Analyzing Cryptosystems by Using Artificial Intelligence

Faez Hassan Ali[1], Mohammed G. S. Al-Safi[2] and Ahmed Ayyoub Yousif[3]
[1] Department of Mathematics, College of Science, Al-Mustansiriyah University, Baghdad-Iraq.
[2] Department of Accounting, Al-Esra'a University College, Baghdad-Iraq.
[3] Department of Mathematics and Computer Applications, College of Science,
Al-Nahrain University, Baghdad-Iraq.
Corresponding author: ahmedayy79@gmail.com.

Abstract

This paper aims to apply the Bees Algorithm for solving system of equations. The solving System of Equations may be linear or nonlinear for a number of unknowns. As an application of System of Equations, we can implement cryptanalysis attack algorithms on stream cipher systems using plaintext attack (or part from it). We consider the Geffe System (which has nonlinear combining function) to be our study case, which is depend on set of Linear Feedback Shift Registers, as a model of stream cipher systems, in the performance of Bees Algorithm by solving System of Equations for any number of variables of the output of Linear Feedback Shift Registers.

The application divided into two stages, first, constructing System of Equations for the suggested cryptosystem, and the second, is attacking the variables of System of Equations which they are also represent the initial key values of the combined of Linear Feedback Shift Registers.

[DOI: 10.22401/ANJS.00.1.14]

Keywords: *Bees Algorithm, Cryptography, Stream Cipher Systems, Linear Feedback Shift Register, System of Nonlinear Equations.*

1. Introduction

Cryptanalysis is the science of studying the methods of analyzinging ciphers. It is a perfect system to identify the problem, while the aim of **Cryptography** is to constructing systems that are hard to be identified, [1]. To be able to attack a cryptosystem successfully, the cryptanalysis is forced to be based on some approaches, such as knowledge of a part of the text encrypted, knowing the characteristic properties of the used language, ect. with some luck.

The Cryptosystems are the systems which depend on the encryption and decryption processes.

One of an Artificial Intelligence (AI) techniques is the Swarm Intelligence (SI) which is an including the study of all collective behavior in decentralized systems. Such SI are made up by one (or more) population (s) of simple agents or individuals interacting locally or globally with one other and with their environment. Although there is decentralized control dictating the behavior of the particles, local interactions among the particles often cause some global pattern to be emerged. There are many instances of systems can be found in nature, involving honey bees, bird flocking, bacteria, ant colonies, animal herding and many more. Swarm-like algorithms, such as Bees Algorithm (BA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) have been applied successfully to solve real world optimization problems in engineering, economy and telecommunication, [2].

Ismail K. Ali. (2009) [3], in his thesis shows that PSO is a good tool for breaking simple transposition and simple substitution ciphers as long as bigram and trigram are used to find the fitness of each particle.

Ahmed T., et al. (2014) [4], introduces an improved cuckoo search algorithm for automata cryptanalysis of transposition ciphers. This algorithm is a search algorithm that is done by adding a procedure to cuckoo search steps analyzing the similarities between population strings to calculate the global maxima of a cost function to find the secret encryption (decryption) key.

Hameed F. A. (2017) [5] in her thesis, implements cryptanalysis system on stream cipher cryptosystems called (PSO) Cryptanalysis System using probable word plaintext attack, choosing different study cases, single Linear Feedback Shift Register

(LFSR), Linear cryptosystem and Threshold generator (as nonlinear cryptosystem) and she obtain good cryptanalysis results.

In this work, the non-linear stream cipher generator will be attacked using the Bee algorithm. The Geffe generator will be our study case. The attacking technique depends on solving the non-linear SoE of Geffe generator. The results show the good achievement of BA in stream cipher cryptanalysis.

2. Modern Cryptosystems

There are basically two different kinds of cryptographic systems (cryptosystems), these cryptosystems are: secret key and public cryptosystems, [6]. First let us redefine some important notations:

- **P** is the Plaintext message and **C** is the Ciphertext message.
- **Key space K**: a set of strings (keys) related to some alphabet, which includes the encryption key e_k and the decryption key d_k.
- The **Encryption** algorithm (process) E: $Ee_k(P) = C$.
- The **Decryption** algorithm (process) D: $Dd_k(C) = P$.
- The two algorithms E and D must have the property that: $Dd_k(C) = Dd_k(Ee_k(P)) = P$.

As it is known, the public key cryptosystem also called **asymmetric cryptosystems**. In a public key (**non-secret key**) cryptosystem, the encryption key e_k and decryption key d_k are not same, that is $e_k \neq d_k$. The secret Key Cryptosystems may also called **symmetric cryptosystems**. In a classical secret-key cryptosystem, the same key ($e_k = d_k = k \in K$), called **secret key**, are used in both encryption and decryption; our aim is this type of cryptosystems. The stream cipher cryptosystems is one of the important types of the secret key cryptosystems, [7].

