International Journal of Bifurcation and Chaos, Vol. 13, No. 5 (2003) 1029–1054
© World Scientific Publishing Company

SYMMETRIC SELF-ORGANIZATION IN A CELLULAR AUTOMATON REQUIRING AN ESSENTIALLY RANDOM FEEDBACK. OBSERVATIONS, CONJECTURES, QUESTIONS

ANDRÉ BARBÉ and FRITZ VON HAESELER
Katholieke Universiteit Leuven, Department of Electrical Engineering, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Received November 29, 2001; Revised January 31, 2002

A triangular cellular array whose cell-states satisfy the local rule of the cellular automaton that generates Pascal's triangle modulo 2 is subjected to a feedback inspired by conditions for rotational symmetry of its state-pattern. We discuss observations and conjectures about the fact that, in order to stabilize effectively in a rotationally symmetric pattern, the underlying recursion (feedback) procedure must have an essentially random nature.

Keywords: Cellular automata; self-organization; asynchronous circuits; Gauss–Seidel algorithm; Pascal's triangle; symmetric patterns.

1. Introduction

As usual, $\mathbb{Z}$ and $\mathbb{N}$ denote the set of integers and the natural numbers (including 0), respectively. Let S be a finite set, then $\underline{s} \stackrel{\text{def}}{=} (s(k))_{k\in\mathbb{Z}} \in S^{\mathbb{Z}}$ represents a one-dimensional bidirectionally infinite sequence with values in S. A one-dimensional cellular automaton (CA) can be considered as a discrete dynamical system which iterates a mapping $A_\phi : S^{\mathbb{Z}} \to S^{\mathbb{Z}} : (A_\phi \underline{s})(k) = \phi(s(k+j_1), s(k+j_2), \ldots, s(k+j_M))$, $j_1, j_2, \ldots, j_M \in \mathbb{Z}$, where ϕ is a mapping from $S^M \to S$, known as the *local evolution rule* of the automaton. Figure 1 gives a schematic *cell-structured* representation. The CA starts in an *initial state* $\underline{s}_0 \in S^{\mathbb{Z}}$. Its state at "time" $t \in \mathbb{N}$ is given by $\underline{s}_t = A^t \underline{s}_0$. The two-dimensional sequence $O_{A_\phi}(\underline{s}_0) \stackrel{\text{def}}{=} (\underline{s}_t)_{t\in\mathbb{N}} = (s_t(k))_{k\in\mathbb{Z},t\in\mathbb{N}}$ is called *the orbit* of the CA and represents the global state-time evolution of the CA.

Cellular automata studies (also for CA defined on finite supports as opposed to the bi-infinite support $\mathbb{Z}$, and possibly extended to higher dimensions) are mainly concerned with the properties of the orbits, and the driving idea is that patterns of interesting complexity may emerge from *locally defined* evolution rules. The vast majority of studies concern complexities which mimic or provide models for certain "real world" phenomena involving locally interacting "cells", others concentrate on links with finite automata and computability aspects; and still other papers consider properties of a more mathematical nature (see, for example, the bundled seminal papers of Wolfram [1994], the collection of papers in [Gutowitz, 1991] and [Ilachinsky, 2001] for a more recent overview and extra references).

The topic in this paper extends a class of studies concerning invariance properties of certain graphical representations of the orbits, such as self-similarity [von Haeseler *et al.*, 2001], periodicity [Barbé, 1988; Barbé & von Haeseler, 2001a], coarse-graining invariance and quasi-periodic behavior [Barbé *et al.*, 1995; Barbé, 1997]. The last two properties, periodicity and coarse-graining invariance, are obtained from enforcing a global condition on the orbit while still satisfying the local evolution rule.

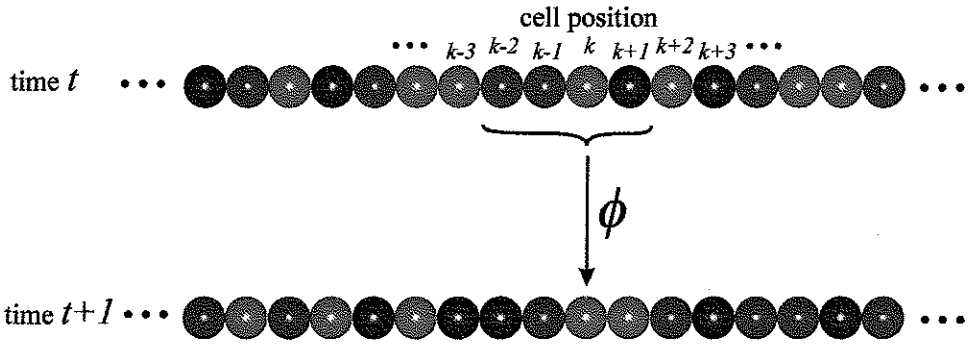


Fig. 1. A one-dimensional cellular automaton. Cells are given an integer address $k \in \mathbb{Z}$ in a natural way, and are attributed a state (here represented as a color) from a finite set of states S. Cells change their state at discrete times $t \in \mathbb{N}$, as defined by a mapping $\phi : S^M \to S$ which determines the state of cell k at time $t+1$ from the states of neighboring cells at time t, according to the evolution rule $s_{t+1}(k) = \phi(s_t(k+j_1), s_t(k+j_2), \ldots, s_t(k+j_M))$ (here $j_1 = -2$, $j_2 = -1$, $j_3 = 0$, $j_4 = 1$).

The phenomena that will be described in this paper find their origin in the observation that *rotationally invariant (symmetric)* patterns may appear in some particular representation of the orbit of the two-valued linear CA defined on $S = \mathbb{F}_2$ (the finite field with elements 0 and 1, with addition and product taken modulo 2) with the simple local rule

$$\phi(s(k), s(k+1)) = (s(k) + s(k+1)) \bmod 2 . \quad (1)$$

Figure 2 shows the abovementioned particular graphical representation of a part of this CA's orbit

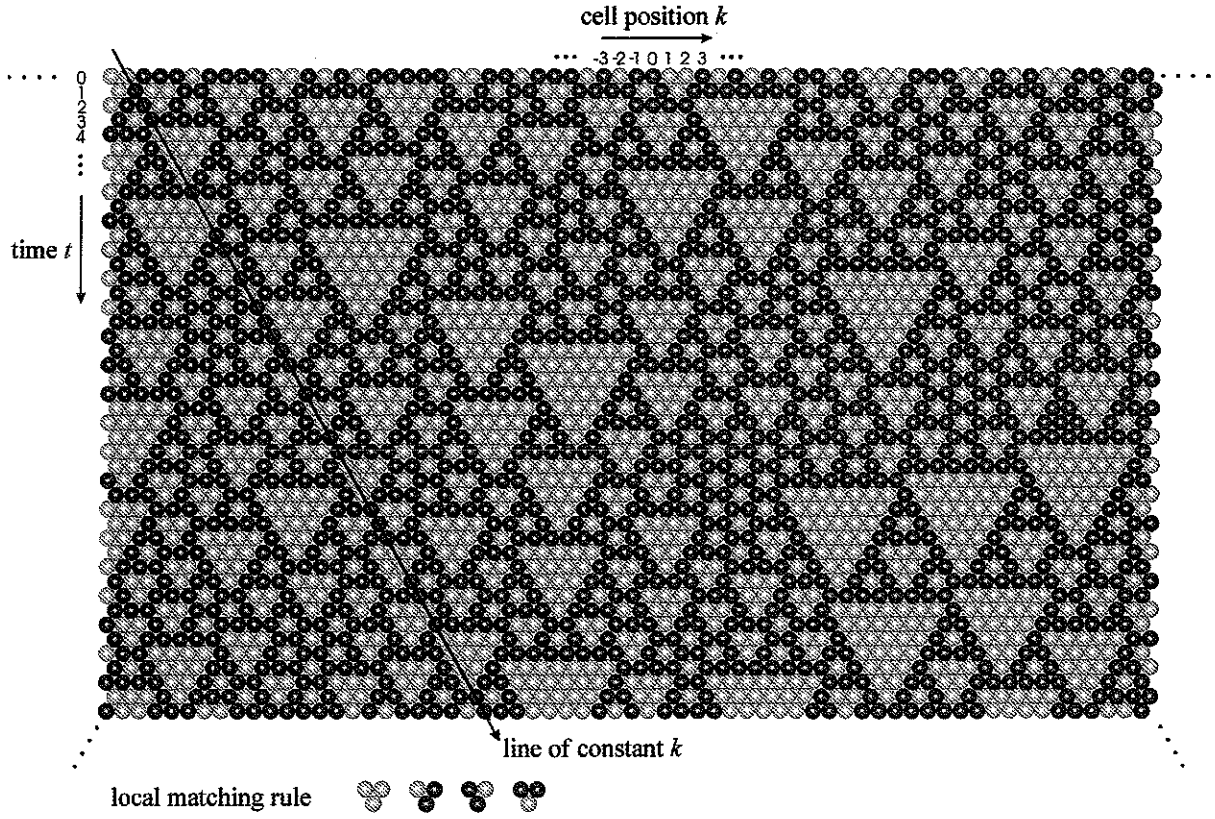


Fig. 2. An orbit $O_{A_\phi}(\underline{s}_0)$ of the binary-valued cellular automaton with local evolution rule $\phi(s(k), s(k+1)) = (s(k) + s(k+1)) \bmod 2$ and with a random initial state $\underline{s}_0$ (top row). Green cells have state-value 0, purple cells have state-value 1. The line of constant cell-position over time runs obliquely. Any orbit satisfies the *local matching rule* shown.

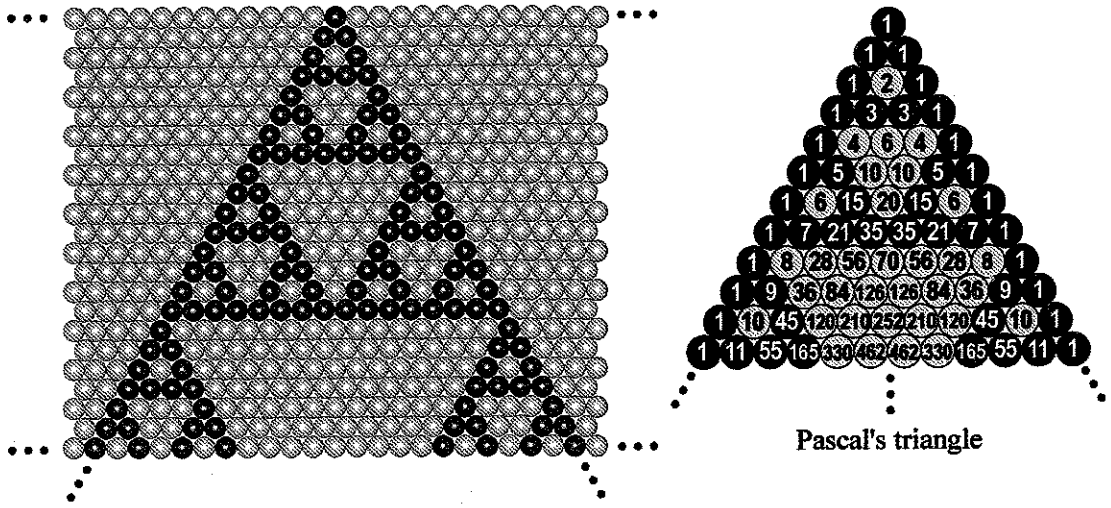
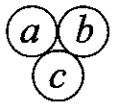


Fig. 3. The Pascal triangle pattern: it is the orbit generated by an initial state (top row) that has a single nonzero (purple) cell in its top row configuration, and is in 1–1 correspondence with Pascal's triangle mod 2, i.e. the Pascal triangle where all odd numbers (the purple ones in the figure on the right) are replaced by 1 and the even numbers (the green ones) by 0.

produced from a random initial state. Each element $O_{A_\phi}(\underline{s}_0)(t,\, k)$ of the orbit is represented by a circular "cell", and lines of constant cell position k are oblique.

Observe that any orbit of the CA under consideration satisfies the particular **local matching rule** in which any elementary triangular configuration of the form



satisfies

$$c = \phi(a,\, b) = (a + b) \bmod 2\,, \qquad (2)$$

the analytical expression of the local matching rule which reflects of course the abovementioned local evolution rule.

The graphical representation of this CA's orbit generated from an initial state $\underline{s}_0 = \underline{\delta}_{k_0}$, such that for a given $k_0 \in \mathbb{Z} : \delta_{k_0}(k_0) = 1$ and $\delta_{k_0}(k) = 0$ when $k \neq k_0$, is known to display a pattern that is isomorphic to *Pascal's triangle modulo 2* (see [Allouche *et al.*, 1996; von Haeseler *et al.*, 1992]). Part of this pattern is shown in Fig. 3: it will be referred to as the *Pascal triangle pattern.* Because of the additivity of the local CA-rule ϕ in focus here, the orbit pattern in Fig. 2 can actually be considered as the cellwise addition modulo 2 of all Pascal triangle patterns generated from the purple cells on its top row. Pascal's triangle and related patterns in CA have been the subject of several studies, see for example [Allouche *et al.*, 1996; Korec, 1993].

Figure 4 highlights several clusters of cells in the orbit of Fig. 2. All these clusters are confined in a top–down equilateral triangle, and also satisfy the local matching rule. Observe that the local matching rule exhibits the (dihedral) symmetry of an equilateral triangle: It is invariant under the group of geometric operations generated by 120°-rotation and median reflection. This fact makes it possible that some of the highlighted cell-clusters display similar symmetries, as indicated. Moreover, the content of any top–down equilateral triangular cluster in this orbit (not only the ones featuring a symmetric pattern) is completely determined by the cell-states of its top-edge. Any pattern which may appear in such a triangular cluster which satisfies the particular local matching rule presented above will be called a **triangle-configuration**, denoted as **$\bigtriangledown$-configuration**. The **size** N of a $\bigtriangledown$-configuration is the number of cells in each of its edges. $\bigtriangledown_N$ will explicitly refer to a $\bigtriangledown$-configuration of size N.

The rest of the paper will deal with isolated $\bigtriangledown$-configurations and more in particular, with the behavior of certain iterative ways for obtaining rotationally symmetric $\bigtriangledown$-configurations, by introducing a proper feedback between certain cells. We first develop the necessary preliminaries, and describe how an electronic realization of a $\bigtriangledown$-configuration gave inspiration for all further considerations. This will again provide a case where a global (nonlocal) condition is enforced while keeping the local matching intact. Properties related to

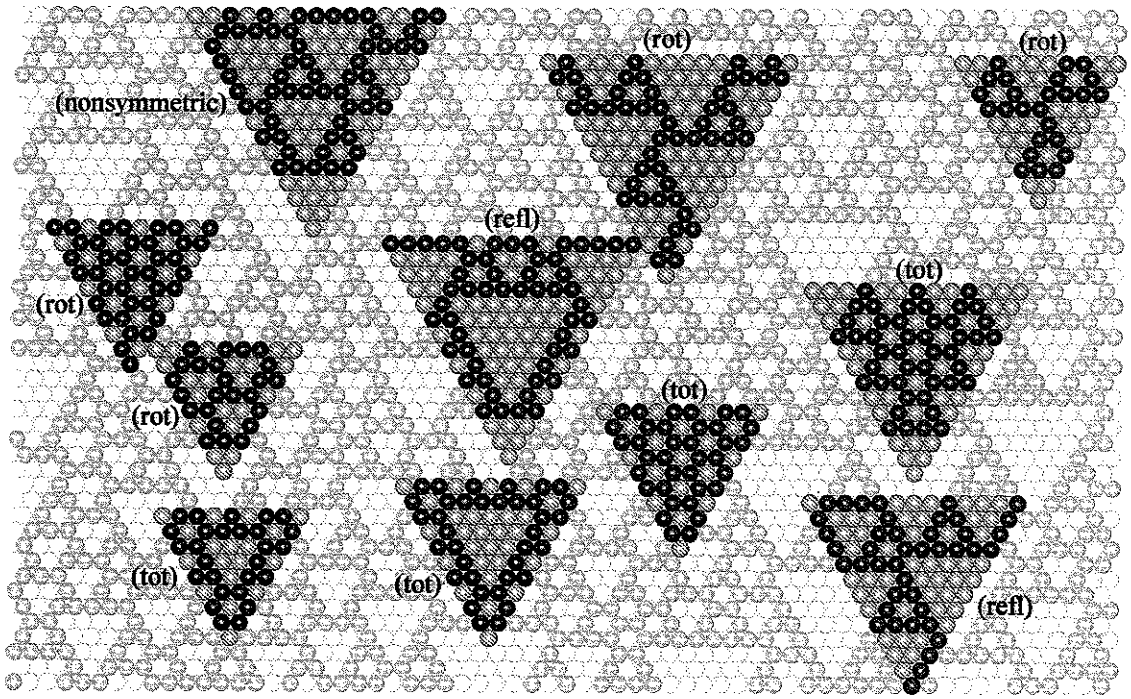


Fig. 4. Triangle-configurations in the orbit of Fig. 2. Among the ones highlighted we find a nonsymmetrical one; rotationally symmetric ones, indicated by (rot), which are invariant under 120°-rotations; reflection symmetric ones, denoted by (refl), which are invariant under reflection w.r.t. a median; and totally symmetric ones, indicated by (tot), which are invariant under rotation as well as under reflection.

symmetric $\bigtriangledown$-configurations as merely static patterns and to their enumeration were already dealt with in [Barbé, 2000].

