Use of MPSO To Break Transposition Cipher System

Mohamed H. Ahmed[1] ,Dr.Salim A.Abbas[2]andAhmed Kareem Shibeeb[3]

[1] *Department of Computer Science., College of Education, Mustansiriyah University, Baghdad, Iraq.*
[2] *Department of Computer Science., College of Education, Mustansiriyah University, Baghdad, Iraq.*
[3] *Department of Computer Systems., Technical Institute-Suwaira, Middle Technical University, Baghdad, Iraq.*
Corresponding Author: Mohamed H. Ahmed

Abstract: *Particle swarm optimization (PSO) based cryptanalysis has acquired much attention because it has fast convergence rate. We investigate the use of Modify Particle Swarm Optimization (MPSO) with some modification in fitness function to break the transposition cipher.MPSO based on using multiple swarm rather than single swarm and exchange information between these swarms to determine the best key. Experimental results of the proposed MPSO appear that the MPSO algorithm reduce the number of tries which are needed to determine the secrete key of the attacked transposition cipher using ciphertext only attack Compared with PSO algorithm. Different parameters are taken for the case study: population size are [20-30] , key size [5-10] and ciphertext [500-10000].*

Keywords*: ciphertext, cryptanalysis, Transposition Cipher, Particle Swarm Optimization Algorithm.*

Date of Submission: 10-10-2018 Date of acceptance: 26-10-2018

I. Introduction

The main problems in cryptography are the evolution of reliable cryptographic system (a cryptography problem) and the search for new effective ways of deciphering existing system (a cryptanalysis problem). A cryptographic approach to secure information presuppose its transformation which enables it to be read only by the authorized person who has the secret key. The authenticity of a cryptographic methods of securing data depends on cryptanalysis immutability of the used system [1].

Cryptanalysis is the process of converting ciphertext to cleartext. It also can be defined as the science of explicating ciphertext . Cryptanalysis assume that the attackers knows the cryptographic system, but the attackers don't know the key or algorithm. A robust method to achieve cryptanalysis on the different cryptographic system using soft computing techniques [2].

This paper focused on using MPSO algorithm to cryptanalysis transposition cipher.

II. Transposition cipher

In transposition systems, the cleartext is left unchanged but rearrange the character's order in such a way that if an unintended recipient get the encryption message and does not know the key then the plaintext would remain unreadable. The main purpose of transposition is to achieve diffusion by dissemination of information to the massage and the key to get out on a large scale across the ciphertext. Transposition cipher also known as a permutation cipher because its rearrange of the characters of the cleartext. A transposition cipher works by breaking a message into fixed size blocks, and then permuting the characters within each block according to the key. The ciphertext in transposition cipher contains all the characters that were in the cleartext, albeit in a different order. In other words, the unigram statistics for the message are unchanged by the encryption process [3].**Example:** Let's have the following Plain text message:

The truth is more important than the fact

1	2	3	4
T	H	E	T
R	U	T	H
I	S	M	O
R	E	I	M
P	O	R	T
A	N	T	T
H	A	N	T
H	E	F	A
C	T	X	X

If the message length is not a multiple of the length of a row, the last columns will be a letter short. An infrequent letter, such as **X** is sometimes used to fill in any short columns.
The size of the permutation is known as the period. For this example, a simple transposition cipher with a period of 4 is used. Let K = (4,2,1,3) be encryption key. Then the message is broken into blocks of 4 characters. Upon encryption the 4th character in the block will be moved to position 1, the 2nd not change, the 1st to position 3 and the 3rd to position 4.

K		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
P	:	T	H	E	T	R	U	T	H	I	S	M	O	R	E	I	M	P	O	R	T
C	:	T	H	t	e	h	u	r	t	o	s	i	m	m	e	r	i	t	o	p	r
K	:	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
P	:	A	N	T	T	H	A	N	T	H	E	F	A	C	T	X	X	X	X	X	X
C	:	T	N	a	t	t	a	h	n	a	e	h	f	x	t	c	x	x	x	x	x

The resulting ciphertext (in lowercase letters) would then be read off as:

Thteh urtos immer itopr tnatt ahnae hfxtc
Notice also that decryption can be achieved by following the same process as encryption using the "inverse" of the encryption permutation. In this case the decryption key , K^{-1} is equal to (3, 2, 4, 1)[4].

