Chapter 4
GENETIC ALGORITHM FOR CRYPTANALYSIS

4.1 Introduction

Genetic algorithms are developed based on the idea of emulating the evolution of a species. A population of individuals is generated, typically randomly. Each of these individuals represents a possible candidate solution to the problem. The solutions are encoded as bit strings (i.e., binary encoding). The solution quality of each individual is evaluated by a fitness function. The population of individuals consists of different keys considered for cryptanalysis and the fitness function is typically given. The natural evolution process is abstracted to three genetic operations; selection, crossover and mutation. In this step, the probability of an individual to be selected is directly proportional to its fitness value. After this, the second operator, crossover, is used to create a new child out of the two selected parents by breaking up the parents’ bit strings at a random position and mutually exchanging one bit string with the other. Similarly come to final and mutation is used where a unsystematically chosen bit in the string is flipped.

Details on genetic algorithms and their application to optimisation problems are extensively treated by Goldberg [12] and Srinivas et al. [44]. Recently biological exaptation has been used to solve problems of early convergence by increasing genetic diversity [10]. An algorithmic presentation [34] of genetic algorithm used for the study. Keys in cryptanalysis studies are represented as a string of bits in the chromosome and genetic operators process this bit string.

K. Matous at el. deals with carefully selected group of optimization problems is addressed to advocate application of genetic algorithms in various engineering optimization domains. Each topic introduced in the present paper serves as a representative of a larger class of interesting problems that arise frequently in many applications such as design tasks, functional optimization associated with various variation formulations, or a number of problems linked to image evaluation [104].
Genetic Algorithm is a programming technique who forms its basis from the biological evolution. [105] Genetic Algorithm is basically used as a problem solving strategy in order to provide with a optimal solution. They are the best way to solve the problem for which little is known. They will work well in any search space because they form a very general algorithm. The only thing to be known is what the particular situation is where the solution performs very well, and a genetic algorithm will generate a high quality solution. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem.[107]

- **Individual** - Any possible solution
- **Population** - Group of all *individuals*
- **Search Space** - All possible solutions to the problem
- **Chromosome** - Blueprint for an *individual*
- **Trait** - Possible aspect of an *individual*
- **Allele** - Possible settings for a *trait*
- **Locus** - The position of a *gene* on the *chromosome*
- **Genome** - Collection of all *chromosomes* for an *individual*

The input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called a *fitness function* that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random [107]

The GA then evaluates each candidate according to the fitness function activity; the candidate with good fitness has high chances to get selected than the one with average fitness. Various functions is basically used to test the fitness of any particular individual. These individuals with high fitness can be termed as promising candidates. These promising candidates are kept and allowed to reproduce. From them multiple copies are made, but the copies are not perfect; random changes are introduced during the copying process. These digital offspring then go on to the next generation, forming a new pool of candidate solutions, and are subjected to a second round of fitness evaluation. Those candidate solutions which were worsened, or made
no better, by the changes to their code are again deleted; but again, purely by chance, the random variations introduced into the population may have improved some individuals, making them into better, more complete or more efficient solutions to the problem at hand. Again these winning individuals are selected and copied over into the next generation with random changes, and the process repeats. The expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be discovered [107].

Genetic algorithms [108] have proven to be an enormously powerful and successful problem-solving strategy. Genetic algorithms have been used in a wide variety of fields to find solutions to problems that are more difficult than those faced by human designers. Thus, the solutions they come up with are often more efficient, more elegant, or more complex than anything comparable a human engineer would produce.

**Selection:** There are many different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation [109].

*Elitist selection:* The fittest members of each generation are guaranteed to be selected.

*Fitness-proportionate selection:* More fit individuals are more likely, but not certain, to be selected.

*Roulette-wheel selection:* A form of fitness-proportionate selection in which the chance of an individual's being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness.

*Scaling selection:* As the average fitness of the population increases, the strength of the selective pressure also increases and the fitness function becomes more discriminating. This method can be helpful in making the best selection later on when all individuals have relatively high fitness and only small differences in fitness distinguish one from another.

*Tournament selection:* Subgroups of individuals are chosen from the larger population, and members of each subgroup compete against each other. Only one individual from each subgroup is chosen to reproduce.
Reproduction: during the reproduction phase the next generation is created using the two basic methods, crossover and mutation. For every new child a pair of parents is selected from which the child inherits its properties. In the crossover process genotype is taken from both parents and combined to create a new child. With a certain probability the child is further exposed to some mutation, which consists of modifying certain genes. This helps to further explore the solution space and ensure, or preserve, genetic diversity. The occurrence of mutation is generally associated with low probability. A proper balance between genetic quality and diversity is therefore required within the population in order to support efficient search.

