

On Modeling and Analyzing Sparsely Networked Large-Scale Multi-agent Systems with Cellular and Graph Automata

Predrag T. Tošić

Open Systems Laboratory, Department of Computer Science,
University of Illinois at Urbana-Champaign, U.S.A.
p-tosic@cs.uiuc.edu

Abstract. Modeling, designing and analyzing large scale *multi-agent systems* (MAS) with anywhere from tens of thousands to millions of autonomous agents will require mathematical and computational theories and models substantially different from those underlying the study of small- to medium-scale MAS made of only dozens, or perhaps hundreds, of agents. In this paper, we study certain aspects of the global behavior of large ensembles of simple reactive agents. We do so by analyzing the collective dynamics of several related models of discrete complex systems based on cellular automata. We survey our recent results on dynamical properties of the complex systems of interest, and discuss some useful ways forward in modeling and analysis of large-scale MAS via appropriately modified versions of the classical cellular automata.

1 Introduction and Motivation

Multi-Agent Systems (MAS) are a research area where artificial intelligence and distributed computing overlap [28]. Hence, research in MAS heavily draws on the existing theories, tools and methodologies from both AI and distributed computing. What we would like to contribute to the more thorough understanding and better design of large-scale MAS are some ideas, paradigms and tools from another scientific discipline, namely, *complex dynamical systems*. Among many mathematical models of discrete dynamical systems, we find one class of models particularly simple yet useful for addressing many fundamental issues in distributed computing in general, and in large-scale multi-agent systems in particular. This class are the classical *cellular automata* and some of their *graph automata* extensions.

Cellular automata (CA) [10, 11, 13, 29, 30] are discrete dynamical systems whose individual components are rather simple, yet that can exhibit highly complex and unpredictable behavior due to these simple components' mutual interaction and synergy. A CA is made of a finite or infinite regular one-, two- or higher-dimensional grid of nodes, where each node behaves like a finite state machine with a fixed, usually small, number of distinct states. The nodes are interconnected together and can affect each other: the future state of a given node

depends on the current states of some of its near-by nodes. The “program” that tells a node how to update its state, based on the states of these neighboring nodes, is deterministic and fixed; it is called the *local update rule*. All nodes of a classical CA update (i) synchronously in parallel with each other, and (ii) according to the same update rule.

What are the important properties of the large-scale distributed computational and communication systems that can be adequately captured by the CA-like models? From distributed computing and multi-agent system perspectives, studying the global dynamics of a CA translates into an exploration of the global behavior of a multi-agent system when (i) the individual agent behaviors are fixed, (ii) the pattern of multi-agent interaction (“network topology”) is fixed, and (iii) both the individual agent behaviors and the interaction patterns among the agents are *homogeneous* across the entire system. However, CA-like models do not allow for an individual agent’s deliberation or adaptation of any sort: the individual agents are strictly *reactive*, and are characterized by a fixed, nonadaptive behavior. Instead, the focus of studying such models is on various *emergent dynamical properties* at the level of *agent ensembles*.

Several modifications of the basic CA model along different dimensions can be argued to provide useful abstractions for loosely coupled distributed computing systems, and, in particular, for the *large-scale* MAS made of simple reactive, autonomously executing agents. We identify the following four as the most important:

- *heterogeneity* of the cellular/graph automata in terms of (i) the individual agent behaviors and (ii) the inter-agent interaction pattern, in contrast to the strict *homogeneity* of the classical CA in both these respects;
- *model of inter-agent communication* insofar as whether the agents locally compute synchronously or asynchronously, and whether they interact with one another synchronously or asynchronously;
- *adaptability* of the individual agents, i.e., are these agents capable of *dynamically changing their behavior* via, e.g., reinforcement learning, or are their individual behaviors *fixed* once the state of their environment is specified;
- *dynamics* of the MAS network topology, i.e., whether the underlying cellular space of a cellular or graph automaton is allowed to change with time.

