# New Symmetrical Ciphering Approach Based on Memetic Algorithm

Z.Kaddouri[1], F.Omary[2]
[1, 2] Laboratory of computer science research
Department of Computer Science, Mohammed V University – Faculty of Sciences Rabat- Morocco.
[1] kaddouri.zakaria@gmail.com
[2] omary@fsr.ac.ma

Abstract— **Memetic algorithms have demonstrated their effectiveness to provide satisfactory solutions to combinatorial optimization problems reputedly hard.**

**In this paper, we present a new encryption system whose internal structure is essentially based on memetic algorithms. Our approach is based on the hybridization between an evolutionary algorithm based on solutions population and a local search adapted to the problem. The two methods are complementary, because the evolutionary algorithm can well sweep the search space, while the local search allows much more research in these areas to find the best solutions.**

**First, we will bring back the problem of encryption to a combinatorial optimization problem as in the Symmetric Encryption Evolutionary SEC. Then we will encode this problem in a specific way to bring us back to scheduling problems. Finally, after building the lists containing the different positions of the characters of the plaintext, we apply the memetic process on the order of these lists for maximum disorder. The performance criteria considered are the execution time and the convergence of the system. To validate the results found, we conducted a comparison to those found by the evolutionary algorithm.**

**Keyword-** Symmetric encryption, Evolutionary algorithm, Hybridization, Memetic algorithms, Scheduling problem, Combinatorial optimization

## I. INTRODUCTION

Traditionally, cryptography provides a means of communicating sensitive information (secret, confidential or private) while making them unintelligible to everyone except for the message recipient. Classical cryptography was a technique that replaces the text manually to be encrypted in order to conceal the original content.

Modern cryptography is based mainly on mathematical and algorithmic concepts [21, 4]. Its purpose is to study methods to ensure a number of security services; integrity, authenticity and confidentiality in information and communication systems [23, 12, 24]. This helps to protect privacy while sending data from sender to receiver [22].

Metaheuristics have optimization methods to solve difficult optimization problems (often from the fields of artificial intelligence, operations research and engineering) in which no classical method is more effective [20]. Recently, the hybridization is a trend observed in many researches conducted on metaheuristics. It takes advantage of the combined benefits of different metaheuristics [7, 14, 15].

Memetic algorithms represent a hybridization between an evolutionary algorithm and a local search method. They were first introduced by Moscato[14]. After, they become very widespread, because the best results obtained by metaheuristics for several combinatorial optimization problems have been found with hybrid algorithms [1, 5].

Generally, evolutionary algorithms are very effective for NP-hard problems, however they are very heavy and too greedy on the computation time [1], hence the idea of hybridization of our evolutionary algorithm by hill climbing which is faster compared to the conventional algorithm.

In this article, we developed a symmetric encryption system. This design is essentially based on the memetic algorithms. For this reason, we have transformed the encryption problem to a combinatorial optimization problem and by well-defined operations; we have succeeded in implementing our memetic algorithm.

This article revolves around four sections. The first is a general description of memetic algorithms. In the second we present a detailed description of our encryption algorithm Symmetic Memetic Ciphering (SMC). The last section is devoted to the results found and their interpretations.

## II. MEMETIC ALGORITHMS

### *A. Definition*

The memetic algorithms were introduced by Dawkins and formalized by Moscato (Dawkins 89, 89 Moscato, Moscato 99) [14, 15]. Their general idea is to hybridize a local search algorithm with a genetic algorithm. They are also called hybrid genetic algorithms and local search hybrids.

*B. Principle of Memetic algorithms*

Indeed, the strength of the genetic algorithm comes from the fact that they are able to sweep globally solutions space unlike other heuristics which are exploring a small area of the space [15]. A disadvantage of a genetic algorithm is that the genetic operators do not allow intensifying the search process in a sufficient way.

The general principle of memetic algorithms is meant to hybridize a local search algorithm with a genetic algorithm. These two methods are complementary because one can detect good regions in the search space while the other focuses intensively to explore these areas of the search space [13]. For this reason the genetic algorithms are often hybridized with local search methods [6]. With this principle we can quickly explore interesting areas of the search space to exploit them in detail [2].

