Chapter 1

Computational Complexity of Counting in Sparsely Networked Discrete Dynamical Systems

Predrag T. Tošić
Department of Computer Science
University of Illinois at Urbana Champaign
p-tosic@cs.uiuc.edu

We study the *computational complexity of counting* various types of configurations in certain classes of *network automata* viewed as discrete dynamical systems. Examples of types of configurations of interest are the stable or fixed point configurations (FPs), the unreachable or *garden of Eden* configurations (GEs), and the predecessor configurations. In particular, we show in this paper that counting the fixed points of Sequential and Synchronous Dynamical Systems (SDSs and SyDSs, respectively) is, in general, computationally intractable. This intractability is shown to still hold when each node of an SDS or SyDS is required to update according to either (i) a monotone linear threshold Boolean function, or (ii) a symmetric Boolean function. Furthermore, the hardness of the exact enumeration of FPs holds even in some severely restricted cases. Concretely, counting FPs of symmetric (alternatively, monotone) Boolean SDSs and SyDSs is hard even when the nodes of an SDS or SyDS use at most two different symmetric (monotone) Boolean functions, and, additionally, when the underlying graphs are constrained so that each node has only c = O(1) neighbors for small values of c. Our results have considerable implications for a number of domains, from *ad hoc* communication networks and loosely coupled multi-agent systems in distributed AI, to connectionist AI (Hopfield networks), statistical physics (the Ising model and spin glasses), and theoretical biology (Kauffman's random Boolean networks).

1 Introduction

We study certain classes of *network automata* that can be used as an abstraction of networks of reactive agents or complex dynamical systems whose complexity stems from coupling and interaction among their simple individual components. These network automata can also be viewed as a theoretical model for the computer simulation of a broad variety of computational, physical, social, and socio-technical distributed infrastructures. In the present work, as well as in several prior, loosely related papers (see, e.g., [2, 3, 4, 5, 6, 7, 8, 18, 20, 21, 22, 24]), the general approach has been to study mathematical and computational *configuration space properties* of such network automata: what are the possible *global behaviors* of the entire system, given the simple local behaviors of its components, and the interaction pattern among those components.

Our recent [20, 21, 22, 24, 25] and ongoing [23] research focus has been on determining *how many* fixed point and other types of configurations of interest such network automata have, and *how hard* is the computational problem of *counting* those configurations. In this paper, we establish the computational intractability of enumerating the fixed point configurations of *sparse Sequential* and *Synchronous Dynamical Systems*, as well as (when applicable) *sparse discrete Hopfield networks*, whose node update rules are restricted to either *symmetric functions* or *monotone linear threshold functions*. Moreover, we show that intractability of the exact enumeration of fixed points holds even when the maximum node degree in the underlying graph is bounded by a small constant. Thus, for the discrete dynamical systems that can be abstracted as network automata, a complex and generally unpredictable global dynamics can be obtained even via *uniformly sparse* couplings of simple local interactions.

2 Preliminaries

Sequential Dynamical Systems (SDSs) have been proposed as an abstract model for computer simulations [5, 6, 7, 8, 9]. This model has been applied in the context of modeling and simulation of large-scale socio-technical systems, such as, e.g., the *TRANSIMS* project at the Los Alamos National Laboratory [10]. 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. $F = (f_1, f_2, \ldots, f_n)$ 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, we arrive at the definition of *Synchronous Dynamical Systems* (SyDSs) [2].

A *configuration* of a Boolean SDS $\mathcal{S} = (G, F, \Pi)$ or an SyDS $\mathcal{S}' = (G, F)$ is a vector $(b_1, b_2, \ldots, b_n) \in \{0, 1\}^n$, where b_i is the state of a node v_i, for $1 \le i \le n$. A configuration $\mathcal{C}$ can also be thought of as a function $\mathcal{C} : V \to \{0, 1\}^n$. A single SDS or SyDS transition from one configuration to another is obtained by updating the state of each node v_i using the corresponding Boolean function f_i. The node updates are carried out either synchronously in parallel (in case of an

SyDS), or in the order specified by the permutation Π (SDS). 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 $\mathcal{C}' = F_{\mathcal{S}}(\mathcal{C})$ 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 $F_{\mathcal{S}} : \{0,1\}^n \to \{0,1\}^n$ is a total function on the set of global configurations. 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 [20, 21].

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 S(y)DS is deterministic, each vertex in its phase space has the out-degree of 1. A global configuration $\mathcal{C}$ is called a *fixed point* (FP) if $F_{\mathcal{S}}(\mathcal{C}) = \mathcal{C}$.

