2-D CA Variation With Asymmetric Neighborship for Pseudorandom Number Generation

Sheng-Uei Guan, Shu Zhang, and Marie Therese Quieta

***Abstract*—This paper proposes a variation of two-dimensional (2-D) cellular automata (CA) by adopting a simpler structure than the normal 2-D CA and a unique neighborship characteristic—asymmetric neighborship. The randomness of 2-D CA based on asymmetric neighborship is discussed and compared with one-dimensional (1-D) and 2-D CA. The results show that they are better than 1-D CA and could compete with conventional 2-D CA under certain array setting, output method, and transition rule. Furthermore, the structures of 2-D CA based on asymmetric neighborship were evolved using some multiobjective genetic algorithm. The evolved 2-D CA could pass DIEHARD tests with only 50 cells, which is less than the minimal number of cells (i.e., 55 cells) needed for neighbor-changing 1-D CA to pass DIEHARD. In addition, a refinement procedure to reduce the cost of 2-D CA based on asymmetric neighborship is discussed. The minimal number of cells found is 48 cells for it to pass DIEHARD. The structure of this 48-cell 2-D CA is identical to that of the evolved $10 * 5$ 2-D CA, except that 2 horizontal cells in the evolved $10 * 5$ 2-D CA are removed.**

***Index Terms*—Asymmetric neighborship, cellular automata (CA), multiobjective genetic algorithm (MOGA).**

I. Introduction

In THE PAST ten years, one-dimensional (1-D) cellular automata (CA) PRNGs were studied extensively [10]–[13], [15]–[23]. However, 1-D programmable CA (PCA) still failed some randomness tests. To improve the randomness of 1-D PCA, the idea of controlling the status of CA cells has been proposed in [20]. In later papers [21] and [22], this idea was further refined to the concept of neighbor-changing CA (NCA). The randomness test results on the improved 1-D PCA showed that they are better than 1-D PCA. In [23], neighbor-changing 1-D PCA are evolved to its best performance with minimal cost. But it is not easy to implement in hardware design due to its irregular structures.

On the other hand, some researchers began to employ two-dimensional (2-D) CA in pseudorandom number generation to improve further the randomness quality of CA PRNGs. Tomassini *et al.* [14] evolved a 64-cell 2-D CA that could pass the DIEHARD test [7]. Though the structural complexity is higher in 2-D CA, test results showed that 2-D CA are comparable to neighbor-changing 1-D CA in randomness.

Manuscript received January 15, 2003; revised August 28, 2003. This paper was recommended by Associate Editor N. K. Jha.

The authors are with the Electrical and Computer Engineering Department, National University of Singapore, Singapore (e-mail: sg_1_1@yahoo.com).

Digital Object Identifier 10.1109/TCAD.2004.823344

In this paper, we propose a variation of 2-D CA that has an array structure and the interconnection among cells is simpler than that in a normal 2-D CA. Due to its special structure, the neighborship among some cells in this 2-D CA is asymmetric. Asymmetric neighborship is a unique property of 2-D CA. The structural complexity of asymmetric-neighborship 2-D CA could be as simple as 1-D CA, except that the neighbor connection of boundary cells is more complex. The ENT [25] and DIEHARD test results on this structure of 2-D CA with asymmetric neighborship show that their randomness could compete with neighbor-changing 1-D CA and normal 2-D CA under certain array setting, output method, and transition rule. Further, multiobjective genetic algorithm (MOGA) is applied to evolve its structure. The evolved $10 * 5$ 2-D CA (with asymmetric neighborship) could generate good random number sequences that pass DIEHARD. Also, we try to reduce the number of cells in the evolved $10 * 5$ 2-D CA (with asymmetric neighborship) while maintaining their randomness quality.

To avoid ambiguity, 2-D CA described in this paper refers to a 2-D CA with asymmetric neighborship property and with such a simple array structure than a normal 2-D CA, which will be discussed later.

The paper is organized as follows. In Section II, we first give an overview on 1-D CA PRNGs, introduce structure and properties of 2-D CA PRNGs, and present randomness test results on $10 * 5$ 2-D CA without using an evolution algorithm. Section III presents the evolution algorithm and randomness results on the evolved 2-D CA. Section IV discusses the process to minimize the cost of 2-D CA. Section V provides a conclusion.

II. CA PRNGs

A. 1- and 2-D CA PRNGs

Following the idea of uniform CA, Hortensius [17] studied rule 90–150 programmable CA (PCA) and rule 30–45 PCA. Their study showed that rule 90–150 PCA has better potential than rule 30–45 PCA in pseudorandom number generation. These two PCAs are 1-bit PCA, where the rule control signal for each programmable cell is 1-bit. Later in 1996, Sipper and Tomassini [15] evolved a 2-bit 50-cell PCA with a mélange of rule 90, 150, and 165, where the rule control signal for each programmable cell is 2-bit. Also, Tomassini *et al.* [13] evolved another 2-bit 50-cell PCA with the rule combination 90, 105, 150, and 165 in 1999. These two 2-bit PCA were evolved

using a cellular programming evolutionary algorithm, while the two 1-bit PCA proposed by Hortensius were handcrafted. The DIEHARD test results showed that although 2-bit PCA are better than 1-bit PCA in randomness, they still fail to pass all the tests in DIEHARD with time spacing as 1.

To further improve the randomness quality of nonuniform CA, we proposed controllable cellular automata (CCA) in [22]. A CCA is a CA in which the action (how the state of a cell is updated in each cycle) of some cells can be controlled via cell control signals. If a cell is under control via some cell control signal, it is a *controllable cell*; otherwise, it is a *basic cell*. CCA is the combination of controllable cells and basic cells. Both controllable cells and basic cells have rule control signals.

The action of a controllable cell is decided by its current cell control signal. A controllable cell can be normal (when the cell control signal is 0) or active (when the cell control signal is 1). When the controllable cell is normal, the computation of the states of the controllable cell and its neighbors are as usual (according to the current rule control signals and the states of its neighbors). When the controllable cell is active, the state computation of the controllable cell and its neighbors are specified by some predefined action. If an active controllable cell keeps its latest state while its neighbors bypass it, it is a neighbor-changing controllable cell. The randomness and properties of neighbor-changing 1-D CA have been discussed in [21]–[23]. The randomness of neighbor-changing 1-D CA is the best among 1-D CA PRNGs but it is not ideal for very large scale integrated design due to its irregular structures.

All of the 1-D CA introduced above are three-neighborhood CA, i.e., state computation depends on the left and right neighbors in addition to itself. In [1], Barry *et al.* expanded the neighborhood size to four and introduced a nonlocal neighborhood connection scheme. They found a number of 64-cell 1-D CA PRNGs passing DIEHARD through exhaustive searching.

CA with 2-D grids are conventional or normal 2-D CA. Generally, two types of neighborhood are considered in a normal 2-D CA: five cells, consisting of the cell along with its four immediate nondiagonal neighbors (known as the von Neumann neighborhood) and nine cells, consisting of the cell along with its eight surrounding neighbors (known as the Moore neighborhood). In this paper, 2-D CA refers to 2-D CA with the von Neumann neighborhood.