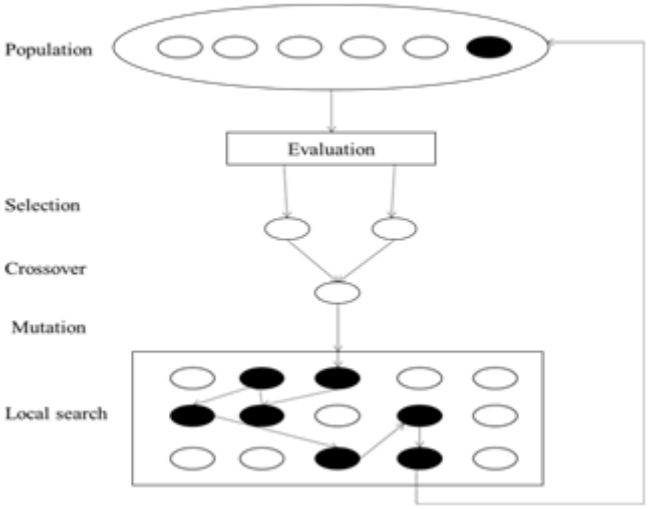


Fig. 1. Diagram of one iteration of memetic algorithms (MA)

## III. DESCRIPTION OF OUR ENCRYPTION ALGORITHM

*A. Mapping the problem*

Let $M$ the binary encoding of message $M_0$:

The message to be encrypted is represented by the Lists, which are elements of the partition of the set *{1, 2, ..., n}*. Each list is made up of the different positions of each binary block.

Let $B_1, B_2, ..., B_n$ of the various blocks $M$.

Denote by $L_i$ $(1 \leq i \leq m)$ a list of the different positions of the block $B_i$ and $card\ (L_i)$ the number of occurrences of $B_i$.

**Note:** This decomposition can be applied only for large messages.

We have $L_i \cap L_j = \emptyset$ *if* $i \neq j,\ \forall i, j \in \{1, 2, ..., m\}$.

Our goal is to create maximum disorder in the positions of characters. For this purpose, we permute iteratively distribution of lists $L_i$ $(1 \leq i \leq m)$ on different blocks of $B$ such that the difference between the cardinal of the new list assigned to each block $B_i$ and cardinal of the original list $L_i$ is maximal. This is an optimization problem and in our case we will appeal to the memetic algorithm to solve it.

*B. Skeleton of the algorithm*

Thereafter we present our adaptation of the memetic algorithm for solving our problem.

**Step 0:** Define a suitable encoding to the problem

**Step 1:** generate an initial population $P_0$ of $q$ individuals $\{X_1, X_2, ..., X_q\}$

*i: = 0;*

Apply the local search procedure on each individual in the population

**Step 2:** Evaluation of individuals.

Let $F$ the evaluation function. Apply $F\ (X_i)$ for all individuals $X_i$ of $P_i$

**Step 3:** Selection of the best individuals

Select the best individuals and group them by pairs.

**Step 4:** Application of genetic operators and the local search.

**1**- Crossover: Apply the crossover operation to pairs selected in **step 3**.

**2**- Mutation: Apply the mutation to individuals from the crossover.

**3**- Local-Search: Apply a local search procedure to individuals from **1** and **2**.

Store new individuals obtained (**1**, **2** and **3**) in a new generation $P_{i+1}$

This process is repeated (**steps 2**, **3** and **4**) until a stop criterion is satisfied.

The local search procedure in the memetic algorithm is performed on all individuals of the initial population, after it is applied to each new individual from the crossover and mutation operator. In other words, we always try to choose the best individuals of the initial population or of the generation built.

In the memetic process we may use a simple method of local search, such as the hill climbing [10, 11] or more complicated methods, such as simulated annealing or Tabu search. In our work, we applied the hill climbing method.

It is an iterative algorithm that starts with an arbitrary solution of a problem and then tries to find the best solution by changing iteratively the current solution.

We can decide whether to consider all solutions in the neighborhood and make the best of all, or to consider a subset of the neighborhood.

We can represent the hill climbing method as follows:

**Step 1:** Generate an initial solution (initialization).

**Step 2:** Generate a list of candidate moves.

**Step 3:** Choose top performer candidate.

**Step 4:** Apply the stopping criterion.

- Continue: Go to step 2 if an improvement is possible
- Stop: select the optimum solution

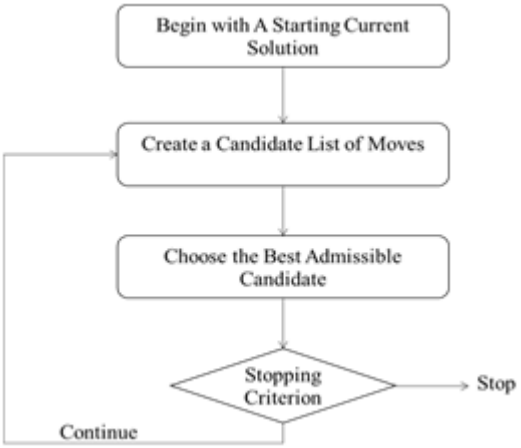


Fig. 2. Flowchart of the hill climbing method.

*C. cryptosystem*

The encryption system is done precisely in 4 steps:

**Step 0:** Coding

An individual (or chromosome) is a vector of size $m$.