Crossover: Once the individuals have been selected the next thing is to produce the offspring. [109] The most common solution for this is something called crossover, and while there are many different kinds of crossover, the most common type is single point crossover. In single point crossover, chooses a locus at which you swap the remaining alleles from one parent to the other. This is complex and is best understood visually. Figure 4.1 shows the structure of a simple GA. The children take one section of the chromosome from each parent. The point at which the chromosome is broken depends on the randomly selected crossover point. This particular method is called single point crossover because only one crossover point exists. Sometimes only child 1 or child 2 is created, but oftentimes both offspring are created and put into the new population. Crossover does not always occur, however. Sometimes, based on a set probability, no crossover occurs and the parents are copied directly to the new population. The probability of crossover occurring is usually 60% to 70%. [109]
Mutation: After selection and crossover, we get new population full of individuals. Some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not all exactly the same, you allow for a small chance of mutation. You loop through all the alleles of all the individuals, and if that allele is selected for mutation, you can either change it by a small amount or replace it with a new value. [110] The probability of mutation is usually between 1 and 2 tenths of a percent. A visual for mutation is shown below.

Mutation is fairly simple. We just change the selected alleles based on what you feel is necessary and move on. Mutation is, however, vital to ensuring genetic diversity within the population [107].

Fitness Function: During the reproduction phase, each individual is assigned a fitness value derived from its raw performance measure given by the objective function. This value is used in the selection to bias towards more fit individuals. Highly fit individuals, relative to the whole population, have a high probability of being selected for mating whereas less fit individuals have a correspondingly low probability of being selected. The error rate is measured in each round of cross validation by dividing “the total number of misclassified examples” into “total number of test examples”.

Issues Related to the Genetic Algorithm [107]

- Certain optimization problems (they are called variant problems) cannot be solved by means of genetic algorithms. This occurs due to poorly known fitness functions which generate bad chromosome blocks in spite of the fact that only good chromosome blocks cross-over.
- There is no absolute assurance that a genetic algorithm will find a global optimum. It happens very often when the populations have a lot of subjects.
- In case of Clinical decision support system GA faces lack of transparency that is used for the decision support systems making it undesirable for physicians. The main challenge in using genetic algorithms is in defining the fitness
criteria. In order to use a genetic algorithm, there must be many components such as multiple drugs, symptoms, treatment therapy and so on must be available in order to solve a problem.

- In case of research study which investigated the possibility of automating parameter selection for an EEG-based P300-driven Brain-Computer Interface (BCI) Genetic Algorithm required lengthy execution times which render it infeasible for online utilization. The GA method was subsequently replaced by the less execution time-intensive N-fold cross-validation (NFCV) for the meta-optimization of feature extraction and pre-processing parameters using Fisher’s Linear Discriminant Analysis (FLDA).

- GA also faces Scalability Issues of Exchanging Building Blocks.

- Like other artificial intelligence techniques, the genetic algorithm cannot assure constant optimization response times. Even more, the difference between the shortest and the longest optimization response time is much larger than with conventional gradient methods. This unfortunate genetic algorithm property limits the genetic algorithms' use in real time applications.

- Genetic algorithm applications in controls which are performed in real time are limited because of random solutions and convergence, in other words this means that the entire population is improving, but this could not be said for an individual within this population. Therefore, it is unreasonable to use genetic algorithms for on-line controls in real systems without testing them first on a simulation model.

- The macromolecular function often relies on the comparison of different structural models of a molecule. In such a comparative analysis, the identification of the part of the molecule that is conformationally invariant with respect to a set of conformers is a critical step, as the corresponding subset of atoms constitutes the reference for subsequent analysis for example by least-squares superposition. A method is presented that categorizes atoms in a molecule as either conformationally invariant by automatic analysis of an ensemble of conformers (e.g. crystal structures from different crystal forms or molecules related by non-crystallographic symmetry).
The algorithm has been tested on several well-known examples and has been found to converge rapidly to reasonable results using a standard set of parameters. In addition to the description of the algorithm, a criterion is suggested for testing the identity of two three-dimensional models within experimental error without any explicit superposition [111].

To apply Genetic Programming to produce many different combinations of features, to extract new features and improve prediction accuracy. To use Evolutionary Algorithms to classify the students and problems directly as well. To apply Evolutionary Algorithms to find Association Rules and Dependency among the groups of problems (Mathematical, Optional Response, Numerical, Java Applet, and so forth) of LON-CAPA homework data sets [111].

Abdul Kadar Muhammad Masum at el. have find a solution of Vehicle Routing Problem using genetic algorithms. The Vehicle Routing Problem (VRP) is a complex combinatorial optimization problem that belongs to the NP-complete class. Due to the nature of the problem it is not possible to use exact methods for large instances of the
VRP. Genetic algorithms provide a search technique used in computing to find true or approximate solution to optimization and search problems. Observed that Heuristic used in addition during crossover or mutation for tuning the system[112].