Preliminaries

It is clear that a $\bigtriangledown_N$-configuration is completely determined by the cell-states on its top row, through application of the local matching rule. Cell-states at the top row of a $\bigtriangledown_N$-configuration will be denoted by the (column)-vector

$$\mathbf{s}_N = [s_1, s_2, \ldots, s_N]^T \in \mathbb{F}_2^N \ (T \text{ means transpose}),$$

as indicated in Fig. 5. The cell values in the right and left edges will be represented by $\mathbf{r}_N = [r_1, r_2, \ldots, r_N]^T \in \mathbb{F}_2^N$ and $\mathbf{l}_N = [l_1, l_2, \ldots, l_N]^T \in \mathbb{F}_2^N$, respectively, in the orientation also shown in Fig. 5. Let $A_N = [a_{ij}] \in \mathbb{F}_2^{N\times N}$, $i, j = 1, 2, \ldots, N$ be the $N \times N$ **Pascal matrix** defined by (see [Barbé, 2000])

$$\begin{aligned} a_{ij} &= \tbinom{i-1}{N-j} \bmod 2 \quad i = 1, 2, \ldots, N;\ j > N - i \\ a_{ij} &= 0 \qquad\qquad\qquad\quad i = 1, 2, \ldots, N;\ j \le N - i\,. \end{aligned} \tag{3}$$

The subscript N will be dropped if it is clear from the context or not necessary. Observe that the matrix A_N features Pascal's triangle modulo 2, as shown below:

$$\left[\begin{array}{cc} \cdots & \begin{array}{ccccccccc} 0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&1&1&0&0&1&1 \\ 0&0&0&1&0&1&0&1&0&1 \\ 0&0&1&1&1&1&1&1&1&1 \\ 0&1&0&0&0&0&0&0&0&1 \\ 1&1&0&0&0&0&0&0&1&1 \end{array} \\ \ddots & \vdots \qquad\qquad\quad \vdots \end{array}\right] \begin{array}{l} \\ \\ \\ A_3 \\ \\ \\ \\ A_7 \\ \\ A_{10} \\ \\ \end{array}$$

Applying the local rule shows that the relationship between the edge-vectors $\mathbf{s}$, $\mathbf{r}$, $\mathbf{l}$ of a $\bigtriangledown_N$-configuration can be expressed as [Barbé, 2000]

$$\mathbf{r} = A\mathbf{s} \tag{4}$$

(all operations modulo 2).

Because the local matching rule has geometric D_3-symmetry, i.e. it is invariant under the group of geometric operations generated by 120°-rotation and median reflection, a $\bigtriangledown$-configuration remains a $\bigtriangledown$-configuration under the geometric operations of

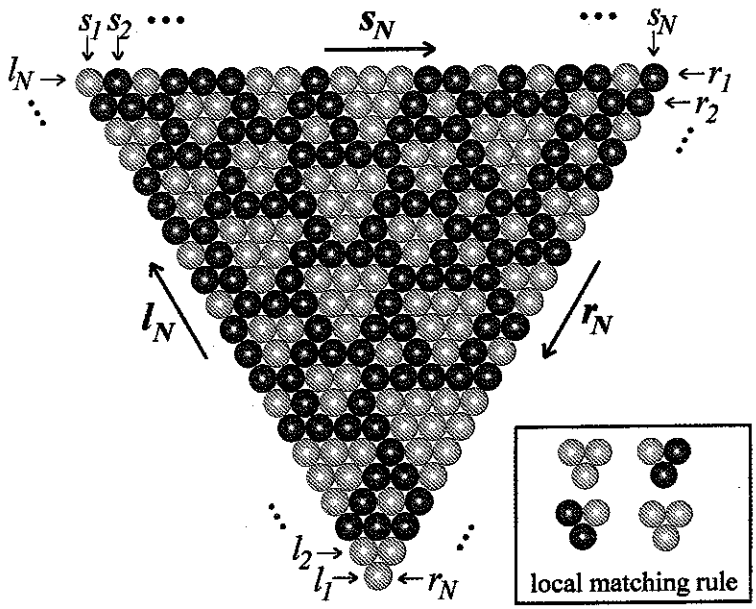


Fig. 5. A $\bigtriangledown$-configuration of size $N = 21$ satisfying the local matching rule displayed.

D_3 on the set of $\bigtriangledown$-configurations. It follows that also

$$\mathbf{l} = A\mathbf{r} \tag{5}$$

$$\mathbf{s} = A\mathbf{l}. \tag{6}$$

A proper combination of (4)–(6) then gives

$$\mathbf{l} = A^2\mathbf{s}, \quad \mathbf{s} = A^2\mathbf{r}, \quad \mathbf{r} = A^2\mathbf{l}. \tag{7}$$

The matrix A^2 is just the 180°-rotation of the matrix A [Barbé, 2000]. From (4)–(6), it also follows that

$$\mathbf{s} = A^3\mathbf{s}, \quad \text{implying that } A^3 = Id, \text{ the identity matrix in } \mathbb{F}_2^{N\times N}. \tag{8}$$

The fact that a $\bigtriangledown$-configuration remains a $\bigtriangledown$-configuration under geometric D_3-operations gives rise to the question on the existence of $\bigtriangledown$-configurations which are **rotationally symmetric** (invariance under 120°-rotations) or **totally symmetric** (invariance under 120°-rotations *and* median reflections). A necessary and sufficient condition for a $\bigtriangledown$-configuration to be rotationally symmetric is that

$$\mathbf{r} = \mathbf{s}, \quad \text{also implying that } \mathbf{s} = \mathbf{r} = \mathbf{l}, \tag{9}$$

or, by (4)

$$A\mathbf{s} = \mathbf{s}. \tag{10}$$

Thus, rotationally symmetric $\bigtriangledown$-configurations are determined by solutions $\mathbf{s}$ of equation (10), i.e. by the fixed points of the mapping $\mathbf{s} \mapsto A\mathbf{s}$ induced by the Pascal-matrix A.

The symmetry condition $\mathbf{s} = \mathbf{r}$ has inspired us for the construction of a real electronic $\bigtriangledown$-configuration of size $N = 10$, the logic scheme of which is given in Fig. 6.

The inputs/outputs of the EXOR-gates are connected to a *light emitting diode*, which lights up when the voltage level is high (state value 1) and remains dimmed when the voltage level is low (state value 0). The picture at the top of Fig. 7 shows the actual circuit.

When the circuit is switched on, all signal levels are 0, with the trivial totally symmetric $\bigtriangledown$-configuration full of zeros. Bringing one of the input/output-nodes to a high level makes the circuit almost immediately settle into a rotationally symmetric pattern like in Fig. 7. After bringing the whole circuit back to the quiescent 0-state, stimulating *the same* node may either produce the same or a different rotationally symmetric pattern (the bottom pictures in Fig. 7 were obtained that way). This already indicates that a certain randomness is involved in the asynchronous operation of the circuit (probably due to random delays and/or fluctuations in the physical initial state of the physical circuit (as opposed to the logic circuit and states)). It should be mentioned that a similar circuit of

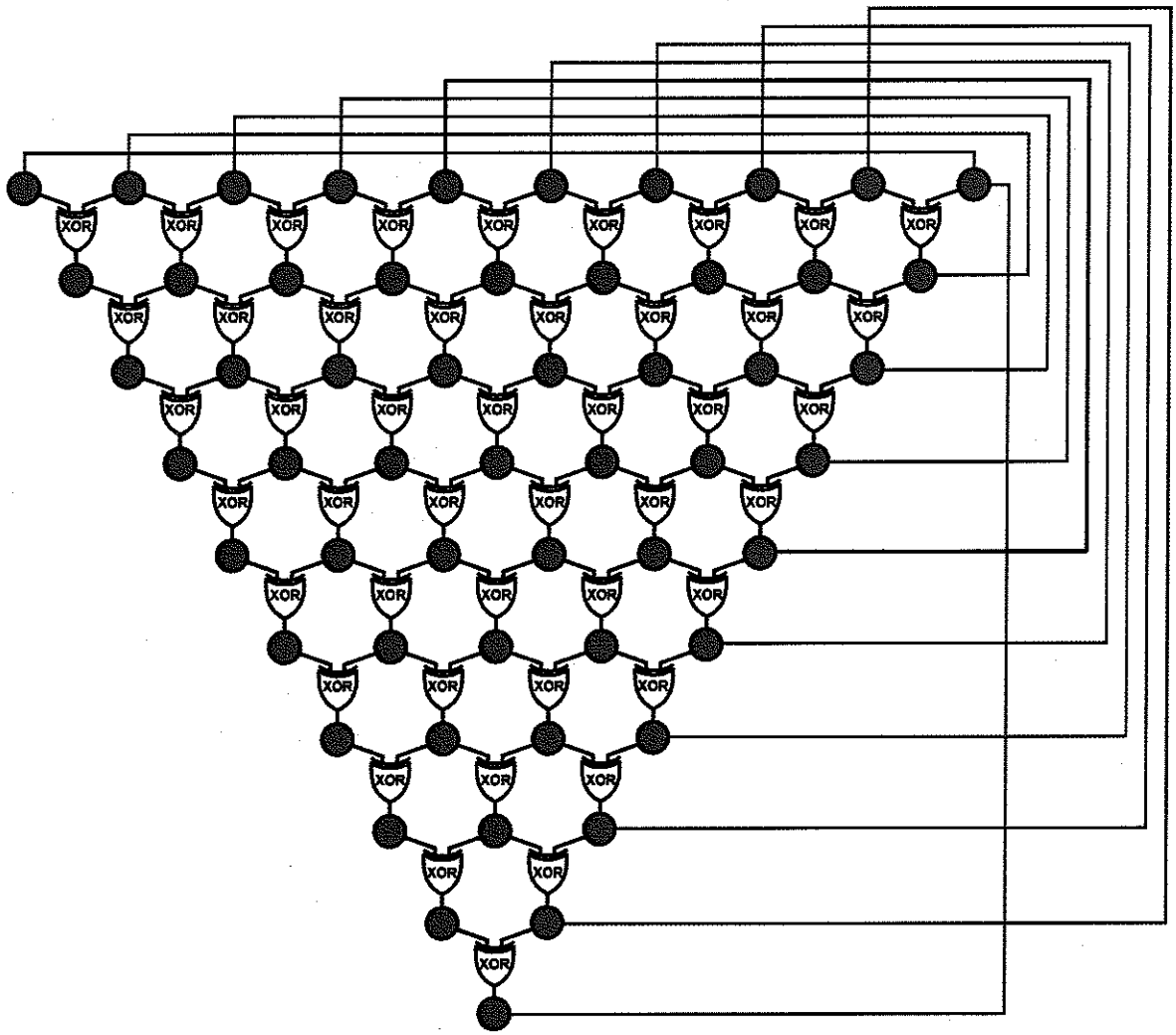


Fig. 6. Logic scheme of the self-organizing electronic circuit.

larger size ($N \geq 13$) did no longer stabilize (the exact reason not being clear, but probably due to an excessive overall gate and interconnection delay).

Suppose that the circuit would have been made to operate in synchronous mode, according to the following

Full feedback (FF) procedure:

Step 1. Start with an initial top row $\mathbf{s}$ in the $\bigtriangledown$-configuration

Step 2. Propagate the cell-states from the top-row to the right edge of the $\bigtriangledown$-configuration, i.e. calculate $\mathbf{r} = A\mathbf{s}$

Step 3. If $\mathbf{r} = \mathbf{s}$, then the $\bigtriangledown$-configuration has rotational symmetry: HALT

Step 4. Feedback the right edge to the top row, i.e. let $\mathbf{s} := \mathbf{r}$; return to Step 2.

This procedure actually realizes the recursion

$$\mathbf{s}^{(k+1)} = A\mathbf{s}^{(k)} \quad k = 0, 1, 2, \ldots \tag{11}$$

for the top-row vector $\mathbf{s}$ (superscripts refer to the iteration step). Then it is clear from Eq. (8) that starting from some $\mathbf{s}^{(0)} = \mathbf{a}$ would give a period-3 cycling of the top-row vector:

$$\mathbf{a} \mapsto A\mathbf{a} \mapsto A^2\mathbf{a} \mapsto A^3\mathbf{a} \overset{(8)}{=} \mathbf{a}$$

and the circuit would never settle in a stable (rotationally symmetric) pattern, unless $\mathbf{s}^{(0)} = \mathbf{a}$ itself already satisfies $A\mathbf{s}^{(0)} = \mathbf{s}^{(0)}$.

The fact that the circuit works in asynchronous mode actually introduces some "erratic" deviations from the synchronous algorithm. The rest of the paper will deal with recursive procedures that deviate

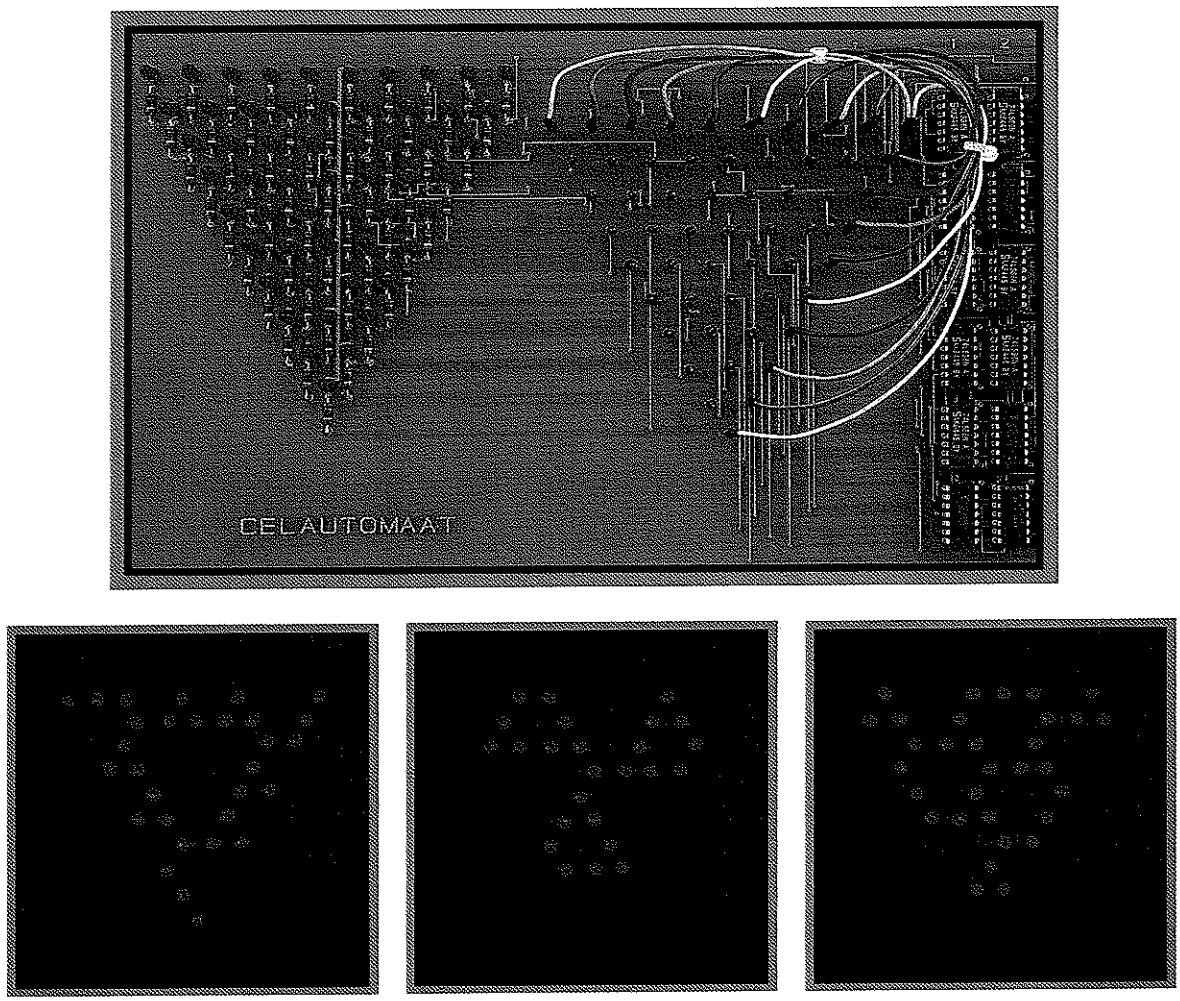


Fig. 7. The real self-organizing circuit and three different rotationally symmetric attractors for the circuit.

from the above FF-procedure by introducing some random errors in the feedback Step 4. In this way, the nonhalting FF-procedure turns into a halting one.

