-r-1$ (the maximum number of consecutive zeros, corresponding to the minimum possible number of consecutive ones, since $bs(1) = n-bs(0)$). Therefore, there are exactly $n-2r-1$ possible distinct values of $bs(0)$, and for each, there are (under the circular boundary conditions assumption) exactly n FP configurations with a single block of $bs(0)$ zeros and a single block of $bs(1) = n - bs(0)$ ones. This gives the total of $n \cdot (n - (2r + 1))$ FP configurations made of *exactly* two stable blocks (one block of 0s and one block of 1s).

The above analysis can be generalized to an arbitrary number of stable blocks $l \geq 2$. In doing so, care needs to be taken not to count certain configurations multiple times. That can be accomplished by appropriately applying the *inclusion-exclusion principle* from classical combinatorics.

Example: should the configuration $0^{r+3}1^{n-2r-7}0^{r+4}$ be considered as made of two stable blocks, with $bs(0) = 2r+7$ and $bs(1) = n-2r-7$, or of three stable blocks, of lengths $r+3$, $n-2r-7$ and $r+4$, respectively? If we count this configuration among the two-block FPs (per analysis above), then we need to apply an appropriate *exclusion principle* to ensure that we don't also count it among the FPs with $l = 3$.

However, the details of these combinatorial considerations do not change the fact that, for each l, determining the number of FPs made of exactly l stable blocks can be done computationally efficiently. The total number of FP configurations of Type (iii), then, is obtained by summing over all "eligible" values of l. Most importantly from a computational standpoint, this process can be completed in time that is (low-degree) polynomial in n, and hence, by the previous discussion, the total number of FPs of the 1-D Majority CA (defined on finite lines or rings) can be determined efficiently.

Applying the combinatorial technique of *generating functions* to the expression for the number of FPs made of exactly l stable blocks, it follows that, given an $l \geq 2$, there are at least

$$|\#FP(l)| \geq 2\frac{(n - rl - 1)!}{(l - 1)! \cdot (n - rl - l)!} \tag{2}$$

fixed points made of exactly l stable blocks (where the multiplicative factor of 2 comes from "flipping zeros and ones" and, being constant, will be ignored in the derivations that follow to simplify the notation).

A lower bound on the total number of FPs of Type (iii) can then be obtained by summing over all l, for $2 \leq l \leq \frac{n}{r+1}$. Therefore, the total number of FPs of all three possible types satisfies the lower bound

$$|\#FP| \geq 2 \sum_{l=1}^{\frac{n}{r+1}} \frac{(n - rl - 1)!}{(l - 1)! \cdot (n - rl - l)!} \tag{3}$$

(Note: generating functions are the standard tool commonly employed in the context of enumerating various types of *integer partitions*. Due to space constraints, we intentionally omit discussing the relationship between an old mathematical problem of integer partitioning, and our problem of enumerating FPs on a 1-D Majority CA. We will discuss the relationship between the number of distinct partitions of a positive integer n and the number of FPs of types (i) and (iii) of a 1-D CA defined on a ring of n nodes and with $\delta = MAJ$ in the expanded, journal version of this paper.)

To establish the claim that $|\#FP|$ grows exponentially with n, it suffices to show that there are exponentially many FPs made of exactly l' stable blocks for a particular, carefully selected value of $l' \in \{2, ..., \frac{n}{r+1}\}$. We observe that the number of FPs of type (iii), in general, decreases as the rule radius r increases, simply because the size of the smallest stable block of consecutive 0s or consecutive 1s increases with r. Without loss of generality, assume $r \geq 2$ and let's consider $l' = \frac{n}{2r}$. We will show that there are exponentially many FPs made of exactly l' stable blocks; the claim of the Theorem will then immediately follow, as the total number of FPs across all permissible values of parameter l will surely be greater than $|\#FP(l')|$ for a single value of l'.

