Information Sciences 382–383 (2017) 81–95

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Welch sets for random generation and representation of reversible one-dimensional cellular automata☆

Juan Carlos Seck-Tuoh-Mora[a,*], Joselito Medina-Marin[a], Norberto Hernandez-Romero[a], Genaro J. Martinez[b,c], Irving Barragan-Vite[a]

[a] AAI-ICBI-UAEH. Carr Pachuca-Tulancingo Km 4.5. Pachuca 42184 Hidalgo, Mexico
[b] Unconventional Computing Centre, University of the West of England, BS16 1QY Bristol, United Kingdom
[c] Escuela Superior de Computo, Instituto Politecnico Nacional, Mexico

ARTICLE INFO

Article history:
Received 28 June 2016
Revised 7 November 2016
Accepted 4 December 2016
Available online 7 December 2016

Keywords:
Cellular automata
Reversibility
Welch indices
Bipartite graph
Block mapping

ABSTRACT

Reversible one-dimensional cellular automata are studied from the perspective of Welch Sets. This paper presents an algorithm to generate random Welch sets that define a reversible cellular automaton. Then, properties of Welch sets are used in order to establish two bipartite graphs describing the evolution rule of reversible cellular automata. The first graph gives an alternative representation for the dynamics of these systems as block mappings and shifts. The second graph offers a compact representation for the evolution rule of reversible cellular automata. Both graphs and their matrix representations are illustrated by the generation of random reversible cellular automata with 6 and 18 states.

1. Introduction

Cellular automata are discrete dynamical systems able to generate complex behavior based on simple local interaction among their components. Reversible cellular automata are characterized to produce an invertible global behavior provoked by a set of local mappings which are not reversible.

In the one-dimensional case, reversible cellular automata have been widely investigated in order to understand their topological, combinatorial and dynamical properties.

Due to their property of keeping the original information of the system, concepts and alternative definitions of reversible cellular automata have been employed in the specification of different cyphering systems [2,8,9,11,19,20,22], and error-correcting codes [13].

Recent works have presented different procedures to characterize reversible behavior in cellular automata. For instance: the generalization of reversibility defined by linear rules over the binary field case under null boundary conditions is explained in [21]. Reversibility of elementary cellular automata rule number 150 is described by circulant matrices in [14]. Reversibility in cellular automata with memory has been studied in [1] and [17].

☆ Dedicated to the memory of Harold V. McIntosh (1929–2015), a leader in the development of cellular automata theory, whose teaching and knowledge were essential to this work.

* Corresponding author:
E-mail addresses: jseck@uaeh.edu.mx (J.C. Seck-Tuoh-Mora), jmedina@uaeh.edu.mx (J. Medina-Marin), nhromero@uaeh.edu.mx (N. Hernandez-Romero), genaro.martinez@uwe.ac.uk (G.J. Martinez), ibarragan@uaeh.edu.mx (I. Barragan-Vite).

http://dx.doi.org/10.1016/j.ins.2016.12.009
0020-0255/

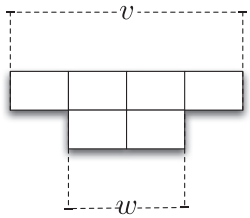

Fig. 1. Evolution of sequence v with four cells for $r = 1$, in this case $\varphi(v)$ has two cells..

Thus, there is a continuous investigation to characterize the properties of reversible cellular automata, both for theoretical research and practical application.

The first topological study of reversible cellular automata is developed by Hedlund in [10] establishing important concepts for the one-dimensional case such as the uniform multiplicity of ancestors and Welch sets. Several investigations followed this paper using Welch sets to define new graphs representing the local behavior inducing global reversibility [15], and using their vector representation to give a maximum bound for the minimum information needed to obtain global invertibility [7]. Another topic of interest has been to enumerate all the possible cases of reversible one-dimensional cellular automata [3,4,18], the last two references taking reversible automata as groupoids with algebraic properties. In this sense, references [6] and [5] establish the properties of these groupoids and their relation with isotopy and Cayley graphs.

Based on this previous work, the aim of this paper is to present an algorithm to create random Welch sets to define reversible one-dimensional cellular automata. From these Welch sets, we define two bipartite graphs, both for representing the forward and reverse local mappings of the associated reversible cellular automaton. We give examples showing that the former graph provides an alternative representation for the dynamics of reversible automata by block mappings and shifts firstly demonstrated by Kari in [12], and the latter needs less space than the original evolution rule.

The paper is organized as follows: Section 2 explains the basic concepts of reversible one-dimensional cellular automata and Welch sets. Section 3 defines the properties of Welch Sets for reversible automata with neighborhood size 2. Section 4 presents the algorithm generating random Welch sets and arbitrary reversible cellular automata. Section 5 exposes the construction of a bipartite graph based on Welch elements to calculate the evolution of reversible cellular automata and represent their dynamics by block mappings and shifts. Section 6 specifies a shorter version of this graph in order to represent the evolution of reversible automata with a reduced number of elements. The last section gives the final remarks of the study.

2. Basic concepts of Welch sets

One-dimensional cellular automata are defined by a discrete set of states S, a neighborhood radius $r \in \mathbb{N}$, an initial configuration $c_0 : \{1, 2, \ldots, m\} \to S$ for some $m \in \mathbb{N}$, and an evolution rule $\varphi : S^{2r+1} \to S$. The dynamics is given applying the evolution rule over every cell in c_0, thus $c_1(i) = \varphi(c_0(i-r)\ldots c_0(i+r))$ taking periodic boundary conditions. In general $c_{t+1}(i) = \varphi(c_t(i-r)\ldots c_t(i+r))$.

For simplicity we shall represent $\varphi(c(i-r)\ldots c(i+r))$ just as $\varphi(c(i))$. Thus, for a string $v \in S^m$ with $m \geq 2r+1$, we define $\varphi(v)$ as follows:

$$\begin{aligned}\varphi(v) &= \varphi(v(r+1))\varphi(v(r+2))\ldots\varphi(v(m-r)) \\ &= w(1)w(2)\ldots w(m-2r) = w\end{aligned} \tag{1}$$

where w is the evolution of v. Eq. (1) shows that $\varphi(v)$ has $2r$ cells less than v. Fig. 1 illustrates the case for a sequence of four cells and $r = 1$; in this work we are using centered evolutions.

In order to simplify our study, we take the simulation of every cellular automaton by another with neighborhood size 2. By Eq. (1), for $v = c_t(i-2r+1)\ldots c_t(i+2r)$ we have that $\varphi(v) = w = c_{t+1}(i-r+1)\ldots c_{t+1}(i+r)$. Then $|v| = 4r$ and $|w| = 2r$. Let us define a new set of states S' such that $|S'| = |S|^{2r}$; thus, we can define a bijection from S' to S^{2r}. With this, there are bijections $v \to s_1 s_2$ and $w \to s_3$ for $s_i \in S'$ such that:

$$\varphi'(s_1 s_2) = s_3 \tag{2}$$

Eq. (2) shows that the original evolution of a cellular automaton can be simulated by another evolution rule φ' with neighborhood size 2 (or radius 1/2) and an extended set of states. Fig. 2 describes this simulation for a cellular automaton with $r = 1$; where t indicates time step.

A cellular automaton is reversible if there exists an integer $r' \in \mathbb{N}$ and an inverse mapping $\varphi^{-1} : S^{2r'+1} \to S$ such that $c_t(i) = \varphi^{-1}(c_{t+1}(i))$. This implies that $\varphi^{-1}(\varphi(c_t(i))) = c_t(i)$.

A reversible cellular automaton can be simulated by another with $r = 1/2$ for both invertible rules, taking $r^\star = \max(r, r')$ and using an extended set of states $S^\star$ with $|S^\star| = |S|^{2r^\star}$. Therefore, we have to study only reversible cellular automata with neighborhood size 2 in both invertible rules in order to understand the other cases.

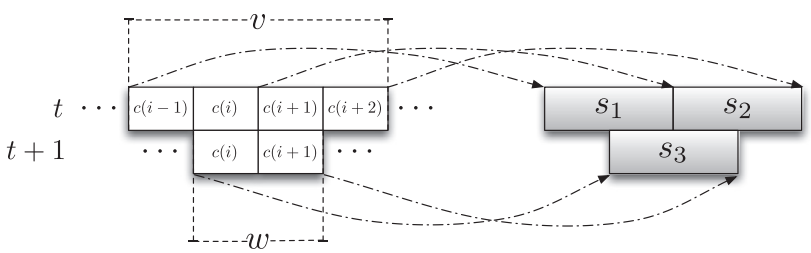

Fig. 2. Evolution of a cellular automaton with $r = 1$ represented by another with $r = 1/2$ according to Eq. (2).

The main properties of reversible cellular automata were firstly defined in the seminal works of Hedlund [10] and Nasu [15]. A detailed explanation of how these properties generate reversible behavior for reversible automata with neighborhood size 2 can be consulted in [18].

The features of reversible one-dimensional cellular automata are summarised in three main properties:

1. *Uniform multiplicity of ancestors. For every string* $w \in S^n$ *there is a set of strings* $A \subset S^{n+1}$ *such that* $|A| = |S|$ *and each* $v \in A$ *holds that* $\varphi(v) = w$. *Each* v *is an ancestor of* w.
2. *Welch indices. There exist* $o_L, o_R \in \mathbb{N}$ *such that for every state* $s \in S$ *there are sets* L_s, $R_s \subset S$ *with* $|L_s| = o_L$ *and* $|R_s| = o_R$ *such that* $\varphi(ab) = s$ *for each* $a \in L_s$ *and every* $b \in R_s$, *and the product* $o_L o_R = |S|$. *These* o_L, o_R *are called*Welch indices.
3. *Unique intersection. Every pair of states* $a, b \in S$ *holds that* $|R_a \cap L_b| = |R_b \cap L_a| = 1$.