2. Random Feedback Procedures

Here we consider the **random m-cell feedback procedure**, defined for $m \in \{1, 2, \ldots, N-1\}$, where N is the size of the $\bigtriangledown$-configuration under consideration. This procedure is obtained from the FF-procedure by replacing Step 4 by

Step 4′. Feedback m *randomly selected cells* from the right edge to the corresponding cells on the top-row and return to Step 2.

If the m randomly selected cells correspond to the components $j_1, j_2, \ldots, j_m$ of the right edge vector $\mathbf{r} = A\mathbf{s}$, then Step 4′ actually induces the mapping

$$[s_1, \ldots, s_{j_1}, \ldots, s_{j_2}, \ldots, s_{j_m}, \ldots, s_N] \mapsto [s_1, \ldots, (A\mathbf{s})_{j_1}, \ldots, (A\mathbf{s})_{j_2}, \ldots, (A\mathbf{s})_{j_m}, \ldots, s_N]$$

for the top-row vector $\mathbf{s}$. Or, as a recursion:

$$\mathbf{s}^{(k+1)} = F^{(k)}\mathbf{s}^{(k)} \quad k = 0, 1, 2, \ldots \tag{12}$$

where $F^{(k)}$ is an $N \times N$-matrix **randomly selected** from the set

$$\mathcal{F}_{A,m} = \{F_{j_1,j_2,\ldots,j_m} | j_1, j_2, \ldots, j_m \in \{1, 2, \ldots, N\} \text{ and } j_1 < j_2 < \cdots < j_m\}, \tag{13}$$

with $F_{j_1,j_2,\ldots,j_m}$ the $N \times N$-matrix obtained from the $N \times N$-identity matrix in which the rows $j_1, j_2, \ldots, j_m$ have been replaced by the corresponding rows in the Pascal matrix A_N.

The following property is trivial:

Property 1. *For all* $m \in \{1, 2, \ldots, N-1\}$, $A\mathbf{s} = \mathbf{s}$ *is equivalent to* $F_*\mathbf{s} = \mathbf{s}$ *for all* $F_* \in \mathcal{F}_{A,m}$ *and vice versa.*

Figure 8 displays a typical evolution of a $\bigtriangledown$-configuration under a random 1-cell feedback.

Based on a large number of observations (exhaustive runs of the algorithm for sizes up to $N = 17$, and numerous trials for larger N up to 230), we formulate

Conjecture 1 (probabilistic formulation). *Let* N *be the size of the underlying* $\bigtriangledown$*-configuration, then the random* m*-cell feedback procedure, as given by Eq. (12), will eventually reach a fixed point of the mapping* $\mathbf{s} \mapsto A\mathbf{s}$ *(and halt) with probability one, for all sizes* N*, for all feedback numbers* m*, and for all initial top-row vectors* $\mathbf{s} \in \mathbb{F}_2^N$.

This conjecture means that for any $\mathbf{s}$, there is at least one product

$$\Pi_{m,\mathbf{s}} = F_{*,q} \ldots F_{*,1}\,, \quad F_{*,i} \in \mathcal{F}_{A,m} \tag{14}$$

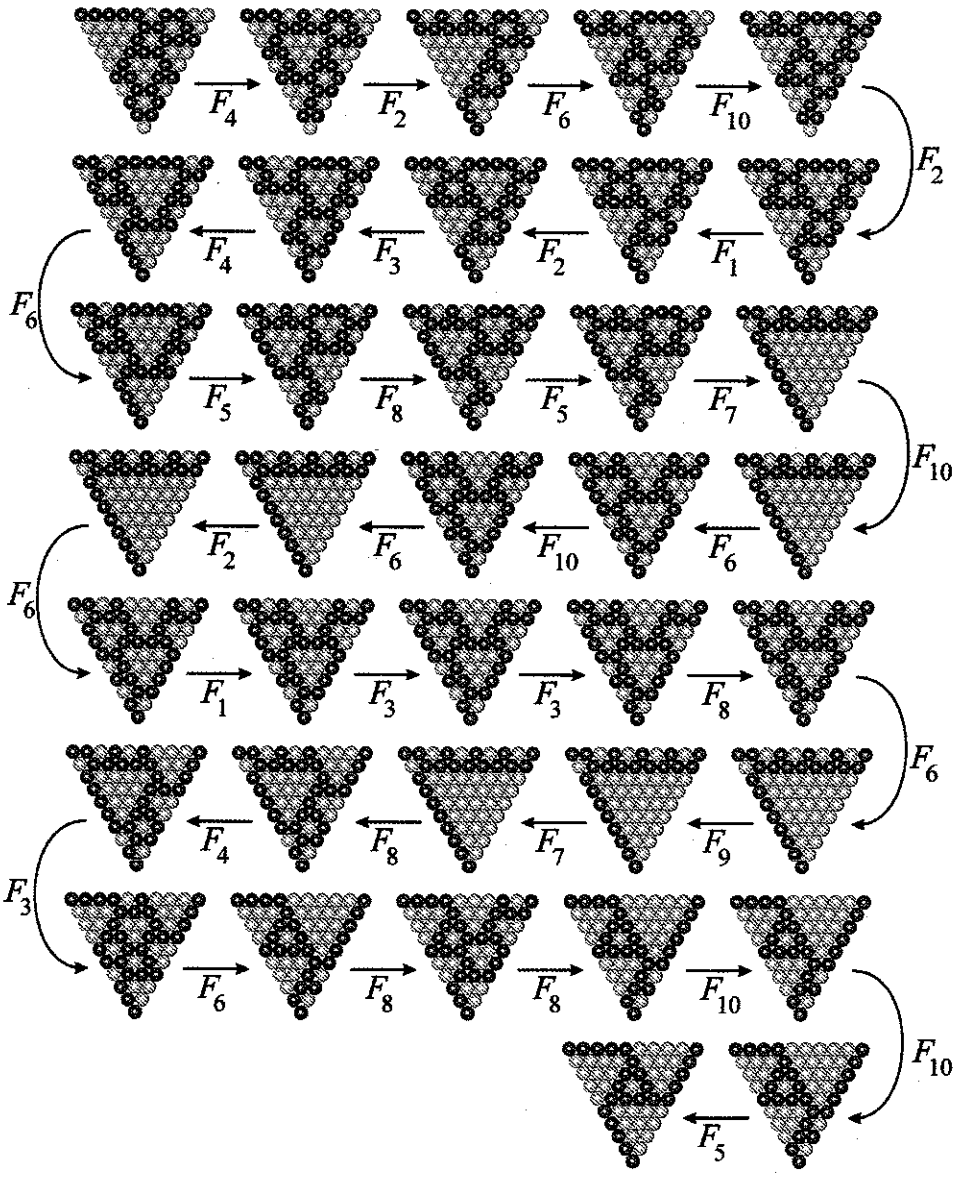


Fig. 8. Evolution of a $\bigtriangledown_{10}$-configuration in $\mathbb{F}_2$ satisfying the rule $\phi(a, b) = (a + b) \bmod 2$ under random 1-cell feedback. The green cells are in state 1, the purple ones in state 0. F_j $(j = 1, \ldots, 10)$ feeds back the state of cell j in the right edge to the corresponding cell on the top edge. The F_j-sequence was generated randomly (equal probability for $j \in \{1, \ldots, 10\}$). Observe that not every iteration step yields a new configuration and that some configurations recur. The final configuration is invariant under all F_j and is rotationally symmetric.

such that $\Pi_{m,\mathbf{s}}\mathbf{s}$ is a fixed point of A, or

$$A\Pi_{m,\mathbf{s}}\mathbf{s} = \Pi_{m,\mathbf{s}}\mathbf{s}\,. \tag{15}$$

The randomness of the procedure implies that this particular product sequence will eventually occur with probability one (in the sense that, given a finite set $\mathcal{S}$ of symbols, any given sequence of symbols from $\mathcal{S}$ will eventually occur with probability one under random selection of the symbols).

We will now report some observations that were made in an attempt to formulate a kind of inductive proof for the conjecture in the m-cell feedback case. Such an inductive proof would be nice, as we can show for small values of N, through exhaustive calculations or analysis of the corresponding state-transition graphs, that the random m-cell feedback procedure halts. Looking for an inductive proof is also stimulated by the iterative constructibility of A and of the triangular configurations for increasing problem size N [Barbé, 2000].

Two possible forms for representing the state transitions will be considered. For both, it will be convenient to encode the state vector $\mathbf{s} = [s_1, s_2, \ldots, s_N]^T$ as the decimal number $c(\mathbf{s}) = \sum_{i=1}^{N} s_i 2^{N-i} \in Z_N \stackrel{\text{def}}{=} \{0, 1, \ldots, 2^N - 1\}$.

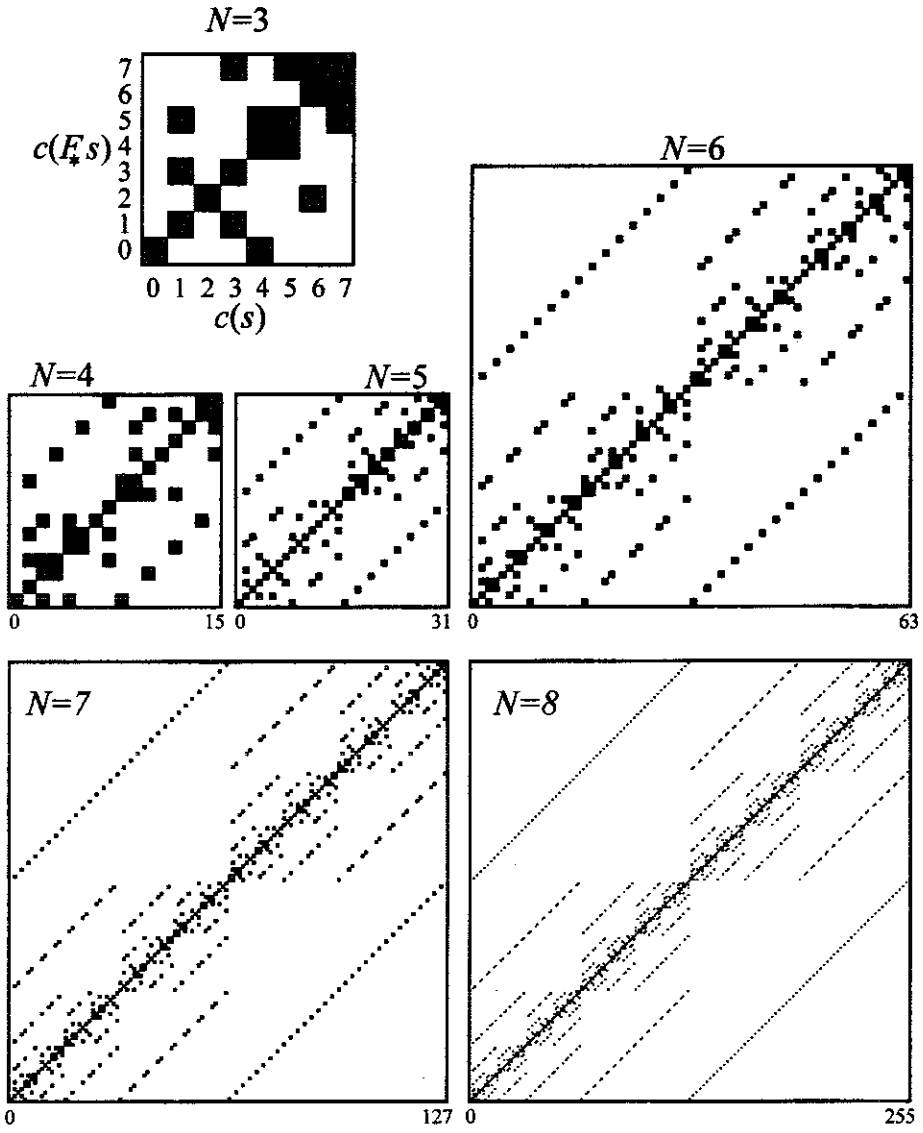


Fig. 9. State transition plots for the random 1-cell feedback algorithm for sizes $N = 3$–8. A state $\mathbf{s}$ is encoded by its decimal representation $c(\mathbf{s})$. Each state $\mathbf{s}$ maps randomly into one of the states represented by a black pixel in the column corresponding to $\mathbf{s}$.

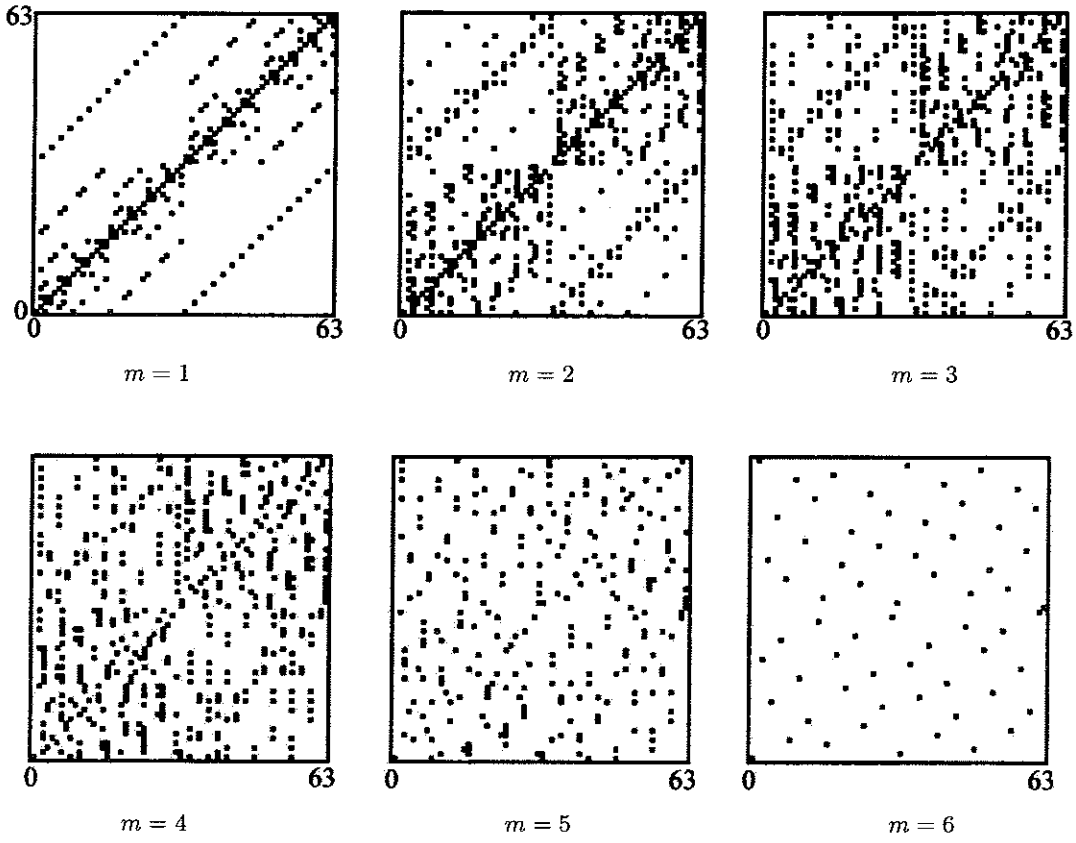


Fig. 10. State transition plots for $N = 6$ and feedback numbers $m = 1$–6.

The first representation, called the **state transition plot**, displays the points $(c(\mathbf{s}), c(F_*\mathbf{s}))$ for all $F_* \in \mathcal{F}_{A,m}$ and all $\mathbf{s} \in F_2^N$ on the point lattice $Z_N \times Z_N$. Such plots are represented in Fig. 9 for $N = 3, 4, \ldots, 8$ and $m = 1$.

Since $A\mathbf{s} = \mathbf{s}$ if and only if $F_*\mathbf{s} = \mathbf{s}$ for all $F_* \in \mathcal{F}_{A,m}$, it follows that the fixed points of A correspond to columns of the transition plot containing a unique point (necessarily on the diagonal). Observe the self-similar aspect between the transition plots. They strongly suggest that there exists an iterative way to construct them. However, any attempt to do so failed.

