One-Dimensional Cellular Automata and Shrinking Generators for Pseudorandom Sequence Generation

by

Nafeesa Sheikh
B.S., University of Fatima Jinnah, 2021

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

We acknowledge and respect the Lək'ʷəŋən (Songhees and Xwsepsəm/ Esquimalt) Peoples on whose territory the university stands, and the Lək'ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

One-Dimensional Cellular Automata and Shrinking Generators for Pseudorandom Sequence Generation

by

Nafeesa Sheikh
B.S., University of Fatima Jinnah, 2021

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Fayez Gebali, Departmental Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Fayez Gebali, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Linear feedback shift registers (LFSRs) based on primitive polynomials are commonly used to generate maximum length sequences (m-sequences). These pseudorandom sequences demonstrate desirable randomness properties such as balance, run, and autocorrelation while exhibiting low linear complexity. One-dimensional Cellular Automata (CA) are employed to produce m-sequences and pseudorandom sequences with high linear complexity and good randomness characteristics. This thesis explores the application of one-dimensional CA with shrinking generators to obtain sequences with high linear complexity and good randomness. Three types of shrinking generators are considered in this thesis. An analysis of the properties of the sequences obtained in relation to the corresponding m-sequences is given.

Contents

List of Tables

List of Figures

List of Acronyms

1D CA	One Dimensional Cellular Automata
AC	Autocorrelation
B	Balance
CA	Cellular Automata
LC	Linear Complexity
LR	Linear Rule
LFSR	Linear Feedback Shift Register
MSR	Maximum Sidelobe Ratio
m-sequence	Maximum Length Sequence
MSSG	Modified Self-Shrinking Generator
NSSG	New Self-Shrinking Generator
OC	Observed Cell
R	Run
RC	Randomized Cell
RR	Random Rule
S	Sequence
SSG	Self-Shrinking Generator
SV	Start Value

ACKNOWLEDGEMENTS

I am grateful to Allah who is the most merciful and my parents for their continuous support, love, prayers, and motivation. I wish to express my sincere gratitude to my supervisor Dr. T. Aaron Gulliver whose guidance, expertise, flexibility, and encouragement contributed greatly to my graduate studies. His expertise and knowledge were essential for the successful completion of my thesis. His flexibility allowed me to work remotely on my thesis. I would also like to thank Umer Khayyam for his help during my degree. I am also thankful to the University of Victoria for financial support and Compute Canada for providing computing resources that enabled me to generate the results for this thesis.

DEDICATION

To my parents, for their continuous support, love, and prayers. To my supervisor, Dr. T Aaron Gulliver for his constant support, guidance, and flexibility.

Chapter 1

Introduction

Randomness refers to a series of events that lack a discernible pattern or predictability. Random numbers, which can be generated through unpredictable processes are widely used in various fields including gaming, gambling, sports, and statistics [1]. For instance, in gaming, random numbers are employed to populate the screen with objects like cars, people, and trees. Slot machines in casinos utilize randomness to stop arbitrarily during game play. Within the field of cryptography [9], random numbers hold particular significance. They serve as the basis for generating encryption and decryption keys, ensuring secure transmission of messages between senders and receivers. The randomness of the key directly affects the vulnerability of the encrypted messages to attacks. Random numbers are also used in generating noise for simulating wireless communication systems.

There are several approaches to creating random numbers. One method involves using a hardware random number generator [14], which relies on physical phenomena like thermal noise or the photoelectric effect to produce truly random sequences. Another approach entails using digital circuits to a create a pseudorandom number generator (PRNG) that emulates statistical randomness. PRNGs have a finite period, meaning that they repeat after a certain interval, but within this period, they exhibit characteristics of statistical randomness, making their output appear similar to truly random numbers.

Binary sequences which consist of a series of 0s and 1s are frequently used in applications like cryptography and simulation. Typically, the more unpredictable the sequence of 0s and 1s, the better the results, but this depends on the application.

This is especially important in tasks where randomness is crucial for security or accuracy. Generating truly random sequences can be impractical and inefficient and often require a reliance on physical phenomena. Consequently, digital circuits are commonly employed to generate pseudorandom sequences [7], which offer a more practical alternative. Pseudorandom sequences are commonly generated using a linear feedback shift register (LFSR). An LFSR consists of flip-flops connected in series along with XOR gates (mod-2 adders) in the feedback path. By combining the outputs of the XOR gates based on primitive polynomials, the LFSR can generate sequences with maximum period, known as maximum length sequences or m-sequences. They possess good statistical randomness properties such as balance, run, and autocorrelation, which are defined below. Note that this thesis only considers binary sequences.

Balance: In a balanced integer sequence, each integer appears with equal frequency, meaning that there is an equal number of each possible value. For instance, in a random binary sequence, there should be an equal number of 0s and 1s [2].

Run: A run refers to a sequence of consecutive identical numbers within a sequence. In an ideal binary sequence, the distribution of runs follows a specific pattern. Half of the runs should have length 1, a quarter of the runs should have length 2, an eighth should have length 3, and so on.

Autocorrelation: The autocorrelation is a measure of how similar a sequence is to a delayed copy of itself [10]. It is given by

$$r(k) = \sum_{m=0}^{N-1} \left[1 - 2\left(s[m] \oplus s[m-k]\right)\right] \tag{1.1}$$

where $s[k]$ is the kth bit of the sequence, and N is the length of the sequence. The ratio of the magnitude of the second largest value to the magnitude of the largest value in the autocorrelation is called the maximum sidelobe ratio (MSR). The lower the MSR, the better the autocorrelation of the sequence.

Linear Complexity: The linear complexity of a sequence refers to the minimum length of an LFSR required to generate that sequence [10]. The goal is often to have a high linear complexity, but m-sequences have low linear complexity.

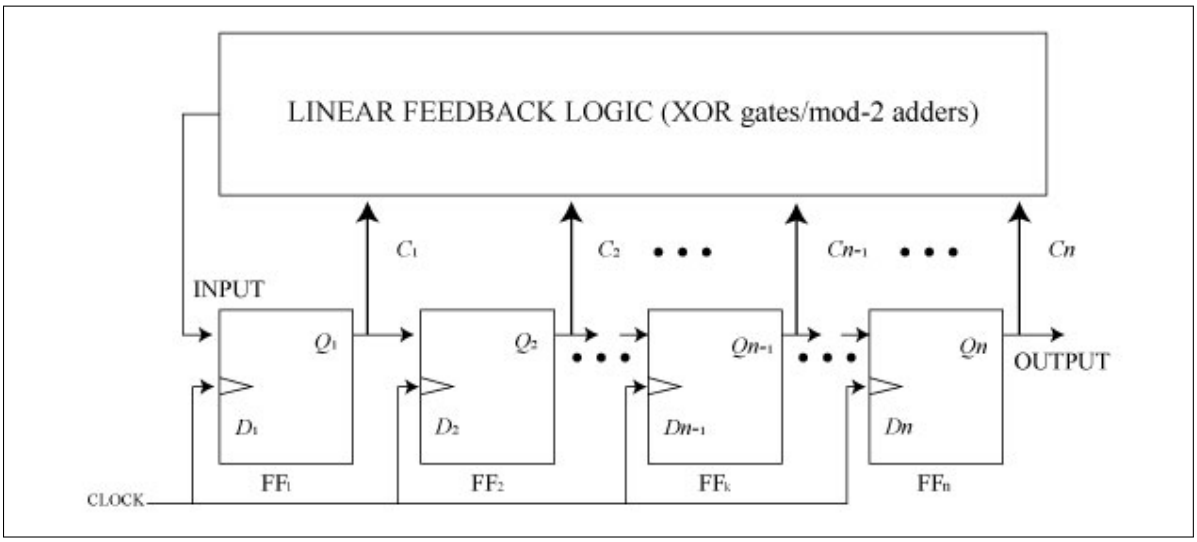

Figure 1.1: The structure of an n-bit linear feedback shift register (LFSR).

1.1 Linear Feedback Shift Registers and Maximum Length Sequences

An n-bit LFSR consists of n flip-flops, synchronized with a clock, and connected in a specific structure. In this structure, each flip-flop is connected to taps and a feedback circuit made up of mod-2 adders. The state of an LFSR is the sequence of bits stored in the flip-flops (D_1 to D_n), and the output of the LFSR is the content of the last flip-flop (D_n). The next state of the LFSR is determined by the feedback circuit. Figure 1.1 shows the structure of an n-bit LFSR consisting of n D flip-flops (D_1 to D_n), n taps (C_1 to C_n) and a feedback circuit comprised of mod-2 adders.

An m-sequence exhibits favorable statistical randomness characteristics such as balance, run, and autocorrelation, but has low linear complexity. These characteristics serve as a benchmark for assessing the quality of pseudorandom sequences. In this thesis, they will be utilized to evaluate the generated sequences. The key properties of an m-sequence are the following [13].

1. **Balance:** In an m-sequence, the number of ones is equal to the number of zeros plus one. It has $2^{n-1}\,(0.5(N+1))$ ones and $2^{n-1}-1\,(0.5(N-1))$ zeros within a period (N) [2]. Thus, the balance property of an m-sequence is optimal.

2. **Run:** For an m-sequence

- there is 1 run of ones of length n,
- there is 1 run of zeros of length $n-1$,
- there are 1 run of ones and 1 run of zeros of length $n-2$,
- there are 2 runs of ones and 2 runs of zeros of length $n-3$,
- there are 4 runs of ones and 4 runs of zeros of length $n-4$,

$$\vdots$$

- there are 2^{n-3} runs of ones and 2^{n-3} runs of zeros of length 1 [2].

3. **Autocorrelation:** The autocorrelation of an m-sequence is given by

$$r(k) = \begin{cases} N, & k = aN \\ -1, & k \neq aN \end{cases} \tag{1.2}$$

where a is an integer.

4. **Linear Complexity:** The linear complexity of an m-sequence is

$$LC = \lceil \log_2 N \rceil = n \tag{1.3}$$

1.2 Cellular Automata

Cellular automata (CA) were introduced in 1940 and their scope and applications in computer systems have since been investigated extensively [15] [16]. CAs are structured as an array of binary cells in one or more dimensions. The next state (0 or 1) of a cell is determined by a rule that takes the current states of the cells in its neighborhood as inputs. All cells are synchronized with an external clock.

The neighbourhood of a cell is the cell itself and adjacent cells whose current states are used to determine the next state of the cell. The neighborhood depends on the number of dimensions of the cellular automaton, it can be one-dimensional, two-dimensional, or higher-dimensional, depending on the nature of the problem. Each cell has a finite number of states such as on/off, black/white, or any other relevant representation. The maximum size of a neighbourhood in a 1D CA is 3. It comprises the cell itself and the cells to its right and left. The neighbourhood of corner cells in

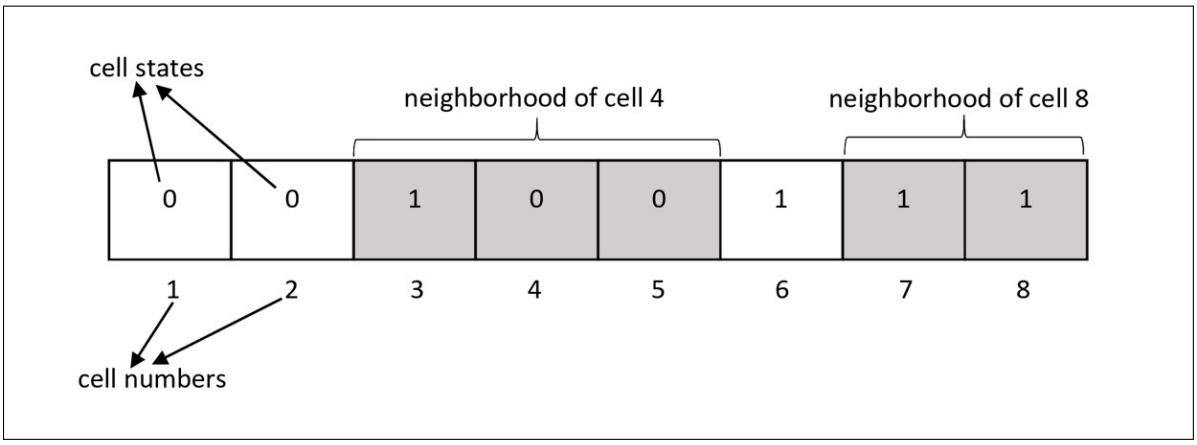

Figure 1.2: An example of a 1D cellular automaton.

a 1D CA has size 2 (the cell itself and the cell adjacent to it).

Figure 1.2 shows a 1D CA of size $n = 8$ and the shaded cells show the neighbourhood of cells 4 and 8, where the cells are numbered from left to right. The state table gives the next state of the cell for each of the possible neighbourhood current states. For a size 3 neighbourhood, there are $2^3 = 8$ states in the state table and $2^8 = 256$ state tables based on different possible next states [2]. The next states generated by the respective state tables are known as rules. These rules are numbered 0 to 255 and are called Wolfram rules [15]. Rules 0, 60, 102, 170, 204 and 240 produce poor results with respect to pseudorandom sequence generation [2]. Rule 0 makes the state of a cell zero, while 204 retains the current state of the cell. Further, rules 60, 102, 170 and 240 divide the CA into two parts.

This thesis considers 1D CA so the maximum size of a neighborhood is 3. Thus, the next state of a cell can be considered a boolean function of 3 inputs $s_i(k+1) = F(s_{i-1}(k), s_i(k), s_{i+1}(k))$ where

- $s_i(k+1)$ is the next state of a cell,
- $s_i(k)$ is the current state of a cell,
- $s_{i-1}(k)$ is the current state of the left cell in the neighborhood,
- $s_{i+1}(k)$ is the current state of the right cell in the neighborhood, and
- $F()$ is the rule.

