Natural Heuristics for Cryptanalysis

T. Mekhaznia
Department of Computing Science
Tebessa University
Tebessa, Algeria
mekhaznia@yahoo.fr

ABSTRACT: *Natural Metaheuristics have been widely used to solve difficult optimization problems. This is mainly due to their ability to converge in reasonable time. However, they remain inefficient where handling large instances. Current research tends to improve these techniques in order to produce hybrid algorithms able to solve these problems. The encryption information has long been a challenge for cryptanalysis. It was always attacked by classical techniques and heuristics but results are limited.*

In this paper, an approach of a PSO algorithm was proposed for cryptanalysis of classical encryption. After comparison with other heuristics in the same category, we show experimentally the superiority of PSO on the instances tested and underline the difficulties encountered.

Keywords: Natural heuristics, PSO, Cryptanalysis

Received: 20 February 2012, Revised 2 April 201, Accepted 7 April 2012

1. Introduction

Cryptanalysis allows to extract readable messages from an encrypted texts without knowing the decryption key. Therefore, being abusive, but contributes to the improvement of data security. Ciphering existed since antiquity. Cryptanalysis by brute force was the first used in this area. Currently, it's no longer effective of its abundant resources consumption. With the arrival of calculators, other alternatives have developed, linear and differential cryptanalysis was among them. They are able to break certain encryption in an acceptable time but are ineffective against modern cryptography seen their specificity and reduced setup. Research tends toward heuristics methods that have the ability to handle more large and diversity of instances.

Taking part of natural heuristics, the particle swarm optimization (PSO for short) have the ability to achieve an acceptable solution with minimal resources [1]. In this context, one approach has been implemented and tested on several instances from classical cryptography algorithms. The paper also includes a comparative performance of similar algorithms. It is organized into five sections: the first being the introduction to the problem, the second provides an overview of the PSO, properties and alternatives. The third section illustrates the problem of cryptanalysis and various forms of resolution. The fourth section presents the model and implementation of the approach. A study comparison with some other results in the literature is presented in the fifth section, followed by a brief discussion and conclusion of work.

2. Natural algorithms

2.1 Introduction

The heuristics let satisfy at least one of the two goals of information technology: the generation of solutions with maximum benefit and minimal resources consumption, however, no proof of optimality of the solution can be pronounced [2].

The PSO, part natural heuristic algorithms dates back to 1995 [3]. They are inspired by social behavior of organisms that live in groups, including the swarming flocks of birds, fish, or colonies of bees. PSO evolve with a population of individuals called *particles* which, at each time interval, the best individuals (according to a predefined criterion as finding food), in their activity, are followed by other towards the better goal. PSO algorithms provide optimal solutions or close to optimal for multidimensional numerical problems [4], optimization [5], automation [6], biometrics identification [7] and other mathematical and scientific fields [8].

2.2 Characteristics

2.2.1 Principle

The population of swarm moves in space research respecting two rules:

Cohesion: not to hurt neighbors or move away from the group,

Alignment: each particle moves in the same way with a relative speed and direction as the whole of collective.

These basic rules allow the attraction and repulsion of each individual and maintain the stability of group movement which tends to converge towards an optimum goal represented by a *cost function*.

2.2.2 Algorithm

In the search space R^n, each particle $\boldsymbol{i}$, at a given time $\boldsymbol{t}$, has a history of activities called *coefficients of confidence* [9]. It has a position $\boldsymbol{x_i^t}$, the best position in its neighborhood $\boldsymbol{p_i^t}$, and the position $\boldsymbol{g^t}$ of particle in the best position of all. At each time *t*, the particle *i* moves with a speed $\boldsymbol{v_i^t} \in R^n$ to a new position calculated based on values $\boldsymbol{x_i}$, $\boldsymbol{p_i}$ and $\boldsymbol{g}$ according to equation [11]:

$$\boldsymbol{v_i \leftarrow c_i v_i + c_p(p_i - x^t) + c_g(g - x_i)} \quad (1)$$

The simplified algorithm of the particle path will be as follows:

Initialize data: population: m, positions $\boldsymbol{x}_i$ *(i=1..n), iter_max*
Search space: n nods
For each particle $\boldsymbol{i}$,
Evaluation cost function $f(\boldsymbol{x}_i)$
Initialize $\boldsymbol{p}_i \leftarrow \boldsymbol{x}_i$
If $(f(\boldsymbol{p}_i) > f(\boldsymbol{g}))$ *update the swarm's best known position:* $\boldsymbol{g} \leftarrow \boldsymbol{p}_i$
Initialize the particle's velocity: $\boldsymbol{v}_i \sim U(v_{min}, v_{max})$
Repeat:
For each particle i:
Pick random numbers: $c_i, c_p, c_g \sim U(C_{min}, C_{max})$
Update particle's velocity: $\boldsymbol{v_i^{t+1}} \leftarrow c_i \boldsymbol{v_i^t} + c_p(\boldsymbol{p_i^t} - \boldsymbol{x_i^t}) + c_g(\boldsymbol{g^t} - \boldsymbol{x_i^t})$
Update the particle's position: $\boldsymbol{x_i^{t+1}} \leftarrow \boldsymbol{x_i^t} + \boldsymbol{v_i^{t+1}}$
If $(f(\boldsymbol{x}_i) > f(\boldsymbol{p}_i))$ *do*
Update the particle's best known position: $\boldsymbol{p}_i \leftarrow \boldsymbol{x}_i$
If $(f(\boldsymbol{p}_i) > f(\boldsymbol{g}))$ *update the swarm's best known position:* $\boldsymbol{g} \leftarrow \boldsymbol{p}_i$
Endfor
Until (**g** *holds the best found solution) or* (*iter_max*)
Report results$(g, f(g))$

2.2.3 Variantes

The basic PSO has been designed for continuous instances. Given the diversity of combinatorial problems, other alternatives have been developed in order to solve other kinds of data, such:

Discrete-PSO [10]: where the particle velocity is limited to the interval [0,1]. **Binary-PSO** [11] is a variant where each particle moves independently of the others, this principle avoids blocking in local minima. **Multivalued-PSO**, version designed to solve combinatorial multidimensional problems. **Genetic-SO** [12], a hybrid technique that combines the advantages of PSO and genetic algorithms to overcome the premature convergence of the process. Other versions deduced from hybridization with such heuristic techniques, including **DMS-PSO** [13], **GPSO** [14], **OPHL** [15], etc.

2.3 State of art

In literature, there is various studies for modeling and development of PSO algorithms: [16] showed that DPSO is not as effective in regard of some specific algorithms, but on the other hand, it is easily adaptable to any combinatorial problem for which does not have a dominant algorithm. [17] Developed a version of DPSO to vary the value of the speed. [18] has developed a model of CPSO using only four parameters. It's allowed to converge quickly. [19] Proposed a hybrid version called MHPSO which integrates the genetic crossing within PSO algorithm in order to expand the exploration of the search space. The approach was designed to solve multidimensional problems. [20] Proposed a version in which all particles are mutual attracted.

3. Classical cryptanalysis

3.1 Definition

Let $\boldsymbol{A = \{a_0, a_1, .., a_{n-1}\}}$ and $\boldsymbol{B = \{b_0, b_1, .., b_{n-1}\}}$ are two sets of ASCII or binary characters and $\boldsymbol{k}$, a bijective function $\boldsymbol{k: A \rightarrow B}$, which transforms two characters $\boldsymbol{a_i, a_j}$ of $\boldsymbol{A}$ to $\boldsymbol{b_i}$. $\boldsymbol{k}$ is called ***encryption key***, it transform a plaintext A to a ciphertext B. The inverse function $\boldsymbol{k^{-1}}$, is called decryption key. The transformation (or encryption) can be a simple substitution, transposition, or a complex mathematical or logical function. Various classic encryption algorithms are available in literature: Vegeneire Cipher [21], Hill encryption [22], affine cipher [23] and various versions of polyalphabetic substitutions. The following example illustrates a cryptogram using a table: The first two entries are the encryption key. Other entries are respectively plain and cipher messages.

k	A B C D E FG H I J K L M N O P Q R S T U V W X Y Z
	P O I U Y T R E Z A M S K J H G F D L Q N B V C X W
A	C R Y P T A N A L Y S I S P E R S U B S T I T U T IO N
B	I D X G Q P J P S X L Z L G Y D L N O L Q Z Q N Q Z H J

Table I. Résults (B) of cifering of (A) using k

3.2 Statistics tables

The frequency of occurrence of any character in the alphabet in a given text is different from one language to another. The frequency of appearance of the alphabet in an English text (***unigram***) is presented in the order ETAON RISHD LFCMU GYPWB VKXJQ Z [24]. In other words, the letter E is the one that appears most often in a text. The frequency of occurrence of pairs of letters (***bigrams***) is given by the following order: TH HE ER RE IN ON AN AT ND ES OF TE ED IT OR AS TO HI and occurrence of letters within the same word is presented in the order: LL EE TT SS OO RR NN PP FF CC. Other specifics regarding the redundancy of certain groups of characters were found. For example: the appearance of the pair NT is often more important than BT, while the pair JX or identical vowels is rarely used.