III. Problem definition

A simple transposition cipher preserves the number of symbols of a given type within a block, and thus is easily cryptanalysis. Transposition cipher which want to be cryptanalyze by soft computing techniques, encrypts cleartext according to this stages [5] :

1) Key with length L, this key takes the form of switching process of the integers from 1 to L. The cleartext of N symbols is written under the key to form matrix of L characters athwart and at least N mod L symbols depth.
2) The cleartext is then encrypted by reading it in rows according to their order in key sequence.

The cryptanalyze of such transposition cipher usually includes two phases:

1) Found the length of the transposition sequence.
2) Determined the permutation of the L integers.

If the length of the key is up to M integers, then the total potential permutations for the transposition system is P where

$$P(L) = \sum_{M} L!$$

IV. Fitness function

Key is used for encryption and decryption, So the primary goal of cryptanalysis is to get the key in order to obtains the plaintext. Cryptanalysis transposition cipher should get the correct key. To obtains the correct key we must still swapping between some positions of the key until obtains the plaintext.

In this research, cryptanalysis of transposition cipher based on Diagram (DG), Trigram (TG) and Quadgram (QG) frequency of cleartext letters with length L=10000 letters. This frequency called the target frequencies(TF) .

All frequencies for the DG,TG and QG have been calculated with overlap and not calculated for the beginnings and ends of the words.

$$\overline{X}^{j} = \frac{\sum_{i='a'}^{'z'} X_{i}^{j}}{L^{j}} \qquad \dots(1.1)$$

Where:
$\overline{X}^{j}$ is the arithmetic mean of X_{i}^{j} frequency
L is the length of plaintext.
j=Diagram, Trigram, Quadgram.
i='a','b',…,'z'.
L^{d}=L-1, L^{t}=L-2 , L^{q}=L-3

The Total of the Higher Frequencies (THF) for cleartext and ciphertext are calculated using equations (1.2) and (1.3) respectively.

$$\text{THF(M)} = \overline{T}^{d} + \overline{T}^{t} + \overline{T}^{q} \qquad \ldots(1.2)$$

For $T_i^d \geq 0.004$, $T_i^t \geq 0.0022$ and $T_i^q \geq 0.002$

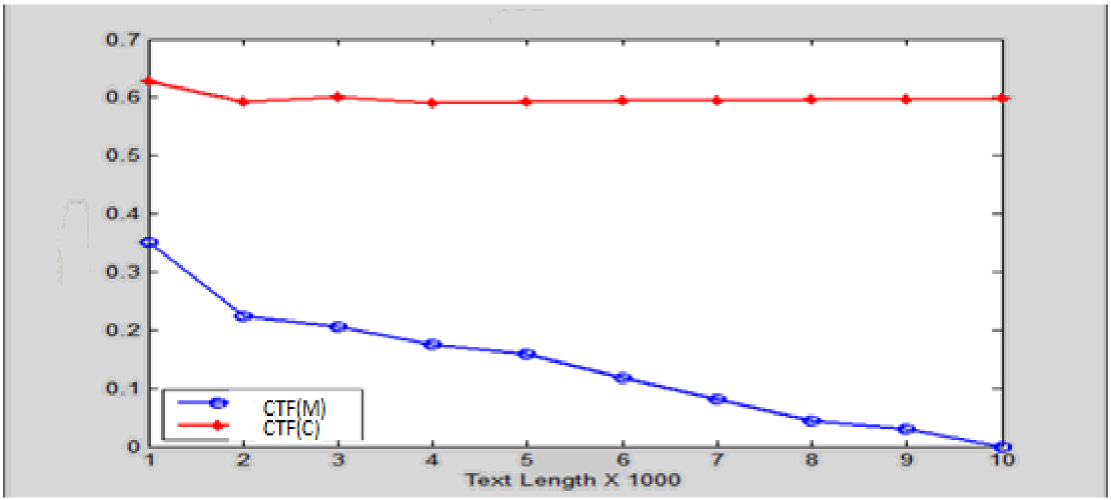
$$\text{THF(C)} = \overline{V}^{d} + \overline{V}^{t} + \overline{V}^{q} \qquad \ldots(1.3)$$

where $\overline{V}^{j}$ is the mean of letters frequency of the ciphertext .