Substituting $l = \frac{n}{2r}$ in Eqn. (2), and ignoring the multiplicative constant 2, we obtain

$$\frac{(n - r \cdot \frac{n}{2r} - 1)!}{(n - r \cdot \frac{n}{2r} - \frac{n}{2r})! \cdot (\frac{n}{2r} - 1)!} = \frac{(\frac{n}{2} - 1)!}{(\frac{n}{2} - \frac{n}{2r})! \cdot (\frac{n}{2r} - 1)!} \tag{4}$$

To simplify the notation, let $m = \frac{n}{2}$. Applying *Stirling's formula* to the expression in Eqn. (4), and simplifying the numerator and the denominator, we obtain

$$\alpha(m) \cdot (1 + \frac{m - r}{m(r - 1)})^{m(1-1/r)} \cdot (\frac{m - 1}{m/r - 1})^{\frac{m}{r}-1} = \alpha(m) \cdot (1 + \beta(m, r))^{m(1-\frac{1}{r})} \cdot (1 + \gamma(m, r))^{\frac{m}{r}-1} \tag{5}$$

where $\alpha(m) \sim \frac{1}{\sqrt{m}}$, and $\beta(m, r)$ and $\gamma(m, r)$ are strictly greater than 1. Moreover, under the assumption that $m = n/2 \gg r$, β will be *bounded away* from 1; in particular, when $r < m/2$, we have $m-r > m/2$, and hence $\beta(m, r) > \frac{1}{2(r-1)}$ for all such $m = n/2$. Lastly, $\gamma(m, r)$ can be simplified to $\frac{m(r-1)}{m-r} > \frac{(m-r)(r-1)}{m-r} = r - 1$, implying that for all values of m and r, $1+\gamma(m, r) \geq r \geq 2$. Therefore, the crude estimate for $|\#FP(l')|$ in equation (5) satisfies

$$\alpha(m) \cdot (1 + \beta(m, r))^{m(1-\frac{1}{r})} \cdot (1 + \gamma(m, r))^{\frac{m}{r}-1} > \\ > \frac{C}{\sqrt{m}} \cdot (1 + \frac{1}{2(r - 1)})^{m(1-\frac{1}{r})} \cdot 2^{\frac{m}{r}-1} \tag{6}$$

for some positive constant C. Consequently,

$$|\#FP(l')| \sim \alpha(m) \cdot (1+\beta(m, r))^{m(1-\frac{1}{r})} \cdot (1+\gamma(m, r))^{\frac{m}{r}-1}$$

is exponential in m and therefore also in $n = 2m$, and therefore the claim of Theorem 1 immediately follows.

■

COROLLARY 1. *The exact number of FPs of a 1-D MAJ (parallel or sequential) CA on n nodes is an exponential function of the number of nodes that can be evaluated computationally efficiently.*

With the above results, we can now fully characterize the number of FPs of 1-D CA for all possible simple threshold update rules:

LEMMA 2. *Given a radius $r \geq 1$, for every integer k such that $0 \leq k \leq 2r+2$, the exact number of fixed point configurations $|\#FP|$ of a 1-D CA on n nodes with the k-threshold node update rules can be efficiently determined (i.e., this problem is in the complexity class* **P**.*)*

- *In general, $|\#FP|$ depends on r, k, n and the boundary conditions.*
- *For $k \neq r+1$ (that is, for all simple threshold rules other than MAJ), $|\#FP| = O(n)$ (where n is the total number of nodes).*
- *In contrast, for $k = r+1$ (that is, the MAJ update rule), we have $|\#FP| = \Omega((1+\beta)^n)$ for some positive real number β.*

Proof idea: Finite 1-D CA with any simple threshold update rule that isn't the Majority function can only have FPs of Type (i) and Type (ii) in Lemma 1. There are only $O(n)$ possible fixed points of Type (ii), hence the second claim of this Lemma; see e.g. [45] for details. That #FP is exponential for the Majority CA follows from Theorem 1; the proof of our main result also outlines the combinatorics via which one can establish the number of FPs of Type (iii); and we have argued elsewhere, that the number of FPs of Types (i) and (ii) can be effectively determined for all simple threshold functions, including the Majority rule [45].