Properties 2 and 3 define that the ancestor strings of any sequence of 2 or more states share a common central part and their differences are at the ends. These properties assure the existence of an inverse evolution rule able to revert the global dynamics of the automaton.

The sets L_s and R_s are the Welch sets associated with state $s \in S$. Notice that indices o_L and o_R are independent of the state defining L_s and R_s. That is, $|L_s| = o_L$ and $|R_s| = o_R$ for all $s \in S$.

Since φ and φ^{-1} are defined with a neighborhood size 2, they can be represented by matrices M_φ and $M_{\varphi^{-1}}$ such that entries $M_\varphi(a, b) = \varphi(ab)$ and $M_{\varphi^{-1}}(a, b) = \varphi^{-1}(ab)$ for every pair of states $a, b \in S$.

3. Properties of Welch sets

The previous properties give a particular structure for M_φ and $M_{\varphi^{-1}}$. Property 1 guarantees that there are $|S|$ entries equal to s in both matrices for every $s \in S$. Property 2 establishes that if $M(s_1, s_2) = M(s_3, s_4)$ then $M(s_1, s_4) = M(s_3, s_2)$ for $s_i \in S$. Finally, Property 3 specifies that for every state $s \in S$ there is a unique state $a \in S$ such that $M_\varphi(a, a) = s$. Thus, in M_φ and $M_{\varphi^{-1}}$ every state has a rectangular arrangement, and both matrices define a rectangular groupoid [5].

For every reversible cellular automaton, we can define a state permutation such that $M_\varphi(s, s) = s$. Due to $\varphi(ss) = s$, it is hold that $\varphi(sss) = ss$. Therefore, $\varphi^{-1}(ss) = s$ and the same state permutation implies that $M_{\varphi^{-1}}(s, s) = s$. This feature simplifies even more our study, we just need to analyze reversible cellular automata with neighborhood size 2 and diagonal elements $M_\varphi(s, s) = M_{\varphi^{-1}}(s, s) = s$ for every $s \in S$ because all the other cases can be represented by this one. Since $|L_s| = o_L$ and $|R_s| = o_R$, state s appears in o_L rows of M_φ, at the same o_R columns.

For $a \in S$, let $M_\varphi(a, *)$ and $M_\varphi(*, a)$ be the a-th row and the a-th column respectively of matrix M_φ. Let $D_\varphi(a)$ be the set of states in row $M_\varphi(a, *)$ and let $I_\varphi(a)$ be the set of states in column $M_\varphi(*, a)$. With this, the following remark can be proved.

Remark 1. $|D_\varphi(a)| = o_L$ for each $a \in S$.

Proof. Since $M_\varphi(a, *)$ has $|S|$ columns and every state s holds that $|R_s| = o_R$, then there are $|S|/o_R = o_L$ states in $M_\varphi(a, *)$. □

A straightforward result from Remark 1 is the following one.

Corollary 1. $|I_\varphi(a)| = o_R$ *for each* $a \in S$.

A relevant property of $D_\varphi(a)$ and $I_\varphi(a)$ is the next one.

Remark 2. $D_\varphi(a) \cap I_\varphi(a) = a$ for every $a \in S$.

Proof. Since $M_\varphi(a, a) = a$ we have that $a \in D_\varphi(a)$ and $a \in I_\varphi(a)$. If there is another state $b \neq a$ such that $b \in D_\varphi(a)$ and $b \in I_\varphi(a)$, then $a \in L_b$ and $a \in R_b$. This implies that $\varphi(aa) = b$ and therefore $M_\varphi(a, a) = b$, which is a contradiction. □

Remark 2 says that $D_\varphi(a)$ and $I_\varphi(a)$ agree only in the diagonal state. For each $s \in S$, let L_s^{-1} and R_s^{-1} be the Welch sets of s associated with the inverse evolution rule φ^{-1}. Accordingly, let $o_{L^{-1}}$ and $o_{R^{-1}}$ be the Welch indices related to φ^{-1}. The following remark establishes the relationship between the Welch indices of φ and φ^{-1}.

Remark 3. $o_{L^{-1}} = o_R$ and $o_{R^{-1}} = o_L$.

Proof. For every state $s \in S$, we have that $a \in L_i^{-1}$ and $b \in R_i^{-1}$ iff $\varphi^{-1}(ab) = s$. This implies that $a \in I_\varphi(s)$ and $b \in D_\varphi(s)$, hence $L_s^{-1} = I_\varphi(s)$ and $R_s^{-1} = D_\varphi(s)$. Remark 1 and Corollary 1 indicate that $|I_\varphi(s)| = o_R$ and $|D_\varphi(s)| = o_L$. Therefore $o_{L^{-1}} = o_R$ and $o_{R^{-1}} = o_L$. □

4. Algorithm to generate random Welch sets

For $U \subseteq S$, we can represent the elements of U by a binary vector g_U: $S \to 0, 1$ of size $1 \times |S|$ such that $g_U(a) = 1$ iff $a \in U$; otherwise, $g_U(a) = 0$. With this, we define two mappings $\alpha(g_U) = U$ and $\beta(U) = g_U$.

The next algorithm computes valid Welch sets for every state $s \in S$ to fill up the matrix M_φ representing the evolution rule of a reversible cellular automaton. The algorithm applies the properties of Welch sets as vector and matrix operations in order to facilitate their computational implementation. The vectors and matrices used in the algorithm are defined as follows:

- G_L: $|S| \times |S|$ matrix representing left Welch vectors as row binary vectors.
- G_R: $|S| \times |S|$ matrix representing right Welch vectors as row binary vectors.
- G_D: $|S| \times |S|$ matrix where every binary row represents the valid states that can be placed at the same row in M_φ.
- G_I: $|S| \times |S|$ matrix where every binary row represents the valid states that can be placed at the corresponding column in M_φ.
- h_D: $1 \times |S|$ integer vector representing the number of states used in each row of M_φ.
- h_I: $1 \times |S|$ integer vector representing the number of states used in each column of M_φ.
- E: $|S| \times |S|$ matrix representing the sum of Kronecker products to validate that the Welch vectors calculated so far compose a valid evolution rule.

Let us represent the identity matrix as $\mathbf{I}$, the zero matrix as $\mathbf{0}$, and the row vector of ones as $\overline{\mathbf{1}}$. For a binary matrix B, let us define its negation by $\neg B$ and its transpose by B^T. With these elements, the algorithm is defined as follows:

1. Initialize $limit_l, limit_r \in \mathbb{N}$ (loop variables).
2. Initialize $s = 1$, $G_L = G_R = \mathbf{I}$, $G_D = G_I = \neg\mathbf{I}$, $h_D = h_I = \overline{\mathbf{1}}$, and $M_\varphi = E = \mathbf{0}$. This step specifies that at the beginning each state $s \in S$ belongs to L_s and R_s. Equally, the initialization determines that s belongs to D_s and I_s and the rest of states are available to fill up each row $M_\varphi(s, *)$ and every column $M_\varphi(*, s)$ respectively.
3. Set $threshold_l = 0$ and $flag_l = 1$ (control variables).
4. Calculate a candidate left Welch set L_s taking state s and $o_L - 1$ random states from $\{\alpha(h_D < o_L) \cap \alpha(G_D(s, *) > 0)\}$. Set $G_L(s, *) = \beta(L_s)$. Validate Property 3 for candidate L_s: for $1 \leq i < s$, if any dot product $G_R(i, *) \cdot G_L(s, *) \neq 1$ then $flag_l = 0$.
5. If $flag_l = 1$, restrict the possible states for candidate R_s. By Remark 2, for every $a \in L_s$, $a \neq s$, set $G_I(s, a) = 0$. For $1 \leq i < s$, take every i such that $|\alpha(G_L(i, *)) \cap \alpha(G_L(s, *))| > 0$. Then the states in the corresponding $\alpha(G_R(i, *))$ cannot be part of R_s to avoid that the same neighborhood has multiple evolutions. Therefore, set $G_I(s, a) = 0$ for every $a \in \alpha(G_R(i, *))$. Enumerate the set of possible states for the candidate right Welch set R_s: if $|\{\alpha(h_I < o_R) \cap \alpha(G_I(s, *) > 0)\}| < o_R - 1$ then $flag_l = 0$.
6. If $flag_l = 0$, set $threshold_l = threshold_l + 1$. If $threshold_l < limit_l$ then set $flag_l = 1$ and return to step 4, otherwise return to step 2.
7. Set $threshold_r = 0$ and $flag_r = 1$ (control variables).
8. Calculate a candidate right Welch set R_s taking state s and $o_R - 1$ random states from $\{\alpha(h_I < o_R) \cap \alpha(G_I(s, *) > 0)\}$. Set $G_R(s, *) = \beta(R_s)$. Validate Property 3 for candidate R_s: for $1 \leq i \leq s$, if any dot product $G_R(s, *) \cdot G_L(i, *) \neq 1$ then $flag_r = 0$.
9. If $flag_r = 1$, calculate the Kronecker product $K_s = G_L(s, *)^T \otimes G_R(s, *)$ and take $K' = K_s + E$. If $K'(i, j) > 1$ for $1 \leq i \leq j \leq S$ then $flag_r = 0$. This means that there is a neighborhood with multiple evolutions.
10. If $flag_r = 0$ set $threshold_r = threshold_r + 1$. If $threshold_r < limit_r$ then set $flag_r = 1$ and return to step 8, otherwise go to step 4.
11. Valid Welch sets have been generated for state s. Update $h_D(a) = h_D(a) + 1$ for each $a \in \{\alpha(G_L(s, *)) - s\}$, and analogously $h_I(b) = h_I(b) + 1$ for each $b \in \{\alpha(G_R(s, *)) - s\}$. Set $E = E + K_s$ and $M_\varphi(a, b) = s$. If $s < |S|$ then $s = s + 1$ and go to step 3. Otherwise return matrices G_L, G_R and M_φ.