Masoud Nosrati deals with usage of genetic algorithms in steganography. Steganography media which is subjected is image files. So, some of recent studies that introduce the steganography methods. Robustness and capacity are two important factors that are considered in almost all steganography methods. Also, different methods use one or both of stego-phases[113].

Currently the security of digital images attracts much attention, especially when these digital images are stored in memory or send through the communication networks. Many different image encryption methods have been proposed to keep the security of these images. Image encryption technique tries to convert an image to another image that is hard to understand. Genetic algorithms (GAs) are a class of optimization algorithms. Many problems can be solved using genetic algorithms through modelling a simplified version of genetic processes. The method based on Genetic Algorithm (GA) which is used to produce a new encryption method by exploitation the powerful features of the Crossover and Mutation operations of (GA)[114].

Asha Gowda Karegowda have dealt with Neural Networks are one of many data mining analytical tools that can be utilized to make predictions for medical data. Model selection for a neural network entails various factors such as selection of the optimal number of hidden nodes, selection of the relevant input variables and selection of optimal connection weights[115]. This paper presents the application of hybrid model that integrates Genetic Algorithm and Back Propagation network(BPN) where GA is used to initialize and optimize the connection weights of BPN. Significant features identified by using two methods, Decision tree and GA-CFS method are used as input to the hybrid model to diagnose diabetes mellitus. The results prove that, GA-optimized BPN approach has outperformed the BPN approach without GA optimization. In addition the hybrid GA-BPN with relevant inputs lead to further improvised categorization accuracy compared to results produced by GA-BPN alone with some redundant inputs.
4.2 Genetic Algorithm steps for cryptanalysis

**Step 1:** Input: Ciphertext of Intercepted, and statistics of the language.

**Step 2:** Initialise the algorithm parameters: the solution pool size M and the maximum number of iterations MAX.

**Step 3:** Randomly generate an initial pool of results PCURR, and then calculate the cost of each of the solutions in the pool.

**Step 4:** For I = 1… MAX do:
- Select M/2 pairs of keys from PCURR to be the parents of the new generation.
- Perform the mating operation on each of the pairs of parents to produce a new pool of solutions PNEW.
- For each of the M children, perform a mutation operation.
- Calculate the cost associated with each of the keys in the new solution pool PNEW.
- Sort PNEW from the most suitable (the least cost) to the least suitable (the most cost).
- Merge PCURR with PNEW to give a list of sorted solutions (discard duplicates). Choose the best M keys to become the new current pool PCURR.

**Step 5:** Output the best solution from PCURR.

The mutation operation in the algorithm is identical to the solution perturbation method used in simulated annealing algorithm discussed earlier. That is, randomly select two elements in the child and swap those elements.

4.3 Experimental Setup and Results

Experiments is carried out to outline the effectiveness of Genetic Algorithm. The Algorithm (GA) is coded using MATLAB 7 module have the Table 4.1 shows results for amounts of cipher text ranging from 100 to 1000 characters. Each size there are some keys which have been broken fully, if the cipher text has more size, the breakable key and success rates will be more accordingly, the time of cryptanalysis increases. The graphical representation for the values of the same is shown in Figure 4.2 illustrates the time comparison of the GA for different cipher text length (100, 200...
and 1000) to attack transposition cipher. This table clearly shows that with increasing the amount of cipher text, the running time of algorithm increases.

<table>
<thead>
<tr>
<th>Amount of Ciphertext (Character)</th>
<th>Genetic Algorithm</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (Minute)</td>
<td>Number of bits matched in the Key</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>30</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>44.3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>22.9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>28.1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>34.5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>65.9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>12.7</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>11.3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>900</td>
<td>70.2</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>38.4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Genetic Algorithm Results
4.4 Concluding Remarks

GA has been successfully applied to a wide range of real-world problems of significant complexity. The population size is taken dynamic and is assumed the square of estimated key length. Mutation operator is randomly performed on 9% of the new generation. The fitness weights used in fitness function GA were 0.01 for unigrams, 0.9 bigrams and 0.611 for trigrams. Experimental results indicated that this improved fitness function is extremely powerful technique for an attack on transposition cipher. Amount of recovered key in transposition cipher using this algorithm is more than the other techniques. With the increase of cipher text, the breakable key and the time of cryptanalysis will increase. The performance of the algorithm is found to be better and considerably faster than exhaustive search method. Genetic Algorithm achieved the amount of Ciphertext is 1000, time taken 38.4 and bits matched 5.

The outcome of this research work is “Overview of Linear Cryptanalysis on S-DES and Block Ciphers using Hill Cipher Method”, Published at International Journal of Computer Applications, New York, USA, Vol. 63, No.21, February 2013, pp 47-52. (Impact factor:0.835).

Cryptanalysis process using GA is time consuming hence the next method concentrates on Simulated Annealing for further research.