Various other projects are present in the literature [25] [26]. The ***ICE*** is the most popular. It includes statistics from several variants of the English language from a dozen English-speaking countries [27]. In general, the average statistics of appearance of characters were compiled in tables, called character frequencies tables**.** They are used as references when decrypting texts to determine the nature of characters according to their occurrence in text. A model is shown in the following example:

3.3 Cost function

When decrypting, the difference between the frequency of the initial character and the one with which it was swapped is even smaller than text produced is close to the plaintext. The cost function most used is illustrated by the equation:

$$cost(k) = 1 - abs\left(\sum_{i=1}^{n}[D(i) - C(i)] + \sum_{i=1}^{n}[D(i,j) - C(i,j)]\right) \quad (2)$$

Char	Freq.	Char	Freq.	Char	Freq.
a	8.167	j	0.153	s	6.327
b	1.492	k	0.772	t	9.056
c	2.782	l	4.025	u	2.758
d	4.253	m	2.406	v	0.978
e	12.702	n	6.749	w	2.360
f	2.228	o	7.507	x	0.150
g	2.015	p	1.929	y	1.974
h	6.094	q	0.095	z	0.074
i	6.966	r	5.987		

Table 2. Character frequency [28]

where k is the deciphering key. D, C denote the portions of the ciphertext and plaintext produced after swapping characters (i) and pairs of character (i, j). The function will be 1 if k is much to encryption key.

3. 4 Importance of classical cryptanalysis

Modern symmetric cryptosystems uses concepts of classical encryption in order to develop their keys by using Feistel networks [29]. The AES algorithm [30] uses a product of substitution and combination. The IDEA algorithm [31], uses three operators: substitution, permutation and XOR. The RC algorithm uses random permutation operations and rotation. The SEAL algorithm uses various logical operations including XOR and a binary transposition. However, all bases of classical encryption are used in building blocks of modern ciphers, which means that the classical ciphers are usually first regarded when it comes to develop new cryptanalysis attacks [32].

4. Approch proposed

4.1 Wave of Swarm of particles (WSOP)

It is a variant of PSO proposed by [33] [34] used to avoid blocking around local minima. Its principle is the existence of the concept of ***wave***, a factor of excitement that will be exercised on certain portions of the search space in order to disperse grouped particles and thus to extend the exploration away from local minima.

4.2 Difficulties in implementing the WSOP

Analysis of PSO variants described in above shows the WSOP is so far from being the best suited algorithm to solve the classical cryptanalysis problems. Several other discrete versions would do likewise. However, cryptanalysis using frequency tables has some specific characteristics illustrated by the following:

- A performance of a high function cost does not imply a totally decrypted text because the characters have an approximate equal costs and are interchangeable (m/w, c/u, x/d, ..) ,

- The fact to swap two wrong characters will change cost function. This fact allow particle to follow, in sometimes the wrong direction and perturb their neighbors.

4.3 Modelization of WSOP

The advantage of this algorithm compared with other PSO, since it is semi-automatic, in other words, the factor of excitement $\boldsymbol{E_f}$ supervises the movement of particles. It records their parameters during the course. It intervene when particle loop around a finite number of nodes many times or a digression of the cost function happens. Its role is to force the concerned particles to change direction in order to avoid minima local.

4.4 Material and method

4.4.1 Initial data and initialization

The field of exploration is a strongly connected graph of $\boldsymbol{n}$ nodes where each node corresponds to a character. The population is composed of a 1) swarm of $\boldsymbol{m}$ particles (keys). Each particle has a position $\boldsymbol{x_i}$, a vilocity $\boldsymbol{v_i}$, a coefficient of enertie $c_{i.}$ and its best position p_i . $\boldsymbol{E_f}$ parameters are: $\boldsymbol{W_m}$, number of blocked particles and $\boldsymbol{W_{iter}}$, number of suspecious blocking iterations.

4.4.2 Iterations

The displacement of particle *i* on the arc $(\boldsymbol{x_i}, \boldsymbol{y_i})$ corresponds to a key generation $\boldsymbol{k_x}^{\boldsymbol{t}}$, obtained by swapping characters x and y and compute cost of the decipherd text function $f(x_i^t)$.