The associated pseudocode is presented in Algorithm 1.

In Algorithm 1, loop parameters $limit_l$ and $limit_r$ define the number of iterations to search valid Welch sets for every state s. In case that a valid R_s has not been calculated after $limit_r$ attempts, a new L_s is calculated to restart the search of an appropriate R_s. If a valid L_s has not been calculated after $limit_l$ attempts, a valid R_s has not been calculated after $limit_l * limit_r$ iterations. This implies that the previous calculated Welch sets could not be so adequate to specify a reversible automaton, especially for large values of $limit_l$ and $limit_r$. Therefore, the search restarts from $s = 1$. Notice that parameters $limit_l$ and $limit_r$ are not directly related to the number of cells in the configuration of the reversible automaton.

The aim of the algorithm proposed in this paper is not to enumerate reversible automata but only to generate a random instance for a larger number of states. In fact, the algorithm has been able to calculate random Welch sets for reversible automata up to 30 states, with any combination of Welch indices.

Algorithm 1 Generate reversible automata by random Welch sets.

Require: Number of states $|S|$, $s = 1$, and loop parameters $limit_l, limit_r \in \mathbb{N}$;

1: **while** $s <= |S|$ **do**
2: **if** s=1 **then** ▷ Initialize data structures
3: $G_L = G_R = \mathbf{I}$, $G_D = G_I = \neg\mathbf{I}$, $h_D = h_I = \overline{\mathbf{1}}$, and $M_\varphi = E = \mathbf{0}$;
4: **end if**
5: $threshold_l = 0$; ▷ Set loop variable to find a valid L_s
6: **while** $threshold_l <= limit_l$ **do**
7: $L_s \leftarrow o_L - 1$ random states in $\{\alpha(h_D < o_L) \cap \alpha(G_D(s,*) > 0)\}$; ▷ Validate Remark 1
8: $G_L(s,*) = \beta(L_s)$; $flag_l = 1$; ▷ Validate Property 3 for L_s
9: **if** $G_R(i,*) \cdot G_L(s,*) \neq 1$ for $1 \leq i < s$ **then** $flag_l = 0$;
10: **end if**
11: **if** $flag_l = 1$ **then**
12: $G_I(s,a) = 0$ for every $a \in L_s$, $a \neq s$; ▷ Restrict states for R_s to hold Remark 2
13: **for all** $i \in \{1 \ldots s-1\}$ such that $|\alpha(G_L(i,*)) \cap \alpha(G_L(s,*))| > 0$ **do**
14: $G_I(s,a) = 0$ for $a \in \alpha(G_R(i,*))$; ▷ To avoid neighborhoods with multiple evolutions
15: **end for**
16: **if** $|\{\alpha(h_I < o_R) \cap \alpha(G_I(s,*) > 0)\}| < o_R - 1$ **then** $flag_l = 0$
17: **else** ▷ Enough available states for R_s
18: $threshold_r = 0$; ▷ Set loop variable to find a valid R_s
19: **while** $threshold_r <= limit_r$ **do**
20: $R_s \leftarrow o_R - 1$ random states in $\{\alpha(h_I < o_R) \cap \alpha(G_I(s,*) > 0)\}$; ▷ Validate Cor. 1
21: $G_R(s,*) = \beta(R_s)$; $flag_r = 1$; ▷ Validate Property 3 for R_s
22: **if** $G_R(s,*) \cdot G_L(i,*) \neq 1$ for $1 \leq i <= s$ **then** $flag_r = 0$;
23: **end if**
24: **if** $flag_r = 1$ **then** ▷ Kronecker product to check neighborhoods already taken
25: $K_s = G_L(s,*)^T \otimes G_R(s,*)$; $K' = K_s + E$;
26: **if** $K'(i,j) > 1$ for $1 \leq i \leq j \leq S$ **then** $flag_r = 0$;
27: **else** ▷ Valid Welch sets have been generated for state s
28: $h_D(a) = h_D(a) + 1$ for $a \in \{\alpha(G_L(s,*)) - s\}$; ▷ Update states used in rows of M_φ
29: $h_I(b) = h_I(b) + 1$ for $b \in \{\alpha(G_R(s,*)) - s\}$; ▷ Update states used in columns of M_φ
30: $E = E + K_s$; ▷ Update sum of Kronecker products
31: **for all** $a \in \{\alpha(G_L(s,*))\}$ and $b \in \{\alpha(G_R(s,*))\}$ **do** ▷ Update evolution rule
32: $M_\varphi(a,b) = s$
33: **end for**
34: **if** $s < |S|$ **then** ▷ There are more states remaining
35: $s = s + 1$; ▷ Advance to the next state
36: $threshold_r = limit_r + 1$; $threshold_l = limit_l + 1$; ▷ Break while loops
37: **else return** G_L, G_R and M_φ; ▷ Reversible rule is complete
38: **end if**
39: **end if**
40: **end if**
41: **if** $flag_r = 0$ **then** $threshold_r = threshold_r + 1$; ▷ Incorrect R_s, search again
42: **end if**
43: **end while**
44: **end if**
45: **end if**
46: **if** $flag_l = 0$ **then** $threshold_l = threshold_l + 1$; ▷ Incorrect L_s, search again
47: **if** $threshold_l > limit_l$ **then** $s = 1$; ▷ Previous Welch sets are inadequate, start again
48: **end if**
49: **end if**
50: **end while**
51: **end while**

This algorithm can be classified as a constrained random local search [16], in the same category as random search heuristics or evolutionary algorithms. Random local search are applied to problems whose structure is not well-understood, as well as to hard combinatorial problems, like the enumeration of reversible automata.

This enumeration is still an open problem. In fact, a set of formulas or a general algorithm to enumerate all reversible automata with $r = 1/2$ and any number of states have not been defined yet. The maximum number of states reported for any procedure so far is 12 [3,18].

Properties of reversible automata are implicit in the generation of the Welch vectors $G_L(s,*)$ and $G_R(s,*)$ in lines 8 and 21 for each $s \in S$. We have that the Welch sets $|\alpha(G_L(s,*))| = o_L$ due to lines 3 and 7, and $|\alpha(G_R(s,*))| = o_R$ due to lines 3 and 20, holding Property 2 (Welch indices). Property 1 (Uniform multiplicity of ancestors) is fulfilled in lines 31–33 where $|S|$ ancestors are specified for every s. Lines 9 and 22 validate Property 3 for all the products between left and right Welch sets. Thus, the obtained matrix M_φ satisfies Properties 1, 2 and 3 and therefore generates a reversible automaton.

One important point in the algorithm is the set of restrictions implemented to validate and reduce the search space. In particular, in the algorithm, Remark 1 in line 7, Remark 2 in line 12, discard multiple evolutions for the same neighborhood in line 14, enough available states for R_s in line 16 and Corollary 1 in line 20, are implemented to restrict the random search of Welch sets, making possible the generation of reversible automata with larger number of states. Property 3 and Kronecker product validate the obtained Welch sets in lines 9, 22 and 26.

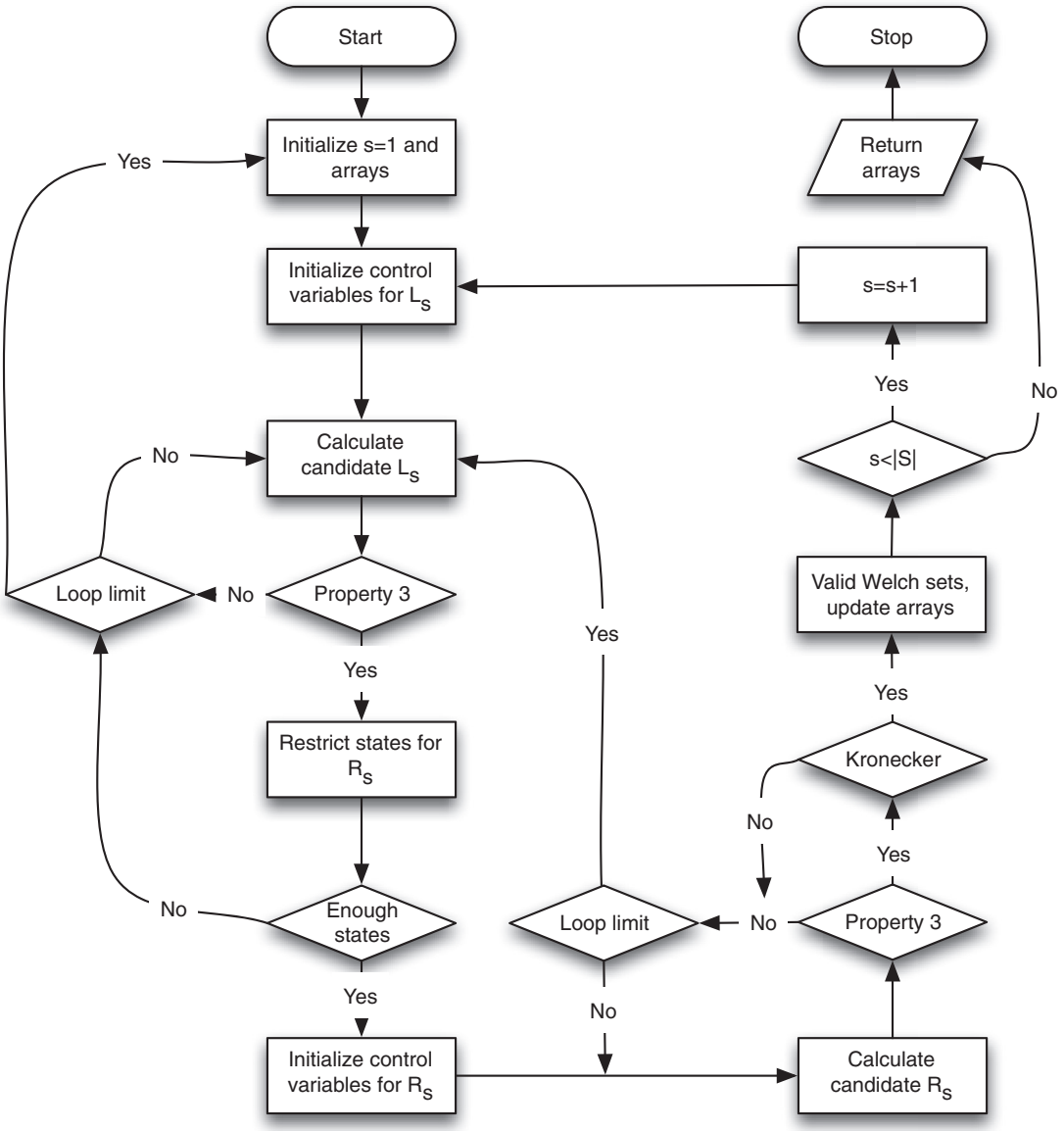

Fig. 3. Flow chart to generate random Welch sets.