Always the THF(M) is greater than THF(C),its means, in order to decrypt message we must look for the maximum THF. Sometimes the THF(M) value and THF(C) value are closer, so we use Coincidence of Target Frequency(CTF) in order to decipher this similarity.

$$CTF = \sum_{i='aa'}^{'zz'} \left| T_i^d - V_i^d \right| + \sum_{i='aaa'}^{'zzz'} \left| T_i^t - V_i^t \right| + \sum_{i='aaaa'}^{'zzzz'} \left| T_i^q - V_i^q \right| \qquad \ldots (1.4)$$

The CTF(M) values for different text lengths is in Growing constantly, while the CTF(C) values are in steady. Figure (1.1) shows the behavior of CTF function for different L of plaintext and ciphertext.



Figure(1.1): the behavior of CTF function for L=(1,(1),10)×10^3 letters of plaintext and ciphertext.

The determination of fitness function to cryptanalyze transposition cipher can be done by depending on the maximum total of the higher frequency (Max THF) and minimum coincidence of target frequency(Min CTF).

TF (n,σ)={Max THF, Min CTF} …(1.5)

Where TF is the target function ,n is the length of the key and σ=(1,2,…,n)[6].

V. Particle swarm optimization (PSO)

PSO is a heuristic global optimization method, originally provided in1995 by Kennedy and Eberhard. Particle Swarm Optimization (PSO) incorporates amalgamate swarming behaviors observed in schools of fish, swarms of bees, flocks of birds and human social behavior. Since PSO is a population-based optimization tool, it can be implemented and applied easily to solve various optimization problems or the problems that can be transformed to optimization problems. The main power of PSO algorithms is its fast convergence as compared with many global optimization algorithms such that Simulated Annealing (SA), Genetic Algorithms (GA), and other global optimization algorithms. In PSO, There are other features such that no complex mathematical functions, no costly mathematical and doesn't required a large amount of memory [7].

The PSO algorithm depends in its implementation in the following two relations:

$V_i[t+1]=wV_i[t]+C1*r1*(pbest_i[t]-X_i[t])+C2*r2*(gbest_i[t]-X_i[t])$ … (1.6)

$X_i[t+1]=X_i[t] + V_i[t+1]$ … (1.7)

Where:

$V_i[t]$is velocity of particle i at iteration t.

$X_i[t]$ is the position of particle i at iteration t.

$V_i(t+1)$ is velocity of particle i at iteration t+1.

$X_i(t+1)$ is the position of particle i at iteration t+1.
r1,r2 is random number between (0,1).
c1 is cognitive acceleration coefficient.
c2 is social acceleration coefficient.
$pbest_i$ represents the best previous position (the position giving the best fitness value) of the i^{th} particle.
$gbest_i$ represents the index of the best particle among all the particles in the population.

Like the other EA, a PSO algorithm is a population based on search algorithm with random initialization and there is an interaction between population members. In PSO, each particle flies through the problem space and has ability of remember previous best position, outrun from generation to another.

```
Algorithm (1): Particle Swarm Optimization (PSO) algorithm
1- Initialize a population of particles with random positions and velocities on d-dimensions in the problem space.
2- PSO operation includes:
    a. For each particle, evaluate the desired optimization fitness function in dvariables.
    b. Compare particle's fitness evaluation with its $pbest_i$. If current value is better than $pbest_i$, then set $pbest_i$equal to the current value, and $pbest_i$ equals to the current location $x_i$.
    c. Identify the particle in the neighborhood with the best success so far, and assign it index to the $gbest_i$.
    d. Change the velocity and position of the particle according to equations (1.6) and (1.7).
3- Loop to step (2) until a criterion is met.
```