Genes are the lists $L_{pi}$ $(1 \leq i \leq m)$.

The $i^{th}$ gene $L_{pi}$ contains the new positions that will take the block $B_i$.

**Step 1:** Creating the initial population

We denote by $P_0$ the initial population which consists of $q$ individuals: $X_1, X_2, ..., X_q$.

Let $x$ the chromosome whose genes are the lists $L_1, L_2, ..., L_m$ which indicate a clear message.

We take randomly $q$ permutations of $\{1,2, ..., m\}$ and we apply them to *Original-Ch* for obtaining an initial population consisting of $q$ potential solutions.

$i: = 0$

**Apply** the local search procedure on each individual of the population $P_0$

**Step 2:** Evaluation of individuals

Let $X_j$ an individual of $P_i$ whose genes are: $L_{j1}, L_{j2}, \ldots, L_{jm}$.

The evaluation function *F* [16, 18, 19] is defined by:

$$F(X_j) = \sum_{i=1}^{m} \left| card(L_{j_i}) - card(L_i) \right|$$

**Step 3:** Selection of the best individuals

In the step of selecting the method of roulette is used to retain the strongest individuals, this method is described in [17].

**Step 4:** Applications of genetic operators and the hill climbing method

We apply genetic operators adapted to the problem of permutations with constraints:

*1-Crossover MPX (Maximal Preservative X)*

This operator is applied to the individuals outcome the step of selecting with a suitable rate [12]. The best rate is about 60% to 100%.

*2 - Mutation*

We choose the mutation which randomly permutes two genes of a chromosome. This operator is applied to the individuals outcome the crossover with a specific rate, preferably from 0.1% to 5% [17].

*3 - Local Search*

Apply the hill climbing to individuals outcome the crossover and the mutation.

Put the new individuals in a new population $P_{i+1}$.

Repeat **steps 2**, **3** and **4** until a stop criterion is satisfied.

***- The stop condition:***

The algorithm stops when the population is not moving fast enough.

The pseudo code of the hill climbing applied to the individuals of the initial population, or resulting from crossover step is given as follows:

***Algorithm***

**Step 1:** Choose an initial solution *i* in *S* (all solutions)

A solution is a vector *v* of size *m*. The content of *v* is the lists $L_i$ $(1 \leq i \leq m)$ of characters position. The $j^{th}$ list $L_j$ contains the new positions which will take the block $B_j$

**Apply** $i = i^*$

$k = 0$

**Step 2**: apply $k = k + 1$ and generate a subset of solutions in $N(i, k)$ so that:

The neighborhood of solutions will be based on the permutations of the order of the lists positions. Specifically, it generates the candidate solutions by the application of random permutation on the position of the two lists of the current solution.

**Step 3:** choose the best solution *i'* from the set of neighboring solutions $N(i, k)$

**Apply** $i = best\ i'$

Let *i'* be a solution of $N(i, k)$ in which the lists are: $L'_{k1}, L'_{k2}, \ldots, L'_{km}$.

We define the evaluation function f in the set of solutions i'

by:

$$f(i') = -\sum_{j=1}^{m} \left| card\left(L'_{k_j}\right) - card(L_j) \right|$$

If $f(i) \leq f(i^*)$, so we found a better solution.

**Apply** $i^* = i$

**Step 4:** If a stop condition is reached, stop.

Otherwise, return to Step 2.

**Final phase of our algorithm**:

Let *Best_Sol* the final solution given by SMC. We build our encryption key (*Memetic-key*) from *Original_Sol* and *Best_Sol*. This key will allow to encrypt the clear message by changing the distribution lists on the various characters of the message *M*.