The associated flow chart is presented in Fig. 3. Several examples of reversible automata generated with the previous algorithm are displayed in Fig. 5 using $limit_l = limit_r = 100$. The corresponding Welch indices are presented below each image.

5. First representation: bipartite graph with Welch elements

For $s \in S$ and its left Welch set L_s, a left Welch element is a pair (a, s) such that $a \in L_s$. Analogously, a right Welch element is a pair (s, a) such that $a \in R_s$.

Given the matrices G_L, G_R and M_φ, we can define a bipartite graph $P = N_L \times N_R$ where N_L is the set of left Welch elements and N_R is the set of right Welch elements for every state in S.

Every left Welch element $(a, b) \in N_L$ holds that $a \in L_b$ for $a, b \in S$. Analogously, every right Welch element $(d, e) \in N_R$ holds that $e \in R_d$ for $d, e \in S$. There is a directed edge from (a, b) to (d, e) labeled by $\varphi^{-1}(bd)$. Similarly, the directed edge from (d, e) to (a, b) is labeled by $\varphi(ea)$. Fig. 4 presents two examples of the bipartite graphs associated with a reversible automaton of 3 states, $o_L = 1$ and $o_R = 3$ (left side); and a reversible automaton of 4 states, $o_L = 2$ and $o_R = 2$. Edges are colored according to φ^{-1} from N_L to N_R, and using φ from N_R to N_L. Since the graphical representation can be very complicated for larger values of number of states and Welch indices, a matrix definition will be used for further analysis.

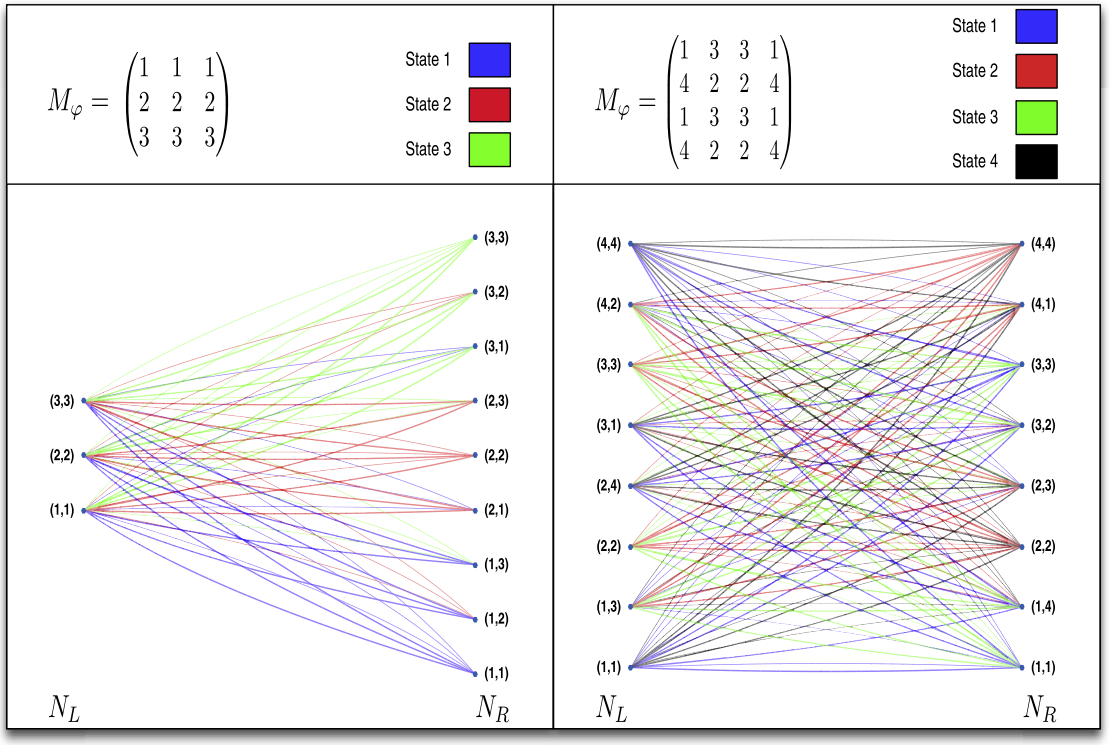

Fig. 4. Digraphs P related to reversible automata of 3 and 4 states respectively..

For $p_L = (a, b) \in N_L$ let us define $\gamma(p_L) = a$ and $\delta(p_L) = b$. In the same way, for $p_R = (d, e) \in N_R$ let us define $\gamma(p_R) = e$ and $\delta(p_R) = d$. Thus, the directed edge (p_L, p_R) is labeled by $\mu(p_L, p_R) = \varphi^{-1}(\delta(p_L)\delta(p_R))$ and the directed edge (p_R, p_L) is labeled by $\eta(p_R, p_L) = \varphi(\gamma(p_R)\gamma(p_L))$.

P can be represented by a matrix M_P with row indices N_L and column indices N_R. Every position $M_P(p_L, p_R)$ has two entries, $\mu(p_L, p_R)$ and $\eta(p_R, p_L)$. With this, the following result can be established.

Theorem 1. *For every $s \in S$ and fixed states $a, e \in S$, there is a unique pair of nodes (p_L, p_R) in P with $\gamma(p_L) = a$ and $\gamma(p_R) = e$ such that $\mu(p_L, p_R) = s$.*

Proof. Every string in S^3 is represented by a directed edge from a node in N_L to other node in N_R. By Property 2 and Remark 1, for every $a \in S$ there is a subset $N'_L \subset N_L$ with $|N'_L| = o_L$ such that $\gamma(p_L) = a$ for each $p_L \in N'_L$. Analogously, for every $e \in S$ there is a subset $N'_R \subset N_R$ with $|N'_R| = o_R$ such that $\gamma(p_R) = e$ for each $p_R \in N'_R$. Thus there are $o_L o_R = |S|$ directed edges in P to represent all the sequences *ase* for fixed states a, e and every $s \in S$. Since there are exactly $|S|$ strings *ase*, there is a unique pair (p_L, p_R) with $p_L \in N'_L$ and $p_R \in N'_R$ such that $\mu(p_L, p_R) = s$. □

Direct results from Theorem 1 is the next one.

Corollary 2. *For every $s \in S$ and fixed states $d, b \in S$, there is a unique pair of nodes (p_R, p_L) in P with $\delta(p_R) = d$ and $\delta(p_L) = b$ such that $\eta(p_R, p_L) = s$.*

Corollary 3. *For every $s \in S$ and fixed states $e, a \in S$, there are $|S|$ pairs of nodes (p_R, p_L) in P with $\gamma(p_R) = e$, $\gamma(p_L) = a$ such that $\eta(p_R, p_L) = s$.*

Proof. For every $e \in S$ there is a subset $N'_R \subset N_R$ with $|N'_R| = o_R$ nodes such that $\gamma(p_R) = e$ for each $p_R \in N'_R$. Besides, for every $a \in S$ there is a subset $N'_L \subset N_L$ with $|N'_L| = o_L$ such that $\gamma(p_L) = a$ for each $p_L \in N'_L$. Thus there are $o_R o_L = |S|$ directed edges in P to represent the neighborhood *ea* such that $\eta(p_R, p_L) = s$. □

As an illustrative example, Table 1 presents a reversible cellular automata with $|S| = 6$, $o_L = 3$ and $o_R = 2$ with Welch sets generated at random applying the algorithm in the previous section.

Fig. 6 displays some evolution examples of this automaton. The matrix M_P associated with this rule is presented in Table 2.

