Automated Cryptanalysis of Transposition Cipher Using Bees Algorithm

Dr. Ismail K. Ali
Alma'mon University College

Abstract

Transposition ciphers are a class of classical encryption algorithms based on rearranging units of plaintext according to some fixed permutation, which acts as the secret key. In this paper, we propose a cryptanalysis tool approach based on a population-based algorithm inspired by the honeybees forage for food called Bees Algorithm to break a transposition cipher. Experimental results demonstrate the applicability of algorithm for the cryptanalysis of transposition ciphers.

Keywords: Cryptanalysis, Transposition Cipher, Bees Algorithm

1. Introduction

There are two basic types of encryption ciphers: substitution and transposition (permutation). The *substitution cipher* replaces bits, characters, or blocks of characters with different bits, characters, or blocks. The *transposition cipher* does not replace the original text with different text, but moves the original text around. It rearranges the bits, characters, or blocks of characters to hide the original meaning [1].

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) [2], 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). The whole point of cryptography is to keep the plaintext (or the key, or both) secret from eavesdroppers (also called attackers, interceptors, or simply the enemy). Cryptanalysis is the science of recovering the plaintext or the key. An attempted cryptanalysis is called an attack. The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme, thus permitting its subversion or evasion.

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; for example, it usually finds good solutions, but there is no proof the solutions could not get arbitrarily bad; or it usually runs reasonably quickly, but there is no argument that this will always be the case. Often, one can find specially crafted problem instances where the heuristic will in fact produce very bad results or run very slowly; however, these instances might never occur in practice because of their special structure. Therefore, the use of heuristics is very common in real world implementations [3].

Bees Algorithm (BA) is a population-based algorithm inspired by the honeybees forage for food; it has demonstrated excellent promise as a heuristic technique in recent

years. Here we have attempted to use BA in the cryptanalysis of transposition cipher. The rest of the paper is organized as follows. Section 2 presents a brief overview of the transposition cipher. The underlying principles of Bees Algorithm are presented in Section 3. Section 4 describes the algorithm proposed. Experimental results of computational tests to evaluate the performance of the algorithm are reported in section5.

2. Transposition Cipher

In cryptography, a transposition cipher is a process of encryption by which the positions held by units of plaintext are shifted according to a regular system or pattern, so that the ciphertext constitutes a permutation of the plaintext. That is, the order of the units is changed at the end of the shifting process. Mathematically, an objective function is used on the characters positions to encrypt and an inverse function to decrypt. The letters themselves are kept unchanged, which implies that the effect is only on their positions only, making their order within the message scrambled according to some well-defined scheme. In a columnar transposition, the message is written out in rows of a fixed length, and then read out again column-by-column, and the columns are chosen in some scrambled order. Both the width of the rows and the permutation of the columns are usually defined by a keyword [4]. Advanced forms of columnar encryption techniques are used for encryption in a matrix representation form [5]. Consider an example of a transposition cipher with a period of five, and a key {4, 2, 1, 5, 3}.
As an example, consider the plaintext:
"MYBEESALGORITHMISANEFFICIENT",
Encrypted using the key (42153):

Table 1: Example of the transposition cipher key and encryption process

M	Y	B	E	E		B	Y	E	M	E
S	A	L	G	O		L	A	O	S	G
R	I	T	H	M		T	I	M	R	H
I	S	A	N	E	⇨	A	S	E	I	N
F	F	I	C	I		I	F	I	F	C
E	N	T	X	X		T	N	X	E	X

Ciphertext: "BLTAIT YAISFN EOMEIX MSRIFE EGHNCX"
Decryption is simply a matter of writing the ciphertext back into the grid using the same ordering of the columns.

3. Bees Algorithm (BA)

The BA is classified as Bee Inspired Algorithms that fall in the field of Swarm Intelligence, and more broadly fall in the fields of Computational Intelligence and Metaheuristics. The bees algorithm is a population-based search algorithm inspired by the natural foraging behavior of honey bees first developed in 2005[6].

In its basic version, the algorithm starts by scout bees being placed randomly in the search space. Then the fitness's of the sites visited by the scout bees are evaluated and Bees that have the highest fitness's are chosen as “selected bees” and sites visited by them are chosen for neighborhood search. Then, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. Searches in the neighborhood of the best e sites are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm. The remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts, those that were the fittest representatives from a patch and those that have been sent out randomly. The algorithm performs a kind of neighborhood search combined with random search and can be used for both combinatorial and functional optimization [7].

The BA requires a number of parameters to be set, namely: the number of scout bees (*n*), the number of patches selected out of *n* visited points (*m*), the number of elite patches out of *m* selected patches (*e*), the number of bees recruited for the best e patches (*nep*), the number of bees recruited for the other (*m-e*) selected patches (*nsp*) and the size of patches (*ngh*) including termination criterion. The pseudo code for the bee's algorithm in its simplest form [7]:

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) // Forming new population
4. Select sites for neighborhood search.
5. Recruit bees for selected sites (more bees for best e 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.