The state table gives the next state of the cell for each of the possible current neighbourhood states. If the state table is generated by a linear function (mod-2 additions) of the current state of the neighbourhood the rule is called linear [2]. From the 256 rules, there are only $2^3 = 8$ (0, 60, 90, 102, 150, 170, 204 and 240) linear rules. Combinations of rules 90 and 150 based on primitive polynomials can be used to generate m-sequences [4] [17]. Rules 90 and 150 can be implemented using XOR gates and mod-2 adders. Rule 90 takes only adjacent cells as inputs (2 inputs), while rule 150 takes adjacent cells and the cell itself as inputs (3 inputs), to generate the next state of the cell [2]. The state table for rule 90 is given in Table 1.1 and the state table for rule 150 is given in Table 1.2.

Current State			Next State
$s_{i-1}(k)$	$s_i(k)$	$s_{i+1}(k)$	$s_i(k+1)$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

Table 1.1: Rule 90 state table.

Current State			Next State
$s_{i-1}(k)$	$s_i(k)$	$s_{i+1}(k)$	$s_i(k+1)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Table 1.2: Rule 150 state table.

The advantage of using CAs is that there are no long feedback paths which can cause delays particularly with large LFSRs. This is because all computations happen

in neighbourhoods, so the feedback paths are minimized. This advantage is exploited in this thesis to generate pseudorandom sequences with large linear complexity using shrinking generators which are discussed below.

1.3 Shrinking Generators

In [5] a new type of pseudorandom sequence generator was presented using two LFSRs that are clocked together. The output bits are produced by shrinking the output sequence of the first LFSR under the control of the second LFSR. The output bit of the second LFSR is selected if the output bit of the first LFSR is 1, otherwise it is discarded. Existing shrinking generators include the self-shrinking generator, modified self-shrinking generator, and new self-shrinking generator. This thesis considers these shrinking generators in constructing sequences using CA.

Self-Shrinking Generator: The Self-Shrinking Generator (SSG) was designed in [8]. It uses only one LFSR to generate a pseudorandom binary sequence. The SSG shrinks two bits of an LFSR sequence to one output bit [6]. For the binary sequence generated by an LFSR $(x) = x_0x_1x_2\ldots$, let $(y) = y_0y_1\ldots$ be the corresponding self-shrunken sequence. Consider two consecutive bits of (x), x_{2i} and x_{2i+1}. If x_{2i} is 1, then x_{2i+1} is output and if x_{2i} is 0, the pair is discarded [3]. The truth table for this generator is given in Table 1.3.

x_{2i}	x_{2i+1}	y_i
0	0	No Output
0	1	No Output
1	0	0
1	1	1

Table 1.3: The self-shrinking generator truth table.

Modified Self-Shrinking Generator: The Modified Self-Shrinking Generator (MSSG) is an improvement of the self-shrinking generator [5]. For the binary sequence generated by an LFSR $(x) = x_0x_1x_2\ldots$, let $(y) = y_0y_1\ldots$ be the corresponding self-shrunken sequence. Consider three consecutive bits of $(x), x_{3i}, x_{3i+1}$ and x_{3i+2}. If $x_{3i} \oplus x_{3i+1}$ is 1, then x_{3i+2} is output and if $x_{3i} \oplus x_{3i+1}$ is 0, the bits are discarded [3]. The truth table for this generator is given in Table 1.4.

x_{3i}	x_{3i+1}	x_{3i+2}	y_i
0	0	0	No Output
0	0	1	No Output
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	No Output
1	1	1	No Output

Table 1.4: The modified self-shrinking generator truth table.

New Self-Shrinking Generator: The New Self-Shrinking Generator (NSSG) utilizes more complex operations than the previous generators to increase the linear complexity [3]. Instead of generating a single bit, two bits are produced as output for every three bits in the original sequence. For the binary sequence generated by an LFSR $(x) = x_0x_1x_2\ldots$, let $(y) = y_0y_1\ldots$ be the corresponding self-shrunken sequence. Consider three consecutive bits of (x), x_{3i}, x_{3i+1} and x_{3i+2}. For each such triple, 2 bits are output as $y_{2i} = x_{3i}, y_{2i+1} = x_{3i+1} \oplus x_{3i+2}$. The truth table for this generator is given in Table 1.5.

x_{3i}	x_{3i+1}	x_{3i+2}	y_{2i}	y_{2i+1}
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	0

Table 1.5: The new self-shrinking generator truth table.

1.4 Thesis Organization

Chapter 2 describes the 1D CA and shrinking generator evaluation system. It outlines the parameters associated with the system and the functions of its modules. It also provides the filtering criteria based on linear complexity, bal-

ance, run, and autocorrelation for selection of rule combinations that produce the best sequences.

Chapter 3 presents the results obtained from the CA evaluation system and provides an analysis of the generated sequences. Filtered results are given for each CA size based on the criteria specified in Chapter 2. These results are used to determine the parameters that produce pseudorandom sequences with high linear complexity and good randomness. Results are obtained with and without using random rules. The properties of the sequences obtained are compared with those in [2].

Chapter 4 provides the conclusions and suggestions for future work associated with pseudorandom sequence generation using shrinking generators and 1D CAs.

Chapter 2

Cellular Automata and Sequence Generation

In this chapter, the 1D CA and shrinking generator evaluation system and the associated parameters are discussed. The use of non-linear rules in CAs is considered to generate pseudorandom sequences that have high linear complexity and good randomness. In the analysis, m-sequences and their properties serve as a benchmark.

A validation mechanism was created using a 1D CA of length n [13]. Each cell was initially assigned a linear rule (90 or 150) to yield an m-sequence. Subsequently, one of these cells was substituted with a random rule (non-linear), and the linear complexity of the resulting sequence was computed. In this thesis, the framework in [13] is expanded to encompass all linear rule combinations with one or more non-linear rule substitutions. The quality of the generated sequences is assessed based on multiple randomness criteria. Furthermore, the sequences obtained are modified using shrinking generators to determine the effectiveness of this approach.

The following parameters are used to characterize the sequences generated [2].

1. **Size (n):** The size is the number of cells in the 1D CA. Figure 2.1 shows a 1D CA of size $n = 5$.

2. **Linear Rules (LR):** Linear rules is the set of 90 and 150 rules initially assigned to the 1D CA. For convenience, rule 90 is denoted by 0 and rule 150 by 1. This forms an n bit vector that represents the CA. For example, the 5 bit 1D CA represented by LR = 01100 is shown in Figure 2.1.

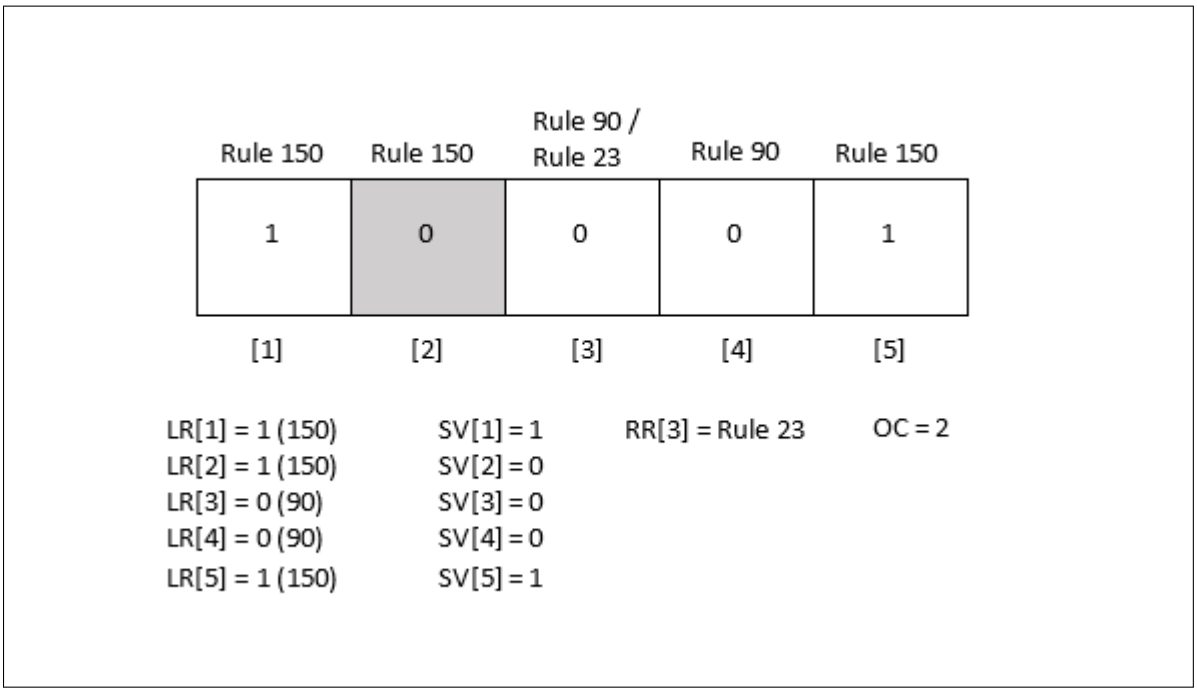

Figure 2.1: An example of a 1D CA of size $n = 5$ in the evaluation system.

3. **Start Value (SV):** Start value is the initial state of the 1D CA. SV is a vector of n bits. In Figure 2.1, the initial state of the CA is $SV = 10001$.

4. **Random Rule (RR) and Randomized Cell (RC):** Random rule is one of the 252 non-linear rules used to replace a linear rule in the 1D CA. The non-linear rule is associated with a cell position, which is the randomized cell. In Figure 2.1, non-linear rule $RR = 23$ replaces rule 150 at cell position $RC = 3$.

5. **Observed Cell (OC):** Observed cell is the cell position that is monitored to obtain a sequence of bits. In Figure 2.1, $OC = 2$ (cell position 2).

6. **Linear Complexity (LC):** The linear complexity of the sequence generated by the OC. The Berlekamp-Massey algorithm is used to compute the linear complexity [11].

7. **Balance (B):** The balance of the sequence generated by the OC.

8. **Run (R):** The run of the sequence generated by the OC. It is an array of size 6 containing runs of length one, two, three, ..., six.

9. **Autocorrelation (AC) and Max Sidelobe Ratio (MSR):** Autocorrelation

is an array containing values of the autocorrelation of the sequence from the OC.

10. **Sequence (S):** The sequence from the OC. The length of the bit stream is restricted to $2^n - 1$, which is equal to the period of an m-sequence generated by an LFSR of the same length as the CA (n).

    In the evaluation system, parameters 1 to 5 are varied and parameters 6 to 10 are analyzed.

2.1 The 1D CA and Shrinking Generator Evaluation System

The 1D CA evaluation system was developed using Python scripting. The operating system used was Linux Ubuntu 22.04. It is first used to analyze sequences generated using linear rules for 1D CA. In Chapter 3 it is used to analyze 1D CA containing a non-linear rule for pseudorandom sequence generation. The evaluation system is comprised of the following modules as shown in Figure 2.2.

Main Control Module: The control module has inputs n, RC, LR, SV, OC and RC. With one set of these parameters, an iteration of the CA module produces a sequence [2]. The number of iterations is determined as follows.

- Possible values of LR is 2^n.
- Possible values of SV is 2^n.
- Possible values of OC is n.
- Possible non-linear rules (RR) is 248.
- Possible values of RC is n. To minimize the number of calculations, the edge cells are excluded from consideration. The impact of non-linear rules on edge cells is diminished due to the presence of a null boundary cell in the neighborhood. Hence, the possible values of RC is reduced to $n - 2$.

For one random cell replacement, the number of iterations are

$$2^n \times 2^n \times n \times 248 \times (n - 2). \tag{2.1}$$

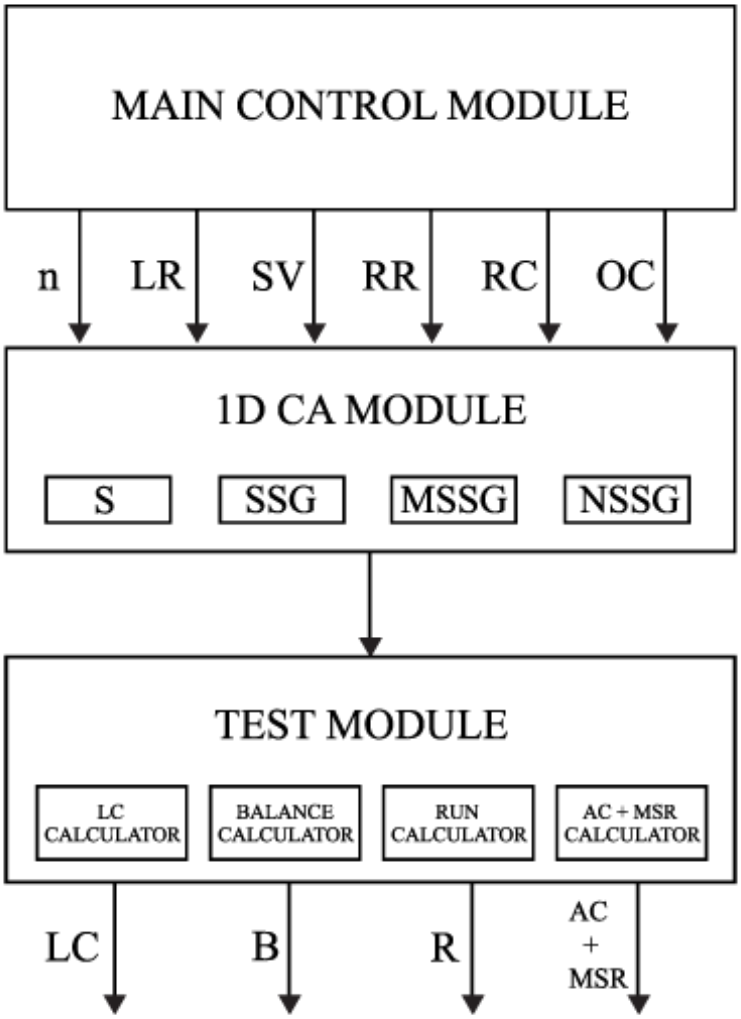

Figure 2.2: The Modules of the 1D CA evaluation system.