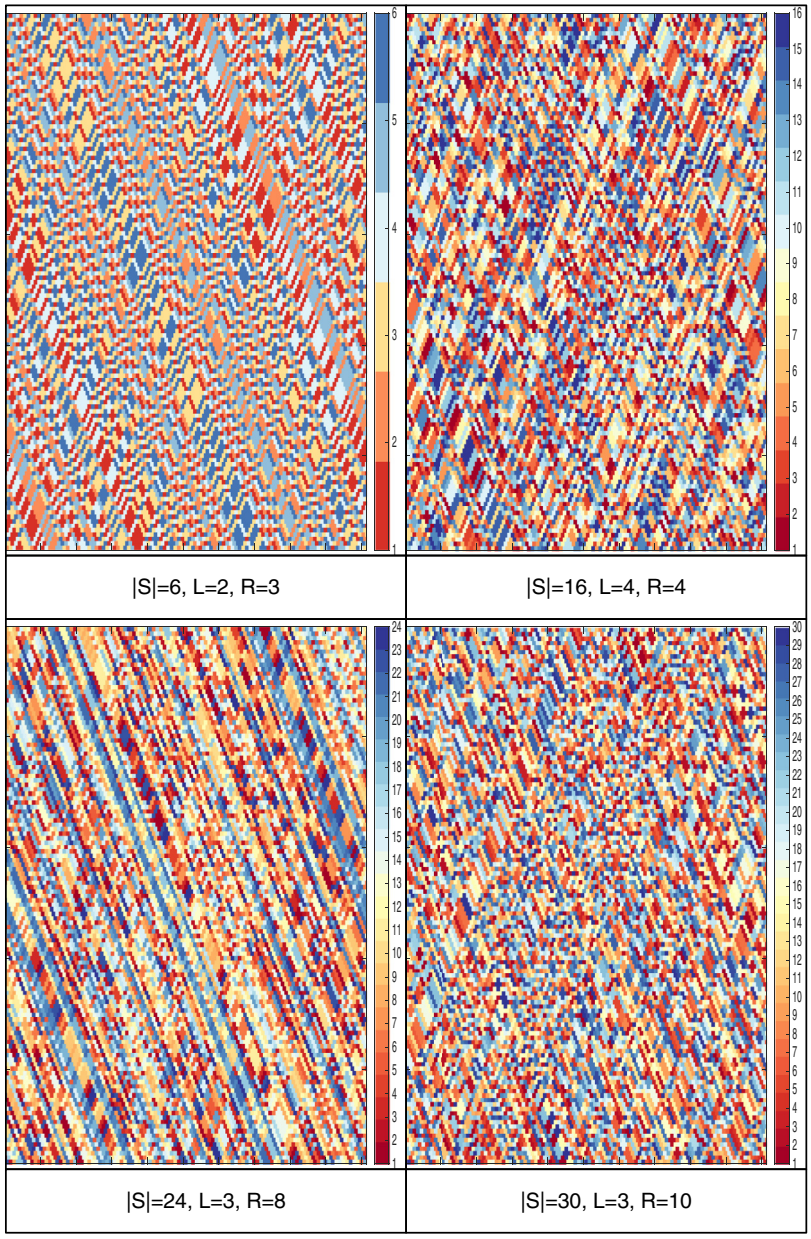

Fig. 5. Different reversible cellular automata with 6, 16, 24 and 30 cell-states generated by the algorithm. Every example is specified by a distinct combination of Welch indices with 100 cells evolving in 120 steps. Cell-state colors are presented on the right side of every evolution.

Table 1
Random reversible cellular automata with 6 cell-states. Indexes of M_φ are implicitly defined, thus $M_\varphi(a, b)$ indicates the evolution of the neighborhood ab.

Evolution rule M_φ	State s	L_s	R_s
4 1 4 1 6 6	1	1 2 6	2 4
3 1 5 1 3 5	2	3 4 5	2 4
3 2 5 2 3 5	3	2 3 5	1 5
4 2 4 2 6 6	4	1 4 6	1 3
3 2 5 2 3 5	5	2 3 5	3 6
4 1 4 1 6 6	6	1 4 6	5 6

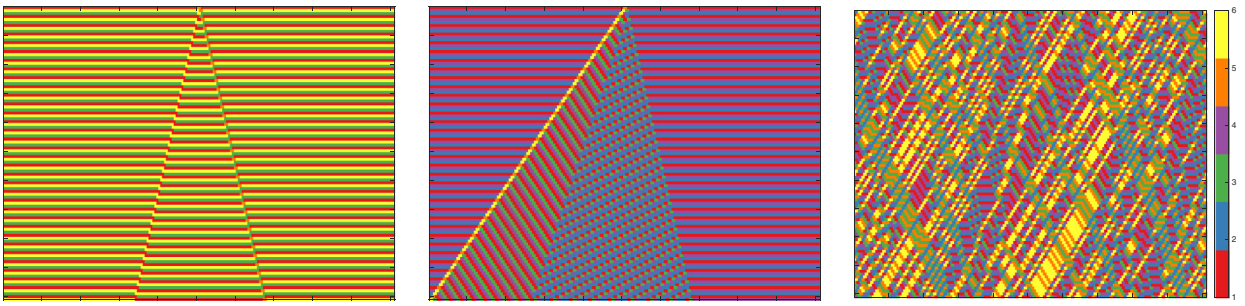

Fig. 6. Evolution examples of the reversible automaton described in Table 1 for 100 cells in 100 time steps.

Table 2
Matrix M_P for the evolution rule and Welch elements presented in Table 1. For row and column indices $p \in N_L$ and $q \in N_R$ respectively, $\gamma(p)$ and $\gamma(q)$ are in bold. In the same way, every $\mu(p, q)$ is in bold too.

N_L/N_R	(1,2)	(1,4)	(2,2)	(2,4)	(3,1)	(3,5)	(4,1)	(4,3)	(5,3)	(5,6)	(6,5)	(6,6)
(**1**,1)	(**2**,3)	(**2**,4)	(**4**,3)	(**4**,4)	(**2**,4)	(**2**,3)	(**4**,4)	(**4**,3)	(**2**,3)	(**2**,4)	(**4**,3)	(**4**,4)
(**2**,1)	(**2**,1)	(**2**,2)	(**4**,1)	(**4**,2)	(**2**,1)	(**2**,2)	(**4**,1)	(**4**,2)	(**2**,2)	(**2**,1)	(**4**,2)	(**4**,1)
(**6**,1)	(**2**,5)	(**2**,6)	(**4**,5)	(**4**,6)	(**2**,6)	(**2**,5)	(**4**,6)	(**4**,5)	(**2**,5)	(**2**,6)	(**4**,5)	(**4**,6)
(**3**,2)	(**2**,5)	(**2**,4)	(**4**,5)	(**4**,4)	(**2**,4)	(**2**,5)	(**4**,4)	(**4**,5)	(**2**,5)	(**2**,4)	(**4**,5)	(**4**,4)
(**4**,2)	(**2**,1)	(**2**,2)	(**4**,1)	(**4**,2)	(**2**,1)	(**2**,2)	(**4**,1)	(**4**,2)	(**2**,2)	(**2**,1)	(**4**,2)	(**4**,1)
(**5**,2)	(**2**,3)	(**2**,6)	(**4**,3)	(**4**,6)	(**2**,6)	(**2**,3)	(**4**,6)	(**4**,3)	(**2**,3)	(**2**,6)	(**4**,3)	(**4**,6)
(**2**,3)	(**1**,1)	(**1**,2)	(**5**,1)	(**5**,2)	(**5**,1)	(**5**,2)	(**1**,1)	(**1**,2)	(**5**,2)	(**5**,1)	(**1**,2)	(**1**,1)
(**3**,3)	(**1**,5)	(**1**,4)	(**5**,5)	(**5**,4)	(**5**,4)	(**5**,5)	(**1**,4)	(**1**,5)	(**5**,5)	(**5**,4)	(**1**,5)	(**1**,4)
(**5**,3)	(**1**,3)	(**1**,6)	(**5**,3)	(**5**,5)	(**5**,6)	(**5**,3)	(**1**,6)	(**1**,3)	(**5**,3)	(**5**,6)	(**1**,3)	(**1**,6)
(**1**,4)	(**1**,3)	(**1**,4)	(**3**,3)	(**3**,4)	(**3**,4)	(**3**,3)	(**1**,4)	(**1**,3)	(**3**,3)	(**3**,4)	(**1**,3)	(**1**,4)
(**4**,4)	(**1**,1)	(**1**,2)	(**3**,1)	(**3**,2)	(**3**,1)	(**3**,2)	(**1**,1)	(**1**,2)	(**3**,2)	(**3**,1)	(**1**,2)	(**1**,1)
(**6**,4)	(**1**,5)	(**1**,6)	(**3**,5)	(**3**,6)	(**3**,6)	(**3**,5)	(**1**,6)	(**1**,5)	(**3**,5)	(**3**,6)	(**1**,5)	(**1**,6)
(**2**,5)	(**6**,1)	(**6**,2)	(**3**,1)	(**3**,2)	(**3**,1)	(**3**,2)	(**6**,1)	(**6**,2)	(**3**,2)	(**3**,1)	(**6**,2)	(**6**,1)
(**3**,5)	(**6**,5)	(**6**,4)	(**3**,5)	(**3**,4)	(**3**,4)	(**3**,5)	(**6**,4)	(**6**,5)	(**3**,5)	(**3**,4)	(**6**,5)	(**6**,4)
(**5**,5)	(**6**,3)	(**6**,6)	(**3**,3)	(**3**,6)	(**3**,6)	(**3**,3)	(**6**,6)	(**6**,3)	(**3**,3)	(**3**,6)	(**6**,3)	(**6**,6)
(**1**,6)	(**6**,3)	(**6**,4)	(**5**,3)	(**5**,4)	(**5**,4)	(**5**,3)	(**6**,4)	(**6**,3)	(**5**,3)	(**5**,4)	(**6**,3)	(**6**,4)
(**4**,6)	(**6**,1)	(**6**,2)	(**5**,1)	(**5**,2)	(**5**,1)	(**5**,2)	(**6**,1)	(**6**,2)	(**5**,2)	(**5**,1)	(**6**,2)	(**6**,1)
(**6**,6)	(**6**,5)	(**6**,6)	(**5**,5)	(**5**,6)	(**5**,6)	(**5**,5)	(**6**,6)	(**6**,5)	(**5**,5)	(**5**,6)	(**6**,5)	(**6**,6)

Based on the proof of Theorem 1, matrix M_P can be used to obtain the evolution of a reversible cellular automaton in both senses. For an initial configuration c with $|c| = m$ and $1 \leq i \leq m$, the forward evolution is obtained following the next steps:

1. For every i with modulus $mod(i, 3) = 1$, take every state $c(i)$ and the set of rows $N'_i \subseteq N_L$ such that $\gamma(p) = c(i)$ for each $p \in N'_i$.
2. For every i with $mod(i, 3) = 0$, take every state $c(i)$ and the set of columns $N'_i \subseteq N_R$ such that $\gamma(q) = c(i)$ for each $q \in N'_i$.
3. For every i with $mod(i, 3) = 2$ and from the previous row and column sets, take the unique pair holding that $\mu(p_{i-1}, q_{i+1}) = c(i)$ where $p_{i-1} \in N'_{i-1}$ and $q_{i+1} \in N'_{i+1}$. The evolution of $c(i-1)c(i)c(i+1)$ is given by $\delta(p_{i-1})\delta(q_{i+1})$.
4. For every i with $mod(i, 3) = 0$ and j with $mod(i+1, 3) = 1$, the evolution of $c(i)c(j)$ is obtained taking the directed edge $\eta(q_i, p_j)$.

