EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 6, No. 3, 2013, 315-334
ISSN 1307-5543 – www.ejpam.com

2-D Reversible Cellular Automata with Nearest and Prolonged Next Nearest Neighborhoods under Periodic Boundary †

Irfan Siap[1], Hasan Akin[2], Selman Uguz[3,*]

[1] *Department of Mathematics, Arts and Science Faculty, Yıldız Technical University, Istanbul, 34210, Turkey*
[2] *Department of Mathematics, Faculty of Education, Zirve University, Gaziantep, 27260, Turkey*
[3] *Department of Mathematics, Arts and Science Faculty, Harran University, Sanliurfa, 63120, Turkey*

Abstract. In this paper, we study a 2-dimensional cellular automaton generated by a new local rule with the nearest neighborhoods and prolonged next nearest neighborhoods under periodic boundary condition over the ternary field ($\mathbb{Z}_3$). We obtain the rule matrix of this cellular automaton and characterize this family by exploring some of their important characteristics. We get some recurrence equations which simplifies the computation of the rank of the rule matrix related to the 2-dimensional cellular automaton drastically. Next, we propose an algorithm to determine the rank of the rule matrix. Finally, we conclude by presenting an application to error correcting codes.

2010 Mathematics Subject Classifications: 37B15; 15A04.

Key Words and Phrases: Cellular automata, Rule matrix, Matrix algebra, Periodic Boundary Condition, error correcting codes

---

1. Introduction

Cellular automata (CAs for brevity), introduced by Ulam and von Neumann [21] in the early 1950's, have been studied by many workers. Von Neumann [21] showed that a cellular automaton (CA) can be universal. However, due to its complexity, von Neumann rules were never implemented on a computer. In the beginning of the eighties, Stephen Wolfram [22] has studied in much detail a family of simple one-dimensional (1-D) CAs rules and showed that even these simplest rules are capable of emulating complex behavior.

---

†This paper was partially presented at 2nd WORLD CONFERENCE on INFORMATION TECHNOLOGY - WCIT-2011, November 23-26, 2011 Antalya, Turkey.
*Corresponding author.

Email addresses: `irfansiap@hotmail.com` (I. Siap), `akinhasan25@gmail.com` (H. Akin), `selmanuguz@gmail.com` (S. Uguz)

Due to various applications of CAs in many disciplines (e.g., mathematics, physics, computer science, chemistry and so on) with different purposes (e.g., simulation of natural phenomena, pseudo-random number generation, image processing, analysis of universal model of computations, cryptography, coding theory, complexity), the study of CAs has received remarkable attention in the last few years [1–3, 5, 9, 10, 12, 13]. The set of papers [1–3], the entropies of 1-D CAs have been investigated. Most of the work for CAs is done for one dimensional case [11]. Lately, two-dimensional (2-D) CA has found applications in traffic modelling. For instance multi-value (including ternary) cellular automaton models for traffic flow are proposed in [17]. CAs have found applications in Cryptography [5], recently multi state CAs have also found applications on Cryptography [16] and especially two dimensional CA has been proposed for multisecret sharing scheme for colored images [4]. The set of papers [6, 10, 12, 13, 18, 20] deals with the behavior of the linear 2-D CAs over binary fields ($\mathbb{Z}_2$) by using matrix algebras setting. Das [9] has studied the characterization of 1-D CAs by means of matrix algebra. Inokuchi *et al*. [11] have investigated the behaviors of 1-D CA generated by the local rule 156. Khan *et al*. [12] developed an analytical tool to study all the nearest neighborhood 2-D CA linear transformations. They proposed a new rule convention to divide the 2-D linear CAs and tried to study the characterization of that 2-D CAs with different rules.

In [18], we have characterized a 2-D finite CA by using matrix algebra built on $\mathbb{Z}_3$. Also, we have analyzed some results about the rule numbers 2460N and 2460P. In [19], we have obtained necessary and sufficient conditions for the existence of Garden of Eden configurations for 2-D ternary CAs. Also by making use of the matrix representation of 2-D CAs, we have provided an algorithm to obtain the number of Garden of Eden configurations for the 2-D CA defined by rule 2460N.

In present paper, we define the 2-D CAs generated by new local rules so called the nearest neighborhoods and prolonged next nearest neighborhoods over the field $\mathbb{Z}_3$ under **P**eriodic **B**oundary **C**ondition (briefly, PBC). We obtain recurrence equations to compute the rank of the rule matrix related to this 2-D CA. We present an application to error correcting codes.

The rest of this paper is organized as follows. In Section 2 we give basic definitions and notations. In Section 3 we obtain the rule matrix corresponding to a 2-D finite CA with PBC generated by the local rule $NPNN$ over the field $\mathbb{Z}_3$. In Section 4 we study the rank of the rule matrix. In Section 5 we give some examples of the main theorem. We also give an algorithm and apply it to a 2-D CA with larger order. In the last Section, we present an application to error correcting codes and we conclude the paper.

2. Preliminaries

In this section, we introduce 2-D CAs over the field $\mathbb{Z}_3$ by using some local rules. We recall the definition of a CA. We consider the 2-dimensional integer lattice $\mathbb{Z}^2$ and the configuration space $\Omega = \{0,1,2\}^{\mathbb{Z}^2}$ with elements

$$\sigma : \mathbb{Z}^2 \to \{0,1,2\}.$$

The value of σ at a point $v \in \mathbb{Z}^2$ will be denoted by σ_v. Let $u_1, \ldots, u_s \in \mathbb{Z}^2$ be a finite set of distinct vectors and $F : \{0,1,2\}^s \to \{0,1,2\}$ be a function. A CA with local rule F is defined as a pair (Ω, T_F), where the global transition map $U_F : \Omega \to \Omega$ is given by

$$(U_F\sigma)_v = F(\sigma_{v+u_1}, \ldots, \sigma_{v+u_s}), v \in \mathbb{Z}^2.$$

The function F is called local rule. The space Ω is assumed to be equipped with a (metrizable) Tychonoff topology; it is easily seen that the global transition map U_F introduced above is continuous in this topology.

The 2-D finite CA consists of $m \times n$ cells arranged in m rows and n columns, where each cell takes one of the values of 0, 1 or 2. A configuration of the system is an assignment of states to all the cells. Every configuration determines a next configuration via a linear transition rule that is local in the sense that the state of a cell at time $(t+1)$ depends only on the states of some of its neighbors at time t using modulo 3. For 2D CA nearest neighbors, there are nine cells arranged in a 3×3 matrix centering that particular cell (see [6, 9, 10] for the details). For 2-D CA there are some classic types of neighborhoods, but in this work only we restrict ourselves to the nearest neighborhood and prolonged next nearest neighborhood (briefly, NPNN). So, we can define the $(t+1)^{\text{th}}$ state of the $(i,j)^{\text{th}}$ cell as follows;

$$\begin{aligned} x_{(i,j)}^{(t+1)} =& a x_{(i-1,j)}^{(t)} + b x_{(i,j+1)}^{(t)} + c x_{(i+1,j)}^{(t)} + d x_{(i,j-1)}^{(t)} + e x_{(i-2,j)}^{(t)} + f x_{(i,j+2)}^{(t)} \\ &+ g x_{(i+2,j)}^{(t)} + h x_{(i,j-2)}^{(t)} (\text{mod } 3), \end{aligned} \tag{1}$$

where $a,b,c,d,e,f,g,h \in \mathbb{Z}_3^* = \{1,2\}$. The dependence will be restricted to the case of being zero or nonzero, in other words if the coefficients in (1) equal to 1 or 2, then this case will be assumed to be the same. This approach is adopted in this paper though these cases may be further distinguished. The linear combination of the neighboring cells on which each cell value is dependent is called the rule number of the 2-D CA over the field $\mathbb{Z}_3$.

Regarding the neighborhood of the extreme cells, there exist two approaches.

- A Periodic Boundary CA is the one which the extreme cells are adjacent to each other.
- A Null Boundary CA is the one which the extreme cells are connected to 0-state.

