# USEs OF GA, PSO and MPSO TO BREAK TRANSPOSITION CIPHER SYSTEM: comparative study

**Dr.Salim A.Abbas* and Mohamed H. Albawi***

**Computer Science. Dept, College of Education, University of Al-Mustansiriya, Baghdad, Iraq.**

## Abstract

GA is an adjustable search method that has the ability for search in smart way to find the best solution and trying to reduce the time that required for obtaining the optimal solution. Particle Swarm Optimization (PSO) algorithm emulate the behavior of a swarm of fish and bird flocks. It's a heuristic global optimization method which can be implemented and applying to solve various optimization problems. The most attractive of using PSO is that it has a fast convergence than the other global optimization methods. Modify PSO (MPSO) is a relatively new approach to attacks transposition cipher which it depends on using multi swarms rather than single swarm and allowing the particles in all swarms to exchange information between them in order to obtains the best solution from all swarms.

This research focuses on use GA, PSO and MPSO to cryptanalyze transposition cipher based on a new tools to determine the fitness function by calculating the Diagram(DG), Trigram(TG) and Quadgram (QG) frequency of letters. It is shown that such algorithms can be used to reduce the number of trails which are needed to determined the initial states of the attacked systems using ciphertext only attack.

Experimental results show the successful applications of GA, PSO and MPSO in cryptanalysis of transposition cipher system. Also, the experimental results indicate that the MPSO is more powerful than the other techniques in cryptanalysis transposition depending on the accuracy of results.

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

## 1.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 knows the key or algorithm. A robust methods to achieve cryptanalysis on the different cryptographic system using soft computing techniques[2].

This paper focused on using GA, PSO and MPSO algorithm to cryptanalysis transposition cipher and make comparative study between them depending on the time consume and accuracy of results.

## 2. Transposition cipher

In transposition systems, the cleartext is left unchanged but rearrange the characters 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].

### *3. 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

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

***4. 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.

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

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 $T_i^q \geq 0.002$

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

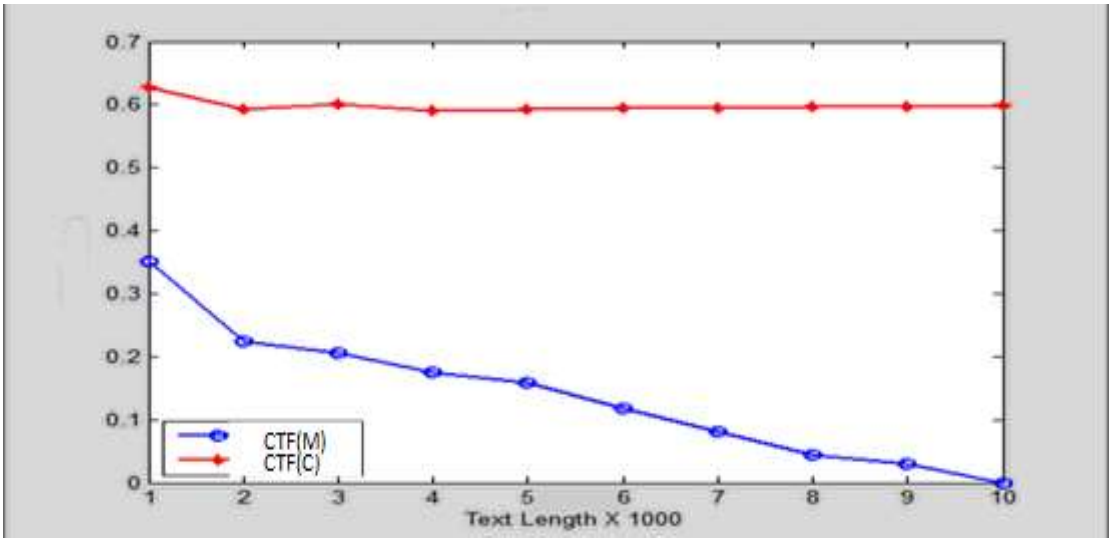
where $\overline{V}^{j}$ is the mean of letters frequency of the ciphertext .

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.

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

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).

$$TF(n,\sigma)=\{\text{Max THF, Min CTF}\} \qquad \ldots(1.5)$$

Where TF is the target function ,n is the length of the key and $\sigma=(1,2,\ldots,n)$[6].

### *5. Use Genetic Algorithm(GA) to cryptanalysis transposition cipher*

GA has been successfully applied to numerous applications in the field of search and optimization. It is recursive procedure that consists of a fixed population size of individuals(chromosomes). These individuals are created randomly or heuristically which represent the initial population. The population evolves by applying three basic operations : selection , crossover and mutation with probability[7].

For the **Initial Population** , The cryptanalysis process begins with randomly generated numbers between{0,1} as the key size for n

chromosomes and sorting these numbers in ascending order. The sequence of these numbers represent the candidate keys (chromosomes). Each chromosome represent the candidate key which it uses to decrypt the ciphertext and then calculate the fitness value to determined the best chromosome(candidate key).

For the **Selection operator**, selects chromosomes in the population for reproduction. The better chromosome has the opportunity to selected more timed to reproduce. Many selection procedures have been proposed, this resarch used Roulette-wheel selection to attack transposition cipher, which it used to selecting potentially useful solutions for reproduction. The chromosome (sequence) with high fitness has a higher probability of participate one or more offspring to the next generation.