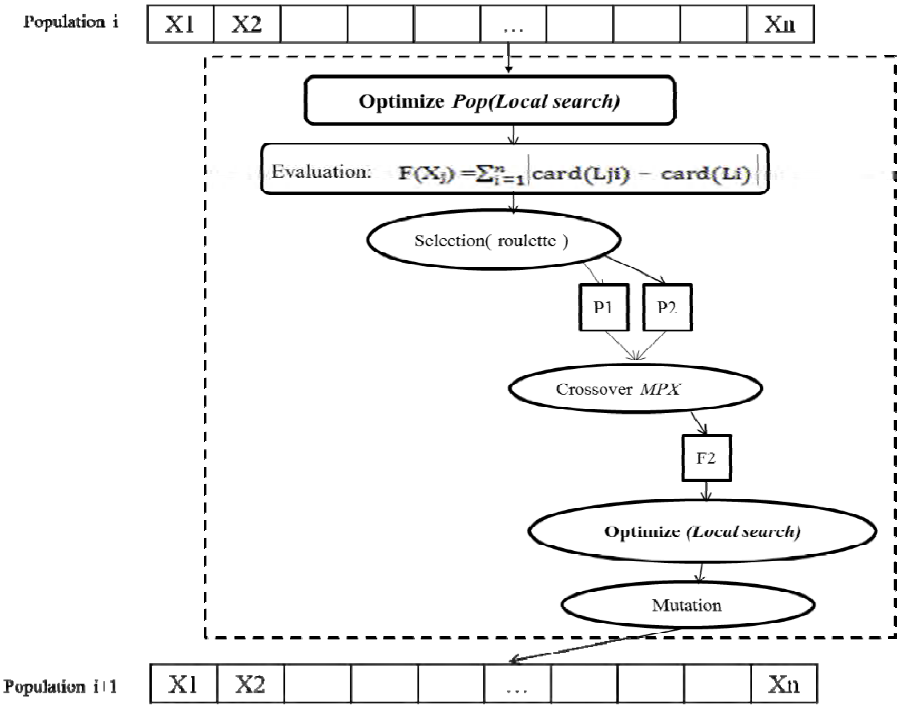


Fig. 3. One iteration of SMC algorithm.

*D. Decryption*

We represent the encrypted text $M'$ by a vector of list. Note by $B'_1, B'_2, ..., B'_M$ different blocks of $M$ and $L'_1, L'_2, ...,L'_m$, their respective lists of positions.

Decoding begins with the inverse of the last operation of the encryption. By using Memetic-key generated by our algorithm, the binary blocks will recover their original positions lists.

The key can be represented by a vector of size m which is called *Key*, so that:

*Key (1)* = $p_1$, *Key (2)* = $p_2$, ..., *Key (i)* = $p_i$, ..., *Key (m)* = $p_m$ where:

The block $B'_{p1}$ will correspond to the list $L'_1$.

The block $B'_{p2}$ will correspond to the list $L'_2$.

The block $B'_{m1}$ will correspond to the list $L'_m$.

Finally, we obtain the plaintext $M$.

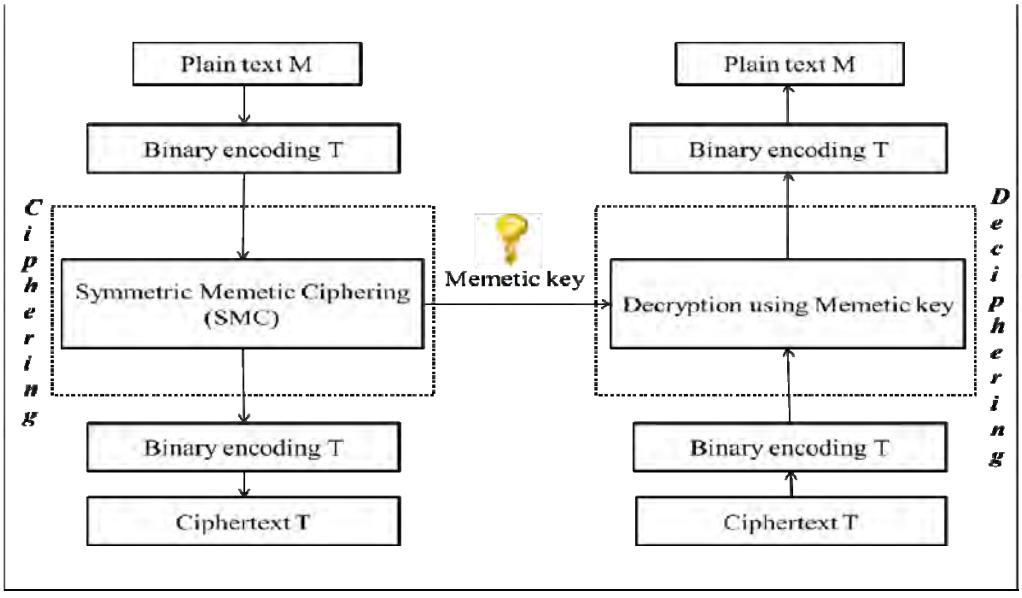


Fig. 4. Diagram of our encryption system SMC

## IV. EXPERIMENTATIONS

To illustrate the performance of the new SMC system, we applied our algorithm to messages of different sizes. For each, we executed the application for different population sizes and we selected the best. The experiments we performed are: comparison of frequency of occurrence, the convergence value of the evaluation function, number of generations reached during this convergence, the length of the encryption key and the execution time.

### *A. Comparison of occurrence frequencies*

The table and figure below compare the frequencies of characters appearance in the plaintext and the encrypted text with SMC.