Among these four dimensions along which classical CA can be generalized, our recent and ongoing research has been focusing on the first two. We analyze some computational complexity implications of extending the classical CA model by allowing a limited degree of heterogeneity in individual agent behaviors, as well as in the underlying communication network, in [19, 20, 21, 23, 25]. We study some implications of the nature of inter-agent communication in [22, 24]. The purpose of the present report is to summarize some of our main findings, and to propose interpretations of those results in the context of modeling and analyzing large-scale MAS.

2 Preliminaries

Classical cellular automata are made of a finite or infinite collection of *identical* finite state machines (FSMs) that are interconnected in some highly regular fashion (e.g., [13, 29, 30]). Each copy of this FSM, called a node, updates its state based on its own current state, and the current states of the nodes in some pre-specified neighborhood, according to its *local update rule* [10, 24, 30]. All nodes update perfectly synchronously in parallel. The underlying communication network topology is called *cellular space* of a CA [10, 24]. For instance, the most studied cellular spaces in one dimension (1D) are finite rings and infinite lines of nodes.

When it comes to the very large CA, perfect synchrony of the classical CA parallel node updates is difficult to justify on either physical or distributed computing grounds. Hence, researchers have also studied various models where this perfect synchrony assumption is dropped [10, 11, 15]. For the complete formal definitions of cellular automata, their cellular spaces, and their various variants with respect to the model of inter-node interaction, we refer the reader to [24].

We next define two classes of discrete dynamical system models that result once some *heterogeneity* is allowed in the parallel and sequential CA models. More formal and elaborate definitions, as well as a detailed discussion of the motivation behind these models, can be found in our earlier work (e.g., [20, 21, 25]). *Sequential Dynamical Systems* (SDSs) are proposed in [3, 4, 5] as an abstract model for computer simulations. This model has been applied in the context of modeling and simulation of large-scale socio-technical systems; an example is the *TRANSIMS* project at the Los Alamos National Laboratory [6]. An SDS $\mathcal{S} = (G, F, \Pi)$ consists of three components. $G(V, E)$ is an undirected graph with $|V| = n$ nodes, where each node has a 1-bit state. $F = (f_1, f_2, \dots, f_n)$ is the *global map* of \mathcal{S} , with f_i denoting a Boolean function associated with node v_i . Π is a permutation of (or a total order on) the nodes in V . If the permutation Π is dropped out, and all the nodes update synchronously in parallel (the way the nodes of classical, parallel CA update their states), we arrive at the definition of *Synchronous Dynamical Systems* (SyDSs).

A *configuration* of a Boolean SDS $\mathcal{S} = (G, F, \Pi)$ or an SyDS $\mathcal{S}' = (G, F)$ is a vector $(b_1, b_2, \dots, b_n) \in \{0, 1\}^n$, where b_i is the value of the state of node v_i , for $1 \leq i \leq n$. A configuration \mathcal{C} can also be thought of as a function $\mathcal{C}: V \rightarrow \{0, 1\}^n$. A single S(y)DS transition from one configuration to another is obtained by updating the state of each node v_i using the corresponding Boolean function f_i . In case of SDSs, these node updates are carried out in the order specified by the permutation Π .

The global map computed by an S(y)DS \mathcal{S} , denoted $F = F_{\mathcal{S}}$, specifies for each configuration \mathcal{C} the next configuration $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}'$ reached by \mathcal{S} after carrying out the updates of all the node states, whether in parallel or in the order given by Π . Thus, the map $F_{\mathcal{S}}: \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a total function on the set of global configurations. This function therefore defines the dynamics of \mathcal{S} . We say that \mathcal{S} moves from a configuration \mathcal{C} to a configuration $F_{\mathcal{S}}(\mathcal{C})$ in a single transition step. Assuming that each node update function f_i is computable in

time polynomial in the size of the description of \mathcal{S} , each transition step will also take polynomial time in the size of the S(y)DS's description.

The *configuration space* (also called *phase space*) $\mathcal{P}_{\mathcal{S}}$ of an SDS or SyDS \mathcal{S} is a directed graph defined as follows. There is a vertex in $\mathcal{P}_{\mathcal{S}}$ for each global configuration of \mathcal{S} . There is a directed edge from a vertex representing configuration \mathcal{C} to that representing configuration \mathcal{C}' if $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}'$. Since an SDS or SyDS is deterministic, each vertex in its phase space has the out-degree of 1.

