OPTIMIZATION OF 2-D LATTICE CELLULAR AUTOMATA FOR PSEUDORANDOM NUMBER GENERATION

MARIE THERESE ROBLES QUIETA

Department of Electrical and Computer Engineering,

National University of Singapore

10 Kent Ridge Crescents, Singapore 119260

E-mail: g0202825@nus.edu.sg

SHENG-UEI GUAN

Department of Electrical and Computer Engineering,

National University of Singapore

10 Kent Ridge Crescents, Singapore 119260

E-mail: eleguans@nus.edu.sg

This paper proposes a generalized approach to 2-d CA PRNGs – the 2-d lattice CA PRNG – by introducing vertical connections to arrays of 1-d CA. The structure of a 2-d lattice CA PRNG lies in between that of 1-d CA and 2-d CA grid PRNGs. With the generalized approach, 2-d lattice CA PRNG offers more 2-d CA PRNG variations. It is found that they can do better than the conventional 2-d CA grid PRNGs. In this paper, the structure and properties of 2-d lattice CA are explored by varying the number and location of vertical connections, and by searching for different 2-d array settings that can give good randomness based on Diehard test. To get the most out of 2-d lattice CA PRNGs, genetic algorithm is employed in searching for good neighborhood characteristics. By adopting an evolutionary approach, the randomness quality of 2-d lattice CA PRNGs is optimized. In this paper, a new metric, *#rn* is introduced as a way of finding a 2-d lattice CA PRNG with the least number of cells required to pass Diehard test. Following the introduction of the new metric *#rn*, a cropping technique is presented to further boost the CA PRNG performance. The cost and efficiency of 2-d lattice CA PRNG is compared with past works on CA PRNGs.

Keywords: pseudorandom number generation, cellular automata, 2-d cellular automata

1. Introduction

Random number generation has been a requirement in a wide range of applications like computer simulations, built-in self test, and cryptography[1]. Most generators are deterministic in nature, in a sense that, random numbers are generated using mathematical methods. These random numbers are the so-called pseudorandom numbers, opposing to the fact that they are not true random numbers generated by physical, non-deterministic means.

There are quite a number of pseudorandom number workhorses. Some examples are linear feedback shift register (LFSR), linear congruential generator (LCG), and cellular automata (CA). In the past decade, researchers have been focusing more on the development of CA as pseudorandom number generator (PRNG) primarily because of its easy implementation in hardware, as CA simply assumes a network of Boolean functions. CA PRNGs have been studied extensively[2-22] and for the past years, they showed outstanding performance and superiority over other PRNGs[2, 15].

CA PRNG structures may vary in complexities given that CA can be employed in one-dimensional string, two-dimensional grid, or three-dimensional solid. Complexity in CA structures increases with good performance. Likewise, complexity in hardware introduces additional implementation costs. Thus, it is considered necessary to design a CA PRNG with the least structural complexity and a cost-effective implementation.

This paper proposes a general 2-d CA structure – the 2-d lattice CA, which can be implemented in hardware in a simple and cost-effective way. The structures of 2-d lattice CA PRNGs lie in between that of 1-d CA and 2-d CA grid PRNGs[16-18]. With the generalized approach, 2-d lattice CA PRNG offers more variations to 2-d CA PRNG. By introducing a new approach to 2-d CA structures, 2-d lattice CA PRNGs are optimized to its best performance without disregarding the cost of implementation. The main consideration of implementation cost is the number of CA cells required to give good randomness performance[12-14, 19]. This paper keeps the same objective in focus.

The paper is organized as follows. Section 2 gives an overview of the cellular automata and its application as PRNGs in past works. Section 3 introduces 2-d lattice CA and some genetic algorithm as an optimization solution. Section 4 presents an approach to analyze CA PRNGs which leads to lesser cost of implementation. Section 5 concludes the paper.

2. Cellular Automata PRNGs

2.1. Cellular automata overview

A cellular automaton is an array of cells where each cell is in any one of its permissible states, i.e., $s \in \{0,1\}$ if Boolean CA is considered. The cells in a CA array are updated synchronously at discrete time steps (clock cycle) by certain functions, called transition rule functions. The transition rule is a function of the previous states of its *k* neighbors for a *k*-neighborhood CA. Normally, each cell's neighborhood considers itself and the cells physically closest to it. For each cell in a 3-neighborhood 1-d CA with *n* number of cells, $x_i(t) = f_i(x_{i-1}(t-1), x_i(t-1), x_{i+1}(t-1)) \forall i = 1, 2, ..., n$ where f_i represents the transition rule for the *i*th cell. In normal 2-d CA grids[16-18], two types of neighborhood are considered: von Neumann neighborhood and Moore neighborhood. The von Neumann neighborhood is a 5-neighborhood CA, consisting of the cell along with its four immediate nondiagonal neighbors. The Moore neighborhood is a 9-neighborhood CA, consisting of the cell along with its eight surrounding neighbors. In this paper, von Neumann neighborhood is assumed so as to reduce connection costs among the cells.

In accordance with Wolfram's convention[6], transition rule functions are defined in Boolean forms. For a *k*-neighborhood CA, there are 2^k possibilities of combining the state values of neighbors. Each combination has an equivalent next state value for a certain cell x_i, depending on the rule function used by that particular cell. For a 3-neighborhood CA, there are $2^3 = 8$ combinations of neighbor states. Performing a rule function on each of the 8 combinations would result in 8 next-state values of x_i. The transition rule names of CA are based on the decimal equivalent of 8 next state values. For instance, if a certain rule function results to 00011110_2 (equivalent to 30_{10}), then the rule function is named rule 30. The most commonly used rule functions in pseudorandom number generation are rule 30, rule 90, rule 105, rule 150, and rule 165.

If all the CA cells obey the same rule, then the CA is said to be uniform; otherwise, it is nonuniform or hybrid[9]. A CA is said to be a periodic boundary CA if the extreme cells (leftmost and rightmost cells) are adjacent to each other. A CA is said to be null-boundary CA if the extreme cells are connected only to its left (right) cell[3] and a fixed state (unchanging in time) is assigned to its supposed to be right (or left) neighboring cell. In this work, 2-d lattice CA is nonuniform. If a cell located at the boundary is directly connected to one cell only, periodic boundary is assumed, so as to introduce another directly connected cell. It will be seen later in the next section.

2.2.Random number feeds to randomness tests suites

Random numbers are obtained from CA by extracting the bit values of the CA cells at each time step. In some works[13-14], bits are sampled so as to increase the randomness quality of the CA PRNGs by avoiding correlations[22] among the pseudorandom numbers produced. Typically, cell spacing and time spacing are the sampling methods used. It has been proven that spacing improves the performance of CA PRNGs. However, efficiency of CA is neglected as some CA bits are not utilized.