The backwards evolution is calculated as follows:

1. For every i with $mod(i, 3) = 1$, take every state $c(i)$ and the set of columns $N'_i \subseteq N_R$ such that $\delta(q) = c(i)$ for each $q \in N'_i$.
2. For every i with $mod(i, 3) = 0$, take every state $c(i)$ and the set of columns $N'_i \subseteq N_L$ such that $\delta(p) = c(i)$ for each $p \in N'_i$.
3. For every $i = mod(i, 3) = 2$ and from the previous rows and columns, take the unique pair with $\eta(qi-1, pi+1) = c(i)$. The inverse evolution of $c(i-1)c(i)c(i+1)$ is given by $\gamma(q_{i-1})\gamma(p_{i+1})$.
4. For every i with $mod(i, 3) = 0$ and j with $mod(i+1, 3) = 1$, the inverse evolution of $c(i)c(j)$ is obtained taking the directed edge $\mu(p_i, q_j)$.

Fig. 7 describes how the previous algorithms are able to calculate the forward and backwards evolution from a given random configuration.

In this figure, for the forward direction, every sequence $w \in S^3$ is mapped into a pair of nodes (p, q) such that $\gamma(p) = w(1)$, $\gamma(q) = w(3)$ and $\mu(p, q) = w(2)$ (bold font). Then the evolution of w is given by $\delta(p)\delta(q)$ (italic font). In the second part of the process, the missing cell-states are obtained taking every directed edge $\eta(q, p)$ (italic font) in order to calculate the complete evolution. The backward evolution can be analogously described. Notice that the last state in the evolution is repeated to represent periodic boundary conditions.

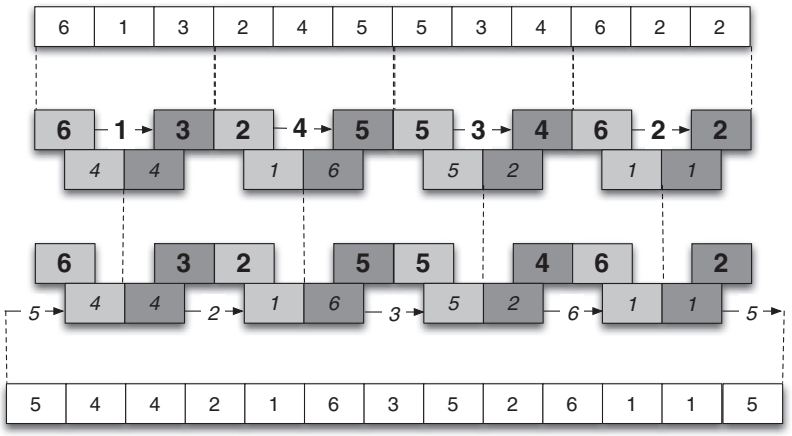

Fig. 7. Forward and backward evolution obtained by means of the bipartite graph *P*.

Table 3
Bijections from Welch elements of L_s and R_s into X and Y respectively.

L_s	Symbol	L_s	Symbol	L_s	Symbol	R_s	Symbol	R_s	Symbol
(**1**,1)	X_1	(**2**,3)	X_7	(**2**,5)	X_{13}	(1,**2**)	Y_1	(4,**1**)	Y_7
(**2**,1)	X_2	(**3**,3)	X_8	(**3**,5)	X_{14}	(1,**4**)	Y_2	(4,**3**)	Y_8
(**6**,1)	X_3	(**5**,3)	X_9	(**5**,5)	X_{15}	(2,**2**)	Y_3	(5,**3**)	Y_9
(**3**,2)	X_4	(**1**,4)	X_{10}	(**1**,6)	X_{16}	(2,**4**)	Y_4	(5,**6**)	Y_{10}
(**4**,2)	X_5	(**4**,4)	X_{11}	(**4**,6)	X_{17}	(3,**1**)	Y_5	(6,**5**)	Y_{11}
(**5**,2)	X_6	(**6**,4)	X_{12}	(**6**,6)	X_{18}	(3,**5**)	Y_6	(6,**6**)	Y_{12}

Although the previous algorithms are more technical and complicated than using the matrix M_φ to obtain the evolution of a reversible automata, their relevance is to show that invertible evolution rules can be represented by a bipartite graph, which opens a future direction of research.

With the nodes of *P* we can obtain a reinterpretation of Kari's proof about the representation of the dynamics of reversible cellular automata by block permutations and shifts [12]. The original proof demonstrates that the evolution of a reversible cellular automata is equivalent to mapping blocks of 6*r* states into other blocks with the same length, and from these blocks, apply a shift of 3*r* places and a second mapping to other blocks of 6*r* states. This demonstration is completely based on the properties of Welch indices and only takes the case when $o_L = 1$.

Suppose that we have a reversible automaton with $r = 1/2$ and $|S| = 4$; if $o_L = 1$ then $|N_L| = 4$ and $|N_R| = 16$. This implies that we can define a bijection from N_L to S and from N_R to S^2. Thus, a permutation from blocks $w \in S^3$ to blocks of the same length is easily defined.

However, if $o_L \neq 1$ a permutation cannot be defined. For the same case assume that $o_L = o_R = 2$, hence $|N_L| = 8$ and $|N_R| = 8$. In order to define a permutation, we should map blocks w into other blocks of the same size, such that they should have 8 different left parts and 8 distinct right parts, which is not possible. This fact does not imply that the original result is wrong, but it is required to generalize the idea that the evolution of a reversible automaton is equivalent to perform two mappings of blocks with a shift between them.

In order to obtain this generalization, the elements of N_L and N_R (or the nodes of *P*) can be mapped into two different sets of symbols X and Y respectively. Instead of defining that every block w permutes into another block of 3 states, it is only necessary that the block maps into a block xy where $x \in X$ and $y \in Y$.

Thus, it is not mandatory that the block xy has the same length that w, therefore the set sizes $|X|$ and $|Y|$ can be arbitrarily controlled to generalize the mappings of blocks with a shift between them for any combination of o_L and o_R such that $o_L o_R = |S|$.

Notice that every sequence $w \in S^3$ maps into a unique pair (p_L, p_R) in *P* such that $\mu(p_L, p_R) = w(2)$. The same happens in the inverse direction, there is a unique pair (p_R, p_L) in *P* such that $\eta(p_R, p_L) = w(2)$. This implies that every left Welch element defined by φ is a right Welch element determined by φ^{-1}.

Thus, we can define two sets of symbols X and Y such that $|X| = |N_L| = |S|o_L$ and $|Y| = |N_R| = |S|o_R$. Every left Welch set defined by φ can be mapped bijectively into an element of X and consequently each right Welch set can be mapped bijectively into an element of Y.

Then, every configuration can be partitioned in blocks of three cells and every block is represented by a pair of symbols xy for $x \in X$ and $y \in Y$.Analogously, every pair yx maps into a unique sequence $w \in S^3$ applying now the bijection from Y into N_R and from X into N_L.

For the reversible cellular automaton in Table 1, let us define a set of bijections from L_s into X and from R_s into Y in Table 3.

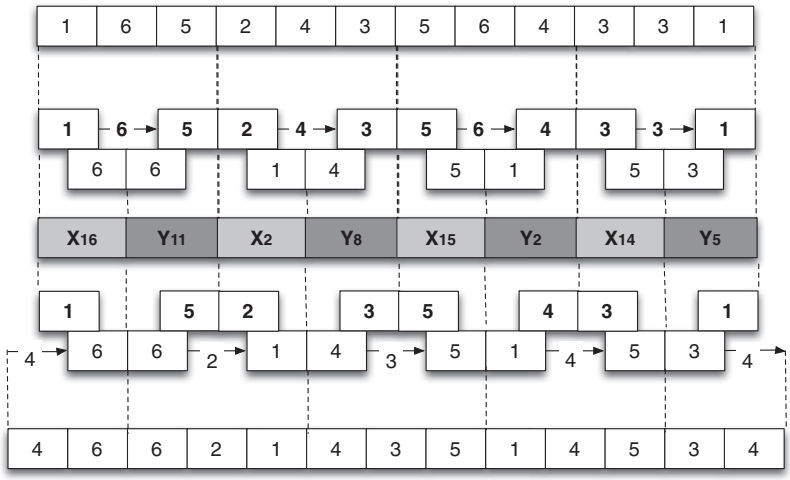

Fig. 8. Dynamics of the reversible cellular automaton represented by block mappings and shifts.