CA Module: The CA module uses the input n which is the size of the CA. RR replaces one linear rule with a non-linear rule at cell position RC. The output sequence S is obtained from cell OC. The SV bit vector is the initial state of the CA. The state of the CA is updated based on the combination of linear rules with one replaced by a non-linear rule.The length of the output sequence generated is twice the period of an m-sequence ($2\times(2^n-1)$). The first half of the output sequence is discarded and the second half is used as the output sequence S. This is done to remove any unwanted effects of the initial conditions. The sequence S of length $2^n - 1$ is passed to the three shrinking generators SSG, MSSG, and NSSG.

Test Module: The test module takes the sequence S from the CA module along with SSG, MSSG, and NSSG as input and calls the following functions.

- **LC Calculator:** The linear complexity is calculated and returned by this function. The Berlekamp-Massey algorithm is employed to determine the LC [13].
- **Balance Calculator:** This function calculates the number of 0s and 1s and subtracts the number of 0s from the number of 1s to produce B.
- **Run Calculator:** This function calculates the runs R of S, SSG, MSSG, and NSSG. The runs are calculated for both 0s and 1s.
- **AC and MSR Calculator:** This function calculates the AC and MSR of S, SSG, MSSG, and NSSG.

2.2 Filtering Criteria

The properties of the generated sequences are evaluated, and the set of best linear rules (LR) in combination with non-linear rules (RR) and their corresponding cell positions (RC) are obtained. This selection is made based on criteria designed with reference to the properties of m-sequences and the randomness tests from [12]. These criteria are given below.

1. Linear complexity is used as the initial filtering criteria. Only those LR, RR, and RC which have $LC \geq n$ for small n (3 and 4) and $LC \geq 2^n/4$ for $n > 4$, are considered. This is because large linear complexity is the goal of this thesis.

2. Rules are chosen that maintain a relatively unchanged LC for all 2^n start values (SV). The objective is to obtain good sequences irrespective of the initial state of the CA, which is similar to m-sequences. The two tests used for balance and run are given below.

   (a) Frequency Test for Balance: In this test, each bit of a sequence is assigned a value -1 or +1 (0 = -1 and 1 = +1), and the sum of the values is

   $$X_N = \sum_{m=0}^{N-1} 2x(m) - 1$$

   where $x(m)$ is the mth bit of S and $N = 2^n - 1$ is the length of S. This is used to calculate the error-function complement

   $$P_B = \text{erfc}\left(\frac{|X_N|}{\sqrt{2N}}\right)$$

   The balance threshold $P_{BTh} = 0.01$ is used to filter out rules for which $P_B > P_{BTh}$ [12].

   (b) Run Test: In this test, the ratio of number of 1s to the length N of the sequence is calculated as

   $$\pi = \sum_{m=0}^{N-1} x(m)/N$$

   where $x(m)$ is the mth bit of S. If the condition

   $$|\pi - 1/2| < \frac{X_N}{\sqrt{N}}$$

   is not satisfied, then the test fails. Then the test statistic

   $$V_N(\text{obs}) = \sum_{m=0}^{N-2} v(m) + 1$$

   is calculated, where $v(m) = 0$ if the $(m+1)$th bit is the same as the mth bit and $v(m) = 1$, otherwise. The corresponding complementary error

function

$$P_R = \text{erfc}\left(\frac{|V_N(\text{obs}) - 2N\pi(1-\pi)|}{2\sqrt{2N\pi(1-\pi)}}\right)$$

is calculated. A run threshold value of $P_{Rth} = 0.01$ is used to filter the parameters so that the combination is kept if $P_R > P_{Rth}$ [12].

(c) Autocorrelation Test: The MSR threshold is $MSR_{Th} = 0.3$. Rules that generate sequences with an MSR greater than MSR_{Th} are filtered out.

Filtered sets of sequences with and without the insertion of non-linear rules for different CA sizes are obtained using the above criteria, and the properties of the sequences produced are compared and analyzed. A comparison is also made between the best sequences from [2] and those obtained in this thesis.

Chapter 3

Results and Analysis

In this chapter, pseudorandom sequences are presented and analyzed for $n = 3$ to $n = 18$. Results are first given without an RR and then sequences are generated using an RR for all CA start values. The results are extracted in Excel files. Based on the filtering criteria discussed in Section 2.2, the following observations are made.

3.1 Initial Observations for $n = 3$ to 10

An even RR applied to a CA with an initial state of $SV = 0$ produces an all-zero sequence. Complementary rules produce identical sequences when the CA is reversed [2]. The final stage of filtering is based on the randomness criteria for balance, run and autocorrelation with thresholds $P_B > 0.01$, $P_R > 0.01$, and $MSR < 0.3$, respectively. Only sequences meeting these criteria are considered suitable, as they have high linear complexity and good randomness.

3.2 Filtered Results for $n = 3$

Results for 1D CA of size $n = 3$ were obtained for $OC = 2$ with and without an RR. Table 3.1 gives the best 1D CA of size $n = 3$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria. Note that these are all m-sequences and have $MSR = 0.14$ and $LC = 3$. Without using an RR, the SSG, NSSG, and MSSG sequences failed the criteria.

Table 3.2 gives the best sequence in [2] and NSSG sequence for a 1D CA of size

$n = 3$ after single cell replacement with $RR = 15$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while these NSSG sequences passed. The LC of this NSSG sequence is 4 which is higher than that of the sequence in [2] which is 3. The balance for the NSSG sequence is -1, whereas the balance for the sequence in [2] is 1, and the MSR is 0.43 for both sequences with $OC = 2$. There are four RR that give better sequences compared to the sequences with $RR = 15$ and they are given in Table 3.2. The NSSG sequences for $RR = 30, 22, 214$, and 122 have $LC = 5$ which is higher than the sequence generated using $RR = 15$. The MSR for these four sequences is 0.43 and the balance is 1.

Table 3.3 gives the LC for different OC. Different RR are used to show the variation in LC when the OC is varied. The parameters LR, RR, RC and SV are kept constant. It is observed that the LC of the sequences is 3 to 6 when the OC is varied.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
LIN	2	2	[90, 90, 150]	[0, 0, 1]	3	1	[2, 1, 1, 0, 0, 0, 0]	0.14	[7, -1, -1, -1, -1, -1, -1]	[1, 1, 1, 0, 1, 0, 0]
LIN	2	2	[90, 90, 150]	[0, 1, 0]	3	1	[2, 1, 1, 0, 0, 0, 0]	0.14	[7, -1, -1, -1, -1, -1, -1]	[0, 0, 1, 1, 1, 0, 1]
LIN	2	2	[150, 90, 90]	[0, 1, 1]	3	1	[2, 1, 1, 0, 0, 0, 0]	0.14	[7, -1, -1, -1, -1, -1, -1]	[0, 1, 1, 1, 0, 0, 1]
LIN	2	2	[150, 150, 90]	[1, 0, 0]	3	1	[2, 1, 1, 0, 0, 0, 0]	0.14	[7, -1, -1, -1, -1, -1, -1]	[0, 1, 1, 0, 1, 0, 0]
LIN	2	2	[150, 150, 90]	[1, 0, 1]	3	1	[2, 1, 1, 0, 0, 0, 0]	0.14	[7, -1, -1, -1, -1, -1, -1]	[0, 1, 1, 1, 0, 0, 0]
LIN	2	2	[90, 90, 150]	[1, 1, 1]	3	1	[2, 1, 1, 0, 0, 0, 0]	0.14	[7, -1, -1, -1, -1, -1, -1]	[0, 1, 0, 0, 1, 1, 1]

Table 3.1: Sequences for $n = 3$ and $OC = 2$ without an RR.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	15	[90, 15, 150]	[0, 1, 0]	3	1	[1, 3, 0, 0, 0, 0, 0]	0.43	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 0, 1, 1, 0]
NSSG	2	2	15	[90, 15, 150]	[0, 1, 0]	4	-1	[0, 2, 1, 0, 0, 0, 0]	0.43	[7, 3, -1, -5, -5, -1, 3]	[0, 0, 1, 1, 1, 0, 0]
NSSG	2	2	30	[150, 30, 90]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	0.43	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 1, 0, 0]
NSSG	2	2	22	[90, 22, 150]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	0.43	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 1, 0, 0]
NSSG	2	2	214	[90, 214, 150]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	0.43	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 1, 0, 0]
NSSG	2	2	122	[90, 122, 150]	[1, 1, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	0.43	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 1, 0, 0]

Table 3.2: Comparison of the best sequence in [2] with the NSSG sequences for $n = 3$, $OC = 2$, and $RR = 15, 30, 22, 214, 122$.

3.3 Filtered Results for $n = 4$

Table 3.4 gives the best 1D CA of size $n = 4$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria. Without using an RR, the SSG and MSSG sequences failed the criteria. Table 3.4 indicates that the NSSG sequences are the best and have $MSR = 0.20$ and $LC = 5$ and 8.

TYPE	OC	RC	RR	LR	SV	LC	B	R	AC	S
[2]	1	2	15	[90, 15, 150]	[0, 1, 0]	3	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[0, 1, 1, 0, 0, 1, 1]
[2]	2	2	15	[90, 15, 150]	[0, 1, 0]	3	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 0, 1, 1, 0]
[2]	3	2	15	[90, 15, 150]	[0, 1, 0]	4	-3	[4, 0, 1, 0, 0, 0, 0]	[7, -1, -1, 3, 3, -1, -1]	[0, 1, 0, 0, 0, 1, 0]
NSSG	1	2	30	[150, 30, 90]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[0, 1, 0, 1, 1, 0, 0]
NSSG	2	2	30	[150, 30, 90]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 1, 0, 0]
NSSG	3	2	30	[150, 30, 90]	[1, 0, 1]	5	-1	[1, 3, 0, 0, 0, 0, 0]	[7, 3, -5, 3, 1, -5, -1]	[1, 0, 0, 1, 1, 0, 0]
NSSG	1	2	22	[90, 22, 150]	[1, 0, 1]	6	1	[1, 0, 2, 0, 0, 0, 0]	[7, -1, -1, 3, 3, -5, -1]	[1, 0, 0, 0, 1, 0, 0]
NSSG	2	2	22	[90, 22, 150]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 0, 0, 1, 0, 0, 0]
NSSG	3	2	22	[90, 22, 150]	[1, 0, 1]	5	1	[3, 2, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 0, 0, 0]
NSSG	1	2	214	[90, 214, 150]	[1, 0, 1]	4	1	[1, 3, 0, 2, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 1, 1, 0, 0]
NSSG	2	2	214	[90, 214, 150]	[1, 0, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 0, 0, 1, 1, 0, 0]
NSSG	3	2	214	[90, 214, 150]	[1, 0, 1]	5	-3	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -1, 3, 3, -5, -1]	[1, 0, 0, 1, 0, 0, 0]
NSSG	1	2	122	[90, 122, 150]	[1, 1, 1]	5	-1	[1, 3, 0, 0, 0, 0, 0]	[7, 5, -1, -1, 3, -5, -1]	[1, 1, 0, 0, 1, 0, 0]
NSSG	2	2	122	[90, 122, 150]	[1, 1, 1]	5	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 1, 1, 1, 0, 1]
NSSG	3	2	122	[90, 122, 150]	[1, 1, 1]	6	1	[1, 3, 0, 0, 0, 0, 0]	[7, -1, -5, -3, 3, -5, -1]	[1, 1, 0, 1, 1, 1, 1]

Table 3.3: LC for different OC, $n = 3$, $RR = 15, 30, 22, 214, 122$, and $RC = 2$.

Table 3.5 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 4$ after single cell replacement with $RR = 169$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while the NSSG sequence passed. The LC of this sequence is 7 which is the same as the sequence in [2], and the balance is 1, whereas the balance for the sequence in [2] is -1. The MSR for the NSSG sequence is 0.20 lower than the sequence in [2] with $OC = 2$. There are three RR that give better sequences compared to the sequences with $RR = 169$ and they are given in Table 3.5. The NSSG sequences for $RR = 144, 173$, and 11 have $LC = 10$ and 11 which is higher than the sequence generated using $RR = 169$. The MSR for these three sequences is 0.20 and the balance is 1.

Table 3.6 gives the LC for different OC. Different RR are used to show the variation in LC when the OC is varied. The parameters LR, RR, RC and SV are kept constant. It is observed that the LC of the sequences is 6 to 11 when the OC is varied.

