Nature-inspired Cryptoanalysis Methods for Breaking Vigenère Cipher

Lucija Brezočnik, Iztok Fister Jr., and Vili Podgorelec

Abstract The protection of sensitive data against unauthorized access remains a primary concern of modern life. Over time, many different approaches have been introduced to tackle this problem, from substitution ciphers in classic cryptography to post-quantum cryptography as a representative of modern cryptography. In this paper, we focus on a polyalphabetic substitution cipher, precisely the Vigenère cipher. For a cryptoanalysis of the latter, we utilized five nature-inspired algorithms, i.e., Differential Evolution, Firefly Algorithm, Particle Swarm Optimization, Artificial Bee Colony Algorithm, and Cuckoo Search, were utilized. Furthermore, different key lengths were analysed to investigate the search behaviour of the selected algorithms. The results of the experiment show that the applicability of the nature- inspired algorithms for cryptoanalysis is very promising. Out of the tested algorithms, the Differential Evolution outperformed other algorithms.

Key words: Cryptoanalysis, Nature-inspired Algorithms, Swarm Intelligence, Vigenère Cipher

1 Introduction

The technique of securing information against third party adversaries is called cryptology. It encompasses cryptography (the art of creating encryption schemes) and

Lucija Brezočnik
Faculty of Electrical Engineering and Computer Science, University of Maribor,
Koroška cesta 46, 2000 Maribor, Slovenia
e-mail: lucija.brezocnik@um.si

Iztok Fister Jr.
e-mail: iztok.fister1@um.si
· Vili Podgorelec
e-mail: vili.podgorelec@um.si

cryptoanalysis (the art of analysing and breaking encryption schemes). In this paper, we will focus exclusively on classic cryptography.

There are two main types of classical ciphers, i.e. transposition ciphers and substitution ciphers [10]. The latter group is further divided into monoalphabetic (e.g. Caesar cipher, Affine cipher, and Atbash cipher), polyalphabetic (e.g. Vigenère cipher, Gronsfeld cipher, and Beaufort cipher), and polygraphic (e. g. Hill cipher) ciphers.

In this paper, we investigate the use of different nature-inspired algorithms in a cryptoanalysis of the Vigenère cipher. This field is relatively new and therefore has not been researched extensively. Although, some methods show promise. In most cases, researchers utilize a Genetic Algorithm (GA) [7–9] for automated cryptoanalysis. Out of the swarm intelligence algorithms, a Particle swarm optimization (PSO) [12] is used in the majority of the times, followed by the Cuckoo Search algorithm (CS) [3] and Bees algorithm (BA) [1].

The primary missions of this paper are:

- to present a short overview of cryptoanalysis methods which include nature-inspired algorithms,
- to analyse a universal cryptoanalysis method that can be combined with any nature-inspired algorithm, and
- to investigate the behaviour of different nature-inspired algorithms for breaking the Vigenère cipher.

The structure of this paper is as follows: Section 2 outlines the Vigenère cipher, while Section 3 acquaints the reader with the fundamentals of nature-inspired algorithms and briefly presents used ones. Section 4 describes proposed nature-inspired cryptoanalysis methods, which were evaluated with the experiment presented in Section 5. The obtained results are gathered and analysed in Section 6, while Section 7 concludes the paper and provides future paths.

2 The Vigenère cipher

The Vigenère cipher was named after Blaise de Vigenère and proposed in 1586 [10]. It is a polyalphabetic cryptosystem, which was famous for centuries as being hard to break. For example, let's take the English alphabet as a base. The English alphabet consists of 26 letters or characters, thus, there exists $26! \approx 4.033 \times 10^{26} \approx 2^{88}$ of possible keys. It is evident, that an exhaustive key search is precluded.

Algebraically, we can present the Vigenère cipher using modular arithmetic. Let's define plaintext, key, and ciphertext as $P = (P_1, P_2, \dots, P_m)$, $K = (K_1, K_2, \dots, K_n)$, and $C = (C_1, C_2, \dots, C_m)$, where m and n denote the length of the plaintext/ciphertext and key, respectively.

Formally, we can now define the encryption E using the key K as:

$$C_i = E_K(P_i) = (P_i + K_i) \mod 26 \tag{1}$$

Decryption D using key K can be represented as

$$D_K(C_i) = (E_i - K_i) \mod 26, \tag{2}$$

where $D_K(C_i)$ denotes the offset of the i-th character of the P.

The encryption of the words *NEW TECHNOLOGIES* with the key *KEY* is presented in Table 1.