3. Stream Cipher Systems

Stream ciphers can be considered as one of an important class of cryptosystems. They encrypt each digit (usually binary digits) independently, taken from a plaintext message on one time, using an encryption function or transformation which must be varies with time, [8].

As usual in stream ciphers, the message units are may be bits (or digits), and the key is usual produced by a some algorithms called **Pseudorandom bit generator** (PBG), as shown in Fig.(1). The plaintext message is encrypted on a method depend on bit-by-bit basis.

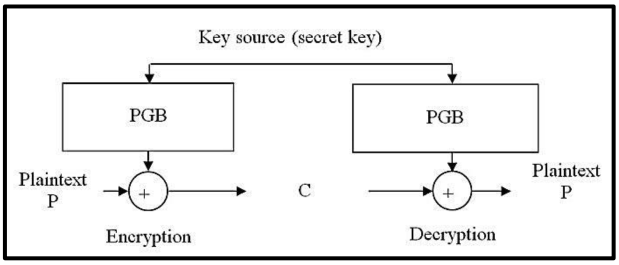

Fig.(1) Block Diagram of Stream cipher cryptosystem.

The secret key is fed into random bit generator to create a high long sequence of binary signals. This key-stream, k is mixed with plaintext p, usually by a bit wise Exclusive-OR modulo 2 addition (XOR) to produce the cipher text stream, using the same random bit generator and seed.

The LFSR's are used widely in cryptography, specially, in stream cipher cryptosystems. Every LFSR consists of two basic units, the LFSR connection function with suitable initial state values, while the other one is the Combining Function (CF), which is a considered as a boolean algebraic function. Most of the stream ciphers are depend on these two basic units.

Most practical stream ciphers are designed around LFSR. In the modern history of electronics, they were very easy to construct. A **shift register** can be defined as an array of number of bit memories and the feedback sequences are just a series or set of XOR logical gates. The stream cipher based on LFSR can give a high security and complexity with only a few logic gates, [9], [10].

4. Bees in Nature

Every bee in a colony search for the feed independently. A known the colony of bees can be extended over long distances and in many directions in the same time to exploit very large number of food sources. A colony grows by deploying its foragers to suitable fields. In basic, flower fields with good amounts of nectar or pollen that may be able to be collected with less effort should be visited by more bees, whereas patches with nectar or pollen should receive fewer bees. The foraging process begins in a colony by bees called scout bees being sent to search for promising flower patches. The scout bees move randomly from one patch to another.

During the harvesting season, the colony continues its exploration, keeping a percentage of the population as scout bees, [11].

4.1 Bees Algorithm (BA)

The challenge of the bees process is to adapt the self-organization behavior of the chosen colony for solving the problems, [12]. The BA can be considered as an optimization algorithm inspired by the natural foraging behavior of honey bees to found the optimal solution.

The algorithm requires a number of parameters to be set, namely:

a. Number of scout bees (n).
b. Number of sites selected out of n visited sites (m).
c. Number of best sites out of m selected sites (e).
d. Number of bees recruited for best e sites (nep).
e. Number of bees recruited for the other (m – e) selected sites (nsp).
f. Initial size of patches (ngh) which includes site and its neighborhood and stopping criterion.

The pseudo code for the Bees algorithm in its simplest form [13].

```
Bees Algorithm (BA)
Input: Number of (scout bees (n), sites selected out of n visited sites (m).
best sites out of m selected sites (e).
bees recruited for best e sites (nep).
bees recruited for the other (m-e) selected sites (nsp). Initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. Maximum of iterations).
Output: Optimal solutions.
step1.  Initialize population with random solutions.
step2.  Evaluate fitness of the population.
step3.  Repeat.
step4.  Select sites for neighborhood search.
step5.  Recruit bees for selected sites (more bees for best e sites) and evaluate fitness’s.
step6.  Select the fittest bee from each patch.
step7.  Assign remaining bees to search randomly and evaluate their fitness’s.
step8.  Until stopping criterion is met.
```

The BA starts with the "n" scout bees chosen randomly in the key search space. The fitness values of the sites which are visited by the scout bees are evaluated in step 2. In step 4, bees that have the best fitness’s are chosen as “selected bees” and sites are visited by them are picked for neighborhood search. Then, in steps 5 and 6, the BA conducts searches in the neighborhood of the selected sites, assigning more bees to search beside or near to the best "e" sites. The bees can be chosen directly according to the good fitness’s associated with the sites they are visiting. Alternatively, the high fitness values are used to find the probability of the bees being selected.