Table 3.7 gives the best sequence in [2] for $n = 3$ with the 1D CA MSSG sequences of size $n = 4$ that passed the filtering criteria. This comparison is made because shrinking reduces the sequence length. The SSG sequences did not pass the criteria. The LC for the sequence in [2] is 3 while the MSSG sequences have $LC = 8$ and 9. The MSR for the sequence in [2] is 0.47, and for the MSSG sequences it is 0.20 or 0.47.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[150, 150, 150, 90]	[0, 0, 0, 1]	8	1	[2, 3, 1, 1, 0, 0, ...]	0.20	[15, 3, -1, -5, -5, 3, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	2	2	[150, 90, 90, 90]	[0, 1, 0, 0]	5	1	[3, 2, 0, 2, 0, 0, ...]	0.20	[15, 3, -1, 3, -1, -1, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[90, 90, 90, 150]	[0, 1, 0, 1]	5	1	[3, 2, 0, 2, 0, 0, ...]	0.20	[15, 3, -1, -1, -1, 3, ...]	[1, 0, 0, 0, 0, 1, ...]
NSSG	2	2	[90, 150, 150, 150]	[0, 1, 1, 0]	8	1	[2, 3, 1, 1, 0, 0, ...]	0.20	[15, 3, -1, -5, -5, 3, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	2	2	[90, 150, 150, 150]	[0, 1, 1, 1]	8	1	[3, 1, 2, 1, 0, 0, ...]	0.20	[15, 3, -1, -5, -5, -1, ...]	[1, 1, 1, 0, 0, 0, ...]
NSSG	2	2	[90, 150, 90, 150]	[1, 0, 1, 0]	8	5	[2, 3, 1, 1, 0, 0, ...]	0.20	[15, 3, -1, -1, -1, 3, ...]	[1, 1, 1, 1, 0, 0, ...]

Table 3.4: Sequences for $n = 4$ and $OC = 2$ without an RR.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	7	-1	[3, 0, 4, 0, 0, 0, ...]	0.47	[15, 3, -1, -9, -5, -5, ...]	[0, 1, 1, 1, 0, 0, ...]
NSSG	2	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	7	1	[4, 2, 1, 1, 0, 0, ...]	0.20	[15, -1, -1, 3, -1, 3, ...]	[0, 0, 0, 1, 0, 0, ...]
NSSG	2	2	144	[150, 144, 150, 90]	[0, 1, 1, 0]	11	1	[8, 2, 1, 0, 0, 0, ...]	0.20	[15, -5, -1, -1, -1, -1, ...]	[1, 0, 0, 1, 0, 1, ...]
NSSG	2	2	173	[90, 173, 90, 150]	[1, 1, 0, 1]	10	1	[6, 3, 1, 0, 0, 0, ...]	0.20	[15, -5, -1, -1, -1, -1, ...]	[1, 0, 1, 0, 1, 1, ...]
NSSG	2	2	11	[150, 11, 90, 90]	[1, 1, 1, 0]	11	1	[8, 2, 1, 0, 0, 0, ...]	0.20	[15, -5, -1, -1, -1, -1, ...]	[1, 0, 0, 1, 0, 1, ...]

Table 3.5: Comparison of the best sequence in [2] with the NSSG sequences for $n = 4$, $OC = 2$, and $RR = 169, 144, 173, 11$.

3.4 Filtered Results for $n = 5$

Table 3.8 gives the best 1D CA of size $n = 5$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria. Without using an RR, the SSG and MSSG sequences failed the criteria. Table 3.8 indicates that the NSSG sequences are the best and have $MSR = 0.11$ and 0.23 and $LC = 12$.

Table 3.9 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 5$ after single cell replacement with $RR = 146$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while the NSSG sequence passed. The LC of this sequence is 16 which is the same as the sequence in [2], and the balance is 13, whereas the balance for the sequence in [2] is -1. The MSR for the NSSG sequence is 0.35 lower than the sequence in [2] with $OC = 2$. There are two RR that give better sequences compared to the sequences with $RR = 146$ and they are given in Table 3.9. The NSSG sequences for $RR = 155$, and 185 have $LC = 22$ which is higher than the sequence generated using $RR = 146$. The MSR for these two sequences is 0.23 and the balance is 5.

Table 3.10 gives the LC for different OC. Different RR are used to show the variation in LC when the OC is varied. The parameters LR, RR, RC and SV are kept constant. It is observed that the LC of the sequences is 16 and 21 to 23 when the OC is varied.

Table 3.11 gives the best sequence in [2] for $n = 4$ with the 1D CA SSG and MSSG sequences of size $n = 5$ that passed the filtering criteria. This comparison is

TYPE	OC	RC	RR	LR	SV	LC	B	R	AC	S
[2]	1	2	169	[90, 169, 150, 150]	[1, 0, 0, 0]	7	-1	[4, 1, 3, 0, 0, 0, ...]	[15, -1, -1, -5, -1, -5, ...]	[0, 0, 0, 1, 0, 1, ...]
[2]	2	2	169	[90, 169, 150, 150]	[1, 0, 0, 0]	7	1	[4, 1, 3, 0, 0, 0, ...]	[15, -1, -1, -5, -1, -5, ...]	[0, 0, 1, 0, 1, 1, ...]
[2]	3	2	169	[90, 169, 150, 150]	[1, 0, 0, 0]	6	1	[3, 3, 2, 0, 0, 0, ...]	[15, -1, -5, -1, -1, -1, ...]	[1, 0, 0, 1, 1, 1, ...]
[2]	4	2	169	[90, 169, 150, 150]	[1, 0, 0, 0]	7	-1	[6, 0, 3, 0, 0, 0, ...]	[15, -1, -1, -5, -1, -5, ...]	[1, 0, 0, 0, 1, 0, ...]
NSSG	1	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	7	-1	[4, 2, 1, 1, 0, 0, ...]	[15, -9, 1, 3, 3, 3, ...]	[1, 0, 0, 0, 1, 0, ...]
NSSG	2	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	7	1	[4, 1, 3, 1, 0, 0, ...]	[15, -1, -1, 3, -1, 3, ...]	[0, 0, 0, 1, 0, 0, ...]
NSSG	3	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	7	1	[4, 2, 1, 0, 0, 0, ...]	[15, 3, 9, -3, 1, 3, ...]	[1, 0, 1, 1, 0, 0, ...]
NSSG	4	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	8	1	[6, 2, 0, 1, 0, 0, ...]	[15, -1, -1, 3, -1, 3, ...]	[0, 1, 0, 1, 0, 0, ...]
NSSG	1	2	144	[150, 144, 150, 90]	[0, 1, 1, 0]	11	5	[8, 1, 1, 0, 0, 0, ...]	[15, -7, -1, -5, -1, -1, ...]	[1, 0, 1, 1, 0, 1, ...]
NSSG	2	2	144	[150, 144, 150, 90]	[0, 1, 1, 0]	11	1	[8, 2, 1, 0, 0, 0, ...]	[15, -5, -1, -1, -1, -1, ...]	[1, 0, 0, 1, 0, 1, ...]
NSSG	3	2	144	[150, 144, 150, 90]	[0, 1, 1, 0]	11	1	[8, 2, 2, 1, 0, 0, ...]	[15, -5, -1, -1, -1, -1, ...]	[1, 0, 0, 0, 0, 1, ...]
NSSG	4	2	144	[150, 144, 150, 90]	[0, 1, 1, 0]	11	-3	[8, 2, 1, 1, 2, 0, ...]	[15, 5, 3, -9, -1, -1, ...]	[1, 1, 0, 1, 1, 1, ...]
NSSG	1	2	173	[90, 173, 90, 150]	[1, 1, 0, 1]	10	1	[8, 2, 1, 0, 0, 0, ...]	[15, 3, -1, -5, -1, -1, ...]	[1, 1, 1, 0, 1, 1, ...]
NSSG	2	2	173	[90, 173, 90, 150]	[1, 1, 0, 1]	10	1	[6, 3, 1, 0, 0, 0, ...]	[15, -5, -1, -1, -1, -1, ...]	[1, 0, 1, 0, 1, 1, ...]
NSSG	3	2	173	[90, 173, 90, 150]	[1, 1, 0, 1]	11	-1	[6, 1, 1, 0, 1, 0, ...]	[15, -9, -1, -1, -1, -1, ...]	[1, 0, 0, 0, 1, 1, ...]
NSSG	4	2	173	[90, 173, 90, 150]	[1, 1, 0, 1]	10	1	[4, 1, 3, 0, 0, 0, ...]	[15, 3, 5, -1, -1, 3, ...]	[1, 1, 1, 0, 0, 1, ...]
NSSG	1	2	11	[150, 11, 90, 90]	[1, 1, 1, 0]	11	1	[8, 2, 2, 0, 1, 0, ...]	[15, 9, 3, -1, -1, -1, ...]	[1, 0, 1, 0, 0, 0, ...]
NSSG	2	2	11	[150, 11, 90, 90]	[1, 1, 1, 0]	10	1	[8, 2, 1, 0, 0, 0, ...]	[15, -1, -1, -5, -1, -1, ...]	[1, 0, 0, 1, 0, 1, ...]
NSSG	3	2	11	[150, 11, 90, 90]	[1, 1, 1, 0]	11	1	[9, 1, 1, 1, 0, 0, ...]	[15, -1, -1, -1, -9, -5, ...]	[1, 1, 0, 1, 1, 1, ...]
NSSG	4	2	11	[150, 11, 90, 90]	[1, 1, 1, 0]	11	1	[5, 2, 1, 2, 0, 0, ...]	[15, 3, -1, -1, -1, -1, ...]	[1, 1, 0, 0, 0, 1, ...]

Table 3.6: LC for different OC, $n = 4$, $RR = 169, 144, 173, 11$, and $RC = 2$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	15	[90, 15, 150]	[0, 1, 0]	3	1	[1, 3, 0, 0, 0, 0, 0]	0.43	[7, -1, -5, 3, 3, -5, -1]	[1, 1, 0, 0, 1, 1, 0]
MSSG	2	2	114	[150, 114, 150, 90]	[0, 0, 1, 0]	9	5	[3, 0, 4, 0, 0, 0, ...]	0.20	[15, 3, 3, -5, -1, 3, ...]	[1, 1, 1, 0, 1, 0, ...]
MSSG	2	2	182	[150, 182, 90, 150]	[1, 0, 0, 0]	8	5	[1, 4, 2, 0, 0, 0, ...]	0.47	[15, 3, -5, -1, 3, -1, ...]	[1, 1, 0, 0, 1, 1, ...]

Table 3.7: Comparison of the best sequence in [2] for $n = 3$ with the MSSG sequences for $n = 4$.

made because shrinking reduces the sequence length. The LC for the sequence in [2] is 7 while the SSG and MSSG sequences have $LC = 5$. The MSR for the sequence in [2] is 0.47, and for the SSG and MSSG sequences it is 0.33.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[90, 90, 90, 90, 150]	[0, 0, 0, 0, 1]	12	-5	[5, 4, 2, 1, 0, 0, ...]	0.23	[31, 7, 3, 3, -5, -1, ...]	[1, 1, 1, 0, 0, 0, ...]
NSSG	2	2	[90, 90, 90, 90, 150]	[0, 0, 0, 1, 0]	12	3	[7, 5, 2, 2, 0, 0, ...]	0.11	[31, -1, -5, -1, 3, -1, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	2	2	[90, 90, 90, 90, 150]	[0, 0, 0, 1, 1]	12	-7	[7, 3, 2, 1, 0, 0, ...]	0.23	[31, 3, 3, 3, -1, -5, ...]	[0, 0, 1, 0, 1, 0, ...]
NSSG	2	2	[90, 90, 90, 90, 150]	[0, 0, 1, 0, 0]	12	-1	[8, 5, 3, 1, 0, 0, ...]	0.23	[31, -1, -5, -5, -1, -5, ...]	[0, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[90, 90, 90, 90, 150]	[0, 0, 1, 0, 1]	12	5	[9, 4, 2, 1, 0, 0, ...]	0.23	[31, -1, -1, -1, 3, -1, ...]	[0, 0, 1, 1, 1, 1, ...]
NSSG	2	2	[90, 90, 90, 150, 90]	[0, 0, 1, 1, 0]	12	3	[7, 3, 2, 1, 0, 0, ...]	0.23	[31, 3, -1, -3, -1, 3, ...]	[0, 1, 0, 0, 1, 1, ...]

Table 3.8: Sequences for $n = 5$ and $OC = 2$ without an RR.

3.5 Filtered Results for $n = 6$

Table 3.12 gives the best 1D CA of size $n = 6$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria. Without using an RR, the SSG and MSSG sequences failed the criteria. Table 3.12 indicates that the NSSG sequences are the best and

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	-1	[14, 1, 5, 0, 0, 0, ...]	0.43	[31, -5, 3, -9, 15, -5, ...]	[0, 1, 0, 0, 0, 1, ...]
NSSG	2	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	13	[7, 2, 4, 2, 0, 0, ...]	0.35	[31, 3, 3, -1, 11, 3, ...]	[1, 1, 1, 1, 1, 0, ...]
NSSG	2	2	155	[150, 155, 90, 150, 150]	[0, 1, 0, 0, 1]	22	5	[9, 4, 2, 2, 0, 0, ...]	0.23	[31, -1, -1, -5, -1, 7, ...]	[1, 0, 1, 0, 1, 1, ...]
NSSG	2	2	185	[150, 185, 90, 150, 150]	[1, 1, 0, 0, 1]	22	5	[9, 4, 2, 2, 0, 0, ...]	0.23	[31, -1, -1, -5, -1, 7, ...]	[1, 0, 1, 0, 1, 1, ...]