Table 1 Example of an encryption with the Vigenère cipher.

Plain text	N	E	W	T	E	C	H	N	O	L	O	G	I	E	S
Key	K	E	Y	K	E	Y	K	E	Y	K	E	Y	K	E	Y
Ciphertext	X	I	U	D	I	A	R	R	M	V	S	E	S	I	Q

3 Nature-inspired Algorithms

Nature-inspired algorithms, as the name implies, draw inspiration from nature [4]. Nowadays they are incorporated in solving many real-world optimization problems and are showing promising results. Because of that, we picked five popular algorithms from that group and utilized them in our proposed method. In the following sections 3.1-3.5, the selected algorithms are briefly presented. An in-depth explanation of nature-inspired algorithms can be seen in [4].

3.1 Differential Evolution

DE is a type of evolutionary algorithm and was proposed by Storn and Price in 1997 [11]. It uses mechanisms inspired by the theory of evolution, i.e. survival of the fittest. The fittest individuals of the population produce more offspring, therefore, their successors are carrying more advantageous genetic material and are more likely to survive. The pseudocode of the DE is outlined in Algorithm 1.

Algorithm 1 Pseudocode of the DE

```
1: init_population();
2: while termination_condition_not_met do
3:     random_choice_of_three_vectors();
4:     calculate_mutation();
5:     calculate_crossover();
6:     calculate_selection();
7:     update_population();
8: end while
```

3.2 Particle Swarm Optimization

PSO was proposed by Kennedy and Eberhart in 1995 [6] when they were inspired by the swarming behaviour of birds and fishes. In the PSO, particles are moving in the search space with given velocity and direction. The latter two change based on the personal best result of each particle and the global best of the swarm. The pseudocode of the PSO is outlined in Algorithm 2.

Algorithm 2 Pseudocode of the PSO

```
1: init_particles();
2: while termination_condition_not_met do
3:     evaluate_new_solution();
4:     save_the_local_best_solution();
5:     save_the_global_best_solution();
6:     update_velocity();
7:     update_position();
8: end while
```

3.3 Firefly Algorithm

Yang invented FA in 2008 [14]. The algorithm mimics the behaviour of the fireflies, but since direct transformation is not possible, the following adjustments were introduced: Fireflies are all unisex, attractiveness is proportional to the light intensity and, the light intensity is determined by a fitness function. The pseudocode of the FA algorithm is presented in Algorithm 3.

Algorithm 3 Pseudocode of the FA

```
1: init_population();
2: while termination_condition_not_met do
3:     evaluate_fireflies();
4:     order_fireflies();
5:     replace_fireflies();
6:     find_the_best_firefly();
7:     move_fireflies();
8: end while
```

3.4 Artificial Bee Colony Algorithm

ABC was defined by Karaboga in 2005 [5] and was inspired by the intelligent Foraging behaviour of honey bee swarm. The ABC algorithm consists of employed bees, onlookers and scouts, where each of them has a different task. Employed bees fly to their food source until the food source is abandoned. Onlooker bees are just watching employed bees and randomly choosing food sources, while scout bees try to find new food sources. The pseudocode of the FA algorithm is presented in Algorithm 4.

Algorithm 4 Pseudocode of the ABC

```
1: init_population();
2: while termination_condition_not_met do
3:     send_employed_bees();
4:     send_onlooker_bees();
5:     send_scouts();
6: end while
```

3.5 Cuckoo Search

Parasitism of some cuckoo species was inspiration for the CS algorithm proposed by Yang and Deb in 2009 [15]. The goal of the algorithm is to transfer the best nests, i.e. nests with the highest quality of eggs, into the next generation. Cuckoos' behaviour of laying eggs into randomly chosen nests and the host bird's decision of throwing the egg away or abandoning the nest is directly drawn from biology. The pseudocode of the FA algorithm is presented in Algorithm 5.

Algorithm 5 Pseudocode of the CS

```
1: init_host_nests_locations();
2: while termination_condition_not_met do
3:     generate_new_solution();
4:     evaluate_new_solution();
5:     randomly_choose_a_nest();
6:     optionally_replace_a_nest();
7:     optionally_build_a_new_nest();
8:     keep_the_best_nests();
9: end while
```

4 Nature-inspired Cryptoanalysis Methods

To perform a cryptoanalysis of the Vigenère cipher, we first have to identify the period of the cipher, i.e. the length of the key. The repetition of the key is one of the biggest weaknesses of this cipher and can be exploited by a Friedman attack that is based on the Index of Coincidence (I_c). The calculation of the I_c is presented in Eq. 3.

$$I_c = \frac{\sum_{i=1}^{j} f_i(f_i - 1)}{n(n-1)}, \tag{3}$$

where n denotes the length of the plaintext and f_i is the frequency count of the i-th letter. Frequency is calculated for all j letters of the alphabet. For an easier calculation, we can use an approximation of the key length [2, 3] presented in the following equation:

$$key\_length = \frac{0.027n}{I_c(n-1) - 0.038n + 0.065} \tag{4}$$

When we set on the key length, we have to find the actual key in the search space of j^{key_length} (for example, using the English alphabet, that is 26^{key_length}). For this difficult task, we utilize the nature-inspired algorithms presented in Section 3.