Let us define the 2-D CA generated by the local rule $NPNN$ with PBC (briefly $NPNNP$). In sequel, for simplicity we will assume that $NPNNP \equiv R$. The 2-D CA $U_R : \Omega \to \Omega$

$$\begin{aligned} (U_R x)_{(i,j)}^{(t)} =& a x_{(i-1,j)}^{(t)} + b x_{(i,j+1)}^{(t)} + c x_{(i+1,j)}^{(t)} + d x_{(i,j-1)}^{(t)} + e x_{(i-2,j)}^{(t)} + f x_{(i,j+2)}^{(t)} \\ &+ g x_{(i+2,j)}^{(t)} + h x_{(i,j-2)}^{(t)} (\text{mod } 3), \\ =& x_{(i,j)}^{(t+1)}. \end{aligned} \tag{2}$$

In this paper, we will only consider a 2-D finite CA generated by the rule $NPNN$ with PBC. It is well known that these CAs are discrete dynamical systems formed by a finite two-dimensional array $m \times n$ composed by identical objects called cells. Let $\Phi : M_{m\times n}(\mathbb{Z}_3) \to \mathbb{Z}_3^{mn}$. Φ takes the t^{th} state $[X_t]$ given by

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \longrightarrow (x_{11}, x_{12}, \ldots, x_{1n}, \ldots, x_{m1}, \ldots, x_{mn})^T, \tag{3}$$

where T is the transpose of the matrix. Therefore, the local rules will be assumed to act on $\mathbb{Z}_3^{mn}$ rather than $\Phi : M_{m\times n}(\mathbb{Z}_3)$.
Suppose binary information matrix is $[X_t]_{m\times n}$, of order $m \times n$:

$$[X_t]_{m\times n} = \begin{pmatrix} x_{11}^{(t)} & \ldots & x_{1n}^{(t)} \\ \vdots & \ddots & \vdots \\ x_{m1}^{(t)} & \ldots & x_{mn}^{(t)} \end{pmatrix}$$

is called the configuration of the 2-D finite CA at time t.
From (3), we can define as follows;

$$(T_R)_{mn\times mn} \cdot \begin{pmatrix} x_{11}^{(t)} \\ \vdots \\ x_{1n}^{(t)} \\ \vdots \\ x_{m1}^{(t)} \\ \vdots \\ x_{mn}^{(t)} \end{pmatrix} = \begin{pmatrix} x_{11}^{(t+1)} \\ \vdots \\ x_{1n}^{(t+1)} \\ \vdots \\ x_{m1}^{(t+1)} \\ \vdots \\ x_{mn}^{(t+1)} \end{pmatrix}.$$

where $x_{i,j}^{(t+1)}$ is defined as Eq. (1). i denotes the i^{th} row of the information matrix $[X_t]_{m\times n}$ and j denotes the j^{th} column of the matrix $[X_t]_{m\times n}$. The matrix $(T_R)_{mn\times mn}$ is called the rule matrix with respect to the 2-D finite $\text{CA}_{m\times n}$ with rule $NPNNP$ (see [6] for details). We can briefly show as

$$[T_R]_{mn\times mn} \begin{pmatrix} X_1^T \\ X_2^T \\ X_3^T \\ \vdots \\ X_m^T \end{pmatrix}_{mn\times 1} = \begin{pmatrix} Y_1^T \\ Y_2^T \\ Y_3^T \\ \vdots \\ Y_m^T \end{pmatrix}_{mn\times 1}$$

For example, for $i = j = 3$

$$x_{33}^{(t+1)} = ax_{23}^{(t)} + bx_{34}^{(t)} + cx_{43}^{(t)} + dx_{32}^{(t)} + ex_{13}^{(t)} + fx_{35}^{(t)} + gx_{53}^{(t)} + hx_{31}^{(t)} (\text{mod } 3).$$

Table 1: An information matrix of order 5×5 with PBC.

$x_{(i+1,j+1)}$	$x_{(i+1,j+2)}$	$x_{(i+1,j-2)}$	$x_{(i+1,j-1)}$	$x_{(i+1,j)}$	$x_{(i+1,j+1)}$	$x_{(i+1,j+2)}$	$x_{(i+1,j-2)}$	$x_{(i+1,j-1)}$
$x_{(i+2,j+1)}$	$x_{(i+2,j+2)}$	$x_{(i+2,j-2)}$	$x_{(i+2,j-1)}$	$x_{(i+2,j)}$	$x_{(i+2,j+1)}$	$x_{(i+2,j+2)}$	$x_{(i+2,j-2)}$	$x_{(i+2,j-1)}$
$x_{(i-2,j+1)}$	$x_{(i-2,j+2)}$	$x_{(i-2,j-2)}$	$x_{(i-2,j-1)}$	$x_{(i-2,j)}$	$x_{(i-2,j+1)}$	$x_{(i-2,j+2)}$	$x_{(i-2,j-2)}$	$x_{(i-2,j-1)}$
$x_{(i-1,j+1)}$	$x_{(i-1,j+2)}$	$x_{(i-1,j-2)}$	$x_{(i-1,j-1)}$	$x_{(i-1,j)}$	$x_{(i-1,j+1)}$	$x_{(i-1,j+2)}$	$x_{(i-1,j-2)}$	$x_{(i-1,j-1)}$
$x_{(i,j+1)}$	$x_{(i,j+2)}$	$x_{(i,j-2)}$	$x_{(i,j-1)}$	$x_{(i,j)}$	$x_{(i,j+1)}$	$x_{(i,j+2)}$	$x_{(i,j-2)}$	$x_{(i,j-1)}$
$x_{(i+1,j+1)}$	$x_{(i+1,j+2)}$	$x_{(i+1,j-2)}$	$x_{(i+1,j-1)}$	$x_{(i+1,j)}$	$x_{(i+1,j+1)}$	$x_{(i+1,j+2)}$	$x_{(i+1,j-2)}$	$x_{(i+1,j-1)}$
$x_{(i+2,j+1)}$	$x_{(i+2,j+2)}$	$x_{(i+2,j-2)}$	$x_{(i+2,j-1)}$	$x_{(i+2,j)}$	$x_{(i+2,j+1)}$	$x_{(i+2,j+2)}$	$x_{(i+2,j-2)}$	$x_{(i+2,j-1)}$
$x_{(i-2,j+1)}$	$x_{(i-2,j+2)}$	$x_{(i-2,j-2)}$	$x_{(i-2,j-1)}$	$x_{(i-2,j)}$	$x_{(i-2,j+1)}$	$x_{(i-2,j+2)}$	$x_{(i-2,j-2)}$	$x_{(i-2,j-1)}$
$x_{(i-1,j+1)}$	$x_{(i-1,j+2)}$	$x_{(i-1,j-2)}$	$x_{(i-1,j-1)}$	$x_{(i-1,j)}$	$x_{(i-1,j+1)}$	$x_{(i-1,j+2)}$	$x_{(i-1,j-2)}$	$x_{(i-1,j-1)}$

The conventional method of defining a rule number for a linear rule in 2-D CA with PBC can be explained in Table 1 where $x_{(k,h)} \in \mathbb{Z}_3$ $(k,h \geq 3)$ and $x_{(i,j)}$ is i^{th} row and j^{th} column entry of the information matrix $[X_t]_{5\times 5}$.

3. Rule Matrix of 2-D Finite CA with Rule $NPNNP$

In this section, we determine the rule matrix corresponding to a 2-D finite CA with PBC generated by the local rule $NPNNP$ over the field $\mathbb{Z}_3$. We want to obtain the transformation Ψ such that Ψ operating on the current information matrix (state) $[X]_{m\times n}$ of dimension $m\times n$ generates the next information matrix $[Y]_{m\times n} = [X']_{m\times n}$.

