Attack on the Simple Substitution Ciphers Using Particle Swarm Optimization

Dr. Hilal Hadi Salih*, Dr. Ahmed Tariq Sadiq*& Ismail K. Ali*

Received on: 6/4/2009
Accepted on: 16/2/2010

Abstract

This paper considers a new approach to cryptanalysis based on simulation of behavior of flocks of birds and schools of fish called Particle Swarm Optimization (PSO). It is shown that such algorithm could be used to break the key for a simple substitution cipher. This paper presents a proposed 2-opt PSO algorithm to enhance the efficiency of PSO algorithm on attacking simple substitution ciphers.

مهاجمة التشفير التعويضي البسيط باستخدام أمثلية حشد الجزيئة خوارزمية

الخلاصة

هذا البحث يعتمد نموذجاً جديداً لتحليل الشفرة معتمداً على طريقة محاكاة تعتمد على سلوكية أسراب من الطيور ومجموعات من الاسماك تسمى أمثلية حشد الجزيئة. تم التوصل الى أن مثل هذه الخوارزميات يمكن ان تستخدم لكسر المفتاح في أنظمة التشفير التعويضي البسيط. في هذا البحث سيتم تقديم خوارزمية مطورة لطريقة أمثلية حشد الجزيئة معتمدة على خوارزمية 2-opt وذلك لغرض كسر التشفير التعويضي البسيط.

*** Computer Science Department, University of Technology/ Baghdad**

https://doi.org/10.30684/etj.28.11.6
2412-0758/University of Technology-Iraq, Baghdad, Iraq

Introduction

Cryptography is the science of secret writing with the goal of hiding the meaning of a message [1]. Generally, cryptography methods are divided into 2 types symmetric and asymmetric. Cryptanalysis can be described as the process of searching for flaws or oversights in the design of cryptosystems (or ciphers). 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) [2].

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 [3].

Two fundamental goals in computer science are finding algorithms with probably 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 pretty 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. Here we have attempted to use heuristic techniques in the cryptanalysis of simple substitution cipher. Particle swarm optimization (PSO) [4] has demonstrated excellent promise as a heuristic technique in recent years. We propose an extension to this concept by introducing a new unique concept of PSO. This technique when applied to cryptanalysis of our candidate cipher has yielded better performance compared to the algorithms implemented in [5] incorporated into it. The rest of the paper is organized as follows. Section 2 presents a brief overview of the simple substitution ciphers. The underlying principle of particle swarm optimization (PSO) and our novel concept of PSO are presented in Section 3 and 4. Results of computational tests to evaluate the performance of these algorithms are reported in section 5.

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 cipher text. 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. Table 1 gives a sample key. Using this representation the ith letter in the plaintext alphabet is encrypted to the it element of the key. The string"the money is in the bag' is encrypted using the key in the table (1).

* Computer Science Department, University of Technology/ Baghdad

Using the key representation in Table 1, the original message can be discovered by reversing the encryption procedure. That is, the cipher text character at position I in the key decrypts to the ith character in the plaintext alphabet. For an alphabet of 26 characters, there are 26! or greater than 4×1026 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 [6, 7].

Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [4], inspired by social behavior of bird flocking or fish schooling and swarm theory. We have used this technique in the cryptanalysis of simple substitution ciphers.

The Principle

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas; function optimization, artificial neural network training and fuzzy system control. PSO simulates the behavior of bird flocking, suppose a group of birds are randomly searching food in an area. Not all the birds know where the food is. The effective strategy is to follow the bird that is nearest to the food. In PSO, each single solution is a "bird" in the search space. We call it "particle". All particles have fitness values that are evaluated by the fitness function to be optimized, and have velocities, which direct the flying of the particles [4].

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generation. In each generation, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and is called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest. After finding the two best values, the particle updates its velocity and position with the following equations [4]:

$$Vi[t+1] = w\ Vi[t] + C1 * r1 * (pbesti[t] - presenti[t]) + C2 * r2 * (gbesti[t] - presenti[t]) \quad \ldots\ldots.(1\text{-}a)$$

$$presenti[t]=presenti[t]+Vi[t] \quad ...(1\text{-}b)$$

where, i =1,2,…..,N; w is the inertia weight, V[t] is the particle's velocity, present[t] is the current particle

(solution), pbest and gbest are defined as stated before, r1 and r2 are two random numbers between (0,1), C1 and C2 are learning factors. However these values of C1, C2 are problem dependent. These are very essential parameters in PSO [4]. Particles' velocities on each dimension are clamped to a maximum velocity Vmax. In our implementation, each particle corresponds to a key. The fitness function of a key discuss in the section (4-C).