We note that our main results above also hold for the higher-dimensional Cartesian grids; for example, in 2-D, we have established that #FP for the Majority rule is exponential in the number of nodes n on the square/rectangular grids, as well as on the tori (as 2-D Cartesian grids with "wrap-around", that is, circular boundary conditions). While combinatorics is somewhat more involved in Cartesian grids of dimensionality 2 or higher, the crux of the argument is the same as what we have shown for 1-dimensional lines and rings. Likewise, the number #FP for simple threshold rules *other than Majority*, on higher-dimensional cellular spaces such as rectangular grids and tori (for 2-D), remains relatively low (in particular, $O(n)$), and can be efficiently determined. We leave out details due to space constraints. Instead, we shift our focus to the Boolean Networks and Graph Automata with simple threshold update rules that *only slightly generalize* the classical CA with those same update rules we have been discussing so far.

We recall, that CA as an abstract model for large MAS is appropriate only in those situations where i) the underlying multi-agent system is highly homogeneous, that is, in a given environment, all agents behave the same way; and ii) the pattern of inter-agent interactions is also highly homogeneous or regular. So the natural question arises: what are the implications for the asymptotic dynamics in general, and the problem of enumerating #FP and other types of configurations in particular, when the general type of the local behaviors (that is, linear threshold update rules) are kept the same as before, but we allow for some amount of heterogeneity in local interactions and/or in the underlying "network topology"? In particular, can provably more complex asymptotic behavior be obtained, if one or both among the network structure (i.e., "cellular space") and the local interactions (that is, node update rules) are made only slightly more general than what has been the case with the finite Simple Threshold CA discussed so far? Justifying such 'slight generalizations" from a MAS standpoint is easy: whether we are modeling a cyber-physical, socio-technical or biological multi-agent system, completely uniform behaviors across all agents in the ensemble, as well as complete uniformity of the underlying network topology, are an exception, not the rule. Now, that sufficiently complex local behaviors and/or inter-agent interaction patterns can lead to a highly complex collective dynamics, would not come as a surprise. However, we have been particularly interested in the situations where the MAS network's *structural complexity* (in terms of the underlying cellular spaces or graphs) as well as heterogeneity in terms of local behaviors are only rather slightly more general, or complex, than those found in the Simple Threshold CA – yet whose dynamics are *much more complex* and in particular harder to predict, than what we have seen insofar as Simple Threshold CA dynamics.

It turns out, that it does not take much of "making Simple Threshold CA more general" with respect to either the network structure or the non-uniformity in local behaviors, to obtain provably very complex dynamics. In particular, our next result shows that, when a Simple Threshold CA is only slightly generalized, a major phase transition in the complexity of predicting the resulting model's dynamics in general, and to enumerate its FPs in particular, takes place almost immediately. We recall that, under the usual assumptions in computational complexity theory, if a counting or enumeration problem is **#P**-complete, that means that exactly enumerating the combinatorial objects of interest (in our case, Fixed Points of a Boolean Network or Graph Automaton) is intractable except for the very small values of the problem parameters (in our case, the number of nodes in a CA, GA or Boolean Network).

THEOREM 2. *Consider a graph automaton defined over a uniformly sparse graph, and with all local update rules being simple threshold Boolean-valued functions. Enumerating all FPs of such a GA is* **#P***-complete, even when the following restrictions are simultaneously imposed: i) each node in the graph has at most 3 neighbors (in particular, the result holds for 3-regular underlying graphs); ii) the heterogeneity in local behaviors is minimal, i.e., each node "gets to choose" from just two given simple threshold update rules.*

