Cryptanalysis of ciphertext substitution using ACO algorithms

Tahar MEKHAZNIA[1], M.Bachir MENAI[2], Abdelmadjid ZIDANI[3]
[1] Computing department, TEBESSA University ALGERIA, mekhaznia@yahoo.fr
[2] Computing department science, CCIS, King Saud University, RIADH, KSA, menai@ksu.edu.sa
[3] Computing department, BATNA University, ALGERIA, zidani@free.fr

Abstract

The document presents a first step towards the automation of several techniques for classical cryptanalysis of ciphertexts by substitution method without manual intervention using ACO algorithms. The tests presented are limited due to the large number of parameters used, including statistical tables of the literary languages of which belong to. The study focuses mainly on how to choose these parameters, a question that remains unanswered so far.

Key words:

ACO algorithm, substitution cipher, cryptanalysis.

1. Introduction

Cryptanalysis is the art of transforming ciphertext into its equivalent plaintext without a priori knowledge of the decryption key. It is also the major challenge of current research in data security.

Typical attacks of ciphertexts are many and varied. The most famous being is the brute force, which consists of trying a multitude of keys in order to get some clear text, which deduct, the encryption key used. However, this technique consumes more of resources and is not interesting in practice.

Actual research is leaning towards using heuristics algorithms, which found interesting to solve a wide range of combinatorial problems. They operate some of the weaknesses of ciphertexts, including the systematic elimination of candidate encryption keys which seem unlikely or consideration of the occurrence of each symbol of the text according to the characteristics of literary language used.

The classical cryptosystems produce ciphertexts using simultaneous the substitution and transposition of characters. Modern techniques such AES, DES, IDEA, uses these techniques in an iterative manner on several levels (*Fiestel chains*) to produce ciphertexts difficult to break.

The ACO technique, inspired by ANT behaviour, and intended to solve various combinatorial optimization problems such as TSP, has been the subject of various recent contributions. In this paper, some versions of this technique are tested for the cryptanalysis of ciphertexts. It is organized as follows: Section 2 presents some recent work in the field, followed by Section 3 which presents the mathematical model for evaluating deciphered texts. Sections 4 and 5 detail the ACO technique and its adaptation to solve the problem of classical cryptanalysis.

Section 6 illustrates the proposed algorithm, followed by preliminary results presented in Section 7. Given that the technique has several variants of ACO algorithms, some of them are presented in Section 8. The summary and comments are illustrated in Section 9.

2. Previous works

The use of heuristic techniques for solving optimization problems in cryptanalysis took part in important research of the last twenty years, starting with Peleg & Rosenfeld [1] who modeled the cryptanalysis of the encrypted text by striking a probabilistic problem. Carrol & Martin [2] have developed an approach of SE for cryptanalysis algorithms using relaxation methods. Forsyth & Safavi-Naini [3], Spillman & Al [4] and Clerk [5] have made use of heuristic techniques such as simulated annealing and genetic algorithms. Bahler & King [7] have used the statistics of appearance of characters in literature and reimplemented the work of Peleg and Rosenfeld. M.faisal & M.Youssef [8] have reimplemented the heuristics for cryptanalysis of substitution ciphers using statistical languages. A. Dimovski & D. Gligoroski [9] has used the same techniques to the attacks of transposition ciphers.

The *ACO technique*, part of the range of heuristic methods has become an important part in this work to solve complex problems, among others, *Travelling Salesman Problem* [22], *Quadratic Assignments* [23], and *Scheduling Problem* [24].

Similarly, the classical cryptanalysis has also been treated using various other heuristics methods, including *Tabu search*, *Simulated Annealing* and *Genetic Algorithms* [5][12][26].

Recent research shows that is only a little work of ACO technique were devoted to research in classical cryptanalysis and the results in this direction are not readily available.

This is evident due to the complexity of the technique and its sensitivity to its many parameters compared to its neighbours, including the Simulated Annealing Method and Taboo Search, which can provide competitive results with minimal effort.

Similarly, the comparative study of some alternative of ACO algorithms, presented in this paper, and their

simultaneous integration in various stages of resolution of the problem has yielded promising solutions.

3. Cryptanalysis per substitution

3.1 Definition :

Let an alphabet $A(n)=(a_0, a_1, .., a_{n-1})$ and $B(n)=(b_0, b_1, .., b_{n-1})$ another alphabet obtained from **A** by the bijective function $K:A\rightarrow B$, which substitutes a character of A with exactly one character in the alphabet B. This function is called *key encryption* of a plaintext belonging to A to a ciphertext belonging to the language B. Of course, the inverse function can be performed using the function K^{-1} assumed known from the receiver of the encrypted message.