Binary PSO

The original PSO is for continuous population but is later extended by Eberhart and Kennedy [8] to the discrete valued population. In the binary PSO thus emerged, the particles are represented by binary values (0 or 1). The velocity and particle updating for binary PSO are the same as in the case of continuous one. However, the final decisions are made in terms of 0 or 1. Sigmoid function in [4] is used to restrict the decision in the range [0, 1].

Discrete PSO

Several adaptations of the method to discrete problems, known as Discrete Particle Swarm Optimization (DPSO). Since, in words of the inventors of PSO, it is not possible to "throw to fly" particles in a discrete space [4], several Discrete Particle Swarm Optimization (DPSO) methods have been proposed.

A DPSO whose particles at each iteration are affected alternatively by its best position and the best position among its neighbors was proposed by Al-Kazemi and Mohan [10]. Pampara et al. [11] solved binary problems by combining continuous PSO and Angle Modulation with only four parameters. Furthermore, several PSO variants applied to problems where the solutions are permutations were considered in [12, 13]. The multi-valued PSO (MVPSO) proposed by Pugh and Martinoli [14] deals with variables with multiple discrete values.

Another DPSO was proposed in [15] for feature selection problems, which are problems whose solutions are sets of items. A new DPSO proposed in [16, 17] does not consider any velocity since, from the lack of continuity of the movement in a discrete space, the notion of velocity loses sense; however they kept the attraction of the best positions.

Parameters of PSO

The convergence and performance of PSO are largely dependent upon parameters chosen w is termed as inertia weight [9] and is incorporated in the algorithm to control the effect of the previous velocity vector of the swarm on the new one. It facilitates the trade-off between the local and the global exploration abilities of the swarm and may result in less number of iterations of the algorithm while searching for an optimal solution. It is experimentally found that inertia weight w in the range [0.8, 1.2] yields a better performance [18]. The velocity lies in the range [-Vmax , Vmax] where, - Vmax denotes the lower range and Vmax is the upper range of the motion of the particle. The roles of C1 and C2 are not so critical in the convergence of PSO, however, a suitably chosen and fine tuned value can lead to a faster convergence of the algorithm. A default value of C1 =C2 = 2 is

suggested for general purpose, but somewhat better results are found with C1 = C2 = 0.5 [19]. However, the values of cognitive parameter, C1 larger than the social parameter C2 are preferred from the performance point of view with the constraint C1 +C2 ≤ 4 [20]. The parameters r1 and r2 used to maintain the diversity of the population in equation (1a).

PSO for Cryptanalysis Substitution Ciphers

PSO is a rather successful method for the continuous optimization problems; however, it is very difficult to adapt it for the discrete case; many studies have been done on it. The focus of this paper is to apply a PSO algorithm to the problem of searching through the key space of a simple substitution cipher. The implementation part of our attack involves a different phase that will be described below:

A. Representation of a Candidate Solution

Since 1995, Particle Swarm Optimization has been successfully applied in continuous problems, and substitution ciphers are the new field of PSO algorithm in discrete space. If the PSO algorithm is designed to tackle discrete problem, a direct correlation must be found between the particle vector and the solution representation of substitution cipher.

Lending idea from GA attack of the simple substitution cipher, 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 PSO.

We define the position vector of a particle represents a feasible solution, that is to say, the position vector means the permutation of the solution key and specifies the order of key to be executed. 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. The order expresses the underlying substitution. Every letter in the scale appear only once in feasible solution.

B. Initial Population

Like other evolutionary algorithms, the PSO algorithm starts from an initial population. The particles are randomly generated between the maximum and the minimum operating limits of the generators.