Definition 1. A configuration \mathcal{C} of an S(y)DS \mathcal{S} is a *fixed point (FP) configuration* if $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}$, that is, if the transition from \mathcal{C} is back to \mathcal{C} itself.

Definition 2. A configuration \mathcal{C} of an S(y)DS is a *cycle configuration (CC)* if there exists 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 is called the *period or length of the temporal cycle*.

Definition 3. A configuration \mathcal{C} of an S(y)DS is a *transient configuration (TC)* if \mathcal{C} is neither a fixed point nor a cycle configuration.

Thus, a fixed point is its own predecessor. A cycle configuration is reachable from itself in two or more transitions, but not in a single transition. Both FPs and CCs are *recurrent configurations*, whereas TCs are never revisited. Fixed point, cycle and transient configurations for the classical, parallel CA are defined identically to the definitions for SDSs and SyDSs. In case of arbitrary *sequential* CA, however, the above definitions need to be modified since one first has to specify what is meant by a single computational step. We refer the reader to our prior work [22, 24] for more details.

3 Parallel vs. Sequential CA: Comparison and Contrast

In this section, we summarize some of the major implications of the interaction or communication model in a multi-agent system that can be abstracted as a collection of *communicating finite state machines* (CFSMs). The two main models of interaction in CFSMs studied in the literature on cellular automata, as well as Hopfield networks [14], are (i) the perfectly synchronous, parallel node updates, and (ii) the sequential, “one-node-at-a-time” node updates. The latter model of inter-node interaction in a CA or a Hopfield network is also often referred to as *asynchronous* (e.g., [15]), although we argue in [24] that this use of terminology is somewhat inaccurate.

It has been discovered in the contexts of *discrete Hopfield networks* (DHNs), parallel and sequential CA that, for the node update rules that are required to be *linear threshold functions* [8, 27], the node updates carried out synchronously in parallel allow for a possibility of temporal cycles. In contrast, sequential node updates do not allow temporal cycles, but only transient and fixed point configurations. That is, in a *finite* CA or a DHN whose nodes update in parallel,

and according to linear threshold rules, dynamical evolution starting from an arbitrary initial configuration either converges to a temporal two-cycle or to a fixed point. In contrast, configuration spaces of the corresponding *sequential* CA and DHNs *cannot have* any CCs.

The standard proof of these two properties that contrast the parallel/synchronous and the sequential/asynchronous models of interaction among the nodes of a CA or a Hopfield network, when the number of nodes is finite, can be found, e.g., in [10]. When the linear threshold update rules are also required to be *monotone* [22, 27], we have slightly extended the fundamental result from [10] along two lines (as well as considerably simplified the original, energy function based proof) in [24]. First, we have introduced an appropriate notion of *fairness* applicable to any CFM model with sequential/asynchronous interaction among the nodes. Under the fairness assumption, we have provided a guarantee that, starting from an arbitrary initial configuration, a sequential CA (or any other graph automaton or DHN) is *guaranteed* to finitely converge to one of its *proper* fixed points [24]. Second, insofar as the finite convergence of parallel or sequential CA computations on *infinite cellular spaces* is concerned, we have characterized the circumstances under which such convergence is guaranteed on any *finite subconfiguration* of such an infinite CA. We summarize this characterization of the fundamental configuration space and convergence properties of sequential and parallel CA in the Proposition below:

Proposition 1. *Let a parallel CA or a sequential SCA be defined over a finite or infinite 1D cellular space, with an arbitrary finite rule radius $r \geq 1$. Let this cellular automaton's local update rule be a monotone linear threshold function. Let's also assume, in the sequential cases, that an appropriate fairness condition holds.¹ Then for any starting configuration $C_0 \in PS(A)$ whatsoever, and any finite subconfiguration $C \subseteq C_0$, there exists a time step $t \geq 0$ such that*

$$F^{t+2}(C) = F^t(C) \tag{1}$$

where, in the case of fair SCA, the equation (1) can be replaced with

$$F^{t+1}(C) = F^t(C) \tag{2}$$

Insofar as the practical implications of these results are concerned, assume a distributed computational infrastructure can be approximated as a ring network of communicating finite state machines. If (i) each node in this infrastructure executes autonomously and (ii) changes its local state according to a monotone linear threshold rule, and (iii) there is a finite bound on the ratios of computational speeds of different nodes, and (iv) there is a finite bound on the inter-node communication delays, then this infrastructure is guaranteed to converge to a stable or steady state after a finite amount of time. In contrast, if the nodes (i.e., agents) execute mostly autonomously but so that *global synchronization* is externally imposed on the nodes insofar as when are the individual nodes allowed

¹ See [24] for a concrete such fairness condition, that is proven to be sufficient for the assertions about sequential CA in the Proposition to hold.

to change their local states, then the global state of this system may fail to ever stabilize and may, instead, wind up endlessly oscillating between two periodic configurations.

4 On the Hardness of Counting Fixed Points of Boolean S(y)DSs

Various configuration space properties of Boolean SDSs and SyDSs, as well as the computational complexity of determining those properties, have been extensively studied since the two models were introduced in the late 1990s. Barrett, Mortveit and Reidys [3, 4, 17], as well as Laubenbacher and Pareigis [16], have investigated mathematical properties of sequential dynamical systems. Barrett et al. have studied the computational complexity of several problems about configuration spaces of S(y)DSs, such as the PREDECESSOR EXISTENCE and PERMUTATION EXISTENCE problems (see, e.g., [1]). Problems related to the existence of FIXED POINT configurations are studied in [2]. Our own subsequent work further builds on the results in [2] by addressing the problems of exact and approximate *enumeration* of various structures such as the fixed points, the predecessor configurations, and the *garden of Eden* configurations [19, 20, 21, 23, 25]. Due to space constraints, we will only summarize the main findings related to the computational complexity of enumerating FP configurations for the two restricted classes of S(y)DSs – those whose nodes update according to (i) monotone and (ii) linear threshold Boolean-valued functions.

Monotone Boolean functions, formulae and circuits [27] have been extensively studied in many areas of computer science, from machine learning to connectionist AI to VLSI design. Cellular and other graph automata with the local update rules restricted to monotone Boolean functions have also been of a considerable interest (e.g., [2, 19, 22]). The problem of counting the FPs in *monotone* Boolean SDSs and SyDSs is originally addressed in [19]. In general, counting FPs of monotone Boolean S(y)DSs either exactly or approximately is computationally intractable. This intractability holds even for the graphs that are simultaneously bipartite, planar, and very sparse *on average* [19]. An example of such graphs are the *star graphs*, in which a single central node is connected to everyone else, and each non-central node is linked only to the central node. We recall that *2CNF* stands for Boolean formulae in *Conjunctive Normal Form* such that each clause contains exactly two literals [9].

Lemma 1. *Exactly enumerating the fixed points of a monotone Boolean SDS or SyDS defined over a star graph, and such that the update rule of the central node is given as a MONOTONE 2CNF formula of size $O(|V|)$, is #P-complete.*

We now turn attention to S(y)DSs whose nodes update according to *linear threshold functions* [2, 22, 24]. In particular, we assume that each node of such an S(y)DS has its update rule given not as a Boolean formula, but as a tuple of integers that encode the appropriate weights and the threshold; for more on

some important complexity-theoretic implications of the exact encoding of the nodes' update rules, we refer the reader to our detailed discussion in [21].

Lemma 2. *Exactly enumerating the FPs of an arbitrary $S(y)DS$ all of whose nodes use Boolean linear threshold update rules with small integer weights is $\#\mathbf{P}$ -complete.*

Moreover, the computational hardness of enumerating FPs still holds even when the following additional restrictions simultaneously hold: (i) the update rules are also *monotone* (i.e., in particular, when no negative weights are allowed), (ii) the underlying graph of an $S(y)DS$ is *uniformly sparse* so that every node has only $O(1)$ neighbors, and (iii) there are only two different update rules, i.e., there are only two possible different local behaviors among the nodes.