The first work on these 2-D grid CA PRNGs was done by Chowdhury *et al.* [5] in 1994. Their results suggested that normal 2-D CA structures are superior to 1-D ones with the same size in pseudorandom number generation. Following their idea, Tomassini *et al.* [14] evolved several 8×8 2-D CA with rule 15, 63, 31, and 47. Their DIEHARD test results showed that some of the evolved 2-D CA could pass all the tests in DIEHARD. Different from 1-D CA, spacing was not used in conventional 2-D CA. Apparently, nonspacing results in higher output efficiency. Compared with 1-D CA, the structure of normal 2-D CA cells and the interconnection among them are more complex than 1-D CA because the number of neighbors is increased from three to five in 2-D CA cell.

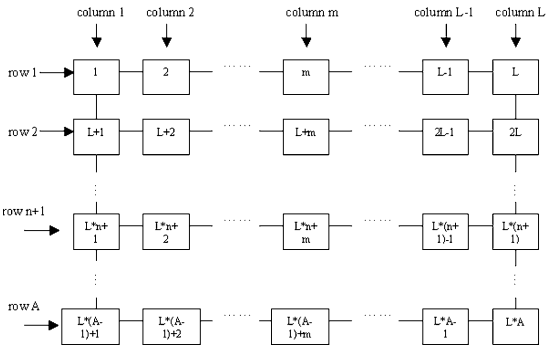

Fig. 1. Basic 2-D CA model.

B. Structure and Properties of 2-D CA With Asymmetric-Neighborship

A simple structure that a 2-D CA can take is an array of several 1-D CA with the boundary cells connected. Or it could be regarded as a normal 2-D CA with all the internal vertical neighbors connections disconnected as shown in Fig. 1.

According to the location of 2-D CA cells, we could place them under three categories:

1) corner cell: the four cells at the array corner;
2) vertical boundary cell: except corner cells, all the cells in the first and last columns are vertical boundary cells;
3) horizontal cell: except corner cells and vertical boundary cells, all the remaining cells in 2-D CA are horizontal cells.

Note that all 2-D CA described in this paper use the array structure presented in Fig. 1.

Aside from the vertical boundary cells, all cells have two directly connected cells, which would consequentially be their left and right neighbors. In the case of vertical boundary cells, they have three directly connected cells. Choosing left and right neighbors from these three cells is another concern. We define *consistent neighborship* as follows. In a column of vertical boundary cells, each cell uses the same neighboring cell position relative to it, as its left and right neighbors. As an example, each of the cells in the first column of array would treat the cells connected directly below and right to it as its left and right neighbors respectively. Simply put, neighbor assignment has a pattern for each column of cells. This is called consistent neighborship. On the other hand, if the cells in one column do not follow a uniform pattern of neighbor assignment, it is called *inconsistent neighborship.* Inconsistency in neighborship is described column-wise, i.e., in the same CA, one column may exhibit consistent neighborship while another column may use inconsistent neighborship.

Using the basic model, neighborships of cells were evolved using a genetic algorithm. From the basic model, only neighborships of vertical boundary cells can be varied. Thus, only vertical boundary cell neighborships were evolved. The configurations obtained are tested using ENT test. The results showed

that inconsistent neighborship performs better than a consistent one (see the Appendix).

To provide a rational analysis, we summarize our approach as follows. First, we start by introducing three different structures under the basic model, which uses consistent neighborship. Next, we analyze these structures by gradually increasing its complexity through connections, i.e., enhancing the basic model. Then, we involve some new CA properties called asymmetric neighborship property. Finally, we find good CA structures by using another genetic algorithm. Evolving CA structure means finding good neighborship of cells, which may be inconsistent.

According to the neighbor connection of vertical boundary cells, 2-D CA may have three different structures under the basic model: 2-D CA-1, 2-D CA-2, and 2-D CA-3. The difference lies in which of the three possible neighbor connections of the vertical boundary cells are chosen as left and right neighbors. Note that we use consistent neighborship for simplicity. Anyhow, we still evolve the neighborship of cells as we draw near to our conclusion.

Though the basic model has lower complexity than a normal 2-D CA, some preliminary experiments showed that the randomness of these 2-D CA (2-D CA-1, 2-D CA-2, 2-D CA-3) are worse than 1-D uniform CA. Compared with neighbor-changing 1-D CA, the randomness of the basic 2-D CA is far behind. As we have known, increasing the structural complexity of CA could improve the randomness quality. This has been proven from the development of 1-D uniform CA to neighbor-changing 1-D CA. Hence, we extend the neighbor connection of corner cells and vertical boundary cells in the basic model and derive an enhanced model of 2-D CA.

This enhanced 2-D CA model is shown in Fig. 2. Some initial experiments (see Appendix) showed that by using additional diagonal connection in the corner cells, randomness may be improved. To further improve the randomness of the basic 2-D CA, we expand the neighbor connection of corner cells and vertical boundary cells to diagonal direction while keeping the neighbor connection of horizontal cells remain the same. All the cells are uniform cells and have two neighbors. The neighbor connection of corner cells and vertical boundary cells are more flexible.

Each corner cell could be connected to three cells as shown in Fig. 2. Each vertical boundary cell could be connected to five cells. For example, vertical boundary cell $\mathrm{L}+1$ could be connected to cell $\mathrm{L}+2$, cell 1, cell $2\mathrm{L}+1$, cell 2, and cell $2\mathrm{L}+2$. But the number of neighbors for each cell remains as three. Hence, for corner cells and vertical boundary cells, there exists more than one combination to select neighbors from more than two connected cells. For each corner cell, there are three combinations to select two neighbors from three cells. For each vertical boundary cell, there are ten combinations to select from five cells.

In the following, we present two 2-D CA structures with different neighbor connections—2-D CA-4 and 2-D CA-5. In 2-D CA-4, the left neighbor of a corner cell is the one next to it the same row; the right neighbor is the one next to it in the diagonal direction. The left neighbor of a vertical boundary cell is the cell next to it in the same row; the right neighbor is the cell above it in the same column. Note that there is no neighbor connection in the diagonal direction for vertical boundary cells but one for each corner cell. The neighbor connection of 2-D CA-5 is shown in Fig. 3(b). There is no neighbor connection in the diagonal direction for corner cells but one for each vertical boundary cell.

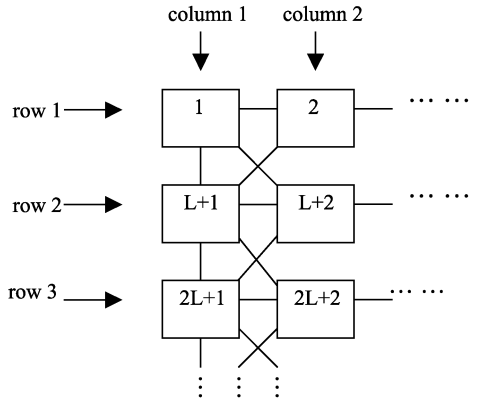

Fig. 2. Enhanced 2-D CA model.

Note that in 2-D CA-4 and CA-5, cell A is a neighbor of cell B does not imply cell B as a neighbor of cell A. This is different from the neighborship in conventional 1-D/2-D CA, in which two cells are neighbors of each other under periodical boundary conditions. In 2-D CA-4 & CA-5, neighborship is not symmetric. This type of neighborship is called as asymmetric neighborship. 2-D CA-1, CA-2, and CA-3 are all 2-D CA based on asymmetric neighborship. For example, in 2-D CA 3, each vertical boundary cell is a neighbor of the cell next to it in the same row while the reverse is not true. Compared with 2-D CA-4 & CA-5 based on the enhanced model, there are fewer asymmetric neighbor connections in the basic model. 2-D CA-1, CA-2, and CA-3 have the simplest structures that adopt the asymmetric neighborship property.