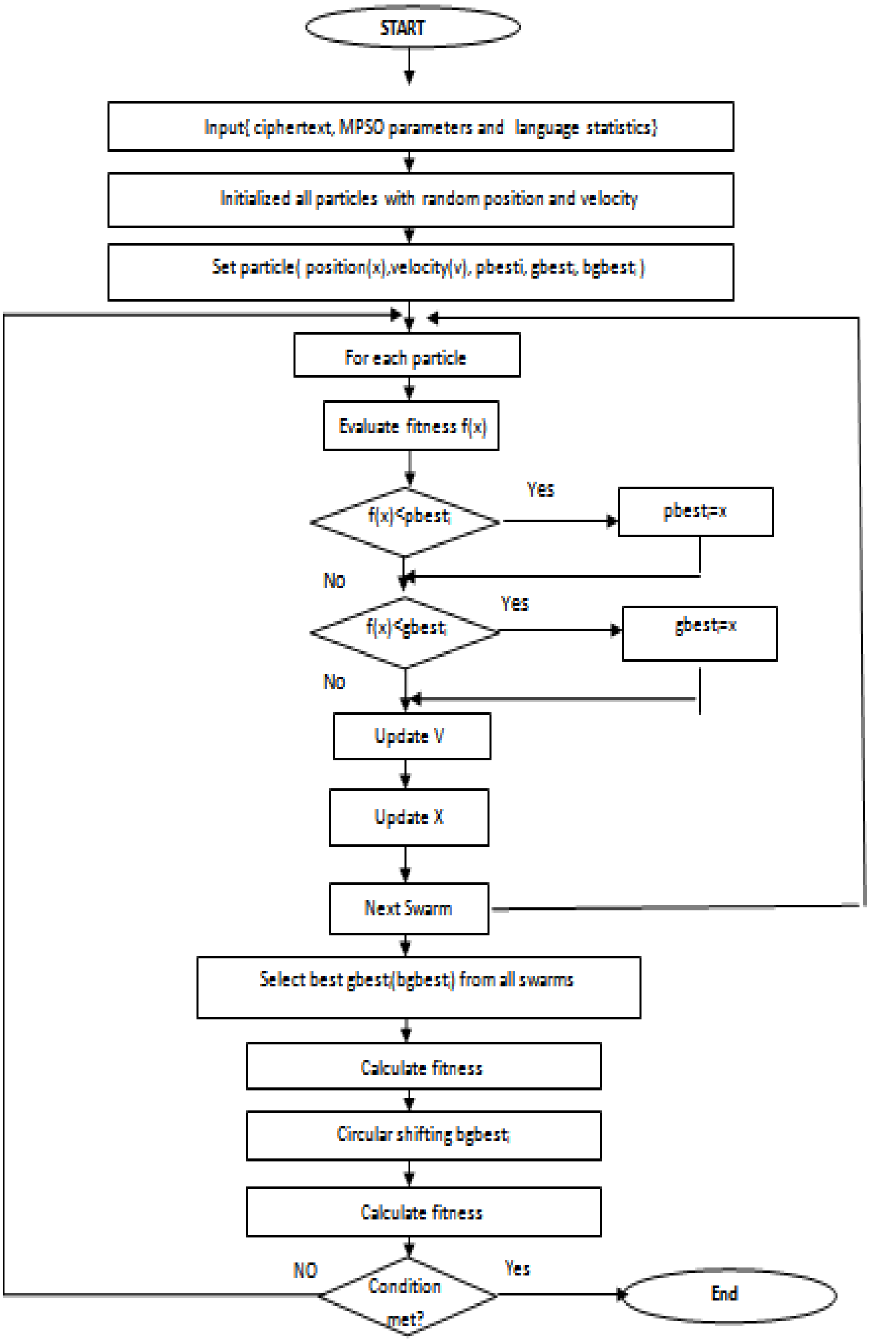
A number of factors will affect the performance of the PSO. These factors are called **PSO parameters**, these parameters are[8]:

1. No. of particles in the swarm which effects on the run time, it must be balanced between the No. of particles and speed.
2. Maximum velocity V_{max} parameter. This parameter limits the maximum jump that a particle can make in one step.
3. Inertia weight w, which it's used to control the impact of the previous history of velocities on the current one.
4. Cognitive and social parameters (c_1 , c_2)are not critical for PSO's convergence. However, proper fine-tuning may result in faster convergence and mitigation of local minima, it better to choose a larger cognitive parameter c_1 than a social parameter c_2 but with limitation to this value $c_1 + c_2 = 4$.
5. r_1 and r_2 which it used to maintain the diversity of the population and it must in the range [0,1].