Since at each node, the particle will choice its next move according to equation (1), the factor of excitation $\boldsymbol{E_f}$ records for each particle its new choice $\boldsymbol{x_i}$ and check it in particle history. It also checks the number of *valid blocks* in the decrypted text (bigrams and trigrams as mentioned in III.B).

Factor of excitation intervenes if $\boldsymbol{x_i}$ appears $(\boldsymbol{W_{iter}})$ times in $(\boldsymbol{W_m})$ particles history or there is a digression of valid blocks $(\boldsymbol{W_{iter}})$ times also.

4.4.3 End of process

Il happens when $\boldsymbol{f(g)}$ reached a fixed value (0.8..1) in general, or after a fixed number of iterations.

4.5 Algorithm WSOP

The proposed algorithm describes the course of particles (without local minima) in order to optimize cost function. It ilustrate as follows:

Inputs: population : m, positions: $\boldsymbol{x}_i$ $(i = 1..n)$, *max_iterations*
particles parameters : $\boldsymbol{v}_i$ $\boldsymbol{c_i}$ $\boldsymbol{p}_i$ $(i = 1..m)$, *g,* $\boldsymbol{c_p}$,$\boldsymbol{c_g}$
Initialize (for all particles)
positions $\boldsymbol{x}_i^0$
best neighbours $\boldsymbol{p}_i^0 \leftarrow \boldsymbol{x}_i^0$
pathHistory $[1..max\_iterations] \leftarrow \boldsymbol{x}_i^0$
$Nbr\_valid\_blocs^0 \leftarrow 0$
Pick random Wave iteration $\boldsymbol{W_{iter}}$ $(\in U|1, W_{max}|)$
Repeat
For each particle $\boldsymbol{i}$ *:*
Initialize particle's velocity: $\boldsymbol{v}_i^0$ $(\in U|\boldsymbol{b_{up}}, \boldsymbol{b_{lo}}|)$*:*
Pick random numbers: $\boldsymbol{c_i}$,$\boldsymbol{c_p}$,$\boldsymbol{c_g}$ $(\in U(0,Cmax))$
Update particle's parameters:
$\boldsymbol{v}_i^{t+1} \leftarrow \boldsymbol{c}_i \boldsymbol{v}_i^t + c_p\,(\boldsymbol{p}_i^t - \boldsymbol{x}_i^t) + \boldsymbol{c_g}\,(\boldsymbol{g}^t - \boldsymbol{x}^t))$
$\boldsymbol{x}_i^{t+1} \leftarrow \boldsymbol{x}_i^t + \boldsymbol{v}_i^t$
$pathHistory[] \leftarrow \boldsymbol{x}_i^{t+1}$
compute $(Nbr\_valid\_blocs^{t+1})$
If $(f(\boldsymbol{x}_i^{t+1}) > f(\boldsymbol{p}_i))$*Update particle's best position:* $\boldsymbol{p}_i \leftarrow \boldsymbol{x}_i^{t+1}$
If $(f(\boldsymbol{p}_i) > f(\boldsymbol{g}))$ *update swarm's best position:* $\boldsymbol{g} \leftarrow \boldsymbol{p}_i$
If$(\boldsymbol{x}_i^{t+1})$ *in pathHistory*[] *for* $\boldsymbol{W_{iter}}$ *times OR*
If$(Nbr\_valid\_blocs^{t+1}) << (Nbr\_valid\_blocs^t)$ *swap* $(\boldsymbol{c_i}, \boldsymbol{c_p}, \boldsymbol{c_g})$
Until ($\boldsymbol{g}$ *holds the best found solution*) *or* (*max_iterations*).
Output: $\boldsymbol{g}$*: best decryption key*

5. Experiments & Conclusion

5.1 Data of experiments

Experiments were carried out on different ASCII text passages extracted from ICE [35]. These were previously encrypted with some classical algorithms such Vigéneire, Polybius and Affine. A simplified model of Feistel network was also tested.

5.2 Evaluation of Algorithm

Treatment was operated with a C++ code on a PC 2.2 GHz and using some standard values of constants that found in literature: particles 20 to 30, rate constants between 0.5 and 1.5 and a text of 200 characters. An average of the results obtained after 150 iterations are presented in the following table.