Given a random initial configuration, Fig. 8 describes the application of block mappings for every sequence $w \in S^3$. First, each w is mapped into its respective pair of nodes (p, q) in P. Then each p is mapped into its corresponding element in X according to Table 3. The analogous process is applied for every y to produce a sequence of symbols $xyxy \ldots xy$. From this sequence, each block yx is mapped into its particular pair (q, p) in P, thereafter every pair (q, p) maps into a unique sequence $w \in S^3$. With this, we obtain the evolution of the initial configuration.

6. Second representation: Bipartite graph with welch sets

With the matrices G_L, G_R and M_φ, another bipartite graph $Q = \mathcal{L} \times \mathcal{R}$ is defined where $\mathcal{L}$ is the family of left Welch sets and $\mathcal{R}$ is the family of right Welch sets. For every $s \in S$, each set $L_s \in \mathcal{L}$ and every $R_s \in \mathcal{R}$. There is a directed edge (L_s, R_s) labeled by s. The directed edges from $\mathcal{R}$ into $\mathcal{L}$ are implicitly defined by Property 3. That is, $(R_a, L_b) = R_a \cap L_b$ for each $R_a \in \mathcal{R}$ and every $L_b \in \mathcal{L}$.

Fig. 9 presents de digraphs Q associated with the reversible automata previously used in Fig. 4. Bold lines describe the edges from $\mathcal{L}$ to $\mathcal{R}$. Notice that there are states with the same Welch sets in the second case, thus only one instance of each set is taken to define the graph.

Q can be represented by a matrix M_Q with row indices $\mathcal{L}$ and column indices $\mathcal{R}$. Every position $(L_a, R_a) = a$, otherwise $(L_a, R_b) = \emptyset$ for $a, b \in S$, $a \neq b$. Since Property 3 is accomplished by the intersection of Welch sets, there is no need to write explicitly the corresponding entries in M_Q.

The maximum order of M_Q is the one of M_φ ($|S| \times |S|$), but only with $|S|$ entries instead of $|S| \times |S|$. All our computational experiments, however, have shown empirically that the order of M_Q is lesser than $|S| \times |S|$ because different states can have the same Welch sets. Therefore, M_Q can be used as a reduced representation of M_φ, which would be favorable for practical applications of reversible automata in order to minimize computational resources.

Matrix M_Q can be used as well to obtain the evolution of a reversible cellular automaton in both directions. For an initial configuration c with $|c| = m$ and $1 \leq i \leq m$, the forward evolution is obtained following the next steps:

1. For every state $c(i)$, take the family $V_L \subseteq \mathcal{L}$ with $c(i) \in L \in V_L$.
2. For every $j = mod(i + 1, m)$ and state $c(j)$, take the family $V_R \subseteq \mathcal{R}$ with $c(j) \in R \in V_R$.
3. Select the unique pair $(L, R) \neq \emptyset$ in M_Q for $L \in V_L$ and $R \in V_R$.
4. The evolution of $c(i)c(j)$ is given by the label $(L, R) = s$.

The inverse evolution is obtained in a more direct way by Property 3.

1. For every state $c(i)$, take the unique entry in M_Q with $(L_{c(i)}, R_{c(i)}) = c(i)$.
2. The inverse evolution of $c(i)c(i + 1)$ is given by $a = R_{c(i)} \cap L_{c(i+1)}$.

As an illustrative example, Table 4 shows a reversible cellular automaton with $|S| = 18$, $o_L = 3$ and $o_R = 6$ with Welch sets generated at random applying the algorithm in Section 4.

Fig. 10 depicts some evolution examples of this automaton. The Welch sets associated with this automaton are described in Table 5.

This table shows that there are only 11 different left Welch sets and 15 distinct right Welch sets. Every Welch set is enumerated according to the first state defining it. With these sets, we specify M_Q in Table 6.

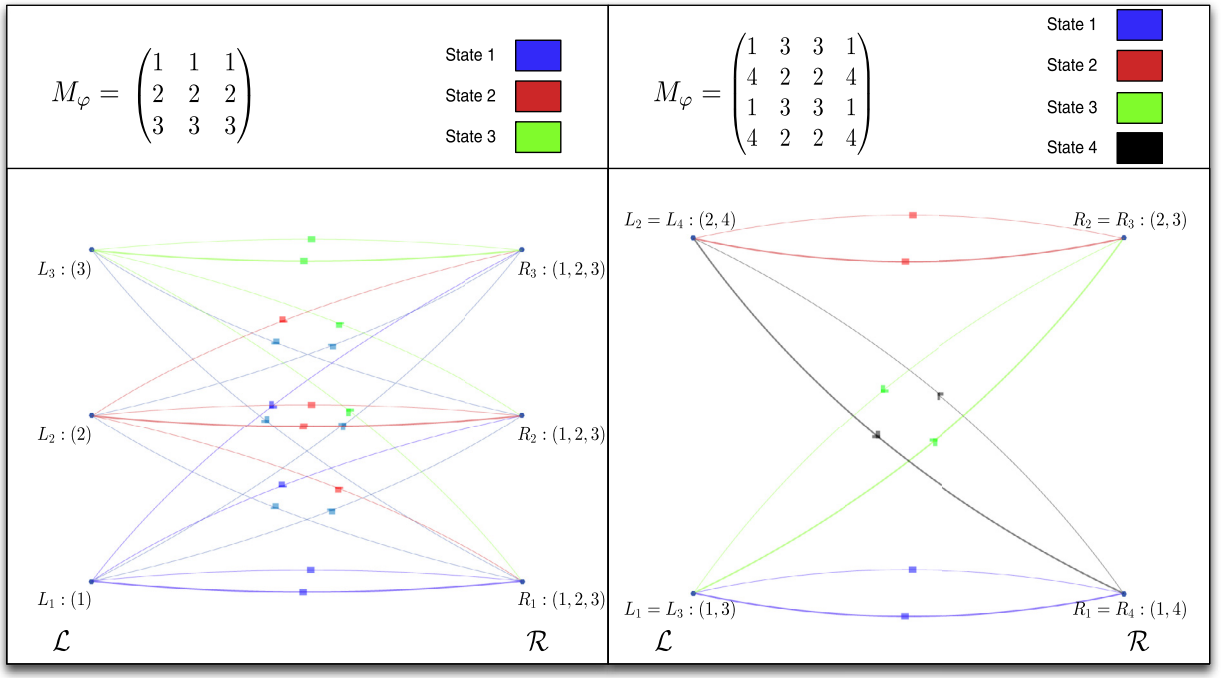

Fig. 9. Digraphs Q related to reversible automata in Fig. 4.

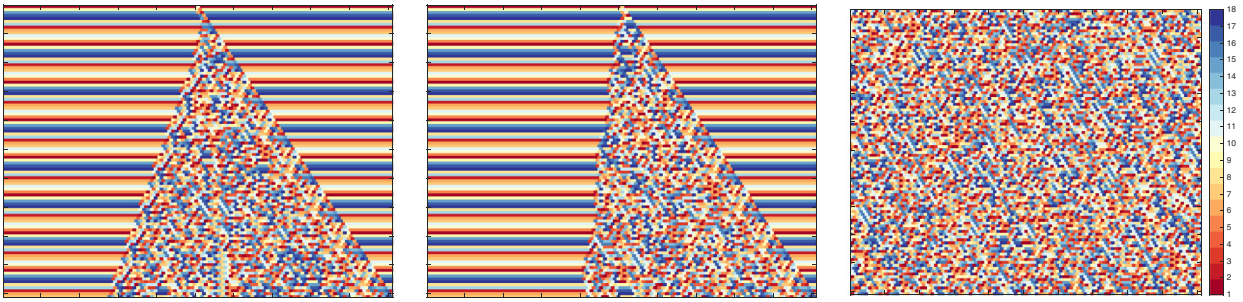

Fig. 10. Evolution examples of the reversible automaton described in Table 4 for 100 cells in 100 time steps.

Table 4
Random reversible cellular automata with 18 cell-states. Row and column indices indicate left and right part of each neighborhood respectively.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	9	6	9	6	6	7	9	7	7	9	6	7	7	6	6	9	9	7
2	4	6	4	6	6	15	4	15	15	4	6	4	15	6	6	4	15	15
3	13	14	14	12	14	12	14	13	12	13	13	13	12	14	14	13	12	12
4	13	14	14	12	14	12	14	13	12	13	13	13	12	14	14	13	12	12
5	9	1	9	1	1	1	9	2	2	9	1	2	2	2	1	9	9	2
6	5	16	5	16	16	7	5	7	7	5	16	7	7	16	16	5	5	7
7	11	8	11	8	10	10	11	10	10	11	10	10	8	8	8	11	11	8
8	13	14	14	12	14	12	14	13	12	13	13	13	12	14	14	13	12	12
9	4	6	4	6	6	15	4	15	15	4	6	4	15	6	6	4	15	15
10	5	1	5	1	1	1	5	2	2	5	1	2	2	2	1	5	5	2
11	11	8	11	8	10	10	11	10	10	11	10	10	8	8	8	11	11	8
12	18	17	18	3	3	17	3	17	3	18	18	3	17	3	17	18	18	17
13	11	1	11	1	1	1	11	2	2	11	1	2	2	2	1	11	11	2
14	5	16	5	16	16	7	5	7	7	5	16	7	7	16	16	5	5	7
15	18	17	18	3	3	17	3	17	3	18	18	3	17	3	17	18	18	17
16	4	16	4	16	16	15	4	15	15	4	16	4	15	16	16	4	15	15
17	18	17	18	3	3	17	3	17	3	18	18	3	17	3	17	18	18	17
18	9	8	9	8	10	10	9	10	10	9	10	10	8	8	8	9	9	8

Table 5
Welch sets for the reversible cellular automaton with 18 cell-states.