VI. Use MPSO algorithm to cryptanalyze transposition Cipher

Modify PSO (MPSO) is a relatively new approach to attack classical cipher system. The general structure of MPSO is shown in figure (1.2). The main operations of MPSOare:

- Using multi swarm rather than single swarm. each swarm consist of number of particles; each particle represents solution(key).
- Exchange information between particles in different swarm. These communications between particles leads to determine the best solution in all swarms.
- After determined the best solution, shifting operation is applying to the best solution. This shifting is performs from right to left.


START
Input{ ciphertext, MPSO parameters and language statistics}
Initialized all particles with random position and velocity
Set particle{ position(x),velocity(v), pbesti, gbest, bgbest }
For each particle
Evaluate fitness f(x)
f(x)<pbesti
Yes
pbesti=x
No
f(x)<gbesti
Yes
gbesti=x
No
Update V
Update X
Next Swarm
Select best gbesti(bgbesti) from all swarms
Calculate fitness
Circular shifting bgbesti
Calculate fitness
NO
Condition met?
Yes
End

6.1. Particle Representation and Initial Swarm

MPSO initialized with a population of random solutions and searches for optimal solution by updating the position and velocity for each particle. In each generation, each particle is updated according to two values:

- $pbest_i$: it's the best previous position(fitness) of the i particle achieved so far.
- $gbest_i$: it's the best position(fitness) of the i particle among all the particles in the population.
- $bgbest_{i:}$ it's the best position(fitness) of the i particle among all swarms.

After finding the two best values, the particle updates its velocity and position by the following equations:

$$V_i[t+1]=wV_i[t]+C1*r1*(pbest_i[t]-X_i[t])+C2*r2*(gbest_i[t]-X_i[t]+C3*r3*(bgbest_i[t]-X_i[t]) \quad \ldots(1.8)$$

$$X_i[t]=X_i[t]+V_i[t] \quad \ldots(1.9)$$

the key. For attacking transposition cipher, population of particles is constructed randomly and an integer value assigned for each particle randomly.

Moreover, Velocity worth ($V_i[t]$) bounded to some minimum and maximum values [Vmax, Vmin] where Vmin= -Vmax. This bound reinforces the local search reconnoitering of the problem space. In this work, Vmax is set to 4 and Vmin set to -4.

6.2 Particle Evaluation

At the beginning of each generation, the fitness of each particle must be calculated. Previous section denoted that the $X_i[t]$ is an integer number referred to permutation of the key to specify that it used to decrypt the ciphertext. In MPSO algorithm, candidate key compare to n-gram (DG, TG and QG) statistics of the decrypted message with those of the language. Equation (1.5) is a formula used to determine the suitability of each key.

6.3 MPSO parameters

Table (1) shows the most parameters of PSO that preferred to be used to decrypt transposition cipher.

Table 1 : MPSO parameters

Parameters	Symbol	Value
Number of particles in the swarm	Numparticles	≥ 10
Number of swarms	NumSwarm	≥ 1
Number of Key	Nkey	≥ 5
Length of text	LEN	≥ 500
The maximum number of iterations	MaxIter	≥ 100
The maximum of velocity	Vmax	4
The minimum of velocity	Vmin	-4
Inertia Weight	W	0.4- 0.9
acceleration parameter	C1,C2,C3	0.5-2
Random number	r1,r2,r3	0-1
Circular Shifting	SH	(0,1)

6.4 Working steps of applying MPSO to cryptanalyze transposition cipher cipher

1- **Input** {ciphertext, MPSO parameters and the language statistics}

2- **Initial population** {Generate more than one swarm. each swarm has set of particle; each particle represents possible key}

3- **Survival of the fittest** {Calculate the fitness value of each particle in every swarm according to equation (1.5) , If the fitness value is better than the best fitness value ($pbest_i$) in history ,set current value as the new $pbest_i$}.

4- **Next generation**{Choose the particle with the best fitness value of all the particles in each swarms as $gbest_i$ and choosing the particle that have best $gbest_i$ from all swarms as $bgbest_i$ }.