C. Fitness Function Assessment

The value of a fitness function is calculated. A higher result of the function implies a higher fitness of the chosen key, and therefore, a closer answer to the correct key. The fitness function can have different forms which are used to the analysis of single characters (unigram analysis), pairs of characters (bigram analysis), triplets of characters (trigram analysis), or any combination of any of these. The most commonly used fitness function for breaking the simple substitution ciphers is based only on unigram and bigram analysis and has the form:

$$\text{Fitness}=(1-\left\{\sum_{i=1}^{26}\left|SF[i]-DF[i]\right|+\sum_{j=1}^{26}\left|SDF[i,j]-DDF[i,j]\right|\right\}/4^{8} \quad \ldots\ldots (2)$$

where, SF[i] and SDF[i, j] are standard frequencies of character i and the pair of symbols (i, j) in the

original language, respectively, and DF[i] and DDF[i, j] are measured frequencies of character i and pair (i, j) in the "decrypted" text. The measured errors (i.e., the difference between standard frequencies and measured frequencies) are normalized and subtracted from 1, so that number closer to 1 represents the higher fitness. In addition, the constants 4 and 8 are used to reduce sensitivity to large errors and to amplify small differences, respectively. Some fitness functions give more weight to one of the factors (unigram or bigram) by inserting weight constants in front of the corresponding error calculation terms [21, 5].

D. Proposed PSO Algorithm

The following is an algorithmic description of the attack on a simple substitution cipher using particle swarm optimization (PSO):

1. The algorithm is given the cipher text (and its length) and the statistics of the language (unigrams, bigrams).
2. Initialize the algorithm parameters. They are:
   a. C1 (Self confidence)
   b. C2 (Swarm confidence)
   c. W (Inertia weight)
   d. Vmax(Themaximumof velocity)
   e. Number of partition of key size
   f. Number of particles in the swarm
   g. Max-iteration (The maximum number of iterations)
3. Randomly generate the initial particles (keys of the simple substitution cipher) to form a swarm.
4. For i=1 To Max-Iteration, do
   a. Calculate the fitness function of each of the particles (keys) using equation (2).
   b. If the current position of the particle is better than the previous history, update the particles to indicate this fact
   c. Find the best particle of the swarm. Update the positions of the particles by using equation (1a and 1b).
5. Copy the best key obtained so far in the output key variable and exit.

E. Improved PSO Algorithm

Once a new solution is generated, a decision must be made whether or not to accept the newly derived vector as next iteration. The general strategy is the greedy technique, in which the solution having better performance is accepted. The algorithm using greedy criterion can converge fairly fast, but it runs the risk of becoming trapped by a local minimum. In typical PSO, all the new derived solutions are transferred into next iteration without selection. This method can preserve the diversity and lead to slow convergence as well. To improve the global convergence, the mechanism of 2-opt is introduced into PSO algorithm. The 2-op algorithm based on exchange of information position, that means in PSO it will exchange particles position (except the best one). The hybrid algorithm effectively integrates both the ability to jump out of the local minima and the capacity of searching the global optimum in PSO algorithm. In original 2-opt, a system is a basic case of the local search optimization heuristics, and as such it is capable of obtaining useful results very fast and

would give us a higher chance of a near to optimal (if not the optimal), the 2-opt is a feasible choice when looking at its results versus the time required to obtain these results. The implementation of the 2-opt is based on the following points:

1. Randomly select two elements in the particle and swap them.
2. Check to see if the new particle is better than previous one.

The swap can be performed in two different ways:

- Search until the first possible improvement is found, and perform the swap.
- Search through the entire key to find all possible improvements, and perform only a swap on the best improvement.

We choose to use the first option, as the second one could possibly run for a much longer time before returning a result, as it has to run through its entire list of neighbors before it performs a swap, whereas the first on performs the swap as soon it hits the first possible improvement. The disadvantage of choosing the first one over the second one is that we might lose a potential good improvement.

Then the improved PSO proposed algorithm is described as the follows:

1. The algorithm is given the cipher text (and its length) and the statistics of the language (unigrams, bigrams).
2. Initialize the algorithm parameters. They are:
   a. C1 (Self-confidence).
   b. C2 (Swarm confidence).
   c. W (Inertia weight).
   d. Vmax (The maximum of velocity).
   e. Number of partition of key size.
   f. Number of particles in the swarm.
   g. Max-iteration (The maximum number of iterations).
3. Randomly generate the initial particles (keys of the simple substitution cipher) to form a swarm.
4. For i=1 To Max-Iteration do
   a. Calculate the fitness function of each of the particles (keys) using equation (2).
   b. If the current position of the particle is better than the previous history, update the particles to indicate this fact
   c. Find the best particle of the swarm. Update the positions of the particles by using equation (1a and 1b).
   d. Use 2-opt technique to select the next iteration.
5. Copy the best key obtained so far in the output key variable and exit.