C. Parameters

Randomness varies with different structures—the neighbor connection of cells. To show this, the randomness of different 2-D CA structures is compared with 1-D uniform CA and neighbor-changing 1-D CA. Other than structure, there are several other factors that could affect the randomness quality of 2-D CA with fixed number of cells. They are setting of L and A (in short, array setting), transition rule selected (rule selection), and output method. Our methodology in studying these issues is to focus on one factor while fixing the other two at one time. Here, we study array setting first, with rule selection and output method following.

Because 2-D CA-1, CA-2, and CA-3 obtain similar randomness, we choose 2-D CA-3 as an example to study the randomness of 2-D CA based on the basic model. On the other hand,

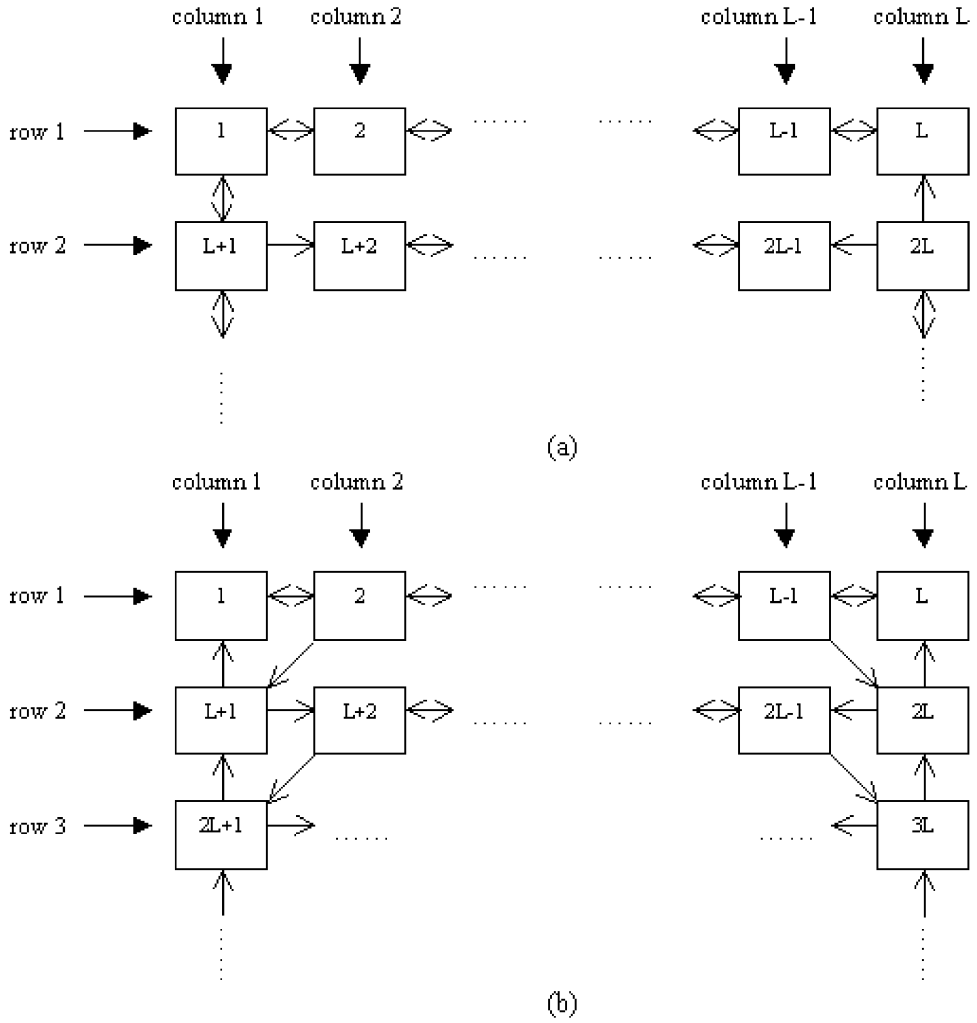

Fig. 3. Some examples of neighbor connections in 2-D CA: (a) neighbor connections in 2-D CA-3, (b) neighbor connections in 2-D CA-5. An arrow pointing from cell A to cell B means cell A is a neighbor of cell B.

we choose 2-D CA-5 as an example to study the randomness of 2-D CA based on the enhanced model.

Array Setting: Most work on conventional 2-D CA was done on regular settings like $4 * 4$, $6 * 6$, and $8 * 8$. Such settings may not be the best choice for our 2-D CA. In this section, we study the randomness of all possible settings in our 2-D CA, based on the basic and enhanced models to find a suitable setting for L and A. The value of A decides the number of vertical boundary cells and the value of L decides the number of horizontal cells in each row. Considering a basic 2-D CA with a fixed number of cells, if L is too large, there will be few vertical boundary cells. We may deduce that maintaining a sufficient number of vertical boundary cells in 2-D CA could be crucial to obtain good randomness quality. On the other hand, if A is too large, the number of horizontal cells in each row will be too small to maintain the randomness quality of horizontal cells. Thus, a suitable choice of L and A will be critical to generate good 2-D CA PRNGs.

Rule Selection: Although the overall structure of the basic and enhanced CA is 2-D, the structure of each horizontal cell is identical to that of a cell in 1-D uniform CA in terms of cell structure and neighbor connection. Work on 1-D uniform CA has shown that their randomness depends highly on the rules used [19]. Due to their structure similarity, it won't be surprising to see that the randomness of 2-D CA also varies under different rules.

Output Methods: We already know that in 1-D CA, adjacent cells have correlations because bits are generated by the horizontal cells. In 2-D CA, we choose to avoid generating the output bits in the horizontal direction. Hence, the random number sequences could be recorded in many methods. Our study is to find out the output bits should be extracted in the vertical direction or horizontal direction. The following methods have been tested. Fig. 4 shows each output method.

D. Randomness Test Results

The randomness of $10 * 5$ CA-1, CA-2, CA-3, CA-4, and CA-5 is shown in Table I. The experiment conditions are identical for the 2-D CA tested. The output sequences are generated by all the cells in the 2-D CA, recorded from the first cell to the last one. The length of the tested sequences for each

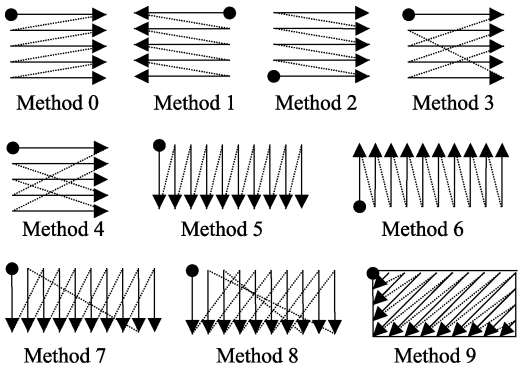

Fig. 4. Diagrams of method 0–method 9.