It is interesting to plot similar graphs for the m-cell feedback cases (see Fig. 10). This shows that the state transition plots become more dense as m grows from 1 to $\lfloor N/2 \rfloor$, while this density again decreases for m going from $\lceil N/2 \rceil$ to N. This is made plausible by the fact that $\mathcal{F}_{A,m}$ contains $\binom{N}{m}$ matrices which bring any state $\mathbf{s}$ to a next state $F_*\mathbf{s}$, thus creating at least the possibility for a denser state transition plot as m approaches $N/2$. But again, there is no obvious structural relationship between these graphs for fixed m over different N (apart from a superficial self-similarity). In terms of these transition plots, Conjecture 1 means that it is always possible to find a "cobweb"-path which brings any initial state to a fixed point.

The second representation concerns the labeled and directed **state transition graph** for fixed N, m. This graph represents the states as vertices. Two vertices (states) $\mathbf{s}_1$ and $\mathbf{s}_2$ are connected by an arrow pointing from $\mathbf{s}_1$ to $\mathbf{s}_2$ and labeled by $F_{j_1,j_2,\ldots,j_m}$ whenever the application of $F_{j_1,j_2,\ldots,j_m} \in \mathcal{F}_{A,m}$ maps $\mathbf{s}_1$ on $\mathbf{s}_2$. Figures 11(a)–11(c) show some examples. In a directed graph, one can define an **absorbing set** as being a minimal set of states such that arrows only enter but do not leave the set. In terms of transition graphs, Conjecture 1 can be reformulated as follows

Conjecture 1 (state transition graph formulation). *For any problem size N and for any feedback number $m = 1, 2, \ldots, N-1$, there are no other absorbing sets in the state transition graph than the singletons containing the fixed points of A, and there is*

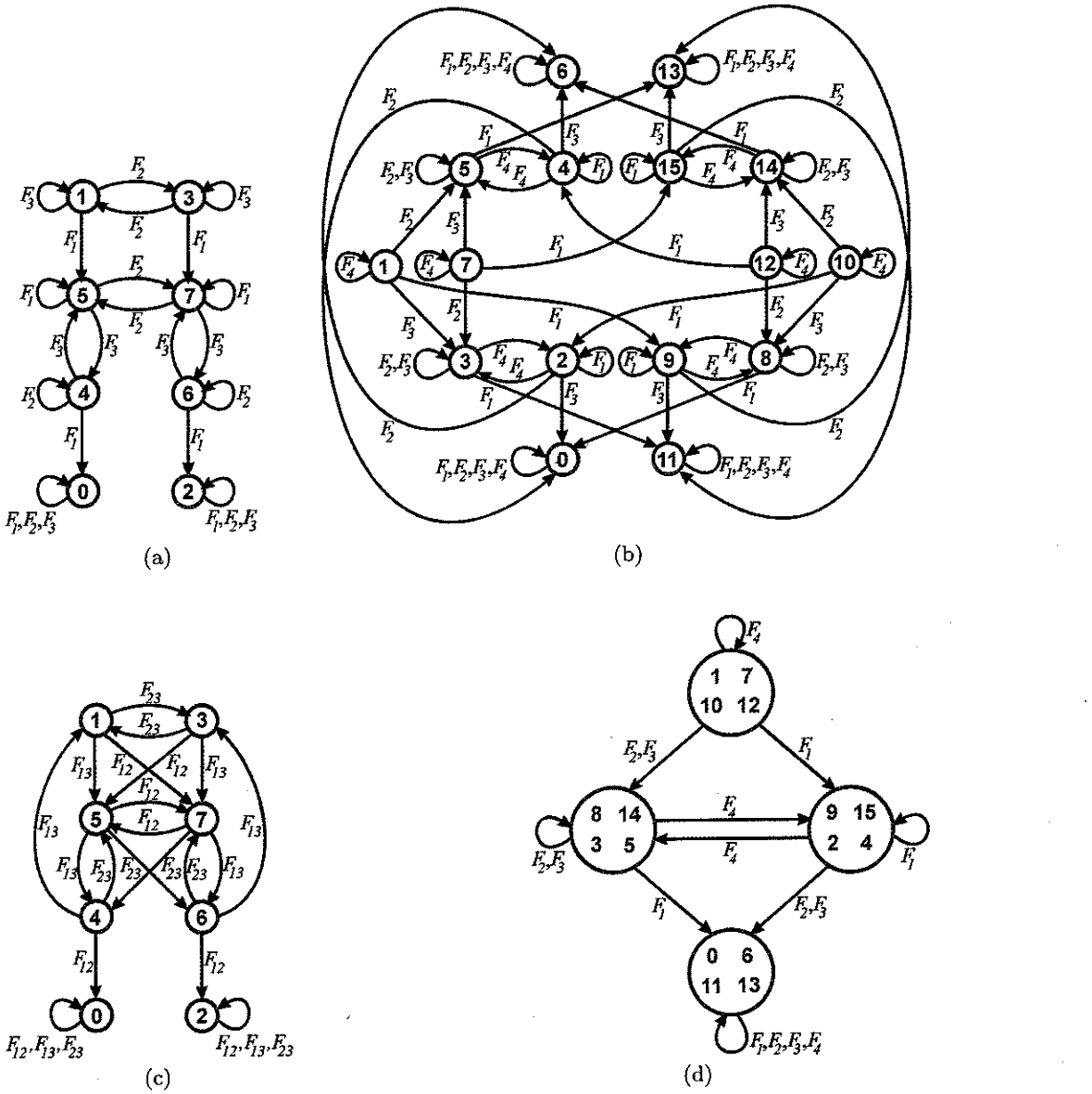


Fig. 11. State transition graphs for (a) $N = 3$, $m = 1$; (b) $N = 4$, $m = 1$ and (c) $N = 3$, $m = 2$. (d) is the reduced transition graph corresponding to (b). The states, represented by the vertices, are coded by their decimal equivalent. Red vertices correspond to absorbing states. As Markov chains, each F_* in (a) and (c) correspond to a probability 1/3; in (b) and (d) to 1/4.

always a path leading from any state to some fixed point.

Note that not all fixed points of A must be reachable from any state: it is sufficient that from any state, there is a path which leads to at least one fixed point. Again, any attempt to prove that the random m-cell procedure implies this property of the transition graph, directly or through inductive arguments, has failed so far. Any direct confirmation that this property holds (e.g. by considering related *adjacency matrices* [Thulasiraman & Swamy, 1992], which are actually directly related to the transition maps; or by applying Markov chain theory), is only feasible for rather small values of N. That is why an inductive argument would be very useful.

The Markov chain we will generally be referring to for fixed m, is the one in which each possible transition is attributed equal probability, i.e. in which each transition induced by $F_* \in \mathcal{F}_{A,m}$ occurs with probability $\binom{N}{m}^{-1}$. If an arrow has k labels, then the corresponding transition has probability $k\binom{N}{m}^{-1}$.

Observe that the state transition graphs of Figs. 11(a)–11(c) show a certain symmetry, and hence a certain redundancy. The reason is that the states can be grouped into equivalence classes, whereby two states $\mathbf{s}$ and $\mathbf{s}' \in F_2^N$ are equivalent whenever $\mathbf{s} = \mathbf{s}' + \mathbf{r}$ for some $\mathbf{r} \in \mathcal{R}$, where $\mathcal{R}$ is the set of states corresponding to a rotationally symmetric configuration. Indeed: $F_*(\mathbf{s}+\mathbf{r}) = F_*(\mathbf{s}) + F_*(\mathbf{r}) = F_*(\mathbf{s}) + \mathbf{r}$ for all $F_* \in \mathcal{F}_{A,m}$, meaning that $\mathbf{s} \overset{F_*}{\mapsto} \mathbf{s}'$ implies $\mathbf{s}+\mathbf{r} \overset{F_*}{\mapsto} \mathbf{s}'+\mathbf{r}$. The state transition graph can thus be defined on the equivalence classes $\{\mathbf{s}+\mathbf{r} | \mathbf{r} \in \mathcal{R}\}$, $\mathbf{s} \in \mathbb{F}_2^N$. Figure 11(d) shows the **reduced** transition graph corresponding to Fig. 11(b). The number of vertices in a reduced transition graph equals $2^{N-\dim\mathcal{R}} = 4^{\lceil (N-1)/3 \rceil}$ (see [Barbé, 2000] for the fact that $\dim\mathcal{R} = \lfloor N/3 \rfloor + ((N \bmod 3) \bmod 2)$). Although this reduction allows to deal with problem-sizes about 3/2 times larger than what is possible when considering nonreduced transition graphs, the basic problem of not being able to construct these graphs or their corresponding transition plots in a recursive way remains. Figure 12 displays the reduced transition plots for $N = 5$ up to $N = 13$ and feedback number $m = 1$ (compare to the nonreduced state transition plots of Fig. 9).

The random 1-cell feedback procedure and the Gauss–Seidel algorithm

The random 1-cell feedback procedure is reminescent of the iterative *Gauss–Seidel* algorithm for

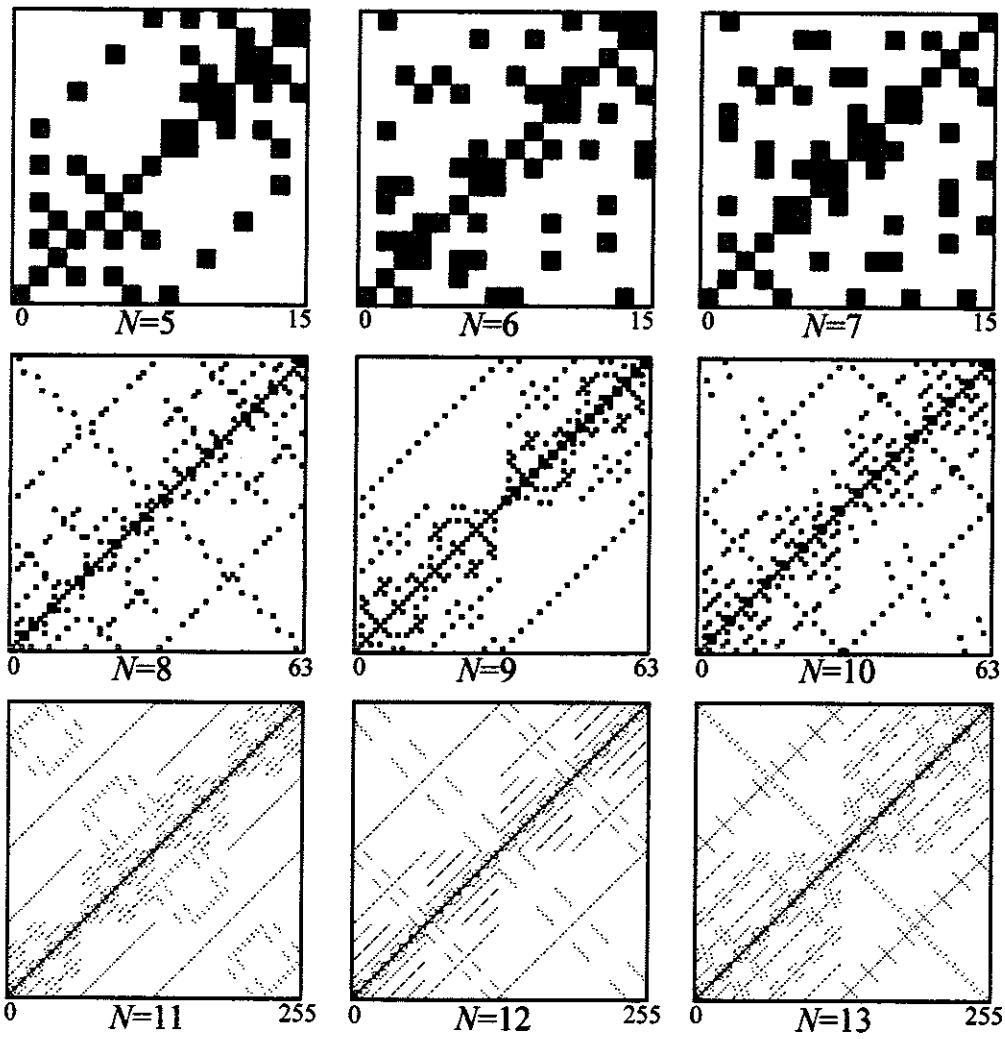


Fig. 12. Reduced transition plots for the random 1-cell feedback algorithm for sizes $N = 5$–13. The state-classes are ordered according to increasing values of their *smallest* state (i.e. the state with lowest code-number).

finding the fixed point of a contraction mapping

$$T : \mathbb{R}^N \to \mathbb{R}^N : \quad \mathbf{s} \mapsto T(\mathbf{s})$$

by the recursion

$$\mathbf{s}^{(k+1)} = S(\mathbf{s}^{(k)}),$$

where $S = T_N \circ T_{N-1} \circ \cdots \circ T_2 \circ T_1$, and $T_i(\mathbf{s}) = [s_1, \ldots, s_{i-1}, (T(\mathbf{s}))_i, s_{i+1}, \ldots, s_N]$. This is similar to Eq. (12) for $m = 1$, when $T(\mathbf{s})$ is replaced by $A\mathbf{s}$. (Any permutation of T_i can be used in the definition of S, and T_i may even correspond to disjoint component-blocks of $\mathbf{s}$ (see, e.g. [Bertsekas & Tsisiklis, 1989]). The Gauss–Seidel algorithm converges to the fixed point of T. Its proof of convergence relies explicitly on the contractivity of the mapping T with respect to some metric, which also implies the uniqueness of the fixed point of T. The fact that the random 1-cell feedback procedure considers random products of the matrices in $\mathcal{F}_{A,1}$, rather than the cyclic product $F_N F_{N-1} \cdots F_2 F_1$, makes it conceptually closer to the original Seidel algorithm [Chabert, 1999].

An important point of difference between what we are considering here and the context in which the (Gauss–) Seidel algorithm works, is that our vectors live in the finite field $\mathbb{F}_2^N$ instead of in $\mathbb{R}^N$. Robert [1995] has considered Gauss–Seidel algorithms (and also randomized versions of them (*itérations chaotiques discrètes*)) for *Boolean mappings*, using a vectorial Boolean metric. There also, convergence depends on some contractivity property of a certain matrix associated with the mapping. Although the mapping $\mathbf{s} \mapsto A\mathbf{s}$ that we are studying fits in that Boolean framework (see [Robert, 1995, Sec. 7.3]), the results on convergence fail because of the fact that the Pascal-matrix A is not contractive, as a consequence of the existence of more than one solution of the equation $A\mathbf{s} = \mathbf{s}$.

The interesting fact concerning the random m-cell feedback algorithm is that convergence occurs *despite* the lack of some contractivity property, which moreover seems incompatible with the existence of multiple fixed points. Moreover, **randomness is essential**: the recursive application of $S = F_N F_{N-1} \cdots F_2 F_1$ (or some other fixed permutation of the matrices in this product) will generally not lead to a fixed point. We have however observed that for N up to 60 and $m = 1$, that there exists a **universal path** defined by some product of matrices $F_i \in \mathcal{F}_{A,1}$, which leads *any* state $\mathbf{s} \in \mathbb{F}_2^N$ to a fixed point of A.

Here follow some universal paths for different problem sizes:

Size N	Universal Path
1	Id(Identity matrix)
2	$F_1F_2F_1$
3	$F_1F_3F_1$
4	F_2F_1
5	$F_2F_4F_2F_1F_5F_2F_1$
6	$F_3F_4F_3F_2F_1$
7	$F_2F_6F_3F_2F_1$
8	$F_4F_2F_7F_4F_3F_6F_4F_3F_2F_1$
9	$F_4F_3F_7F_4F_3F_2F_1F_9F_4F_3F_2F_1$
10	$F_5F_6F_5F_4F_3F_2F_1$
11	$F_5F_7F_4F_8F_4F_2F_{10}F_5F_4F_3F_2F_1$
12	$F_4F_3F_{10}F_6F_4F_2F_{11}F_6F_5F_4F_3F_2F_1$
13	$F_6F_8F_4F_3F_{11}F_6F_5F_4F_3F_2F_1$
14	$F_7F_8F_6F_5F_{10}F_6F_4F_2F_{13}F_7F_6F_5F_4F_3F_2F_1$
15	$F_4F_{12}F_7F_6F_5F_{11}F_7F_4F_3F_{13}F_7F_6F_5F_4F_3F_2F_1$
16	$F_8F_4F_{13}F_8F_6F_4F_2F_{15}F_8F_7F_6F_5F_4F_3F_2F_1$

These universal paths, constructed by some well-argued *symbolic manipulations* induced by the F_i-mappings, are assumed to be the shortest ones. They are not unique, nor do they necessarily correspond to a shortest path from some arbitrary $\mathbf{s}$ to one of the absorbing states. Again, no iterative way for constructing these universal paths seems to be available.

If there is a universal path, it follows that there is a product Π of F_j-matrices such that Eq. (15) is satisfied for all $\mathbf{s} \in F_2^N$, i.e. $A\Pi = \Pi$. The existence of universal paths has been established for size N up to 60 by letting a computer program grow random products Π until the condition $A\Pi = \Pi$ was satisfied.