TABLE 1
Frequency analysis in the plaintext and the encrypted text with SMC

| Characters | Frequency analysis in the plaintext | Frequency analysis in the ciphertext with SMC |
|---|---|---|
| 1 | 223 | 123 |
| 2 | 192 | 95 |
| 3 | 169 | 94 |
| 4 | 108 | 92 |
| 5 | 98 | 88 |
| 6 | 95 | 83 |
| 7 | 88 | 70 |
| 8 | 83 | 43 |
| 9 | 75 | 43 |
| 10 | 57 | 42 |
| 11 | 52 | 40 |
| 12 | 39 | 39 |
| 13 | 33 | 36 |
| 14 | 21 | 33 |
| 15 | 20 | 33 |
| 16 | 20 | 33 |
| 17 | 16 | 31 |
| 18 | 12 | 29 |
| 19 | 10 | 28 |
| 20 | 10 | 26 |
| 21 | 10 | 23 |
| 22 | 9 | 21 |
| 23 | 7 | 21 |
| 24 | 6 | 20 |
| 25 | 6 | 19 |
| 26 | 5 | 16 |
| 27 | 3 | 13 |
| 28 | 1 | 11 |
| 29 | - | 11 |
| 30 | - | 10 |
| 31 | - | 9 |
| 32 | - | 8 |
| 33 | - | 2 |

Frequency analysis is based on the fact that in each language certain letters or combinations of letters appear with some frequency. This information allows cryptanalysts to make assumptions about the plaintext provided that the encryption algorithm preserves the frequency distribution.

An attack of this kind is to be rejected. Indeed, due to the binary coding and implementation of the encryption system SMC, the appearance frequencies of characters are not recognized; therefore, cryptanalysis by using frequency analysis cannot be based on wrong statistics.

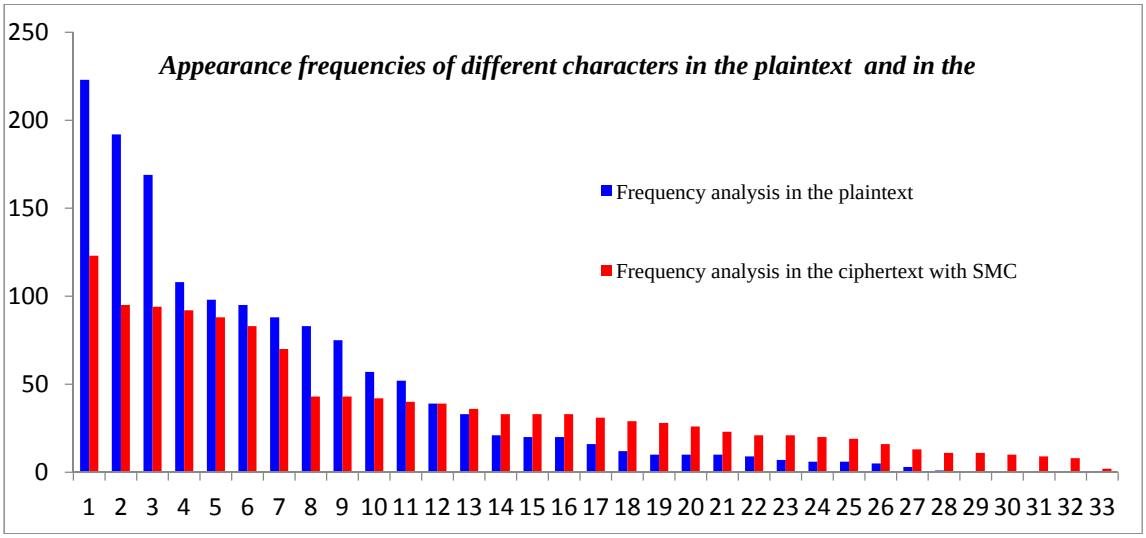


Fig. 5. Graphical representation of the appearance frequencies of different characters in the plaintext, the ciphertext with SMC.

*B. Configuration*

We record the results on the number of sufficient iterations for the convergence of the system to find the best parameters to achieve the optimal solution in an ideal time.

The following table shows the results:

TABLE II
Summary of results

| Size of Plaintext | Pop | Size of blocks | 5 | 6 | 7 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|
| 1 000 characters | 20 | Number of iterations | 37 | 33 | 31 | 34 | 42 | 43 | 48 |
| | | Convergence | 0 | 0 | 55 | 0 | 0 | 31 | 0 |
| | 30 | Number of iterations | 38 | 49 | 55 | 40 | 43 | 42 | 56 |
| | | Convergence | 0 | 0 | 15 | 0 | 43 | 0 | 22 |
| | 40 | Number of iterations | 41 | 45 | 50 | 48 | 46 | 60 | 59 |
| | | Convergence | 0 | 39 | 0 | 77 | 0 | 87 | 0 |
| 3 000 characters | 20 | Number of iterations | 36 | 46 | 52 | 47 | 67 | 58 | 61 |
| | | Convergence | 0 | 0 | 0 | 0 | 19 | 78 | 0 |
| | 30 | Number of iterations | 48 | 44 | 54 | 66 | 70 | 59 | 77 |
| | | Convergence | 0 | 0 | 74 | 0 | 20 | 89 | 33 |
| | 40 | Number of iterations | 42 | 53 | 67 | 41 | 69 | 72 | 72 |
| | | Convergence | 46 | 0 | 88 | 0 | 0 | 20 | 0 |
| 6 000 characters | 20 | Number of iterations | 34 | 38 | 61 | 55 | 63 | 77 | 69 |
| | | Convergence | 0 | 18 | 29 | 0 | 0 | 34 | 0 |
| | 30 | Number of iterations | 39 | 44 | 50 | 68 | 65 | 60 | 67 |
| | | Convergence | 0 | 0 | 66 | 88 | 0 | 17 | 0 |
| | 40 | Number of iterations | 43 | 36 | 57 | 48 | 55 | 68 | 58 |
| | | Convergence | 40 | 29 | 0 | 42 | 0 | 99 | 0 |
| 10 000 characters | 20 | Number of iterations | 49 | 41 | 55 | 64 | 53 | 69 | 64 |
| | | Convergence | 0 | 0 | 36 | 0 | 55 | 0 | 0 |
| | 30 | Number of iterations | 36 | 33 | 46 | 55 | 57 | 62 | 77 |
| | | Convergence | 0 | 78 | 0 | 96 | 0 | 0 | 33 |
| | 40 | Number of iterations | 34 | 38 | 51 | 48 | 53 | 60 | 66 |
| | | Convergence | 96 | 38 | 0 | 43 | 0 | 66 | 98 |

We can see that in most of the examples treated in our experiments, we find that the best values of the optimum (convergence value) are reached for a population of equal size 20, whose binary blocks size $k = 5$, $k = 6$ or $k = 7$. In some cases, population sizes equal to 30 or 40 gave good results, but the number of iterations in this case has grown significantly, which results in an additional cost in terms of execution time.

*C. The key length*

The key length is an important security parameter. The key to our system is composed of two elements: Memetic key and block size '*k*'.

We calculate the number of different blocks in existing texts to determine the size of the encryption key, the table below summarizes the study on some texts of different sizes.

TABLE III
Summary results showing the number of different binary blocks generated by SMC

| Size of blocks / Size of Plaintext | **5** | **6** | **7** | **9** | **10** | **11** |
|---|---|---|---|---|---|---|
| 1 000 Characters | 24 | 52 | 99 | 303 | 315 | 476 |
| 3 000 Characters | 26 | 56 | 106 | 330 | 316 | 604 |
| 6 000 Characters | 31 | 64 | 118 | 345 | 385 | 633 |
| 10 000 Characters | 32 | 67 | 120 | 353 | 407 | 748 |

- The size of the key Memetic is product of the various blocks and 8 bits.

A Small size of cryptographic key is now considered as insecure (The smaller the key size, the more attacks against the algorithm will be less complex). Indeed, the modern calculating capabilities allow finding the key by enumerating all possible keys. The length of the encryption key generated by our system has a variable length and depends on the number of different blocks existing in the texts. If we compare the length of the minimum key generated by our system to the recommended size for symmetric systems, we can conclude that our system provides protection to a higher level than other existing systems.

*D. Execution time*

*1) Comparison between the execution time of SMC and the execution time of SEC*

In this section, we compare the execution time of our encryption system SMC to the system encryption SEC. The table below summarizes the comparison.

TABLE IV
Comparison between the execution time of SMC and the execution time of SEC

| $N^O$ | Size of Plaintext | Ciphering System | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| I | 1000 Characters | SEC | 55 | 52 | 54 | 54 | 55 |
| | | SMC | 40 | 46 | 47 | 41 | 46 |
| II | 3000 Characters | SEC | 51 | 54 | 57 | 56 | 58 |
| | | SMC | 54 | 52 | 55 | 50 | 50 |
| III | 6000 Characters | SEC | 57 | 55 | 62 | 60 | 58 |
| | | SMC | 50 | 51 | 56 | 52 | 55 |
| IV | 10000 Characters | SEC | 55 | 62 | 59 | 54 | 56 |
| | | SMC | 53 | 63 | 60 | 50 | 52 |