5- **Sorting** {sort the particles(key) in Progressive order according to their Fitness Values}.

6- **Shifting** {Circular Shifting to the $bgbest_i$ from right to left}.

7- **Calculate the fitness**{Calculate the fitness for $bgbest_i$ according to equation (1.5)}.

8- **Evolve particle** {For each particle:

i. Calculate particle velocity according equation (1.8)

ii. Update particle position according equation (1.9)

1. While maximum iterations or stopping criteria is not hookup, Particles' velocities on each dimension are clamped to Vmax.

2. If the sum of accelerations would cause exceed the velocity to the Vmax, Then the velocity is limited to the Vmax.

9- Terminate the PSO{Terminate when a solution has been found or other termination criteria has been met}.

VII. Experimental results

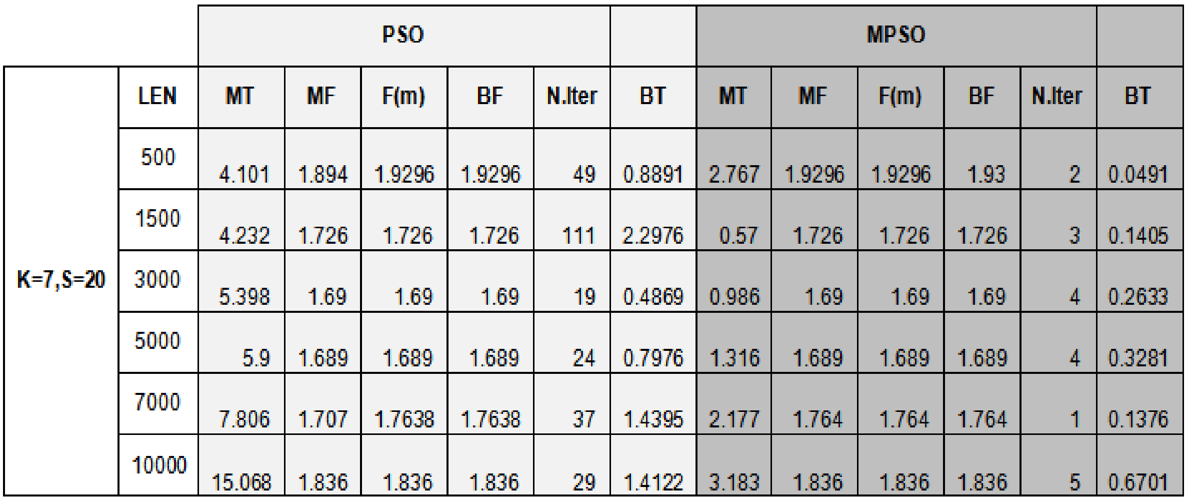
All experiments presented in this paper were performed on text using capital English characters alphabet, i.e. A-Z. All punctuation, space and structure (sentences/paragraphs) has been removed from the text before encryption. The MPSO has been implemented successfully on different size of ciphertext provided to attack. In this section MPSO used to decrypt transposition cipher with key length [5,7,10] and text length [500,1500,3000,5000,7000,10000] with different number of generation and different number of particles. Table (2) illustrated the result of applying MPSO to decrypt message with different size encrypt with key length=5. This algorithms applied to 5 example with the same key length (5). The maximum iteration in this table =500.
Where **K**=NO.Key, **N.S**=NO.Particle or genes, **MT**=Mean of Time, **MF**=Mean of Fitness, **F(m)**=Fitness of plaintext, **BF**=Best Fitness, **BT**=Best Time, **BI**=Best Iteration.

Table (2): Result of applying PSO andMPSO decrypt message with different size using key length=5 .

