# Graph-Based Modelling of Maximum Period Property for Nonlinear Feedback Shift Registers

Eric Filiol[1,3] and Pierre Filiol[2]
[1]*Thales Digital Factory, Thales Group, Paris, France*
[2]*Lab-STICC, ENSTA Bretagne, Brest, France*
[3]*ENSIBS, Vannes, France*

Keywords: NLFSR, Stream Cipher, Binary Sequence, Maximum Period, Graph Representation, Incidence Matrix.

Abstract: NonLinear Feedback Shift Registers (NLFSRs) are key primitives to design pseudorandom generators in modern stream ciphers, especially when the feedback function is of low degree. Contrary to their linear counterparts (LFSRs) for which a general and comprehensive theory has been established, many fundamental problems related to NLFSRs remain open. In particular finding a systematic procedure of acceptable complexity for constructing NLFSRs with a guaranteed long period is still a general open problem and only a few results have been obtained so far. In this paper, we present the results of a exhaustive exploratory search and analysis of NLFSRs of low degree. We first model NLFSRs as graphs using their incidence matrix and express the maximum period property as graph properties. This enables to reduce the number of possible candidates greatly that can be tested finally for the maximum period property by HPC on GPGPUs and Massively Parallel Processor Array (MPPA).



## 1 INTRODUCTION

Binary sequences produced by feedback shift registers (FSRs) are widely used in stream ciphers and random generators. These registers are the key primitive used in these cryptographic systems.

A binary $n$-stage feedback shift register is defined as a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$

$$(x_{n-1},\ldots,x_0) \mapsto (f(x_{n-1},\ldots,x_0),x_{n-1},\ldots,x_1) \quad (1)$$

where $f$ is a Boolean function, called feedback function, $\mathbb{F}_2$ denotes the binary field and $\mathbb{F}_2^n$ the $n$-dimensional vector space over $\mathbb{F}_2$ consisting of the $n$-tuples of elements of $\mathbb{F}_2$. Whenever $f$ is a linear transformation, we deal with a *Linear Feedback Shift Register* (LFSR) otherwise ($f$ is nonlinear) with a Nonlinear Feedback Shift Register (NLFSR). In this paper, we focus on NLFSRs defined by a bijective mapping (nonsingular mapping).

Consider a binary sequence $\sigma = (\sigma_i)_{i=0}^{\infty}$. From the $n$ first fixed terms $\sigma_0,\sigma_1,\ldots,\sigma_{n-1}$ (called the initial state vector), we derived the register output sequence uniquely defined by the recurrence relation for all $i > 0$:

$$\sigma_{n+i} = f(\sigma_i,\sigma_{i+1},\ldots,\sigma_{i+n-1}) \quad (2)$$

If there exists an integer $p > 0$ such that $\sigma_{i+p} = \sigma_i$ for all $i \in \mathbb{F}_2^n \setminus \{(0,0,\ldots,0)\}$, the sequence is called periodic of period $p$. The most desirable property for NLFSRs (as well as for LFSRs) is to have a maximal possible period length of $2^n - 1$. If we iterate $f$ over $\mathbb{F}_2^n$, we then have two cycles (see Figure 1): one of length 1 (the loop over the single point $(0,0,\ldots,0)$ and a cycle of length $2^n - 1$. In this case, NLFSRs generate maximal length sequences or *m-sequences* (Golomb, 1981).

From the cryptographic or random number generation perspective, it is strongly desirable that NLFSRs' feedback function fulfils the following conditions (Augustynowicz, 2018):

- The number of feedback function's linear and nonlinear terms should remain as small as possible. It is especially a desirable property for hardware implementation (number of logic gates).
- The algebraic degree of the feedback function should be the lowest possible (at least 2 however).

The goal is somehow that NLFSRs are as close as possible to all the main advantages of LFSRs such as low power consumption, easy implementation and high efficiency while providing a better resistance against known attacks (Kuznetsov et al., 2022, page 5).