5. Experimental Results

PSO and improved PSO were used for cryptanalysis of simple substitution. We summarize the results of these experiments in this section. The algorithm was applied to cipher text 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 (100) iterations were usually enough to break the cipher text and the algorithm was fast enough that this took a few seconds.

The experiment shown in Figure 1 was performed on simple

substitution cipher using basic PSO and improved PSO algorithm. The fitness function used was in equation (2). As we can see, the fitness value has started from about 0.42 and come up to 0.85 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.

Also, PSO outperforms all other optimization techniques in discrete optimization problems. In this case, we have observed similar findings. As we will see in the next experiment, improved PSO performs slightly better than basic PSO, in the same figure we can see the results of improved PSO (IPSO). At the beginning, it was ensured that swarms start at different positions. However, it is very difficult to ensure that trajectories of the swarms do not intersect during the execution. The X- axis here shows the number of iterations. As we can see, the fitness value has started from above 0.50 and has come up to 0.87. The increasing in fitness value continues even after 300 iterations. Thus, PSO is a very promising approach for solving the problems of cryptanalysis of simple substitution 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. One of the ideas suggested in this paper, is the exploration of improved PSO in speed execution environment.

The best parameters value of PSO and 2-op PSO to attack substitution cipher show in table (2).

Conclusions

The goal of this work is to illustrate the feasibility of using particle swarm optimization algorithm as a cryptanalysis tool. Hence, our algorithm is applied to simple substitution ciphers. Also, this paper proposed a hybrid approach involving basic PSO and 2-opt technique. The advantages of the hybrid approach are in making use of the fast convergence property of PSO. The experimental results show how PSO and improved PSO are successful in finding the key for a simple substitution ciphers and give a good performance. The proposed algorithms are open avenues for further research in cryptanalysis and other more complicated cryptosystems.

References

[1] Christof Paar and Jan Pelzl, "Understanding Cryptography", Springer-Verlag Press, Berlin, Germany, 2009.

[2] Mao, W., "Modern Cryptography: Theory & Practice". Upper Saddle River, NJ: Prentice Hall PTR, 2004.

[3] Giddy J. P. and Safavi-Naini R., "Automated Cryptanalysis of Transposition Ciphers". The Computer Journal, 37(5):429–436, 1994.

[4] Kennedy J. and Eberhart R., "Particle Swarm Optimization", in Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[5] Spillman, R., Jansses, M., and Kepner, M., "Use of Genetic Algorithm in the Cryptanalysis of Simple Substitution Ciphers", Cryptologia, vol. 17, issue 1, January 1993,pp. 31-34, 1993.

[6] Douglas R. Stinson., "Cryptography: Theory and Practice". CRC Press, Boca Raton, Florida, USA, 1995.

[7] Henry B. and Fred P., "Cipher Systems: The Protection of Communications", Wiley-Inter Science, London, 1982.

[8] Eberhart R.C. and Kennedy J., "A New Optimizer Using Particle Swarm Theory", Proc. 6th Symposium on Micro Machine and Human Science, pp. 39–43, 1995.

[9] Khanesar M., Teshnehlab M., and Shoorehdeli M., "A Novel Binary Particle Swarm Optimization", Proc. 15th Mediterranean Conference on Control and Automation, 2007.

[10] Al-Kazemi, B and Mohan C., "Multi-phase Discrete Particle Swarm Optimization". In: Fourth International Workshop on Frontiers in Evolutionary Algorithms, Kinsale, Ireland, 2002.

[11] Pampara G., Franken N. and Engelbrecht A., "Combining Particle Swarm Optimization with Angle Modulation to Solve Binary Problems". In: Proceedings of the IEEE Congress on Evolutionary Computing, vol 1, pp 89-96, 2005.

[12] Onwubolu G. and Clerc M., "Optimal Operating Path for Automated Drilling Operations by a New Heuristic Approach Using Particle Swarm Optimization". International Journal of Production Research 42(3):pp.473-491, 2004.

[13] Pang W., Wang K., Zhou C. and Dong L., "Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem". In: Proceedings of the 4th International Conference on Computer and Information Technology (CIT04), IEEE Computer Society, vol 1, pp 89-96, 2004.