For the **Crossover operator**, two chromosomes are combine to produce a new generation that possesses both their characteristic. There are several crossover techniques, this work used arithmetical crossover with probability 1.0 to attack transposition cipher.

For the **Mutation operator**, this process is used to maintain diversity in population from one generation to the next generation in order to obviate local minima .For this study, a simple two point mutation is used. This process uses to select two individuals in chromosome and swap between them. The mutation operator is randomly applied on population if the random number that generated to represent the candidate key is less than 0.1.

The fitness rating helps the cryptanalysis algorithm to achieve attacking transposition cipher . Two, three and four letter combinations (DG, TG and QG) are commonly found in English actually occurs in the decrypted text. It is always possible that GA can award relatively high fitness to chromosome that chance produces relatively high numbers of DG, TG and QG. In this study, equation(1.5) are used to calculate the fitness function of GA to attacks transposition cipher.

For the GA parameters there are a set of values which are considered as the most appropriate to attacks transposition cipher by GA. Table (1.1) shows the different parameters of GA to cryptanalysis transposition cipher.

***Table (1.1) :GA parameters to attack transposition cipher***

| Parameters | Symbol | Value |
|---|---|---|
| Length of key | KeyLen | [5-12] |
| Length of text | TxtLen | [500-10000] |
| Number of chromosomes | Pop_size | [10 – 80] |
| Maximum number of Generation | Gen | [1000-2000] |
| Probability of crossover | Pc | 1.0 |
| Probability of mutation | Pm | < 0.1 |

### *5.1 Working steps of applying GA to attack transposition cipher*

1- **Input** {ciphertext, GA parameters and the language statistics(Digrams, Trigrams and Quardagram)}
2- **Initial population** {Generate set of chromosomes randomly, each one of them represent candidate key}
3- **Decryption** {Decrypt the ciphertext with each candidate key by reading it off in columns in the order dictated by the integers making up the candidate keys}
4- **Fitness Assessment** {Calculate the fitness value of each chromosome (key) according to equation (1.5)}
5- **Sorting** { Sorting the chromosomes (key) in ascending order based on their fitness}
6- **Generate a new population** {Generates a new population by repeating the following steps :
   a. Select parents (candidate keys) from current population according to their fitness value using Roulette-wheel selection.
   b. Applying crossover operation for each parent in population to form offspring.
   c. Applying mutation operation on the offspring with low probability.
   d. Place all new offspring in the new population.
7- **Decryption** {Decrypt the ciphertext with each new candidate keys}
8- **Fitness Assessment** {Calculate the fitness value of new chromosomes (key) according to equation (1.5)}
9- **Sorting** { Sorting the new chromosomes (key) in ascending order based on their fitness}