The fitness function is calculated the same for all algorithms using Eq. 5.

$$f(K) = \sum_{i=1}^{j} |FE_i - FD_i|, \tag{5}$$

where FE_i and FD_i denotes the frequency of the English language and frequency of the deciphered cipher text, respectively.

5 Experiment

The method was implemented in the Python programming language. All nature-inspired algorithms used in this method were taken from the external library NiaPy [13], which is a micro-framework for building nature-inspired algorithms. The experiment was run on a computer that has a Windows operating system and an Intel Core i7 processor with 16 GB of RAM.

The plain text length was 2,582 characters, which does not include spaces, numerals and special characters. To encode the plain text, we used four different sized keys, i.e. TEXT, CIPHER, VIGENERE, and CRYPTOGRAPHY, and the English alphabet. After the parameter tuning, the following parameter settings were used based on their best performance:

- DE (NP = 20, F = 0.5, CR = 0.9),
- PSO (NP = 200, c_1 = 1.3, c_2= 1.3, w = 0.7),

- FA ($NP = 30$, $\alpha = 0.5$, $\beta_{min} = 0.2$, $\gamma = 1.0$),
- ABC ($NP = 40$, $Limit = 2$),
- CS ($NP = 20$, $pa = 0.9$, $\alpha = 0.25$).

Each configuration of the algorithms was run independently five times. The termination condition was set to 50,000 fitness evaluations ($nFES$).

6 Results

This section outlines the results of cryptoanalysis methods for breaking the Vigenère cipher.

Table 2 presents the maximum number of correctly recovered key characters (MAX_KC), the minimum number of correctly recovered key characters (MIN_KC), and the average number of correctly recovered key characters (AVG_KC) per algorithm for all key sizes. It is apparent that only the DE algorithm managed to recover all of the key characters. PSO performed the second-best (average of 3.8, 5.4, 6, and 7.8 of correctly recovered key characters), closely followed by ABC (average of 4, 5.2, 6.4, and 7 of correctly recovered key characters). The FA algorithm recovered all keys only for the key size four, but its average numbers do not differ drastically from the ABC algorithm. The worst results were obtained by the CS algorithm, which had the smallest average number of recovered key characters for all key sizes.

We also measured the time that it took an algorithm to break the cipher (Table 3). The DE algorithm not only performed the best in recovering the highest number of key characters, but also took the least amount of time doing that. Like in Table 2, PSO performed second best, followed by ABC. The CS algorithm was relatively fast but did not manage to recover even a size 4 key.

Since each algorithm was run independently five times, we wanted to see how many times an algorithm managed to recover the whole key in those five times (Figure 1). The DE algorithm recovered the right key in every run for the key sizes four and six. At key size 8 was mistaken only once and at the key size 12, it succeeded to recover the right key one time. DE is the only algorithm that managed to do that.

Table 2 Number of correctly recovered key characters per algorithm.

Key size	**MAX_KC**					**MIN_KC**					**AVG_KC**				
	DE	PSO	FA	ABC	CS	DE	PSO	FA	ABC	CS	DE	PSO	FA	ABC	CS
4	4	4	4	4	3	4	3	3	4	2	4	3.8	3.8	4	2.6
6	6	6	5	6	3	6	5	5	4	3	6	5.4	5	5.2	3
8	8	8	6	8	5	6	5	4	5	2	7.6	6	5	6.4	3.8
12	12	9	8	8	5	11	6	5	6	3	11.2	7.8	6.6	7	4

Table 3 Time analysis (in seconds) of obtaining the maximum number of correctly recovered key characters. In cases when all key characters were recovered correctly, they are marked as bold.