TABLE I
ENT TEST RESULTS ON CA-1, CA-2, CA-3, CA-4, AND CA-5

*L*A*	*entropy*	*1-SCC*	*chi-square*
50-cell 1-D uniform CA	7.982535	0.994142	0.65
10*5 2-D CA –1	7.835810	0.980746	0.0
10*5 2-D CA -2	7.817134	0.974727	0.0
10*5 2-D CA -3	7.608064	0.972144	0.0
10*5 2-D CA -4	7.882832	0.986124	0.855
10*5 2-D CA -5	7.981707	0.992714	0.96
50-cell uniform 1-D NCA	7.980437	0.993134	0.98

CA is 10 000 bytes. 100 initial seeds are tested for each individual CA. The transition rule used is rule 150. Both of them obtain better randomness than 1-D uniform CA. Table I shows that 2-D CA under the basic model could not outperform 1-D uniform CA. The comparison between the basic model (CA-1, CA-2, CA-3) and the enhanced model (CA-4, CA-5) shows that the randomness of 2-D CA is improved under the enhanced model. Yet when compared with neighbor-changing 1-D CA, the randomness of CA-4 and CA-5 is still lower, which may mean that these CA need to be further evolved to compete with neighbor-changing 1-D CA. The randomness of CA-5 is better than that of CA-4. Considering that the structural complexity of CA-5 is higher too, we could deduce that the randomness of these CA is closely related to their complexity levels.

Using CA-3 and CA-5 as examples, 50-cell 2-D CA is studied under various array settings. Table II shows the ENT test results. The experiment conditions are identical for the 2-D CA tested and the same as that in Table I. The results show that the randomness of 2-D CA varies a lot under different array settings. The randomness of CA-5 is better than that of CA-3 under most array settings. One reason could be that there are more asymmetric neighbor connections in CA-5. Another reason may be due to the neighbor connections in the diagonal direction in CA-5.

In 36-cell 2-D CA, $9 * 4$ 2-D CA-5 can get the best randomness. In 50-cell 2-D CA, $10 * 5$ 2-D CA-5 can get the best randomness. In 64-cell 2-D CA, $16 * 4$ 2-D CA-5 can get the best randomness. It shows that regular size is not the best choice in 2-D CA. In conclusion, $10 * 5$ CA-5 obtains the best ENT results. Our study in the following will focus on $10 * 5$ CA-5.

TABLE II
ENT TEST RESULTS OF 50-CELL CA

*L*A*	*entropy*	*1-SCC*	*chi-square*
50-cell 1-D uniform CA	7.981345	0.991525	0.685
2*25 2-D CA-3	7.978867	0.992828	0.63
2*25 2-D CA-5	7.980220	0.992639	0.765
5*10 2-D CA-3	7.603398	0.963734	0.0
5*10 2-D CA-5	7.978731	0.992862	0.68
10*5 2-D CA-3	7.610381	0.964993	0.0
10*5 2-D CA-5	7.981707	0.992714	0.96
25*2 2-D CA-3	7.981585	0.990267	0.835
25*2 2-D CA-5	5.656873	0.833827	0.0
50-cell uniform 1-D NCA	7.981728	0.992898	0.985

TABLE III
ENT TEST RESULTS OF 2-D CA-3/5 UNDER DIFFERENT RULES

	*L*A*	*entropy*	*1-SCC*	*chi-square*
rule-150	10*5 2-D CA-3	7.607533	0.969035	0.0
	10*5 2-D CA-5	7.981478	0.990234	0.94
	50-cell 1-D uniform CA	7.981612	0.990887	0.65
rule-90	10*5 2-D CA-3	7.784564	0.980134	0.0
	10*5 2-D CA-5	7.981331	0.991578	0.86
	50-cell 1-D uniform CA	7.981898	0.990731	0.76
rule-30	10*5 2-D CA-3	3.131734	0.900043	0.0
	10*5 2-D CA-5	5.042612	0.847323	0.005
	50-cell 1-D uniform CA	7.979645	0.992367	0.66
rule-105	10*5 2-D CA-3	7.582839	0.965751	0.0
	10*5 2-D CA-5	7.981503	0.991421	0.93
	50-cell 1-D uniform CA	7.981508	0.992467	0.72
rule-165	10*5 2-D CA-3	7.874589	0.979989	0.0
	10*5 2-D CA-5	7.981501	0.991564	0.92
	50-cell 1-D uniform CA	7.981932	0.990767	0.76

In the previous experiments, we use rule 150. In the following, we test the randomness of $10 * 5$ 2-D CA-3 and CA-5 under different rules. We test rule 30, 90, 105, and 165 because they are reported as the best rules for 1-D uniform CA [17]. Table III shows the ENT test results. The experiment condition is the same as that in Table I and Table II.

The ENT test results show that 2-D CA-3 cannot generate good random number sequences under all the rules tested, while 2-D CA-5 outperform 1-D uniform CA under all the transition rules except rule 30. Generally, rule 30 is not a good choice for $10 * 5$ 2-D CA, but we cannot exclude the possibility that it could generate good results under other array settings and output methods. Comparison on the remaining rules shows that the randomness improvement of 2-D CA-5 is most prominent with rule 150. Hence, we choose to apply rule 150 on 2-D CA cells in our work.

The ENT/DIEHARD test results of 2-D CA-5 under output method 0 to 9 are shown in Table IV. The experiment condition is the same as that in Tables I–III. The length of sequences for DIEHARD test is 10 000 000 bytes. It shows that method 5–8 could generate better random number sequences than the other methods, which shows that output in the vertical direction is better than output in the horizontal or diagonal direction. We can deduce that output bit extraction in the vertical direction

TABLE IV
ENT/DIEHARD TEST RESULTS OF 2-D CA-5
UNDER DIFFERENT OUTPUT METHODS

output method		*entropy*	*1-SCC*	*chi-square*	*diehard*
50-cell 1-D uniform CA		**7.981109**	**0.991703**	**0.75**	**4**
10*5 2-D CA-5	**Method 0**	**7.981936**	**0.992653**	**0.94**	**16**
	Method 1	**7.981733**	**0.992253**	**0.945**	**16**
	Method 2	**7.981522**	**0.992215**	**0.93**	**16**
	Method 3	**7.981465**	**0.992314**	**0.90**	**16**
	Method 4	**7.981395**	**0.991767**	**0.935**	**16**
	Method 5	**7.981579**	**0.992905**	**0.915**	**17**
	Method 6	**7.981358**	**0.992534**	**0.955**	**17**
	Method 7	**7.981256**	**0.991365**	**0.95**	**17**
	Method 8	**7.981530**	**0.992886**	**0.945**	**17**
	Method 9	**7.981631**	**0.991463**	**0.895**	**16**
50-cell 1-D NCA without spacing		**7.981205**	**0.991787**	**0.98**	**17**

presents lesser correlation than in the horizontal direction. Thus, we choose to use vertical direction in generating the output. The performance of method 5–7 is similar while the implementation of method 5 is a more natural way. Hence, we choose method 5 as the output method in the following experiments.

In the following experiments, all the 2-D CA are $10 * 5$ 2-D CA using rule 150 and output method 5. The best randomness quality of 2-D CA is obtained by 2-D CA-5, failing one test in DIEHARD. But we have shown in [22] that 1-D 50-cell NCA could pass all the tests in DIEHARD with time spacing at 1. The question is can we find other structures that generate better random number sequences than 2-D CA-5 and at the same time pass DIEHARD? In the next section, the structure of 2-D CA is to be evolved.

III. EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION APPROACHES

The objective of the evolution process is to find some 2-D CA that could generate good random number sequences. Because DIEHARD test is too time-consuming, we use ENT test results to evaluate the randomness of these CA in this work. Different from Tomassini *et al.*'s method [14] that uses entropy value alone as the objective to evolve 2-D CA, we choose to use entropy value along with SCC and chi-square values as objectives to get a more comprehensive evaluation. Some 2-D CA may get good entropy values but fair SCC values while some other 2-D CA may get reverse results. Because entropy, SCC, and chi-square values may not be aligned, we cannot simply add them together to generate one single value as objective. Hence, single-objective evolution algorithm does not make sense here.

MOGA is then used to evolve the structure of 2-D CA. MOGA is widely used to solve engineering problems where simultaneous optimization of multiple, often competing, objectives is required. Various schemes have been developed in recent years [6]. These techniques could be divided into two categories: the population-based approach and the Pareto-based approach. On the whole, the population-based approach has a common deficiency that it tends to generate solutions such that the performance of one objective is extremely good while the other objectives are not so [6]. Hence, we use the Pareto-based approach.

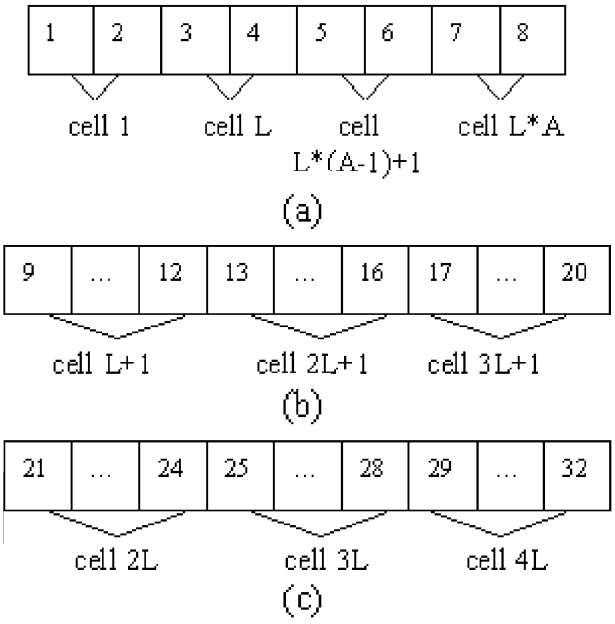

Fig. 5. The chromosome structure: (a) bits 1–8 for four corner cells; (b) bits 9–20 for three vertical boundary cells in the first column; and (c) bits 21–32 for three vertical boundary cells in the last column.

Different from the population-based approaches, the Pareto-based approach performs selection/reproduction by referring not only to the objective values themselves but also to the dominance property of them. Among several proposed schemes, we choose Fonseca and Fleming's [2] as our basic algorithm. In their scheme, the rank of each individual is defined as one plus the number of individuals in the current population that dominates it.

A. Evolutionary Algorithm—MOGA

The structure of an individual $10 * 5$ 2-D CA includes three parts: the interconnection of horizontal cells and their neighbors, the interconnection of the four corner cells and their neighbors, and the interconnection of six vertical boundary cells and their neighbors. Because the interconnections of horizontal cells and their neighbors are fixed in these CA, the structure of $10 * 5$ 2-D CA is decided by the interconnection of the four corner cells, the six vertical boundary cells and their neighbors.

For each corner cell, there are three possible combinations and for each vertical boundary cell, there are ten possible combinations. Different corner cells could have different neighbor connections and the same for vertical boundary cells. Hence, we have two bits for each corner cell and four bits for each vertical boundary cell in the chromosome to identify their neighbor connections Fig. 5 shows the structure of the chromosome. Bits 1–8 stand for the neighbor connection of

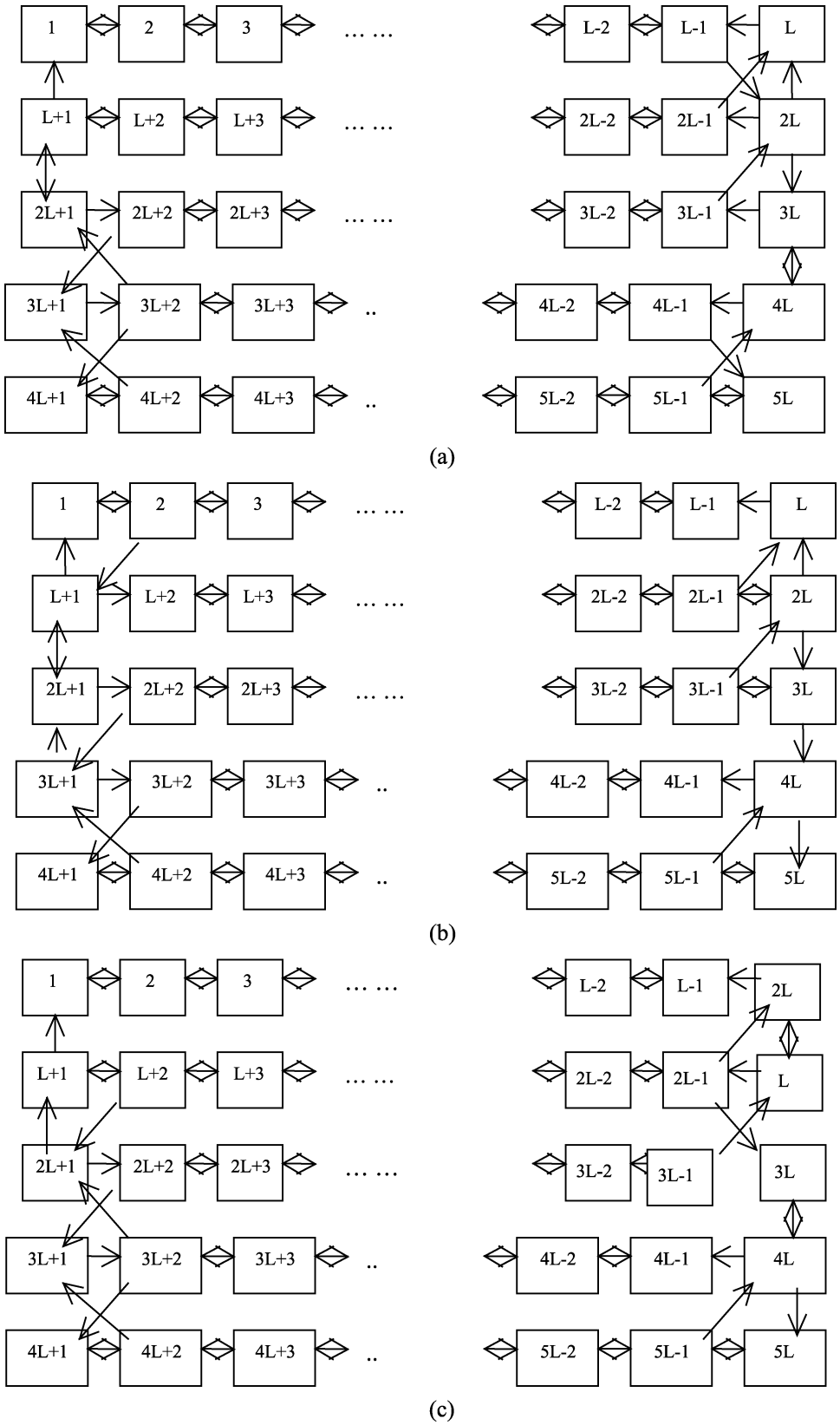

Fig. 6. Structures of the three evolved $10 * 5$ 2-D CA. (a) Structure of chromosome 1. (b) Structure of chromosome 2. (c) Structure of chromosome 3. An arrow pointing from cell A to cell B means cell A is a neighbor of cell B.

four corner cells, bits 9–20 stand for the neighbor connection of three vertical boundary cells in the first column, and bits 21–32 stand for the neighbor connection of three vertical boundary cells in the last column.