Filiol, E. and Filiol, P.
Graph-Based Modelling of Maximum Period Property for Nonlinear Feedback Shift Registers.
DOI: 10.5220/0012839300003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 832-837
ISBN: 978-989-758-709-2; ISSN: 2184-7711

The paper hence focus on nonlinear feedback functions whose Algebraic Normal Form (ANF) is given by:

$$f(x_{n-1},\ldots,x_0) = \sum_{i=0}^{n-1} c_i.x^i + x^n + x_j.x_k \qquad (3)$$

for all possible pair $j,k$ such that $0 < j,k < n$ and where $c_i$ are binary coefficients describing whether the register cell is considered ($c_i = 1$) or not ($c_i = 0$). We have conducted an exhaustive exploratory analysis to find all feedback functions up to the degree $n = 28$. To reduce the computing time it has been necessary to find a new way of NLFSR modelling. For that purpose, we represent NLFSR as directed graphs whose incidence matrix exhibits specific properties to express the maximal period property. It is worth mentioning that the present study can be easily applied to any other forms of feedback polynomials (for instance more quadratic terms).

It is the first exhaustive search to date whereas previous works only published a very few number of results due to the search complexity. From the results obtained, we have identified a number of news results that can be of high interest to explore further for $n > 28$.

In the rest of this article we will consider all operations in the finite field $(\mathbb{F}_2, +, .)$.

The paper is organized as follows. Section 2 analyses the overall complexity of searching of NLFSRs producing m-sequence and presents the related works. Section 3 introduces our combinatorial model for NLFSRs and formalizes the maximal period property in terms of algebraic equations. Then Section 4 details the particular implementation aspects that have been used to perform an effective computation. Finally Section 5 presents the detailed results of our exhaustive search and identify a few new interesting properties before concluding in Section 6.

# 2 PRELIMINARIES

## 2.1 Complexity Analysis of NLFSR Search

To date, there are no theoretical results that allow to easily find maximum period NLFSRs as is the case for LFSRs (Golomb, 1981). In this section we look at the approach currently being favoured in recent years. Let us rewrite (3) in a more simple way:

$$f(x_{n-1},\ldots,x_0) = l(x_{n-1},\ldots,x_0) + x_j.x_k \qquad (4)$$

where $l(x) = l(x_{n-1},\ldots,x_0) = \sum_{i=0}^{n-1} c_i.x_i + x^n$ is the linear part of $f$.

The search for such NLFSRs of length $n$ can be formalized according the two following steps:

1. Among the $2^n$ possible candidates corresponding $l(x)$, for a given pair $i, j$ fixing the degree-2 monomial, we retain those which validate a certain number of algebraic or combinatorial properties $I_1, I_2, \ldots, I_k$. If these properties are independent, with respective probabilities of being realized by a good candidate $P(I_i) = p_i$, then at the end of this stage we retain $N = 2^n.\prod_{i=1}^{k} p_i$. This step has an incompressible complexity of $2^{n-1}$ (a symmetry property presented in Section 2.3 enables to cut search work in half).
2. For each valid candidate for $l(x)$, we check whether it is in the maximum period by calculating the cycle. Complexity is in $2^n$ in the worst case. The average complexity is $2^{n-1}$ from (Golomb, 1981, Corollary 11, p. 183). No result is known for this step that would allow us to reduce this complexity (except in certain very specific cases, see (Golomb, 1981, Chapter VII)).

The overall worst-case complexity is therefore $2^{2n-1}$ and the average-case complexity is $2^{2n-2}$. To reduce this complexity, the focus must be on the first stage to reduce the number of candidates to be tested in the second stage. Some of these properties are already known (see Section 2.3). In this paper we are going to add many others thanks to results from graph theory and matrix calculus on the associated incidence matrix. This significantly reduces the overall complexity of the exhaustive search. This has enabled us to perform an exploratory analysis up to $n = 28$.

## 2.2 Related and Previous Work

Since the Jansen's seminal thesis in 1989 (Jansen, 1989), the main approach for searching for NLFSRs of the simplest form considers more or less sophisticated exhaustive search. In (Dabrowski et al., 2014), this approach has been initiated with parallel computing.