The cryptanalysis is to build a clear message based on an encrypted message without knowing the key K (or K^{-1}). The basic logic is to test all possible combinations of keys, a number of size exceeding 26!≈288 where exhaustive tests are excluded.

3.2 Appearance of characters :

One of the characteristics of a literary language is the number of appearance of its characters in plain text. The letter E for example, is one that occurs most often in English text, while the letter Z is the least visible. We talk however *unigrams*. Similarly, the occurrence of syllables in more than one character is more frequent for *bigrams*, like TH, HE and *trigrams* like THE, ING or AND. Statistical tables of characters frequencies[10] were developed by researchers are exploited in cryptanalysis of ciphertexts.

3.3 Index of coincidence:

Let us a text t of length n. The index of coincidence of a character c of t is given by the relation [11]:

$$I_c(t)=\frac{1}{(n/n-1)}\sum_{i=a}^{z}p_i(p_i-1) \qquad (1)$$

where p_i is the number of occurrences of character c in the text t. Of course, we assumed the text included the 26 letters of the alphabet only. For commercial or scientific texts, other criteria are taken in consideration depending on the nature of the message. If, for example, each character appears with equal frequency in a text, the index of coincidence would be about 1/26≈0.04. The reality is that some characters have higher frequencies than others, and have an index of coincidence greater. The overall index of coincidence is given by the relationship

$$I_c=\sum_{0}^{25}p_i^2 \qquad (2)$$

The value of this index varies from one language to another. It is, for example for English equal to 0.065 and, equal to 0.074 for the French language.

3.4 Cost function

Using tables of frequencies of appearance of characters, the substitution of one character by another in a ciphertext is closer to reality if the frequency difference between the two characters is minimal, i.e., the new text generated close to the original plaintext. This difference would be zero if it falls on good character. The cost function is defined by the following equation [12]

$$Cout(K)=a\sum\left|R^U-D^U\right|+b\sum\left|R^B-D^B\right|+c\sum\left|R^T-D^T\right| \qquad (3)$$

where R, D respectively denotes the portions of the encrypted text and it equivalent obtained after substitution using the key K. A portion may be a unigram, bigram, trigram or more.
a, b and c denotes coefficients in the range [0,1] and justified by experiments.

4. The ACO Technique

In its moving, an ***ant*** leaves a constant and continuous quantity of pheromone on its way. The choice of direction of movement is subordinated by the pheromone trail left by other ants who followed the same path before. The pheromone evaporates on contact with air in a const manner also. The path with less phenomenal tends to disappear and be replaced by new path shorter. After a certain number of movements of ants, only paths most frequented constantly resist to evaporation and provide the minimum distance between the nest and the food.

5. Adaptation of the ACO technique to our problem

5.1 Basic data :

- The field of exploration is a strongly connected graph of 26 nodes (27 if we added the 'space' to alphabet) where the ants move in a manner that does not pass through a node more than once.
- The basic key of cryptanalysis K_0 is chooses randomly.
- The distance $d(i,j)$ between two nodes i and j of graph is a function depending on the difference between the original cost of the text and cost of a new text obtained by using the key K_0.
- The starting node is chosen randomly for each ant.

5.2 Iterations :

Each ant is moving in a discreet way, from its starting node i through all the remaining nodes to construct a *Hamiltonian path*. A control function of single pass on each node must be activated. On each moves from node i, the probability of choosing the next node j (not visited yet) depends on the distance $d(i,j)$ which separates the current node i and the amount of pheromone τ_{ij} on arc (i,j) by the relation [8]:

$$p(i\rightarrow j)=\frac{\tau(i,j)^{\alpha}d(i,j)^{\beta}}{\sum\tau(i,j)^{\alpha}d(i,j)^{\beta}} \qquad (4)$$

where α, β are constant variables whose values will be justified experiments.

The distance d, parameter not significant in cryptanalysis, can be obtained from the function of the cost of the arc according to the relation:

$$d(i,j) = cout(i,j)^{\chi} \qquad (5)$$

with χ, a constant defined in experiments.