In previous works[12-14, 19], randomness quality of CA PRNGs is tested by two well-known test suites: ENT test[23] and Diehard test[24]. ENT test is a collective term for the three tests which are the Chi-square test, entropy test, and the serial correlation coefficient (SCC) test. Normally, before testing the randomness of a CA PRNG with Diehard, it is subjected first to the ENT test[13-15, 19]. Marsaglia's Diehard test is described by Hellekalek[22] as a well-established battery of empirical tests. Most researchers working on PRNGs prefer to test its performance by Diehard test as it is believed to be a very stringent test[1, 7, 17, 21, 22]. The Diehard test is a collection of 18 battery of statistical tests. In some works[7], pseudorandom numbers were tested with some of the 18 tests in Diehard, excluding the three stringent tests which are OPSO, OQSO, and DNA test. Similar to other works[8, 12-14, 17, 19], 2-d lattice CA is tested with all of the 18 battery of tests. Thus, if a PRNG can pass the Diehard test, then, that PRNG is said to have good randomness quality.

Each randomness test suite has certain requirements to be satisfied. In particular, Diehard test requires the set of pseudorandom numbers to be in 8-bit, 16-bit or 32-bit random words[24]. For some unknown reasons, other works on CA PRNG that use Diehard test as measure of randomness did not talk about how random words are generated by different number of CA cells. For benchmarking purposes, 8-bit pseudorandom word generators for Diehard test are assumed in this work. Also, sampling or spacing is not employed, that is to say, all cell values at all time steps are used to generate the pseudorandom numbers. In the succeeding paragraphs, a discussion on random word feeds to Diehard test is provided.

The state of a CA, $X(t)$ at discrete time step *t* is defined as the *n*-tuples formed from the states of the individual cells, $X(t) = [x_1(t), x_2(t), ..., x_n(t)]$ where *n* is the total number of CA cells. This means that a string of *n* bits is produced by the CA *X*(*t*) at a particular discrete time step *t*. If the strings of *n* bits for all time steps are juxtaposed, thus forming *t***n* bits, a general equation of a CA can be defined as,

$$X(t) = [x_1^{tn+1}(t), x_2^{tn+2}(t), \dots, x_n^{(t+1)n}(t)], \forall t \geq 0 .$$

For example, at *t*=0 and *n*=50,

$$X(0) = [x_1^1(0), x_2^2(0), \dots, x_{50}^{50}(0)] .$$

Similarly, at *t*=1 and *n*=50,

$$X(1) = [x_1^{51}(1), x_2^{52}(1), \dots, x_{50}^{100}(1)] .$$

At discrete time step *t*=0, initial random seeds are used for all CA cells, $x_i(0) \forall i = 1, 2, ..., n$. These are obtained by using the C++ function generating random numbers.

For an *m*-bit CA PRNG, pseudorandom numbers r_j are generated using the equation

$$r_j = [x^{jm+1}, x^{jm+2}, \dots, x^{(j+1)m}], \forall j \geq 0 .$$

As previously mentioned, in this paper, 8-bit (*m*=8) pseudorandom words are generated. Thus, even if the total number of CA cells *n*, is not divisible by 8, all the CA cells' bit values at all time steps are utilized for random number generation.

For example, the 1st random number r_0 is generated by the first 8 bits.

$$r_0 = [x^1, x^2, \dots, x^8] .$$

The 7th random number (note that *j*=6) is

$$r_6 = [x^{49}, x^{50}, \dots, x^{56}].$$

If the total number of CA cells is 50 as in the previous example of $X(t)$, then, the bits forming the random word r_6 is not necessarily taken from one time-step states of CA cells. Here, generating the random numbers is not directly time-dependent. Instead, a few random words are generated by each time step and at times, a random word may be a function of cell values at two or more consecutive time steps. As seen in the example of r_6 in relation to $X(t)$ example, bits x^{49}, x^{50} are extracted from cell values at t=0 and bits $x^{51},\dots,x^{56}$ are extracted from cell values at t=1.

2.3.CA PRNGs

One-dimensional CA PRNGs were the focus of researchers in the last decade. Wolfram first used CA as a PRNG[3]. He used uniform rule 30 1-d CA and showed that that it can produce fairly random temporal bit sequences. Following uniform CA, Hortensius[9] studied rule 90-150 programmable CA (PCA) and rule 30-45 PCA. Unlike the uniform CA, PCA is a nonuniform CA which uses rule control signals to vary the rule for each cell at every time step. In PCA, 1-bit and 2-bit control ensue two choices and four choices of rules, respectively. Tomassini et al. further explored PCA 90-165[8] and 2-bit control PCA referred to as PCA 90-105[17]. The 2-bit PCA were evolved using a cellular programming evolutionary algorithm. Results showed that the latter of Tomassini's work is better than Hortensius' work.

Controllable cellular automata (CCA)[13-14] was proposed to further improve the quality of nonuniform 1-d CA. Aside from rule control signals, a cell control signal is introduced to alter CA properties. Although CCA can outperform PCA to a large extent, it entails more complex properties than PCA. Following the development of CCA, self-programmable CA (SPCA) [12] was proposed. SPCA uses the states of the cells in the same CA in previous time steps by having memory cells storing bit values way back time step *t-2*. SPCA can pass Diehard at 21 cells.

Chowdhury et al. first proposed a methodology which generates pseudorandom number using 2-d CA grid[16]. They showed that 2-d CA PRNG is superior to 1-d CA PRNG using the same number of cells. Tomassini et al. worked on 2-d CA as well[17]. In his work, he studied time spacing parameters and recommended time spacing of 2 for practicality. Also, he concluded that evolved 2-d CA grid PRNGs can outperform the evolved 1-d CA PRNGs[8] and Chowdhury et al.'s 2-d CA grid PRNGs[16].

Recently, 2-d CA variation[19] with asymmetric neighborship properties was introduced. Asymmetric neighborship property means that a certain cell *a* considers cell *b* as its neighbor but cell *b* does not consider cell *a* as neighbor. This property is applied to the proposed 2-d CA variation, which is much simpler than a normal 2-d CA grid[16-18]. The 2-d CA structure is a collection of several 1-d CA connected at the boundary cells. Evolutionary algorithms were used to find the neighborship and structures of this 2-d CA variation. The paper concludes that diagonal neighbor connections are indispensable in the 2-d CA variation in order to pass all the 18 tests of Diehard. Also, the proposed 2-d CA variation can compete with normal 2-d CA grid[16-18] in terms of performance and entails lower implementation cost.

Table 1 provides a summary of all existing CA PRNG designs and the randomness quality of each PRNG based on the number of tests that it can pass in Diehard. PCA and CCA PRNGs can pass all tests in Diehard, however, they are very inefficient as not all the cells are used in the random number generation. Moreover, these CA PRNGs require at least 50 cells to pass all the tests in Diehard. At present, SPCA is the most novel 1-d CA PRNG design that shows potentials in generating high quality random numbers with few CA cells. SPCA PRNGs can pass Diehard at very minimal cost, i.e., 21 cells by using memory cells which stores previous bit values of CA cells.

Table 1. Past works on CA PRNG and its performance in Diehard test.