Proof sketch: The original variant of this fundamental result was originally established by us for the aforementioned Sequential and Syncrhonous Dynamical Systems in [50]; subsequently, in [43], we have also established another variation of this hardness-of-counting result for uniformly sparse discrete Hopfield Nets, as well. The approach to establishing these hardness of counting results is to construct a *weakly parsimonious reduction*, that is, a polynomially-computable reduction that (approximately) preserves the number of solutions, from a known **#P**-hard counting problem to our problem. To that end, we have taken advantage of the results about the hardness of counting all satisfying truth assignments to a *Monotone 2CNF Boolean formula* (2-Mon-CNF). Moreover, since this hardness of counting still holds when the underlying 2-Mon-CNF is a sparse formula, that is, one in which each literal (a Boolean variable or its negation) occurs in only $O(1)$ clauses [21, 51], the weakly parsimonious reductions from #2-Mon-CNF to #FP of various GA and DHN models establish the **#P**-hardness of the problem of enumerating #FPs for several types of those Boolean Network, DHN and Graph Automata models. (Note, that it is immediate that this counting problem, for all finite underlying GA, CA and Boolean Network models, belongs to the class **#P**; so, once the hardness part is established via the weakly parsimonious reductions as elaborated upon in our prior work [50, 43], the claimed **#P**-completeness of counting FPs of the Boolean Networks and other models of interest follows immediately.)

To summarize, the FP configurations of simple threshold CA on finite cellular spaces can be very few (such as, in most extreme cases, *only two* for AND and OR rules) or *exponentially many* (for the MAJ rule); yet, either way, their exact number can be computationally efficiently determined. This is in stark contrast with respect to other, slightly more general types of binary-valued network automata such as the aforementioned SDSs and SyDSs, as well as discrete Hopfield networks, whose nodes update according to (possibly different) simple threshold update rules. In fact, in case of the S(y)DS network automata, for many classes of restricted update rules and underlying graphs, even *approximate counting* of their FPs is, under the usual assumptions in computational complexity theory, provably computationally hard [49, 46, 50].

6. CONCLUSIONS

As a step toward understanding the collective behavior of various large-scale multi-agent and other decentralized systems, we adopt a formal dynamical system approach to abstracting and then mathematically analyzing those systems and their formal dynamics. The primary methodological approach to studying properties of a dynamical system is to study its *configuration space*. In this paper, we consider a restricted class of cellular and graph automata models of large multi-agent ensembles viewed as discrete-time, discrete-state dynamical systems.

We have specifically focused on the problem of counting how many *fixed point* configurations such dynamical systems have, when each of their nodes has only two different states, and each such node updates according to a simple threshold Boolean function. In the case of classical (finite) CA, where every node updates according to the same rule, the counting problem of our interest can be shown to be computationally tractable and therefore easy to solve explicitly, at least in principle. In contrast, when different nodes are allowed to compute according to different simple threshold update rules, and the underlying network topology is (slightly) more general than that of CA defined on Cartesian cellular spaces, a phase transition in the complexity of determining the number of fixed points occurs. This has been demonstrated for binary-valued SDS, SyDS and DHN models, where it's been established that the corresponding counting problems are, in general, computationally intractable [50]. Therefore, an agent ensemble that can be abstracted as a deterministic cellular or graph automaton may display a straight-forward to predict behavior, or a provably complex behavior, depending on the specifics of the inter-agent local interactions, that is, how exactly the network's nodes interconnect and interact with one another, and how homogeneous is this network of agents in terms of possible different types of local behaviors – even when the allowable local behaviors are restricted to Boolean simple threshold functions (inspired by the simplest models of neurons and other biological networks).

In summary, we have identified two interesting phase transitions when it comes to (un)predictability of the asymptotic dynamics of a restricted class of CA and GA abstractions of large-scale MAS. First, among homogeneous MAS (in which all reactive agents change their local states according to the same deterministic Boolean-valued function), if local behaviors are restricted to simple threshold update rules, then the total number of stable configurations (and consequently, the total number of possible dynamic trajectories) is fairly small (at most linear in the number of agents), for all rules but one. The "exception" rule is the MAJORITY rule, for which there are exponentially many Fixed Points and therefore possible dynamic trajectories. Second, once we allow for modest heterogeneity in local behaviors, the complexity threshold is immediately crossed: a MAS where each agent still updates according to a simple threshold rule, but not all agents have to use the exact same rule, in general have unpredictable dynamics. Moreover, presence of just two different local behaviors is sufficient is sufficient for this phase transition in dynamics complexity. In particular, enumerating all fixed points (and, by extension, all possible dynamic trajectories) of such a simple network of agents is provably computationally intractable, even when each agent has at most (or exactly) three neighbors, and a choice of only two simple threshold update rules.