The above observations on universal paths induce

Conjecture 2. *There exists a universal path for the random 1-cell feedback algorithm for any problem-size N.*

Questions:

- do universal paths also exist for the m-cell feedback procedures?
- is there an efficient way to find a shortest universal path? (all shortest universal paths?)

We conjecture that the answer to the first question is affirmative.

Notice that a proof of Conjecture 2 would imply the validity of Conjecture 1 (the converse is not necessarily true).

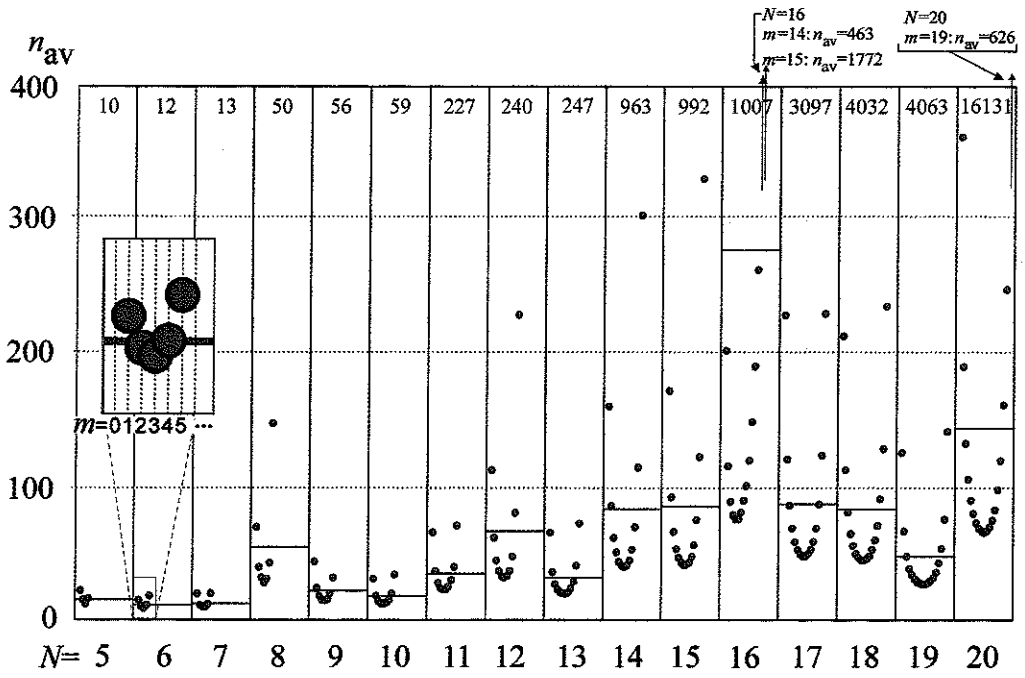


Fig. 13. The red dots represent the average number of iterations (n_{av}) before the random m-cell feedback algorithm finds a rotationally symmetric $\bigtriangledown$-configuration, for sizes $N = 5$–20, while the feedback number m ranges from 1 to $N-1$ (m is increasing from left to right in each N-column, as shown in the insert). The blue horizontal lines for a given N correspond to the global average of n_{av} over all m. The numbers at the top of the columns correspond to the average number of random draws without replacement of the elements in $\mathbb{F}_2^N$ before an element is drawn that corresponds to a rotationally symmetric $\bigtriangledown_N$-configuration. For $N = 5$ to $N = 15$, these results are based on data generated by 10 runs for each possible initial state in $\mathbb{F}_2^N$, and that for all $m = 1, 2, \ldots, N-1$. For $N = 16$, 17, only the results of two runs for each of the $2^{16} = 65536$ (respectively $2^{17} = 131072$) states were generated. For $N = 18$, two runs for each fifth initial state (i.e. states 1, 6, 11, 16, 21, 26, ..., 262141) were used; for $N = 19$, two runs for each thirteenth state (i.e. states 1, 14, 27, 40, ..., 524277) and for $N = 20$, two runs for each 31st state (i.e. states 1, 32, 63, 94, ..., 1048545).

Let us now return to the transition plots (see Fig. 10), and the observation that their densities increase as m approaches $N/2$. This seems to open at least the possibility for a larger number of paths between two given states, and maybe also for shorter paths on the average. That this is so is corroborated by the results of experiments estimating the average number of iterations of the random m-cell feedback procedure in order to reach a fixed point. Figure 13 shows the results. For $N = 16$ there seems to be an exception to the rule that the average number of iterations is minimal for $m = N/2$: this minimum occurs for $m = 5 < N/2$. For what other sizes would this be the case?

Figure 14 shows the experimental probability functions (relative frequency) of the number of iteration steps before halting for problem sizes $N = 15$ and $N = 16$, based on the same data used for Fig. 13. The shape for $N = 15$ is typical for most N. The one for $N = 16$ however, which tends to be of a more pure exponential type, also occurs for $N = 8$ (and we conjecture that this type occurs for $N = 2^k$, for which on average also substantially more time is needed to come to a halt than for neighboring values of 2^k).

Let us consider for a while the pure random way of finding a solution of $A\mathbf{s} = \mathbf{s}$ which randomly draws a vector $\mathbf{s}$ from the set of all 2^N vectors in $\mathbb{F}_2^N$, *without replacement*. The procedure stops when the last selected $\mathbf{s}$ satisfies $A\mathbf{s} = \mathbf{s}$. With $m = 2^{\lfloor N/3 \rfloor + (N \bmod 3) \bmod 2}$ the number of solutions satisfying $A\mathbf{s} = \mathbf{s}$ for problem-size N (see [Barbé, 2000]), the probability function $p(k)$ for success at the kth draw (starting at $k = 0$ for the initial draw), is given by

$$p(0) = \frac{m}{2^N}$$

$$p(k) = \left(1 - \frac{m}{2^N}\right)\left(1 - \frac{m}{2^N - 1}\right)\left(1 - \frac{m}{2^N - 2}\right)\cdots \left(1 - \frac{m}{2^N - (k-1)}\right)\frac{m}{2^N - k}$$

$$\text{for } 1 \le k \le 2^N - m\,.$$

The corresponding average number of steps $\mathrm{av}(N) = \sum_{k=0}^{2^N-1} kp(k)$ is given in the following table (rounded to the nearest integer):

N	av(N)	$2^{N-\dim\mathcal{R}}$	N	av(N)	$2^{N-\dim\mathcal{R}}$
2	2	4	17	3097	4096
3	2	4	18	4032	4096
4	2	4	19	4063	4096
5	10	16	20	16131	16384
6	12	16	21	16256	16384
7	13	16	22	16319	16384
8	50	64	23	65027	65536
9	56	64	24	65280	65536
10	59	64	25	65407	65536
11	227	256	26	261123	262144
12	240	256	27	261632	262144
13	247	256	28	261887	262144
14	963	1024	29	1046531	1048576
15	992	1024	30	1047552	1048576
16	1007	1024	...	...	...

Hereby it can be observed that for larger values of N, av(N) is roughly approximated by $2^{N-\dim\mathcal{R}}$, the number of reduced states already mentioned before. Comparing the averages in this table to the ones displayed in Fig. 13 reveals that the random m-cell feedback procedure for $N \geq 9$ is generally more efficient than the random draw procedure, except for $N = 16$, $m = 15$. It is also interesting to note that for all N which are not too small (say $N \geq 8$), the probability function $p(k)$ is of a quasi-exponential type (closely approximated by $[1/(1+\mathrm{av}(N))][\mathrm{av}(N)/(1+\mathrm{av}(N))]^k$). This makes it remarkable that for problem sizes $N = 8$, 16 (and, as conjectured for $N = 2^k$ in general), for which we already found that it takes on average substantially more time to find a solution by the random m-cell feedback procedure than for neighboring sizes, the experimentally obtained probability function is also of a more exponential type (cf. Fig. 14 for $N = 16$).

3. Generalizations

3.1. *Other generating sets for a $\bigtriangledown$-configuration*

The random m-cell feedback algorithm presented above is based on the fact that the top-row vector in the underlying $\bigtriangledown$-configuration completely determines the $\bigtriangledown$-configuration: the N top-row cells can be considered as constituting a **generating set** for the $\bigtriangledown$-configuration, as knowledge of their states completely determines the $\bigtriangledown$-configuration through application of the local matching rule. The top-row cells will be called the **natural generating set** G^{N}. Our algorithm then implements a random feedback from the cell-states in the 120° right-rotated version of G^{N} (i.e. the right edge) to G^{N} (top row). The 120° right-rotated version of G^{N} will be denoted by G^{N}_R.

Now, the top-row cells do not form the only generating set: Many others are possible. A generating set consists of N cells in a $\bigtriangledown_N$-configuration whose states can be independently fixed, and which together completely determine the $\bigtriangledown_N$-configuration (since the set of $\bigtriangledown_N$-configurations forms a vector space of dimension N, any generating set contains N elements). Figure 15 displays a few examples.

For generating sets (a) to (j), it is easy to see that the $\bigtriangledown$-configuration can be determined by successive applications of the local matching rule in elementary triangles. These generating sets will be called **primitive**. This is not the case in the examples (k) to (o), where dependencies among non-neighboring cells (but also dictated by the local matching rule) have to be taken into account in order to fill the whole $\bigtriangledown$-configuration. Such generating sets will be called **nonprimitive**. This is illustrated in Fig. 16 for examples of size $N = 6$.

Considering the nonuniqueness of generating sets, it is natural to extend what has been done before for the natural generating set G^{N} to any generating set G, by feeding back the corresponding cells of the 120° right-rotated version of G, denoted by G_R, to G, as shown in Fig. 17.

Denoting by $\mathbf{s}$ and $\mathbf{r}$ the top-row vector and right-edge vector of a $\bigtriangledown$-configuration, respectively, as before, and by $\mathbf{x}$ and $\mathbf{y}$ the vectors corresponding to the cell-states for a generating set G and its right-rotated version G_R, as shown in Fig. 17, it can be shown that

$$\mathbf{x} = P\mathbf{s}\,, \tag{16}$$

where $P = [p_{ij}]_{i,j=1,2,\ldots,N}$ is an $N \times N$-matrix with elements in $\mathbb{F}_2$ ($x_j = \sum_{i=1}^{N} p_{ij}s_i$ expresses the dependency of component x_j on the top-row vector $\mathbf{s}$, and can be determined by applying the local

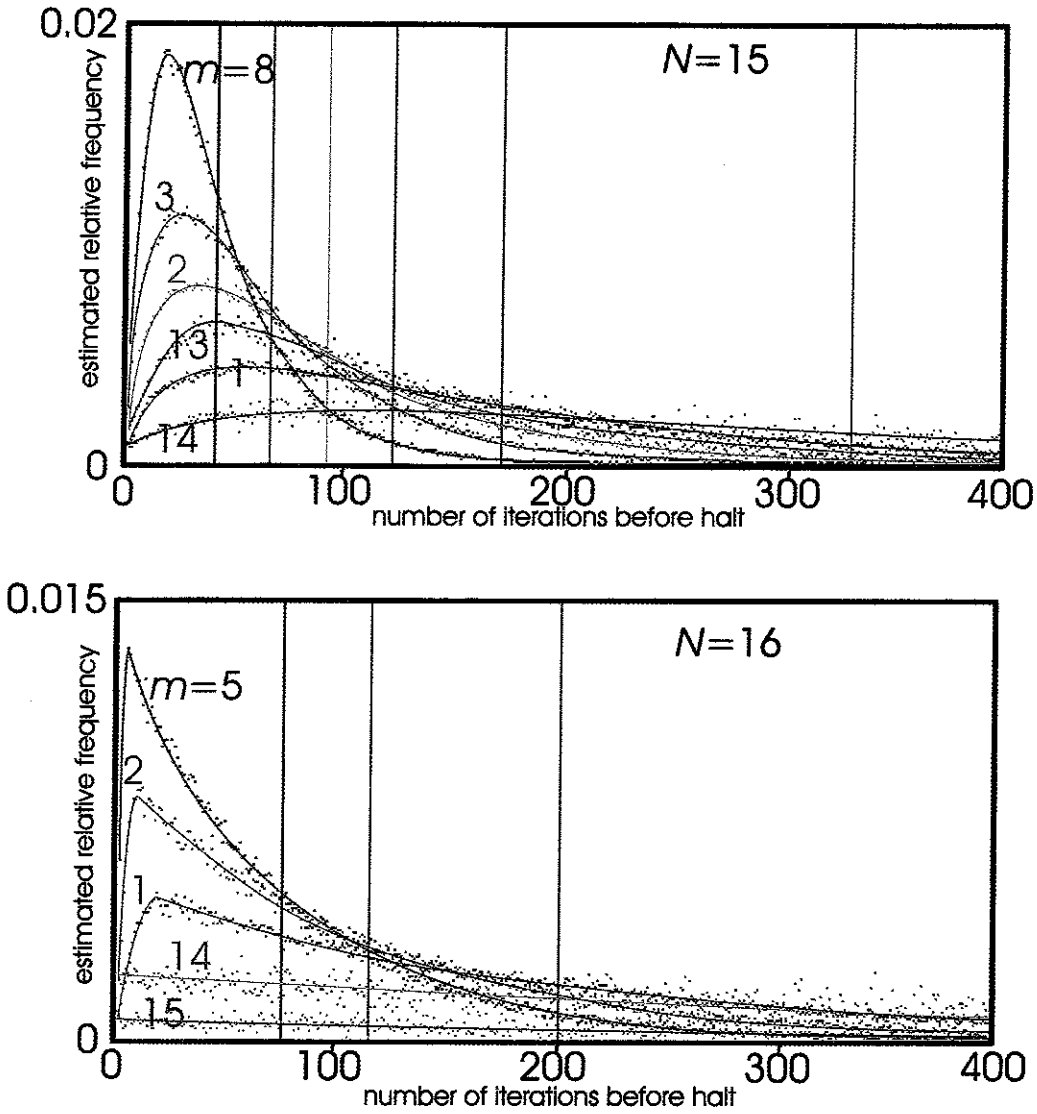


Fig. 14. The experimental relative frequency of halting times (represented by the dots) for the random m-cell feedback procedure for problem sizes $N = 15$ and 16 and different feedback numbers m. The vertical lines locate the averages. The solid curves give a rough estimate of the shape of the corresponding probability functions. Observe the difference in nature between the curves for both cases: for $N = 16$, they tend to be more exponential.

matching rule (2)). For the example of Fig. 17,

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} . \quad (17)$$

P is invertible, because it establishes a 1 to 1 mapping between the vectors associated with two distinct generating sets.

By rotational symmetry of the local matching rule, we also have

$$\mathbf{y} = P\mathbf{r} . \quad (18)$$

It follows from (18), (5) and (16) that

$$\mathbf{y} = P\mathbf{r} = PA\mathbf{s} = PAP^{-1}\mathbf{x} , \quad (19)$$

meaning that the random multicell feedback algorithm based on the new generating set G actually tries to find a fixed point of the recursion

$$\mathbf{x}^{(k+1)} = A'\mathbf{x}^{(k)} , \quad \text{with } A' = PAP^{-1} .$$

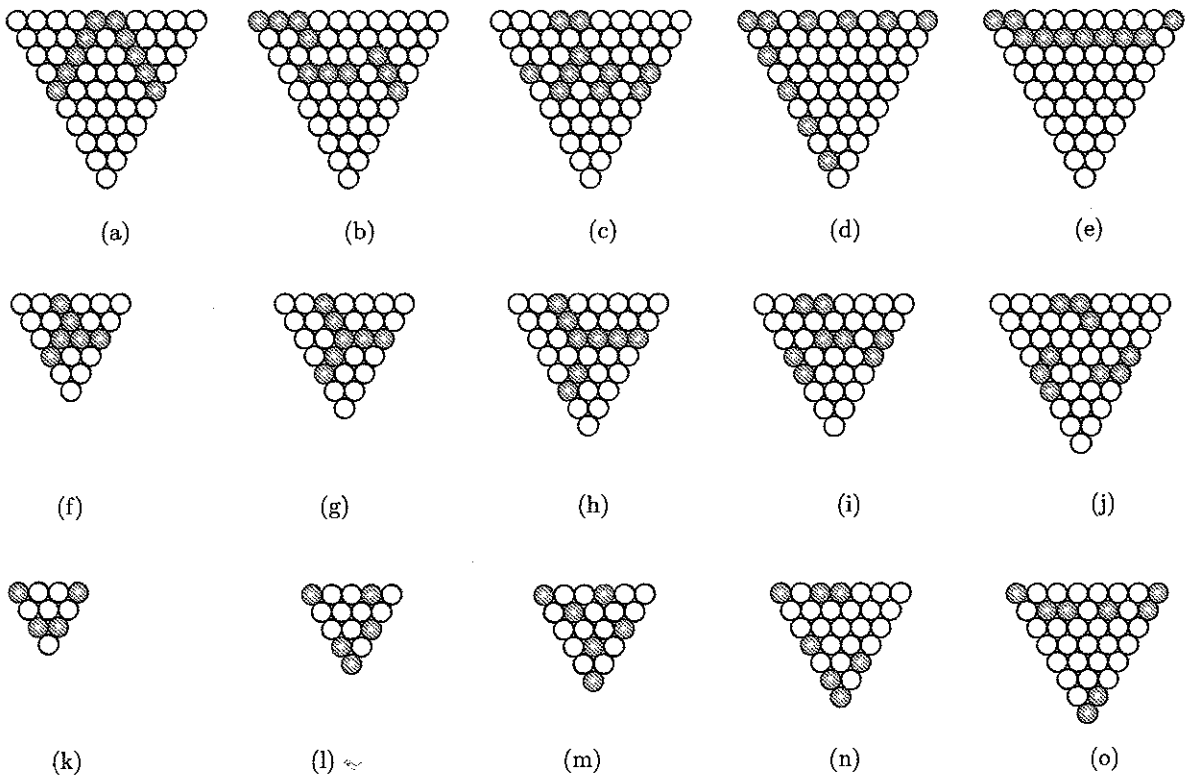


Fig. 15. Some alternative generating sets, formed by the green cells. The generating sets from (a) to (j) are primitive, the ones from (k) to (o) are nonprimitive.