Theorem 1. *Let $a,b,c,d,e,f,g,h \in \mathbb{Z}_3^*$, $m \geq 5$ and $n \geq 5$. Then, the rule matrix of T_R from $\mathbb{Z}_3^{mn}$ to $\mathbb{Z}_3^{mn}$ which takes the t^{th} state $[X_t]$ (as identified in* (3)*) to the* $(t+1)$*- state $[X_{t+1}] = [Y]$ is given by:*

$$(T_R)_{mn\times mn} = \begin{pmatrix} S & cI & gI & 0 & \cdots & \cdots & eI & aI \\ aI & S & cI & gI & \cdots & \cdots & 0 & eI \\ eI & aI & S & cI & gI & \cdots & 0 & 0 \\ 0 & eI & aI & S & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & eI & aI & S & cI & gI \\ gI & \cdots & \cdots & 0 & eI & aI & S & cI \\ cI & gI & \cdots & \cdots & 0 & eI & aI & S \end{pmatrix}_{mn\times mn} \tag{4}$$

where each submatrix is of order $n \times n$, and

$$S_{n\times n} = \begin{pmatrix} 0 & b & f & 0 & 0 & 0 & \cdots & h & d \\ d & 0 & b & f & 0 & 0 & \cdots & 0 & h \\ h & d & 0 & b & f & 0 & \cdots & 0 & 0 \\ 0 & h & d & 0 & b & f & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f & \cdots & \cdots & \cdots & 0 & h & d & 0 & b \\ b & f & \cdots & \cdots & \cdots & 0 & h & d & 0 \end{pmatrix}_{n\times n}. \tag{5}$$

Proof. In order to determine the rule matrix T_R we need to determine the action of T_R on the bases vectors. First, we consider the linear transformation Ψ from $m\times n$ matrix space to itself. Next, we relate the transformation Ψ with rule matrix T_R. Let e_{ij} denote the matrix of size $m\times n$ where the (i,j) position is equal to one and the rest of the entries equal to zero. It is well known that these vectors give the standard basis for this space (see [14]). Given e_{ij}, the image of e_{ij} which is $\Psi(e_{ij})$ is related to the four nearest neighbors and four prolonged next nearest neighbors. $\Psi(e_{ij})$ equals to a linear combination of its eight neighbors in the following way:

$$\Psi(e_{ij}) = f e_{i,j-2} + be_{i,j-1} + de_{i,j+1} + he_{i,j+2} + ge_{i-2,j} + ce_{i-1,j} + ae_{i+1,j} + ee_{i+2,j}$$

with a care on the bordering components of the matrix. Due to the neighboring relations that govern the rule, especially observing the bordering relations as mentioned above, we obtain the following:

$$\begin{aligned}
\Psi(e_{1,1}) =& f e_{1,n-1} + be_{1,n} + de_{1,2} + he_{1,3} + ge_{m-1,1} + ce_{m,1} + ae_{2,1} + ee_{3,1},\\
\Psi(e_{1,2}) =& f e_{1,n} + be_{1,1} + de_{1,3} + he_{1,4} + ge_{m-1,2} + ce_{m,2} + ae_{2,2} + ee_{3,2},\\
\Psi(e_{1,l}) =& f e_{1,l-2} + be_{1,l-1} + de_{1,l+1} + he_{l,j+2}+\\
& ge_{m-1,l} + ce_{m,l} + ae_{2,l} + ee_{3,l}, \quad 3 \leq l \leq n-2\\
\Psi(e_{1,n-1}) =& f e_{1,n-3} + be_{1,n-2} + de_{1,n} + he_{1,1} + ge_{m-1,n-1} + ce_{m,n-1} + ae_{2,n-1} + ee_{3,n-1},\\
\Psi(e_{1,n}) =& f e_{1,n-2} + be_{1,n-1} + de_{1,1} + he_{1,2} + ge_{m-1,n} + ce_{m,n} + ae_{2,n} + ee_{3,n}.
\end{aligned}$$

Now, let us apply to Ψ the second row of information matrix

$$\begin{aligned}
\Psi(e_{2,1}) =& f e_{2,n-1} + be_{2,n} + de_{2,2} + he_{2,3} + ge_{m,1} + ce_{1,1} + ae_{3,1} + ee_{4,1},\\
\Psi(e_{2,2}) =& f e_{2,n} + be_{2,1} + de_{2,3} + he_{2,4} + ge_{m,2} + ce_{1,2} + ae_{3,2} + ee_{4,2},\\
\Psi(e_{2,l}) =& f e_{2,l-2} + be_{2,l-1} + de_{2,l+1} + he_{2,j+2}\\
& + ge_{m,l} + ce_{1,l} + ae_{3,l} + ee_{4,l}, \quad 3 \leq l \leq n-2\\
\Psi(e_{2,n-1}) =& f e_{2,n-3} + be_{2,n-2} + de_{2,n} + he_{2,1} + ge_{m,n-1} + ce_{1,n-1} + ae_{3,n-1} + ee_{4,n-1},\\
\Psi(e_{2,n}) =& f e_{2,n-2} + be_{2,n-1} + de_{2,1} + he_{2,2} + ge_{m,n} + ce_{1,n} + ae_{3,n} + ee_{4,n}.
\end{aligned}$$

For $2 \leq k \leq m-1$, we have

$$\begin{aligned}
\Psi(e_{k,1}) =& f e_{k,n-1} + be_{k,1} + de_{k,2} + he_{k,3} + ge_{k-2,1} + ce_{k-1,1} + ae_{k+1,1} + ee_{k+2,1},\\
\Psi(e_{k,2}) =& f e_{k,n} + be_{k,n} + de_{k,3} + he_{k,4} + ge_{k-2,2} + ce_{k-1,2} + ae_{k+1,2} + ee_{k+2,2},\\
\Psi(e_{k,l}) =& f e_{k,l-2} + be_{k,l-1} + de_{k,l+1} + he_{k,j+2}\\
& + ge_{k-2,l} + ce_{k-1,l} + ae_{k+1,l} + ee_{k+2,l}, \quad 3 \leq l \leq n-2\\
\Psi(e_{k,n-1}) =& f e_{k,n-3} + be_{k,n-2} + de_{k,n} + he_{k,1} + ge_{k-2,n-1} + ce_{k-1,n-1} + ae_{k+1,n-1} + ee_{k+2,n-1},\\
\Psi(e_{k,n}) =& f e_{k,n-2} + be_{k,n-1} + de_{k,1} + he_{k,2} + ge_{k-2,n} + ce_{k-1,n} + ae_{k+1,n} + ee_{k+2,n}.
\end{aligned}$$

$$\begin{aligned}
\Psi(e_{m-1,1}) =& f e_{m-1,n-1} + be_{m-1,n} + de_{m-1,2} + he_{m-1,3} + ge_{m-3,1} + ce_{m-2,1} + ae_{m,1} + ee_{1,1},\\
\Psi(e_{m-1,2}) =& f e_{m-1,n} + be_{m-1,1} + de_{m-1,3} + he_{m-1,4} + ge_{m-3,2} + ce_{m-2,2} + ae_{m,2} + ee_{1,2},\\
\Psi(e_{m-1,l}) =& f e_{m-1,l-2} + be_{m-1,l-1} + de_{m-1,l+1} + he_{m-1,l+2}\\
& + ge_{m-3,l} + ce_{m-2,l} + ae_{m,l} + ee_{1,l}, \quad 3 \leq l \leq n-2\\
\Psi(e_{m-1,n-1}) =& f e_{m-1,n-3} + be_{m-1,n-2} + de_{m-1,n} + he_{m-1,1} + ge_{m-3,n-1} + ce_{m-2,n-1} + ae_{m,n-1} + ee_{1,n-1},\\
\Psi(e_{m-1,n}) =& f e_{m-1,n-2} + be_{m-1,n-1} + de_{m-1,1} + he_{m-1,2} + ge_{m-3,n} + ce_{m-2,n} + ae_{m,n} + ee_{1,n}.
\end{aligned}$$

Finally, we also have

$$\begin{aligned}
\Psi(e_{m,1}) =& f e_{m,n-1} + be_{m,n} + de_{m,2} + he_{m,3} + ge_{m-2,1} + ce_{m-1,1} + ae_{1,1} + ee_{2,1},\\
\Psi(e_{m,2}) =& f e_{m,n} + be_{m,1} + de_{m,3} + he_{m,4} + ge_{m-2,2} + ce_{m-1,2} + ae_{1,2} + ee_{2,2},
\end{aligned}$$

$$\Psi(e_{m,l}) = f e_{m,l-2} + b e_{m,l-1} + d e_{m,l+1} + h e_{m,l+2}$$
$$+ g e_{m-2,l} + c e_{m-1,l} + a e_{1,l} + e e_{2,l}, \quad 3 \le l \le n-2$$
$$\Psi(e_{m,n-1}) = f e_{m,n-3} + b e_{m,n-2} + d e_{m,n} + h e_{m,1} + g e_{m-2,n-1} + c e_{m-1,n-1} + a e_{1,n-1} + e e_{2,n-1},$$
$$\Psi(e_{m,n}) = f e_{m,n-2} + b e_{m,n-1} + d e_{m,1} + h e_{m,2} + g e_{m-2,n} + c e_{m-1,n} + a e_{1,n} + e e_{2,n}.$$

Now, let E_i be the column vector in $\mathbb{Z}_3^{mn}$ where all entries equal to zero expect the entry positioned at i which equals to one. These vectors also give the standard basis of the space $\mathbb{Z}_3^{mn}$. The transition from the matrix space basis to the standard basis of $\mathbb{Z}_3^{mn}$ is given by

$$\Phi(e_{ij}) = E_{(i-1)n+j},$$

where $1 \le i \le m$ and $1 \le j \le n$.