Table 3.9: Comparison of the best sequence in [2] with the NSSG sequences for $n = 5$, $OC = 2$, and $RR = 146, 155, 185$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	AC	S
[2]	1	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	-3	[12, 2, 5, 0, 0, 0, ...]	[31, -5, 3, -9, 15, -5, ...]	[0, 0, 1, 0, 0, 0, ...]
[2]	2	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	-1	[14, 1, 5, 0, 0, 0, ...]	[31, -9, 7, -9, 15, -9, ...]	[0, 1, 0, 0, 0, 1, ...]
[2]	3	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	-3	[20, 0, 1, 2, 0, 0, ...]	[31, -13, 15, -9, 3, 3, ...]	[1, 0, 1, 0, 1, 0, ...]
[2]	4	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	3	[2, 3, 6, 0, 1, 0, ...]	[31, 7, -9, -13, -1, 3, ...]	[0, 0, 0, 1, 1, 1, ...]
[2]	5	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	3	[4, 2, 6, 0, 1, 0, ...]	[31, 7, -9, -13, -1, 3, ...]	[1, 0, 0, 0, 1, 1, ...]
NSSG	1	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	12	[9, 2, 4, 5, 1, 0, ...]	[31, -5, 3, 15, 11, -1, ...]	[1, 0, 0, 1, 1, 0, ...]
NSSG	2	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	13	[7, 2, 4, 2, 0, 0, ...]	[31, 3, 3, -1, 11, 3, ...]	[1, 1, 1, 1, 1, 0, ...]
NSSG	3	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	13	[7, 5, 4, 6, 0, 0, ...]	[31, 3, 3, -1, 11, 3, ...]	[1, 0, 1, 0, 1, 0, ...]
NSSG	4	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	13	[8, 2, 4, 2, 2, 0, ...]	[31, 3, 9, -13, 11, 3, ...]	[1, 1, 0, 1, 0, 0, ...]
NSSG	5	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	13	[10, 4, 4, 1, 0, 0, ...]	[31, -9, 3, 2, 11, 3, ...]	[1, 1, 1, 1, 1, 0, ...]
NSSG	1	2	155	[150, 155, 90, 150, 150]	[0, 1, 0, 0, 1]	22	5	[13, 4, 2, 3, 0, 1, ...]	[31, 7, -1, 3, 3, 7, ...]	[1, 0, 1, 0, 0, 0, ...]
NSSG	2	2	155	[150, 155, 90, 150, 150]	[0, 1, 0, 0, 1]	22	5	[9, 2, 4, 0, 0, 0, ...]	[31, -1, -1, -5, -1, 7, ...]	[1, 0, 1, 0, 1, 1, ...]
NSSG	3	2	155	[150, 155, 90, 150, 150]	[0, 1, 0, 0, 1]	21	5	[10, 5, 2, 3, 0, 0, ...]	[31, -1, -1, -1, -1, -1, ...]	[1, 1, 1, 0, 1, 1, ...]
NSSG	4	2	155	[150, 155, 90, 150, 150]	[0, 1, 0, 0, 1]	21	5	[7, 4, 1, 2, 1, 0, ...]	[31, -5, -1, -5, -1, 7, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	5	2	155	[150, 155, 90, 150, 150]	[0, 1, 0, 0, 1]	22	5	[10, 4, 2, 2, 5, 0, ...]	[31, -13, 15, -5, -1, 7, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	1	2	185	[150, 185, 90, 150, 150]	[1, 1, 0, 0, 1]	23	5	[12, 4, 4, 2, 0, 0, ...]	[31, -1, -1, -5, -1, 7, ...]	[1, 0, 1, 1, 1, 1, ...]
NSSG	2	2	185	[150, 185, 90, 150, 150]	[1, 1, 0, 0, 1]	22	5	[9, 4, 2, 2, 0, 0, ...]	[31, -1, -1, -5, -1, 7, ...]	[1, 0, 1, 0, 1, 1, ...]
NSSG	3	2	185	[150, 185, 90, 150, 150]	[1, 1, 0, 0, 1]	23	5	[9, 4, 1, 1, 0, 0, ...]	[31, -9, 15, -5, -9, 7, ...]	[1, 0, 0, 0, 1, 0, ...]
NSSG	4	2	185	[150, 185, 90, 150, 150]	[1, 1, 0, 0, 1]	22	5	[12, 4, 2, 2, 2, 0, ...]	[31, -1, 0, -5, -5, 1, ...]	[1, 1, 1, 0, 0, 1, ...]
NSSG	5	2	185	[150, 185, 90, 150, 150]	[1, 1, 0, 0, 1]	22	5	[11, 4, 2, 2, 1, 0, ...]	[31, 7, 5, -5, -13, 7, ...]	[1, 1, 1, 0, 0, 1, ...]

Table 3.10: LC for different OC, $n = 5$, $RR = 46, 155, 185$, and $RC = 2$.

have $MSR = 0.24$ and 0.30 and $LC = 14$.

Table 3.13 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 6$ after single cell replacement with $RR = 89$ and $OC = 2$. The SSG sequences and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this sequence is 8, which is higher than the sequence in [2]. The MSR is 0.94, which is higher than the sequence in [2] and does not satisfy the MSR criteria with $OC = 2$. There are three RR that give better sequences compared to the sequences with $RR = 89$ and they are given in Table 3.13. The NSSG sequences for $RR = 167, 151$, and 107 have $LC = 50$ and 46 which is higher than the sequence generated using $RR = 89$. The MSR for these three sequences is 0.17 or 0.30 and the balance is 5.

Table 3.14 gives the best sequence in [2] for $n = 5$ with the 1D CA SSG sequence of size $n = 6$ that passed the filtering criteria. This comparison is made because shrinking reduces the sequence length. The MSSG sequences did not pass the criteria. The LC for the sequence in [2] is 16, and the SSG sequences have $LC = 29$ and 30. The MSR for the sequence in [2] and the SSG sequences is 0.43.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	169	[90, 169, 150, 150]	[0, 0, 0, 1]	7	-1	[3, 0, 4, 0, 0, 0, ...]	0.47	[15, 3, -1, -9, -5, -5, ...]	[0, 1, 1, 1, 0, 0, ...]
SSG	2	2	231	[150, 231, 150, 150, 90]	[1, 0, 1, 0, 1]	5	4	[1, 0, 0, 0, 0, 0, ...]	0.33	[6, 2, 2, 2, 2, 2, ...]	[1, 1, 1, 1, 1, 0, ...]
MSSG	2	2	28	[90, 28, 150, 90, 90]	[1, 1, 0, 1, 1]	5	4	[1, 0, 0, 0, 1, 0, ...]	0.33	[6, 2, 2, 2, 2, 2, ...]	[1, 1, 1, 1, 1, 0, ...]

Table 3.11: Comparison of the best sequence in [2] for $n = 4$ with the SSG and MSSG sequences for $n = 5$.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[90, 90, 90, 90, 90, 150]	[1, 1, 0, 0, 1, 1]	14	3	[16, 8, 3, 3, 0, 0, ...]	0.24	[63, 3, -1, 7, 3, 3, ...]	[0, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[90, 150, 90, 150, 150, 90]	[1, 1, 0, 1, 0, 0]	14	3	[14, 9, 3, 4, 0, 1, ...]	0.30	[63, 3, -5, -1, -5, -1, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	2	2	[150, 90, 90, 90, 90, 90]	[1, 1, 0, 1, 1, 0]	14	3	[16, 8, 3, 3, 0, 0, ...]	0.24	[63, 3, -1, 7, 3, 3, ...]	[0, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[90, 150, 150, 90, 150, 90]	[1, 1, 0, 1, 1, 1]	14	3	[14, 9, 3, 4, 0, 1, ...]	0.30	[63, 3, -5, -1, -5, 11, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	2	2	[150, 90, 90, 90, 90, 90]	[1, 1, 1, 0, 0, 0]	14	3	[16, 8, 3, 3, 0, 0, ...]	0.24	[63, 3, -1, 7, 7, 3, ...]	[1, 0, 0, 0, 0, 0, ...]
NSSG	2	2	[90, 90, 90, 90, 90, 150]	[1, 1, 1, 0, 0, 1]	14	3	[14, 8, 5, 3, 0, 1, ...]	0.24	[63, 3, -5, -5, 3, 7, ...]	[1, 1, 1, 1, 0, 1, ...]

Table 3.12: Sequences for $n = 6$ and $OC = 2$ without an RR.

3.6 Filtered Results for $n = 7$

Table 3.15 gives the best 1D CA of size $n = 7$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria Without using an RR, the SSG, and MSSG sequences failed the criteria. Table 3.15 indicates that the NSSG sequences are the best and have $MSR = 0.21$ and 0.24 and $LC = 16$.

Table 3.16 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 7$ after single cell replacement with $RR = 18$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this sequence is 64, which is lower than the sequence in [2]. The MSR is 0.65 which is higher than the sequence in [2] and does not satisfy the MSR criteria with $OC = 2$. There are two RR that give better sequences compared to the sequences with $RR = 18$ and they are given in Table 3.16. The NSSG sequences for $RR = 181$, and 121 have $LC = 84$ and 76 which is higher than the sequence generated using $RR = 18$. The MSR for these two sequences is 0.40 and the balance is 25.

Table 3.17 gives the best sequence in [2] for $n = 6$ with the 1D CA SSG and MSSG sequence of size $n = 7$ that passed the filtering criteria. This comparison is made because shrinking reduces the sequence length. The LC for the sequence in [2] is 5, and the SSG and MSSG sequences have $LC = 64$ and 76. The MSR for the sequence in [2] is 0.87, and for the SSG and MSSG sequences it is 0.33 and 0.27.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	3	89	[90, 90, 89, 90, 90, 90]	[0, 0, 1, 0, 0, 1]	5	3	[1, 13, 12, 0, 0, 0, ...]	0.87	[63, 11, -37, -33, 11, 55, ...]	[0, 1, 1, 1, 0, 0, ...]
NSSG	2	3	89	[90, 90, 89, 90, 90, 90]	[0, 0, 1, 0, 0, 1]	8	5	[7, 6, 7, 6, 0, 0, ...]	0.94	[63, -61, 59, -57, 55, -53, ...]	[1, 0, 0, 0, 0, 1, ...]
NSSG	2	3	167	[150, 90, 167, 90, 90, 90]	[0, 0, 0, 0, 1, 1]	50	5	[12, 11, 5, 2, 0, 1, ...]	0.17	[63, 3, -9, 3, -1, -9, ...]	[1, 1, 0, 0, 1, 1, ...]
NSSG	2	3	151	[90, 90, 151, 150, 150, 90]	[1, 1, 0, 1, 0, 1]	46	5	[12, 10, 5, 4, 0, 0, ...]	0.30	[63, 3, -17, -5, 3, 11, ...]	[0, 1, 1, 1, 0, 1, ...]
NSSG	2	3	107	[90, 90, 107, 150, 150, 90]	[1, 1, 1, 0, 1, 0]	46	5	[12, 10, 5, 4, 0, 0, ...]	0.30	[63, 3, -17, -5, 3, 11, ...]	[0, 1, 1, 1, 0, 1, ...]

Table 3.13: Comparison of the best sequence in [2] with the NSSG sequences for $n = 6$, $OC = 2$, and $RR = 89, 167, 151, 107$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	146	[90, 146, 150, 150, 90]	[0, 0, 0, 0, 1]	16	-1	[14, 1, 5, 0, 0, 0, ...]	0.43	[31, -5, 3, -9, 15, -5, ...]	[0, 1, 0, 0, 0, 1, ...]
SSG	2	2	75	[90, 75, 90, 150, 90, 90]	[0, 0, 0, 1, 1, 1]	29	5	[16, 6, 5, 0, 4, 0]	0.43	[63, 3, 3, -5, -9, -1, ...]	[1, 1, 1, 0, 1, 0, ...]
SSG	2	2	135	[90, 135, 150, 150, 150, 150]	[0, 0, 1, 1, 1, 0]	30	1	[15, 4, 10, 0, 2, 0, ...]	0.43	[63, 3, -5, -9, 3, -5, ...]	[1, 0, 1, 1, 0, 1, ...]

Table 3.14: Comparison of the best sequence in [2] for $n = 5$ with the SSG sequences for $n = 6$.

3.7 Filtered Results for $n = 8$

Table 3.18 gives the best 1D CA of size $n = 8$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria. Without using an RR, the SSG and MSSG sequences failed the criteria. Table 3.18 indicates that the NSSG sequences are the best and have $MSR = 0.29$ and $LC = 18$.

Table 3.19 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 8$ after single cell replacement with $RR = 225$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this NSSG sequence is 127, which is close to the sequence in [2] with $OC = 2$. The value MSR for the NSSG sequence is 0.61, which is higher than the sequence in [2] and does not satisfy the MSR criteria. There is one RR that give better sequence compared to the sequences with $RR = 225$ and that is given in Table 3.19. The NSSG sequence for $RR = 89$ has $LC = 131$ which is higher than the sequence generated using $RR = 225$. The MSR for the sequence is 0.35 and the balance is 110.

Table 3.20 gives the best sequence in [2] for $n = 7$ with the 1D CA SSG sequence of size $n = 8$. This comparison is made because shrinking reduces the sequence length. The MSSG sequences did not pass the criteria. The LC for the sequence in [2] is 67, and the SSG sequence has $LC = 128$ and 76. The MSR for the sequence in [2] is 0.24, and for the SSG sequence it is 0.23.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[150, 90, 150, 90, 150, 90, 90]	[0, 0, 0, 0, 1, 0, 1]	16	7	[29, 16, 8, 9, 0, 1, ...]	0.24	[127, 3, -9, -5, -13, 3, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[90, 150, 150, 150, 150, 90, 150]	[0, 0, 1, 0, 0, 0, 0]	16	9	[28, 17, 9, 8, 0, 1, ...]	0.21	[127, 3, -13, -1, -9, -1, ...]	[0, 0, 1, 1, 1, 1, ...]
NSSG	2	2	[150, 150, 150, 90, 90, 90, 150]	[0, 0, 1, 1, 0, 0, 1]	16	9	[29, 15, 10, 8, 0, 1, ...]	0.21	[127, 3, -9, -13, -1, 3, ...]	[0, 1, 0, 0, 1, 0, ...]
NSSG	2	2	[150, 90, 90, 90, 150, 150, 150]	[0, 0, 1, 1, 1, 0, 0]	16	9	[29, 15, 10, 8, 0, 1, ...]	0.21	[127, 3, -9, -13, -1, 3, ...]	[0, 1, 0, 0, 1, 0, ...]
NSSG	2	2	[150, 90, 90, 150, 150, 90, 150]	[0, 1, 0, 0, 0, 0, 1]	16	7	[34, 12, 9, 6, 0, 1, ...]	0.24	[127, 3, 15, 3, 19, 3, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[150, 90, 150, 150, 90, 90, 150]	[0, 1, 0, 0, 1, 1, 0]	16	7	[34, 12, 9, 6, 0, 1, ...]	0.24	[127, 3, 15, 3, 19, 3, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	[150, 150, 150, 90, 90, 90, 150]	[0, 1, 0, 1, 1, 1, 1]	16	9	[29, 15, 10, 8, 0, 1, ...]	0.21	[127, 3, -9, -13, 3, -1, ...]	[0, 0, 1, 0, 1, 1, ...]