[14] Pugh J. and Martinoli A., "Discrete Multi-Valued Particle Swarm Optimization". In: Proceedings of IEEE Swarm Intelligence Symposium, vol 1, pp 103-110, 2006.

[15] Correa E., Freitas A. and Johnson C., "A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bio-informatic data set". In: Proceedings of GECCO 2006, pp35-42, 2006.

[16] Martinez-Garcia F., Moreno-Perez J., "Jumping Frogs Optimization: A New Swarm Method for Discrete Optimization". Tech. Rep. DEIOC 3/2008, Dep. of Statistics, O.R. and Computing, University of La Laguna, Tenerife, Spain, 2008.

[17] Moreno-Perez J., Castro-Gutierrez J., Martinez-Garcia F., Melian B, Moreno-Vega J. and Ramos J., "Discrete Particle Swarm Optimization for the p-median Problem". In: Proceedings of the 7th Metaheuristics International Conference, Montreal, Canada, 2007.

[18] Shi Y. and Eberhart R., "A Modified Particle Swarm Optimizer", Proc. IEEE Conference on Evolutionary Computation, 1998.

[19] Kumar A., and Zhang D., "Palm Print Authentication Using Multiple Representation", Pattern

Recognition Journal, vol. 38, pp.1695-1704, 2005.

[20] Carlisle A. and Dozier G., "An off-the-Shelf PSO", Proc .Particle Swarm Optimization Workshop, pp. 1–6, 2001.

[21] Clark, A., "Modern Optimization Algorithms for Cryptanalysis". In Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, November 29 - December 2,pp. 258-262, 1994.

Table (1) Example Substitution Cipher

KEY	
Plaintext	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext	PQOWIEURYTLAKSJDHFGMZNXBCV
ENCRYPTION	
Plaintext	T H E M O N E Y I S I N T H E B A G
Ciphertext	M R I K J S I C Y G Y S M R I Q P U

Table (2)
Parameters Value of PSO and 2-op PSO to Attack Substitution Cipher

Parameter	Symbol	Value
Self-confidence	**C_1**	**2**
Swarm confidence	**C_2**	**2**
Inertia weight	**W**	**1.2**
The maximum of velocity	**V_{max}**	**26**
Number of particles in the swarm	**Swarm_Size**	**25-40**
The maximum number of iterations	**Max-Iter**	**100-300**

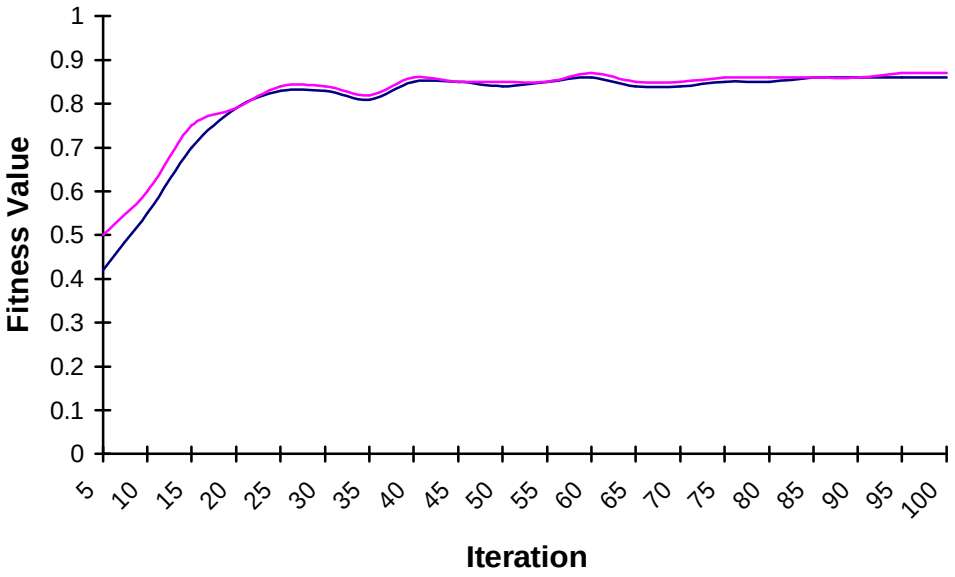

Figure (1) Performance of PSO & IPSO

* Computer Science Department, University of Technology/ Baghdad