**10- Evolve generation** { evolve generation by repeating step 6,7,8,9, and 10 for each generation}.

**11- Terminate the GA** { Terminate when termination criteria has been met}.

**12- Save and exit**{ save the chromosome(key) that have the highest fitness as best solution and exit.

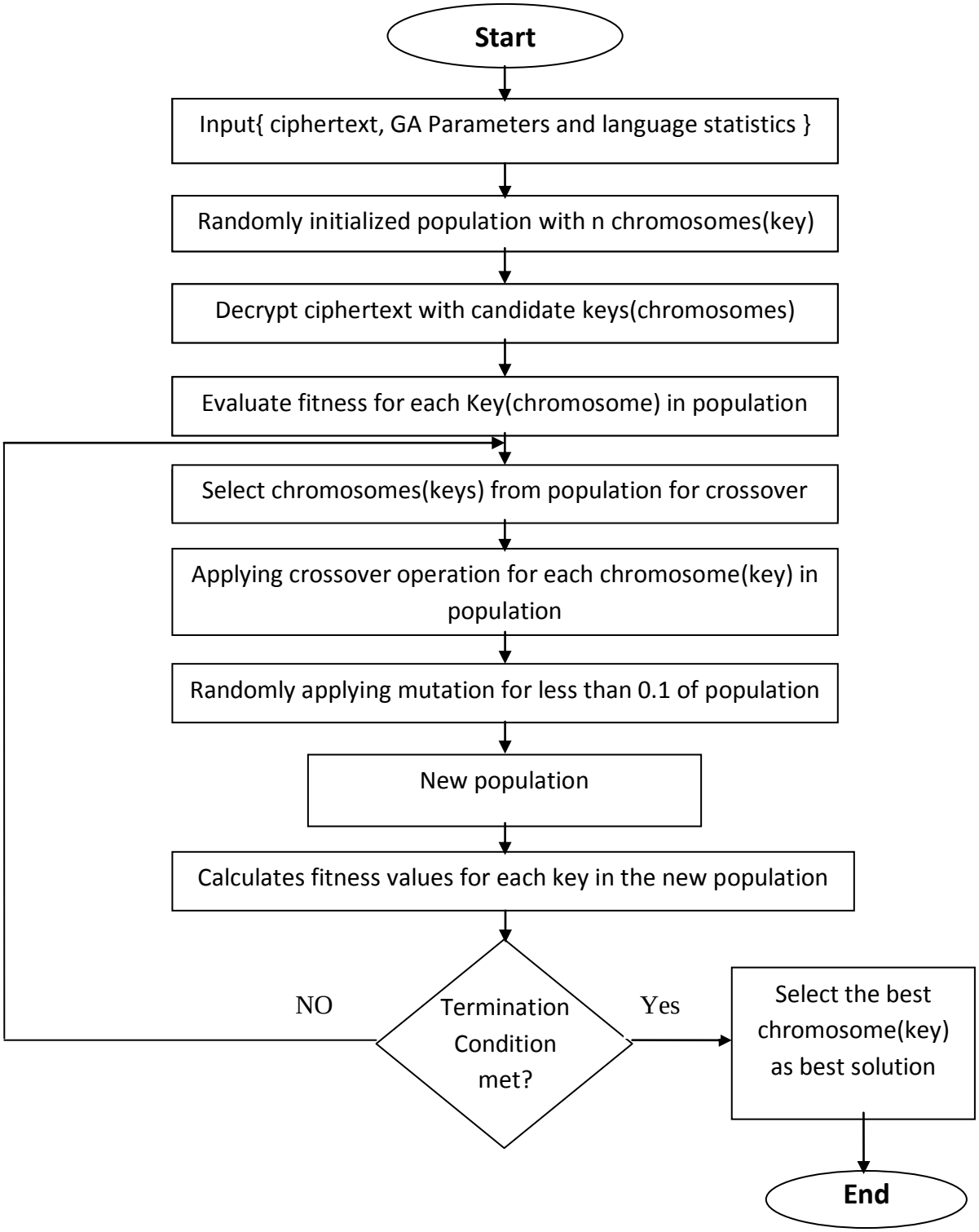



***Figure (1.2):Flow chart of applying GA to break transposition cipher***

## 6. Use PSO algorithm to cryptanalysis transposition Cipher

PSO is a population based stochastic optimization technique which could be implemented and applied easily to solve various optimization problems. It's part of swarm intelligence, So its took inspiration from the Biological. The main advantages of PSO are its easy to implement and there are few parameters to adjust[8].

### 6.1 Particle representation, Initial population and particles evaluation

For the initial population, The cryptanalysis process begins with randomly generated numbers between{1,-1} as the key size for n particles and sorting these numbers in ascending order($X_i[t]$). The sequence of these numbers represent the candidate keys (particles). Randomly generates Velocity worth ($V_i[t]$) for each particle which it's bounded to some minimum and maximum values [Vmax, Vmin] where Vmin= -Vmax and it uses to reinforces the local search reconnoitering of the problem space. In this work, Vmax is set to 4 and Vmin set to -4 .

Each particle represent the candidate key and use to decrypt the ciphertext and then calculate the fitness value to determined the best particle(key).

To evolve solutions(keys), each particle is updated according to two values:

- $pbest_i$: which it's the best previous position for i particles achieved so far.
- $gbest_i$: which it's the best position of the i particles among all particles in swarm(population).

After finding the two best values, the particle updates its velocity and position according to two equations:

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

$$X_i[t+1]=X_i[t] + V_i[t+1] \quad \dots (1.7)$$

For the evaluation process, The fitness value for each particle(candidate key) must be calculated for each generation. In this work, the candidates key(particles) are 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.

### *6.2 PSO parameters*

Table (1.2) shows the most parameters of PSO that preferred to be used to decrypt transposition cipher.

***Table (1.2) : PSO parameters to attack transposition cipher***

| Parameters | Symbol | Value |
|---|---|---|
| Number of particles in the swarm | Pop-size | [10-80] |
| Number of Key | KeyLen | [5-12] |
| Length of text | TxtLen | [500-1000] |
| The maximum number of generation | Gen | [1000-2000] |
| The maximum of velocity | Vmax | 4 |
| The minimum of velocity | Vmin | -4 |
| Inertia Weight | W | [0.4- 0.9] |
| Acceleration parameter | C1,C2 | [0.5-2] |
| Random number | r1,r2 | [0-1] |

## 6.3 Working steps of applying PSO algorithm to attack transposition cipher

1- **Input** {ciphertext, PSO parameters and the language statistics(Digrams, Trigrams and Quardagrams)}
2- **Initial population** {Generate set of particles (candidate key) randomly, each particle represent candidate key } .
3- **Decryption** {Decrypt the ciphertext with each candidate key(particle)}.
4- **Fitness Assessment** {Calculate the fitness value for each particle (candidate key) according to equation (1.5), If the fitness value of the current position is better than the best fitness value ($pbest_i$) in history ,then set the current position as the new $pbest_i$ and Choose the particle(candidate(key) that has the best fitness value from all the particles in swarm as the $gbest_i$}.
5- **Sorting** { Sorting the particles (keys) in ascending order based on their fitness}.
6- **Evolve particles** { For each particle :
   i. Calculate particle velocity according to equation (1.6):
      1. If the particle velocity exceed the Vmax, Then the particle velocity is limited to the Vmax.
      2. If the particle velocity exceed the Vmin, Then the particle velocity is limited to the Vmin.
   ii. Update particle position according to equation (1.7)
7- **Evolve generation** { evolve generation by repeating step 3,4,5,6, and 7 for each generation}.