Later on in (Poluyanenko, 2017), the author studies NLFSRs implementation on FPGAs and discusses issues of their optimization. Search method of NLFSRs generating M-sequence was given. It was based on a practical synthesis and explores the possibility of NLFSR implementation of FPGA.

In (Augustynowicz, 2018), the authors consider a multi-stages hybrid algorithm which uses Graphics Processor Units (GPU) and developed for processing data-parallel throughput computation. They focus and give results for feedback polynomials of the form $l(x) + x_i.x_k + x_j.x_l$ (two quadratic monomials). Later

the same authors (Augustynowicz and Kanciak, 2020) optimize their search methods by applying particular vector processor instructions. Their aim was to maximize the advantage of Single Instruction Multiple Data (SIMD) and Single Instruction Multiple Threads (SIMT) execution patterns. Their results are only partial and contains errors (especially regarding the number of sparse feedback polynomials, see Section 5).

Finally in 2022, the authors of (Poluyanenko, 2017) have extended their results given in (Kuznetsov et al., 2022), giving a bit more feedback polynomials. Unfortunately only a very few number of results are obtained due to the high computing complexity.

As a result, no exhaustive enumeration of maximum period NLFSR polynomials is yet available. Such a result would perhaps allow to identify unsuspected properties which could help to find a general theory of construction of these NLFSRs as is the case for their linear counterparts (LFSRs). This is the aim of the present article.

Studies considering exhaustive search all try to reduce the first step of search described in Section 2 by using a very few number of algebraic properties satisfied by the feedback coefficients $a_i$. We recall them in Section 2.3.

### 2.3 Known Algebraic Properties

Golomb (Golomb, 1981) in 1982 identifies a very important property that defines the condition for a shift register cycle to be branchless. In other words, the feedback function is bijective: each $x$ has a unique successor $f(x)$ and any $f(x)$ has a unique predecessor (see Fig. 2).

**Theorem 1.** *(Golomb, 1981) The cycles generated by a feedback shift register have no branch points if and only if its feedback function can be decomposed as*

$$f(x_0,\ldots,x_{n-l}) = x_0 + g(x_l,\ldots,x_{n-1}) \qquad (5)$$

It implies that any integer encoding the linear part $l(x)$ must be an odd value (monomial $x_0$ is always present). Later on Chan, Game and Rushanan (Chan et al., 1993) identified three more generic algebraic properties.

- The Hamming weight of the integer encoding $l(x)$ is even (we add the bit corresponding to monomial $x_0$ so $c_0 = 1$).
- The Hamming weight of the integer encoding $l(x)$ must be at least equal to 2.
- If $x_0 + g(x_l,\ldots,x_{n-1})$ generates a quadratic $m$-sequence, then $x_0 + g(x_{n-1},\ldots,x_1)$ generates a quadratic $m$-sequence as well. Then we can divide the search over half the pairs $\{i,j\}$ defining the degree 2 monomial in Equation 4. Each time we have a solution, we generate the conjugate solution for replacing all indices $i$ by $n-i$ in Equation 4.

In the next section, we present how to have more statistically independent such equations for a larger reduction.

## 3 NLFSR AND GRAPH INCIDENCE MATRIX

Modelling NLFSRs using graph incidence matrices was first mentioned by Gonzalo, Ferrero and Soriano (Gonzalo et al., 2002) in a rather imprecise and succinct manner. No results were given. No analysis of the independence of potential equations was presented. The computational and algorithmic aspects were not discussed, even though they are fundamental as soon as $n > 10$. Indeed, the size of the matrix grows exponentially with $n$, which limits their approach to small values of $n$. Our work is based on their approach, which we have effectively implemented and optimised.