4. BA for Cryptanalysis Transposition Ciphers

The focus of this work is to apply a BA to the problem of searching through the key space of a simple transposition cipher. The implementation part of our cryptanalysis involves a different phases that is will be described below:

4.1 Initial Population

Individuals in the population represent candidate solutions to the task assignment problem. Each solution is encoded as a vector of integers. For a problem with key of length *N*, the length of the vector which can be considered as a string of *N*. Moreover, the content of each cell that shows a value can get a number between 1 and *N* representing the key allocated to that problem. In order to generate an initial population with *n* individuals, a random number between 1 and *N* is assigned to each vectors.

4.2 Fitness Function Calculation

The frequency of appearance of some character of the alphabet within a given text is different from a language in the other one. In addition, it is also different in texts at the level of the same language as in the case of literary, political or commercial texts. The most ten common bigrams and trigrams (in order) of English language are shown in Table 2 given to them in fitness calculation [8]. Due to permuting the characters frequency of unigrams (single letters) doesn't change in the cipher text, makes no effect on the cipher. Quadgrams are computationally expensive so they aren't considered in fitness calculation. When candidate solution (key permutation) fitness is to be calculated, it should first decrypt the cipher text then compute the frequencies of bigrams and trigrams in the decrypted text and sum the scores.

Fitness function is evaluated based on the bi-grams (two letter words) frequency and trigrams (three letter words) frequency in the decrypted cipher text using equation 1.

$$\textbf{\textit{Fitness}} = \Big(\sum_{i=1}^{10} frq[B_i] + \sum_{j=1}^{10} frq[T_j] \Big) / \mathbf{L} \qquad (1)$$

Where $frq[B_i]$ denotes the frequency of appearance of the pairs of letters (*bigrams*), $frq[T_j]$ denotes the frequency of appearance of the triple of letters (*trigrams*), and L the text length.

Table 2: The most common bigrams and trigrams

English Bigrams		English Trigrams	
Letter	**Frequency**	**Letter**	**Frequency**
th	**0.03883**	**the**	**0.03508**
he	**0.03681**	**and**	**0.01593**
in	**0.02284**	**ing**	**0.01147**
er	**0.02178**	**her**	**0.00822**
an	**0.02141**	**hat**	**0.00651**
re	**0.01749**	**his**	**0.00597**
nd	**0.01572**	**tha**	**0.00594**
on	**0.01418**	**ion**	**0.00561**
en	**0.01383**	**for**	**0.00555**
at	**0.01336**	**ent**	**0.00531**

4.3 Proposal BA in the Cryptanalysis Transposition Cipher

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

Input: The cipher text and its length (L), the key size (permutation size or period), the score table such as the one in Table 2, and algorithm parameters (*n*, *m*, *e*, *nep*, *nsp*, *Max_Iter*).

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

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

$$\textbf{\textit{Fitness}} = \left(\sum_{i=1}^{10} frq[B_i] + \sum_{j=1}^{10} frq[T_j] \right) / \mathbf{L}$$

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.
Step 8: Until stopping criterion is met.
Step 9: Copy the best key obtained so far in the output key variable and exit.

4.4 BA Parameters Selection

The appropriate values of the BA parameters are obtained by examining different values of these parameters and making many trial and error runs. The algorithm tested includes a substantial number of settings where it would be difficult to treat them simultaneously. Table 3 shows the values of the parameters adopted for BA. The values were decided empirically.

Table 3: Parameters selection of cryptanalysis transposition cipher

Parameters	Symbol	Value
Number of scout bees	*n*	10 – 20
Number of sites selected out of n visited sites	*m*	3 – 5
Number of best sites out of m selected sites	*e*	1
Number of bees recruited for best e sites	*nep*	7
Number of bees recruited for the other(*m-e*) selected sites	*nsp*	2
Number of iterations	*Iter*	200-1000

5. Experimental Results

A Metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space. The number of possible rearrangements of a length *N* key is *N*! (*N* factorial). Notice that if the length key size is 20 then there are 20! = 2,452,902,008,176,640,000 arrangements of the columns. As with all the results for the classical ciphers, the first comparison is made based upon the amount of ciphertext provided to the attack. The algorithm was run on differing amounts of ciphertext. The results in Table 4 represent the average number of key letters correctly placed for a key size of 15 by using BA technique.

The BA solver returns a key candidate within score of fitness evaluations. Sometimes, the algorithm returns solutions that are close to the key. For example, two letters in the permutation might be in the incorrect order, or the permutation may start on the wrong column. Despite this, the plaintext remains mostly readable, especially if the period of the transposition cipher is large. To investigate this property, we use a success metric which shifting the number of columns which are placed before their correct successor.
Table 5 shows results for the transposition cipher based on the period. It should be noted that for period less than fifteen, the algorithm could successfully recover the key all the most of time. The table shows that the BA was the most powerful as a cryptanalysis tool.