8- **Terminate the PSO** { Terminate when termination criteria has been met}.
9- **Save and exit**{ save the particle (key) that have the highest fitness and exit}

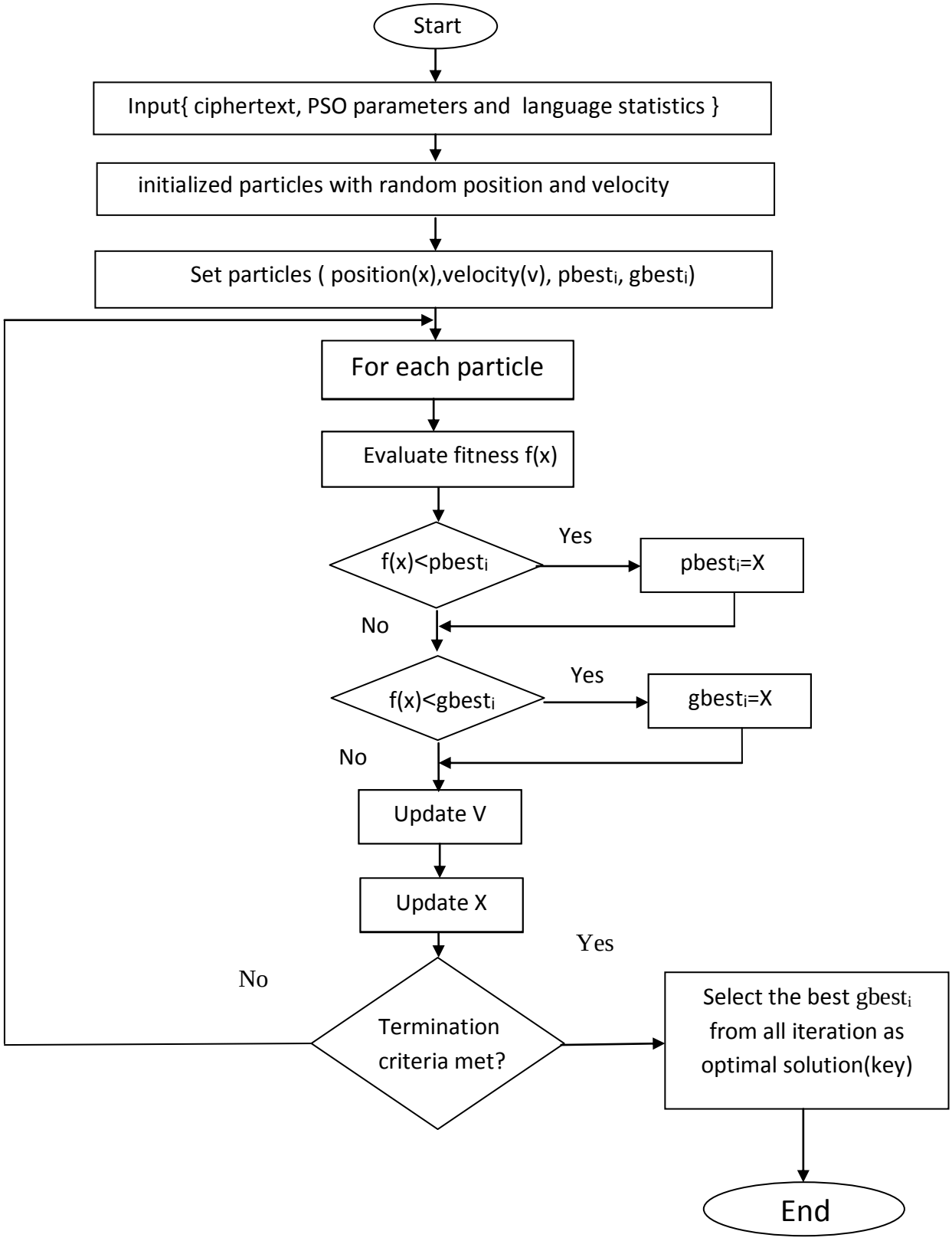



**Figure (1.3):Flowchart of applying PSO algorithm to break transposition**

**7. Use MPSO algorithm to cryptanalysis transposition Cipher**

Modify PSO (MPSO) is a relatively new approach to attack classical cipher system. The main operations of MPSO are :

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

**7.1 Particle Representation, Initial Swarm and evaluation**

For the initial population and particle representation, MPSO uses the same processes of PSO algorithm that are described in section(6.1), But in MPSO, population consist of more than one swarm, So the initial population performed for all swarms.

Each particle in different swarms represent the candidate key and use to decrypt the ciphertext and then calculate the fitness value to determined the best particle(key).

To evolve particles(keys), each particle is updated according to three values:

1. $pbest_i$: it's the best previous position of the particle i achieved so far.
2. $gbest_i$: it's the best position of the particle i among all particles in each swarm .
3. $bgbest_i$: it's the best position of the particle i among all swarms.

After finding $pbest_i$, $gbest_i$ and bgbesti, each particle update its velocity and position by the following equations:

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

$$X_i[t+1] = X_i[t] + V_i[t] \quad \ldots(1.9)$$

***7.3 MPSO parameters***

Table (1.3) shows the most parameters of PSO that preferred to be used to decrypt transposition cipher.