3 Complexity of Counting: Main Results

Counting or *enumeration* problems naturally arise in many contexts, from approximate reasoning and Bayesian belief networks in AI (e.g., [19]), to network reliability and fault-tolerance [26], to statistical physics [16]. Counting problems often arise in the context of discrete dynamical systems, as well [12, 20]. Being able to efficiently solve certain counting problems is essential for the full understanding of the underlying dynamical system's qualitative behavior. Perhaps the most fundamental counting problem about any dynamical system is to determine (or estimate) the number of *attractors* of that system [1]; these attractors correspond to our *fixed points*. For example, in *deterministic* discrete dynamical systems that are temporal cycle-free, such as the *asynchronous* (that is, sequential) discrete Hopfield networks with *linear threshold* update rules [15], the number of fixed points equals the number of possible dynamical evolutions of the system. Another interpretation of the FP count in the context of Hopfield networks is that it tells us *how many distinct patterns* such a network can *memorize* [1, 15].

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 [7, 8, 18], as well as Laubenbacher and Pareigis [17], investigate various mathematical properties of sequential dynamical systems. Barrett *et al.* study the computational complexity of several problems about configuration spaces of S(y)DSs in [4, 5]. Problems related to the existence of *fixed point* configurations are studied in [6].

Our subsequent work further builds on the results in [6] 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 [20, 21, 22, 24, 25]. Due to space constraints, in the rest of this paper we only summarize the main results on the computational complexity of enumerating FP configurations for certain restricted classes of Boolean S(y)DSs, and for the related binary-valued discrete Hopfield networks.

3.1 Counting Fixed Points of Symmetric Boolean SDSs and SyDSs

References [7, 8, 24] study SDSs with *symmetric Boolean* node update functions. These are the functions with Boolean domain (and range), and such that a function's value does not depend on the order in which the input bits are specified; that is, it only depends on how many of its inputs are 1. Symmetric functions provide one possible way to model the *mean field effects* used in statistical physics and studies of other large-scale systems [6].

We establish in [21, 24] that counting fixed points of Boolean SDSs and SyDSs with symmetric update rules is, in general, intractable:

Lemma 3.1 *[] The problems of counting the fixed point configurations of* symmetric Boolean *SDSs and SyDSs (abbreviated as* #FP-SYM-S(Y)DS*) are* #**P**-*complete.*

In the original construction used to establish the above result [21], the underlying graphs are allowed to have one or more nodes with an unbounded number of neighbors; such node(s) would correspond to the agents that can *directly interact* with many other agents. Intuitively, it does not come as a surprise that determining the exact number of stable configurations of a networked discrete dynamical system is hard when there are agents in the system that are capable of interacting with other agents *globally*. In most large-scale networks and multi-agent systems in practice, however, an agent can directly communicate with only a small number of other agents, i.e., the underlying communication topology is *sparse*. When the restrictions are imposed onto this network topology so that *every agent* can interact only locally, and only with a handful of other agents, then one would intuitively expect that many properties of the collective dynamics, such as determining the number of FPs, should become tractable. However, we argue below that, in general, that intuition does not hold.

Namely, we prove in [24, 25] that enumerating FPs of symmetric Boolean S(y)DSs remains intractable, even when the underlying graph, that is, the communication network topology of the agents, is *uniformly sparse*. More concretely, the #FP-SYM-S(Y)DS problem remains #**P**-complete even when each node of an SDS or SyDS has only $O(1)$ neighbors [24]. In particular, the intractability of the FP enumeration problem holds even when the underlying graph is 3-*regular* and also *bipartite* [25].

Proposition 3.1 *[] The problems of enumerating the fixed point configurations of symmetric Boolean SDSs and SyDSs are* #**P***-complete even when the underlying graphs are required to be both* 3-regular *and* bipartite.

The full proof of this interesting result can be found in our recent paper [25].

3.2 Counting Fixed Points of Monotone Linear Threshold Boolean SDSs and SyDSs

Monotone Boolean functions, formulae and circuits have been extensively studied in many areas of computer science, from machine learning to connectionist AI to VLSI design [27]. Cellular and other network automata with the local update rules restricted to monotone Boolean functions have also been of a considerable interest (e.g., [6, 20]). The problem of counting the FPs in *monotone* Boolean SDSs and SyDSs is originally addressed in [20]. 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* [20]. 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 *(monotone) 2CNF* stands for Boolean formulae in *Conjunctive Normal Form* such that each clause contains exactly two (unnegated) literals.

Lemma 3.2 *[] Counting exactly 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 $O(n)$, where n is the number of nodes in the star graph, is* $\#\mathbf{P}$*-complete.*

Moreover, as a corollary to the results of D. Roth in [19] and S. Vadhan in [26], the problem of *approximately* counting FPs of monotone Boolean S(y)DSs to within $2^{n^{1-\epsilon}}$, in the same setting as in Lemma 3.2, is **NP**-hard [20].