REFERENCES

[1] S. Amoroso and Y. Patt. "Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures", *J. Computer and System Sciences* (JCSS), vol. 6, pp. 448–464, 1972

[2] R. J. Bagley and L. Glass. "Counting and Classifying Attractors in High Dimensional Dynamical Systems", *J. Theoretical Biology*, vol. 183, pp. 269–284, 1996

[3] C. Barrett, B. Bush, S. Kopp, H. Mortveit, C. Reidys. "Sequential Dynamical Systems and Applications to Simulations", Technical Report, Los Alamos National Laboratory (LANL), 1999

[4] C. Barrett, H. Hunt, M. Marathe, S. Ravi, D. Rosenkrantz, R. Stearns. "Dichotomy Results for Sequential Dynamical Systems", LANL Tech. Report, LA-UR-00-5984, 2000

[5] C. Barrett, H. Hunt, M. Marathe, S. Ravi, D. Rosenkrantz, R. Stearns. "On Some Special Classes of Sequential Dynamical Systems", *Annals of Combinatorics*, vol. 7, pp. 381–408, 2002

[6] C. Barrett, H. Hunt, M. Marathe, S. Ravi, D. Rosenkrantz, R. Stearns. "Reachability problems for sequential dynamical systems with threshold functions", *Theoretical Comp. Sci.*, vol. 295 (1-3), pp. 41–64, 2003

[7] C. Barrett, H. Hunt, M. Marathe, S. Ravi, D. Rosenkrantz, R. Stearns, P. Tosic. "Gardens of Eden and Fixed Points in Sequential Dynamical Systems",

Discrete Math. Theoretical Comp. Sci. (DMTCS), Proc. AA DM-CCG, pp. 95–110, 2001

[8] K. Culik, L. P. Hurd, S. Yu. "Computation theoretic aspects of cellular automata", *Physica D,* vol. 45 (1-3), pp. 357–378, 1990

[9] N. Davey, L. Calcraft, R. Adams. "High capacity, small world associative memory models", *Connection Science*, vol. 18 (3), pp. 247–264, 2006

[10] B. Durand. "Inversion of 2D cellular automata: some complexity results", *Theoretical Computer Science*, vol. 134 (2) , pp. 387 – 401, 1994

[11] B. Durand. "A random NP-complete problem for inversion of 2D cellular automata", *Theoretical Computer Science*, vol. 148 (1), pp. 19 – 32, 1995

[12] P. Floreen, P. Orponen. "On the computational complexity of analyzing Hopfield nets", *Complex Systems* vol. 3, pp. 577 – 587, 1989

[13] P. Floreen, P. Orponen. "Attraction radii in binary Hopfield nets are hard to compute", *Neural Computations* vol. 5, pp. 812 – 821, 1993

[14] M. R. Garey, D. S. Johnson. *"Computers and Intractability: A Guide to the Theory of NP-completeness",* W. H. Freeman & Co., San Francisco, CA, 1979

[15] M. Garzon. *"Models of Massive Parallelism: Analysis of Cell. Automata and Neural Networks"*, Springer, 1995

[16] M. Gouda, C. Chang. "Proving Liveness for Networks of Communicating Finite State Machines." *ACM Transactions Prog. Languages and Systems (TOPLAS)*, vol. 8 (1), pp. 154 – 182, 1986

[17] E. Goles, S. Martinez. *"Neural and Automata Networks: Dynamical Behavior and Applications"*, Math. and Its Applications series (vol. 58), Kluwer, 1990