***Table (1.3) : MPSO parameters***

| Parameters | Symbol | Value |
|---|---|---|
| Number of particles in the swarm | Pop-size | [10-80] |
| Number of swarms | MJ | ≥ 1 |
| Number of KeyS | KeyLen | [5-12] |
| Length of text | TxtLen | [500-10000] |
| The maximum number of generation | Gen | [1000-2000] |
| 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] |
| Shifting | SH | (0,1) |

## 7.4 Working steps of applying MPSO to attack transposition cipher

1- **Input** {ciphertext, PSO parameters and the language statistics(Diagrams, Trigrams and Quardagrams)}
2- **Initial population** {Generate more than one swarm, for each swarm generate a set of particles (candidate key) randomly} .
3- **Decryption** {Decrypt the ciphertext with each candidate key(particle) from all swarms }
4- **Fitness Assessment** {Calculate the fitness value for each particle (candidate key) in different swarms according to equation (1.5), If the fitness value for the current position of the particle is better than the best fitness value ($pbest_i$) of the same particle in history ,then set the current position as the new $pbest_i$ and Choose the

particle(candidate(key) that has the best fitness value from all particles in single swarm as the $gbest_i$ and choose the particle that has the best $gbest_i$ from all swarms as $bgbest_i$ }

5- **Sorting** { For each swarm, Sorting the particles (keys) in ascending order based on their fitness }.

6- **Shifting** {Applying circular shifting operation to the $bgbest_i$ }.

7- **Decryption** {Decrypt the ciphertext with $bgbest_i$ }.

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

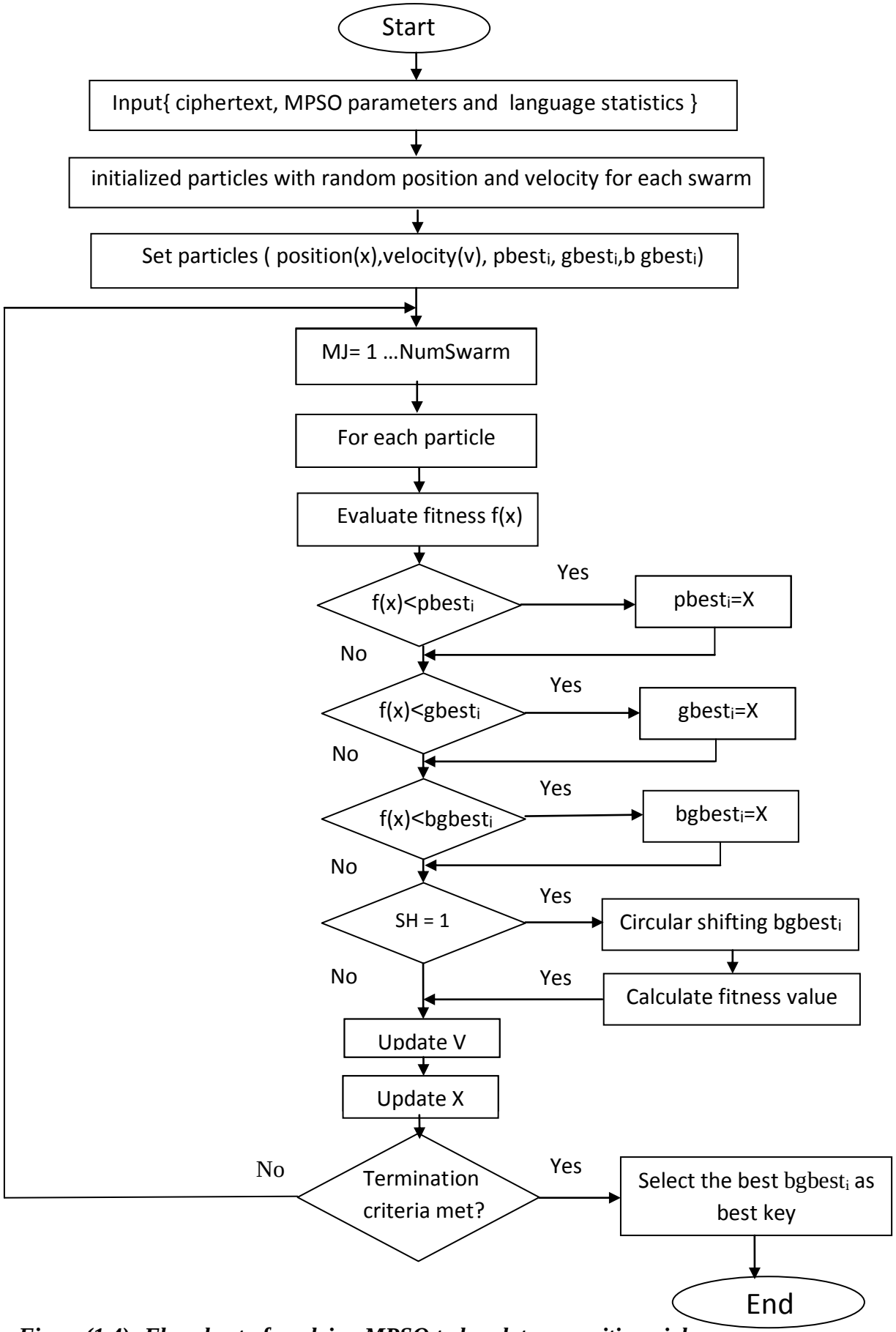
9- **Evolve particles** { For each particle :

   i. Calculate particle velocity according to equation (1.8):
      1. If the particle velocity exceed the Vmax, Then the particle velocity is limited to the Vmax.
      2. If the particle velocity exceed the Vmin, Then the particle velocity is limited to the Vmin.
   ii. Update particle position according to equation (1.9)

10- **Evolve generation** { evolve generation by repeating step 3,4,5,6,7,8,9 and 10 for each generation}.