Table 3.15: Sequences for $n = 7$ and $OC = 2$ without an RR.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	6	18	[90, 150, 150, 150, 90, 18, 150]	[1, 1, 1, 0, 0, 1, 0]	67	49	[35, 17, 8, 2, 4, 1, ...]	0.24	[127, -5, -5, -5, 3, -13, ...]	[1, 1, 0, 1, 1, 0, ...]
NSSG	2	6	18	[90, 150, 150, 150, 90, 18, 150]	[1, 1, 1, 0, 0, 1, 0]	64	43	[37, 17, 9, 4, 0, 1, ...]	0.65	[127, -1, -1, -1, -1, -5, ...]	[0, 1, 0, 1, 1, 0, ...]
NSSG	2	2	181	[90, 181, 90, 150, 90, 90, 90]	[0, 0, 0, 0, 0, 0, 1]	84	25	[32, 22, 9, 6, 0, 0, ...]	0.40	[127, -9, -17, -5, -9, -5, ...]	[1, 1, 0, 1, 1, 0, ...]
NSSG	2	2	121	[150, 121, 150, 150, 90, 150, 150]	[0, 0, 0, 0, 0, 1, 1]	76	25	[44, 17, 11, 4, 0, 0, ...]	0.40	[127, -25, -1, -5, 15, -9, ...]	[1, 0, 0, 1, 0, 1, ...]

Table 3.16: Comparison of the best sequence in [2] with the NSSG sequences for $n = 7$, $OC = 2$, and $RR = 18, 181, 121$.

3.8 Filtered Results for $n = 9$

Table 3.21 gives the best 1D CA of size $n = 9$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that passed the filtering criteria. Without using an RR, the SSG and MSSG sequences failed the criteria. Table 3.21 indicates that the NSSG sequences are the best and have $MSR = 0.27$ and $LC = 18$ and 20.

Table 3.22 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 9$ after single cell replacement with $RR = 163$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this sequence is 256, which is close to the sequence in [2]. The MSR for the NSSG sequence is 0.52, which is higher than the sequence in [2] and does not satisfy the MSR criteria with $OC = 2$. There is no other RR that gave a better sequence.

Table 3.23 gives the best sequence in [2] for $n = 8$ with the 1D CA MSSG sequence of size $n = 9$. This comparison is made because shrinking reduces the sequence length. The SSG sequences did not pass the criteria. The LC for the sequence in [2] is 128, and the MSSG sequence has $LC = 441$. The MSR for the sequence in [2] is 0.23, and for the MSSG sequence it is 0.27.

3.9 Filtered Results for $n = 10$

Table 3.24 gives the best 1D CA of size $n = 10$ sequences generated without using an RR and using linear rules 90 and 150, $OC = 2$, $RC = 2$, and all non-zero SV that

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	3	89	[90, 90, 89, 90, 90, 90]	[0, 0, 1, 0, 0, 1]	5	3	[1, 13, 12, 0, 0, 0, ...]	0.87	[63, 11, -37, -33, 11, 55, ...]	[0, 1, 1, 1, 0, 0, ...]
SSG	2	2	166	[90, 166, 150, 150, 90, 90, 90]	[0, 0, 0, 0, 0, 0, 0, 1]	64	49	[255, 11, -5, -9, 15, 59, ...]	0.33	[127, 3, -5, -9, 3, -5, ...]	[1, 0, 0, 1, 1, 0, ...]
MSSG	2	2	110	[90, 110, 90, 90, 90, 90, 150]	[1, 1, 0, 0, 1, 0, 1]	76	25	[133, 11, 35, -9, 0, 4, ...]	0.27	[127, 1, -15, 8, 3, -1, ...]	[1, 0, 0, 0, 0, 1, ...]

Table 3.17: Comparison of the best sequence in [2] for $n = 6$ with the SSG and MSSG sequences for $n = 7$.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[150, 150, 90, 90, 90, 150, 150, 90]	[0, 0, 0, 1, 0, 0, 0, 0]	18	1	[65, 30, 16, 14, 0, 2, ...]	0.29	[255, -1, 3, -21, -13, 31, ...]	[1, 0, 0, 0, 0, 1, ...]
NSSG	2	2	[90, 90, 90, 90, 150, 150, 90, 150]	[0, 0, 0, 1, 0, 0, 1, 1]	18	1	[69, 27, 16, 9, 0, 1, ...]	0.29	[255, 3, 27, 7, 15, 31, ...]	[1, 1, 0, 0, 1, 0, ...]
NSSG	2	2	[150, 90, 150, 150, 90, 90, 90, 90]	[0, 0, 0, 1, 0, 1, 1, 0]	18	1	[69, 27, 16, 9, 0, 1, ...]	0.29	[255, 3, 27, 7, 15, 31, ...]	[1, 1, 0, 0, 1, 0, ...]
NSSG	2	2	[150, 90, 150, 150, 90, 90, 90, 90]	[0, 0, 0, 1, 1, 0, 0, 0]	18	1	[71, 25, 16, 10, 0, 2, ...]	0.29	[255, -1, 27, 7, 11, 31, ...]	[1, 0, 0, 0, 0, 0, ...]
NSSG	2	2	[90, 150, 150, 90, 90, 90, 150, 150]	[0, 0, 1, 0, 0, 0, 1, 0]	18	1	[62, 33, 19, 11, 0, 2, ...]	0.29	[255, -1, -9, -17, -9, 27, ...]	[0, 0, 1, 0, 1, 1, ...]
NSSG	2	2	[150, 150, 90, 90, 90, 150, 150, 90]	[0, 0, 1, 0, 1, 0, 0, 0]	18	1	[64, 31, 17, 13, 0, 2, ...]	0.29	[255, -1, -1, -17, -9, 27, ...]	[0, 0, 0, 1, 0, 0, ...]

Table 3.18: Sequences for $n = 8$ and $OC = 2$ without an RR.

passed the filtering criteria. Without using an RR, the SSG and MSSG sequences failed the criteria. Table 3.24 indicates that the NSSG sequences are the best and have $MSR = 0.29$ and 0.30 and $LC = 22$ and 18.

Table 3.25 gives the best sequence in [2] and NSSG sequence for a 1D CA of size $n = 10$ after single cell replacement with $RR = 154$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this sequence is 502, which is lower than the sequence in [2]. The MSR for the NSSG sequence is 0.51, which is higher than the sequence in [2] and does not satisfy the MSR criteria with $OC = 2$. There is one RR that gives a better sequence compared to the sequences with $RR = 154$ and is given in Table 3.25. The NSSG sequence for $RR = 185$ has $LC = 443$. The MSR for the sequence is 0.29 and the balance is 45.

Table 3.26 gives the best sequence in [2] for $n = 9$ with the 1D CA SSG sequence of size $n = 10$. This comparison is made because shrinking reduces the sequence length. The MSSG sequences did not pass the criteria. The LC for the sequence in [2] is 257, and the SSG sequence has $LC = 502$. The MSR for the sequence in [2] is 0.13, and for the SSG sequence it is 0.10.

3.10 Observations for $n = 3$ to 10

The sequences generated after a single cell replacement for 1D CAs of size $n = 3, 4, 5, 6, 7, 8, 9$ and 10 were filtered based on the criteria in Section 2.2, and the following observations can be made.

1. The LC of the sequences remains approximately the same when the OC (in-

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	225	[90, 150, 90, 225, 150, 150, 90, 150]	[1, 1, 1, 1, 0, 0, 0, 0]	128	55	[57, 31, 19, 8, 4, 2, ...]	0.23	[255, 11, -5, -9, 15, 59, ...]	[1, 0, 1, 1, 0, 0, ...]
NSSG	2	4	225	[90, 150, 90, 225, 150, 150, 90, 150]	[1, 1, 1, 1, 0, 0, 0, 0]	127	121	[67, 33, 11, 7, 0, 4, ...]	0.61	[255, 11, 19, 91, 63, -13, ...]	[0, 1, 1, 0, 0, 1, ...]
NSSG	2	4	89	[90, 150, 90, 89, 150, 150, 90, 150]	[1, 0, 0, 1, 1, 0, 0, 0]	131	110	[57, 133, 511, 7, 0, 4, ...]	0.35	[255, 201, -119, 41, 63, -13, ...]	[1, 0, 0, 0, 0, 1, ...]

Table 3.19: Comparison of the best sequence in [2] with the NSSG sequence for $n = 8$, $OC = 2$, and $RR = 89, 225$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	6	18	[90, 150, 150, 150, 90, 18, 150]	[1, 1, 1, 0, 0, 1, 0]	67	49	[35, 17, 8, 2, 4, 1, ...]	0.24	[127, -5, -5, -5, 3, -13, ...]	[1, 1, 0, 1, 1, 0, ...]
SSG	2	4	125	[90, 150, 90, 125, 150, 150, 90, 150]	[1, 0, 0, 1, 0, 0, 0, 0]	128	55	[57, 31, 19, 8, 4, 2, ...]	0.23	[255, 11, 5, 9, 15, 59, ...]	[1, 0, 1, 1, 0, 0, ...]

Table 3.20: Comparison of the best sequence in [2] for $n = 7$ with the SSG sequence for $n = 8$.

cluding the end cells) is varied with all other parameters fixed. Examples of this behavior for $n = 3, 4$ and 5 are given in Tables 3.3, 3.6, and 3.9, respectively. In each of the tables, LR, RR, RC and SV are constant, and only OC is varied. The value of OC has a negligible effect, so it is fixed for further analysis.

2. The sequences generated after a single cell replacement with all RR in the range 1 to 256 for 1D CAs of size $n = 3, 4, 5, 6, 7, 8, 9$, and 10, were filtered, and only 21 RR gave the best sequences that passed the filtering criteria. These RR are $11, 15, 18, 22, 30, 89, 107, 121, 122, 144, 146, 151, 154, 155, 163, 167, 169, 173, 181, 183,$ and 225. These RR are used for CA sizes $n > 10$.

3. The 1D CA evaluation system was first used to analyze all possible combinations of LR in CAs of size $n = 3, 4, 5, 6, 7, 8, 9$, and 10. The outputs were stored in Excel files and then filtered to obtain the combinations that generated the best sequences. These combinations are used for further analysis.

4. For 1D CAs of different sizes with fixed values of RC, OC and LR, all possible combinations of SV for $n = 3, 4, 5, 6, 7, 8, 9$, and 10 were tested that generate non-zero sequences. $SV = 0$ was ignored as it gives all-zero sequences. All SV results were then filtered to obtain the combinations that generated the best sequences. These combinations are used to reduce the computational complexity for larger sizes.

Based on these observations, the best sequences for $n = 11$ to $n = 18$ are obtained where the SV, LR and OC are fixed, and only the best RR given above are considered to reduce the computational complexity and execution time.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[90, 150, 150, 150, 150, 150, 90, 90, 90]	[0, 1, 0, 0, 1, 1, 0, 0, 1]	20	1	[133, 61, 32, 24, 0, 4, ...]	0.27	[511, -1, 19, -1, 15, 3, ...]	[0, 0, 0, 1, 0, 0, ...]
NSSG	2	2	[90, 90, 150, 90, 90, 90, 150, 90, 150]	[0, 1, 0, 0, 1, 1, 0, 1, 1]	18	1	[129, 63, 30, 23, 0, 5, ...]	0.27	[511, 3, 11, 15, 19, -13, ...]	[1, 0, 1, 0, 1, 0, ...]
NSSG	2	2	[150, 90, 90, 90, 90, 90, 150, 150, 150]	[0, 1, 0, 0, 1, 1, 1, 1, 0]	20	3	[124, 68, 33, 23, 0, 1, ...]	0.27	[511, 3, -13, 3, -1, -1, ...]	[1, 0, 1, 1, 1, 0, ...]
NSSG	2	2	[150, 90, 90, 90, 90, 90, 150, 150, 150]	[0, 1, 0, 0, 1, 1, 1, 1, 1]	20	1	[123, 67, 34, 22, 0, 3, ...]	0.27	[511, 3, -13, 3, -5, -5, ...]	[1, 0, 1, 1, 1, 0 ...]
NSSG	2	2	[90, 150, 90, 150, 150, 150, 90, 90, 90]	[0, 1, 1, 0, 1, 1, 1, 1, 1]	20	1	[123, 67, 34, 22, 0, 3, ...]	0.27	[511, 3, -13, 3, -5, 11, ...]	[1, 0, 1, 1, 1, 1 ...]
NSSG	2	2	[90, 150, 150, 150, 150, 150, 90, 150, 150]	[0, 1, 1, 1, 1, 0, 0, 0, 1]	18	1	[129, 63, 30, 23, 0, 5, ...]	0.27	[511, 3, 11, 15, 19, -13, ...]	[1, 0, 1, 0, 1, 0, ...]
NSSG	2	2	[90, 90, 90, 150, 150, 150, 150, 150, 90]	[0, 1, 1, 1, 1, 1, 0, 0, 1]	20	1	[133, 61, 32, 24, 0, 4, ...]	0.27	[511, -1, 19, -1, 15, 3, ...]	[0, 0, 0, 1, 0, 0, ...]