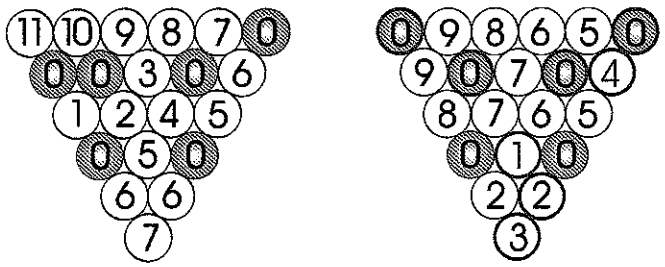


Fig. 16. Two types of generating sets. The left figure shows a primitive generating set (green cells): the ∇-configuration can be determined from successive applications of the local matching rule. The number in a cell indicates the number of times that the local rule has to be applied before the state of the cell is known. The green cells in the right figure form a nonprimitive generating set: the state of the cells containing a colored number (red, blue) can only be determined from the fact that the global effect of the local matching rule implies that the sum modulo 2 of all cells encircled with the same color equals 0; cells with a black positive number are obtained by direct application of the local matching rule.

Note that the fixed points of the map $\mathbf{x} \mapsto A'\mathbf{x}$ (based on a generating set G) correspond to those of $\mathbf{s} \mapsto A\mathbf{s}$ (based on the natural generating set G^{N}). Indeed: $A'\mathbf{x} = \mathbf{x}$ is equivalent to $PAP^{-1}\mathbf{x} = \mathbf{x}$, and implies, through (16), that $A\mathbf{s} = \mathbf{s}$ (and vice-versa).

For the example of Fig. 12,

$$A' = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} . \quad (20)$$

The $F_{j_1,j_2,\ldots,j_m}$ featuring in the original random m-cell feedback procedure for finding the solution of $A\mathbf{s} = \mathbf{s}$ are now replaced by the $F'_{j_1,j_2,\ldots,j_m}$, obtained from the $N \times N$ identity matrix by replacing rows $j_1, j_2, \ldots, j_m$ in it by the corresponding rows from $A' = PAP^{-1}$. Observe however, that in general

$$F'_{j_1,j_2,\ldots,j_m} \neq PF_{j_1,j_2,\ldots,j_m}P^{-1} . \quad (21)$$

Even worse, the set of matrices $P\mathcal{F}_{A,m}P^{-1}$ has nothing to do with the set $\mathcal{F}_{A',m}$, i.e. these sets are different. This is unfortunate, as otherwise, there

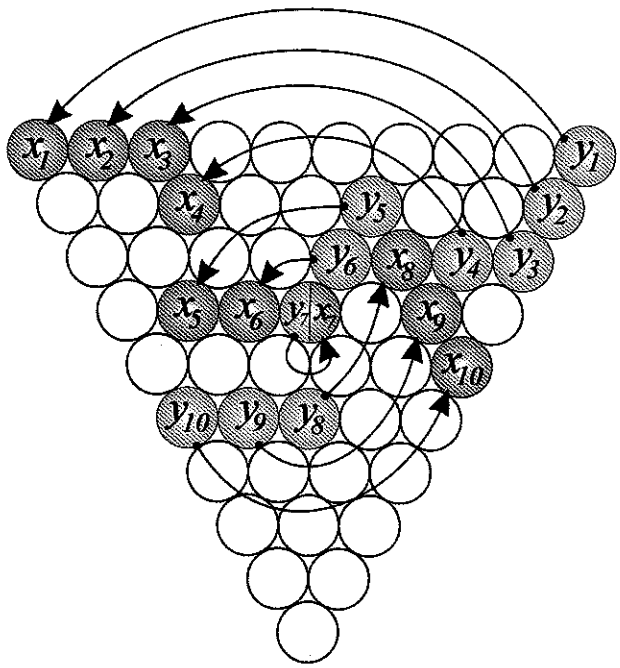


Fig. 17. Feedback paths for the generating set of Fig. 15(b). Cell-states are fed back from the right rotated version of the generating set (blue cells, states y_i) to the corresponding cells in the generating set (green cells, states x_i).

would be a complete isomorphism between the random m-cell feedback procedure for any generating set. As this is not the case, we formulate

Conjecture 3. *The random m-cell feedback procedure halts with probability one for any problem size N, for all feedback numbers $m < N$, and for any generating set G. Or: in the corresponding state transition graphs, there is always a path from any state to an equilibrium state.*

This conjecture is also corroborated by (limited) experiments: no counterexamples have been found so far.

Looking at the state transition graph related to an m-cell feedback algorithm, different generating sets generally produce different state transitions. This is illustrated in Fig. 18 for the reduced state transition graphs for $N = 4$, for three different generating sets and for 1-, 2- and 3-cell feedback.

Special generating sets for which Conjecture 3 can be proven

Let us look again at the generating sets of Figs. 15(f)–15(j). It is clear how similar sets as (f, g) can be constructed for sizes $N = 3k$, $N = 3k + 1$, respectively, and sets similar to (h, i) for sizes $N = 3k + 2$; and also a set similar to (j) for size $N = 3k$. These generating sets are particular because they are rotationally symmetric, except for one cell in the center if $N = 3k + 2$. It follows that their 120°-rotated version almost coincides with the original generating set. For such (almost) symmetric generating sets (there are more than just those displayed in Figs. 15(f)–15(j)), it is not too difficult to determine a universal sequence of corresponding $F'_{j_1,j_2,\ldots,j_m}$-matrices which lead to a rotationally symmetric solution. The reason for this is that the replacement of a component in $\mathbf{x}$, associated with some cell, only changes the $\mathbf{y} = A'\mathbf{x}$-component which corresponds to the same cell. For example, for a $\bigtriangledown_8$-configuration with the generating set from Fig. 19(a), a universal path for the 1-cell feedback procedure is

$$F'_7F'_8F'_7F'_5F'_4F'_2F'_1\,,$$

for the 4-cell feedback algorithm:

$$F'_{4567}F'_{5678}F'_{4567}F'_{3456}F'_{2345}F'_{1234}\,,$$

and for the 7-cell feedback algorithm:

$$F'_{1235678}F'_{1235678}F'_{1345678}F'_{1234567}F'_{2345678}F'_{1234567}\,.$$

(For convenience, and as there is no confusion possible by doing so, the commas in the subscript have been deleted. The $F'_{j_1,j_2,\ldots,j_m}$ are here associated with

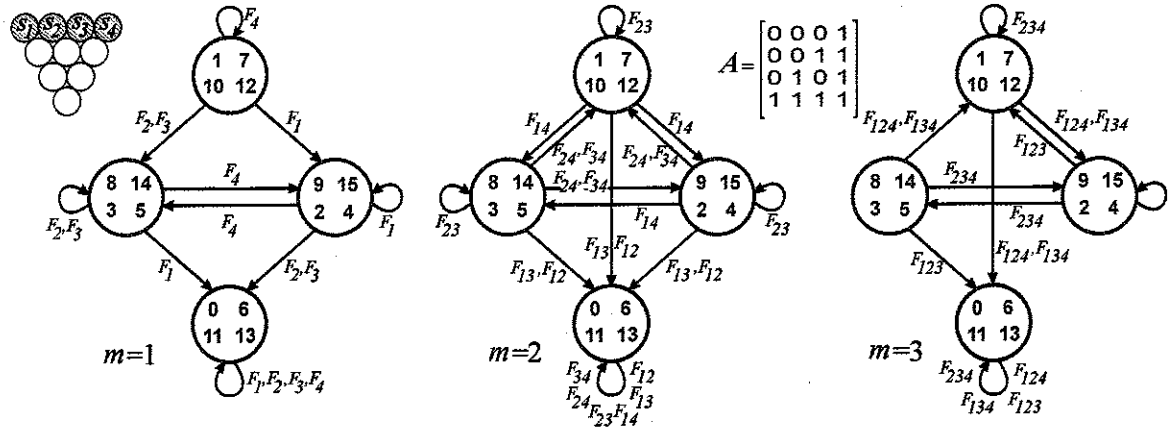
$$A' = \begin{bmatrix} 0&0&1&0&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&1 \end{bmatrix}, \qquad (22)$$

and the m-cell feedback algorithm solves $A'\mathbf{x} = \mathbf{x}$.)

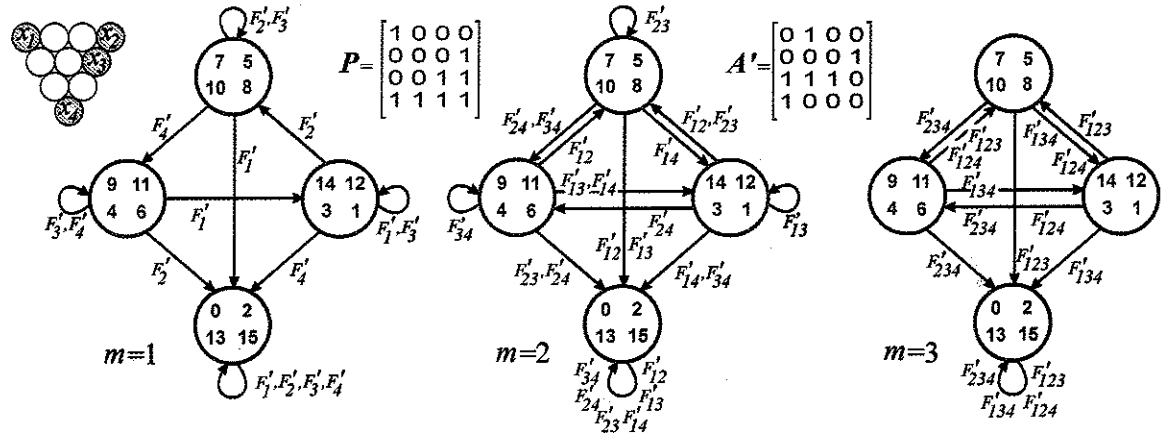
The sequence starting in Fig. 19(b) shows how the values in the generating set change under the universal 4-cell feedback path.

Questions:

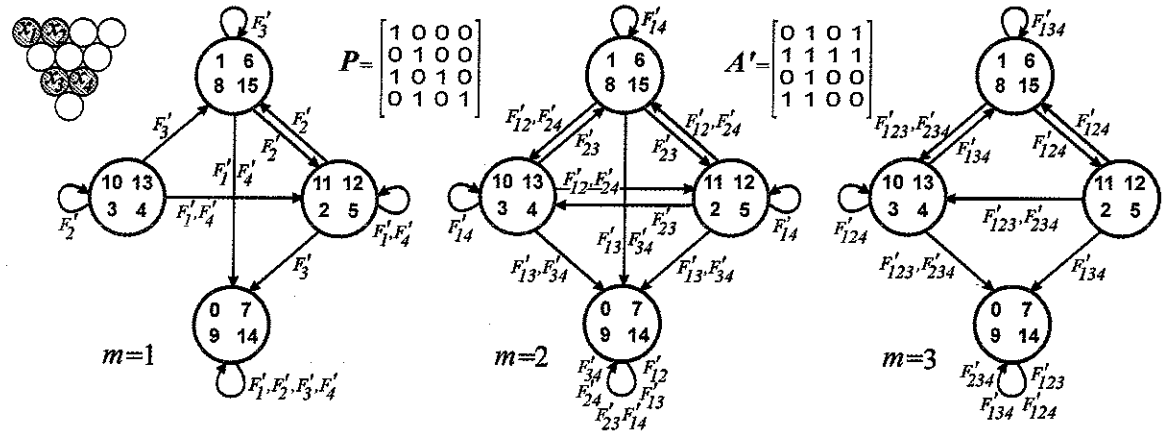
- does Conjecture 2 also hold for the m-cell feedback procedure based on any generating set, i.e. is there a corresponding universal path for all N, m and all generating sets?
- how to determine all possible generating sets for a $\bigtriangledown_N$-configuration?



(a)



(b)



(c)

Fig. 18. The reduced state transition graphs for $N = 4$ and feedback numbers $m = 1, 2, 3$, for the three different generating sets displayed under (a)–(c). (a) corresponds to the natural generating set. The states are represented by their corresponding decimal codes $\sum_{i=1}^{4} x_i 2^{N-i}$. The states as shown in (a)–(c) are in 1–1 correspondence to each other; e.g. state 6 in (c) and state 5 in (b) correspond to state 7 in (a), meaning that these states all correspond to the same ∇-configuration. The $F'_{j_1(,j_2,j_3)}$-matrices are obtained from the identity matrix in which rows $j_1(, j_2, j_3)$ are replaced by the corresponding rows in the A'-matrix displayed [$A' = PAP^{-1}$, where P is the matrix linking $\mathbf{x}$ to the top row vector $\mathbf{s}$ ($\mathbf{x} = P\mathbf{s}$)].

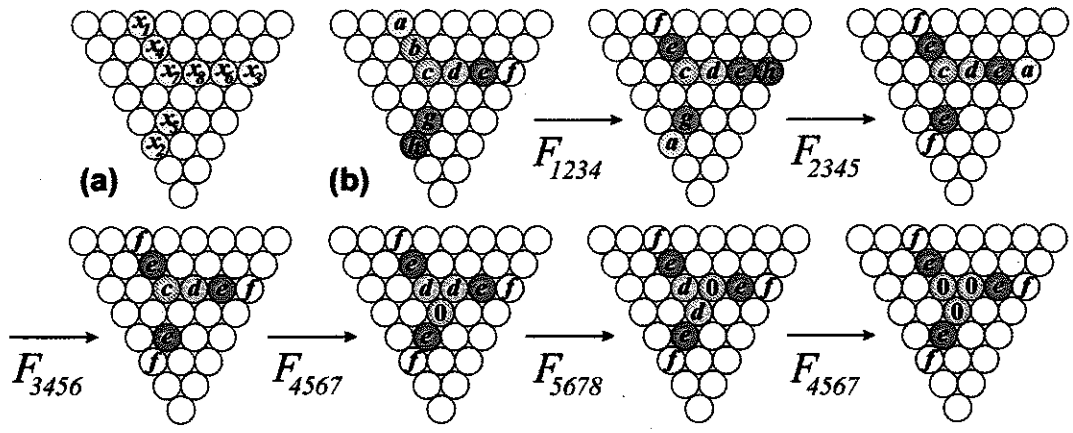


Fig. 19. Symbolic evolution under a universal path for 4-cell feedback for the generating set of Fig. 13(h). The components of the x-vector correspond to the cell-states of the generating set as shown under (a). (b) displays the symbolic evolution of the states in the generating set under a universal 4-cell feedback path (each symbol is also attributed a color).

An answer to the second question can be found in [Barbé & von Haeseler, 2002], and establishes a link with matroid theory.

3.2. *More general versions of the random m-cell feedback algorithm*

3.2.1. *Composite random multicell feedback*

The random m-cell feedback algorithm can be changed into a **random $(m_1, m_2, \ldots, m_l)$-cell feedback algorithm** (with $m_1 < m_2 < \cdots < m_l$, $m_i \in \{1, 2, \ldots, N-1\}$), by selecting at each recursion step the number of cells to be fed back in a random way among the set $\{m_1, m_2, \ldots, m_l\}$. When $(m_1, m_2, \ldots, m_l) = (1, 2, \ldots, N-1)$, we will speak about the **full-range random feedback procedure.**

This newly introduced procedure takes at each iteration a random step on a randomly selected state-transition graph among those transition graphs that correspond to the specified m_i-cell random feedback algorithms. Or, the state transition graph of this procedure is the union of all m_i-related transition graphs (i.e. union of all edges in the graphs, labels included), whereby at each iteration step only one of the possible paths leaving from the present state is taken in a random way. For example, the reduced state transition graph for the random (1, 3)-cell feedback algorithm for $N = 4$, with the generating sets displayed in Fig. 18, is the union of the corresponding transition graphs for $m = 1$ and $m = 3$.