11- **Terminate the MPSO** { Terminate when termination criteria has been met}.

12- **Save and exit**{ save the particle (key) that have the highest fitness value from all swarms and exit}.




***Figure(1.4): Flowchart of applying MPSO to break transposition cipher***

### *8. 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 three algorithms have been implemented successfully on different size of ciphertext provided to attack. In this section GA, PSO and MPSO are used to decrypt transposition cipher with key length [5,7,10,12] and text length [500,1500,3000,5000,7000,10000] with different number of generation and different number of particles. Table (1.4) show the average time(AvT), average of the best fitness(AvF) and best fitness(BF) of the implementation of GA ,PSO and MPSO algorithm to decrypt message with different size encrypted by using key length=5. These algorithms applied to 5 case study with maximum number of generation =500 and population size=10.

**Table (1.4): The comparison results between GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=5**

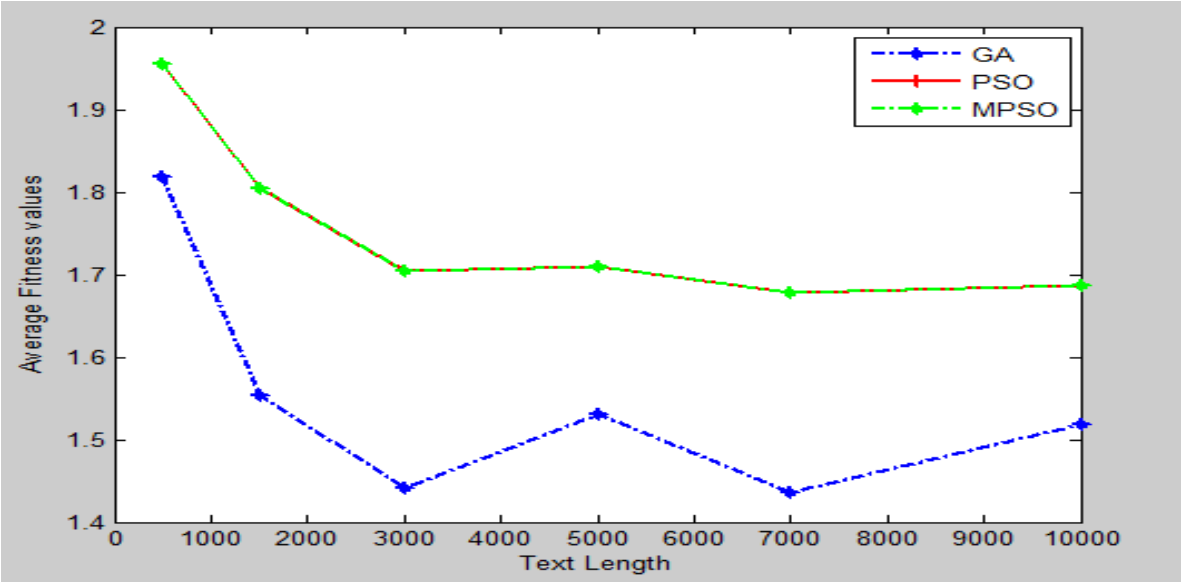
| TxtLen | AvT | | | AvF | | | BF | | |
|---|---|---|---|---|---|---|---|---|---|
| | GA | PSO | MPSO | GA | PSO | MPSO | GA | PSO | MPSO |
| 500 | 0.45 | 0.165 | 0.052 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 |
| 1500 | 2.862 | 0.109 | 0.068 | 1.805 | 1.805 | 1.8046 | 1.8046 | 1.805 | 1.8046 |
| 3000 | 1.827 | 0.101 | 0.056 | 1.634 | 1.705 | 1.705 | 1.7052 | 1,705 | 1.705 |
| 5000 | 3.948 | 0.348 | 0.051 | 1.639 | 1.709 | 1,709 | 1.7091 | 1.709 | 1,709 |
| 7000 | 2.531 | 0.388 | 0.116 | 1.6776 | 1.678 | 1.6776 | 1.6776 | 1.678 | 1.6776 |
| 10000 | 4.081 | 0.258 | 0.074 | 1.687 | 1.687 | 1.6874 | 1.6874 | 1.687 | 1.6874 |

Table (1.5) illustrate the results of applying GA, PSO and MPSO to decrypt ciphertext with different size encrypt with key length=7,maximum number of generation =1000 and population size=30.