Algorithm of Ciphertext	**Key length**	**Cost**	Avg. much Chars (max 27)	**Time (ms)**
Polyalph. Substitution	4	0.85	21.11	850
Transposition	8	0.88	21.78	1220
Vigéneire	6	0.92	25	800
Delastelle	5 x 5	0.58	18.35	1058
Feistel network (2 rounds)	XOR % 26	0.43	15	1520

Table 3. Résults of cryptanalysis by Wpso

A similar study was conducted with other heuristics, ACO[36], AG[37] and GPSO for a laps time of 2000 ms. The number of much char obtained are illustrated by the following table:

Algorithm of ciphertext	Number of much Chars (max 27)		
	Genetic Algorithm	Group SOAlgorithm	ACO Algorithm
Polyalph. Substitution	8	15	9
Vigéneire	8	18	10
Polybe/Delastelle	10	21	13
Feistel network(2 rounds)	7	13	8

Table 4. Résults of cryptanalysis by other natural heuristics

6.3 Synthesis with literature

Various heuristics have been subject of experimentation in the attacks of ciphers. The results were varied depending on many circumstances: nature of the texts chosen, encryption algorithm, statistical tables used and the values of parameters of the algorithms tested. The most significant conclusions of this work are summarized by the following:

• The results presented by [38] using genetic algorithms were interesting: fully decrypted texts. Nevertheless, the size of these texts seems quite large: 2000 characters, which exceeds the size most common encrypted messages exchanged across networks, which is, statistically between 90 to 300 characters.

• A comparison between GA and TabuSearch algorithms was conducted by [39], The results were in favor of the GA in testing small texts (less than 800 characters). Otherwise, it is the TS that becames well. However, AG is more consuming in resource.

• H. Hadi [40] proposed a hybrid approach using PSO and IPSO for cryptanalysis of substitution cipher. The tests gave a result almost identical for both versions: a cost function of 0.9 was reached. However the text was too short: twenty characters. The study found the best parameter values used.

• Ganapathi [41] had presented a comparison between GA and PSO to search for Vigéneire key encryption. Tests showed that the technique PSO may be substantially

• better than the AG. The experiments were made on 200 characters of text encrypted with keys contains less than 10 characters.

6.4 Conclusion

In this paper, we presented a comparison of some natural heuristics algorithms. Experimentation data is a set of texts encrypted by various classical algorithms. The synthesis of the tests proved that the PSO algorithms can give better results than other revolutionary algorithms of the same class.

The proposed approach is a modified WSOP algorithm which associates with a control module. It allows to supervise continuously the path of the particles and thus to avoid any blockage in a local minima, challenge difficult to avoid in most cases.

References

[1] Gherboudj, A., Chikhi, S. (2011). BPSO 'Algorithms for Knapsack Problem', CCIS, Springer.

[2] Olamaei, J., Niknam, T., Gharehpetian, G. (2008). Application of particle swarm optimization for distribution feeder reconfiguration considering distributed generators', *AMC*.

[3] Eberhart, R. C., Kennedy, J. (1995). A new optimizer using particles swarm theory,6-th symposium on micro-machine and human science, IEEE.

[4] Eberhart, R., Kennedy, J. (1995). A new optimizer using particle swarm theory, Micro Machine and Human Science. MHS *In*: proceedings of the Sixth International Symposium on, 4 (6) 39-43.

[5] Durán, O., Pérez, L., Batocchio, A. (2011).Optimization of modular structure using Particule Swarm optimization, ESA 2011.

[6] Onwubolu, G. C., Clerc, M. (2004). Optimal operating path for automated drilling operations by a new heuristic approach using particle swarm optimisation", International Journal of Production Research, 42 (3) 473-491.

[7] Lin, C. H. Chen, J. L., Gaing, Z. L. (2010). Combining Biometric Fractal Pattern and Particle Swarm Optimization-Based Classifier for Fingerprint Recognition,MPE.

[8] Parsopoulos, K. E., Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization, Natural Computing 1, 2-3, p. 235-306.

[9] CLERC et SIARRY. (2003). Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essaims particulaires

[10] SHEN, Q., SHI,W., KONG, W. (2007). Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, Computational Biology and Chemistry.

[11] Kennedy, J., Eberhart, R.C., Shi, Y. (2001). *Swarm intelligence*, Morgan Kaufmann Publishers, San francisco.

[12] Gandelli, A., Grimaccia, F., Mussetta, M., Pirinoli, P., Zich, R. E. (2007). Development and Validation of Different Hybridization Strategies between GA and PSO, Proc. of the IEEE Congress on Evolutionary Computation, Sept. , Singapore, p. 2782–2787

[13] Zhao, S. Z., Liang, J. J., Suganthan, P. N., Tasgetiren, M. F. (2008). Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization, in Proceedings IEEE Congress on Evolutionary Computation, p. 3845–3852.