### 3.1 Combinatorial Model for NLFSR

Let us consider a NLFSR of size $n$ whose feedback polynomial has the general form $f(x) = x^n + \sum_{i=0}^{n-1} c_i.x_i + x_j.x_k$ for $0 < j,k < n$. For each possible pair $\{j,k\}$ we search for all $n$-uples $(c_0,\ldots,c_{n-1})$ for which the NLFSR fulfils the maximum period property. We can model any NLFSR by a directed graph of $2^n$ points. Any maximal period NLFSR more precisely is a $(2^n,2^n)$-graph with two cycles: one cycle of length 1 (loop on the null point) and one cycle of length $2^{n-1}$.

To illustrate our approach, let us consider the feedback function $x_4 + x_2 + x_1 + x_0 + x_1.x_2$ denoted for short as $0,1,2,(1,2)$. Fig. 1 describes the two corresponding cycles.

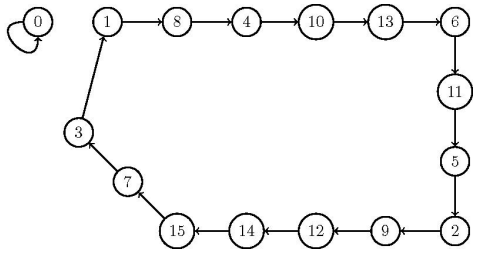


Figure 1: Graph for $0,1,2,(1,2)$.

To simplify notations for $i = (i_{n-1},\ldots,i_1,i_0) \in$

$\mathbb{F}_2^n$ let us note $\vec{i_0} = (0, i_{n-1}, \ldots, i_1)$ and $\vec{i_1} = (1, i_{n-1}, \ldots, i_1)$ the right-shifted versions $i$ whose most significant bit is updated respectively with 0 and 1. This notation describes the state of an NLFSR, which changes from state $i$ to state $\vec{i_0}$ or $\vec{i_1}$ depending on the feedback value $f(i)$ of the NLFSR calculated on state $i$.

Let us consider its incidence matrix whose entries are expressed as linear polynomials in the unknown $c_i$ as follows: $\forall i = (i_{n-1}, i_{n-2}, \ldots, i_1, i_0) \in \mathbb{F}_2^n$ then $f(i)$ equals 0 or 1 and hence produce either $\vec{i_0}$ or $\vec{i_1}$ depending on the values of coefficients $c_i$. We can then define the formal incidence matrix $A = [a_{i,j}]$ where $i$ and $j$ are in $\mathbb{F}_2^n$

$$\forall i \in \mathbb{F}_2^n \setminus \{(0,0,\ldots,0)\} \begin{cases} a_{i,\vec{i_0}} = & 1 + f(i) \\ a_{i,\vec{i_1}} = & f(i) \end{cases} \quad (6)$$

For instance let us consider the formal feedback polynomials $f(x) = x_4 + c3.x_3 + c_2.x_2 + c_1.x_1 + c_0 + x_1.x_2$ and $i = (0,1,1,1)$. We have $a_{i,\vec{i_1}} = c_2 + c_1$ and $a_{i,\vec{i_0}} = 1 + c_2 + c_1$. If we consider the formal feedback polynomial $f'(x) = x_3 + c_2.x_2 + c_1.x_1 + x_0 + x_1.x_2$, the corresponding formal incidence matrix of order $2^n - 1 = 7$ is given in (7).

It is important to note that for $i = (0,0,1)$ and $i' = (1,1,1)$ have only one possible successor if we want the cycle to have maximal period (respectively $\vec{i_1}$ and $\vec{i'}_1$. Recall also from Theorem 1 that $c_0 = 1$. We can observe that the sum of matrix entries linewise is always equal to 1 as well as columnwise (in fact $c_0 = 1$).

### 3.2 Formalisation of Max-Period Property

The Max-Period property can be expressed in different ways (see Figure 1).

It is a well known result that each entry $a_{i,j}^k$ of a power matrix $A^k$ describes the number of paths of length $k$ between $i$ and $j$ (Brouwer and Haemers, 2012). If we want a NLFSR be in maximal period then there must be no loop ($a_{i,i}^k = 0$) and there must exist a unique path between between $i$ and $j$ (refer to Fig. 2) for any value of $k \geq 1$.