5.3 Pheromone update:

After each iteration, the update of the pheromone over the arc along visited path is given by the relation [13]

$$\tau(i,j) = \tau(i,j) + \Delta\tau(i,j) \qquad (6)$$

where $\Delta\tau(i,j)$ is a positive quantity depending on the variant of the algorithm used. It is, for example, in virtual ants' case, equal to Q/L where L is the length of the Hamiltonian cycle produced by a given ant, and Q, the cost of text determined by (1) and the cycle length is the sum of the lengths of arcs that constitute it. Equation (6) would be defined as

$$\tau(k) = \tau(k) + \frac{Q}{\sum_{j,j\in K} cout(i,j)} \qquad (7)$$

with k, the key used and i, j vertices belonging to the cycle traversed by the ant.

The amount of pheromone added is inversely proportional to the cost of the text. It's lower, as we approach the actual decryption key.

5.4 Evaporation :

The evaporation of pheromone is made after a given number of iterations independent of the progress of ants, uniformly over all the arcs of the graph [14] according to the relation:

$$\tau(i,j) = (1-\rho)\tau(i,j) \qquad (8)$$

where ρ is a constant in [0,1] interval that is important to choose, because if it gets too close, it's causes an effect of stagnation of pheromone on the arcs where the persistence of bad solutions. Similarly, the approximation of 1 implies a rapid evaporation, therefore, loss of use of the bow in question.

6. AntSystem Algorithm (AS)

6.1 Description

The AS algorithm that we proposed comprises broadly, the following tasks:

```
Building a starting solution
 repeat
   construct a solution (one round) for each ant
   improve solution by local search
   reward the best solutions by adding pheromone
   evaporate pheromone trails
Until  (maxCycles or bestSolution)
```

The development of these steps leads to the following form:

```
Calculate the cost of text according to (3) (solution S_optimale)
Set the lengths of the arcs according to (5)
Place m ants randomly on the vertices of the graph
For nb_iterations = 1 to max_iter
  For k = 1 to nb_fourmi
    Building a path S(k) according to (4)
    Calculate the cost of the solution S(k)
    If (nb_iterations% Evap) = 0 Evaporer_phéromone
  EndFor
  If (S(k) is better than S_optimale) S_optimale ← S(k)
EndFor
```

6.2 Function:

The implementation of the AS algorithm requires its structure in several steps:

a. Choice of input parameters**:**

- The choice of the city of departure is random for each ant. However, experiments reveal that an equitable distribution of ants on every city converges more toward the optimal solution.
- Initialize the pheromone graph. A quantity τ_{min} positive or null is assigned to each arc. The optimal value of this quantity is justified when testing.
- Resetting the distances between cities. Recent studies suggest a positive a dummy value initially, similar to the initial value of pheromone [13]. Our experiment shows that the solution converges better if this value is determined separately for each arc and, it's obtained by the difference in the cost of initial key and actual key. The table of distances can be changed if needed during treatment if new keys appear interesting.
- A tabu list for each ant is initialized to its starting city.

b. Moves of ants:

- **Choice of next city:** each ant must visit all cities of the graph (change one character by another in the key reference). The changes are inserted in the tabu list while moving in preventing them from making choices that already made. The choice of each future change is based on the probability mentioned in the relation (4).

  Of course, if the result is zero or below a nominal value (in a dry arc with pheromone or a quantity less than a threshold), the ant will be blocked in stalemate. It will be destroyed and replaced by another who will resume the way with new data.
- **Pheromentation** : It will be during the journey of ant, or in return path as if it's a real or virtual ant. The amount deposited can be a constant or variable. Several cases are considered in experiments.
- **Evaporation**: It will be after a certain time of moves as a homogeneously on all arcs regardless of the number or types of ants. The evaporation period can be a step of move, a cycle or generally, after a given number of iterations.

The amount of pheromone deposited or evaporated can be limited by bounds to keep some warmth on the set of arcs for a maximum use [15].

c. End of algorithm:

Since the solution of the problem can not be reached quickly, stopping the algorithm will be done after a certain number of iterations or from the boundary of a temporary period. Preferably it is selected as multiple of the number of cities in the graph so that all ants complete their respective paths.

During the evolution of ants, the least expensive paths will be retained. They can serve as baseline data for other parts of generations of ants.

6.3 Complexity :

The algorithm can be divided into 5 sections. The complexity analysis is done for each section separately [13]:

a. Initialization data, including tables of pheromones and distances. The complexity is the order of $O(|L|+m)=O(n^2+m)$.With m, the number of ants, n, the number of cities and L, the length of path.
b. Complete a cycle of course. Requires scanning all cities, where the complexity is $O(n^2 \cdot m)$.
c. End of the cycle and calculation of deposits pheromone. Complexity is $O(|L|)=O(n^2)$.
d. Evaporation of pheromone: $O(|L|)=O(n^2)$
e. Determining the best path: $O(n \cdot m)$.