We will argue in the sequel that the hardness of the exact enumeration of FPs for monotone S(y)DSs holds even when the underlying graphs are required to be *uniformly sparse.* We will also show that the problem of enumerating the stable configurations of uniformly sparse *discrete Hopfield networks* (DHNs) is computationally intractable, as well.

A *discrete Hopfield network* (DHN) [15] is made of n binary-valued nodes; the set of each node's possible states is, by convention, $\{-1, +1\}$. Associated to each pair of nodes (v_i, v_j) is (in general, real-valued) *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 *threshold,* $h_i \in \mathbf{R}$. A node v_i updates its state x_i from time step t to step $t+1$ according to

$$x_i^{t+1} \;\leftarrow\; sgn(\sum_{j=1}^{n} w_{ij} \cdot x_j^t - h_i) \tag{1}$$

We will not bother to explicitly distinguish between an S(y)DS's or DHN's node, v_i, and this node's state, usually denoted by s_i or x_i; the intended meaning will be clear from the context. In the standard DHN model, the nodes update synchronously in parallel, just like the nodes of an SyDS do. We point out that the *asynchronous Hopfield networks,* where the nodes update sequentially, one at a time, have also been studied [11, 12]. In these sequential DHNs, however, it is not required that the nodes update according to a *fixed permutation* like in our SDS model; these differences are inconsequential insofar as the fixed points are concerned [18].

In the Hopfield networks literature, the weight matrix W is usually assumed symmetric, i.e., for all pairs of indices $\{i, j\}$, $w_{ij} = w_{ji}$ holds. We call a DHN *symmetric* if its weight matrix W is symmetric. A DHN is called *simple* if $w_{ii} = 0$ for all $i = 1, ..., n$ [12].

In [11], Floreen and Orponen establish that the problem of determining the number of fixed point configurations of simple discrete Hopfield networks, with symmetric weight matrices $W = [w_{ij}]$ such that all the weights w_{ij} are integers and $w_{ii} = 0$ along the main diagonal, is $\#\mathbf{P}$-complete. The Hopfield networks for which this intractability of counting stable configurations is shown, however, are relatively dense, i.e., with a considerable number of weights $w_{ij} \neq 0$ [11]. In contrast, our results in Lemma 3.2 for monotone Boolean SDSs and SyDSs that are rather similar to binary-valued discrete Hopfield networks only allow a single node to have a large neighborhood; see [20, 22] for more details.

We argue next that counting the FPs remains computationally intractable even if *no node has more than* $O(1)$ *neighbors.* Namely, the hardness of counting FPs is shown to hold even when several considerable restrictions on the underlying graph structure, as well as the nature and the number of the node update rules, are *simultaneously* imposed. BOOL-MON-S(Y)DS below stands for a *monotone Boolean* SDS or SyDS.

Proposition 3.2 *[] Counting the fixed points of* BOOL-MON-S(Y)DS*s* exactly *is* $\#\mathbf{P}$*-complete, even when all of the following restrictions on the structure of such an S(y)DS* simultaneously *hold:*

- *the node update rules are* monotone linear threshold functions*;*
- *each node of this S(y)DS has at most three neighbors;*
- *at most two different positive integer weights are used by each local update rule – more specifically, it suffices to consider* $w_{ij} \in \{0, 1, 2\}$*;*
- *only two different monotone linear threshold rules are used by the S(y)DS's nodes.*

The proof is omitted due to space constraints; it can be found in [23].

As a consequence, we have the following result about synchronous as well as asynchronous discrete Hopfield networks with sparse weight matrices:

Corollary 3.1 *[] Determining the exact number of stable configurations of a synchronous (parallel) or an asynchronous (sequential) discrete Hopfield network with* n *nodes is* $\#\mathbf{P}$*-complete even when all of the following restrictions on the weight matrix* $W = [w_{ij}]$ *simultaneously hold: (i)* W *is symmetric:* $w_{ij} = w_{ji}$ *for all pairs of indices* $i, j \in \{1, ..., n\}$*; (ii)* $w_{ii} = 1$ *along the main diagonal for all* $i \in \{1, ..., n\}$*; (iii)* $w_{ij} \in \{0, 1, 2\}$ *for all pairs of indices* $i, j \in \{1, ..., n\}$*; and (iv) each row and each column of* W *has* at most three *(alternatively,* exactly three*) nonzero entries off the main diagonal.*

Bibliography

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

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

[3] Barrett, Chris, Harry B. Hunt, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, Richard E. Stearns, "Dichotomy Results for Sequential Dynamical Systems", Los Alamos National Laboratory Report, LA–UR–00–5984 (2000)

[4] Barrett, Chris, Harry B. Hunt, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, Richard E. Stearns. "Predecessor and Permutation Existence Problems for Sequential Dynamical Systems", Los Alamos National Laboratory Report, LA–UR–01–668 (2001)