[18] E. Goles, S. Martinez (eds.) *"Cellular Automata, Dynamical Systems and Neural Networks"*, Math. and Its Applications series (vol. 282), Kluwer, 1994

[19] E. Goles, S. Martinez (eds.) *"Cellular Automata and Complex Systems"*, Nonlinear Phenomena and Complex Systems series, Kluwer, 1999

[20] F. Green. "**NP**-Complete Problems in Cellular Automata", *Complex Sys.*, vol. 1 (3), pp. 453 – 474, 1987

[21] C. Greenhill. '"The Complexity of Counting Colourings and Independent Sets in Sparse Graphs and Hypergraphs", *Computational Complexity,* vol. 9, pp. 52 – 72, 2000

[22] J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities", *Proc. Nat. Academy Science*, vol. 79, pp. 2554 – 2558, 1982

[23] M. Jerrum, A. Sinclair. "Polynomial-time approximation algorithms for the Ising model", *SIAM J. Comp.,* vol. 22, pp. 1087 – 1116, 1993

[24] S. A. Kauffman. "Metabolic stability and epigenesis in randomly connected nets", *J. Theor. Biology,* vol. 22, pp. 437 – 467, 1969

[25] S. A. Kauffman. *"The Origins of Order: Self-Organization and Selection in Evolution",* Oxford Univ. Press, Oxford, UK, 1993

[26] J. Kari. "Reversibility and surjectivity problems of cellular automata", *Journal of Computer and System Sciences* (JCSS), vol. 48, pp. 149 – 182, 1994

[27] R. Laubenbacher, B. Pareigis. "Finite Dynamical Systems", Tech. Report, Dept. of Math. Sciences, New Mexico State University, Las Cruces, NM, USA, 2000

[28] B. Martin. "A Geometrical Hierarchy of Graphs via Cellular Automata", Proc. MFCS'98 Satellite Workshop on Cellular Automata, Brno, Czech Republic, 1998

[29] R. J. McEliece et al. "The Capacity of the Hopfield Associative Memory", *IEEE Trans. Information Theory*, vol. 33 (4), pp. 461 – 482, 1987

[30] H. Mortveit, C. Reidys. "Discrete, sequential dynamical systems", *Discrete Mathematics,* vol. 226, pp. 281 – 295, 2001

[31] J. von Neumann, *"Theory of Self-Reproducing Automata",* edited and completed by A. W. Burks, Univ. of Illinois Press, Urbana, 1966

[32] C. Nichitiu and E. Remila. "Simulations of Graph Automata", Proc. MFCS'98 Satellite Workshop on Cellular Automata, Brno, Czech Republic, 1998

[33] P. Orponen. "Computational complexity of neural networks: a survey", *Nordic J. Comp.*, vol. 1 (1), pp. 94 – 110, 1994

[34] Zs. Roka. "One-way cellular automata on Cayley graphs", *Theoretical Computer Science*, vol. 132 (1-2), pp. 259 – 290, Sept. 1994

[35] K. Sutner. "On the computational complexity of finite cellular automata", *Journal of Computer and System Sciences,* vol. 50 (1), pp. 87 – 97, Feb. 1995

[36] K. Sutner. "Computation theory of cellular automata", *MFCS'98 Satellite Workshop on Cellular Automata,* Brno, Czech Republic, 1998

[37] T. Toffoli, H. Margolus. "Invertible cellular automata: A review", *Physica* **D** vol. 45, pp. 229 – 253, 1990

[38] P. Tosic. "A Perspective on the Future of Massively Parallel Computing: Fine-Grain vs. Coarse-Grain Parallel Models", Proc. 1st ACM Conference on Computing Frontiers (CF'04), pp. 488 – 502, Ischia, Italy, Apr. 2004

[39] P. Tosic. "On Counting Fixed Point Configurations in Star Networks", APDCM Workshop, *Proc. IEEE Int'l Parallel & Distributed Processing Symp.* (IPDPS'05), Denver, Colorado, USA, 2005