Total number of cycles, either C_{max} is variable and depends on the condition for stopping the algorithm. The overall complexity of the algorithm would be

$$O(n^2+m+C_{max} \cdot n^2 \cdot m)$$

well, in case maximum, it's about of

$$O(NC_{max} \cdot n^2 \cdot m)$$

7. Experiments

7.1 Parameter test :

The problem involves a large number of parameters which is difficult to treat them simultaneously at the same time given the lack of mathematical model to justify the choice of certain values. Similarly, the range of choices and step of move is also undefined. The extensive use of wide ranges consumes enormous resources and does not validate certain values outright, because each variant of algorithm (as discussed below) has its proper values. The following table illustrates some of these values:

parameter	range	step	item
Alpha, beta	0-1	0.1	Probability of choice next step
Gamma	0-2	0.2	Rate of pheromentation
Teta	1-5	1	Distance between cities
Lambda1, Lambda2			Cost of text (Unigram, bi-gram)
Evap	0.1-1	0.1	rate evaporation
BI, BS	40-200	5	Edges of pheromones
Maxcars	90-300		size of ciphertext
Cevap	40-100	10	period of evaporation
Bonus	0-%Ph		Pheromone elitistes ants
NbFourmis	50-110	10	Number of ants
NbIter	26-520	N*26	Iterations number

The arguments of choices of ranges of above values are based solely on the results of experiments using the alternative min-max algorithm (§ 8.4) where results are acceptable but in reduced cases.

7.2 Graph cities :

Two test models were made for 26 cities (key standard alphabet) and 27 cities (alphabet and space). The selected values of other parameters are mentioned when presenting the results of experiments.

The distance between two cities is an insignificant parameter in the cryptanalysis of ciphertext. In our case, a city is defined as a character in the alphabet key. Moving from one city to another is the substitution of one character with another within this key and, contrary to classical problems like the TSP, where the distance is a physical quantity unchanging over time.

As input parameter, [8] suggests that this distance is the difference between the cost of the original text and the encrypted text obtained with the initial key. The idea of [15] and for reasons of ambiguity, this parameter can simply be removed from the relationship and consider only the parameter of the pheromone. Some other studies have proposed it as a fixed constant equal for all cities.

7.3 Local search :

Given the large number of parameters, some values taken randomly at the start can lengthen the processing time; see to distort the optimal path of ants. To determine optimal initial values, preliminary reduced tests have been made on various data bases. Better results, specifically the distances between cities, were taken as input parameters for further testing.

Similarly, several generations of ants were tested in the footsteps of other ants. The results are more acceptable especially in resource consumption.

8. Variants of algorithm

8.1 Real ants :

The real ant deposes the pheromone during its movement in real time in a homogeneous and continuous manner. Its activity ends after arrival at the last summit of the graph. It may however be trapped in an impasse (dry arc or containing an insignificant amount of pheromone), which causes the blockage of all ants have taken the same portion of the graph in one direction or another with a surpheromentaion of surrounding arcs.

The value of art is to alert following ants arriving in the same area as the destination selection is justified. The drawback of the method is the swing of activity to specific paths faster. Similarly, some paths will not be used. The arcs chosen were not always fast. They are only parts of the road in what could be interrupted by an impasse because the overall vision is not available.

In practice, this version of algorithm is used to decrypt a portion of the message only and seems to be interesting less used as a starting point for other models tested.

The experiments were realized with 3 kinds of keys: Simple, like Cesar or AtBash keys, medium, as Vigenere key, and more difficult, like Delastelle method.

The average results obtained with a text size in range [90-190] characters and using generations of

ants in range of [50-110] are following:

Ants	parameters	Key	Char. corrects	
			Best	Average
30	α=0.8 β= 0.4 γ=-0.5 τ=-0.5 Evap=72 BI=6 BS=20 Cvp=20	Simple	25	20,3
20	α=0.2 β= 0.6 γ=-1.0 τ =0.2 Evap=1 BI=1 BS=12 Cvp=52	Middle	18	16,1
22	α=0.9 β= 0.8 γ=-1.0 τ =-0.1 Evap=0.5 BI=4 BS=12 Cvp=52	Difficult	11	10,8

8.2 Virtual ants :

In reality, the filing of the pheromone will be during displacement in a homogeneous and continuous manner, which is what real ants do. The virtual ants update the pheromone trails once led in the end, since the amount deposited depends on the length of the path constructed (cost of decrypted text) and only the ants have constructed a complete path are entitled to deposit the pheromone, as they will do on way home. Of course, this deposit will require an additional step to reverse course and memorization of the path.

