Self-synchronization of Cellular Automata: An Attempt to Control Patterns

J.R. Sánchez[1] and R. López-Ruiz[2]

[1] Fac. Ingeniería, Universidad Nacional de Mar del Plata,
Justo 4302, 7600 Mar del Plata, Argentine
jsanchez@fi.mdp.edu.ar
[2] Department of Computer Science and BIFI, Facultad de Ciencias,
Universidad de Zaragoza, 50009 - Zaragoza, Spain
rilopez@unizar.es

Abstract. Cellular automata display configurations that are constant in time. We implement a stochastic synchronization between the present configurations of the system and its precedent ones in order to search for these constant patterns. For most of the known evolution rules with complex behavior a dynamic competition among all the possible constant patterns is established and no stationary regime is reached. For the particular rule coded by the decimal number 18, a self-synchronization phenomenon can be obtained, even when strong modifications to the synchronization method are applied.

1 Introduction

Cellular automata (CA) are discrete dynamical systems, discrete both in space and time. The simplest one dimensional version of a cellular automaton is formed by a lattice of N sites or cells, numbered by an index $i = 1, \ldots, N$, and with periodic boundary conditions. In each site, a local variable σ_i taking a binary value, either 0 or 1, is asigned. The binary string $\sigma(t)$ formed by all sites values at time t represents a configuration of the system. The system evolves in time by the application of a rule Φ. A new configuration $\sigma(t+1)$ is obtained under the action of the rule Φ on the state $\sigma(t)$. Then, the evolution of the automata can be writen as

$$\sigma(t+1) = \Phi\left[\sigma(t)\right]. \tag{1}$$

If coupling among nearest neighbors is used, the value of the site i, $\sigma_i(t+1)$, at time $t+1$ is a function of the value of the site itself at time t, $\sigma_i(t)$, and the values of its neighbors $\sigma_{i-1}(t)$ and $\sigma_{i+1}(t)$ at the same time. Then, the local evolution is expressed as

$$\sigma_i(t+1) = \phi(\sigma_{i-1}(t), \sigma_i(t), \sigma_{i+1}(t)), \tag{2}$$

being ϕ a particular realization of the rule Φ. For such particular implementation, there will be 2^3 different local input configurations for each site and, for each one of them, a binary value can be assigned as output. Therefore there will be 2^8

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part III, LNCS 3993, pp. 353–359, 2006.

different rules ϕ, also called the *Wolfram rules*. Each one of these rules produces a different dynamical evolution. In fact, dynamical behavior generated by all 256 rules were already classified in four generic classes. The reader interested in the details of such classification is addressed to the original reference [1].

CA provide us with simple dynamical systems, in which, we would like to find different methods of synchronization. A stochastic synchronization technique was introduced by Morelli and Zanette [2] that works in synchronizing two CA evolving under the same rule Φ. The two CA are started from different initial conditions and they are supposed to have partial knowledge about each other. In particular, the CA configurations, $\sigma^1(t)$ and $\sigma^2(t)$, are compared at each time step. Then, a fraction p of the total different sites are made equal (synchronized). The synchronization is stochastic since the location of the sites that are going to be equal is decided at random. Hence, the dynamics of the two coupled CA, $\sigma(t) = (\sigma^1(t), \sigma^2(t))$, is driven by the successive application of two operators:

1. the deterministic operator given by the CA evolution rule Φ, $\Phi[\sigma(t)] = (\Phi[\sigma^1(t)], \Phi[\sigma^2(t)])$, and
2. the stochastic operator Γ_p that produces the result $\Gamma_p[\sigma(t)]$, in such way that, if the sites are different ($\sigma_i^1 \neq \sigma_i^2$), then Γ_p sets both sites equal to σ_i^1 with the probability $p/2$ or equal to σ_i^2 with the same probability $p/2$. In any other case Γ_p leaves the sites unchanged.