State s	L_s	ID	R_s	ID
1	5 10 13	L_1	2 4 5 6 11 15	R_1
2	5 10 13	L_1	8 9 12 13 14 18	R_2
3	12 15 17	L_3	4 5 7 9 12 14	R_3
4	2 9 16	L_4	1 3 7 10 12 16	R_4
5	6 10 14	L_5	1 3 7 10 16 17	R_5
6	1 2 9	L_6	2 4 5 11 14 15	R_6
7	1 6 14	L_7	6 8 9 12 13 18	R_7
8	7 11 18	L_8	2 4 13 14 15 18	R_8
9	1 5 18	L_9	1 3 7 10 16 17	R_5
10	7 11 18	L_8	5 6 8 9 11 12	R_{10}
11	7 11 13	L_{11}	1 3 7 10 16 17	R_5
12	3 4 8	L_{12}	4 6 9 13 17 18	R_{12}
13	3 4 8	L_{12}	1 8 10 11 12 16	R_{13}
14	3 4 8	L_{12}	2 3 5 7 14 15	R_{14}
15	2 9 16	L_4	6 8 9 13 17 18	R_{15}
16	6 14 16	L_{16}	2 4 5 11 14 15	R_6
17	12 15 17	L_3	2 6 8 13 15 18	R_{17}
18	12 15 17	L_3	1 3 10 11 16 17	R_{18}

Table 6
Matrix M_Q for the reversible automaton of 18 cell-states.

L/R	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_{10}	R_{12}	R_{13}	R_{14}	R_{15}	R_{17}	R_{18}
L_1	1	2													
L_3			3											17	18
L_4				4									15		
L_5					5										
L_6						6									
L_7							7								
L_8								8	10						
L_9					9										
L_{11}					11										
L_{12}										12	13	14			
L_{16}						16									

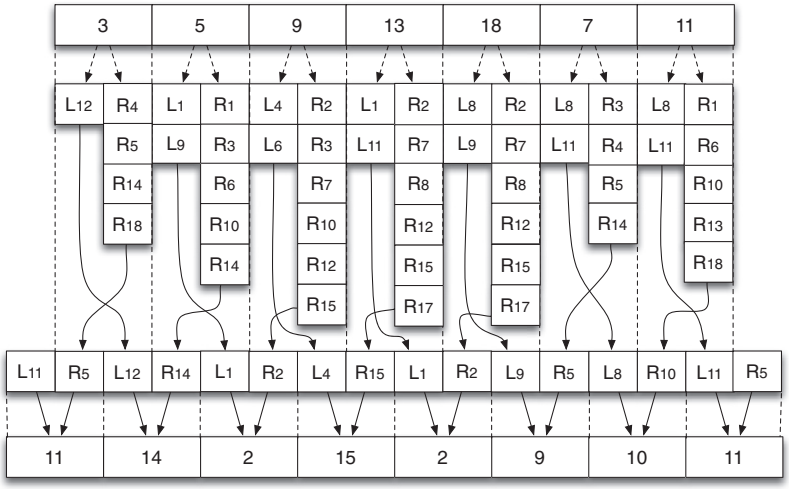

Fig. 11. Evolution of the reversible cellular automaton in Table 4 described by mappings and intersection between Welch sets.

Fig. 11 presents an example calculating the evolution for the reversible cellular automaton in Table 4 from a random initial configuration. Every state is mapped into the Welch sets containing it and then the only nontrivial intersection, between left and right Welch sets, is taken to calculate the evolution. Notice that the Welch sets at both ends are repeated to represent periodic boundary conditions.

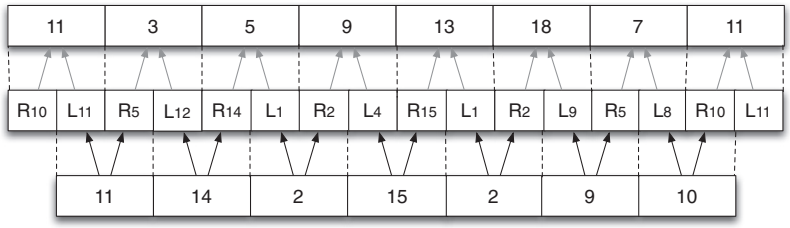

Fig. 12. Reversible evolution computed by the intersection of Welch sets.

For the same automaton, the inverse evolution is easily calculated taking the intersections of the Welch sets associated with every state (Fig. 12). This case exemplifies how the forward and backwards evolution of a reversible automaton is calculated using M_Q, which has only $|S|$ elements instead of the $|S| \times |S|$ elements in the M_φ.

7. Final remarks

This paper has presented an algorithm based on the properties of Welch indices in order to calculate a random reversible cellular automaton. This algorithm is able to calculate reversible automata with tens of states.

The properties of Welch indices are employed as well to specify two types of bipartite graphs, which represent the dynamics of reversible automata.

The first graph takes as nodes the Welch elements, and the edges among them are utilized to represent the evolution in both directions of reversible automata.

The nodes of this graph are useful as well to generalize the result that the dynamical behavior of reversible automata is equivalent to perform two mappings of blocks with a shift between them.

The second graph takes every different Welch set to establish a reduced version of the evolution rule associated with a given reversible automaton. This graph can be used as well to calculate the evolution of the reversible automaton with the intersection property of Welch sets. Therefore, this graph can be useful to minimize the computational resources required in applications of reversible automata in data cyphering and error-correcting codes.

The main objective of the paper is to show the utility of Welch sets to generate and represent reversible automata by bipartite graphs. These results may open a new research area, where cellular automata and graph theory experts can utilize the properties and operations of bipartite graphs to complement the study of the structural and dynamical behavior of reversible automata in future papers.

Additional further work may imply the definition of an algorithm to calculate random reversible automata in more dimensions, or to enumerate reversible automata for a desired combination of Welch indices.

Acknowledgment

This work has been supported by CONACYT project No. CB-2014-237323 and by IPN Collaboration Network “ Grupo de Sistemas Complejos del IPN ”.

References

[1] R. Alonso-Sanz, Reversible cellular automata with memory of delay type, Complexity 20 (1) (2014) 49–56.
[2] A. Bakhshandeh, Z. Eslami, An authenticated image encryption scheme based on chaotic maps and memory cellular automata, Opt. Lasers Eng. 51 (6) (2013) 665–673.
[3] T. Boykett, Efficient exhaustive listings of reversible one dimensional cellular automata, Theor. Comput. Sci. 325 (2) (2004) 215–247.
[4] T. Boykett, Orderly algorithm to enumerate central groupoids and their graphs, Acta Mathematica Sinica 23 (2) (2007) 249–264.
[5] T. Boykett, Rectangular groupoids and related structures, Discrete Math. 313 (13) (2013) 1409–1418.
[6] T. Boykett, J. Kari, S. Taati, Conservation laws in rectangular ca., J. Cell. Automata 3 (2) (2008) 115–122.
[7] E. Czeizler, J. Kari, A Tight Linear Bound on the Neighborhood of Inverse Cellular Automata, in: Automata, languages and programming, Springer, 2005, pp. 410–420.
[8] K.M. Faraoun, Design of fast one-pass authenticated and randomized encryption schema using reversible cellular automata, Commun. Nonlin. Sci. Numer. Simul. 19 (9) (2014) 3136–3148.
[9] K.M. Faraoun, Fast encryption of rgb color digital images using a tweakable cellular automaton based schema, Opt. Laser Technol. 64 (2014) 145–155.
[10] G.A. Hedlund, Endomorphisms and automorphisms of the shift dynamical system, Theory Comput. Syst. 3 (4) (1969) 320–375.
[11] W.-T. Hu, M.-C. Li, C. Guo, L.-F. Yuan, A reversible steganography scheme of secret image sharing based on cellular automata and least significant bits construction, Math. Prob. Eng. 2015 (2015).
[12] J. Kari, Representation of reversible cellular automata with block permutations, Math. Syst. Theory 29 (1) (1996) 47–61.
[13] M.E. Koroglu, I. Siap, H. Akın, Error correcting codes via reversible cellular automata over finite fields, Arabian J. Sci. Eng. 39 (3) (2014) 1881–1887.
[14] A. Martín del Rey, G. Rodríguez Sánchez, Reversible elementary cellular automaton with rule number 150 and periodic boundary conditions over $\mathbb{F}_p$, Int. J. Mod. Phys. C 26 (11) (2015) 1550120.
[15] M. Nasu, Local maps inducing surjective global maps of one-dimensional tessellation automata, Math. Syst. Theory 11 (1) (1977) 327–351.
[16] F. Neumann, I. Wegener, Randomized local search, evolutionary algorithms, and the minimum spanning tree problem, Theor. Comput. Sci. 378 (1) (2007) 32–40.

[17] J.C. Seck-Tuoh-Mora, G.J. Martínez, R. Alonso-Sanz, N. Hernández-Romero, Invertible behavior in elementary cellular automata with memory, Inf. Sci. 199 (2012) 125–132.
[18] J.C. Seck-Tuoh-Mora, S.V.C. Vergara, G.J. Martínez, H.V. McIntosh, Procedures for calculating reversible one-dimensional cellular automata, Physica D 202 (1) (2005) 134–141.
[19] A. Souyah, K.M. Faraoun, Fast and efficient randomized encryption scheme for digital images based on quadtree decomposition and reversible memory cellular automata, Nonlin. Dyn. (2015) 1–18.
[20] X. Wang, D. Luan, A novel image encryption algorithm using chaos and reversible cellular automata, Commun. Nonlin. Sci. Numer. Simul. 18 (11) (2013) 3075–3085.
[21] B. Yang, C. Wang, A. Xiang, Reversibility of general 1d linear cellular automata over the binary field z2 under null boundary conditions, Inf. Sci. 324 (2015) 23–31.
[22] X. Zhang, R. Lu, H. Zhang, C. Xu, A new public key encryption scheme based on layered cellular automata., TIIS 8 (10) (2014) 3572–3590.