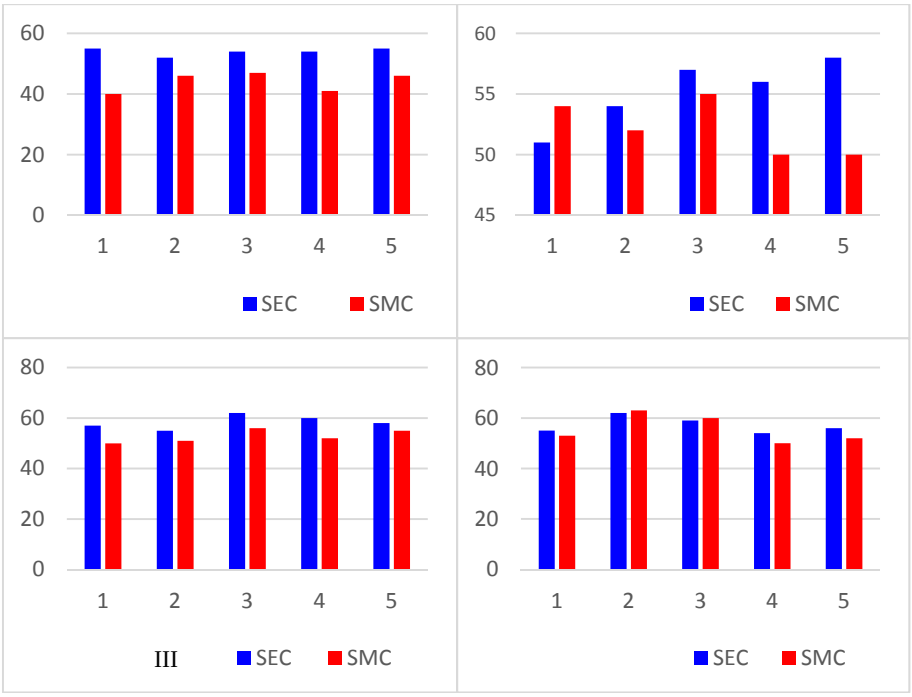


Fig. 6. Graphical representation of comparison between the execution time of SMC and the execution time of SEC

Since most of the processes used in our encryption system are random, this makes the measurement of execution time more complex. Otherwise, tests obtained by our program execution time show that our new SMC system is generally faster than the SEC system.

*2) Comparison between modern algorithms and memetic algorithm*

The encryption time is the time taken to encrypt the entire file measured in milliseconds. Table V and Fig. 7 show the results of the encryption time using 3DES, RSA and the proposed algorithm SMC for the input file size equal to 20 Ko.

The parameters used for each algorithm are:

3DES: no parameter influencing the execution time.

RSA: The parameters that affect the execution time are the values of prime numbers $p$ and $q$ (the time increases according to the value of $p$ and $q$).

We take $p$ =11 and $q$ = 107.

SMC: The parameters that influence the execution time is the population size and the block size. We chose the size of population equal to 20 and the size of block equal to 7 bits.

Table V
Time of encryption and decryption

| System | Time of encryption | Time of decryption |
|---|---|---|
| Triple DES | 75 ms | 70 ms |
| RSA | 150 ms | 160 ms |
| SMC | 55 ms | 19 ms |

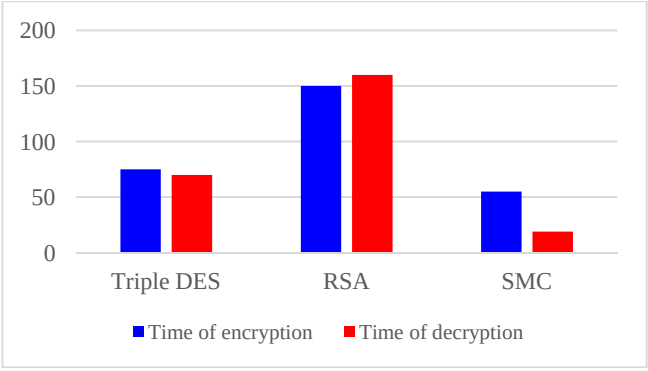


Fig. 7. Graphical representation of encryption and decryption

In consideration of their specific properties, asymmetric ciphers are generally less efficient than their symmetrical counterparts: the processing times are longer and for an equivalent level of security, the keys must be much longer. The 3DES uses operations which take less time in software implementation. SMC uses random process iteratively which also takes less time. The proposed algorithm shows good performance compared to other algorithms.

## V. CONCLUSION

In this article, we are interested in the application of memetic algorithm, which is a hybrid metaheuristic combining an evolutionary algorithm and a local search method for solving a scheduling problem in the domain of symmetric encryption. The objective is to improve some performance criteria of the ancient system.

The results of experiments obtained are compared to those of the evolutionary encryption algorithm to show the improvement made to the latter by introducing it to a local search procedure. The results show the performance of our memetic encryption algorithm.

The local search method introduced in the hybridization process is the hill climbing. It will be beneficial and interesting to use another method of local search more evolved as Tabu search or simulated annealing. We can also increase security of our system by combining it with another encryption method such as [8, 9].