In order to determine the columns of the rule matrix T_R we maintain the following method:

$$\begin{aligned}
\Phi(\Psi(e_{ij})) =& \Phi(f e_{i,j-2} + b e_{i,j-1} + d e_{i,j+1} + h e_{i,j+2} + g e_{i-2,j} + c e_{i-1,j} + a e_{i+1,j} + e e_{i+2,j}) \\
=& f\Phi(e_{i,j-2}) + b\Phi(e_{i,j-1}) + d\Phi(e_{i,j+1}) + h\Phi(e_{i,j+2}) \\
&+ g\Phi(e_{i-2,j}) + c\Phi(e_{i-1,j}) + a\Phi(e_{i+1,j}) + e\Phi(e_{i+2,j}) \\
=& f E_{(i-1)n+j-2} + bE_{(i-1)n+j-1} + dE_{(i-1)n+j+1} + hE_{(i-1)n+j+2} \\
&+ gE_{(i-3)n+j} + cE_{(i-2)n+j} + aE_{in+j}) + eE_{(i+1)n+j}
\end{aligned} \tag{6}$$

which gives the $((i-1)n + j)^{th}$ column of the rule matrix T_R. For instance,

$$dE_2 + hE_3 + f E_{n-1} + bE_n + aE_{n+1} + eE_{2n+1} + gE_{(m-2)n+1} + cE_{(m-1)n+2}$$

which gives the first column of the rule matrix T_R. Further,

$$bE_1 + dE_3 + hE_4 + f E_n + aE_{n+2} + eE_{2n+2} + gE_{(m-2)n+2} + cE_{(m-1)n+2}$$

which gives the second column of the rule matrix T_R.
Similarly one can obtain the last column of the rule matrix as following:

$$aE_n + eE_{2n} + gE_{(m-2)n} + cE_{(m-1)n} + dE_{(m-1)n+1} + hE_{(m-1)n+2} + f E_{mn-2} + bE_{mn-1}.$$

Inductively, we can similarly obtain the rest of the columns. □

Example 1. *If we take* $m = 5$ *and* $n = 5$*, then we get the representation matrix* T_R *of order* 25×25*. We consider a configuration of size* 5×5 *with PBC.*

Table 2: An information matrix of order 5×5 with PBC.

x_{44}	x_{45}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{41}	x_{42}
x_{54}	x_{55}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{51}	x_{52}
x_{14}	x_{15}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{11}	x_{12}
x_{24}	x_{25}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{21}	x_{22}
x_{34}	x_{35}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	$x_{31)}$	x_{32}
x_{44}	x_{45}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{41}	x_{42}
x_{54}	x_{55}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{51}	x_{52}
x_{14}	x_{15}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{11}	x_{12}
x_{24}	x_{25}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{21}	x_{22}

Now, we apply the local rule in Eq. (1) *to all cells of the first row of information matrix* 5×5*, then we obtain the first block row of rule matrix* T_R *and next apply the rule R to all the cells of the second row of information matrix* 5×5*, we obtain the second block row of rule matrix* T_R*. Similarly, we can obtain the other block rows of rule matrix* T_R*.*

From Theorem 1, one can get the matrix as following:

$$(T_R)_{25\times 25} = \begin{pmatrix} S & cI_5 & gI_5 & eI_5 & aI_5 \\ aI_5 & S & cI_5 & gI_5 & eI_5 \\ eI_5 & aI_5 & S & Ic_5 & gI_5 \\ gI_5 & eI_5 & aI_5 & S & cI_5 \\ cI_5 & gI_5 & eI_5 & aI_5 & S \end{pmatrix},$$

where each submatrix is of order 5×5*, and* $S_{5\times 5} = \begin{pmatrix} 0 & b & f & h & d \\ d & 0 & b & f & h \\ h & d & 0 & b & f \\ f & h & d & 0 & b \\ b & f & h & d & 0 \end{pmatrix}_{5\times 5}$.

4. Characterization of 2-D Finite CA with Rule $NPNNP$

The dimension of the kernel of a map gives a clue to draw the state transition diagram (see [6, 10, 12]). In order to determine dimension of the kernel of a 2-D CA, we need to study the rank of the rule matrix $(T_R)_{mn\times mn}$. The following theorem presents an algorithm for computing the rank:

Let T_i denote the ith row and $T_i[j]$ denote the j the entry of the ith row of matrix T respectively. By definition, we have

$$\begin{aligned} T_1 =&[S, cI, gI, 0, 0, \dots, 0, eI, aI] \in M_{n\times n}(\mathbb{Z}_3)^m \\ T_2 =&[aI, S, cI, gI, 0, 0, \dots, 0, eI] \in M_{n\times n}(\mathbb{Z}_3)^m \\ T_3 =&[eI, aI, S, cI, gI, 0, 0, \dots, 0] \in M_{n\times n}(\mathbb{Z}_3)^m \\ T_4 =&[gI, 0, 0, 0, \dots, eI, aI, S, cI] \in M_{n\times n}(\mathbb{Z}_3)^m \\ T_5 =&[cI, gI, 0, 0, \dots, 0, eI, aI, S] \in M_{n\times n}(\mathbb{Z}_3)^m. \end{aligned}$$

Define the σ map as follows:

$$\begin{aligned} \sigma :& M_{n\times n}(\mathbb{Z}_3)^m \to M_{n\times n}(\mathbb{Z}_3)^m \\ & \sigma([A_1, A_2, A_3, \dots, A_{m-1}, A_m]) = [0_n, A_1, A_2, A_3, \dots, A_{m-1}], \end{aligned}$$

where 0_n represents the zero square matrix of order n. Further, if $A = [A_1, A_2, A_3, \dots, A_{m-1}, A_m]$, then $A[i] = A_i$ represents the ith entry of A. Further, $B[A_1, A_2, \dots, A_m] = [BA_1, BA_2, \dots, BA_m]$.

Theorem 2. *Let the rule matrix* $(T_R)_{mn\times mn}$ *be given in Theorem 1 and* $m \geq 5$*. Assume that*

$$T_1^{(1)} = T_1,$$

$$\begin{aligned}
T_1^{(k+1)} =& -\frac{1}{e}T_1^{(k)}[k]\sigma^{(k-1)}(T_3)+T_1^{(k)} \textit{ for } 1\le k\le m-4,\\
T_2^{(1)} =& T_2,\\
T_2^{(k+1)} =& -\frac{1}{e}T_2^{(k)}[k]\sigma^{(k-1)}(T_3)+T_2^{(k)} \textit{ for } 1\le k\le m-4\\
T_4^{(1)} =& T_4,\\
T_4^{(k+1)} =& -\frac{1}{e}T_4^{(k)}[k]\sigma^{(k-1)}(T_3)+T_4^{(k)} \textit{ for } 1\le k\le m-4,\\
T_5^{(1)} =& T_5,\\
T_5^{(k+1)} =& -\frac{1}{e}T_5^{(k)}[k]\sigma^{(k-1)}(T_3)+T_5^{(k)} \textit{ for } 1\le k\le m-4.
\end{aligned}$$