CA PRNG design	**No. of cells**	**No. of tests passed in Diehard**	**Remarks**
PCA[8]	50	17	Uses time and cells spacing
CCA0/CCA2[13]	50	18	Uses time and cells spacing
SPCA 150/105[12]	21	18	Uses memory cells
2-d CA grid[17]	64	18	Uses more cells
2-d CA variation[19]	48	18	Uses diagonal connections

Conversely, 2-d CA can not be simply compared with SPCA based on randomness quality alone. Complexity and hardware implementation should also be taken into consideration. Additional cost is introduced by SPCA because of the memory cells and connections. In the case of 2-d CA, additional connection cost is also required.

In the two-dimensional domain, 2-d lattice CA can be directly compared with other works like 2-d CA grid[16-18] and 2-d CA variation[19]. The 2-d CA variation[19] has been empirically proven to outperform other 2-d CA PRNG works[16-17]. It can pass Diehard with few cells, i.e. 48 cells, and in addition, it decreases hardware cost by improving the cell connections. But as will be shown later, 2-d CA variation is a special case of 2-d lattice CA. Thus, the final simulation results of 2-d lattice CA at the end of the paper will be compared with 2-d CA variation[19].

3. 2-d Lattice Cellular Automata

3.1. 2-d lattice CA structure

The structure of a 2-d CA variation[19] was presented as the simplest 2-d CA representation. 2-d lattice CA is a more general approach to this 2-d CA variation and to a 2-d CA grid[16-18] as well. Fig. 1 shows the structure of 2-d lattice CA.

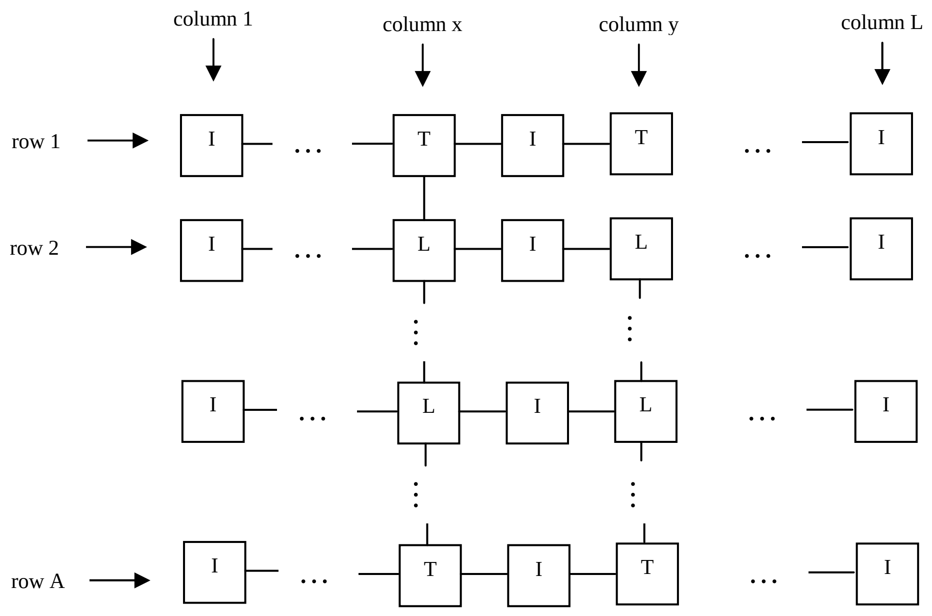

Fig. 1. The structure of a 2-d Lattice CA

2-d lattice CA is basically an array of 1-d CA stacked contiguously and vertically connected at some columns. In this paper, the term *vertical line connection* is abbreviated as *VL*. Given an L*A 2-d lattice CA, where L is the number of columns and A is the number of rows, the cells can be classified under three types according to location:

i) T-cells (labeled as T in Fig. 1). These are the cells in the first and last rows that are positioned along the VL. Each t-cell has three cells directly connected to it.
ii) Lattice cells (labeled as L in Fig. 1). Aside from the cells in the first and last rows, the cells that are positioned along the VL are called lattice cells. Each lattice cell has four cells directly connected to it.
iii) Intermediate cells (labeled as I in Fig. 1). The cells which are neither t-cell nor lattice cell are called intermediate cells. Each intermediate cell has two cells directly connected to it.

Unlike the 2-d CA variation[19], 2-d lattice CA need not have diagonal connections of cells in order to improve the randomness. 2-d lattice CA structure makes the 2-d CA variation simpler by doing away with diagonal connections. It also makes the 2-d CA grid[16-17] less costly by removing some VLs in selected columns of CA cells.

A key factor that affects the randomness quality of 2-d lattice CA PRNGs is the positioning of VLs. There are two VLs and these are positioned at the extreme sides (boundary), i.e. first and last columns[19]. There may be better positions for these two VLs other than the extreme sides. This issue of positioning VL in 2-d lattice CA is studied by experimenting on L*A 2-d lattice CA setting, where L=10 and A=5. It has been empirically proven[19] that 10*5 2-d CA setting is the best setting for a 50-cell 2-d CA.

Table 2 shows the ENT test results comparison of some of the 2-d lattice CA PRNGs by varying the VL positioning. The experiment conditions are identical for all the 2-d lattice CA PRNGs tested. The output sequences are generated by all the cells following method 5[19], where the output bits are extracted vertically from top to bottom and left to right. All cells are uniform cells and the transition rule function used is rule 150. For the intermediate cells, the cells immediately left and right to it are its left and right neighbors, respectively. If the cells in the extreme sides are intermediate cells, periodic boundary condition is assumed. For the lattice cells and t-cells, the cells immediately below and right to it are its left and right neighbors, respectively. If the t-cell is in the last row, the cells immediately above and right to it are its left and right neighbors, respectively. If the lattice cell or t-cell is in the last column, the cells immediately below and left to it are its right and left neighbors, respectively. The length of the tested sequences is 10,000 bytes. 500 initial seeds are tested for each CA PRNG. For notation purpose, *x*-*y* 2-d lattice CA means that cells are vertically connected along columns *x* and *y*, assuming that column 1 is the 1st column of a 10*5 2-d lattice CA PRNG.

Table 2. ENT test results comparing VL positions at VL=2.

x-y 2-d lattice CA	***Chi –square***	***Entropy***	***1-SCC***
1-2 2-d lattice CA	0.330000	7.971467	0.990955
1-4 2-d lattice CA	0.391000	7.972949	0.991529
1-10 2-d lattice CA	0.000000	7.458368	0.950632
2-5 2-d lattice CA	0.470000	7.973961	0.991148
3-6 2-d lattice CA	0.433000	7.974107	0.990701
4-7 2-d lattice CA	0.464000	7.973400	0.991600

Since L=10, the 2-d CA variation[19] is equivalent to the 1-10 2-d lattice CA PRNG in Table 2. From the results, there are better performing 2-d lattice CA structures than the 2-d CA variation[19]. These initial results of experimentation by ENT test triggers off the objective of searching for more 2-d CA variations. This suggests that proper positioning of VL alone can enhance the 2-d CA PRNGs performance and diagonal connections of the 2-d CA structure are unnecessary.