Therefore the temporal evolution of the system can be written as

$$\sigma(t+1) = (\Gamma_p \circ \Phi)[\sigma(t)] = \Gamma_p[(\Phi[\sigma^1(t)], \Phi[\sigma^2(t)])]. \tag{3}$$

A simple way to visualize the transition to synchrony can be done by displaying the evolution of the difference automaton (DA),

$$\delta_i(t) = \mid \sigma_i^1(t) - \sigma_i^2(t) \mid . \tag{4}$$

The mean density of active sites for the DA

$$\rho(t) = \frac{1}{N} \sum_{i=1}^{N} \delta_i(t), \tag{5}$$

represents the Hamming distance between the automata and verifies $0 \leq \rho \leq 1$. The automata will be synchronized when $\lim_{t\to\infty} \rho(t) = 0$. As it has been described in [2] that two different dynamical regimes, controlled by the parameter p, can be found in the system behavior:

$$p < p_c \rightarrow \lim_{t\to\infty} \rho(t) \neq 0 \text{ (no synchronization)},$$
$$p > p_c \rightarrow \lim_{t\to\infty} \rho(t) = 0 \text{ (synchronization)},$$

being p_c the parameter for which the transition to the synchrony occurs. When $p \lesssim p_c$ complex structures can be observed in the DA time evolution. In Fig. 1, typical cases of such behavior are shown near the synchronization transition. Lateral panels represent both CA evolving in time where the central strip displays

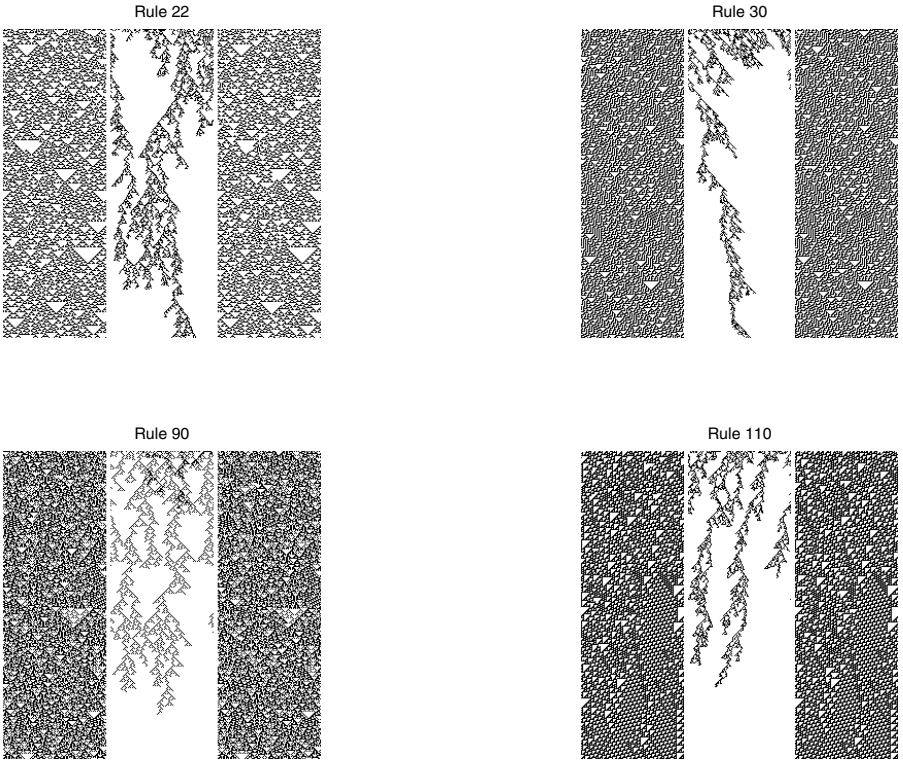

Fig. 1. Spatio-temporal patterns near the synchronization transition. Lateral strips represent both CA evolving in time where the central strip displays the evolution of the corresponding DA. Time goes from top to bottom. Lattice size is $N = 100$ and the number of iterations is $T = 250$. The stochastic coupling between both CA is $p = 0.23$.

the evolution of the corresponding DA. When p comes close to the critical value p_c the evolution of $\delta(t)$ becomes rare and resembles the problem of structures trying to percolate in the plane [3]. A method to detect this kind of transition, based in the calculation of a statistical measure of complexity for patterns, has been proposed in the Refs. [4].