If

$$B=\begin{pmatrix} T_1^{(m-3)}[m-3] & T_1^{(m-3)}[m-2] & T_1^{(m-3)}[m-1] & T_1^{(m-3)}[m]\\ T_2^{(m-3)}[m-3] & T_2^{(m-3)}[m-2] & T_2^{(m-3)}[m-1] & T_2^{(m-3)}[m]\\ T_4^{(m-3)}[m-3] & T_4^{(m-3)}[m-2] & T_4^{(m-3)}[m-1] & T_4^{(m-3)}[m]\\ T_5^{(m-3)}[m-3] & T_5^{(m-3)}[m-2] & T_5^{(m-3)}[m-1] & T_5^{(m-3)}[m] \end{pmatrix}$$

is a 4×4 *block matrix consisting of square sub matrices each of order n, then*

$$rank((T_R)_{mn\times mn})=(m-4)\cdot n+rank(B).$$

Proof. We apply induction on m. First, we observe that the submatrix consisting of all rows except the first, the second and last two rows is in the upper triangular form and it has a full rank which is $(m-4)\cdot n$. Now, if we multiply the third row T_3 by $-\frac{1}{e}T_1^{(1)}[1]T_3[1]$ and add this to $T_1^{(1)}[1]$, then the first entry of the new first row $T_1^{(2)}$ becomes zero. So we replace the first row by $T_1^{(2)}=-\frac{1}{e}T_1^{(1)}[1]\sigma^{(0)}(T_3)+T_1^{(1)}$. Next, if we multiply the fourth row $\sigma^{(1)}(T_3)$ by $-\frac{1}{e}T_1^{(1)}[2]\sigma^{(1)}(T_3)[2]$ and add it to $T_1^{(1)}[2]$ the second entry of the new first row $T_1^{(3)}$ becomes zero. Inductively, after $m-4$ steps, the only nonzero entries of $T_1^{(m-3)}$ are $T_1^{(m-3)}[m-3]$, $T_1^{(m-3)}[m-2]$, $T_1^{(m-3)}[m-1]$ and $T_1^{(m-3)}[m]$. Similarly, by applying elementary row operations to the second row, the only non zero entries of the second row are $T_2^{(m-3)}[m-3]$, $T_2^{(m-3)}[m-2]$, $T_2^{(m-3)}[m-1]$ and $T_2^{(m-3)}[m]$ (see (7)). Same procedure is applied to the last two rows. After applying the elementary row operations mentioned above, we get

$$T'_R = \left(\begin{array}{cccccc|cccc} 0 & 0 & 0 & 0 & \cdots & 0 & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & & B & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & & & & \\ \hline eI & aI & S & cI & gI & \cdots & 0 & 0 & 0 & 0 \\ 0 & eI & aI & S & \cdots & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\ 0 & \cdots & 0 & eI & aI & S & cI & gI & 0 & 0 \\ 0 & \cdots & \cdots & 0 & eI & aI & S & cI & gI & 0 \\ 0 & 0 & \cdots & \cdots & 0 & eI & aI & S & cI & gI \end{array}\right)$$

$$= \left(\begin{array}{c|c} 0 & B \\ \hline C & D \end{array}\right). \tag{7}$$

The left lower block of C in (7) has full rank $(m-4)\cdot n$. The rank of the matrix in (7) equals to $(m-4)\cdot n+$ rank(B). □

A straightforward corollary which gives a lower and an upper bound for the rank of the rule matrix $(T_R)_{mn\times mn}$ is presented below.

Corollary 1. *Let the rule matrix $(T_R)_{mn\times mn}$ be defined as in Theorem 1. Then,*

$$(m-4)\cdot n \leq rank((T_R)_{mn\times mn}) \leq m\cdot n.$$

5. Examples

In this section, we give some applications of the main theorem. We also give an algorithm and apply it to a 2-D CA with larger order.

Example 2. *Let $n = 5$, $m = 5$ and $a = 2, b = 2, c = 1, d = 1, f = 2, h = 1, e = 1, g = 2$. Then we get*

$$(T_R)_{25\times 25} = \begin{pmatrix} S & I_5 & 2I_5 & I_5 & 2I_5 \\ 2I_5 & S & I_5 & 2I_5 & I_5 \\ I_5 & 2I_5 & S & I_5 & 2I_5 \\ 2I_5 & I_5 & 2I_5 & S & I_5 \\ I_5 & 2I_5 & I_5 & 2I_5 & S \end{pmatrix},$$

$$S = \begin{pmatrix} 0 & 2 & 2 & 1 & 1 \\ 1 & 0 & 2 & 2 & 1 \\ 1 & 1 & 0 & 2 & 2 \\ 2 & 1 & 1 & 0 & 2 \\ 2 & 2 & 1 & 1 & 0 \end{pmatrix}.$$

Direct computation of the rank of $(T_R)_{25\times 25}$ *gives* 20*. On the other hand, if we apply Theorem* (2)*, then we proceed in the following way:*

$$\begin{aligned}
T_1^{(1)} =& T_1 = [S, I_5, 2I_5, I_5, 2I_5],\\
T_1^{(2)} =& -\frac{1}{e}T_1^{(1)}[1]\sigma^{(0)}(T_3) + T_1^{(1)}\\
=& -S[I_5, 2I_5, S, I_5, 2I_5] + [S, I_5, 2I_5, I_5, 2I_5]\\
=& [0_5, -2S + I_5, -S^2 + 2I_5, -S + I_5, -2S + 2I_5],
\end{aligned}$$

$$\begin{aligned}
T_2^{(1)} =& T_2 = [2I_5, S, I_5, 2I_5, I_5],\\
T_2^{(2)} =& -\frac{1}{e}T_2^{(1)}[1]\sigma^{(0)}(T_3) + T_2^{(1)}\\
=& -(2I_5)[I_5, 2I_5, S, I_5, 2I_5] + [2I_5, S, I_5, 2I_5, I_5] = [0_5, S - I_5, -2S + I_5, 0_5, 0_5],
\end{aligned}$$

$$\begin{aligned}
T_4^{(1)} =& T_4 = [2I_5, I_5, 2I_5, S, I_5],\\
T_4^{(2)} =& -\frac{1}{e}T_4^{(1)}[1]\sigma^{(0)}(T_3) + T_4^{(1)}\\
=& -2I_5[I_5, 2I_5, S, I_5, 2I_5] + [2I_5, I_5, 2I_5, S, I_5] = [0_5, 0_5, -2S + 2I_5, S - 2I_5, 0_5],
\end{aligned}$$

$$\begin{aligned}
T_5^{(1)} =& T_5 = [I_5, 2I_5, I_5, 2I_5, S],\\
T_5^{(2)} =& -\frac{1}{e}T_5^{(1)}[1]\sigma^{(0)}(T_3) + T_5^{(1)}\\
=& -(I_5)[I_5, 2I_5, S, I_5, 2I_5] + [I_5, 2I_5, I_5, 2I_5, S] = [0_5, 0_5, -S + I_5, I_5, S - 2I_5],
\end{aligned}$$

Thus,

$$B = \begin{pmatrix} T_1^{(2)}[2] & T_1^{(2)}[3] & T_1^{(2)}[4] & T_1^{(2)}[5] \\ T_2^{(2)}[2] & T_2^{(2)}[3] & T_2^{(2)}[4] & T_2^{(2)}[5] \\ T_4^{(2)}[2] & T_4^{(2)}[3] & T_4^{(2)}[4] & T_4^{(2)}[5] \\ T_5^{(2)}[2] & T_5^{(2)}[3] & T_5^{(2)}[4] & T_5^{(2)}[5] \end{pmatrix} = \begin{pmatrix} -2S + I_5 & -S^2 + 2I_5 & -S + I_5 & -2S + 2I_5 \\ S - I_5 & -2S + I_5 & 0_5 & 0_5 \\ 0_5 & -2S + 2I_5 & S - 2I_5 & 0_5 \\ 0_5 & -S + I_5 & I_5 & S - 2I_5 \end{pmatrix}.$$

Hence, $rank(B) = 15$*. Therefore,* $rank((T_R)_{36\times 36}) = (m-4)\cdot n + rank(B) = (5-4)\cdot 5 + 15 = 20$*. As a result, the CA is irreversible.*