***Table (1.5): The comparison results between GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=7***

| TxtLen | AvT | | | AvF | | | BF | | |
|---|---|---|---|---|---|---|---|---|---|
| | GA | PSO | MPSO | GA | PSO | MPSO | GA | PSO | MPSO |
| 500 | 4.779 | 4.424 | 0.688 | 1.819 | 1.955 | 1.955 | 1.955 | 1.955 | 1.955 |
| 1500 | 16.46 | 14.64 | 0.758 | 1.553 | 1.805 | 1.805 | 1.802 | 1.805 | 1.805 |
| 3000 | 5.896 | 6.88 | 0.775 | 1.442 | 1.705 | 1.705 | 1.7054 | 1.705 | 1.705 |
| 5000 | 20.17 | 10.23 | 3.286 | 1.53 | 1.709 | 1.709 | 1.709 | 1.709 | 1.709 |
| 7000 | 21.59 | 231.6 | 1.817 | 1.436 | 1.678 | 1.678 | 1.677 | 1.678 | 1.678 |
| 10000 | 29.62 | 16.37 | 2.739 | 1.519 | 1.687 | 1.687 | 1.687 | 1.687 | 1.687 |

Figure (1.5) show the growth of average fitness value for GA,PSO and MPSO in cryptanalysis message with different size encrypted using key length=7.




***Figure(1.5): The growth of average fitness value of GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=7.***

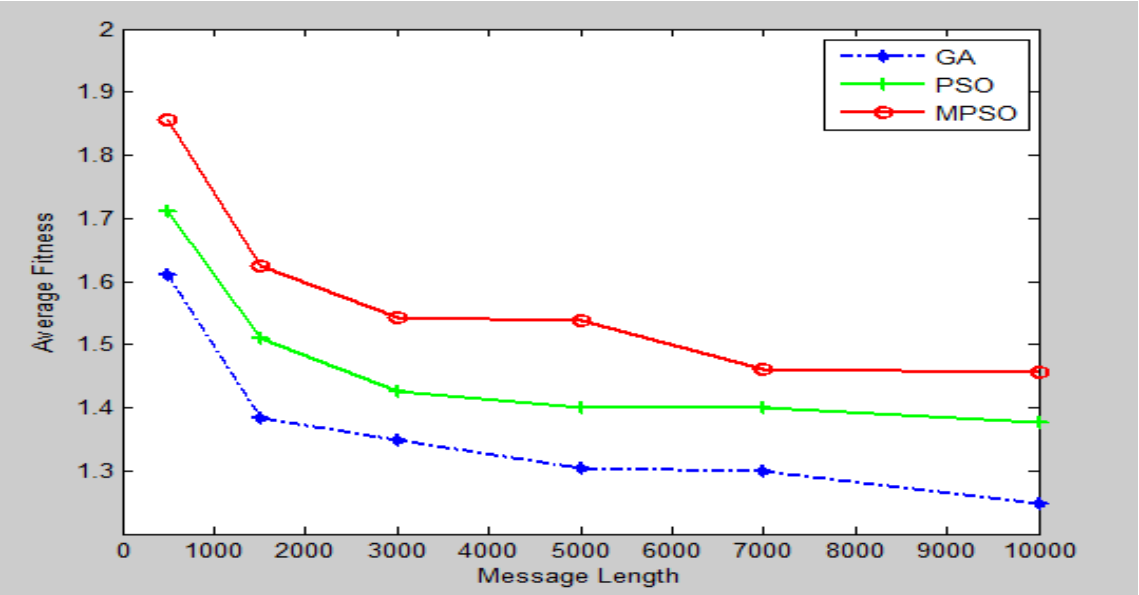
From tables(1.4,1.5), notice that the GA,PSO and MPSO have little differences in obtaining good results, Since all of these algorithms gave excellent results, but MPSO algorithm characterized in average time(AvT). Results showe there is a great affinity to GA, PSO and MPSO to break ciphertext encrypt with key length=5.

Table (1.6) show the resultS of applying GA,PSO and MPSO to decrypt ciphertext with different size encrypt with key length=10, maximum number of generation=2000 and population size =40.

**Table (1.6): The comparison results between GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=10**

| TxtLen | AvT | | | AvF | | | BF | | |
|---|---|---|---|---|---|---|---|---|---|
| | GA | PSO | MPSO | GA | PSO | MPSO | GA | PSO | MPSO |
| 500 | 17.11 | 16.99 | 69.73 | 1.611 | 1.712 | 1.856 | 1.7 | 1.955 | 1.955 |
| 1500 | 13.84 | 19.2 | 62.02 | 1.384 | 1.51 | 1.624 | 1.457 | 1.536 | 1.805 |
| 3000 | 20.59 | 25.14 | 69.43 | 1.348 | 1.425 | 1.542 | 1.39 | 1.523 | 1.705 |
| 5000 | 23.82 | 44.41 | 80.16 | 1.303 | 1.4 | 1.537 | 1.379 | 1.498 | 1.709 |
| 7000 | 36.04 | 33.54 | 83.75 | 1.3 | 1.401 | 1.46 | 1.33 | 1.425 | 1.678 |
| 10000 | 45.29 | 50.36 | 92.33 | 1.247 | 1.376 | 1.455 | 1.283 | 1.404 | 1.687 |

Figures (1.6) show the growth of average fitness value for GA,PSO and MPSO in cryptanalysis message with different size encrypted using key length=10.