2 Self-synchronization of Cellular Automata

2.1 First Self-synchronization Method

Let us now take a single cellular automaton [5, 6]. If $\sigma^1(t)$ is the state of the automaton at time t, $\sigma^1(t) = \sigma(t)$, and $\sigma^2(t)$ is the state obtained from the application of the rule Φ on that state, $\sigma^2(t) = \Phi[\sigma^1(t)]$, then the operator Γ_p can be applied on the pair $(\sigma^1(t), \sigma^2(t))$, giving rise to the evolution law

$$\sigma(t+1) = \Gamma_p[(\sigma^1(t), \sigma^2(t))] = \Gamma_p[(\sigma(t), \Phi[\sigma(t)])]. \tag{6}$$

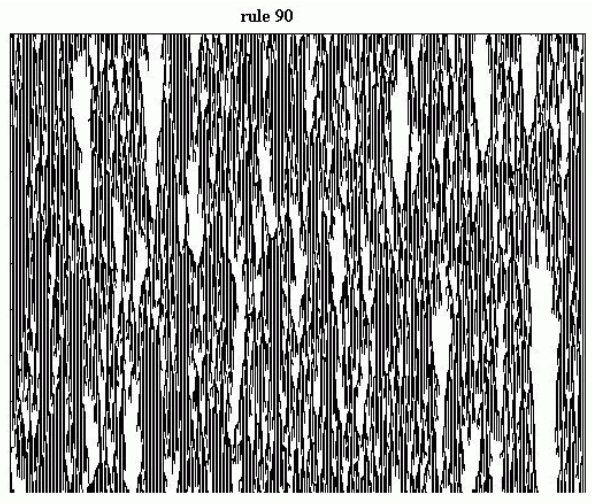

Fig. 2. Rule 90 has two stable patterns: one repeats the 011 string and the other one the 00 string. Such patterns are reached by the first self-synchronization method but there is a dynamical competition between them. In this case $p = 0.9$. Binary value 0 is represented in white and 1 in black. Time goes from top to bottom.

The application of the Γ_p operator is as follows. When $\sigma_i^1 \neq \sigma_i^2$, the sites i of the state $\sigma^2(t)$ are updated to the correspondent values taken in $\sigma^1(t)$ with a probability p. The updated array $\sigma^2(t)$ is the new state $\sigma(t+1)$.

It is worth to observe that if the system is initialized with a configuration constant in time for the rule Φ, $\Phi[\sigma] = \sigma$, then this state σ is not modified when the dynamic equation (6) is applied. Hence the evolution will produce a pattern constant in time. However, in general, this stability is marginal. A small modification of the initial condition gives rise to patterns variable in time. In fact, as the parameter p increases, a competition among the different marginally stable structures takes place. The dynamics drives the system to stay close to those states, although oscillating continuously and randomly among them. Hence, a complex spatio-temporal behavior is obtained. Some of these patterns can be seen in Fig. 2. However, in rule 18, the pattern becomes stable and, independently of the initial conditions, the system evolves toward this state, which is the null pattern in this case.

2.2 Second Self-synchronization Method

Now we introduce a new stochastic element in the application of the operator Γ_p. To differentiate from the previous case we call it $\tilde{\Gamma}_{\tilde{p}}$. The action of this operator consists in applying at each time the operator Γ_p, with p chosen at random in the interval $(0, \tilde{p})$. The evolution law of the automaton is in this case:

$$\sigma(t+1) = \tilde{\Gamma}_{\tilde{p}}[(\sigma^1(t), \sigma^2(t))] = \tilde{\Gamma}_{\tilde{p}}[(\sigma(t), \Phi[\sigma(t)])]. \tag{7}$$