		PSO						**MPSO**					
	LEN	**MT**	**MF**	**F(m)**	**BF**	**N.Iter**	**BT**	**MT**	**MF**	**F(m)**	**BF**	**N.Iter**	**BT**
K=5,S=20	500	0.215	1.944	1.944	1.944	1	0.002	0.051	1.944	1.944	1.944	1	0.0484
	1500	0.132	1.725	1.725	1.725	1	0.0126	0.09	1.725	1.725	1.725	1	0.0587
	3000	0.162	1.692	1.692	1.692	1	0.0122	0.077	1.692	1.692	1.692	1	0.0738
	5000	0.245	1.688	1.688	1.688	1	0.0238	0.099	1.688	1.688	1.688	1	0.0944
	7000	0.196	1.764	1.764	1.764	1	0.0623	0.117	1.764	1.764	1.764	1	0.1149
	10000	0.299	1.837	1.837	1.837	1	0.0831	0.153	1.837	1.837	1.837	2	0.1495

The results from above table shows that both algorithms PSO and MPSO are found the correct key for the different ciphertext length but the average of time for MPSO is better than PSO. Results showed there is a great affinity between PSO and MPSO to break ciphertext encrypt with key length=5.
Table (3) illustrate the result of applying PSO and MPSO to decrypt message with different size encrypt with key length=7. This algorithms applied to 5 examples with the same key length(7). The maximum iteration in this table =1000.

Table (3): Results of applying PSO and MPSO to decrypt message with different size using key length=7.

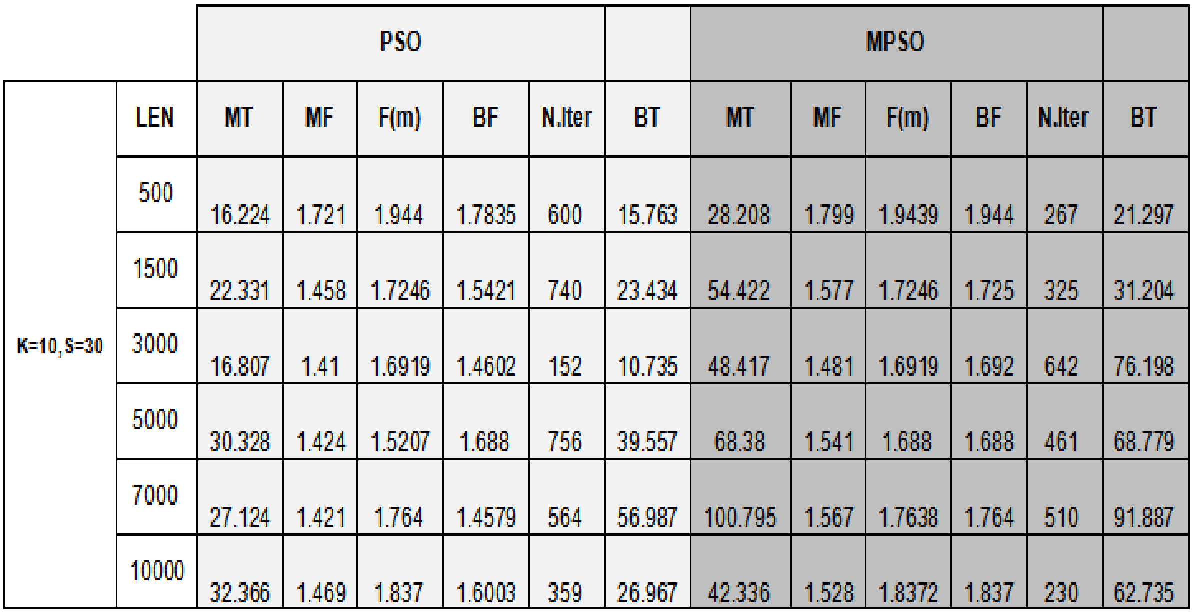


		PSO						**MPSO**					
	LEN	**MT**	**MF**	**F(m)**	**BF**	**N.Iter**	**BT**	**MT**	**MF**	**F(m)**	**BF**	**N.Iter**	**BT**
K=7,S=20	500	4.101	1.894	1.9296	1.9296	49	0.8891	2.767	1.9296	1.9296	1.93	2	0.0491
	1500	4.232	1.726	1.726	1.726	111	2.2976	0.57	1.726	1.726	1.726	3	0.1405
	3000	5.398	1.69	1.69	1.69	19	0.4869	0.986	1.69	1.69	1.69	4	0.2633
	5000	5.9	1.689	1.689	1.689	24	0.7976	1.316	1.689	1.689	1.689	4	0.3281
	7000	7.806	1.707	1.7638	1.7638	37	1.4395	2.177	1.764	1.764	1.764	1	0.1376
	10000	15.068	1.836	1.836	1.836	29	1.4122	3.183	1.836	1.836	1.836	5	0.6701

Also, the results from above table shows that both algorithms PSO and MPSO are found the correct key for the different ciphertext length butthe result showed that the MPSO is most powerful than PSO on the average. For example, MPSO find the correct key for all text length with iteration does not exceed 5 iterations, while PSO iteration arrived to 111 iterations, such that in the length of 1500 character. Also the mean of time in MPSO less than in PSO.