Proposition 2. *Exactly enumerating the fixed point configurations of monotone Boolean SDSs and $SyDS$ s is $\#\mathbf{P}$ -complete, even when all of the following restrictions simultaneously hold:*

- *the monotone update rules are linear threshold functions;*
- *at most two different positive integer weights are used by each local update rule;*
- *each node has only $O(1)$ neighbors in the underlying graph of this $S(y)DS$;*
- *only two different monotone linear threshold rules are used by the $S(y)DS$'s nodes.*

5 Concluding Remarks

We propose using cellular automata (CA) and their graph automata extensions for modeling large-scale multi-agent systems made of reactive, autonomously executing agents. We focus on two particular aspects of CA. One is the issue of (a)synchrony of the node updates, as an abstraction of communication (or other forms of inter-agent interaction) among robotic, software or other kinds of autonomous agents. We analyze some important implications of (a)synchrony of the inter-agent interactions in terms of the collective dynamics of an ensemble made of simple reactive agents. In particular, we show that, if the agents update their states according to monotone linear threshold rules, then the asynchronous local updates have an advantage of guaranteeing that the global state of the entire multi-agent system would eventually stabilize.

The second issue addressed are some implications for the global system behavior when some *heterogeneity* is introduced into (i) the communication network topology and (ii) the individual agent behaviors. In that context, we show that enumerating all stable global configurations of a MAS abstracted as an appropriate graph automaton, and therefore all its possible dynamical evolutions, is computationally intractable – even if the underlying network topology is *uniformly sparse* and each agent's behavior rather simple. Thus, even a very modest amount of heterogeneity in the individual agents' behaviors, coupled with some

non-uniformity in the inter-agent interaction pattern, can make the fundamental properties of such multi-agent system's global dynamics infeasible to predict.

In summary, classical cellular automata and their appropriately tuned extensions provide a simple, yet elegant and useful setting for formally reasoning about collective dynamics of large-scale multi-agent systems made of reactive autonomous agents.

Acknowledgements. The author would like to thank Gul Agha, Harry B. Hunt, Michael Loui, Madhav Marathe and the anonymous referees. This work was supported in part by the ONR MURI Grant N00014-02-1-0715 and the NSF Grant CNS 05-09321.

References

1. C. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns. "Predecessor and Permutation Existence Problems for Sequential Dynamical Systems", Los Alamos National Laboratory Report, LA-UR-01-668, 2001
2. C. L. Barrett, H. B. Hunt, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, P. T. Tosic. "Gardens of Eden and Fixed Points in Sequential Dynamical Systems", Proc. AA DM-CCG, *Discrete Math. & Theoretical Comp. Sci.*, pp. 95-110, 2001
3. C. Barrett, H. Mortveit, and C. Reidys. "Elements of a theory of simulation II: sequential dynamical systems" *Appl. Math. & Comput.*, vol 107/2-3, pp. 121-136, 2000
4. C. Barrett, H. Mortveit and C. Reidys. "Elements of a theory of computer simulation III: equivalence of SDS", *Appl. Math. & Comput.*, vol. 122, pp. 325-340, 2001
5. C. Barrett and C. Reidys. "Elements of a theory of computer simulation I: sequential CA over random graphs" *Applied Math. & Comput.*, vol. 98, pp. 241-259, 1999
6. R.J. Beckman, et. al. "TRANSIMS: Case Study", Dallas Ft-Worth. Los Alamos National Laboratory, LA UR 97-4502, 1999
7. P. Floreen, P. Orponen. "On the Computational Complexity of Analyzing Hopfield Nets", *Complex Systems* vol. 3, pp. 577-587, 1989
8. P. Floreen, P. Orponen. "Complexity Issues in Discrete Hopfield Networks", *NeuroCOLT Technical Report Series*, NC-TR-94-009, October 1994
9. M. R. Garey and D. S. Johnson. "*Computers and Intractability: A Guide to the Theory of NP-completeness*", W. H. Freeman and Co., San Francisco, California, 1979
10. E. Goles, S. Martinez. "*Neural and Automata Networks: Dynamical Behavior and Applications*", Math. and Its Applications series (vol. 58), Kluwer, 1990
11. E. Goles, S. Martinez (eds.) "*Cellular Automata and Complex Systems*", Nonlinear Phenomena and Complex Systems series, Kluwer, 1999
12. C. Greenhill. "The Complexity of Counting Colourings and Independent Sets in Sparse Graphs and Hypergraphs", *Comput. Complexity*, vol. 9, pp. 52-72, 2000
13. H. Gutowitz (ed.). "*Cellular Automata: Theory and Experiment*", N. Holland, 1989
14. J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities", *Proc. Nat'l Academy Sci. (USA)*, vol. 79, pp. 2554-2558, 1982