Table 4: The amount of key recovered versus available ciphertext transposition size of 15

Amount of Ciphertext Length	Average Number of Key Recovered
100	8.25
150	9
250	10.50
400	11.50
600	12.60
800	12.75
1000	13

Table 5: The amount of key recovered versus transposition size using 1000 known ciphertext characters

Transposition Key Size	Average Number of Key Recovered
7	6.75
9	7.90
11	9
15	13
20	16.10
25	19

Figure 1 shows the performance of BA. Thus, BA is a very promising approach for solving the problem of cryptanalysis of simple transposition cipher and any discrete optimization problem in general. However, this is a fairly new technique and there is lot of scope for research in this area.

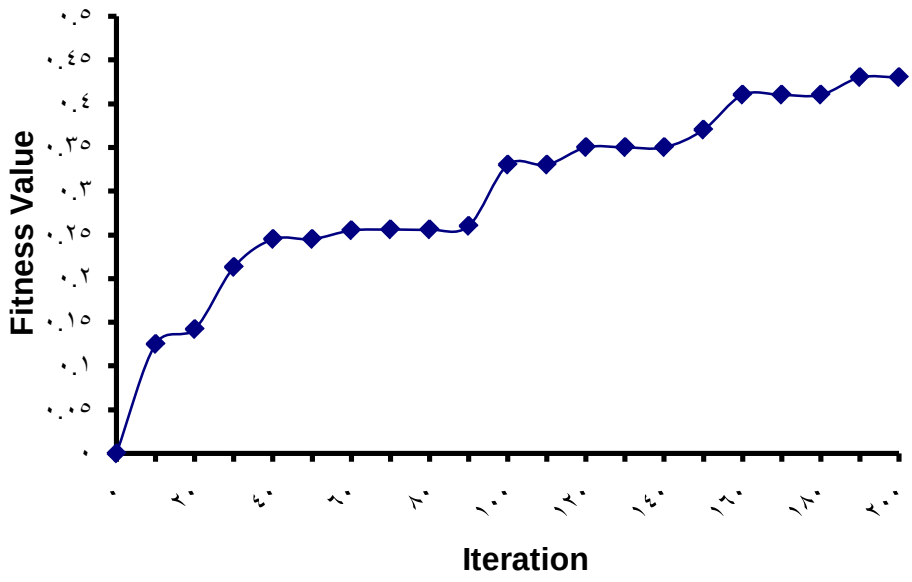

Figure 1: Performance of BA

6. Conclusion

This paper reviews works on cryptanalysis of classical ciphers using BA approach. The types of classical ciphers involved are the simple transposition cipher. In this paper, we have argued that bee's algorithm is a valuable tool in the cryptanalysis of certain classes of cipher. We believe this is the first application of a BA approach to cryptanalysis of transposition ciphers. Experimental results demonstrate BA has good performance, few parameters need to be tuned for the best possible performance, and advantage of easy implementation and it is very efficient in finding optimal solutions performance in fields of time and/or memory requirements. . The proposed algorithms are open avenues for further research in cryptanalysis other more complicated cryptosystems.

References

1. ***"A Classical Cipher, Transposition ciphers"***, retrieved from http://en.wikipedia.org/wiki/Classical_cipher. download in 10/1/2013.

2. U.S. Department of Commerce/National Bureau of Standards ***" Data Encryption Standard "***, Federal Information Processing Standards Publication 46-1, 1988.

3. Duran-Novoa, R., Leon-Rovira, N., Aguayo-Tellez, H., Said, D. ***"Inventive Problem Solving Based on Dialectical Negation Using Evolutionary Algorithms and TRIZ Heuristics"***. *Computers in Industry* 62(4), pp. 437-445, 2011.

4. ***"Transposition ciphers, columnar transposition"***, retrieved from http://en.wikipedia.org/wiki/Transposition_cipher. download in 10/1/2013.

5. Kester, Q. A.,***"A Public-Key Exchange Cryptographic Technique Using Matrix***", Adaptive Science & Technology (ICAST), 2012 IEEE 4th International Conference on, vol., no., pp.78-81, 25-27 Oct. 2012.

6. D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim and M. Zaidi, ***"The Bees Algorithm"***, Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.

7. Pham D.T., Ghanbarzadeh A., Koc E., Otri S., and Zaidi M., ***" The Bees Algorithm–A Novel Tool for Complex Optimization Problems"***, in second Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006), Elsevier. Cardiff, UK. pp. 454-459, 2006.

8. ***"Relative frequencies of letters"***, retrieved from http://www.cryptograms.org/letter-frequencies.php, download in11/10/2012.