It is clear that the validity of the previous conjectures also implies that the random $(m_1, m_2, \ldots, m_l)$-cell feedback algorithms satisfy the same property as expressed by these conjectures (the converse is not necessarily true). Figure 20 shows the average number of steps $n_{\text{av}}(N)$ for the full-range random feedback for sizes N up to 36 for the natural generating set. Observe also here

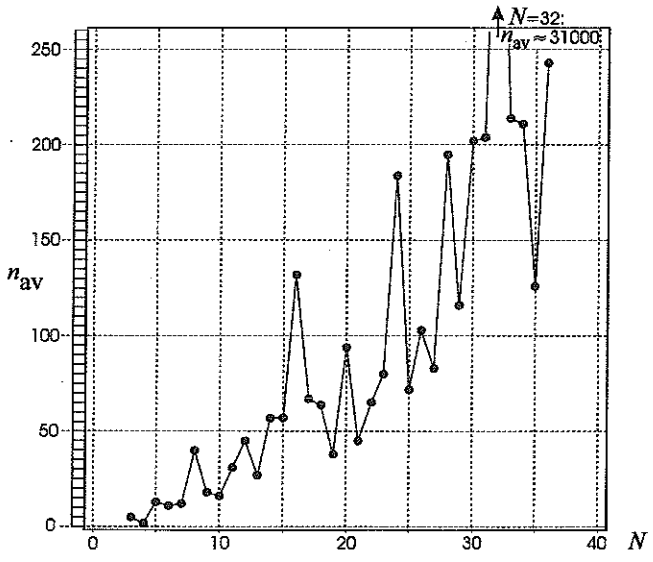


Fig. 20. Average number of steps for the full-range random feedback procedure to halt for $N = 3$ to 36 where at each iteration step the feedback number is taken with equal probability from $\{1, 2, \ldots, N-1\}$ (for $N = 3$ to 19: average taken over all initial states and over 10 runs per initial state; for $N = 20$ to 31: average taken over all initial conditions with code $1+k(2^{(N-12)}+7)$, $k = 0, 1, 2, \ldots$ and 10 runs per initial state); for $N = 32$ to 36: average taken over all states with code $1 + k(2^{20} + 7)$ and 5 runs per initial state, except for $N = 32$ where only 2 runs per initial state were taken.

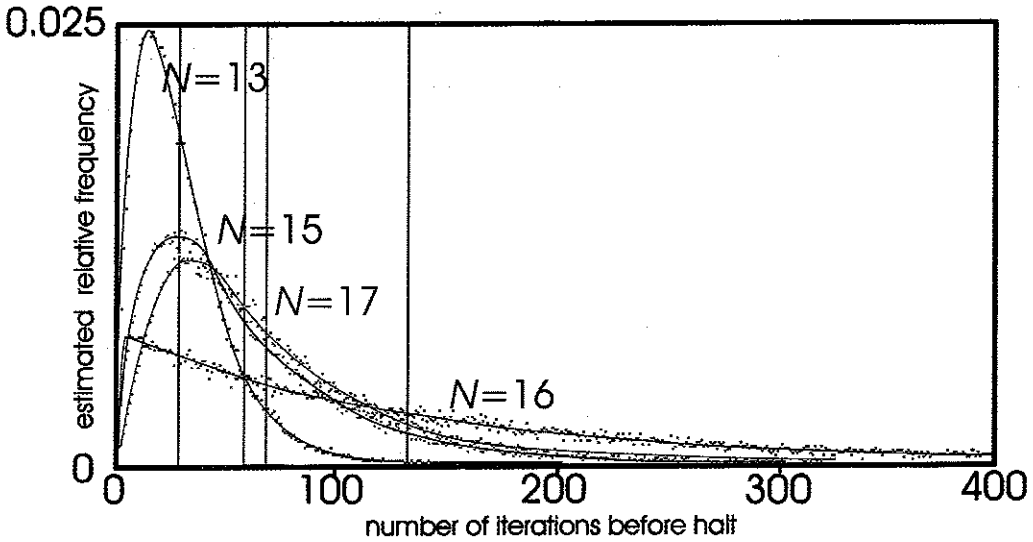


Fig. 21. The experimental relative frequency of halting times for the full-range random feedback procedure for problem sizes $N = 13, 15, 16, 17$ (10 runs for all initial states were taken into account).

that n_{av} is not monotonically increasing. Compare also with the global averages (horizontal lines) displayed in Fig. 13 which correspond to the average, over all m, of the average number of steps before halting for the random m-cell feedback algorithm. Both averages evolve in a similar way, but the ones in Fig. 13 exceed those in Fig. 20. Notice again the huge peaks at $N = 8, 16, 32$, with intermediate peaks at $N = 12, 20, 24, 28, 36 \ldots$.

Figure 21 shows the estimated probability functions of the number of iterations before halting for this full-range feedback procedure, for the sizes $N = 13$ to $N = 17$. Notice also here the more exponential nature for $N = 16$.

In an attempt to be able to say more about the evolution of n_{av} as N grows, experiments were done for N up to 232, but in which averages were taken over substantially less samples than was the case for obtaining the graphs in Fig. 20: 100 runs of the algorithm were performed in which the initial state was randomly selected for sizes $N \leq 63$, and only 10 runs for $N \geq 64$. Figure 22 displays a logarithmic plot of the obtained estimates. Although based on the few runs, this graph gives a good idea about the evolution of $n_{\text{av}}(N)$ and clearly hints at a kind of *scaling* in the evolution of n_{av}. For example, consider the N-interval $[28, 36]$. It contains a huge peak at $N = 32$. This behavior is somehow replicated on doubling scales, see the intervals $[56, 72]$ with a peak at $N = 64$, and $[112, 144]$ with a peak at $N = 128$, etc. The corresponding n_{av}-values thereby also jump to a next higher level. This scaling peak-behavior also occurs for neighborhoods at $N = 48, 96, 192, \ldots$, at $N = 56, 112, 224, \ldots$, etc. As a matter of fact, the behavior in the interval $[1, M]$ seems to be replicated also in $[1, 2M]$, at least qualitatively, by a proper scaling. This scaling is clearly nonlinear in n_{av}. A few words about the n_{av}-levels indicated by the arrows are at place here. Actually, we have not been able so far to get experimental values of n_{av} for the corresponding problem sizes, because of excessive computation time [for $N = 64$, we went as far as 38×10^6 steps without the algorithm coming to a halt]. For all other sizes showing an arrow, the full-range random feedback algorithm did not stop after one million steps, which was considered as an indication that the corresponding n_{av} is extremely high. In a sense, this is not so surprising: it was already noticed that for $N = 8, 16, 32$, there was a peak-behavior in $n_{\text{av}}(N)$. On the logarithmic plot, the evolution of these peak-heights seems quadratic, or even exponential. A rough quadratic extrapolation gives $\log_2 n_{\text{av}}(64) \approxeq 45$, even much higher for exponential extrapolation. This corresponds to an estimate for $n_{\text{av}} \approxeq 4 \times 10^{13}$.

A variant of the full-range random feedback procedure feeds back, on each iteration, each cell from G_R to G with a certain probability.

3.2.2. *Symmetric versions*

Instead of feeding back the y-values of the cells in the right-rotated version G_R of the generating

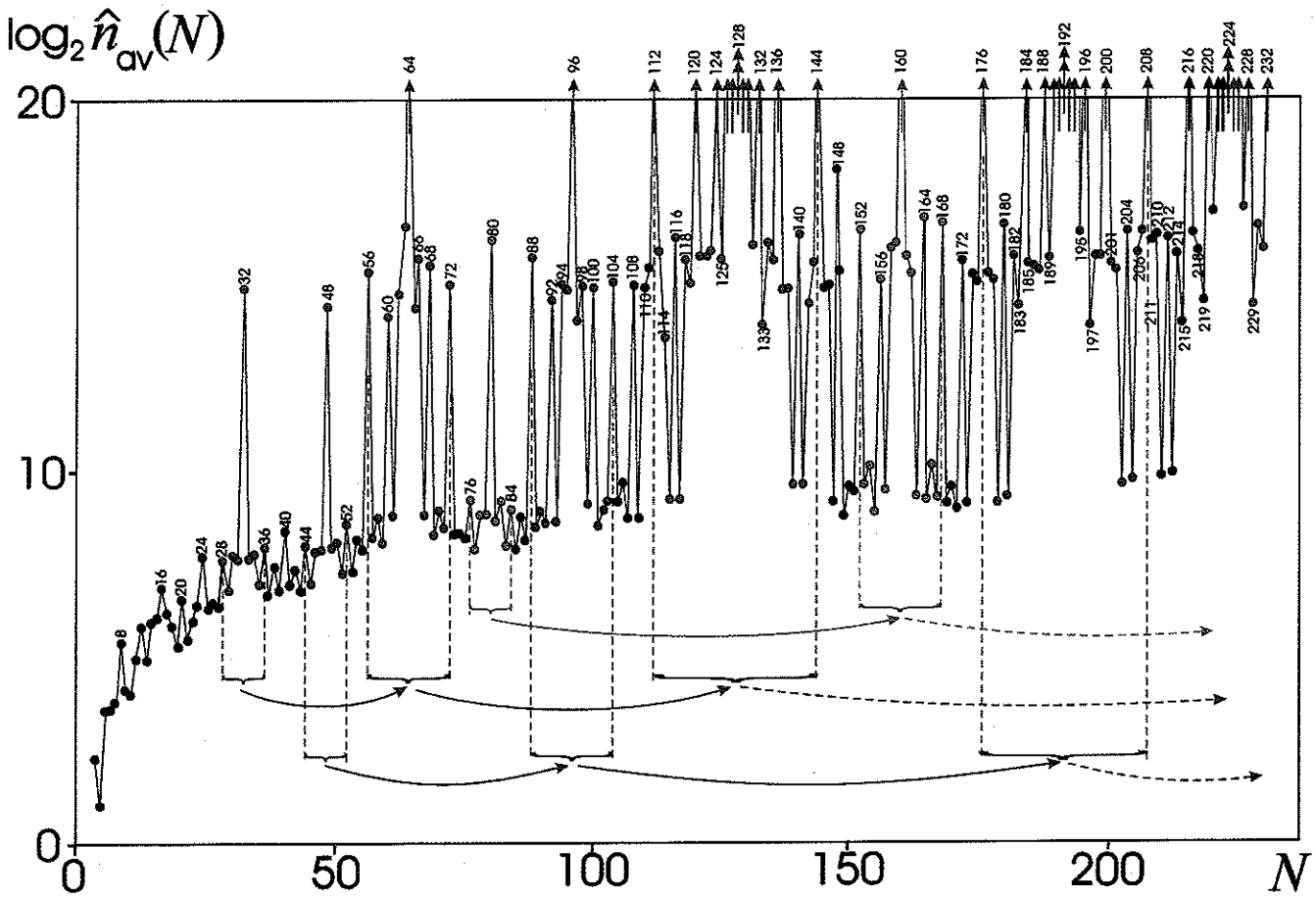


Fig. 22. Logarithmic plot of the estimates for the average number of iterations $n_{\mathrm{av}}(N)$ before halting for the full-range random feedback algorithm, based on 100 runs for $N \leq 63$ and 10 runs for $N \geq 64$. For sizes N for which an arrow is displayed, the procedure was aborted before halting when the number of iterations exceeded one million steps. A rough extrapolation suggests a level of magnitude of 10^{13} for n_{av} for these sizes corresponding to the single arrows. A still higher level occurs at the double arrows. Notice the scaling behavior when doubling sizes, as indicated by the colored intervals. As far as levels of magnitude are concerned, there is also some left–right reflecting symmetry in the neighborhood of the peaks.

set G to the corresponding $\mathbf{x}$-values in G, one could also consider a similar feedback from the left-rotated version G_L (with values $\mathbf{z}$) to G. Because of the symmetry of the local rule involved, this would make no fundamental difference. The only modification is that the Pascal-matrix A and the related F, A', F'-matrices have to be replaced by $\tilde{A} = A^2$ (which is the 180°-rotated version of A), and by the corresponding $\tilde{F}$, $\tilde{A}'$, $\tilde{F}'$, respectively. Thus, $\tilde{F}_{j_1,j_2,\ldots,j_m}$ is obtained by replacing rows $j_1, j_2, \ldots, j_m$ in the identity matrix by the corresponding rows of $\tilde{A}$; $\tilde{A}' = P\tilde{A}P^{-1}$, P again relating $\mathbf{x}$ to the top-row $\mathbf{s}$, i.e. $\mathbf{x} = P\mathbf{s}$; $\tilde{F}'_{j_1,j_2,\ldots,j_m}$ is obtained by replacing rows $j_1, j_2, \ldots, j_m$ in the identity matrix by the corresponding rows of $\tilde{A}'$. This **left-to-top feedback version** then iterates the maps $\mathbf{x} \mapsto \tilde{F}'_*\mathbf{x}$, where the $*$ refers in the usual way to the cells which are (randomly) selected in the feedback. On halting, this left feedback algorithm yields the solution for $(\mathbf{z} =)\tilde{A}'\mathbf{x} = \mathbf{x}$, which is also a necessary and sufficient condition for a rotationally symmetric $\bigtriangledown$-configuration (it is equivalent to $P\tilde{A}P^{-1}\mathbf{x} = \mathbf{x} \Leftrightarrow P^{-1}P\tilde{A}P^{-1}\mathbf{x} = P^{-1}\mathbf{x} \Leftrightarrow \tilde{A}\mathbf{s} = \mathbf{s} \Leftrightarrow A\mathbf{s} = \mathbf{s}$, $\mathbf{s}$ referring to the top-row vector in the $\bigtriangledown$-configuration). As left-feedback for a particular generating set has always a right-feedback counterpart, it is clear that if the previously stated conjectures (related to the right-feedback version of the algorithm) are true, then they are also true for the left-feedback version. Figure 23 displays the reduced state transition graphs for the left m-cell feedback for the same generating set as in Fig. 18. Observe that these graphs are isomorphic, in a non-trivial way, with the corresponding right-feedback ones in Fig. 18. For example, for $m = 2$, the isomorphism establishes under the interchange of the

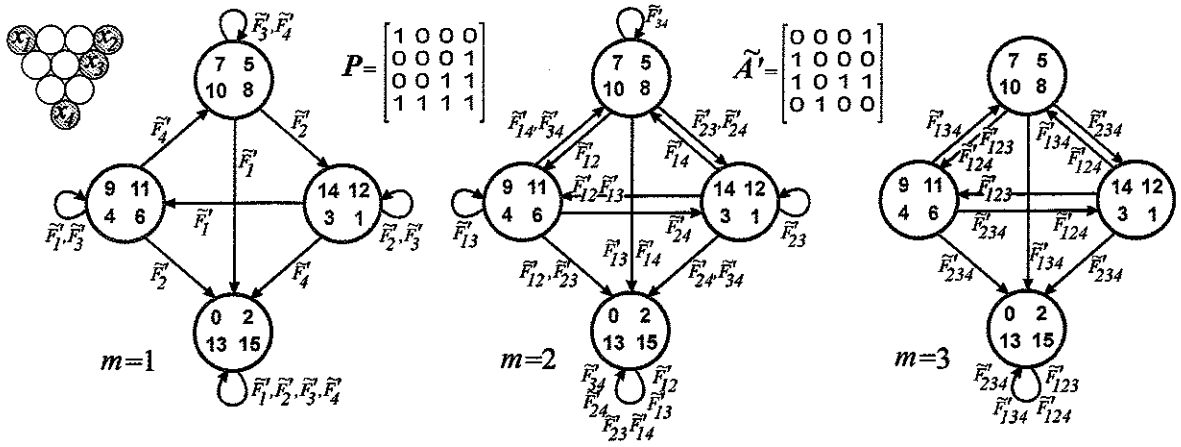


Fig. 23. The reduced state transition graphs for the left m-cell feedback algorithm for $N = 4$ and the same generating set as in Fig. 18(b). The transition graphs for the left–right symmetric version of the algorithm are the unions of the corresponding graphs above and in Fig. 18(b).

classes {5, 7, 8, 10} and {4, 6, 9, 11} and the replacements $F'_{12} \leftrightarrow \tilde{F}'_{12}$, $F'_{13} \leftrightarrow \tilde{F}'_{23}$, $F'_{23} \leftrightarrow \tilde{F}'_{13}$, $F'_{14} \leftrightarrow \tilde{F}'_{14}$, $F'_{24} \leftrightarrow \tilde{F}'_{14}$, $F'_{34} \leftrightarrow \tilde{F}'_{34}$. Is such an isomorphism between the left- and right-feedback transition graphs a general feature, and if so, what are the proper transformations of labels and states involved?