Aside from VL positions, different number of VLs also results in another 2-d CA variation. If VL=2, then, it is equivalent to the 2-d CA variation[19]. Maximizing VL, i.e., VL=L, equates to 2-d CA grid[16-18]. Initial experiments show that there are better 2-d CA structures other than these two, e.g., when 2<VL<L. This 2-d lattice CA property is further explored in the later sections of this paper.

3.2. Evolutionary approach

In our evolutionary approach to CA PRNG structures and properties[14-15, 19], ENT test suite result is used as objective. This is because generating the random number sequences to be tested with Diehard is too time-consuming. Diehard test requires more random numbers to be tested than the ENT test. Thus, in this work, some comparisons are made using ENT test, e.g., in preliminary experiments and evolution process of 2-d lattice CA PRNGs. The Diehard test is used to verify the results of the CA PRNGs which give good ENT test results.

ENT test is a collective term for three tests: chi-square, entropy, and serial correlation coefficient (SCC). The overall evaluation for the ENT test can be obtained from the *F value* as given in Equation (1). In comparing good quality CA PRNGs, the entropy (*ent*) and SCC values normally have comparable results with minimal discrepancies unlike the chi-square value. Because the chi-square test is an important indication of randomness, it is given the highest weight in the calculation of *F*.

$$F = (entropy - 7) * 30\% + (1 - |SCC|) * 30\% + f(chi - square) * 40\% \qquad (1)$$

$$\text{where:} \quad f(chi - square) = \begin{cases} 0; & \text{if chi-square } > 90\% \ or < 10\% \\ 1; & \text{if } 10\% < \text{chi-square} < 90\% \end{cases}$$

The rationale behind the evolutionary approach in this paper is to search for CA PRNG structures and properties that can give satisfactory ENT results. According to some initial experiments, a CA PRNG with good ENT result does not guarantee that it can pass all of the 18 tests in Diehard. Alternatively, CA PRNGs demonstrating satisfactory ENT results are more likely to pass at least 15 tests in Diehard, excluding the three stringent tests (OPSO, OQSO, DNA tests). It is observed that if a certain CA PRNG has $F< 0.9$ (assuming that number of initial pseudorandom seeds, $S = 100$), the CA PRNG will certainly not pass Diehard test[13-14]. Hence, ENT result is considered satisfactory if $F>0.9$. Ultimately, Diehard test suite is the foremost gauge of evaluating randomness quality of PRNGs. Thus, in this work, more experiments on Diehard test are conducted.

The chromosomes structure of an L*A 2-d lattice CA is divided into three parts: the location of the vertical line connection VL (maximum is L), the neighborship assignment for t-cells, and the neighborship assignment for lattice cells. In this work, four bits are assigned to choose the location of VL. Each t-cell has three directly connected neighbors.

To choose two out of three cells to act as neighbors, the neighborships of t-cells are evolved using two bits for each t-cell. Each lattice cell has four directly connected neighbors. To choose two out of four cells to act as neighbors, the neighborships of lattice cells are evolved using three bits for each lattice cell. Although von Neumann neighborhood is used here, 3-neighborhood CA is still assumed, i.e., each cell uses the previous states of itself and two of its directly connected cells to get the next state value. Some directly connected cells although do not function as neighbors but they function as rule control cells. This demonstrates the asymmetric neighborship property of the 2-d CA variation[19].

The general chromosome structure of a 2-d lattice CA is described in details in the Appendix. Following the example in the Appendix, the structure of a 10*5 2-d lattice CA with two vertical connections is shown in Fig. 2. $T_{i\text{-}j}$ and $L_{i\text{-}j}$ are used as naming conventions for $(j+1)$th t-cells and lattice cells, respectively, lying at the vertical connection directed by VL_i.

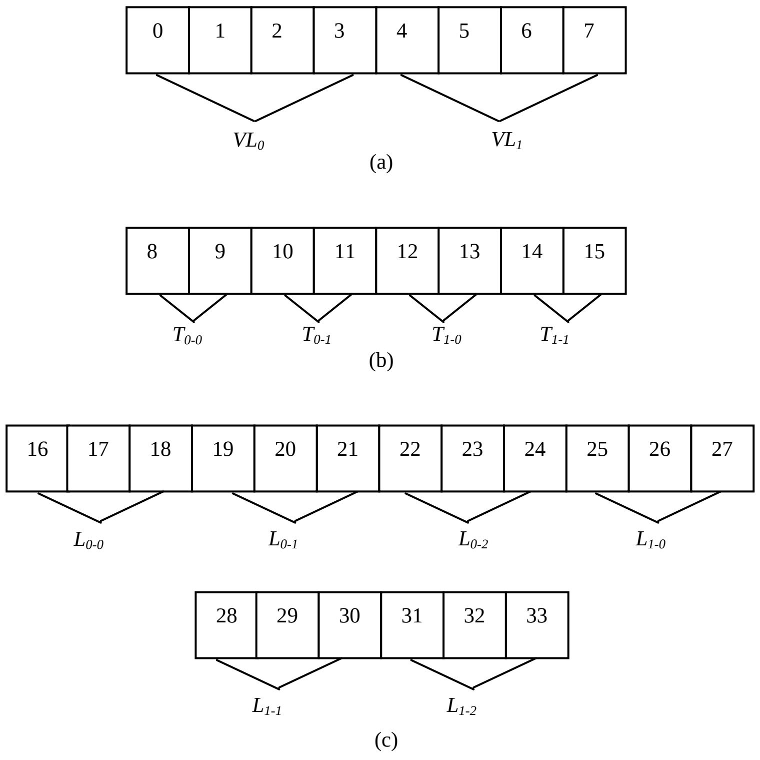

Fig. 2. The chromosome structure of 10*5 2-d lattice CA with 2 VLs: (a) bits 0-7 for the 2 VLs, (b) bits 8-15 for the 4 t-cells, and (c) bits 16-33 for the 6 lattice cells.

In all the experiments conducted, the input of evolution process is randomly generated by a C++ function. Population size is set at 40. The stopping criterion is the maximum stagnation steps. If the best chromosomes keep unchanged for 100 steps, the evolution is stopped. The 1-point crossover rate is set at 0.95. The bit mutation rate is set at 0.05 per chromosome. During reproduction, half of the better-performing parents and child chromosomes are copied into the next generation. The objective of fitness calculation follows the *F* value as described in Eq. (1).

3.3.2-d lattice PCA

According to past researches on CA PRNGs, applying rule control signals to CA cells improves the randomness quality[8, 11-14, 17, 19]. This idea of rule control signal is also employed in this work in order to improve the 2-d lattice CA PRNG performance. Each lattice cell has four directly connected cells, two of which acting as neighbors. The other two cells, then, can act as rule control signals for these lattice cells. To make it simpler and less costly, only one cell is used as rule control signal. Consequently, lattice cells become PCA cells. T-cells remain to be uniform CA cells.