Key size	DE	PSO	FA	ABC	CS
4	**35.7**	**178.8**	**349.7**	**180.2**	181.7
6	**38.3**	**184.9**	232.8	**187.3**	185.4
8	**40.2**	**214.9**	426.4	**246.6**	205.3
12	**52.8**	216.4	312.1	200.3	229.5

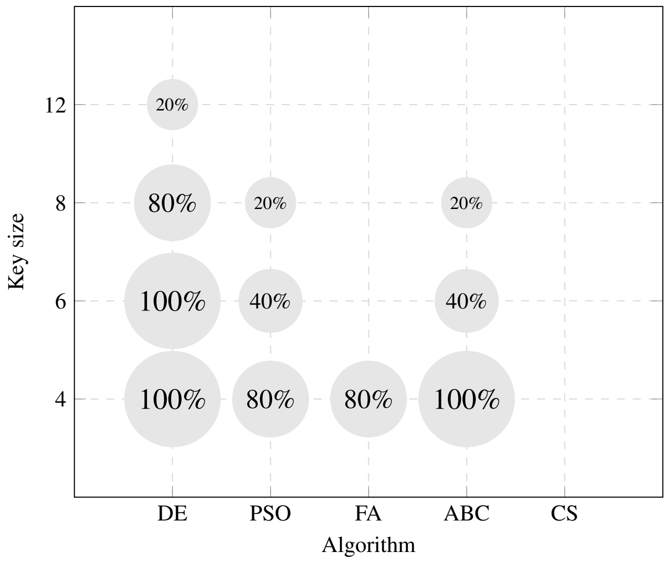

Fig. 1 Bubble plot for the percentage of all correctly obtained key characters in five independent runs regarding two variables: Algorithm (x-axis) and key size (y-axis).

7 Conclusion

A method that offers a solution for breaking the Vigenère cipher was analysed. It was tested with four different key sizes on the same plaintext. The results are revealing, in that the DE algorithm outperformed other algorithms considering the number of correctly recovered key characters and time.

The modular design of the method offers further utilization of other nature-inspired algorithms. Therefore, an in-depth comparison of the algorithms for a cryptoanalysis of the Vigenère cipher could be carried out. Furthermore, since all of the nature-inspired algorithms are sensible on their parameter settings, it would be a good way to perform parameter tuning. In the future, we are planning to introduce nature-inspired algorithms to modern cryptography.

Acknowledgements The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

References

1. I. K. Ali. Cryptanalysis of simple substitution ciphers using bees algorithm. *Baghdad College of Economic sciences University*, (36):373–382, 2013.
2. H. Beker and F. Piper. *Cipher systems: the protection of communications*. Northwood Books, 1982.
3. A. K. Bhateja, A. Bhateja, S. Chaudhury, and P. Saxena. Cryptanalysis of vigenere cipher using cuckoo search. *Applied Soft Computing*, 26:315–324, 2015.
4. L. Brezočnik, I. Fister, and V. Podgorelec. Swarm intelligence algorithms for feature selection: A review. *Applied Sciences*, 8(9), 2018.
5. D. Karaboga. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical report, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
6. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of International Conference on Neural Networks (ICNN'95)*, volume 4, pages 1942–1948, Perth, WA, 1995. IEEE.
7. P. K. Mudgal, R. Purohit, R. Sharma, and M. K. Jangir. Application of genetic algorithm in cryptanalysis of mono-alphabetic substitution cipher. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 400–405. IEEE, 2017.
8. S. Omran, A. Al-Khalid, and D. Al-Saady. Using genetic algorithm to break a mono-alphabetic substitution cipher. In *2010 IEEE Conference on Open Systems (ICOS 2010)*, pages 63–67. IEEE, 2010.
9. S. Omran, A. Al-Khalid, and D. Al-Saady. A cryptanalytic attack on vigenère cipher using genetic algorithm. In *2011 IEEE Conference on Open Systems*, pages 59–64. IEEE, 2011.
10. D. R. Stinson. *Cryptography: theory and practice*. Chapman and Hall/CRC, 2005.
11. R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
12. M. F. Uddin and A. M. Youssef. Cryptanalysis of simple substitution ciphers using particle swarm optimization. In *2006 IEEE International Conference on Evolutionary Computation*, pages 677–680. IEEE, 2006.
13. G. Vrbančič, L. Brezočnik, U. Mlakar, D. Fister, and I. Fister Jr. NiaPy: Python microframework for building nature-inspired algorithms. *Journal of Open Source Software*, 3, 2018.
14. X.-S. Yang. Firefly Algorithm. In *Nature-Inspired Metaheuristic Algorithms*, page 128. Luniver Press, 2008.
15. X.-S. Yang and Deb, Suash. Cuckoo Search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 210–214. IEEE, 2009.