The randomness of 2-D CA is evaluated using ENT. 100 initial seeds are tested and the average performance is taken as the randomness of the tested CA. The three objectives are the average entropy value, average 1-SCC value, and the chi-square pass rate. The relationship of these objectives has been extensively studied in [22]. Each 32-bit chromosome identifies one CA structure. The detail is presented in Algorithm 1.

Algorithm 1: Evolution of the 2-DCA structure

Input: P chromosomes randomly initiated.
//***evolution***
While (stopping criteria is not true) do
  Calculate the objective values of each chromosome;
  Calculate the Pareto-rank of each chromosome;
  Perform crossover and mutation to generate P child chromosomes;
  Calculate the objective values of each child chromosomes;
  Calculate the Pareto-rank of each child chromosome;
  Copy the first half of parent chromosomes and first half of child chromosomes to the next generation;
End while
Output P chromosomes in the last generation

The input of the evolution process is P chromosomes that are randomly initiated. The output is P chromosomes in the last generation. The total population size P is set at 80. The stopping criterion is the maximum stagnation steps (T), which is set at 300 $(T = 300)$. Or, if the best chromosome keeps unchanged for 200 continuous evolution steps, the evolution process will be stopped. The 1-point crossover rate is set at 0.95. The bit mutation rate is set at 0.05. During reproduction, half of the better-performing parents and child chromosomes will be copied into the next generation.

B. Evolution Results

The final objective of the evolution process is to find some 2-D CA to pass DIEHARD. However, due to the limitation of the computation ability, we use ENT test results to evaluate the randomness of these 2-D CA during the evolution process. DIEHARD test is applied to the whole population every ten evolution steps to check whether the evolved CA could pass DIEHARD. In the last population, three out of 80 chromosomes are found to pass DIEHARD. These three chromosomes are considered as the final evolution results. The structures of the evolved CA are shown in Fig. 6. It shows that neighbor connections in the diagonal direction are indispensable to obtain good randomness quality in these CA. The structural complexity of the evolved three CA is similar.

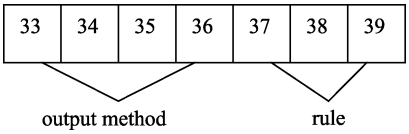

Fig. 7. Bits 33–39 in the expanded chromosome.

Compared with 50-cell 1-D NCA, the randomness of the evolved $10 * 5$ 2-D CA is better because 50-cell 1-D NCA could not pass DIEHARD without any spacing. According to the work of Tomassini *et al.* [14], an evolved $8 * 8$ 2-D CA could pass DIEHARD without any spacing while the randomness of their 2-D CA with smaller size is not known yet. We may conclude that our 2-D CA could be as good as the conventional 2-D CA at least. But the structural complexity of our 2-D CA is lower than that of the conventional 2-D CA with the same size.

C. Discussion on Evolution Results

In this section, we evolve transition rule and output method along with our 2-D CA structures to find whether output method 5 and rule 150 are the best choice as indicated in Section II. The chromosome shown in Fig. 5 is expanded with seven additional bits. Fig. 7 shows the expanded part of the chromosome. Bits 33–36 represent the output method 0–9. Bits 37–39 represent the eight rules. The evolution algorithm is the same as Algorithm 1. All the evolution parameters are the same too.

Evolving $10 * 5$ 2-D CA for 1000 steps, we find that all the chromosomes in the last population use rule 150. Most of the chromosomes use output method 5, while a few use output method 6. The randomness of these few chromosomes is further tested with DIEHARD under output method 5 and 6. Their DIEHARD test results are the same under the two output methods although they obtain better ENT test results under output method 6. In addition, we evolve output method and transition rule along with the structures of these $8 * 6$ 2-D CA. Similar results are obtained for $8 * 6$ 2-D CA. Rule 150 is used by all the chromosomes in the last generation and output method 5 is used by most of them. The evolution results on these $10 * 5$ and $8 * 6$ CA confirm our previous suggestion that output method 5 and rule 150 are the best choice for $10 * 5$ CA or CA under similar array settings.

Referring to the structures of three evolved $10 * 5$ CA presented in the last section, we can see that their complexity is close. It is possible to find some other $10 * 5$ CA with less complex structures to pass DIEHARD by increasing evolution effort. Fig. 8 shows the number of chromosomes passing DIEHARD during the evolution process.

Using DIEHARD to test the chromosomes every ten evolution steps, we can see that the number of chromosomes passing DIEHARD increases. However, after approximately 150 time steps, i.e., from time steps 250 to 400, the number of chromosomes that can pass DIEHARD keeps unchanged. Given that our main objective is to find a structure that can pass DIEHARD, we are contented with getting a satisfactory result at lesser evolution time. This is because the evolution process is time-consuming

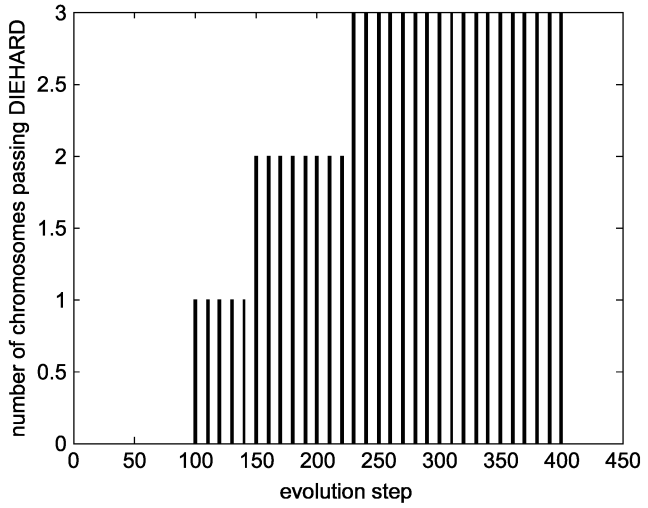

Fig. 8. Number of chromosomes passing DIEHARD at different evolution steps.