As described in the previous section, the left and right neighbors of each lattice cell are evolved by some genetic algorithms. After evolution, the evolved neighborship characteristics of 2-d lattice CA PRNGs are fixed. This means that each lattice cell has been assigned two of its four possible neighbors. The rule control cell is then chosen from the remaining two directly connected cells.

Careful selection of the rule control cell for each lattice cell should be observed since the rule control cell is a directly connected cell, thereby, introducing correlations with the lattice cell under consideration. For instance, if two cells *a* and *b* are directly connected in the same row, cell *a* may regard cell *b* as its neighbor, and cell *b* may consider cell *a* as its rule control cell. There are numerous occurrences of this example that need to be considered. So, in order to avoid an extensive and time-consuming search, an automated selection of rule control cells is employed. Depending on preference, the various modes of rule control cell selection are the following:

i) Mode 1: 1 – left, 2 – above, 3 – right, 4 – below

ii) Mode 2: 1 – above, 2 – below, 3 – left, 4 – right

iii) Mode 3: 1 – left, 2 – right, 3 – above, 4 – below

iv) Mode 4: 1 – above, 2 – left, 3 – below, 4 – right

where: 1 has the highest preference.

Using Mode 1 as an example, if the left cell is not acting as neighbor, it would be the first choice as rule control cell. Otherwise, the cell above, which is preference 2, is chosen. This means that higher preference cells, not acting as neighbors, are selected first to provide the rule control signals.

Take note that these modes of rule control cell selection are applied only to lattice cells which are PCA cells. More 2-d CA PRNG variations may be experimented with by making t-cells programmable (e.g., PCA) like the lattice cells. Additional costs are introduced though.

Table 3 shows how 2-d lattice CA is improved by making lattice cells PCA cells. The structures for 10*5 2-d lattice CA PRNGs with 3 VLs are evolved for 200 evolution steps. At each evolution step, the average *F* value of 100 initial seeds tested decides the next generation chromosomes. Other experiment conditions for the CA transition follow the first experiment in the previous section. The chromosome structures of the 10*5 2-d lattice CA PRNG with 3 VLs describe the positioning of the vertical connections, the neighborships of all t-cells, and the neighborships of all lattice cells. After evolution, the automated selection of rule control cells is then applied to the best structures obtained. These are subjected to Diehard test for 10 initial seeds. As shown in Table 3, 2-d lattice CA with lattice PCA cells can almost pass Diehard test.

Table 3. Average Diehard test results of 2-d lattice CA with and without any rule control cell for lattice cells using 10 initial seeds

10*5 2-d lattice CA with 3 VLs	**Average number of tests passed in Diehard test**	
	Without rule control	**With rule control**
2-d lattice CA structure 1	14.8	16.9
2-d lattice CA structure 2	14.8	16.9
2-d lattice CA structure 3	14.6	17.4
2-d lattice CA structure 4	11.4	17

The modes of rule control cell selection are further analyzed at different number of VLs, whereVL=3, 4, and 5. Again, the rule control selection concept is incorporated in the evolved 2-d lattice CA PRNG structures. It is also possible to evolve the 2-d lattice CA structures along with the modes of selection. But since there are only four different modes, it is more practical to find it by hand.

Fig. 3 compares the performance of 2-d lattice CA PRNGs with VL=3, 4, and 5 under different modes of rule control selection. Each point in the graph indicates an average

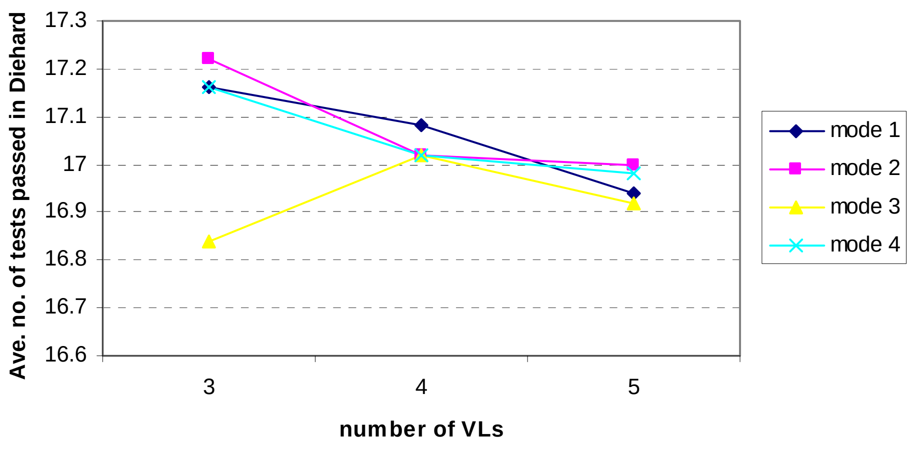

Fig. 3. Average Diehard test results for different modes of selecting rule control cell.

Diehard test result of 5 different 2-d lattice CA PRNG structures evolved. The four modes of rule control selection are applied to each 2-d lattice CA PRNG structure, which are then tested with 10 initial seeds.

Fig. 3 shows that mode 3 is not a good choice of method for choosing rule control cell. Recall that in mode 3, the preferences are: 1 – left, 2 – right, 3 – above, 4 – below. This shows that left and right cells, if possible, should not be chosen as control signals. This suggests that correlations between the neighbors possibly exist. For intermediate cells, there are only 2 directly connected cells, which act as left and right neighbors. Thus, if the cell's neighbor uses the cell for rule control, then, correlations are more probable. From among all the modes of selection, mode 2 can be regarded as the best mode to use. It is expected since mode 2 prioritizes the selection of cells above or below lattice cells as neighbors, which avoids correlations among left and right neighbors. With these results, mode 2 rule control selection is applied in the experiments hereafter.

3.4.Discussion

Section 3 focuses on the internal properties of 2-d lattice CA PRNGs, e.g., positioning of the vertical connections, the neighboship characteristics of cells, and the application of rule control signals to lattice cells. Each of these internal properties affects the other. If two consecutive columns are both vertically connected by VL, then, the neighborship characteristic (and rule control selection property) is different from the case wherein two consecutive columns are not both vertically connected. There are a lot of 2-d lattice CA

PRNG variations and with the aid of genetic algorithms, a faster and more efficient way of searching for good CA PRNGs is accomplished.

The objective of this paper is to impart more possible alternatives of designing 2-d CA PRNGs. What the past works on 2-d CA PRNGs presented are merely straightforward approaches to 2-d CA, which bear fairly good performance and not necessarily effective and efficient way for design and implementation. Accordingly, there are quite a number of 2-d lattice CA PRNGs with varying internal structures and properties which can outperform the conventional 2-d CA PRNG and 2-d CA variation[19]. A clear-cut means of achieving this is with the use of simple genetic algorithms.

Aside from 2-d lattice CA PRNG variations by internal structures and properties modifications, the array setting of the 2-d lattice CA is another crucial consideration. Array settings and the number of vertical connections are significant factors affecting the randomness of 2-d CA PRNGs. These are the important factors that visibly set apart 2-d CA from 1-d CA. In the next section, optimization of 2-d lattice CA PRNG is handcrafted by searching for good array settings and number of VLs.