[14] Nalini, N. (2006). Experiments on Cryptanalysing Block Ciphers via Evolutionary Computation Paradigms, M. F., International Conference on Evolutionary Computing, Cavtat, Croatia, June 12-14.

[15] Chen, S. (2009). Locust Swarms – A New Multi-Optima Search Technique, in Proceedings of the IEEE Congress on Evolutionary Computation, p. 1745–1752.

[16] CLERC .(2003). Discrete Particle Swarm Optimization illustrated by the Traveling Salesman Probleme.

[17] Yang, S., Wang, M., Jiao, L. (2004). A Quantum Particle Swarm Optimization, Congress on Evolutionary Computing.

[18] Pampara, G., Franken, N., Engelbrecht, A. P. (2005). Combining Particle Swarm Optimisation with angle modulation to solve binary problems, IEEE Congress on Evolutionary Computing.

[19] Labed, S., Guerboudj, A., Chikhi, S. (2011). A Modified Hybrid Particle swarm Optimization, IJCA.

[20] Mendes, R., Kennedy, J., Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better, IEEE Transactions on Evolutionary Computation.

[21] Bruen, Aiden, A., Forcinito, Mario, A. (2012). Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century, Wiley, J., Sons, ISBN 978-1-118-03138-4.

[22] Gupta, I., al. (2007). Cryptanalysis of an Extension of the Hill Cipher, ACM.

[23] Biryukov, A., al. (2003). A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms, EUROCRYPT.

[24] Zim, Herbert Spencer. (1962). Codes and secret writing, (abridged edition). Scholastic Book Services, fourth printing.

[25] Beker, Henry, Piper, Fred. (1982). Cipher Systems: The Protection of Communications, Book Wiley-Interscience.

[26] Lewand Robert. (2000). Cryptological Mathematics, The Mathematical Association of America.

[27] Nelson, Gerald, Wallis, Sean, A., arts, Bas. (2000). Exploring Natural Language, Working with the British Component of the International Corpus of English.

[28] Beker, Henry, Piper, Fred. (1982). Cipher Systems, The Protection of Communications. Wiley-Interscience. p. 397.

[29] Kak, A. (2012). Block Ciphers and the DES, lecture notes on Computer and network security.

[30] Douglas Selent. (2010). Advanced Encryption Standard, River Academic Journal, 6 (2).

[31] Mahapatra, A., dash, R. Data Encryption And Decryption By Using Hill Cipher Technique And Self Repetitive Matrix, Thesis, National Institute of Technology, Rourkela.

[32] Verma, A. K., Mayank Dave., Joshi, R. C. (2007). Genetic Algorithm and Tabu Search attack on the Mono-Alphabetic Subsitution Cipher in Adhoc networks, Journal of Computer Science.

[33] Xinchao, Z. (2009). A perturbed particle swarm algorithm for numerical optimization, ASC.

[34] Hendtlass, T. (2005). WoSP: A Multi-Optima Particle Swarm Algorithm, *In*: Proceedings IEEE Congress on Evolutionary Computation.

[35] Nelson, Gerald, Wallis, Sean, Bas, A. (2002). Exploring Natural Language. Working with the British Component of the International Corpus of English, John Benjamins Publishing Company.

[36] Mekhaznia, T., Menai, M. B., Zidani, A. (2010). Cryptanalysis of ciphertext substitution using ACO algorithms, ICMOSS10,Algerie.

[37] Mekhaznia, T., Menai, M. B., Zidani, A. (2012). Cryptanalyse du chiffrement par substitution et transposition l'aide d'algorithmes heuristiques,ICIST12, Tunisia.

[38] Verma, A. K., Mayank Dave., Joshi, R. C. (2007). Genetic Algorithm and Tabu Search attack on the Mono-Alphabetic Subsitution Cipher in Adhoc networks, *Journal Of computer Sciences*.

[39] Verma, A. K., Mayank Dave., Joshi, R. C. (2007). Genetic Algorithm and Tabu Search attack on the Mono-Alphabetic Subsitution Cipher in Adhoc networks, *Journal Of computer Sciences*.

[40] Hadi salih, H., Tarik Sadiq, A. (2010). Attack on the Simple Substitution Ciphers Using Particle Swarm Optimization , ETJ.

[41] Sivagurunathan, G., Purusothaman, T. (2011). Reduction of Key Search Space of Vigenere Cipher Using Particle Swarm Optimization, JCS.