Searches in the local or global neighborhood of the best "e" sites which represent more promising solutions are made more detailed by recruiting more bees to follow them than the other selected good bees. Together with scouting, this differential recruitment is a good key operation of the BA. However, in step 6, for each patch, only the bee with the best fitness will be selected to form the next good bee population. In nature, there is no such a restriction. This restriction is introduced here to decrease the number of points to be explored.

In step 7, the remaining bees in the population are must be assigned randomly around the search space scouting for new potential solutions. These steps are must be repeated until a stopping criterion is met. At the end of search iterations, the colony will have two parts to its new population representatives from each chosen patch and other scout bees assigned to conduct random searches, [14].

4.2 The Parameters of BA

From our experience, the following parameters are adopted to be used: Number of Bees (N_Bee = 20, 30), Number of Jobs (n), Number of selected sites (m = 3 − 5), Number of elite sites out of m selected sites (e = 2), Number of bees for elite sites (nep = 5), Number of bees other selected points (nsp = 3) and some hundreds number of generations (NG).

5. Constructing System of Equations

Suppose that the tested LFSR is maximum LFSR (n-LFSR), then its period is $P = 2^L - 1$, where L is LFSR length. Let SR_L be a LFSR has length L, let $B_0=(\alpha_{-1},\alpha_{-2},\ldots, \alpha_{-L})$ be the initial vector of SR_L, s.t. a_{-j}, $1 \le j \le L$, be the component j of the vector B_0, this mean, a_{-j} is the initial bit of stage j of SR_L, let $C_0^T = (c_1,\ldots,c_L)$ be the feedback vector, $c_j \in \{0,1\}$, if $c_j = 0$ this means the stage j is unconnected else its connected. Let $S = \left\{s_i\right\}_{i=0}^{n-1}$ be the n-sequence or $S = (s_0,s_1,\ldots,s_{n-1})$ generated from SR_L. The generation of S is as follows:

$$s_i = \alpha_i = \sum_{j=1}^{L} \alpha_{i-j}c_j \quad i = 0,1,\ldots \quad \ldots\ldots\ldots\ldots\ldots (1)$$

Equation (1) represents the linear recurrence relation, [10].

Of course, the vectors L, C_0 and S are known, while the goal is finding the Vector B_0.

Let A be a L×L matrix, which is represents the initial phase of SR_L

$A = (C_0|I_{L\times L-1})$, where $A^0 = I$

Let B_1 be the new initial of SR_L after just one shift, such that

$B_1 = B_0\times A = (\alpha_{-1},\alpha_{-2},\ldots,\alpha_{-L})$

$$\begin{pmatrix} c_1 & 1 & \cdots & 0 \\ c_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ c_L & 0 & \cdots & 0 \end{pmatrix} = (\sum_{j=1}^{L} \alpha_{-j}c_j, \alpha_{-1},\ldots,\alpha_{1-L}).$$

In general,

$B_i = B_{i-1}\times A$, $i = 0,1,2,\ldots \ldots\ldots\ldots\ldots$ (2)

Then by using equation (2), we obtain:

$B_i=B_{i-1}\times A=B_{i-2}\times A^2 = \ldots = B_0\times A^i$ ...... …(3)

The matrix A^i can be considered the i phase of SR_L.

We will notice:

$A^2 = [C_1C_0|I_{L\times L-2}]$ and continue, so we get:

$A^i = [C_{i-1}\ldots C_0|I_{L\times L-i}]$, where $1 \le i < L$.

When $C_P = C_0$ then $A^{P+1} = A$.

Now we have to calculate C_i by:

$C_i = A\times C_{i-1}$, $i = 1,2,\ldots$ .... (4)

Lets rewritten relation (1) as:

$B_0\times C_i = s_i$, $i = 0,1,..,L-1$ .... (5)

In general:

$B_0\times\Psi = S$ .... (6)

Where Ψ represents the matrix of all vectors C_i s.t.

$\Psi = (C_0C_1\ldots C_{L-1})$ .... (7)

The system of equations (SoE) can be represented as:

$B = [\Psi^T|S^T]$ .... (8)

So B be the extended (augmented) matrix of the SoE.

6. Use BA to Attack Stream Cipher System

In this paper, BA will be used to solve SoE's of Geffe system with length n equations are needed to solve the SoE.

6.1 Problem Definition

In this manner, the SoE of Geffe cryptosystem, needs:

$n = L_1\times L_2+L_1\times L_3+L_3$ .... (9)

where L_i is length of register i, to solve the system.

6.2 Coding Scheme

For the purpose of this study, SoE is decoded by binary representation. As an example, the equation $\alpha_1+\alpha_7 = 1$ of just one LFSR with length 7 decoded by the equation string (1000001-1), where the absolute value (right side) of the equation is the real key of the cryptosystem. The equations can be constructed and kept to be used again, such that these equations are constant for fixed LFSR's length, connection polynomial and combining function. As this representation indicates, the size of the equations space is $2^n - 1$ (ignoring the zero string). When n as large as possible, then a purely random search is not acceptable.