[40] P. Tosic. "Cellular automata for distributed computing: models of agent interaction and their implications", *Proc. IEEE Conf. on Systems, Man and Cybernetics* (SMC'05), vol. 4, pp. 3204 – 3209, The Big Island, Hawaii, USA, 2005

[41] P. Tosic. "On Modeling and Analyzing Sparsely Networked Large-Scale Multi-agent Systems with Cellular and Graph Automata", Proc. Intl. Conf. Computational Science (ICCS-06), Reading, UK; LNCS vol. 3993, pp. 272 – 280, Springer, 2006

[42] P. Tosic. "Cellular Automata Communication Models: Comparative Analysis of Parallel, Sequential and Asynchronous CA with Simple Threshold Update Rules", *Intlernational Journal of Natural Computing Research* (IJNCR), vol. 1 (3), pp. 66 – 84, 2010

[43] P. Tosic. "On Simple Models of Associative Memory:

Network Density Is Not Required for Provably Complex Behavior ", Proc. Brain Informatics & Health (BIH-2016), Omaha, Nebraska; Lecture Notes in Computer Science (LNCS) series, vol. 9919, pp. 61 – 71, Springer, 2016

[44] P. Tosic, G. Agha. "Concurrency vs. Sequential Interleavings in 1-D Threshold Cellular Automata", APDCM Workshop, Proc. IEEE Int'l Parallel & Distributed Processing Symp. (IPDPS'04), Santa Fe, New Mexico, USA, 2004

[45] P. Tosic, G. Agha. "Characterizing Configuration Spaces of Simple Threshold Cellular Automata", Proc. ACRI'04, Amsterdam, Netherlands, Springer LNCS, vol. 3305, pp. 861 – 870, 2004

[46] P. Tosic, G. Agha. "On Computational Complexity of Counting Fixed Points in Symmetric Boolean Graph Automata", Proc. 4th Int'l Conf. Unconventional Comp. (UC'05), Sevilla, Spain; Springer LNCS, vol. 3699, pp. 191 – 205, 2005

[47] P. Tosic, G. Agha. "On Computational Complexity of Predicting Dynamical Evolution of Large Agent Ensembles", Proc. 3rd European Workshop Multiagent Systems (EUMAS'05), pp. 415 – 426, Flemish Academy of Sciences, Brussels, Belgium, Dec. 2005

[48] P. Tosic, G. Agha. "Parallel vs. Sequential Threshold Cellular Automata: Comparison and Contrast", Proc. 1st Euro. Conf. Complex Systems (ECCS-2005), Euro. Complex Systems Society, Paris, France, 2005

[49] P. Tosic. "On the Complexity of Counting Fixed Points and Gardens of Eden in Sequential Dynamical Systems on Planar Bipartite Graphs", *International Journal on Foundations of Computer Science* (IJFCS), vol. 17 (5), pp. 1179 – 1203, World Scientific, 2006

[50] P. Tosic. "On the complexity of enumerating possible dynamics of sparsely connected Boolean network automata with simple update rules", *Discrete Math. & Theoretical Comp. Sci.* (DMTCS); pp. 125–144, 2010

[51] S. Vadhan. "The Complexity of Counting in Sparse, Regular and Planar Graphs", *SIAM Journal of Computing*, vol. 31 (2), pp. 398 – 427, 2001

[52] L. Valiant. "The Complexity of Computing the Permanent", *Theoretical Computer Science*, vol. 8, pp. 189 – 201, 1979

[53] L. Valiant. "The Complexity of Enumeration and Reliability Problems", *SIAM J. Comp.,* vol. 8 (3), 1979

[54] S. Wolfram. "Computation theory of cellular automata", *Commun. Math. Physics,* vol. 96, 1984

[55] S. Wolfram. *"Theory and applications of cellular automata"*, World Scientific, 1986

[56] S. Wolfram (ed.). *"Cellular Automata and Complexity (collected papers)"*, Addison-Wesley, 1994