4. Optimization of 2-d Lattice CA PRNGs

4.1 Evolution process of different 2-d lattice CA setting and its performance

2-d CA setting of L*A greatly affects the randomness quality of PRNGs[19]. However, the analysis of array setting in 2-d CA variation[19] is not enough as it only covers the results based on ENT test. In this section, different L*A settings are compared using Diehard test as the gauge. For each 2-d lattice CA setting, the number of VLs is also examined.

For a step-by-step coherent analysis, the approach of the experiments for each L*A 2-d lattice CA PRNG is as follows. First, the internal structures and properties of the L*A 2-d lattice CA PRNG is evolved, adhering to the chromosome structure presented in the Appendix and evolution parameters discussed in the previous section. Next, the five better-performing chromosomes for the L*A 2-d lattice CA under consideration are subjected to Diehard test. Finally, different L*A settings are analyzed, and an improvement strategy for 2-d lattice CA PRNGs to pass Diehard test is applied.

For each L*A 2-d lattice CA setting, different VLs are investigated. Because evolution time increases with chromosome length, a certain maximum number of VLs for each L*A setting is limited by the chromosome length required for the evolution process. Here, the maximum chromosome length is set to 100 bits. Table 4 shows the calculation of chromosome length in each L*A 2-d CA. Following the Appendix notation, *vl* is the number of vertical connections, *t* is the number of t-cells, *l* is the number of lattice cells, and *cl* is the total chromosome length.

Table 4. Chromosome length calculation of L*A 2-d lattice CA setting.

L	**A**	**L*A**	***vl***	***t***	***l***	***cl***
9	5	45	2	6	4	34
9	5	45	5	15	10	85
5	9	45	3	21	6	87
15	3	45	5	5	10	55
15	3	45	9	9	18	99
3	15	45	2	26	4	94
11	4	44	2	4	4	28
11	4	44	7	14	14	98
4	11	44	2	18	4	70
7	6	42	2	8	4	40
7	6	42	5	20	10	100
6	7	42	2	10	4	46
6	7	42	4	20	8	92
8	5	40	3	9	6	51
8	5	40	5	15	10	85
5	8	40	2	12	4	52
5	8	40	3	18	6	78
10	4	40	3	6	6	42
10	4	40	7	14	14	98
4	10	40	3	24	6	96
13	3	39	4	4	8	44
13	3	39	8	8	16	88
3	13	39	2	22	4	82
12	3	36	3	3	6	33
12	3	36	8	8	16	88
3	12	36	2	20	4	76
9	4	36	3	6	6	42
9	4	36	7	14	14	98
4	9	36	3	21	6	87

As mentioned earlier, after the evolution process of each item in Table 4, Diehard test is conducted on these evolved 2-d lattice CA structures. To roughly illustrate and compare the randomness quality of L*A 2-d lattice CA, the 5 chromosomes of each item are categorized based on the number of tests passed in Diehard: 0~9, 10~14, 15~18. Table 5 shows the percentage of chromosomes (how many chromosomes out of 5) that can be classified into a particular category. Also, the maximum number of tests in Diehard that an L*A 2-d lattice CA can pass is shown in the table.

Table 5 Diehard test results of some 2-d lattice CA configurations

				Number of tests passed in Diehard			
L	**A**	**L*A**	***vl***	**0 ~ 9**	**10 ~ 15**	**15 ~ 18**	**maximum**
9	5	45	2		0.60	0.40	16
9	5	45	5		0.40	0.60	16
5	9	45	3	0.60	0.40		not impt.*
15	3	45	5			1.00	17
15	3	45	9		0.20	1.00	17
3	15	45	2	1.00			not impt*
11	4	44	2		0.60	0.40	16
11	4	44	7		0.60	0.40	16
4	11	44	2	0.60	0.40		not impt*
7	6	42	2	0.20	0.40		15
7	6	42	5		0.20	0.80	17
6	7	42	2	0.60	0.40		not impt*
6	7	42	4	0.80		0.20	15
8	5	40	3	0.80	0.20		14
8	5	40	5	0.20	0.80		14
5	8	40	2	0.80	0.20		not impt*
5	8	40	3	1.00			not impt*
10	4	40	3	0.20	0.80		13
10	4	40	7	0.40	0.60		14
4	10	40	3	0.80	0.20		not impt*
13	3	39	4	0.40		0.60	16
13	3	39	8	0.20		0.80	16
3	13	39	2	0.60	0.40		not impt*
12	3	36	3	0.60	0.40		12
12	3	36	8	0.80	0.20		11
3	12	36	2	1.00			not impt*
9	4	36	3	0.80	0.20		10
9	4	36	7	0.40	0.60		11
4	9	36	3	0.60	0.40		not impt*

* If none of the chromosomes can pass at least 15 tests passed in Diehard, then, the result is considered not important.

In Table 5, the results of categorizing chromosomes show that the more number of VLs in 2-d lattice CA, the better the randomness quality based on Diehard test. But as pointed out in the previous section and some previous works, connection cost matters. The number of vertical connections should be minimized as possible so as to entail lower implementation cost. Another observation that can be made from Table 5 is that, an L*A setting with L>A is better than a setting with A>L for the same number of cells. This suggests that the vertical output method (method 5 as described in Section 3.1) may have a relation with the 2-d lattice CA settings. If A>L, then there are more vertically connected cells than horizontally connected cells, thus, resulting to correlations between output bits that are extracted vertically.

4.2. Introducing the metric #rn

CA PRNGs performance has always been attributed to the number of CA cells. However, it seems that the results in Table 5 do not really justify this notion. For instance, a 39-cell 2-d lattice CA PRNG is better than a 40-cell based on the Diehard test. In this section, a new metric *#rn* is introduced in order to explain the results of Table 5.

As described in Section 2.2, the random word produced is 8-bit in this paper. In the following equations,

$$r_0 = [x^1, x^2, \ldots, x^8]$$

$$r_j = [x^{jm+1}, x^{jm+2}, \ldots, x^{(j+1)m}], \forall j \geq 0 ,$$

there exists a condition wherein x^1 and x^{jm+1} bits are extracted from the same cell, which is the first CA cell in this case. At the first occurrence of such condition, i.e., at minimum *j*, it can be said that one *output cell cycle* has been completed. *#rn* refers to the number of 8-bit random numbers generated during each output cell cycle. Thus, *#rn* = *j*. For an *n* number of output cells, the ideal scenario is when, the maximum *#rn* is reached. The maximum *#rn* is actually the total number of output cells *n*. For instance, if there are 45 output cells, then there are 45x8 CA bits per cycle. Thus, *#rn* is 45x8/8, i.e. 45. Analyzing the metric *#rn* with the results in Table 5 leads to Table 6 results. Table 6 is a simplified version of Table 5, wherein *#rn* is included and the substandard L*A 2-d lattice settings when A>L are disregarded.