Similarly, by rotational symmetry, these conjectures would also apply for a proper feedback from the G_R- to the G_L-cells and vice versa (G_L is the right rotated version of G_R, so feedback from G_L to G_R does not at all differ from feedback from G_R to G).

A next version combines left- and right m-cell feedback, whereby at each iteration step, m randomly selected cells are fed back either from G_R to G or from G_L to G in a random fashion. This will be called the **left–right symmetric version** of the random m-cell feedback algorithm. Similar extensions can be conceived for the random $(m_1, m_2, \ldots, m_l)$-cell feedback procedure. Again, our conjectures stretch out to these extensions. The (reduced) transition graphs for the left–right symmetric versions of the algorithm are the unions of the corresponding graphs in Fig. 23 and Fig. 18(b).

A still further extension makes everything **fully symmetric**: it feeds back corresponding cells from G, G_R, G_L from the left or from the right in a random fashion.

3.2.3. *The multigenerating-set random feedback algorithm*

Here we present the ultimate version of the random feedback algorithm. It contains the previous versions as special cases. Let $\mathcal{G}_s$ be a specific collection of generating sets of a $\bigtriangledown$-triangle, then the procedure runs as follows:

Step 1. Start with an initial $\bigtriangledown$-configuration: $\bigtriangledown = \bigtriangledown^{(0)}$

Step 2. Select (randomly or not) an arbitrary generating set $G \in \mathcal{G}_s$, and attribute the state $\mathbf{x}$ to it which corresponds to $\bigtriangledown$ obtained in the previous step.

Step 3. Select (randomly or not) the left- or right-rotated version of the generating set G for implementing the feedback, i.e. select $G_* \in \{G_R, G_L\}$.
if $G_* = G_R$, propagate $\mathbf{x}$ to the corresponding $\mathbf{y}$ on G_R (for example, by calculating $\mathbf{y} = PAP^{-1}\mathbf{x}$, where P is the matrix linking $\mathbf{x}$ and the top-row vector $\mathbf{s}$ in $\bigtriangledown$, i.e. $\mathbf{x} = P\mathbf{s}$)
if $G_* = G_L$, propagate $\mathbf{x}$ to the corresponding $\mathbf{z}$ on G_L (for example, by calculating $\mathbf{z} = PA^2P^{-1}\mathbf{x}$).

Step 4. If $\mathbf{y} = \mathbf{x}$ when $G_* = G_R$ (or if $\mathbf{z} = \mathbf{x}$ when $G_* = G_L$), then the $\bigtriangledown$-configuration has rotational symmetry: HALT

Step 5. Select an arbitrary feedback number m (randomly or not) from $\{1, 2, \ldots, N-1\}$, and feedback m randomly chosen G_*-cells to the corresponding G-cells, i.e. select j_i, $i = 1, \ldots, m$, $j_i < j_{i+1}$ in a random way from $\{1, 2, \ldots, N-1\}$ and set $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ such that

$$x_k := x_k \quad \text{for } k \neq j_i,\ i = 1, \ldots, m$$

and

$$x_k := y_k \quad \text{for } k = j_i,\ i = 1, \ldots, m$$

when $G_* = G_R$,

$$x_k := z_k \quad \text{for } k = j_i,\ i = 1, \ldots, m$$

when $G_* = G_L$.

Step 6. Determine the new $\bigtriangledown$-configuration corresponding to this $(G, \mathbf{x})$-pair (for example, by calculating $\mathbf{s} = P^{-1}\mathbf{x}$ and applying the local matching rule), and go back to Step 2.

Then we formulate

Conjecture 4. Multigenerating-Set Conjecture. *The multigenerating-set random feedback algorithm halts with probability one for all problem sizes N, for all collections $\mathcal{G}_s$ of generating sets, and for all initial states of the underlying $\bigtriangledown_N$-configuration. Or, in terms of the corresponding state transition graphs: there is always a path from any state to an equilibrium state.*

As the related state transition graphs are proper unions of transition graphs for random m-cell feedback algorithms, this conjecture is implied by Conjecture 3. The opposite is not necessarily true. Also, if $\mathcal{G}_s$ contains one of the (almost) rotationally symmetric generating sets introduced earlier, the conjecture is true.

3.3. *Alternative symmetric local rules*

Everything presented so far is related to $\bigtriangledown$-configurations over $\mathbb{F}_2$ which satisfy the local matching rule (2)

$$c = \phi(a, b) = (a + b) \bmod 2\,.$$

However, there are other local matching rules for which the question of the existence of rotationally symmetric $\bigtriangledown$-configurations makes sense: namely all those for which the function $c = \phi(a, b)$ has a *quasigroup* (or latin-square) structure [Barbé & von Haeseler, 2001b]. We will consider here the example of a $\bigtriangledown$-configuration over $\mathbb{F}_3 = \{0, 1, 2\}$ subject to addition and multiplication modulo 3, and satisfying the local matching rule

$$c = \phi(a, b) = (2a + 2b) \bmod 3\,.$$

Figure 24 shows a few examples of rotationally symmetric solutions for this case.

The random feedback procedures as presented in the previous sections can also be applied to this case. They try to find fixed points of the mapping $\mathbf{s} \mapsto B\mathbf{s}$ in $\mathbb{F}_3$ ($\mathbf{s}$ being the top-row vector of the $\bigtriangledown$-configuration), or equivalently solutions of $B\mathbf{s} = \mathbf{s}$, where the matrix B is obtained from Pascal's triangle mod 3 in which the even numbered rows are

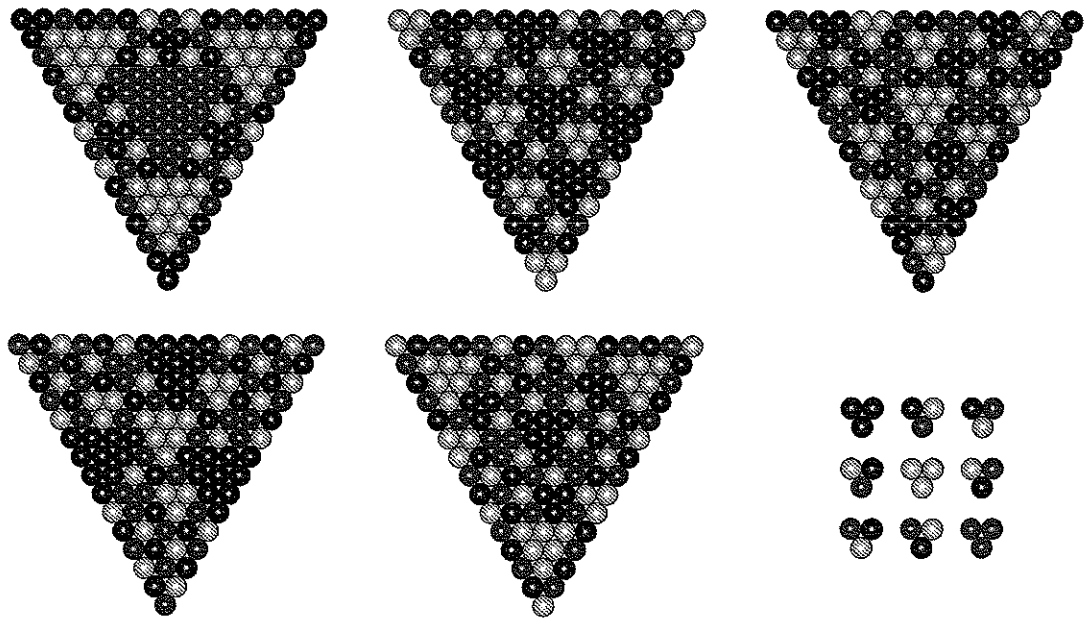


Fig. 24. A few rotationally symmetric $\bigtriangledown_{15}$-configurations in $\mathbb{F}_3$ satisfying the local rule $\phi(a, b) = (2a + 2b) \bmod 3$ which produces the local matchings displayed at the right bottom. The state values are: 0 = blue, 1 = green, 2 = red.

multiplied by 2 mod 3:

$$B = \begin{bmatrix} \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{2} & \mathbf{2} \\ \cdots & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \cdots & 0 & 0 & 0 & 0 & \mathbf{2} & 0 & 0 & \mathbf{2} \\ \cdots & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \cdots & 0 & 0 & \mathbf{2} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{1} & \mathbf{2} \\ \cdots & 0 & \mathbf{1} & 0 & 0 & \mathbf{2} & 0 & 0 & \mathbf{1} \\ \cdots & \mathbf{2} & \mathbf{2} & 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \\ & \vdots & & & & \vdots & & & \vdots \end{bmatrix} \tag{23}$$

But do the conjectures also apply to the new situation? The answer is definitely *no*! Experiments concerning the random m-cell feedback algorithm show that *for some problem sizes N and for some generating sets, the algorithm does not always stop, while for other N-values, it does.* For example, the random 1-cell feedback algorithm using the natural generating set does not always stop for

$$N = 5,\ 11,\ 13,\ 14,\ 17,\ 23,\ 29,\ 31,\ 32, \ldots \tag{24}$$

i.e. for some initial states, the algorithm never halts, for other initial states, it may or may not halt. This sequence will be called the **nonhalting sequence** for the particular algorithm and generating set under consideration. The first two numbers are certain, they were derived using an exhaustive search, the other numbers are just guesses: they are based on observation of very long iteration times without coming to a halt. Also, for sizes not belonging to this sequence, all the experiments performed came to a halt.

Moreover, the nonhalting sequence (24) for the 1-cell feedback algorithm using the natural generating set *seems also to be the nonhalting sequence for any feedback number $m(< N)$*. This would also imply that it is the nonhalting sequence for any composite random $(m_1, m_2, \ldots, m_l)$-cell feedback.

The fact that the m-cell feedback procedure does not halt for a given problem size means that, in the corresponding state transition graph, an absorbing set of states is reached which differs from the singleton absorbing states that correspond to a rotationally symmetric $\bigtriangledown$-configuration. The nonhalting sequence depends on the generating set: e.g. for the generating set of Fig. 15(a) the nonhalting sequence seems to be, for any legitimate feedback number, $N = 8,\ 11,\ 20,\ 23,\ 25,\ 26,\ 28, \ldots$; for the generating set of Fig. 15(d), this sequence seems to be the same as for the natural generating set displayed under (24). For the generating set of Fig. 15(e), the nonhalting sequence seems to be $N = 4,\ 6,\ 8,\ 10,\ 12,\ 14,\ 16, \ldots$ (also for any feedback number $m < N$). These observations lead to the following conjecture and question for the $\mathbb{F}_3$-local rule under consideration here

Conjecture 5. *Nonhalting sequences only depend on the generating set and are valid for any legitimate feedback number $m < N$.*

Question: What is the nonhalting sequence associated with a given generating set?

Experiments have shown that the m-cell algorithm with the natural generating set can be modified such that it halts for all N in the $\mathbb{F}_3$ case under study (at least this is conjectured), by introducing an additional random element in the feedback step: If a cell receiving a feedback is already in the same state as the cell issuing the feedback, the receiving cell should nevertheless change its state to any of the other possible states with some probability. This allows jumps in the transition graph which are forbidden by the $F_{j_1,j_2,\ldots,j_m}$-maps, but which may free the state from its imprisonment in a wrong absorbing set with some probability. Of course, once the rotational symmetry of the $\bigtriangledown$-configuration is detected, the computation stops.

A few experiments have been done for the totally symmetric local matching rule $\phi(a, b) = (4a + 4b) \bmod 5$ in $\mathbb{F}_5$. Also here, the existence of nonhalting sequences which are valid for all legitimate feedback numbers has been observed. Therefore we conjecture that the $\mathbb{F}_2$-case considered in the previous sections is the only one for which any of the proposed algorithms do halt for any problem size.

4. Conclusion

We have introduced several versions of a recursive procedure that produces solutions for $A\mathbf{s} = \mathbf{s}$, $\mathbf{s} \in \mathbb{F}_2^N$ where $A \in \mathbb{F}_2^{N \times N}$ is the Pascal-matrix defined in Eq. (3), the solutions being related to rotationally symmetric $\bigtriangledown$-configurations. The procedure must have an essentially random character. Several conjectures and open questions were discussed. The two main challenges are providing proofs for the following two conjectures.

1. For any problem size N, for any generating set, and for any feedback number m ($m \in \{1, 2, \ldots, N-1\}$, the (reduced) state transition graph for the m-cell feedback procedure always

has a path from any state to one of the attractor states which correspond to the solutions of $A\mathbf{s} = \mathbf{s}$.

2. (stronger conjecture, implying the first one) There is a universal path in the state transition graphs mentioned under 1.

Thus, the problem is not so much to show that these properties hold for a specific problem size (although the problem seems to have exponential complexity, as the number of vertices in the reduced state transition graph is $4^{\lceil (N-1)/3 \rceil}$), but to show that these properties hold for all possible problem sizes.

Moving outside the $\mathbb{F}_2$-framework raises the interesting question of determining the nonhalting sequence, i.e. the sequence of problem sizes for which the m-cell feedback algorithm does not always halt.

Acknowledgments

This work was partially supported by The Concerted Action project GOA-Mefisto of the Flemish Community, by the Belgian Program on Interuniversity Attraction Poles of the Belgian Prime Minister's Office for Science, Technology and Culture (IUAP P4-02), by the FWO (Fund for Scientific Research Flanders) project G.0080.01, and by the Research Council of the K.U. Leuven.

References

Allouche J.-P., von Haeseler, F., Peitgen, H.-O., Skordev, G. [1996] "Linear cellular automata and Pascal's triangle," *Discr. Appl. Math.* **66**, 1–22.

Barbé, A. [1988] "Periodic patterns in the binary difference field," *Compl. Syst.* **2**, 209–233.

Barbé, A., von Haeseler, F., Peitgen, H.-O., Skordev, G. [1995] "Coarse-graining invariant patterns of one-dimensional two-state linear cellular automata," *Int. J. Bifurcation and Chaos* **5**, 1611–1631.

Barbé, A. [1997] "Complex order from disorder an from simple order in coarse-graining invariant orbits of certain two-dimensional linear cellular automata," *Int. J. Bifurcation and Chaos* **7**, 1451–1496.

Barbé, A. [2000] "Symmetric patterns in the cellular automaton that generates Pascal's triangle modulo 2," *Discr. Appl. Math.* **105**, 1–38.

Barbé, A. & von Haeseler, F. [2001a] "Periodic patterns in orbits of certain linear cellular automata," *Proc. Conf. Discrete Models: Combinatorics, Computation, and Geometry*, eds., Cori, R., Mazoyer, J., Morvan, M. & Mosseri, R. (Maison de l'Informathique et des Mathématiques Discrètes, Paris), pp. 79–93.

Barbé, A. & von Haeseler, F. [2001b] "Cellular automata, quasigroups and symmetry," *Aequationes Math.* **62**, 211–248.

Barbé, A. & von Haeseler, F. [2002] "The Pascal matroid as a home for generating sets of configurations defined by quasigroups," Sista-Cosic Report nr.02-09, Dept. of Electr. Eng., KU Leuven, Belgium.

Bertsekas, D. P. & Tsitsiklis, J. N. [1989] *Parallel and Distributed Computation* (Prentice-Hall, London).

Chabert, J.-L. [1999] *A History of Algorithms* (Springer-Verlag, Berlin).

Gutowitz, H. (ed.) [1991] *Cellular Automata: Theory and Experiment, A Bradford Book* (MIT Press, Cambridge, MA); [1990] reprinted from *Physica* **D45**.

Ilachinsky, A. [2001] *Cellular Automata, A Discrete Universe* (World Scientific, Singapore).

Korec, I. [1993] "Irrational speeds of configurations growth in generalized Pascal's triangles," *Theoret. Comput. Sci.* **112**, 399–412.

Robert, F. [1995] *Les Systèmes Dynamiques Discrets* (Springer-Verlag, Berlin).

Thulasiraman, K. & Swamy, M. N. S. [1992] *Graphs: Theory and Algorithms* (John Wiley, NY).

von Haeseler, F., Peitgen, H.-O. & Skordev, G. [1992] "Pascal triangle, dynamical systems and attractors," *Ergod. Th. Dyn. Syst.* **12**, 479–489.

von Haeseler, F., Peitgen, H.-O. & Skordev, G. [2001] "Self-similar structure of rescaled evolution sets of cellular automata I, II," *Int. J. Bifurcation and Chaos* **11**, 913–941.

Wolfram, S. [1994] *Cellular Automata and Complexity* (Addison-Wesley, Reading, MA).