TABLE V
RANDOMNESS OF $10 * 5$, 49-CELL, 48-CELL, 47-CELL CA WITH ASYMMETRIC NEIGHBORSHIP

Array size		chi-square	entropy	1-scc	diehard
10*5 CA		0.937	7.981508	0.992002	18
49-cell CA	10*5 – (0,5)	0.946	7.981581	0.991706	18
	10*5 – (1,5)	0.933	7.981463	0.991813	18
	10*5 – (2,5)	0.9475	7.981480	0.992049	18
	10*5 – (3,5)	0.9405	7.981573	0.992040	18
	10*5 – (4,5)	0.9405	7.981498	0.991921	18
	10*5 – (0,6)	0.937	7.981562	0.991772	18
	10*5 – (0,3)	0.949	7.981545	0.991991	18
48-cell CA	10*5 – (0,5) & (3,6)	0.9315	7.981532	0.991905	18
	10*5 – (0,5) & (2,4)	0.93	7.981905	0.991619	18
	10*5 – (1,4) & (3,5)	0.931	7.981601	0.991897	18
	10*5 – (2,5) & (4,4)	0.932	7.981765	0.992019	18
	10*5 – (0,5) & (0,6)	0.90	7.981532	0.991905	16
	10*5 – (2,6) & (2,4)	0.91	7.981405	0.990619	16
47-cell CA	10*5 – (0,5) & (2,5) & (4,5)	0.911	7.981423	0.991822	16
	10*5 – (0,5) & (2,4) & (3,6)	0.901	7.981435	0.991798	16
	10*5 – (0,5) & (0,6) & (0,7)	0.91	7.981430	0.991820	16
	10*5 – (1,5) & (2,4) & (3,6)	0.90	7.981432	0.991768	16
	10*5 – (2,5) & (2,6) & (4,5)	0.91	7.981424	0.991702	16
9*5 2-D CA		0.92	7.981476	0.991802	16

and there is no guarantee that CA with simpler structures could be found in the end.

IV. REFINEMENT PROCEDURE TO REDUCE THE COST OF 2-D CA WITH ASYMMETRIC NEIGHBORSHIP

Since we know that $10 * 5$ CA with asymmetric neighborship could pass DIEHARD we explore whether the cost of CA could be reduced while maintaining the randomness quality instead of going on with the evolution. Here, we try to reduce the cost by decreasing the number of cells in the evolved $10 * 5$ CA. The evolved $10 * 5$ CA-1 (chromosome 1) is used as the basic structure in this section.

There is no doubt that the good performance of 2-D CA mainly comes from the versatile interconnections among corner cells/horizontal cells and their neighbors. Hence, it is natural to keep these neighbor connections unchanged during our exploration. We keep the number and location of corner cells and vertical boundary cells fixed while reducing the number of horizontal cells in the evolved CA. First, we test the randomness of $9 * 5$ CA by deleting one horizontal cell in each row. The ENT/DIEHARD test results are shown in Table V. It shows that the randomness of $9 * 5$ CA drops dramatically. It could only pass 16 tests in DIEHARD.

Because the randomness of $9 * 5$ 2-D CA is unsatisfactory, the procedure goes back to $10 * 5$ CA by reducing the number of horizontal cells in the evolved $10 * 5$ CA one by one. The difference between the evolved $10 * 5$ CA and 49-cell 2-D CA is that there is one horizontal cell less in the latter. Table V shows the ENT and DIEHARD test results on 49-cell 2-D CA. The results show that 49-cell 2-D CA could perform as well as $10 * 5$ CA and the location of the deleted horizontal cell may not affect the randomness.

Next, we test 48-cell 2-D CA by having two horizontal cells deleted from the $10 * 5$ CA. The randomness test results are shown in Table V too. It shows that 48-cell 2-D CA may pass DIEHARD and the performance depends on the location of the two horizontal cells deleted. If the two horizontal cells deleted are in the same row, the 48-cell 2-D CA could not pass DIEHARD. If the two horizontal cells deleted are located in different rows, the 48-cell 2-D CA could pass DIEHARD. Table V also shows the randomness of some 47-cell 2-D CA with three horizontal cells deleted. None of the tested 47-cell CA could pass DIEHARD no matter where the deleted horizontal cells are located. Because the searching process is not complete, we could not exclude the possibility that some 47-cell 2-D CA could pass DIEHARD. However, it would be straightforward to carry on the refinement procedure.

V. CONCLUSION

In this paper, we have introduced a variation of CA—2-D CA with an array structure and discussed the asymmetric neighborship in them. The array setting, rule selection and output method of such CA have been studied and some suggestions given. The structures of CA with asymmetric neighborship were evolved using MOGA. The evolved $10 * 5$ CA could generate good random number sequences that pass DIEHARD. Compared with 1-D NCA and normal 2-D CA in similar size, our 2-D CA could perform better or as well. Further, we explored the possibility to reduce the number of horizontal cells in the evolved CA with asymmetric neighborship. We found that two horizontal cells could be deleted from the evolved $10 * 5$ 2-D CA without degrading the randomness quality when the two horizontal cells are not deleted from the same row.

APPENDIX
CONSISTENT VERSUS INCONSISTENT NEIGHBORSHIP

Recall the difference between consistent and inconsistent neighborship by citing examples for each in Fig. 9. Note that cells 1–3 are vertical boundary cells in the first column and cells 4–6 are vertical boundary cells in the last column of 2-D CA. Genetic algorithm was used to derive the neighborship of vertical boundary cells that will give the best ENT results for a $10 * 5$ 2-D CA. Corner cells and horizontal cells are uniform cells and use their two directly connected cells as neighbors.

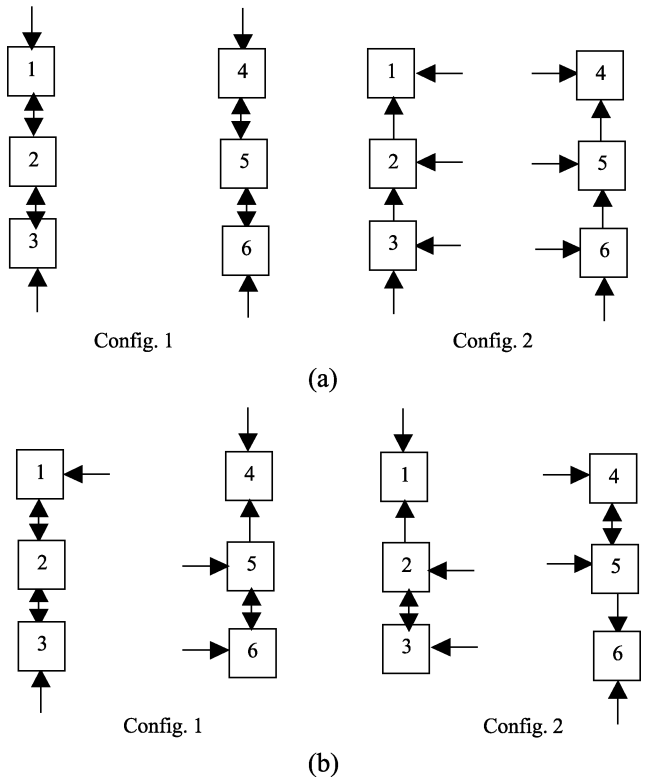

Fig. 9. Vertical boundary cell neighborship connection. (a) Consistent neighborship. (b) Inconsistent neighborship. An arrow pointing from cell A to cell B means cell A is a neighbor of cell B.