Table 6. Comparing maximum Diehard test performance of L*A 2-d lattice settings with the metric *#rn*.

L	**A**	**L*A**	***vl***	**maximum**	***#rn***
9	5	45	2	16	45
9	5	45	5	16	45
15	3	45	5	17	45
15	3	45	9	17	45
11	4	44	2	16	11
11	4	44	7	16	11
7	6	42	2	15	21
7	6	42	5	17	21
8	5	40	3	14	5
8	5	40	5	14	5
10	4	40	3	13	5
10	4	40	7	14	5
13	3	39	4	16	39
13	3	39	8	16	39
12	3	36	3	12	9
12	3	36	8	11	9
9	4	36	3	10	9
9	4	36	7	11	9

The results of Table 6 showed that the higher the metric *#rn* of 2-d lattice CA is, the better the Diehard performance, provided that A>L in the 2-d lattice CA setting. Intuitively, a larger *#rn* is better because it avoid correlations in some ways. Randomness tests suites take into account the *sequence* of random words produced by PRNGs. If random words are frequently produced by the same set of cells, $x^{jm+1}, x^{jm+2}, \ldots, x^{(j+1)m}$, then it is more probable that these random words are correlated. Accordingly, a high value for *#rn* implies that the random words are outputed by the same set of cell in between long intervals.

4.3. Cropping technique

A more in-depth analysis of the metric *#rn* is carried out by introducing a technique called cropping of cells. This is done by cropping the last cell of the 2-d lattice CA, meaning, the last cell is not used to generate output bits. By doing so, the *#rn* increases to its maximum. Table 7 shows the corresponding *#rn* before and after cropping.

Table 7. *#rn* before and after cropping for each L*A 2-d lattice CA

L	**A**	**L*A**	***#rn* w/o cropping**	***#rn* with cropping**
9	5	45	45	-
15	3	45	45	-
11	4	44	11	43
7	6	42	21	41
8	5	40	5	39
10	4	40	5	39
13	3	39	39	-
12	3	36	9	35
9	4	36	9	35

In order to test the effect of increasing *#rn* by cropping, the five structures for each L*A 2-d lattice CA are tested under Diehard as shown in Table 8. From Table 8, it shows that maximizing *#rn* generally improves the performance of 2-d lattice CA PRNG.

Table 8. Diehard test result comparisons of LxA_VL 2-d lattice CA with and without cropping of the last cell.

	no. of tests passed w/o cropping	no. of tests passed with cropping		no. of tests passed w/o cropping	no. of tests passed with cropping
8x5_3			**8x5_5**		
chrom0	8	7	chrom0	10	16
chrom1	9	13	chrom1	12	16
chrom2	14	16	chrom2	14	17
chrom3	9	12	chrom3	10	13
chrom4	8	14	chrom4	9	12
10x4_3			**10x4_7**		
chrom0	11	15	chrom0	8	11
chrom1	7	7	chrom1	14	17
chrom2	11	16	chrom2	8	11
chrom3	13	16	chrom3	10	17
chrom4	12	17	chrom4	13	16
12x3_3			**12x3_8**		
chrom0	8	5	chrom0	8	10
chrom1	12	10	chrom1	5	8
chrom2	12	10	chrom2	6	8
chrom3	9	7	chrom3	11	12
chrom4	6	7	chrom4	7	6
9x4_3			**9x4_7**		
chrom0	7	10	chrom0	7	11
chrom1	8	13	chrom1	3	7
chrom2	6	8	chrom2	10	11
chrom3	10	14	chrom3	10	12
chrom4	9	12	chrom4	11	13
11x4_2			**11x4_7**		
chrom0	11	12	chrom0	13	17
chrom1	14	17	chrom1	15	16
chrom2	13	17	chrom2	16	16
chrom3	15	17	chrom3	12	15
chrom4	16	18	chrom4	14	17
7x6_2			**7x6_5**		
chrom0	11	12	chrom0	11	16
chrom1	10	11	chrom1	17	16
chrom2	8	8	chrom2	17	16
chrom3	15	16	chrom3	16	16
chrom4	15	15	chrom4	17	16

A summary of all the 2-d lattice CA settings with cropping applied is shown in Table 9.

Table 9. Summary of all 2-d lattice CA settings performance under Diehard test.

L	**A**	**L*A**	***vl***	***#rn* w/o cropping**	***#rn* with cropping**	**diehard performance at max *#rn***
9	5	45	2	45	-	16
9	5	45	5	45	-	16
15	3	45	5	45	-	17
15	3	45	9	45	-	17
11	4	44	2	11	43	18
11	4	44	7	11	43	17
7	6	42	2	21	41	16
7	6	42	5	21	41	16
8	5	40	3	5	39	16
8	5	40	5	5	39	17
10	4	40	3	5	39	17
10	4	40	7	5	39	17
13	3	39	4	39	-	16
13	3	39	8	39	-	16
12	3	36	3	9	35	10
12	3	36	8	9	35	12
9	4	36	3	9	35	14
9	4	36	7	9	35	13
7	5	35	5	35	-	14
11	3	33	8	33	-	9
6	5	30	4	15	-	9
9	3	27	6	27	-	9

There are some points to be considered in Table 8. First, although it is said that maximum *#rn* should be used to boost 2-d lattice CA performance, there is still a limit to the improvement it offers. It can be seen from Table 8 that if L*A ≤ 33 cells, the 2-d lattice CA is already much inferior in terms of the Diehard test results. Second, L*A setting does affect the performance of 2-d lattice CA PRNG. An example is L*A=45. 9x5 and 15x3 2-d lattice CA settings do not give the same results. Third, performance-wise, the best L*A setting is 11x4 wherein cropping is applied. This setting can pass all of the 18 tests in Diehard. Finally, 10x4 or 8x5 lattice settings may be considered good cost-wise. This is because the settings use fewer cells and do not differ much on the performance with 11x4 2-d lattice CA settings.

The 2-d Lattice CA PRNG designs mentioned are better than the 2-d CA grid PRNG[17] or 2-d CA variation PRNG[19] not only in terms of Diehard test performance, but also, in hardware implementation and cost. 2-d Lattice CA do away with redundant vertical connections used by 2-d CA grid PRNG and avoids complexities of diagonal connections used by 2-d CA variation PRNG, by using some added properties as discussed earlier. Comparing 2-d Lattice CA PRNG with 1-d CA PRNG, the former requires lesser number of cells to pass the Diehard test.

5. Conclusion and Future Works

This paper proposed a generalization of 2-d CA PRNGs with the introduction of the vertical connections to ordinary 1-d CA PRNGs. Our results showed that 2-d CA PRNGs with certain number of vertical connections perform better in Diehard than 2-d CA grid PRNGs , using the von Neumann neighborhood.

Also, a new metric *#rn* and 'cropping technique' has been presented as a way to find optimal 2-d CA PRNGs in terms of the number of cells. Thus, 2-d CA lattice PRNGs were evolved to its maximum performance. Considering the Diehard performance, 11x4 2-d lattice CA PRNG is said to be a good choice since it can pass all of the 18 tests in Diehard. If cost and efficiency are vital factors, then, 10x4 or 8x5 2-d lattice CA PRNG can be used since it can almost pass the Diehard test.