6.3 Initial Population

For the initialization process we can initialize the population by a random sample of combinations of 0 and 1 with n-string length represents the probable initial values LFSR's. The creation of the population must submit to what we called non-zero initial condition. By this condition we can avoid the zero initial of LFSR's. For example, we wish to initiate initial values of LFSR with length 7, we ignore the initial value 0000000 for single LFSR or other cryptosystems. Another example the string 010011001<u>000</u> is ignored for linear cryptosystem consists of three LFSR's with lengths 5, 4 and 3 respectively, since the initial of the third register 000. The ***Initial Population Algorithm*** shown below describes the initial population process.

Initial Population Algorithm (IPA)

```
INPUT     : read NB, p; {Number of Bees}, {size of Population}
OUTPUT    : Population of Bees.
PROCESS   : for i = 1: NB
              for j = 1 : p
                 $s_j$ = RANDOM(2);
             end;
                 $Bee_i = (s_1,s_2,…,s_n)$;
             end;
END.
```

6.4 Evaluation Function (Fitness Function)

This function is used to determine the "best" representation. The process of the evaluation function selection is as follows:

1. From Population, a Bee k, k = 1,2,…,n initial string of length n bits, so we get the string $X_k = (X_{k1},X_{k2},…,X_{kn})$.
2. The string bit X_{kj} product with corresponding equation string bit Y_{ij}, where $1 \leq j \leq n$ such that the equation string is $Y_i = (Y_{i1},Y_{i2},…,Y_{in})$ and calculate the observed value:
$$O_{ki}=X_{k1}\times Y_{i1}\oplus X_{k2}\times Y_{i2}\oplus\ldots\oplus X_{kn}\times Y_{in}$$
$$=\sum_{j=1}^{n} X_{kj}\times Y_{ij} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$
3. Compare the observed value O_{ki} with key value K_i which represents the known output value of the cryptosystem, by using mean absolute error (MAE) s.t.
$$MAE_k = \frac{1}{n}\sum_{i=1}^{n}\left|O_{ki} - K_i\right| \quad \ldots\ldots\ldots\ldots\ldots\ldots (11)$$
4. The Fitness value is
$$Fitness_k = 1 - MAE_k$$
$$= 1 - \frac{1}{n}\sum_{i=1}^{n}\left|O_{ki} - K_i\right| \quad \ldots\ldots\ldots\ldots (12)$$

where

n: The size of the bee string or equation string.

X_{kj}: is the initial value j in String X_k.

Y_{ij}: is the equation variable j in the string Y_i.

O_{ki}: is the observed value i of string X_k calculated from Equation (10).

K_i: is the key bit (actual value) i.

When the measured (observed) value O_{ki} matches the key bit K_i, for all $1 \leq i \leq n$, then the summation terms MAE_k in Equation (11) evaluate to 0 so the fitness value is 1. The fact that a fitness value of 0 is never achieved does not affect the algorithm since high fitness values are more important than low fitness values. As a result, the search process is always moving towards fitness values closer to or equal 1. The steps of the ***Fitness Algorithm*** are as follows:

Fitness Algorithm (FA)

```
INPUT     : read X vector; {Initial string with size n from Population}
            read Y vector; {Equation string from data base file}
            read K vector ;{ Actual key=absolute value of SoE}
OUTPUT    : Fitness value;
PROCESS   : for i = 1 : n
$O_i = \sum_{j=1}^{n} X_j \times Y_j$ ; {XOR sum, $O_i$ is observed key}
$Z_i = |O_i-K_i|$;
  end;
 $MAE = \frac{1}{n}\sum_{i=1}^{n} Z_i$ ;{ MAE is the Mean Absolute Error}
        Fitness = 1-MAE;
END.
```

6.5 Evolution Process

In this part we attempt to make evolution to the population of Bees to improve the fitness values by modified the bees with good fitness in the population. The main steps of the ***Evolution Population Algorithm*** are as follows:

Evolution Population Algorithm (EPA)

```
INPUT     : read Bees Population;
OUTPUT    : New Population;
PROCESS   : sort (population)
{by descanting fitness order}
 N=[1 e;e+1 m]; V=[nep nsp];
 h=1,2;
         for i = N(h,1) : N(h,2)
       for j = 1:V(h)
```

```
                    Tmp=random(Bee);
          Fit=CALL FA;
          if Pop.fit(i) < Fit
                         Pop(i) = Tmp;
                         Pop.fit(i)=Fit;
                    end;
               end;
          end;
CALL IPA (m+1:size of population);
END;
```

6.6 Constructing a SoE for Geffe Cryptosystem

Two stopping criterions are be used to stop the BA cryptanalysis system, first criterion, some hundred generations are enough to reach this level of fitness. The second, when the fitness value reaches (1.0), so no need to reach the high number of generation. The algorithm was fast enough that this took less than few minutes.

