Cryptanalysis of Simple Substitution Ciphers Using Bees Algorithm

Dr. Ismail K. Ali
Alma'mon University College
Abstract

This paper describes a new application of the Bees Algorithm to the optimization to break the key for the problem of cryptanalysis a simple substitution ciphers. The algorithm, which is a swarm-based algorithm inspired by the food foraging behavior of honey bees, was also employed to select the best combination of parameters and key representation to maximize key accuracy. The paper presents the results obtained to demonstrate the strengths of the Bees Algorithm as a cryptanalysis tool.

Keywords: Bees Algorithm, substitution cipher, Cryptanalysis.

1. Introduction

cryptanalysis can be described as the process of searching for flaws or oversights in the design of cryptosystems. A typical cipher takes a clear text message (known as the plaintext) and some secret keying data (known as the key) as its input and produces a scrambled (or encrypted) version of the original message (known as the ciphertext).

Classical ciphers were first used hundreds of years ago. As far as security is concerned, they are no match for today’s ciphers; however, this does not mean that they are any less important to the field of cryptology. Their importance stems from the fact that most of the ciphers in common use today utilize the operations of the classical ciphers as their building blocks. For example, the Data Encryption Standard (DES) [1], an encryption algorithm used widely in the finance community throughout the world, uses only three very simple operators, namely substitution, permutation (transposition) and bit-wise exclusive-or (admittedly, in a complicated fashion). Given their simplicity, and the fact that they are used to construct other ciphers, the classical ciphers are usually the first ones considered when researching new attack techniques such as the ones discussed here.

The use of automated techniques in the cryptanalysis of cryptosystems is desirable as it removes the need for time-consuming (human) interaction with a search process. Making use of computing technology also allows the inclusion of complex analysis techniques, which can quickly be applied to a large number of potential solutions in order to weed out unworthy candidates.

Two fundamental goals in computer science are finding algorithms with provably good run times and with provably good or optimal solution quality. A heuristic is an algorithm that gives up one or both of these goals. This paper attempted to use Bees Algorithm (BA) in the cryptanalysis of simple substitution cipher. The rest of the paper is organized as follows. The underlying principles and algorithm of Bees has shown in Section 2 and 3. Section 4 presents a brief overview of the simple substitution cipher. A result of computational tests to evaluate the performance of algorithm is reported in section 6.

2. Bees in Nature

Bees Algorithm is an optimization algorithm proposed by Pham et al [2] based on the food foraging behavior of honeybees in nature. The Bees Algorithm has become one of the most successful optimization algorithms due to its successful implementation in various applications for optimization problems such as neural network training [3], identifying homogenous data clustering [4] and a preliminary design problem [5].
A colony of honey bees can extend itself over long distances (more than 10 km) and in multiple directions simultaneously to exploit a large number of food sources [6, 7]. The foraging process of bees begins in a colony by scout bees being sent to search for promising flower patches. Flower patches with large amounts of nectar or pollen that can be collected with less effort tend to be visited by more bees, whereas patches with less nectar or pollen receive fewer bees. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees [7]. When they return to the hive, those scout bees that have found a patch rated above a certain quality threshold deposit their nectar or pollen and go to the “dance floor” to perform a dance known as the "waggle dance" [6]. This mysterious dance is essential for colony communication, and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness) [6, 8]. This information helps the colony to send its bees to flower patches precisely, without using guides or maps. Each individual’s knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it [8]. After waggle dancing on the dance floor, the dancer (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. More follower bees are sent to more promising patches. This allows the colony to gather food quickly and efficiently.

While harvesting from a patch, the bees monitor its food level. This is necessary so that they can decide upon the next waggle dance when they return to the hive [8]. If the patch is still good enough as a food source, then it will be advertised in the waggle dance and more bees will be recruited to that source.

3. The Bees Algorithm

In bees algorithm, the colony of artificial bees contains two groups of bees are scout and employed bees. The scout bees have the responsibility is to find a new food source. The responsibility of employed bees is to determine a food source within the neighborhood of the food source in their memory and share their information with other bees within the hive.

The basic steps of the Bees Algorithm are explained in Figure 1.

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
4. Select sites for neighborhood search.
5. Recruit bees for selected sites (more bees for best sites) and evaluate fitness's.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitness's.
8. End While.

Figure 1: Pseudo code of the basic bee's algorithm
Also, the algorithm requires a number of parameters to be set in Table 1.