Table (4) shows the mean of time, mean of fitness, number of iteration, best fitness and best time required to break the ciphertext with text length [500-10000] letters and key size (10). The maximum iteration in this table =1000.

Table (4):Result of applying GA and PSO to decrypt message with different size using key length=10.



		PSO						MPSO					
	LEN	MT	MF	F(m)	BF	N.Iter	BT	MT	MF	F(m)	BF	N.Iter	BT
K=10,S=30	500	16.224	1.721	1.944	1.7835	600	15.763	28.208	1.799	1.9439	1.944	267	21.297
	1500	22.331	1.458	1.7246	1.5421	740	23.434	54.422	1.577	1.7246	1.725	325	31.204
	3000	16.807	1.41	1.6919	1.4602	152	10.735	48.417	1.481	1.6919	1.692	642	76.198
	5000	30.328	1.424	1.5207	1.688	756	39.557	68.38	1.541	1.688	1.688	461	68.779
	7000	27.124	1.421	1.764	1.4579	564	56.987	100.795	1.567	1.7638	1.764	510	91.887
	10000	32.366	1.469	1.837	1.6003	359	26.967	42.336	1.528	1.8372	1.837	230	62.735

From the above table,The results showed that MPSO much better than PSO because MPSO found the orrect key when applying to attacks text with length from 500 letters to 10000 letters encrypt with 10 key length, but with the same conditions PSO doesn't find the decrypt key.

VIII. Conclusions

This paper focused on using MPSO in cryptanalysis of transposition cipher. MPSO algorithms are used to find the correct key and recover the plaintext. MPSO algorithm based on new calculation fitness depending on the DG, TG and quadgram (QG) and coincidence of desired frequency. Experimental results show that the MPSO is very efficient to decrypt transposition cipher. MPSO algorithm has less number of possible visited keys required for breaking the ciphertext than the number of possible visited keys required in the Brute Force attack, which is equal to N!

References

[1]. Chris Bourke, "**Cryptography and Computer Security**".University of Nebraska, Lincoln, NE 68588, USA, CSCE 477-877,2015.

[2]. E.C. Laskari, G.C. Meletiou,Y.C. Stamatiou and M.N. Vrahatis,"**Cryptography and Cryptanalysis Through Computational Intelligence**".Springer-Verlag Berlin Heidelberg ,2007.

[3]. William Stallings," **Cryptography and Network Security** ". 5th Edition,2010.

[4]. Bethany Delman,"**Genetic algorithms in cryptography**". Master thesis,Rochester Institute of Technology,2004.

[5]. A.Menezes, P. Van Oorchot and S.Vanstone," **Handbook of applied cryptography**". Boca Raton, CRC Press,1997.

[6]. Faez Hassan Ali, "**Improving Exact and Local Search Algorithms for Solving Some Combinatorial Optimization Problems** ".Ph.D thesis, al-mustansiriya university college of science, department of mathematics,2015.

[7]. Bijaya Ketan Panigrahi,Swagatam Das,Ponnuthurai Nagaratnam Suganthan and Subhransu Sekhar Dash,"**Swarm, Evolutionary and Memetic Computing**".Springer-Verlag Berlin Heidelberg,ISSN 0302-9743, 2010.

[8]. Kennedy J. and Eberhart R. C, "**Particle Swarm Optimization**", Proceedings of IEEE International Conference on NN, Piscataway, pp. 1942-1948, 1995.

Mohamed H.Ahmed. "Use Of Mpso To Break Transposition Cipher System." IOSR Journal of Computer Engineering (IOSR-JCE) 20.5 (2018): 25-32.