Let's have 3-SR_{L_j} (since Geffe system has 3 SR's) with length L_j, j = 1,2,3, with following feedback vector:

$$C_{0j}=\begin{pmatrix} c_{01j} \\ c_{02j} \\ c_{03j} \end{pmatrix}$$

and has unknown initial value vector $A_{0j} = (a_{-1j},\ldots,a_{-Lj})$, so SR_{L_j} has $A_j = (C_{0j}| I_{L_j\times L_j-1})$

By using recurrence Equation (4),

$C_{ij} = A_j\times C_{i-1,j}$, i = 1,2, ... (13)

by using equation (5):

$B_{0j}\times C_{ij} = s_{ij}$, i = 0,1,…,L−1

and S_j=(s_{0j},s_{1j},…,$s_{n-1,j}$).

S_j represents the result output vector of SR_{L_j}, which of course, is unknown too. n represents the number of variables generated the LFSR's with consider to CF, which is also the number of equations which are be needed to solve the SoE. Of course, there are 3-SoE's (one SoE for each SR_{L_j} with unknown absolute values).

Now, let B_0 be the extended vector for n variables, which consists of initial values from all LFSR's and Ψ is the matrix of all C_i vectors considering the CF, C_i represents the extended vector of all feedback vectors C_{ij}, then $B_0\times\Psi = S$.

Now we apply this construction process for Geffe Cryptosystem, using Equations (4), (5) and (13).

The CF of this generator is [5]:

$F(x_1,x_2,x_3) = x_1x_2 \oplus x_2x_3 \oplus x_3$ .... (14)

for this reason $n=r_1r_2+r_2r_3+r_3$.

The initial value of this cryptosystem is:

$B_0 = B_{01}B_{02}+B_{02}B_{03}+B_{03} = (\delta_0,\delta_1,\ldots,\delta_{n-1})$,

(+ is concatenation to the vectors), such that:

$\delta_0 = \alpha_{-11}\alpha_{-12}$, $\delta_1=\alpha_{-11}\alpha_{-22},\ldots,\delta_{n-1}=\alpha_{r_3 3}$.

In the same way, Equation (14) can be applied on the feedback vector C_{ij}:

$C_i = C_{i1}C_{i2}+C_{i2}C_{i3}+C_{i3}$.

And the sequence S will be:

$S = S_1S_2+S_2S_3+S_3$

such that

$s_i = s_{i1}s_{i2} \oplus s_{i2}s_{i3} \oplus s_{i3}$,

s_i is the element i of S.

So the SoE which be changed to SoE can be gotten by equation (7).

Figure (2) shows the sequence S which is generated from Geffe Generator.

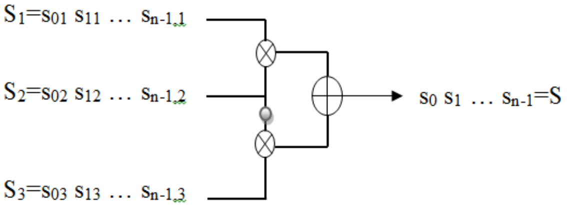
$S_1=s_{01}\ s_{11}\ \ldots\ s_{n-1,1}$
$S_2=s_{02}\ s_{12}\ \ldots\ s_{n-1,2}$
$S_3=s_{03}\ s_{13}\ \ldots\ s_{n-1,3}$
$s_0\ s_1\ \ldots\ s_{n-1}=S$

Fig.(2) The sequence S generated from Geffe Cryptosystem.

Example (1)

Let the Geffe system has 3-LFSR's with length 2,3 and 4, with following connection vectors:

$C_{01}=\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $C_{02}=\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ and $C_{03}=\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$, then n=22.

Suppose we have the output sequence:

S=(1,0,1,1,0,1,1,1,1,1,0,1,1,0,1,0,0,1,1,0,0,1)

$C_{01}=C_{31}=C_{61}=C_{91}=C_{12,1}=C_{15,1}=C_{18,1}=C_{21,1}=\begin{pmatrix} 1 \\ 1 \end{pmatrix}$,

$C_{11}=C_{41}=C_{71}=C_{10,1}=C_{13,1}=C_{16,1}=C_{19,1}=\begin{pmatrix} 0 \\ 1 \end{pmatrix}$,