Table 3.21: Sequences for $n = 9$ and $OC = 2$ without an RR.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	8	163	[150, 90, 90, 150, 150, 90, 90, 163, 150]	[0, 0, 0, 0, 0, 1, 0, 1, 0]	257	3	[128, 61, 31, 19, 6, 3, ...]	0.13	[511, 3, 7, 11, -9, 3, ...]	[0, 0, 0, 1, 0, 0, ...]
NSSG	2	8	163	[150, 90, 90, 150, 150, 90, 90, 163, 150]	[0, 0, 0, 0, 0, 1, 0, 1, 0]	256	1	[126, 66, 32, 18, 0, 5, ...]	0.52	[511, 15, -29, 47, 7, 7, ...]	[1, 0, 1, 1, 1, 1, ...]

Table 3.22: Comparison of the best sequence in [2] with the NSSG sequence for $n = 9$, $OC = 2$, and $RR = 163$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	225	[90, 150, 90, 225, 150, 150, 90, 150]	[1, 1, 1, 1, 0, 0, 0, 0]	128	55	[57, 31, 19, 8, 4, 2, ...]	0.23	[255, 11, -5, -9, 15, 59, ...]	[1, 0, 1, 1, 0, 0, ...]
MSSG	2	2	154	[90, 154, 150, 90, 90, 90, 150, 90, 150]	[0, 1, 0, 0, 1, 1, 0, 1, 1]	441	1	[129, 23, 0, 123, 0, 15, ...]	0.27	[511, 13, 101, 5, 43, -1, ...]	[1, 0, 1, 0, 1, 0, ...]

Table 3.23: Comparison of the best sequence in [2] for $n = 8$ with the MSSG sequence for $n = 9$.

TYPE	OC	RC	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	[150, 90, 90, 90, 90, 90, 150, 90, 90, 90]	[0, 0, 0, 0, 0, 1, 1, 1, 0, 1]	22	5	[267, 116, 60, 54, 0, 5, ...]	0.29	[1023, 3, 43, -1, 47, 47, -33, ...]	[1, 0, 1, 1, 1, 0, ...]
NSSG	2	2	[90, 90, 90, 90, 90, 90, 150, 90, 150, 150]	[0, 0, 0, 0, 1, 0, 1, 0, 1, 0]	22	45	[241, 132, 72, 55, 0, 6, ...]	0.29	[1023, 3, -53, -33, 15, 31, ...]	[1, 1, 1, 1, 0, 0, ...]
NSSG	2	2	[90, 90, 90, 150, 150, 90, 90, 90, 90, 90]	[0, 0, 0, 0, 1, 1, 0, 0, 1, 1]	18	9	[253, 126, 70, 47, 0, 7, ...]	0.30	[1023, 3, -5, -13, -1, 3, ...]	[1, 1, 1, 0, 0, 1, ...]
NSSG	2	2	[90, 90, 90, 90, 90, 150, 150, 90, 90, 90]	[0, 0, 0, 0, 1, 1, 0, 1, 1, 1]	18	9	[266, 117, 61, 53, 0, 5, ...]	0.30	[1023, 3, -5, -13, -1, 3, ...]	[1, 1, 1, 0, 0, 1, ...]
NSSG	2	2	[150, 90, 90, 90, 90, 90, 150, 90, 90, 90]	[0, 0, 0, 1, 0, 1, 1, 0, 1, 1]	22	5	[267, 116, 60, 53, 0, 5, ...]	0.29	[1023, 3, 43, -1, 47, 51, ...]	[1, 1, 0, 1, 1, 1, ...]
NSSG	2	2	[150, 90, 90, 90, 90, 90, 150, 90, 90, 90]	[0, 0, 0, 0, 0, 0, 1, 1, 1, 1]	18	5	[267, 116, 61, 54, 0, 5, ...]	0.29	[1023, 3, 43, -1, 47, 47, ...]	[1, 0, 1, 1, 1, 0, ...]

Table 3.24: Sequences for $n = 10$ and $OC = 2$ without an RR.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	154	[150, 150, 150, 154, 150, 150, 90, 90, 90, 90]	[0, 0, 0, 0, 0, 0, 1, 1, 1, 1]	514	12	[259, 128, 63, 34, 16, 7, ...]	0.10	[1023, -5, 3, -5, -17, 3, ...]	[0, 0, 0, 0, 1, 1, ...]
NSSG	2	4	154	[150, 150, 150, 154, 150, 150, 90, 90, 90, 90]	[0, 0, 0, 0, 0, 0, 1, 1, 1, 1]	502	9	[243, 135, 71, 46, 0, 12, ...]	0.51	[1023, -9, -21, 63, -1, 59, ...]	[1, 0, 1, 0, 1, 1, ...]
NSSG	2	4	185	[150, 150, 150, 185, 150, 90, 90, 90, 90, 150]	[1, 1, 0, 0, 1, 0, 1, 1, 0, 1]	443	45	[267, 135, 61, 0, 0, 12, ...]	0.29	[1023, -53, 21, 63, 47, 3, ...]	[1, 1, 1, 0, 0, 0, ...]

Table 3.25: Comparison of the best sequence in [2] with the NSSG sequences for $n = 10$, $OC = 2$, and $RR = 154, 185$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	163	[150, 90, 90, 163, 150, 90, 90, 90, 150]	[0, 0, 0, 0, 0, 1, 0, 1, 0]	257	3	[128, 61, 31, 19, 6, 3, ...]	0.13	[511, 3, 7, 11, -9, 3, ...]	[0, 0, 0, 1, 0, 0, ...]
SSG	2	4	127	[150, 90, 150, 127, 150, 150, 90, 150, 150, 150]	[1, 1, 0, 0, 0, 0, 1, 1, 0, 1]	502	26	[259, 128, 63, 34, 16, 7, ...]	0.10	[1023, -21, 63, -75, -17, 93, ...]	[0, 0, 0, 0, 1, 1, ...]

Table 3.26: Comparison of the best sequence in [2] for $n = 9$ with the SSG sequence for $n = 10$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	86	[90, 150, 90, 86, 150, 90, 90, 90, 90, 150, 90]	[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]	1024	12	[512, 256, 128, 64, 32, 16, ...]	0.06	[2047, -29, -9, -57, -5, -25, ...]	[0, 0, 0, 0, 0, 0, ...]
NSSG	2	4	86	[90, 150, 90, 86, 150, 90, 90, 90, 90, 150, 90]	[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]	1027	25	[512, 256, 128, 64, 32, 16, ...]	0.16	[2047, -57, -5, -25, -5, 3, ...]	[1, 1, 0, 0, 1, 0, ...]
NSSG	2	4	225	[90, 150, 90, 225, 150, 90, 90, 90, 90, 150, 90]	[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]	1032	25	[512, 133, 98, 164, 128, 162, ...]	0.25	[2047, -17, -23, 5, 5, 63, ...]	[1, 1, 1, 0, 0, 0, ...]

Table 3.27: Comparison of the best sequence in [2] with the NSSG sequences for $n = 11$, $OC = 2$, and $RR = 86, 225$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	154	[150, 150, 150, 154, 150, 150, 90, 90, 90, 90]	[0, 0, 0, 0, 0, 0, 1, 1, 1, 1]	514	12	[259, 128, 63, 34, 16, 7, ...]	0.13	[1023, -5, 3, -5, -17, 3, ...]	[0, 0, 0, 0, 1, 1, ...]
SSG	2	4	127	[90, 150, 90, 127, 150, 90, 90, 90, 90, 150, 90]	[1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0]	502	26	[259, 128, 63, 34, 16, 7, ...]	0.13	[1023, -21, 63, -75, -17, 93, ...]	[0, 0, 0, 0, 1, 1, ...]

Table 3.28: Comparison of the best sequence in [2] for $n = 10$ with SSG sequence for $n = 11$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	99	[90, 99, 150, 90, 90, 150, 150, 150, 150, 90, 150, 90]	[1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1]	2046	15	[1022, 510, 262, 122, 68, 36, ...]	0.23	[4095, -1, -9, -25, -25, 7, ...]	[1, 0, 1, 1, 1, 1, ...]
NSSG	2	2	99	[90, 99, 150, 90, 90, 150, 150, 150, 150, 90, 150, 90]	[1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1]	2049	18	[1027, 505, 260, 125, 78, -2, ...]	0.99	[4095, -1, 7, -17, -45, 35, ...]	[0, 1, 1, 1, 1, 1, ...]
NSSG	2	2	163	[90, 163, 150, 90, 90, 150, 150, 150, 150, 90, 150, 90]	[1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1]	2559	25	[1027, 505, 260, 125, 78, -2, ...]	0.33	[4095, 51, 66, 32, -45, -35, ...]	[0, 1, 1, 1, 1, 1, ...]

Table 3.29: Comparison of the best sequence in [2] with the NSSG sequences for $n = 12$, $OC = 2$, and $RR = 99, 163$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	4	86	[90, 150, 90, 86, 150, 90, 90, 90, 90, 150, 90]	[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]	1024	12	[512, 256, 128, 64, 32, 16, ...]	0.06	[2047, -29, -9, -57, -5, -25, ...]	[0, 0, 0, 0, 0, 0, ...]
MSSG	2	4	225	[90, 150, 90, 225, 150, 90, 90, 90, 90, 150, 90, 90]	[0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0]	1226	26	[1027, 228, 123, 34, 16, 7, ...]	0.31	[4095, -521, 163, 475, 217, 93, ...]	[1, 1, 0, 0, 1, 1, ...]

Table 3.30: Comparison of the best sequence in [2] for $n = 11$ with MSSG sequence for $n = 12$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	163	[150, 163, 150, 150, 150, 150, 150, 150, 150, 90, 150, 150, 150]	[1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1]	4098	22	[2048, 1024, 512, 256, 128, 64, ...]	0.43	[8191, -2641, 21, 51, 47, 47, -33, ...]	[1, 0, 1, 1, 1, 0, ...]
NSSG	2	2	225	[90, 225, 90, 90, 90, 90, 90, 90, 90, 150, 90, 150, 150]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	4100	22	[4040, 672, 340, 172, 90, 42, ...]	0.25	[8191, -21, 19, -1397, 1327, 2047, ...]	[0, 1, 1, 1, 0, 0, ...]

Table 3.31: Sequences for $n = 13$, $OC = 2$, and $RR = 163, 255$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
[2]	2	2	99	[90, 99, 150, 90, 90, 150, 150, 150, 150, 90, 150, 90]	[1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1]	2046	15	[1022, 510, 262, 122, 68, 36, ...]	0.23	[4095, -1, -9, -25, -25, 7, ...]	[1, 0, 1, 1, 1, 1, ...]
SSG	2	2	127	[90, 127, 150, 90, 150, 150, 90, 90, 90, 150, 90, 90, 150]	[1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1]	3997	21	[4027, 1228, 253, 434, 46, 90, ...]	0.31	[8191, 1521, 3163, 175, 57, 93, ...]	[1, 0, 0, 1, 0, 1, ...]

Table 3.32: Comparison of the best sequence in [2] for $n = 12$ with SSG sequence for $n = 13$.

3.11 Filtered Results for $n = 11$ to 18

Table 3.27 gives the best sequence in [2] and NSSG sequences for a 1D CA of size $n = 11$ after single cell replacement with $RR = 86$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this sequence is 1027, which is higher than the sequence in [2]. The MSR for the NSSG sequence is 0.16 which is higher than the sequence in [2] with $OC = 2$. There is one RR that gives a better sequence compared to the sequences with $RR = 86$ and is given in Table 3.27. The NSSG sequence for $RR = 225$ has $LC = 1032$. The MSR for this sequence is 0.25 and the balance is 25.

Table 3.28 gives the best sequence in [2] for $n = 10$ with the 1D CA SSG sequence of size $n = 11$. This comparison is made because shrinking reduces the sequence length. The MSSG sequences did not pass the criteria. The LC for the sequence in [2] is 514, and the SSG sequence has $LC = 502$. The MSR for the sequence in [2] is 0.13, and for the SSG sequence it is also 0.13.

Table 3.29 gives the best sequence in [2] and NSSG sequences for a 1D CA of size $n = 12$ after single cell replacement with $RR = 99$ and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequence passed. The LC of this sequence is 2049, which is higher than the sequence in [2] with $OC = 2$. The MSR for the NSSG sequence is 0.99, which is lower than the sequence in [2]. There is one RR that gives a better sequence compared to the sequences with $RR = 99$ and is given in Table 3.29. The NSSG sequence for $RR = 163$ has $LC = 2559$. The MSR for this sequence is 0.33 and the balance is 25.

Table 3.30 gives the best sequence in [2] for $n = 11$ with the 1D CA MSSG sequence of size $n = 12$. This comparison is made because shrinking reduces the sequence length. The SSG sequences did not pass the criteria. The LC for the sequence in [2] is 1024, and the MSSG sequence has $LC = 1226$. The MSR for the sequence in [2] is 0.06, and for the MSSG sequence it is also 0.31.

Table 3.31 gives the best sequences for a 1D CA of size $n = 13$ after single cell replacement which have $RR = 163$ and 225 and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while these NSSG sequences passed. The LC of these sequences is 4098 and 4100, the MSR is 0.43 and 0.25, and the balance is 22.