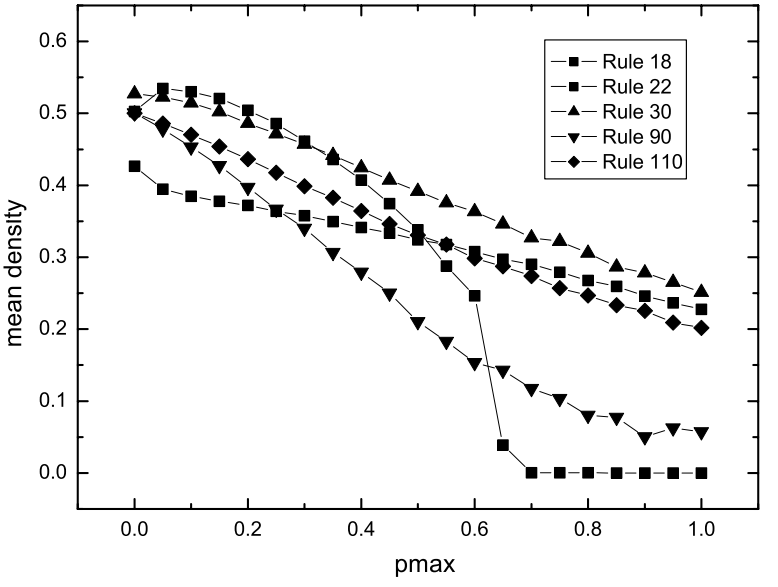

Fig. 3. Mean density ρ vs. $pmax = \tilde{p}$ for different rules evolving under the second synchronization method. The existence of a transition to a synchronized state can be clearly observed for rule 18.

The DA density between the present state and the previous one, defined as $\delta(t) = \mid \sigma(t) - \sigma(t-1) \mid$, is plotted as a function of $\tilde{p}$ for different rules Φ in Fig. 3. Only when the system becomes self-synchronized there will be a fall to zero in the DA density. Let us observe again that the behavior reported in the first self-synchronization method is newly obtained in this case. Rule 18 undergoes a phase transition for a critical value of $\tilde{p}$. For $\tilde{p}$ greater than the critical value, the method is able to find the stable structure of the system. For the rest of the rules the freezing phase is not found. The dynamics generates patterns where the different marginally stable structures randomly compete. Hence the DA density decays linearly with $\tilde{p}$ (see Fig. 3).

2.3 Third Self-synchronization Method

At last, we introduce another type of stochastic element in the application of the rule Φ. Given an integer number L, the surrounding of site i at each time step is redefined. A site i_l is randomly chosen among the L neighbors of site i to the left, $(i-L, \ldots, i-1)$. Analogously, a site i_r is randomly chosen among the L neighbors of site i to the right, $(i+1, \ldots, i+L)$. The rule Φ is now applied on the site i using the triplet (i_l, i, i_r) instead of the usual nearest neighbors of the site. This new version of the rule is called Φ_L, being $\Phi_{L=1} = \Phi$. Later the operator Γ_p acts in identical way as in the first method. Therefore, the dynamical evolution law is:

$$\sigma(t+1) = \Gamma_p[(\sigma^1(t), \sigma^2(t))] = \Gamma_p[(\sigma(t), \Phi_L[\sigma(t)])]. \tag{8}$$

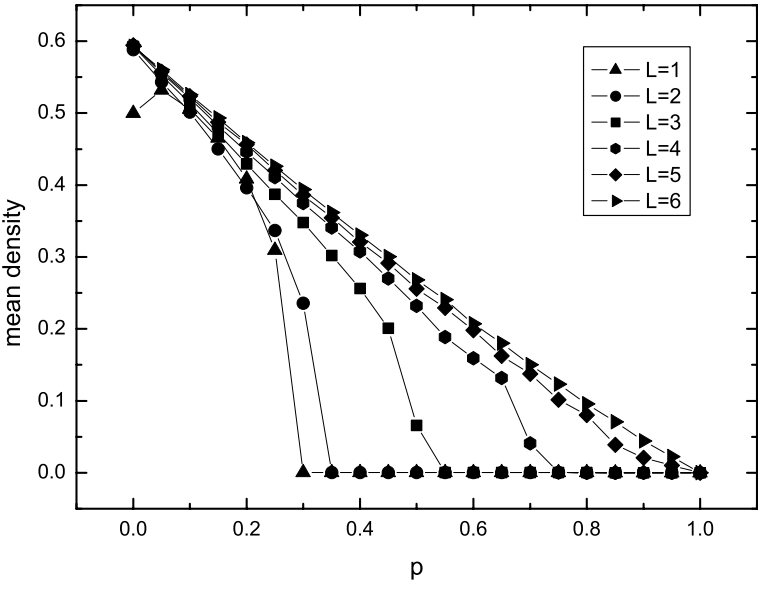

Fig. 4. Mean density ρ vs. p for rule 18 evolving under the third self-synchronization method. The existence of a transition to a synchronized state can be observed despite of the randomness in the election of neighbors within a range L, up to $L = 4$.