$C_{21}=C_{51}=C_{81}=C_{11,1}=C_{14,1}=C_{17,1}=C_{20,1}=\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

$$C_{02}=C_{72}=C_{14,2}=C_{21,2}=\begin{pmatrix}1\\0\\1\end{pmatrix},\ C_{12}=C_{82}=C_{15,2}=\begin{pmatrix}1\\1\\1\end{pmatrix},\ C_{22}=C_{92}=C_{16,2}=\begin{pmatrix}0\\1\\1\end{pmatrix},\ C_{32}=C_{10,2}=C_{17,2}=\begin{pmatrix}1\\1\\0\end{pmatrix},C_{42}=C_{11,2}=C_{18,2}=\begin{pmatrix}0\\0\\1\end{pmatrix},C_{52}=C_{12,2}=C_{19,2}=\begin{pmatrix}0\\1\\0\end{pmatrix},C_{62}=C_{13,2}=C_{20,2}=\begin{pmatrix}1\\0\\0\end{pmatrix}.C_{03}=C_{15,3}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

$$,C_{13}=C_{16,3}=\begin{pmatrix}1\\0\\1\\1\end{pmatrix},C_{23}=C_{17,3}=\begin{pmatrix}1\\1\\1\\1\end{pmatrix},C_{33}=C_{18,3}=\begin{pmatrix}0\\1\\1\\1\end{pmatrix}$$

$$C_{43}=C_{19,3}=\begin{pmatrix}1\\1\\1\\0\end{pmatrix},C_{53}=C_{20,3}=\begin{pmatrix}0\\1\\0\\1\end{pmatrix},\ C_{63}=C_{21,3}=\begin{pmatrix}1\\0\\1\\0\end{pmatrix},\ C_{73}=\begin{pmatrix}1\\1\\0\\1\end{pmatrix},\ C_{83}=\begin{pmatrix}0\\0\\1\\1\end{pmatrix},\ C_{93}=\begin{pmatrix}0\\1\\1\\0\end{pmatrix},\ C_{10,3}=\begin{pmatrix}1\\1\\0\\0\end{pmatrix},\ C_{11,3}=\begin{pmatrix}0\\0\\0\\1\end{pmatrix},\ C_{12,3}=\begin{pmatrix}0\\0\\1\\0\end{pmatrix},\ C_{13,3}=\begin{pmatrix}0\\1\\0\\0\end{pmatrix},\ C_{14,3}=\begin{pmatrix}1\\0\\0\\0\end{pmatrix}.$$

by applying equation (5), C_0^T will be:
C_0^T=(1,0,1,1,0,1,1,0,0,1,0,0,0,0,1,0,0,1,1,0,0,1).
Therefore,

$$B=\begin{bmatrix}1&0&1&1&0&1&1&0&0&1&0&0&0&0&1&0&0&1&1&0&0&1&1\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\ 0&0&0&0&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\quad\text{..................................}(15)$$

7. Design of BA Cryptanalysis System

The BA cryptanalysis system can be view as two main parts, first, is the SoE constructing which described SoE constructing algorithm which is as follows:

SoE Constructing Algorithm (SoECA)

```
INPUT          : read $C_0^T$ vector, $B_0$;
{ initial value of LFSR }
           Calculate A = $(C_0|I_{L\times L-1})$;
           read $B_0$ vector;
{ Initial values of LFSR}
OUTPUT: Augmented matrix B;
                                   {store in a file}
PROCESS: for i = 0 : n-1
                            $C_i = A\times C_{i-1}$;
                            $s_i = B_0\times C_i$;
              end;
              S = $(s_0, s_1,\ldots,s_{n-1})$;
              $\Psi$= $(C_0, C_1,\ldots, C_{n-1})$;
               B  = $[\Psi^T|S^T]$;
END.
```

The second part is the BA cryptanalysis part, which illustrated in BA cryptanalysis algorithm, as shown below:

BA Cryptanalysis Algorithm

```
INPUT          : read Equation Information File of LFSRs;
                 read Actual Key of LFSR;
                 read G; {Number of Generations }
INITILIZE   : CALL IPA;
OUTPUT      :Best Bee which has highest Best.Fit;
PROCESS     : CALL SoECA;
     for i =1 : G
                    CALL FA; { Fitness }
                    CALL EPA; { Evolution }
       if  Best.Fit = 1.0   STOP
     end;
END.
```

8. Experimental Results

Geffe Cryptosystem

Three LFSR's are used, each has the following information:

1. First LFSR with length 3, has $1+x+x^3$ as characteristic polynomial with initial key values 001.
2. Second LFSR with length 5, has $1+x^2+x^5$ as characteristic polynomial with initial key values 00001.
3. Third LFSR with length 7, has $1+x+x^7$ as characteristic polynomial with initial key values 0000001.

When the SoE constructeed, we get 57 (=3×5+5×7+7) equations with 57 variables of SoE for the Geffe cryptosystem are shown in Table (1).