15. T. E. Ingerson and R. L. Buvel. "Structure in asynchronous cellular automata", *Physica D: Nonlinear Phenomena*, vol. 10 (1-2), pp. 59-68, January 1984
16. R. Laubenbacher and B. Pareigis. "Finite Dynamical Systems", Technical report, Dept. of Mathematical Sciences, N. Mexico State Univ., Las Cruces, 2000
17. H. Mortveit, C. Reidys. "Discrete sequential dynamical systems", *Discrete Mathematics*, pp. 281-295, vol. 226, Issue 1-3, 2001
18. D. Roth. "On the Hardness of Approximate Reasoning", *Artificial Intelligence*, vol. 82, pp. 273-302, 1996
19. P. Tošić. "On Counting Fixed Point Configurations in Star Networks", APDCM Workshop within *The 19th IEEE Int'l Parallel & Distributed Processing Symp.*, 2005; in *Proc. IEEE-IPDPS '05* (CD-Rom)
20. P. Tošić. "On Complexity of Counting Fixed Point Configurations in Certain Classes of Graph Automata", *Electronic Colloquium on Computational Complexity*, ECCC-TR05-051 (revision 2), April 2005
21. P. Tošić. "Counting Fixed Point and Gardens of Eden of Sequential Dynamical Systems on Planar Bipartite Graphs", *Electronic Colloquium on Computational Complexity*, ECCC-TR05-091, August 2005
22. P. Tošić, G. Agha. "Characterizing Configuration Spaces of Simple Threshold Cellular Automata", *Proc. of the 6th Int'l Conf. on Cellular Automata for Research and Industry* (ACRI'04), Springer LNCS series, vol. 3305, pp. 861-870, 2004
23. P. Tošić, G. Agha. "On computational complexity of counting fixed points in certain classes of graph automata", *Proc. of the 4th Int'l Conf. on Unconventional Computation*, (UC'05), Springer LNCS series, vol. 3699, pp. 191-205, 2005
24. P. Tošić, G. Agha. "Parallel vs. Sequential Threshold Cellular Automata: Comparison and Contrast", session *Complex Systems Methods 1*, in *Proc. European Conference on Complex Systems* (ECCS'05), Euro. Complex Systems Society, 2005
25. P. Tošić, G. Agha. "On Computational Complexity of Predicting Dynamical Evolution of Large Agent Ensembles", *Proc. of the 3rd European Workshop on Multiagent Systems* (EUMAS'05), pp. 415-426, Flemish Academy of Sciences, 2005
26. S. Vadhan. "The Complexity of Counting in Sparse, Regular and Planar Graphs", *SIAM J. Computing*, vol. 31 (2), pp. 398-427, 2001
27. I. Wegener. "*The Complexity of Boolean Functions*", Teubner Series Comp. Sci., Wiley, 1987
28. G. Weiss (ed.), "*Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*", The MIT Press, Cambridge, Massachusetts, 1999
29. S. Wolfram. "Theory and applications of cellular automata", World Scientific, 1986
30. S. Wolfram (ed.). "Cellular Automata and Complexity (collected papers)", Addison-Wesley, 1994