Example 3. *Let* $n = 6$*,* $m = 6$ *and* $a = 2, b = 2, c = 1, d = 1, f = 2, h = 1, e = 2, g = 1$*. Then we have,*

$$(T_R)_{36\times 36} = \begin{pmatrix} S & I_6 & I_6 & 0_6 & 2I_6 & 2I_6 \\ 2I_6 & S & I_6 & I_6 & 0_6 & 2I_6 \\ 2I_6 & 2I_6 & S & I_6 & I_6 & 0_6 \\ 0_6 & 2I_6 & 2I_6 & S & I_6 & I_6 \\ I_6 & 0_6 & 2I_6 & 2I_6 & S & I_6 \\ I_6 & I_6 & 0_6 & 2I_6 & 2I_6 & S \end{pmatrix},$$

$$S = \begin{pmatrix} 0 & 2 & 2 & 0 & 1 & 1 \\ 1 & 0 & 2 & 2 & 0 & 1 \\ 1 & 1 & 0 & 2 & 2 & 0 \\ 0 & 1 & 1 & 0 & 2 & 2 \\ 2 & 0 & 1 & 1 & 0 & 2 \\ 2 & 2 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Direct computation of the rank of $(T_R)_{36\times 36}$ *gives* 18. *On the other hand, if we apply Theorem* (2), *then we proceed in the following way:*

$$\begin{aligned}
T_1^{(1)} =& T_1 = [S, I_6, I_6, 0_6, 2I_6, 2I_6],\\
T_1^{(2)} =& -\frac{1}{e} T_1^{(1)}[1]\sigma^{(0)}(T_3) + T_1^{(1)}\\
=& -2S[2I_6, 2I_6, S, I_6, I_6, 0_6] + [S, I_6, I_6, 0_6, 2I_6, 2I_6]\\
=& [0_6, -S + I_6, -2S^2 + I_6, -2S, -2S + 2I_6, 2I_6],\\
T_1^{(3)} =& -\frac{1}{e} T_1^{(2)}[2]\sigma^{(1)}(T_3) + T_1^{(2)}\\
=& -2(-S + I_6)[0_6, 2I_6, 2I_6, S, I_6, I_6] + [0_6, -S + I_6, -2S^2 + I_6, -2S, -2S + 2I_6, 2I_6]\\
=& [0_6, 0_6, -2S^2 + S, 2S^2 - S, 0_6, 2S],
\end{aligned}$$

$$\begin{aligned}
T_2^{(1)} =& T_2 = [2I_6, S, I_6, I_6, 0_6, 2I_6],\\
T_2^{(2)} =& -\frac{1}{e} T_2^{(1)}[1]\sigma^{(0)}(T_3) + T_2^{(1)}\\
=& -2(2I_6)[2I_6, 2I_6, S, I_6, I_6, 0_6] + [2I_6, S, I_6, I_6, 0_6, 2I_6]\\
=& [0_6, S - 2I_6, -S + I_6, 0_6, -I_6, 2I_6],\\
T_2^{(3)} =& -\frac{1}{e} T_2^{(2)}[2]\sigma^{(1)}(T_3) + T_2^{(2)}\\
=& -2(S - 2I_6)[0_6, 2I_6, 2I_6, S, I_6, I_6] + [0_6, S - 2I_6, -S + I_6, 0_6, -I_6, 2I_6]\\
=& [0_6, 0_6, -2S, -2S^2 + S, -2S, -2S],
\end{aligned}$$

$$\begin{aligned}
T_4^{(1)} =& T_4 = [I_6, 0_6, 2I_6, 2I_6, S, I_6],\\
T_4^{(2)} =& -\frac{1}{e} T_4^{(1)}[1]\sigma^{(0)}(T_3) + T_4^{(1)}\\
=& -2I_6[2I_6, 2I_6, S, I_6, I_6, 0_6] + [I_6, 0_6, 2I_6, 2I_6, S, I_6]\\
=& [0_6, -I_6, -2S + 2I_6, 0_6, S - 2I_6, I_6],\\
T_4^{(3)} =& -\frac{1}{e} T_4^{(2)}[2]\sigma^{(1)}(T_3) + T_4^{(2)}\\
=& -2(-I_6)[0_6, 2I_6, 2I_6, S, I_6, I_6] + [0_6, -I_6, -2S + 2I_6, 0_6, S - 2I_6, I_6]\\
=& [0_6, 0_6, -2S, 2S, S, 0_6],
\end{aligned}$$

$$
\begin{aligned}
T_5^{(1)} =& T_5 = [I_6, I_6, 0_6, 2I_6, 2I_6, S],\\
T_5^{(2)} =& -\frac{1}{e} T_5^{(1)}[1]\sigma^{(0)}(T_3) + T_5^{(1)}\\
=& -2(I_6)[2I_6, 2I_6, S, I_6, I_6, 0_6] + [I_6, I_6, 0_6, 2I_6, 2I_6, S] = [0_6, 0_6, -2S, 0_6, 0_6, S],
\end{aligned}
$$

$$
\begin{aligned}
T_5^{(3)} =& -\frac{1}{e} T_5^{(2)}[2]\sigma^{(1)}(T_3) + T_5^{(2)}\\
=& -2(0_6)[0_6, 2I_6, 2I_6, S, I_6, I_6] + [0_6, 0_6, -2S, 0_6, 0_6, S]\\
=& [0_6, 0_6, -2S, 0_6, 0_6, S].
\end{aligned}
$$

Thus we have

$$
B = \begin{pmatrix} T_1^{(3)}[3] & T_1^{(3)}[4] & T_1^{(3)}[5] & T_1^{(3)}[6] \\ T_2^{(3)}[3] & T_2^{(3)}[4] & T_2^{(3)}[5] & T_2^{(3)}[6] \\ T_4^{(3)}[3] & T_4^{(3)}[4] & T_4^{(3)}[5] & T_4^{(3)}[6] \\ T_5^{(3)}[3] & T_5^{(3)}[4] & T_5^{(3)}[5] & T_5^{(3)}[6] \end{pmatrix} = \begin{pmatrix} -2S^2 + S & 2S^2 - S & 0_6 & 2S \\ -2S & -2S^2 + S & -2S & -2S \\ -2S & 2S & S & 0_6 \\ -2S & 0_6 & 0_6 & S \end{pmatrix}.
$$

Hence, rank$(B) = 6$. *Therefore, rank*$((T_R)_{36\times 36}) = (m-4)\cdot n + rank(B) = (6-4)\cdot 6 + 6 = 18$. *As a result, the CA is irreversible.*

5.1. An Algorithm for Computing The Rank of $(T_R)_{mn\times mn}$

Now we can summarize the method introduced above for computing the rank of the rule matrix as follows:

Step 1. Determine respectively the first two rows T_1 and T_2 and the last two rows T_4 and T_5 which consists of block of matrices. Set $T_1^{(1)} = T_1$, $T_2^{(1)} = T_2$, $T_4^{(1)} = T_4$ and $T_5^{(1)} = T_5$.

Step 2. If $m > n+1$, compute the characteristic polynomial of S by applying Cayley-Hamilton theorem [14].

Step 3. For $1 \le k \le m-4$, compute

$$
\begin{aligned}
T_1^{(k+1)} =& -\frac{1}{e} T_1^{(k)}[k]\sigma^{(k-1)}(T_3) + T_1^{(k)},\\
T_2^{(k+1)} =& -\frac{1}{e} T_2^{(k)}[k]\sigma^{(k-1)}(T_3) + T_2^{(k)},\\
T_4^{(k+1)} =& -\frac{1}{e} T_4^{(k)}[k]\sigma^{(k-1)}(T_3) + T_4^{(k)},\\
T_5^{(k+1)} =& -\frac{1}{e} T_5^{(k)}[k]\sigma^{(k-1)}(T_3) + T_5^{(k)}.
\end{aligned}
$$

Hence, determine the matrices $T_1^{(m-3)}[m-3], \ldots, T_1^{(m-3)}[m], T_2^{(m-3)}[m-3], \ldots, T_2^{(m-3)}[m], T_4^{(m-3)}[m-3], \ldots, T_4^{(m-3)}[m]$ and $T_5^{(m-3)}[m-3], \ldots, T_5^{(m-3)}[m]$.