Table (1)
The 1st (7) equations of SoE for Geffe cryptosystem.

i	Separating equations	Binary. Representation of real equations
1	$a_1+a_3=1, b_2+b_5=1, c_1+c_7=1$ (101)(01001)(1000001)	(0100100000010010000000100000100000000000000010000011000001)-1
2	$a_1+a_2+a_3=1, b_1+b_4=0,$ $c_1+c_6+c_7=0$ (111)(10010)(1000011)	(1001010010100101000011000000000000000100001100000001000011)-1
3	$a_2+a_3=1, b_2+b_3+b_4=1,$ $c_1+c_5+c_6+c_7=1$ (011)(01110)(1000111)	(0000001110011100000000010001111000111100011100000001000111)-1
4	$a_1+a_2=0, b_1+b_2+b_4=0,$ $c_1+c_4+c_5+c_6+c_7=0$ (110)(11010)(1001111)	(1101011010000001001111100111100000001001111000000001001111)-0
5	$a_3=1, b_1+b_2+b_3+b_5=1,$ $c_1+c_3+c_4+c_5+c_6+c_7=1$ (001)(11101)(1011111)	(0000000001110110111111011111101111100000001011111011111)-1
6	$a_2 =0, b_1+b_4+b_5=1,$ $c_1+c_2+c_3+c_4+c_5+c_6+c_7=1$ (010)(10011)(1111111)	(0000010011000001111111000000000000000111111111111111111111)-1
7	$a_1=0, b_2+b_3+b_5==1,$ $c_2+c_3+c_4+c_5+c_6+c_7=1$ (100)(01101)(0111111)	(0110100000000000000000001111110111111000000001111110111111)-1
⋮	⋮	⋮

For this example, only 10 initial keys were in the population. The system began by generating 10 random initial key as shown in Table (2).

Table (2)
10 random initial keys in the Bees population.

Key	Random Initial Individuals												Fitness
1	-0.81	0.30	0.66	0.48	-0.69	-0.02	0.72	-0.47	0.85	-0.85	0.74	-0.38	0.5263
	0	1	1	1	0	0	1	0	1	0	1	0	
2	0.22	0.20	0.24	0.80	0.81	0.03	0.03	0.34	0.55	-0.94	0.50	0.80	0.5439
	1	1	1	1	1	1	1	1	1	0	1	1	
3	0.78	-0.25	0.78	0.69	0.23	0.66	0.71	-0.67	-0.71	-0.33	-0.27	0.63	0.6316
	1	0	1	1	1	1	1	0	0	0	0	1	
4	0.04	-0.20	-0.94	-0.03	-0.20	0.03	-0.10	-0.89	-0.15	-0.73	-0.88	-0.23	0.6491
	1	0	0	0	0	1	0	0	0	0	0	0	
5	-0.88	-0.14	0.30	-0.28	0.20	-0.81	-0.51	-0.19	-0.47	-0.75	-0.78	0.54	0.5614
	0	0	1	0	1	0	0	0	0	0	0	1	
6	-0.23	0.57	-0.68	0.78	0.56	-0.93	0.18	-0.85	0.02	0.28	0.08	-0.24	0.3860
	0	1	0	1	1	0	1	0	1	1	1	0	
7	-0.73	-0.15	-0.93	0.35	0.84	-0.22	0.06	-0.12	-0.49	-0.99	0.66	-0.12	0.4035
	0	0	0	1	1	0	1	0	0	0	1	0	
8	-0.33	-0.62	-0.34	-0.56	-0.48	0.85	0.70	0.40	0.59	-0.49	0.70	0.11	0.4737
	0	0	0	0	0	1	1	1	1	0	1	1	
9	-0.01	0.12	0.97	-0.97	0.73	0.32	0.63	0.54	0.06	-0.22	0.53	-0.63	0.5789
	0	1	1	0	1	1	1	1	1	0	1	0	
10	0.50	-0.61	0.32	0.95	-0.16	0.90	-0.04	-0.68	0.86	-0.12	-0.38	0.48	0.5088
	1	0	1	1	0	1	0	0	1	0	0	1	
Av.													0.5263

As we expected that none of the random keys are close to the actual key which is reflected in the fact that the average fitness for these keys is 0.5263. The best of these 10 random keys, (key4) has a fitness value 0.6491.

Table (3) shows the improvement in the results of finding the real initial key for Geffe cryptosystem.

Table (3)
Results for 1000 Generations for Geffe Cryptosystem.