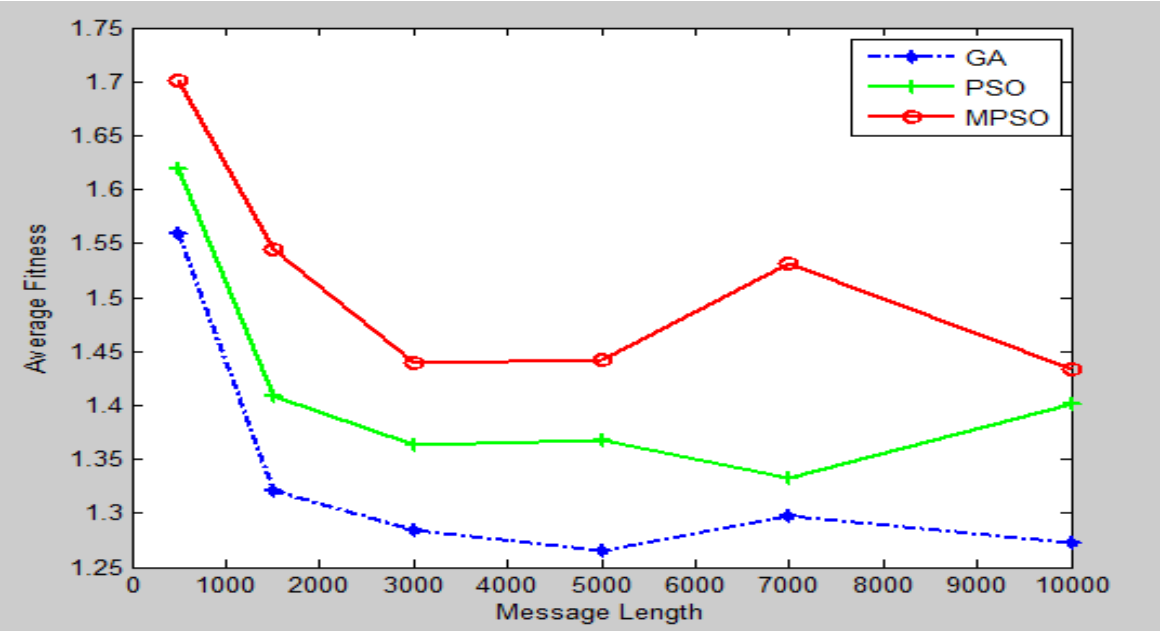

***Figure(1.6): The growth of average fitness value of GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=10.***

Table (1.7) show the results of applying GA,PSO and MPSO to decrypt ciphertext with different size encrypt with key length=12, maximum number of generation=2000 and population size =80.

***Table (1.7): The comparison results between GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=12***

| TxtLen | AvT | | | AvF | | | BF | | |
|---|---|---|---|---|---|---|---|---|---|
| | GA | PSO | MPSO | GA | PSO | MPSO | GA | PSO | MPSO |
| 500 | 44.6 | 94.71 | 176.6 | 1.559 | 1.619 | 1.701 | 1.606 | 1.632 | 1.742 |
| 1500 | 75.47 | 105.2 | 255.1 | 1.321 | 1.409 | 1.544 | 1.382 | 1.438 | 1.695 |
| 3000 | 55.19 | 213.6 | 347.7 | 1.285 | 1.363 | 1.439 | 1.36 | 1.43 | 1.482 |
| 5000 | 168.9 | 315.5 | 406.4 | 1.266 | 1.368 | 1.442 | 1.355 | 1.435 | 1.442 |
| 7000 | 117.6 | 440.3 | 523.4 | 1.298 | 1.332 | 1.531 | 1.331 | 1.339 | 1.771 |
| 10000 | 96.17 | 373.2 | 798.1 | 1.272 | 1.402 | 1.433 | 1.34 | 1.518 | 1.584 |

Figures (1.7) show the growth of average fitness value for GA,PSO and MPSO in cryptanalysis message with different size encrypted using key length=12.




***Figure(1.7): The growth of average fitness value of GA,PSO and MPSO in cryptanalysis message with different size encrypted by using key length=12.***

From tables(1.6 , 1.7) and figures(1.6, 1.7), It's clear that the performance of PSO and MPSO are better than the performance of GA because the fitness values of applying PSO and MPSO in cryptanalysis are better than the fitness values of the GA, but it takes more time than GA. Also, From previous experiments can be observed that the performance of MPSO algorithm is better than PSO algorithm. Where the average fitness values and best fitness values of MPSO algorithm is better than GA and PSO in most experiment, But as expected, the time consumption by MPSO algorithm is larger than the rest algorithms.

## *9. Conclusions*

This paper focused on using GA, PSO and MPSO in cryptanalysis of transposition cipher. From this work, several conclusions can be drawn from applying these algorithms in cryptanalysis transposition cipher systems.

1. It is pointed out that the analysis of the proposed methods only valid for transposition cipher systems .
2. Our original goal was clearly met. The GA proved highly successfully in cryptanalysis of transposition cipher where the key length less than 8 while the PSO and MPSO proved highly successfully in cryptanalysis transposition cipher for key length greater than 8.

3. The accuracy of the recover keys in transposition cryptanalysis using MPSO is better than the other techniques.
4. The performance of GA is less than the performance of the other two techniques.
5. The time consuming in cryptanalysis of transposition cipher systems based on GA and PSO are less than MPSO for most cases.

## 9. 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] Mitchell Melanie," An Introduction to Genetic Algorithms". Cambridge, MIT Press, Fifth printing, 1999.

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