If $m > n+1$, in every iteration while computing $T_i^{(k)}$ $(i = 1, 2, 4, 5)$, since S is a matrix, arithmetic can be carried out modulo the characteristic polynomial of S which saves reasonable time.

Step 4. Compute the rank of

$$B = \begin{pmatrix} T_1^{(m-3)}[m-3] & T_1^{(m-3)}[m-2] & T_1^{(m-3)}[m-1] & T_1^{(m-3)}[m] \\ T_2^{(m-3)}[m-3] & T_2^{(m-3)}[m-2] & T_2^{(m-3)}[m-1] & T_2^{(m-3)}[m] \\ T_4^{(m-3)}[m-3] & T_4^{(m-3)}[m-2] & T_4^{(m-3)}[m-1] & T_4^{(m-3)}[m] \\ T_5^{(m-3)}[m-3] & T_5^{(m-3)}[m-2] & T_5^{(m-3)}[m-1] & T_5^{(m-3)}[m] \end{pmatrix}$$

Therefore, $rank((T_R)_{mn\times mn}) = (m-4)\cdot n + rank(B)$.

Here we give an example that makes use of the algorithm introduced above.

Example 4. *Let $n = 6$, $m = 1000$ and $a = 2, b = 1, c = 1, d = 1, f = 2, h = 2, e = 2, g = 1$. Hence, the representation matrix of this two dimensional cellular automata is of size 6000×6000. In order to compute its rank, we apply the algorithm:*

Step 1. We determine respectively the first two rows T_1 and T_2 and the last two rows T_4 and T_5 which consists of block of matrices. Set $T_1^{(1)} = T_1$, $T_2^{(1)} = T_2$, $T_4^{(1)} = T_4$ and $T_5^{(1)} = T_5$.

Step 2. Since $m > n+1$, the characteristic polynomial of S $charpoly(S) = S^6 + S^3$. So, in each computational step we apply the Cayley-Hamilton theorem [14] to reduce the algebra dramatically.

Step 3. By applying the reduction formula we get the rows of matrix B which are given in the next step.

Step 4. So

$$B = \begin{pmatrix} 2S+2S^2 & 2S^2+1 & 2S^2+S^3 & S^4+S+2S^2+2 \\ S^3+2+2S^2 & 2S+2S^2 & S^3+2S^4+2S+1 & 2S^3+S^4+S \\ 2S^3+S^4+S & S^4+S+2S^2+2 & S^4+2S+S^2+2S^5 & 2S^3+S^4+S+1 \\ S^3+2S^4+2S+1 & 2S^2+S^3 & 2S+S^5+2S^2+2 & S^4+2S+S^2+2S^5 \end{pmatrix}.$$

Therefore, $rank(B) = 18$ and
$rank((T_R)_{mn\times mn}) = (m-4)\cdot n + rank(B) = (996)\cdot 6 + 18 = 5994$.

P.S. In Example 3, running these four algorithms for $m = 1000$ and $n = 6$ took approximately 522 seconds in an ordinary computer (Intel Core DuoCPU, 1.6GHZ, 1GB RAM).

6. Bit Error Correcting-Detecting Code Based On Nonsingular Transition Matrix 2D Cellular Automata

One Dimensional Cellular Automata based bit error correcting codes (ECC) proposed by Chaudhuri et al. [8] in 1994 (see [7]). They studied CA based ECC over binary fields. In this section, we extend CA based bit error correcting code (CA-ECC) over ternary fields. CA based bit error correcting codes have two main advantages if compared to standard linear coding. The first advantage is that if errors occur only in the information or in check bits, decoding is very simple and fast. Even in the case where the errors are in both information part and check part the decoding steps are less than the classical syndrome decoding method. The second advantage is that the structure of cellular automata allows parallel computing which leads to faster computations and simple hardware implementations.

Now, we present some basics of error correcting codes. Further and more detailed information on this topic can be found in [15]. It is well-known that $V = \mathbb{Z}_3^n$ is a $\mathbb{Z}_3$-vector space.

Definition 1. *A subspace C of V is called a linear code of length n. The matrix with rows consisting of the basis of C is called a generator matrix of the code. The elements of C are called codewords.*

Error correcting codes are applied in digital media. An information is encoded in order to be able to detect errors or even correct them. An information of length k is encoded to an n tuple an regarded as an element (codeword) of C. After the transmission process if a linear code can detect and correct one error then the code is called one error correcting code. The number of nonzero entries of a vector v in V is called the Hamming weight of v. The smallest nonzero weight among all codewords of a linear code C is called the minimum Hamming weight of C. A linear code C of length n, dimension k and minimum Hamming weight d is represented by $[n, k, d]$. For a detailed information the reader can refer to [15]. These three parameters play an important role in error correcting codes. Especially the minimum Hamming weight of a code is is very crucial.

Theorem 3. *[15] If C is a linear code with minimum Hamming weight d, then C can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

Let T be $n \times n$ nonsingular transition matrix of 2D-cellular automata with $NPNN$. Suppose that there exists $1 \leq k \leq n$, $k \in \mathbb{Z}^+$ such that $G = \left[I_n | T^k \right]$ (I_n, $n \times n$ identity matrix) generates a linear code that corrects at least one error.

6.1. Encoding

Let $I = \left(i_1, i_2, \ldots, i_n \right) \in \mathbb{Z}_3^n$ be information bits, where n is the rank of nonsingular transition matrix. Then $CW = \left(I, T^k [I] \right) = \left(i_1, i_2, \ldots, i_n, c_{n+1}, c_{n+2}, \ldots, c_{2n} \right)$ is a code word ($C = T^k [I] = \left(c_{n+1}, c_{n+2}, \ldots, c_{2n} \right)$ is the check bits).

Now, we give a decoding scheme for CA based single bit error correcting code which is an extension from binary [8] to ternary case.

6.2. Decoding

Let $CW' = \left(I', T^k [I] \right) = \left(i'_1, i'_2, \ldots, i'_n, c'_{n+1}, c'_{n+2}, \ldots, c'_{2n} \right)$ be received word, then we compute syndrome vector as follows:

$S = 2T^k \left[I' \right] \oplus C'$.

Case 1. Let all errors occur in the information bits, then the syndrome of the information part is

$S_n = 2T^k \left[I'\right] \oplus C'$ and $T^{-k}\left[S_n\right]$ is error vector of information part. In this case the syndrome vector of check part S_c is all zero vector. Then, the error vector is

$E = \left(T^{-k}\left[S_n\right], S_c\right) = \left(T^{-k}\left[S_n\right], 00\ldots 0\right)$

Case 2. Let all errors occur in the check part, then, the syndrome of check part (at the same time error vector of check part) is

$S_c = T^k\left[I'\right] \oplus 2C'$.In this case the syndrome vector of information part S_n is all zero vector, then the error vector is

$E = \left(T^{-k}\left[S_n\right], S_c\right) = \left(00\ldots 0, T^k\left[I'\right] \oplus 2C'\right)$.

Case 3. Both information and check parts are in error. Then, we take S_c all possible error vectors and execute decoding Cases 1 and 2.