Gen	*Best Fit.*	*Av.*	*Key no.*	*T/s*	*Best Initial Key*
1	0.6491	0.5263	4	0.33	000111011100
4	0.7895	0.5319	9	0.46	000000001010
22	0.8070	0.5491	12	1.22	000000101011
35	0.8596	0.5605	19	1.78	000000101010
124	0.9123	0.6495	20	5.47	000000001110
623	1.000	0.6918	11	24.66	000000000000

The best initial keys after (623) generations was: 001 for the 1st LFSR, 00001 for the 2nd LFSR and 0000001 for 3rd LFSR, which they are equal to the real initial keys. Fig.(3) shows the results developing of Table (3).

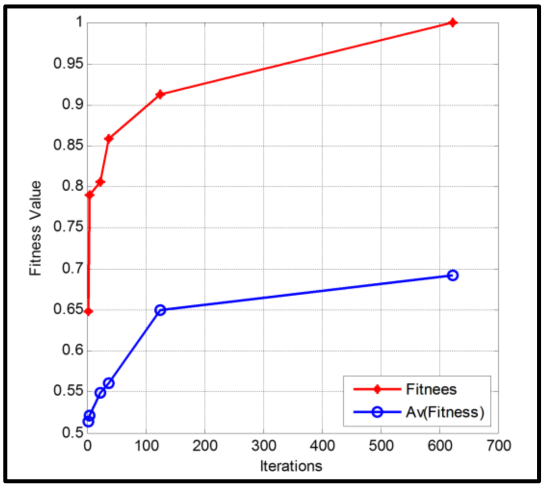

Fig.(3) The results improving of Table (3).

9. Conclusions

This research concludes the following aspects:

1. Although the proposed system is employed for sum of shift registers length (sum ≤ 12), it was provide the base of building BA cryptanalysis system valid for shift registers with high length attacking.
2. As a logical mathematical situation, if the proposed system gives a fitness value less than 1.0, this mean, no results obtained so we must run the system a gain, since the SoE must has unique solution for fixed absolute values, no another solution gives fitness equal 1.0.
3. Percentages reported are based on number of tests and different numbers of the tests must be always used, and that what will done in this research.

References

[1] Ekdhal P., “On LFSR Based Stream Ciphers Analysis and Design”, Ph.D. Thesis, Dept. of Economics, West Virginia University, Nov., 2003.

[2] Liu Y, Passino K. M., “Swarm Intelligence: Literature Overview”, Dept. of Electrical Engineering The Ohio State University March 30, 2000.

[3] Ali I. K., "Intelligent Cryptanalysis Tool Using Particle Swarm Optimization", Ph.D. Thesis, University of Technology, Department of Computer Science, 2009.

[4] Ahmed T., Laith A., Hashim K., “Attacking Transposition Cipher Using Improved Cuckoo Search”, Journal of Advanced Computer Science and Technology Research, Vol.4 No.1, PP.22-32, March 2014.

[5] Hameed, F. A., “Using Swarm Intelligence in Cryptanalysis of Nonlinear Stream Cipher Cryptosystem”, M.Sc., Department of Mathematics, College of Science, University of Baghdad, 2017.

[6] Yan, S. Y., “Number Theory for Computing”, Springer-Verlag Berlin, 2000.

[7] Schneier B., “Applied Cryptography”, John Wiley & Sons, 1997.

[8] Mohammed M. S, Mohammad G. S. Al-Safi, Faiaz H. A., “Dynamic Stream Ciphering Algorithm”, IOSR Journal of Computer Engineering (IOSR-JCE), V. 16, Issue 2, Ver. VIII, PP 72-78, (Mar-Apr. 2014) www.iosrjournals.org.

[9] Juntao G., Xuelian L., Yupu H., "Fault Attack on the Balanced Shrinking Generator"**;** Wuhan University Journal of Natural Science Vol.11 No.6 P.1773-1776, 2006.

[10] Golomb, S.W., “Shift Register Sequences” San Francisco: Holden Day 1967, (Reprinted by Aegean Park Press in 1982).

[11] Muhammad I., Rathiah H., Noor E. A., “An Overview of Particle Swarm Optimization Variants”; Procedia Engineering Vol. 53, PP. 491-496, 2013.

[12] Chong C., Low M. Yoke H., Sivakumar A., Gay K., “A Bee Colony Optimization Algorithm to Job Shop Scheduling”, Proceedings of the 2006 Winter Simulation Conference, N.J., USA. WSC, PP. 1954-196, 2006.

[13] Ashraf A., Michael P., Marco C., “Bees Algorithm”, Manufacturing Engineering Center, Cardiff University, Wales, UK, 2009.

[14] Pham D. T., Ghanbarzadeh A., Koc E., Otri S., Zaidi M. “The Bee's Algorithm – a Novel Tool for Complex Optimization Problems”. In: Pham D.T., Eldukhri E., Soroka A. J. ed(s) 2nd Virtual International Conference on Intelligence Production Machines and Systems. Elsevier, Oxford, pp 454-459, 2006.