Table 3.32 gives the best sequence in [2] for $n = 12$ with the 1D CA SSG sequence of size $n = 13$. This comparison is made because shrinking reduces the sequence

length. The MSSG sequences did not pass the criteria. The LC for the sequence in [2] is 2046, and the SSG sequence has $LC = 3997$. The MSR for the sequence in [2] is 0.23, and for the SSG sequence it is 0.31.

Table 3.33 gives the best sequences for a 1D CA of size $n = 14$ after single cell replacement which have $RR = 11, 89, 163$ and 225 and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequences passed. The LC of these sequences is 7102, 7592, 8012 and 8193, the MSR is 0.21, 0.33, 0.35 and 0.23, and the balance is 65, 71, and 77.

Table 3.34 gives the best sequences for a 1D CA of size $n = 15$ after single cell replacement which have $RR = 15, 107$ and 173 and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while this NSSG sequences passed. The LC of these sequences is 10100, 10812 and 10992, the MSR is 0.31, 0.34 and 0.41, and the balance is 361 and 367.

Table 3.35 gives the best sequences for a 1D CA of size $n = 16$ after single cell replacement which has $RR = 178$ and 213 and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while these NSSG sequences passed. The LC of these sequences is 16112 and 16378, the MSR is 0.31, and the balance is 307 and 327.

Table 3.36 gives the best sequences for a 1D CA of size $n = 17$ after single cell replacement which have $RR = 121, 163$ and 183 and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while these NSSG sequences passed. The LC of these sequences is 18514, 18378, and 18992, the MSR is 0.23, 0.29 and 0.41, and the balance is 418 and 420.

Table 3.37 gives the best sequences for a 1D CA of size $n = 18$ after single cell replacement which have $RR = 89, 144$ and 163 and $OC = 2$. The SSG and MSSG sequences failed to pass the filtering criteria discussed in Section 2.2 while these NSSG sequences passed. The LC of these sequences is 17054, 22514, and 19514, the MSR is 0.33, 0.23 and 0.35, and the balance is 511, 515.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	11	[90, 11, 150, 90, 150, 90, 150, 150, 150, 90, 150, 90, 150, 150]	[1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1]	7102	65	[2014, 4584, 1221, 408, 612, 201, ...]	0.21	[16383, 169, 235, -911, 109, 41, ...]	[1, 1, 1, 1, 0, 0, ...]
NSSG	2	2	89	[90, 89, 150, 90, 150, 90, 150, 150, 150, 90, 150, 90, 150, 150]	[0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1]	7592	71	[3114, 1004, 2501, 213, 142, -110, ...]	0.33	[16383, -169, -65, -219, -99, 11, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	163	[90, 163, 150, 90, 150, 90, 150, 150, 150, 90, 150, 90, 150, 150]	[1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1]	8012	77	[5013, -1094, 6131, -512, 452, 66, ...]	0.35	[16383, -69, -65, -9, 199, 11, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	225	[90, 225, 150, 90, 150, 90, 150, 150, 150, 90, 150, 90, 150, 150]	[0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1]	8192	77	[4114, 2084, 1021, 508, 242, 130, ...]	0.23	[16383, -69, -65, -9, 199, 11, ...]	[0, 0, 1, 1, 0, 1, ...]

Table 3.33: Sequences for $n = 14$, $OC = 2$, and $RR = 11, 89, 163, 225$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	15	[90, 15, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150]	[1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1]	10100	367	[7897, 1278, 5711, 5788, 211, 315, ...]	0.31	[32767, 1295, -3957, 6343, -2011, 4295, ...]	[1, 0, 1, 0, 0, 1, ...]
NSSG	2	2	107	[90, 107, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150]	[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1]	10812	361	[7897, 4488, -1704, 6718, 126, 0, ...]	0.34	[32767, 3215, -1397, 3343, -4101, 1505, ...]	[1, 0, 0, 0, 0, 1, ...]
NSSG	2	2	173	[90, 173, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150]	[0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1]	10992	367	[7897, 2788, 2714, 2788, 0, 0, ...]	0.41	[32767, 395, -397, -5313, 5011, 8375, ...]	[0, 1, 0, 1, 0, 1, ...]

Table 3.34: Sequences for $n = 15$ and $OC = 2$ with an $RR = 15, 107, 173$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	178	[90, 178, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150]	[0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0]	16112	307	[14897, 2788, 2714, 2788, 0, 0, ...]	0.31	[64767, 1158, 2797, -5003, 5011, 8375, ...]	[0, 1, 0, 1, 0, 1, ...]
NSSG	2	2	213	[90, 213, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90]	[0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0]	16378	327	[16431, 2793, 1487, 620, 326, 174, ...]	0.31	[64767, 7542, 9635, 967, 963, 963, ...]	[0, 1, 0, 1, 0, 1, ...]

Table 3.35: Sequences for $n = 16$, $OC = 2$, and $RR = 178, 213$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	121	[90, 121, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150, 150]	[1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1]	18514	418	[16895, 2155, 2714, 2788, 4173, 0, ...]	0.23	[128767, 8395, 1597, 2313, 5011, 8375, ...]	[1, 1, 1, 1, 0, 1, ...]
NSSG	2	2	163	[90, 163, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150, 150]	[1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1]	18378	420	[18151, 4173, 3647, 150, 326, 155, ...]	0.29	[128767, 8375, 9635, 1597, 963, 963, ...]	[1, 1, 0, 1, 0, 1, ...]
NSSG	2	2	183	[90, 183, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150, 150]	[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]	18992	418	[18231, 2793, 3287, 620, 0, 174, ...]	0.41	[128767, 15421, 6589, 125, 788, 963, ...]	[0, 1, 1, 1, 1, 1, ...]

Table 3.36: Sequences for $n = 17$, $OC = 2$, and $RR = 121, 163, 183$.

TYPE	OC	RC	RR	LR	SV	LC	B	R	MSR	AC	S
NSSG	2	2	89	[90, 89, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150, 90, 90]	[1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1]	17054	511	[19875, 4811, 3114, 2877, 1256, 521, ...]	0.33	[256327, 12983, 6197, 2113, 1651, 3035, ...]	[1, 1, 0, 0, 0, 1, ...]
NSSG	2	2	144	[90, 144, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150, 90, 90]	[1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1]	22514	515	[18875, 5178, 3071, 1278, 7273, 1220, ...]	0.23	[256327, 20783, 51897, 5201, 4501, 1335, ...]	[1, 1, 0, 0, 0, 1, ...]
NSSG	2	2	163	[90, 163, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 150, 90, 90]	[1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1]	19514	515	[19875, 1278, 2771, 4278, 8273, 1220, ...]	0.35	[256327, 20783, 51897, 5201, 4501, 1335, ...]	[1, 0, 0, 1, 1, 1, ...]

Table 3.37: Sequences for $n = 18$, $OC = 2$, and $RR = 89, 144, 163$.

3.12 Execution Time

The 1D CA evaluation system was implemented on a computer with a Ryzen 7-4800 processor with 16 cores, 16 GB RAM, and the 64-bit Linux Ubuntu 22.04 operating system. The execution time depends on the number of iterations which is given by

$$(2^n - 1) \cdot 2^n \cdot 255 \tag{3.1}$$

so each increment in n increases the execution time by approximately 4. Table 3.38 gives the execution times in seconds for $n = 3$ to 10 and Table 3.39 gives the execution times for $n = 11$ to 18. These execution times were based on the values of SV, LR and RR defined to give the best sequences, as discussed in Section 3.10.

n	Iterations	Time (s)
3	14,280	0.99
4	61,200	8.81
5	252,960	12.58
6	1,028,160	56.26
7	4,145,280	235
8	16,646,400	846
9	66,716,160	3372
10	267,125,760	16448

Table 3.38: Execution times for $n = 3$ to 10.

n	Time (s)
11	38.64
12	151
13	616
14	2745
15	9318
16	37269
17	146357
18	582709

Table 3.39: Execution times for $n = 11$ to 18.

Chapter 4

Conclusion

The objective of this thesis was to use 1D CA and shrinking generators to produce pseudorandom sequences with high linear complexity and good randomness. An evaluation system was developed to test sequences generated by 1D CAs and shrinking generators which contain combinations of linear rules (rules 90 and 150) with single cells replaced by a non-linear rule (rules 1 to 256). Initially, an extensive investigation was done on CAs of sizes $n = 3, 4, 5, 6, 7, 8, 9$, and 10. All SV, LR and RR combinations were tested to get the best sequences. The same evaluation system was used for further analysis for $n = 11$ to 18. The generated sequences were filtered based on the criteria for balance, run and autocorrelation. The criteria for balance and run were $P_B > 0.01$ and $P_R > 0.01$, respectively, for all start values. For autocorrelation, the filtering criterion was a maximum sidelobe ratio of $MSR < 0.3$.

The results of the filtering process indicated that LC remains approximately the same when OC was varied. An extensive investigation of RR was conducted and it was found that there were 21 RR that gave the best NSSG sequences, and those sequences passed the filtering criteria. These RR were 11, 15, 18, 22, 30, 89, 107, 121, 122, 144, 146, 151, 154, 155, 163, 167, 169, 173, 181, 183, and 225, and only these RR were used for investigation with $n = 11$ to 18. The SSG and MSSG sequences failed to pass the filtering criteria, and only the NSSG sequences passed when all RR were tested. These NSSG sequences were compared with the best sequences in [2]. The LC of the NSSG sequences was high compared to the LC of the sequence in [2] but do not always satisfy the criterion for autocorrelation as with some SV, the MSR was greater than 0.3. However, without using the RR, the best sequences always satisfy the criterion for autocorrelation as the MSR was below 0.3. Based on the observations for $n = 3$ to 10, the best sequences for $n = 11$ to 18 were obtained where

the SV, LR and OC were fixed and only the best RR given above were considered to reduce the computational complexity and thus execution time.

Another comparison was made because shrinking reduces the sequence length and due to this reason, only the NSSG sequences passed the filtering criteria. In this comparison, the best sequence in [2] for $n = 3$ to 12 is compared with SSG and MSSG sequences for $n = 4$ to 13 and it was found that the SSG and MSSG sequences passed the filtering criteria. The LC of the SSG and MSSG sequences was high compared to the LC of the best sequence in [2]. In conclusion, the 1D CA and shrinking generators can be used to generate pseudorandom sequences of high linear complexity and good randomness.

4.1 Future Work

In this thesis, the 1D CA and shrinking generator evaluation system was used to generate pseudorandom sequences. The best sequences were obtained using the shrinking generators and the filtering criteria. 2D CAs and shrinking generators can be considered in the future to generate sequences. This thesis analyzed sequences generated by a single replacement of a linear rule by a non-linear rule. In the future, multiple replacements can be considered.

Bibliography

[1] A. Stevenson, Oxford Dictionary of English. Oxford University Press, 3rd Edition, Oxford, England, 2010.

[2] S. Acharya. "Cellular Automata Pseudorandom Sequence Generation". M.A.Sc. Thesis, Dept. of Electrical and Computer Eng., University of Victoria, Victoria, BC, Canada, 2017.

[3] M. Aggarwal and T.A. Gulliver. "A New Self-Shrinking Generator". Preprint, 2022.

[4] K. Cattell and J.C. Muzio. "Synthesis of One Dimensional Linear Hybrid Cellular Automata". *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 15, no. 3, pp. 325-335, 1996.

[5] D. Coppersmith, H. Krawczyk, and Y. Mansour. "The Shrinking Generator". In *Advances in Cryptology*, Lecture Notes in Computer Science, vol. 773, pp. 22-39. Springer-Verlag, Berlin, Germany, 1993.

[6] A. Kanso and A. Ali Adel. "Modified Self-Shrinking Generator". *Electrical and Computer Eng.*, vol. 36, no. 5, pp. 993-1001, 2001.

[7] M.G. Kendall and B.B. Smith. "Randomness and Random Sampling Numbers". *Journal of the Royal Statistical Society*, vol. 101, no. 1, pp. 147-160, 1938.

[8] W. Meier and O. Staffelbach. "The Self-Shrinking Generator". In *Advances in Cryptology*, Lecture Notes in Computer Science, vol. 950, pp. 205-214. Springer-Verlag, Berlin, Germany, 1995.

[9] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. Handbook of Applied Cryptography. CRC Press, Boca Raton, FL, USA, 2001.

[10] A. Mitra. "On the Properties of Pseudo Noise Sequences with a Simple Proposal of Randomness Test". *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 2, no. 9, pp. 1997-2002, 2008.

[11] G.H. Norton. "The Berlekamp-Massey Algorithm via Minimal Polynomials. *arXiv preprint arXiv:1001.1597*, 2010.

[12] A.L. Rukhin, J. Soto, J.R. Nechvatal, M.E. Smid, E.B. Barker, S.D. Leigh, M. Levenson, M. Vangel, D.L. Banks, N.A. Heckert, J.F. Dray, S.C. Vo, and L.E. Bassham III. "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications". rev. 1a, National Institute of Standards and Technology (NIST) and U.S. Department of Commerce, 2010.

[13] R. Scurr. "Sequences and Cellular Automata". ENEL 427 Final Report, University of Canterbury, Christchurch, New Zealand, 1998.

[14] T.E. Tkacik. "A Hardware Random Number Generator". In *International Workshop on Cryptographic Hardware and Embedded Systems*, Lecture Notes in Computer Science, vol. 2523, pp. 450-453. Springer, Berlin, Germany, 2002.

[15] S. Wolfram. Cellular Automata and Complexity: Collected Papers. Westview Press, 1st Edition, Boulder, CO, USA, 1994.

[16] S. Wolfram. A New Kind of Science. Wolfram Media, Champaign, IL, USA, 2002.

[17] S. Wolfram. "Random Sequence Generation by Cellular Automata". *Advances in Applied Mathematics*, vol. 7, no. 2, pp. 123-169, 1986.