Now we give an example that illustrates the algorithm proposed above:

Example 5. *Let us take the following transition matrix* $(T_R)_{25\times 25} = \begin{pmatrix} S & I_5 & 2I_5 & I_5 & 2I_5 \\ 2I_5 & S & I_5 & 2I_5 & I_5 \\ I_5 & 2I_5 & S & I_5 & 2I_5 \\ 2I_5 & I_5 & 2I_5 & S & I_5 \\ I_5 & 2I_5 & I_5 & 2I_5 & S \end{pmatrix}$,

where $S = \begin{pmatrix} 0 & 1 & 2 & 1 & 1 \\ 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 & 0 \end{pmatrix}$. *For* $k = 2$, $G = \left[I_{25} | (T_R)^2_{25\times 25}\right]$ *generates*

$C \to [50, 25, 10]_3 - code$, *and* $d(C) = 10$ *then* C *can correct* 4 *errors.*

Let $I = 1111111111222222222222222 \in \mathbb{Z}_3^{25}$ *be a word. Then we generate check bits*

$$C = (T_R)^2_{25\times 25}[I] = 1111100000000001111100000.$$

$$CW = \left(I, (T_R)^2_{25\times 25}[I]\right) = 11111111112222222222222221111100000000001111100000 \in \mathbb{Z}_3^{50}.$$

Case 1. Four errors occur in the information bits. Let the received word be

$$\begin{aligned} CW' =& \widehat{0000}111111222222222222222111110000000000111110000 0 = \left(I'|C'\right). \\ S =& 2T^2\left[I'\right] \oplus C' \\ =& 2020011012112112212122021 \oplus 1111100000000001111100000 \\ =& 0101111012112110020222021. \\ S_{25} =& S \oplus S_c = 0101111012112110020222021. \end{aligned}$$

$$I_e = T^{-2}\left[S_{25}\right] = 1111000000000000000000000.$$
$$I = I' \oplus I_e = \widehat{0000}111111222222222222222 \oplus 1111000000000000000000000$$
$$= 1111111111222222222222222.$$
$$C = C' = 1111100000000001111100000$$
$$E = 11110000000000000000000000000000000000000000000000.$$

Case 2. Four errors occur in the check part. Let the received word be

$$CW' = 1111111111222222222222222\widehat{2222}100000000001111100000 = \left(I'|C'\right).$$
$$S = T^2\left[I'\right] \oplus 2C' = 1111100000000001111100000 \oplus 1111200000000002222200000$$
$$= 2222000000000000000000000.$$
$$S_{25} = 0000000000000000000000000$$

and

$$S_c = 2222000000000000000000000.$$
$$I_e = T^{-2}\left[S_{25}\right] = 0000000000000000000000000.$$
$$C_e = S_c = 2222000000000000000000000.$$
$$C = C' \oplus C_e = \widehat{2222}100000000001111100000 \oplus 2222000000000000000000000$$
$$= 1111100000000001111100000.$$
$$E = 00000000000000000000000002222000000000000000000000.$$

Case 3. Both information and check part are in error. Let the received word be

$$CW' = 1111\widehat{00}11112222222222222221111100002\widehat{2}00001111100000.$$
$$S = 2T^2\left[I'\right] \oplus C' = 2\,(1021002221101201102020122) \oplus 1111100002200001111100000$$
$$= 0120101111102100012110211.$$

After first and second cycles fail to detect errors take S_c all possible error vectors in the check part. Consider the case when

$$S_c = 0000000001100000000000000.$$
$$S_{25} = S \oplus S_c = 0120101112202100012110211.$$
$$I_e = T^{-2}\left[S_{25}\right] = 0000110000000000000000000.$$
$$I = I' \oplus I_e = 1111\widehat{00}1111222222222222222 \oplus 0000110000000000000000000$$
$$= 1111111111222222222222222.$$
$$C = C' \oplus C_e = 111110000\widehat{22}00001111100000 \oplus 0000000001100000000000000$$
$$= 1111100000000001111100000.$$
$$E = 00001100000000000000000000000000001100000000000000.$$

In the Case-3, where errors occur in the both information and check part, firstly the check part is corrected by classical syndrome decoding which require in total $\sum_{i=0}^{3}\binom{25}{i}$ *steps. Secondly the information part is corrected by applying Case-1 which requires only one step. On the other hand if the classical decoding method is used,* $\sum_{i=0}^{4}\binom{50}{i}$ *steps are required. So as* n *is larger the advantage of using CA becomes clearer.*

7. Conclusion

2-D finite CA with the rule $NPNNP$ has been defined on the field $\mathbb{Z}_3$. The rule matrix of the 2-D finite CA has been obtained. Characterization of 2-D finite CA with the rule $NPNNP$ has been investigated. Properties of the 2-D finite CA over other fields (see [3]) remain to be of great research interest. In this paper we focuss on algebraic representation for a novel 2-D CA over ternary fields. We give some examples of the algorithms established here. Finally, we present an application to error correcting codes and discuss the advantages of using CA. Some other applications of this family of codes especially to cryptography, simulation of natural phenomena, pseudo-random number generators, etc. is waiting to be explored.

ACKNOWLEDGEMENTS The work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK), project number: 110T713.

References

[1] H. Akın. On the directional entropy of $\mathbb{Z}^2$ -actions generated by additive cellular automata, Applied Mathematics and Computation. **170** (1), 339-346. 2005.

[2] H. Akın. On the topological directional entropy, Journal of Computational and Applied Mathematics. **225** (2), 459-466. 2009.

[3] H. Akın and I. Siap. On cellular automata over Galois rings, Information Processing Letters. **103** (1), 24-27. 2007.

[4] G. Alvarez, L. H. Encinas, and A. Martin del Rey. A multisecret sharing scheme for color images based on cellular automata, Information Sciences. **178**, 4382-4395. 2008.

[5] S.R. Blackburn, S. Murphy, and K.G. Peterson. Comments on theory and applications of cellular automata in cryptography, IEEE Transactions on Computers. **46**, 637-638. 1997.

[6] P. Chattopdhyay, P. P. Choudhury, and K. Dihidar. Characterisation of a particular hybrid transformation of two-dimensional cellular automata, Computers Mathematics with Applications. **38**, 207-216. 1999.

[7] S.J. Cho, U.S. Choi, and S.H. Heo. Design of double error correcting codes based on cellular automata, Journal of Applied Mathematics and Computing. **21**, 545-553. 2006.

[8] D.R. Chowdhury, S. Basu, I.S. Gupta, and P.P. Chaudhuri. Design of CAECC-Cellular Automata Based Error Correcting Code, IEEE Transactions on Computers. **43**, 759-764. 1994.

[9] A. K. Das. Additive cellular automata: Theory and application as a built-in self-test structure, PH.D. Thesis, I.I.T. Kharagpur, India. 1990.

[10] K. Dihidar and P. P. Choudhury. Matrix algebraic formulae concerning some exceptional rules of two dimensional cellular automata, Information Sciences **165**, 91-101. 2004.

[11] S. Inokuchi and T. Sato. On limit cycles and transient lengths of cellular automata with threshold rules, Bulletin of Informatics and Cybernetics. **32** (1), 2360. 2000.

[12] A. R. Khan, P. P. Choudhury, K. Dihidar, S. Mitra, and P. Sarkar. VLSI architecture of a cellular automata, Computers Mathematics with Applications. **33**, 79-94. 1997.

[13] A. R. Khan, P. P. Choudhury, K. Dihidar, and R. Verma. Text compression using two dimensional cellular automata, Computers Mathematics with Applications. **37**, 115-127. 1999.

[14] S. Lipschutz. Theory and problems of linear algebra, Mc Graw Hill Inc. 1990.

[15] F.J. MacWilliams and N.J.A Sloane. The Theory Of Error Correcting Codes, North-Holland Pub. Co., 1977.

[16] M. Mihaljevic, Y. Zheng, and H. Imai. A family of fast dedicated one-way hash functions based on linear cellular automata over GF(q), IEICE Transactions on Fundamentals **E82-A**, 40-47. 1999.

[17] K. Nishinar and D. Takahashi. Multi-value cellular automaton models and metastable states in a congested phase, Journal of Physics A: Mathematical and General. **33**, 7709-7720. 2000.

[18] I. Siap, H. Akın, and F. Sah. Characterization of two dimensional cellular automata over ternary fields, Journal of the Franklin Institute, **348**, 1258-1275. 2011.

[19] I. Siap, H. Akın, and F. Sah. Garden of Eden configurations for 2-D cellular automaton with rule 2460N, Information Science. **180** (18), 3562-3571. 20107

[20] I. Siap, H. Akın, and S. Uguz. Structure and reversibility of 2-dimensional hexagonal cellular automata, Computers Mathematics with Applications, **62**(11), 4161-4169. 2011.

[21] J. Von Neumann. The theory of self-reproducing automata, (Edited by A.W.Burks), Univ. of Illinois Press, Urbana. 1966.

[22] S. Wolfram. Statistical mechanics of cellular automata, Reviews of Modern Physics. **55**(3), 601-644. 1983.