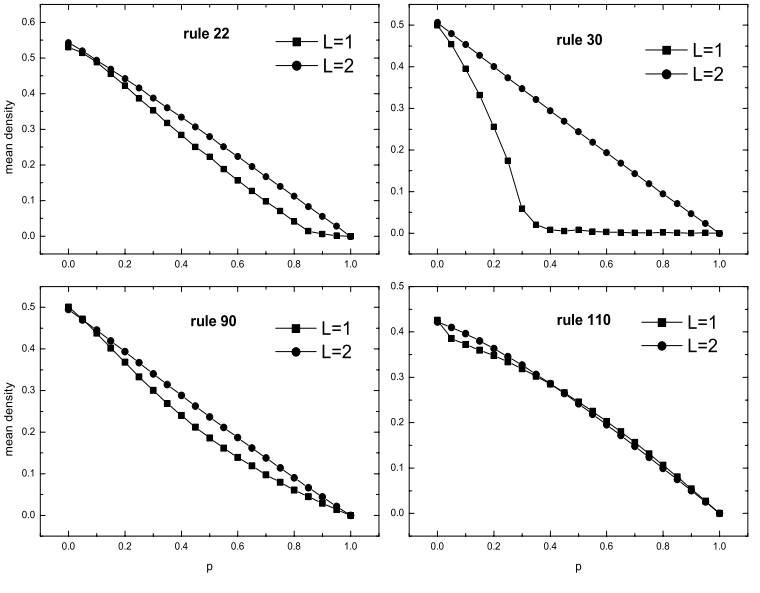

Fig. 5. Mean density ρ vs. p for different rules evolving under the third self-synchronization method. The density of the system decreases linearly with p.

The DA density as a function of p is plotted in Fig. 4 for the rule 18 and in Fig. 5 for other rules. It can be observed again that the rule 18 is a singular case that, even for different L, maintains the memory and continues to self-synchronize. It means that the influence of the rule is even more important than the randomness in the election of the surrounding sites. The system self-synchronizes and decays to the corresponding stable structure. Contrary, for the rest of the rules, the DA density decreases linearly with p even for $L = 1$ as shown in Fig. 5. The systems oscillate randomly among their different marginally stable structures as in the previous methods.

3 Conclusions

Inspired in stochastic synchronization methods for CA, different schemes for self-synchronization of a single automaton have been proposed and analyzed in this work. Self-synchronization of a single automaton can be interpreted as a strategy for searching and controlling the structures of the system that are constant in time. In general, it has been found that a competition among all such structures is established, and the system ends up oscillating randomly among them. However, rule 18 is a unique position among all rules because, even with random election of the neighbors sites, the automaton is able to reach the configuration constant in time.

References

1. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. **55** (1983) 601-644
2. Morelli, L.G., Zanette, D.H.: Synchronization of stochastically coupled cellular automata. Phys. Rev. E **58** (1998) R8-R11 ; Synchronization of coupled extended dynamical systems: a short review. Int. J. Bifurcation and Chaos **13** (2003) 1-16
3. Pomeau, Y.: Front motion, metastability and subcritical bifurcations in front motion. Physica D **23** (1986) 3-11
4. Sánchez, J.R., López-Ruiz, R.: A method to discern complexity in two-dimensional patterns generated by coupled map lattices. Physica A **355** (2005) 633-640 ; Detecting synchronization in spatially extended systems by complexity measurements. Discrete Dynamics in Nature and Society **9** (2005) 337-342
5. Toffoli, T., Margolus, N.: *Cellular automata machines: a new environment for modeling* (1987) MIT-Press
6. Ilachinski, A.: *Cellular automata: a discrete universe* (2001) World Scientific