TABLE VI
CONSISTENT VERSUS INCONSISTENT NEIGHBORSHIP

	chi-square	*entropy*	*1-SCC*
Consistent Configuration 1	**0.0**	**7.603085**	**0.963640**
Consistent Configuration 2	**0.0**	**7.819772**	**0.970303**
Inconsistent Configuration 1	**0.937**	**7.981572**	**0.992048**
Inconsistent Configuration 2	**0.94**	**7.981458**	**0.992074**

TABLE VII
BASIC VERSUS ENHANCED MODEL

	diehard
Inconsistent Configuration 1 under the basic model	11
Inconsistent Configuration 1 under the enhanced model	14

Using ENT test to evaluate the performance, it is shown in Table VI that consistent neighborship outperforms consistent one.

BASIC VERSUS ENHANCED MODEL

Configuration 1 with inconsistent neighborship is used to compare the performance of enhanced model with the basic model. To describe the enhanced model, the basic model is modified such that the corner cells will have diagonal cells directly connected to it. The diagonal cells are used as neighbors. All other parameters including neighborship of vertical boundary cells remain the same. Under the DIEHARD test using ten initial seeds, it can be seen from Table VII that the enhanced model outperforms the basic model.

REFERENCES

[1] B. Shackleford, M. Tanaka, R. Carter, and G. Snider, "High-performance cellular automata random number generators for embedded probabilistic computing systems," in *Proc. NASA/DOD Conf. Evolvable Hardware*, 2002, pp. 191–200.

[2] C. M. Fonseca and P. J. Fleming, "Genetic algorithm for multiobjective optimization: Formation, discussion and generalization," in *Proc. 5th ICGA*, 1993, pp. 416–423.

[3] D. E. Knuth, *The Art of Computer Programming*, 3rd ed. Reading, MA: Addison-Wesley, 1998, vol. 2, Seminumerical Algorithms.

[4] D. J. Neeble and C. R. Kime, "Cellular automata for weighted random number generation," *IEEE Trans. Comput.*, vol. 46, pp. 1219–1229, Nov. 1997.

[5] D. R. Chowdhury, I. S. Gupta, and P. P. Chaudhuri, "A class of two-dimensional cellular automata and applications in random pattern testing," *J. Elect. Testing: Theory Applicat.*, vol. 5, pp. 65–80, 1994.

[6] D. Cvetkovic and I. Parmee, "Preference and application in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 42–57, Feb. 2002.

[7] Diehard, G. Marsaglia. (1998). Available: http://stat.fsu.edu/~geo/diehard.html [Online]

[8] J. Li and X. San, "Tree-structured linear cellular automata," in *Proc. Canad. Conf. Elect. Comput. Eng.*, vol. 1, May 1996, pp. 261–264.

[9] J. Li, K. Soon, and X. San, "Tree-structured linear cellular automata and their application as PRNGs," in *Proc. Int. Test Conf.*, Nov. 1997, pp. 858–867.

[10] M. Matsumoto, "Simple cellular automata as pseudorandom m-sequence generators for built-in self-test," *ACM Trans. Modeling Comput. Simul.*, vol. 8, no. 1, pp. 31–42, 1998.

[11] M. Mihaljevic, "Security examination of a cellular automata based pseudorandom bit generator using an algebraic replica approach," in *Proc. Appl. Algebra, Algorithms, Error Correcting Codes, Lecture Notes Comput. Sci.*, vol. 1255, 1997, pp. 250–262.

[12] M. Mihaljevic and H. Imai, "A family of fast keystream generators based on programmable linear cellular automata over GF(q) and time-variant table," *IEICE Trans. Fundamentals*, vol. E82-A, no. 1, pp. 32–39, 1999.

[13] M. Tomassini, M. Sipper, M. Zolla, and M. Perrenoud, "Generating high-quality random numbers in parallel by cellular automata," *Future Generation Comput. Syst.*, vol. 16, pp. 291–305, 1999.

[14] M. Tomassini, M. Sipper, and M. Perrenoud, "On the generation of high-quality random numbers by two-dimensional cellular automata," *IEEE Trans. Comput.*, vol. 49, pp. 1146–1151, Oct. 2000.

[15] M. Sipper and M. Tomassini, "Generating parallel random number generators by cellular programming," *Int. J. Modern Phys.*, vol. 7, no. 2, pp. 181–190, 1996.

[16] P. D. Hortensius, R. D. Mcleod, and H. C. Card, "Parallel random number generation for VLSI system using cellular automata," *IEEE Trans. Comput.*, vol. 38, pp. 1466–1473, Oct. 1989.

[17] P. D. Hortensius, R. D. Mcleod, W. Pries, D. M. Miller, and H. C. Card, "Cellular automata-based pseudorandom number generators for built-in self-test," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 842–859, Aug. 1989.

[18] S. Nandi, B. K. Kar, and P. P. Chaudhuri, "Theory and applications of cellular automata in cryptography," *IEEE Trans. Comput.*, vol. 43, pp. 1346–1357, Dec. 1994.

[19] S. Wolfram, *Theory and Applications of Cellular Automata: Including Selected Papers 1983–1986*, Singapore: World Scientific, 1986.

[20] S.-U. Guan and S. Zhang, "An encryption method based on dynamic cellular automata," in *Proc. Int. ICSC Congr. Intell. Syst. Applicat.*, Univ. Wollongong, Australia, #1514–074, Dec. 12–15, 2000.

[21] ——, "A family of controllable cellular automata for pseudorandom number generation," *Int. J. Modern Phys., Comput.*, vol. 13, no. 8, pp. 1047–1074, 2002.

[22] ——, "An evolutionary approach to the design of controllable cellular automata structure for random number generation," *IEEE Trans. Evol. Comput.*, pp. 23–36, Feb. 2003.

[23] ——, "Incremental evolution of cellular automata for random number generation," Int. J. Modern Phys. Comput., vol. 14, no. 7, pp. 881–886, 2003, to be published.

[24] J. Von Neumann, "The general and logical theory of automata," in *J. von Neumann Collected Works*, A. H. Taub, Ed. New York: Macmillan, 1951.

[25] ENT Test Suite Available: http://www.fourmilab.ch/random [Online]

Sheng-Uei Guan received the M.Sc. and Ph.D. degrees from the University of North Carolina, Chapel Hill.

He is currently with the Electrical and Computer Engineering Department at National University of Singapore. He has worked in a prestigious research and development organization for several years, serving as a Design Engineer, Project Leader, and Manager. He has also served as a Member on the Chinese Information and Communication National Standard Draft Committee. After leaving the industry, he joined Yuan Ze University, Taipei, Taiwan, for three and a half years. He served as Deputy Director for the Computing Center and as Chairman for the Department of Information and Communication Technology. Later, he joined the Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia, where he helped to create a new Multimedia Systems stream.

Shu Zhang received the B.Sc. degree in computer science from Huazhong University of Science and Technology, Hubei, China, and the M.Eng. degree from the National University of Singapore, Singapore.

She is a research engineer in the Department of Electrical and Computer Engineering at National University of Singapore. Her current research interests include cellular automata, evolutionary computation, and artificial intelligence.

Marie Therese Quieta received the B.Sc. in electronics and communications engineering from De La Salle University, Manila, Philippines, 2001. She is currently pursuing M.Eng. at the National University of Singapore, Singapore.

She is currently a Research Scholar under AUN/SEED-Net JICA Project. Her research interests include cellular automata, evolutionary computation, and neural networks.