Exploiting this formalisation efficiently requires to manage two issues:

- A complexity issue. If almost all matrix computation have polynomial complexity, the actual complexity is exponential since the size of the matrix is in $O(2^n)$. We then need to find ways of managing this explosion in complexity.
- The different equations have to be statistically independent in order to minimize their number. Optimally $n$ statistically independent equations should drastically reduce the number of suitable candidates at the end of the first step of the exhaustive search.

To obtain several independent algebraic equations while limiting the computing effort, we calculated the successive powers $A^2, A^4, A^8 \ldots, A^{2^k}$ of the incidence matrix $A$. The results confirm that this approach does indeed yield statistically independent equations. The probability for a candidate to satisfy all of diagonal equations is indeed $\frac{1}{2^k}$. By restricting ourselves to $k \leq \frac{n}{4}$, the computational effort remains moderate while guaranteeing a very significant reduction in the first stage of the exhaustive search described in Section 2. It is worth noting that the statistical independence of the diagonal equations obtained is compliant with the fact that the cycle-length distribution is flat (Golomb, 1981, Section 2.2). We have evaluated the statistical independence of these additional equations and we have observed that they are always satisfied if the diagonal equation is. So they do not bring any new bit of information. It means that each squaring iteration provides only one significant equation.

## 4 COMPUTATIONAL APPROACH

Modelling the maximum-period NLFSR search problem using graphs means that their incidence matrices are square matrices of size $N = 2^n - 1$. As far as naive matrix multiplication is concerned, the computation of the diagonal equations is of cubic complexity but the size of the data is of exponential complexity in $2^{n-1}$. It is therefore not possible to compute the matrix products directly (naive product of matrices) as soon as $n > 18$.

However these matrices are extremely sparse. The initial matrix defined in Equation 7 has $2^{n+1} - 4$ entries out of $2^{2n} - 2^{n+1} + 1$ possible entries. The matrix sparsity is asymptotically defined by $\mathcal{S}(A) = \frac{1}{2^{n-1}}$ when $n \to \infty$.

We used a specific compact matrix representation. There are several possible forms of representation and we have opted for the form described in Table 1. The matrix entries are not integer or real values, but formal polynomials whose maximum number of monomials is $2^{2^n}$. This maximum number is never reached in practice (for small values of $k$), which means that effective calculations can be carried out with limited memory requirements. This form consists of storing the $(i, j)$ coordinates of the only non-zero inputs, together with their polynomials. We give this representation for the matrix (7) and in its general expression in Table 1.

$$A = \begin{pmatrix} 0 & 0 & 0 & c_0 & 0 & 0 & 0 \\ 1+c_1 & 0 & 0 & 0 & c_1 & 0 & 0 \\ 1+c_0+c_1 & 0 & 0 & 0 & c_0+c_1 & 0 & 0 \\ 0 & 1+c_2 & 0 & 0 & 0 & c_2 & 0 \\ 0 & c_0+c_2+1 & 0 & 0 & 0 & c_0+c_2 & 0 \\ 0 & 0 & c_1+c_2 & 0 & 0 & 0 & 1+c_1+c_2 \\ 0 & 0 & c_0+c_1+c_2 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (7)$$

Table 1: Compact Representation of (7) (left) - General Case (right).

| | | |
|---|---|---|
| 1 | 4 | $c_0$ |
| 2 | 1 | $1+c_1$ |
| 2 | 5 | $c_1$ |
| 3 | 1 | $1+c_0+c_1$ |
| 3 | 5 | $c_0+c_1$ |
| 4 | 2 | $1+c_2$ |
| 4 | 6 | $c_2$ |
| 5 | 2 | $c_0+c_2+1$ |
| 5 | 6 | $c_0+c_2$ |
| 6 | 3 | $c_1+c_2$ |
| 6 | 7 | $1+c_1+c_2$ |
| 7 | 3 | $c_0+c_1+c_2$ |

| | | |
|---|---|---|
| ... | ... | ... |
| i | $\vec{i}_0$ | ... |
| i | $\vec{i}_1$ | ... |
| i + 1 | $(\vec{i+1})_0$ | ... |
| i + 1 | $(\vec{i+1})_1$ | ... |
| ... | ... | ... |

This form of matrix product computation is fast, efficient and optimizes memory resources. The algorithm is described in Table 1. Note that for clarity of presentation, the form described in Table 1 has been implemented a bit differently so as to allow direct access to the various items without having to use local search in the table (thus eliminating tests).