**Table 1: Parameters selection of Bees Algorithm**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of scout bees</td>
<td>$n$</td>
</tr>
<tr>
<td>Number of sites selected out of $n$ visited sites</td>
<td>$m$</td>
</tr>
<tr>
<td>Number of best sites out of $m$ selected sites</td>
<td>$e$</td>
</tr>
<tr>
<td>Number of bees recruited for best $e$ sites</td>
<td>$nep$</td>
</tr>
<tr>
<td>Number of bees recruited for the other ($m-e$) selected sites</td>
<td>$nsp$</td>
</tr>
<tr>
<td>Initial size of patches, which includes site and its neighborhood</td>
<td>$ng'h$</td>
</tr>
</tbody>
</table>

### 4. Simple Substitution Cipher

The simple substitution cipher (sometimes referred to as the monoalphabetic substitution cipher to distinguish it from the polyalphabetic substitution cipher). Each symbol in the plaintext maps to a (usually different) symbol in the ciphertext. The processes of encryption and decryption are best described with an example, such as the one following.

For example, a simple substitution cipher key can be represented as a permutation of the plaintext alphabet. Using this representation the $i$th letter in the plaintext alphabet is encrypted to the $i$th element of the key. The string “THE MONEY IS IN THE BAG” is encrypted using the key in the Table 2.

**Table 2: Example substitution cipher**

<table>
<thead>
<tr>
<th>KEY</th>
<th>Plaintext</th>
<th>Ciphertext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext</td>
<td>ABCDEFGHIJKLMNOPQRSTUVWXYZ</td>
<td>PQOWIEURYTLAKSJDHFGMZNXBCV</td>
</tr>
<tr>
<td>ENCRYPTION</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Using the key representation in Table 2, the original message can be discovered by reversing the encryption procedure. That is, the ciphertext character at position $i$ in the key decrypts to the $i$th character in the plaintext alphabet. For an alphabet of 26 characters, there are $26!$ or greater than $4\times10^{26}$ possible keys for a simple substitution cipher. This number is far too large to allow a brute force attack even on the fastest of today’s computers. However, because of the properties of the simple substitution cipher they are relatively easy to cryptanalysis [9, 10].
5. Use BA in the Cryptanalysis of Simple of Substitution Ciphers

The general strategy with the substitution ciphers is to substitute symbols from the plaintext alphabet with different symbols from the ciphertext alphabet. The weakness with this strategy is that character frequency distributions are not significantly altered by the encryption process. Thus, most attacks on substitution ciphers attempt to match the character frequency statistics of the encrypted message with those of some known language. Character frequency statistics (or \(n\)-grams) indicate the frequency distribution of all possible instances of \(n\) adjacent characters (for example, THE is a very common 3-gram (or trigram) in the English language). The attack on the simple substitution cipher is particularly simple since the frequency of any \(n\)-gram in the plaintext (or unencrypted) message will correspond exactly to the frequency of the corresponding encrypted version in the ciphertext. The search for the corresponding \(n\)-gram frequencies can be automated using combinatorial optimization algorithms such as BA. Here; a BA is utilized in attacks in the simple substitution cipher.

5.1 Representation of a Candidate Solution

An artificial bee's environment is defined as an artificial representation of the space. Those artificial bees perform only activities defined by the model and a corresponding computer program. The way in which artificial bees communicate with each other is defined. The substitution ciphers are the new field of BA in discrete space, a direct correlation must be found between the bee vector and the solution representation of substitution cipher. The idea from BA attack of the simple substitution cipher is the permutation form; the position of a character in the vector represented permutation means the order or sequence of the key is scheduled. The main emphasis is placed on how to develop an alternative method for the key substitution by utilizing the features of BA. For the purpose of this study, a key for decoding a cipher is given by an ordered list of the 26 letters of the alphabet. As this representation indicates, the size of the key space is 26! or greater than 4.03291 \(\times 10^{26} \approx 2^{88}\) which suggests that a purely random search is not acceptable.

5.2 Initial Population

Like other evolutionary algorithms, the BA starts from an initial population. The bees are randomly generated between the minimum and the maximum operating limits of the generators. For the initialization process either use some heuristics different alphabetic strings, or initialize the population by a random sample of permutation of \{A, B,\ldots, Z\}.

5.3 Fitness Function Calculation

The fitness function is the main factor of the algorithm. The choice of fitness measure Depends entirely on the language characteristics must be known. The technique used by Clark[11] to compare candidate key is to compare \(n\)-gram statistics of the decrypted message with those of the language (which are assumed known). Equation 1 is a general formula used to determine the suitability of a proposed key, here \(K\) is known as language Statistics i.e. for English, [A,B,\ldots, Z], \(D\) is the decrypted
message statistics, and u/b/t are the unigram, bigram and trigram statistics. The values of $\alpha$, $\beta$ and $\gamma$ allow assigning of different weights to each of the three $n$-gram types where $\alpha + \beta + \gamma = 1$. The measured errors (i.e., the difference between standard frequencies and measured frequencies) are normalized and subtracted from 1, so that the number closer to 1 represents the higher fitness.