## REFERENCES


[1] Bouly H., D-C. Dang, and A. Moukrim. 2010. A memetic algorithm for the team orienteering problem. 4OR, 8(1): 49–70.
[2] Dang D., R. Nesrine Guibadj, and A. Moukrim. 2011. A pso-based memetic algorithm for the team orienteering problem. In EvoApplications, pages 471–480.
[3] Delfs H. and H. Knebl. March 2007. Introduction to Cryptography: Principles and Applications. Springer.
[4] Florin G. and S. Natkin. 2002. Les techniques de la cryptographie.CNAM.
[5] Hongfeng W., W. Dingwei, Y. Shengxiang. 2009. A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems Soft Comput 13:763–780.
[6] Ishibuchi H., T. Yoshida and T. Murata. 2003. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Trans Evol Comput 7(2): 204–223.
[7] JIN D., D. HE, D. Liu and C. Baquero. Octobre 2010. Genetic Algorithm with Local Search for Community Mining in Complex Networks. Tools with Artificial Intelligence, ICTAI ’10. 22nd International Conference on, IEEE.
[8] Kaddouri Z., F. Omary and A. Abouchouar. February 2013. Binary Fusion Process to the Ciphering System “Sec Extension To Binary Blocks. Journal of Theoretical and Applied Information Technology,.
[9] Kaddouri Z., F. Omary, A. Abouchouar and M. daari. June 2013. Balancing Process To The Ciphering System Sec. Journal of Theoretical and Applied Information Technology.
[10] Kumar R., S. Tyagi and M. Sharma. May 2013. Memetic Algorithm: Hybridization of Hill Climbing with Selection Operator. International Journal of Soft Computing and Engineering (IJSCE) ISSN: Volume-3, Issue-2.2231-2307.
[11] Lozano M., F. Herrera, N. Krasnogor, D. Molina. 2004. Real-coded memetic algorithms with crossover hill-climbing. Evol Comput 12(3):273–302.
[12] Migga Kizza J. February 2009. A Guide to Computer Network Security,USA: Springer.
[13] Moscato P. 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report C3P 826, Cal-teech Concurrent Computation Program.
[14] Moscato P. 1999 « Memetic algorithms: a short introduction », McGraw-Hill Ltd., UK, Maidenhead, UK, England. pp. 219–234.
[15] Moscato P. 1999. New ideas in optimization. Chapter Memetic algorithms: a short introduction, pages. McGraw-Hill Ltd., UK, Maidenhead, UK, England. 219–234.
[16] Mouloudi A., F. Omary, A. Tragha and A. Bellaachia. Novembre 2006. An Extension of evolutionary Ciphering System. International Conference on Hibrid Information Tchnology.
[17] Omary F. 2006. Thesis. Applications des algorithmes évolutionnistes à la cryptographie. University of science- Rabat.
[18] Omary F., A. Tragha, A. Bellaachia, A. Mouloudi. February 28 2007. Design and Evaluation of Two Symmetrical Evolutionist-Based Ciphering Algorithms. International Journal of Computer Science and Network Security (IJCSNS), pp 181-190.
[19] Omary F., A. Tragha, A. Lbekkouri, A. Bellaachia, A. Mouloudi. 2005. An Evolutionist Algorithm to Cryptography. Brill Academic Publishers – Lecture Series And Computational Sciences Volume 4, pp.1749-1752
[20] Reeves C.R. 1993. Modern Heuristic Techniques for Combinatorial Problems. John Wiley & Sons, Inc.
[21] Schneier B. 1996. Cryptographie appliquée. Seconde edition (John Wiley & Sons).
[22] Stallings W. November 2005. Cryptography and Network SecurityPrinciples and Practices, USA: Prentice Hall, Fourth Edition,
[23] Stinson D. 2003. Cryptographie Théorie et pratique. Traduction de Serge Vaudenay, Gildas Avoine et Pascal Junod. (2ieme edition). Paris Vuibert Informatique.
[24] Wayner P. December 2008. Disappearing Cryptography: Information Hiding: Steganography & Watermarking, USA: Morgan Kaufmann, Third Edition.


## AUTHOR PROFILE


Zakaria Kaddouri PhD research scholar and member of research laboratory in computer science at the faculty of sciences, Mohamed V University in Rabat, where he has obtained a Doctorate degree in computer sciences, computer security option in 2014. His research interests include Design and construction of new cryptosystems.

Fouzia Omary Received a Doctorate of high Graduate studies degree in theories of computer Sciences from Mohammed V university, 1988 and Doctorate of state degree (or PhD) July 2006 in computer sciences from the same university. In 1983, she joined the faculty of science, Mohammed V University, Morocco where she is currently a professor in Department of computer sciences, and director of the research laboratory in computer science.