**Data:** Matrix $A$ of size $N$
**Result:** Compute $A^2$
Allocate result table *MatRes* of size $2.N$;
**for** *x from 1 to N* **do**
  `/* For nonzero entries of A */`;
  `/* Coord. (i, j) & polynomial */`;
  $i = A[x][1]$
  $j = A[x][2]$
  $p_x = A[i][3]$
  **for** *all y such that* $A[y][1] == j$ **do**
    `/* Nonzero entries (j,k) */`
    $p_y = A[j][y]$;
    $MatRes[i][A[y][2]] += p_x * p_y$;
  **end**
**end**
Return MatRes;

Algorithm 1: Fast Large Sparse Matrix Square Algorithm.

It is worth noting a few important points:

- Since the number of entries for the second `For` loop is constant, the overall complexity is that of main loop. Hence the overall complexity is in $O(N)$. However, for the initial matrix we have $N = 2^n - 1$. The number of non zero matrix entries roughly doubles with each squaring. So after $k$ squaring, the number of non zero entries is $2^k.N$.
- At the end of the squaring procedure, the result matrix *MatRes* is already ordered according to matrix line indices $i$. There is consequently no need of an additional sorting step.

# 5 RESULTS AND DISCUSSION

## 5.1 Results

We have applied our method to search exhaustively all NLFSRs with feedback polynomials of the form given by (3). Until now we have completed this search up to $n = 27$ (for $n = 28$ search is in progress). For small value of $n$ ($n \leq 20$), we also performed a naive exhaustive search in order to validate our algebraic approach by comparing both results obtained. Our approach has been fully confirmed.

This research work required four months of computing on an AMD Ryzen 32/64-core Linux machine with 256 Mb of RAM and equipped with a Kalray 256-core Massively Parallel Processor Array (MPPA) and a Nvidia RTX 2080 Ti GPGPU. We did not parallelize Algorithm 1, preferring to run threads on the different pairs $(i, j)$ for the monomial $x_i.x_j$.

Table 2 presents the definitive results for $15 \leq n \leq 26$. The $(i, j)$ rate describes the proportion of pairs $\{i, j\}$ for which at least one solution has been found. The minimal weight of $l(x)$ is of high importance since they provide the most simple form for hardware implementation while maintaining excellent non-linearity for cryptographic designs.

## 5.2 Analysis of Results

This exhaustive research enabled us first to compare our results with the few published ones. While the solutions we found systematically included and confirmed the few solutions already published, we were

Table 2: Results of Exhaustive Search.

| $n$ | # polynomials | $(i,j)$ rate | Min. weight of $l(x)$ |
|---|---|---|---|
| 15 | 204 | 0.7821 | 3 |
| 16 | 250 | 0.9428 | 3 |
| 17 | 302 | 0.9083 | 5 |
| 18 | 332 | 0.8235 | 5 |
| 19 | 404 | 0.8627 | 3 |
| 20 | 436 | 0.8596 | 5 |
| 21 | 554 | 0.8947 | 5 |
| 22 | 524 | 0.8666 | 5 |
| 23 | 568 | 0.8268 | 5 |
| 24 | 616 | 0.8650 | 5 |
| 25 | 756 | 0.8731 | 7 |
| 26 | 764 | 0.8933 | 7 |
| 27 | 737 | 0.8615 | 7 |
| 28 | $> 120$ | $> 0.3418$ | $\leq 7$ |