$$\text{Fitness} = 1 - \left( \alpha \sum_{i=1}^{26} |K^u[i] - D^u[i]| + \beta \sum_{i,j=1}^{26} |K^b[i,j] - D^b[i,j]| + \gamma \sum_{i,j,k=1}^{26} |K^t[i,j,k] - D^t[i,j,k]| \right)$$ (1)

In general, the larger the $n$-grams, the more accurate the assessment is likely to be. It is usually an expensive task to calculate the trigram statistics; this is, perhaps, it omitted in the above equation. The complexity of determining the fitness is $O(N^3)$ (where $N$ is the alphabet size) when trigram statistics are being determined, compared with $O(N^2)$ when bigrams are the largest statistics being used.

5.4 Parameters Selection

As the goal of this study is to verify the impact of the choice of social topologies in the behavior of the algorithm, the tuning parameters are fixed. They are set to the values that are widely used by the community and that are deemed the most appropriate ones. Table 3 shows the different parameters used.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of scout bees</td>
<td>$n$</td>
<td>10 – 20</td>
</tr>
<tr>
<td>Number of sites selected out of n visited sites</td>
<td>$m$</td>
<td>3 – 5</td>
</tr>
<tr>
<td>Number of best sites out of m selected sites</td>
<td>$e$</td>
<td>1</td>
</tr>
<tr>
<td>Number of bees recruited for best e sites</td>
<td>$nep$</td>
<td>7</td>
</tr>
<tr>
<td>Number of bees recruited for the other($m-e$) selected sites</td>
<td>$nsp$</td>
<td>2</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>$Iter$</td>
<td>300-1000</td>
</tr>
</tbody>
</table>

5.5 Proposal Bees Algorithm (BA) in the Cryptanalysis of Substitution Cipher

The following is an algorithmic description of the cryptanalysis on a simple substitution cipher using bee's algorithm (BA):

**Input:** The cipher text, the statistics of the language (unigrams, bigrams), the algorithm parameters ($n$, $m$, $e$, $nep$, $nsp$, $Iter$).

**Output:** The key having the highest fitness as found by BA.

**Step 1:** Randomly generate the initial bees (keys of the simple substitution cipher) to form a population.
Step 2: Calculate the fitness function of each of the bees (keys) using equation.

$$Fitness = 1 - (\alpha \sum_{i=1}^{26} |K^a[i] - D^a[i]| + \beta \sum_{i,j=1}^{26} |K^b[i,j] - D^b[i,j]|)$$

Step 3: Repeat

Step 4: Select sites for neighborhood search.

Step 5: Recruit bees for selected sites (more bees for best e sites) and evaluate fitness’s.

Step 6: Select the fittest bee from each patch.

Step 7: Assign remaining bees to search randomly and evaluate their fitness’s.

Step 8: Until stopping criterion is met.

Step 9: Copy the best key obtained so far in the output key variable and exit.

6. Experimental Results

Bees Algorithm (BA) was used in cryptanalysis of simple substitution ciphers. We summarize the results of these experiments in this section. The algorithm was applied to ciphertext created using a simple substitution cipher and the attack was run a number of times with a variety of parameter values. In general it was found that (300) iterations were usually enough to break the ciphertext and the algorithm was fast enough that this took a few seconds.

The experimental shown in Figure 2 was performed on simple substitution cipher using BA. As we can see, the fitness value has started from about 0.50 and come up to 0.87 in about 100 iterations. In addition, we can see a rapid increase in fitness function at the beginning and the rate of increases as we run the experiments for longer period. This is because as the fitness value increases, it becomes more and more difficult to find a better key. Figure 3 shows how the average (over 30 randomly selected keys) number of corrected key elements varies with the amount of known ciphertext.

![Figure 2: Performance of BA](image-url)
7. Conclusions and Future Work
The goal of this work is to illustrate the feasibility of using bee's algorithm as a cryptanalysis tool. Hence, our study is applied to simple substitution ciphers. The experimental results show how BA is successful in finding the key for a simple substitution ciphers and give a good performance. Bees Algorithm (BA) has demonstrated excellent promise as a heuristic technique in recent years. This technique when applied to cryptanalysis of our candidate cipher has yielded better performance in fields of time and/or memory requirements.

The research described in this document includes the BA to the cryptanalysis of a classical cryptosystems. The proposed algorithm is open avenues for further research in cryptanalysis other more complicated cryptosystems, such as polyalphabetic substitution cipher or modern ciphers such as public key cipher, stream cipher and block cipher. Another area for investigating and studying deep into the use of BA to construct strong Boolean functions in the field of cryptography.
References