[5] Barrett, Chris, Harry B. Hunt, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, Richard E. Stearns, "Reachability problems for sequential dynamical systems with threshold functions", *Theoretical Computer Science*, vol. 295, issues 1–3, pp. 41–64 (2003)

[6] Barrett, Chris L., Harry B. Hunt, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, Richard E. Stearns, Predrag T. Tosic, "Gardens of Eden and Fixed Points in Sequential Dynamical Systems", Proc. AA DM–CCG, *Discrete Mathematics & Theoretical Computer Science*, pp. 95–110 (2001)

[7] Barrett, Chris, Henning Mortveit and Christian Reidys, "Elements of a theory of simulation II: sequential dynamical systems" *Applied Mathematics & Computation*, vol. 107 (2–3), pp. 121–136 (2000)

[8] Barrett, Chris, Henning Mortveit and Christian Reidys, "Elements of a theory of computer simulation III: equivalence of SDS", *Applied Mathematics & Computation*, vol. 122, pp. 325–340 (2001)

[9] Barrett, Chris and Christian Reidys, "Elements of a theory of computer simulation I: sequential CA over random graphs" *Applied Mathematics & Computation*, vol. 98, pp. 241–259 (1999)

[10] Beckman, R. J., et. al. "TRANSIMS – Release 1.0 – The Dallas-Forth Worth case study", Tech. Report LA UR 97–4502, Los Alamos National Laboratory, Los Alamos, New Mexico (1999)

[11] Floreen, P. and P. Orponen, "On the Computational Complexity of Analyzing Hopfield Nets", *Complex Systems*, vol. 3, pp. 577–587 (1989)

[12] Floreen, P. and P. Orponen, "Complexity Issues in Discrete Hopfield Networks", *NeuroCOLT Technical Report Series*, NC–TR–94–009 (1994)

[13] GOLES, Eric, and Servet MARTINEZ, *"Neural and Automata Networks: Dynamical Behavior and Applications"*, Mathematics and Its Applications series (vol. 58), Kluwer (1990)

[14] GOLES, Eric and Servet MARTINEZ (eds.) *"Cellular Automata, Dynamical Systems and Neural Networks"*, Mathematics and Its Applications series (vol. 282), Kluwer (1994)

[15] HOPFIELD, John J., "Neural networks and physical systems with emergent collective computational abilities", *Proc. National Academy of Sciences (USA)*, vol. 79, pp. 2554–2558 (1982)

[16] JERRUM, M. and A. SINCLAIR, "Polynomial-time approximation algorithms for the Ising model", *SIAM Journal of Computing*, vol. 22, pp. 1087–1116 (1993)

[17] LAUBENBACHER, R. and B. PAREIGIS, "Finite Dynamical Systems" Technical report, Department of Mathematical Sciences, N. Mexico State University, Las Cruces, New Mexico (2000)

[18] MORTVEIT, Henning and Christian REIDYS, "Discrete sequential dynamical systems", *Discrete Mathematics*, vol. 226 , Issue 1–3, pp. 281–295 (2001)

[19] ROTH, Dan, "On the Hardness of Approximate Reasoning", *Artificial Intelligence*, vol. 82, pp. 273–302 (1996)

[20] TOSIC, Predrag T., "On Counting Fixed Point Configurations in Star Networks", *Advances in Parallel & Distributed Comp. Models* workshop, in *Proc. 19th IEEE Int'l Parallel & Distributed Processing Symposium* (2005)

[21] TOSIC, Predrag T., "On Complexity of Counting Fixed Point Configurations in Certain Classes of Graph Automata", *Electronic Colloquium on Computational Complexity*, ECCC–TR05–051 (2005)

[22] TOSIC, Predrag T., "Counting Fixed Points and Gardens of Eden of Sequential Dynamical Systems on Planar Bipartite Graphs", *Electronic Colloquium on Computational Complexity*, ECCC–TR05–091 (2005)

[23] TOSIC, Predrag T., "Computational Complexity of Some Enumeration Problems in Uniformly Sparse Boolean Network Automata", to appear in *Proc. of the Second Euro. Conference on Complex Systems* (ECCS'06), European Complex Systems Society (2006)

[24] TOSIC, Predrag T. and Gul AGHA, "On computational complexity of counting fixed points in certain classes of graph automata", *Proc. 4th Int'l Conf. on Unconventional Computation*, Springer LNCS series, vol. 3699, pp. 191–205 (2005)

[25] TOSIC, Predrag T. and Gul AGHA, "On Computational Complexity of Predicting Dynamical Evolution of Large Agent Ensembles", *Proc. 3rd European Workshop on Multiagent Systems* (EUMAS'05), pp. 415–426 (2005)

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

[27] WEGENER, Ingo, *"The Complexity of Boolean Functions"*, Teubner Series in Computer Science, Wiley (1987)