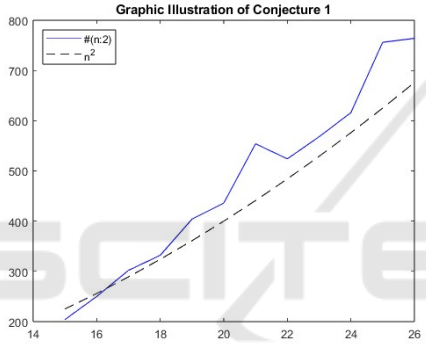


Figure 2: Graphic Illustration of Conjecture 1.

able to disprove certain results. For instance Augustynowicz and K. Kanciak (Augustynowicz and Kanciak, 2020, p. 21, Table VII) claim that no feedback polynomials of weight 7 does exist for $n = 26$ and $n = 27$ while we have found several ones. For instance $n = 27$ $x_0 + x_1 + x_2 + x_3 + x_7 + x_{10} + x_{18} + x_6.x_{14}$. A number of other of their results are also incomplete or wrong.

We have also initiated an in-depth analysis of the results to identify some interesting properties. For example, the number of solutions varies significantly according to the respective parities of the indices $i$ and $j$ of the monomial $x_i.x_j$.

Finally we can formulate the following conjecture concerning the number of polynomials as a function of $n$.

**Conjecture 1.** *The number of feedback polynomials of the form* (3)*, denoted* $\#(n:2)$*, is in* $O(n^2)$.

This Conjecture is illustrated in Figure 2.

# 6 CONCLUSION

In this paper we have presented how combinatorial modelling of an NLFSR via the incidence matrix of the state graph can help to significantly reduce the computational effort in the exhaustive search for feedback polynomials. This enabled us to carry out this search up to $n \leq 28$ for a minimal form that is very important in the design of encryption algorithms.

This exhaustive search will be carried out for $n > 28$. However, memory requirements quite soon exceeds current capacities ($n = 31$ requires a machine with 1Tb RAM). We are therefore considering emulating RAM with disk space and use MapReduce-type functions on Distributed File Systems (DFS) (Leskovec et al., 2014, Chap. 2). The computation time will be longer, but this is of relative importance for an exhaustive one-time search (once and for all search).

# REFERENCES

Augustynowicz, P. (2018). Scalable method of searching for full-period nonlinear feedback shift registers with gpgpu. new list of maximum period nlfsrs. *International Journal of Electronics and Telecommunications*, 64.

Augustynowicz, P. and Kanciak, K. (2020). The search of square m-sequences with maximum period via gpu and cpu. *Infocommunications Journal*, XI:17.

Brouwer, A. E. and Haemers, W. H. (2012). Spectra of Graphs. New York, NY

Chan, A. H., Games, R. A., and Rushanan, J. J. (1993). On quadratic m-sequences. In *Fast Software Encryption, Cambridge Security Workshop*, page 166–173, Berlin, Heidelberg. Springer-Verlag.

Dabrowski, P., Labuzek, G., Rachwalik, T., and Szmidt, J. (2014). Searching for nonlinear feedback shift registers with parallel computing. *Inf. Process. Lett.*, 114(5):268–272.

Golomb, S. W. (1981). *Shift Register Sequences*. Aegean Park Press, USA.

Gonzalo, R. P., Ferrero, D., and Soriano, M. (2002). Nonlinear feedback shift registers with maximum period.

Jansen, C. (1989). *Investigations on Nonlinear Streamcipher Systems: Construction and Evaluation Methods*. PhD thesis, TU Delft.

Kuznetsov, A., Potii, O., Poluyanenko, N., Gorbenko, Y., and Kryvinska, N. (2022). *Stream Ciphers in Modern Real-time IT Systems: Analysis, Design and Comparative Studies*.

Leskovec, J., Rajaraman, A., and Ullman, J. D. (2014). *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd edition.

Poluyanenko, N. (2017). Development of the search method for nonlinear shift registers using hardware, implemented on field programmable gate arrays. *EUREKA: Physics and Engineering*, 1:53–60.