With this work, more rooms are opened for future research. 2-d CA PRNGs are proved to be better than 1-d CA PRNGs and this paper further strengthens that proposition with the introduction of new variations of 2-d CA. Hence, 2-d lattice CA can be further explored and evolved to provide better PRNGs.

The metric *#rn* may be used to analyze Diehard test not only in 2-d CA, but in 1-d CA/PCA PRNGs as well. Other than the cropping technique, other sampling technique methods that can improve Diehard performance may also be explored. For instance, cropping technique may be extended to two or more cells. This means that two cells per time step are not used in the generation of output bits. Another way is to use dynamic sampling method by dynamically changing the cropped cell/s in time.

Appendix – The chromosome structure of 2-d lattice CA

In an L*A 2-d lattice CA with number of VLs equal to *vl*, the number of t-cells, *t* and the number of lattice cells, *l* can be calculated as: $t = 2vl$ and $l = vl(A-2)$. The total number of bits for each part is: a) total number of bits for all VL, *vlb=4vl*, b) total number of bits for all t-cell neighborship assignment, *tb=2t*, and c) total number of bits for all lattice cell neighborship assignment, *lb=3l.*

Thus, the chromosome structure of an L*A 2-d lattice CA may be described by the following equations:

$$VL_i = [b_{4i+1}, b_{4i+2}, b_{4i+3}, b_{4i+4}], \forall i = 0,1,..,(vl-1)$$

$$T_{i-j} = [b_{vlb+4i+2j}, b_{vlb+4i+2j+1}], \forall i = 0,1,...,(vl-1) \;\&\; j = 0,1$$

$$L_{i-j} = [b_{vbl+tb+4i+3j}, b_{vbl+tb+4i+3j+1}, b_{vbl+tb+4i+3j+2}], \forall i = 0,1,...,(vl-1) \;\&\; j = 0,...,A-3$$

where VL_i, $T_{i\text{-}j}$, *and* $L_{i\text{-}j}$ constitute the three parts of the chromosomes.

The chromosome length or the total number of bits for each chromosome, *cl* is given by the equation:

$$cl = vlb + 2t + 3l\ .$$

For example, in a 10*5 2-d lattice CA with 2 VLs, the chromosome structure is described by the following:

$$VL_0 = [b_0, b_1, b_2, b_3]\,,$$

$$VL_1 = [b_4, b_5, b_6, b_7]\,,$$

$$\text{t cells at}\, VL_0 :\ T_{0-0} = [b_8, b_9]\ ,\ T_{0-1} = [b_{10}, b_{11}]\,,$$

$$\text{t cells at}\, VL_1 : T_{1-0} = [b_{12}, b_{13}]\,,\ T_{1-1} = [b_{14}, b_{15}]\,,$$

$$\text{lattice cells at}\, VL_0 : L_{0-0} = [b_{16}, b_{17}, b_{18}]\,,\ L_{0-1} = [b_{19}, b_{20}, b_{21}]\,,\ L_{0-2} = [b_{22}, b_{23}, b_{24}]\,,$$

$$\text{lattice cells at}\, VL_1 : L_{1-0} = [b_{25}, b_{26}, b_{27}]\,, L_{1-1} = [b_{28}, b_{29}, b_{30}]\ ,\ L_{1-2} = [b_{31}, b_{32}, b_{33}]\,,$$

$$cl = 34.$$

References

1. P. Hellekalek, *In Mathematics and Computers in Simulation*, 46, pp. 485-505 (1998).

2. S. Wolfram, *Cryptography with Cellular Automata, Advances in Cryptography*: *Proceeding*

of CRTPTO 85, *Lecture Notes in Computer Science,* Vol. 218 (1985), pp. 429–

432.

3. S. Wolfram, *Theory and Applications of Cellular Automata: Including Selected Papers 1983-1986* (World Scientific Publishing Co., Inc., River Edge, N.J., 1986).

4. M. Matsumoto, *ACM Trans. Modeling and Computer Simulation* 8(1), 31 (1998).

5. M. Mihaljevic, *Proc. Applied Algebra*, *Algorithms and Error Correcting Codes*, *Lecture*
notes in Computer Science, Vol. 1255 (1997), pp. 250–262.

6. M. Mihaljevic and H. Imai, *IEICE Trans. Fundamentals* E82-A(1), 32 (1999).

7. M. Tomassini, M. Sipper, M. Zolla, and M. Perrenoud, *Future Generation Computer Systems* 16, 291 (1999).

8. M. Sipper and M. Tomassini, *Int. J. Mod. Phys.* 7(2), 181 (1996).

9. P. D. Hortensius, R. D. Mcleod, and H. C. Card, *IEEE Trans. Comput.* 38, 1466 (1989).

10. P. D. Hortensius, R. D. Mcleod, W. Pries, D. M. Miller, and H. C. Card, *IEEE Trans. Computer-Aided Design* 8(8), 842 (1989).

11. S. Nandi, B. K. Kar, and P. Pal Chaudhuri, *IEEE Trans. Comput.* 43, 1346 (1994).

12. S.U. Guan and S.K. Tan, *Knowledge-based Intelligent Information and Engineering Systemts, PT 1, Proceedings* (2003).

13. S.U. Guan, and S. Zhang, *IEEE Trans. Evolutionary Computation*. 7(1): 23-36 (2003).

14. S.U. Guan, and S. Zhang, *International Journal of Modern Physic C*, Vol. 13, No. 8 (2002).

15. J.C. Isaacs, R.K. Watkins, and S.Y. Foo, *Engineering Applications of Artificial Intelligence*, 16 (5-6): 491-499 (2003).

16. D. R. Chowdhury, I. S. Gupta, and P. P. Chaudhuri, *J. Electrical Testing*: *Theory and Applications* 5, 65 (1994).

17. M. Tomassini, M. Sipper, and M. Perrenoud, *IEEE Trans. Comput.* 49, 1146 (2000).

18. J. Von Neumann, *J. von Neumann Collected Works*, ed. A. Taub.

19. S.U. Guan, S. Zhang, and M.T. Quieta, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23-3 (2004).

20. B. Shackleford,, M. Tanaka, R.J. Carter, and G. Snider, *In Proc. of the 2002 NASA/DOD Conference on Evolvable Hardware* (2002).

21. B. Shackleford, M. Tanaka, R.J. Carter, and G. Snider, *Field-programmable Gate Arrays: Proceedings of the 2002 ACM/SIGDA 10th International Symposium*, pp. 106-112 (2002).

22. J.C. Isaacs, R.K. Watkins, and S.Y. Foo, *2001 MAPLD International Conference* (2001).

23. ENT test suite, http://www.fourmilab.ch/random.

24. G. Marsaglia, “Diehard”, http://stat.fsu.edu/~geo/diehard.html, 1998.