In practice, and being fixed an initial cost of the text, only decryption keys obtaining a plaintext whose cost is below the threshold are retained.

The advantage of this version of algorithm is that ants do not lead at the top end, will not be entitled to deposit pheromone, which has the effect of quickly distinguish the importance of certain portions of paths and favor some of them.

In following experiments, two variants of algorithms were tested:

a. Amount of pheromone deposited proportional to the length of the path:

After completing a Hamiltonian cycle and calculate its cost, each ant on its way back, lays a quantity of pheromone laid by the relation:

$$\tau(i,j) = \tau(i,j) + \frac{cout(i,j)^{t}}{\sum_{j,j\in K} cout(i,j)} \quad (9)$$

where ***k*** is the cycle completed and ***ι***, a constant justified in testing.

The best results obtained with a text of 100 characters are illustrated in the following table:

Ants	paramètres	Key	Char. corrects	
			Best	Average
40	α=1 β= 0.5 γ=-0.5 τ =-0.5 Evap=60 BI=5 BS=20 Cvp=20	Simple	22	19
20	α=0.11 β= 0.8 γ=-1.0 τ =0.2 Evap=1 BI=1 BS=12 Cvp=55	Middle	15	13,42
22	α=0.9 β= 0.9 γ=-1.0 τ =-0.1 Evap=0.5 BI=1 BS=12 Cvp=52	difficult	10	7,81

The results are significantly better than those obtained when tested on real ants because the technique promotes early some ways and avoids an unnecessary pheromentation for other arcs. The tests were preceded by a choice of parameters as noted in § 7.3 exhaustively, however the resource consumption is significantly.

b. Only the best path will pheromented:

Once all ants have completed their respective cycles, only the one whose term had the best cost path is allowed to a return in the opposite direction and depositing pheromone.

With a number of iterations close to 600, the best results with a text of 120 characters were illustrated in the following table:

parameters	Key	Characters corrects	
		Best	Average
α=0.7 β= 0.5 γ=-0.5 τ =-0.5 Evap=65 BI=4 BS=18 Cvp=20	Simple	18	12,33
α=0.11 β= 0.7 γ=-1.0 τ =0.2 Evap=1 BI=1 BS=12 Cvp=55	Middle	14	10,62
α=0.8 β= 0.5 γ=1 1 Evap=0.4 BI=40 BS=200 Cvp=40	Difficult	10	9

The results of the first alternative of the algorithm seems more interesting, however, extended the running time of the second version could yield more promising results, but its disadvantage is the rapid orientation of ants to specific areas of graph with a "dry" of other areas that remain untapped. This orientation "*quenching*" is due to a poor choice of basic parameters.

8.3 Elitist ants :

Proposed by [16], the idea is to grant an additional *bonus* of pheromone on the arcs of each path with the best result in a cycle. In other words, try to retrace some ants called *elitits* on the maximum arcs of the cycle in question in order to keep it rich in pheromone.

In our case, the idea is to keep a key which gave a better result intermediate still in view and modify just few of its characters in order to get a key that gives a better result. If this is not the case, we return to the reference key in way to modify other characters. This method allows highlighting the most interesting path in the graph; however it consumes more computing resources.

The bonus is a positive quantity of pheromone added to the arcs involved in the final best key as the relation:

$$\tau(i,j) = \tau(i,j) + \frac{Bonus}{\sum_{i,j\in K} cout(i,j)} \quad (10)$$

The experiment shows that the amount is significantly higher close to the value of pheromone deposited by other ants. The best results obtained with a number of iterations close to 850, a population of 60 ants and a text of 180 characters is illustrated in following table:

parameters	Key	Char corrects
α=0.8 β= 0.5 γ=-1.0 τ =0.2 Evap=1 BI=40 BS=200 Cvp=55	difficult	13

This version of algorithm also seems faster than his previous in treatment time.

When there are too many ants, they encourage the harvesting of local zones, it causes the problem of '*state of stagnation*'.. where the disadvantage of the method.

These anomalies can be reduced by evaporation in real time of pheromone during its deposition while keeping the privilege of operating in local ways (as presented in § 8.1). The update of the pheromone will be given by the following relation:

$$\tau(i,j) = (1-\rho)\tau(i,j) + \rho \frac{Bonus}{\sum_{i,j\in K} cout(i,j)} \qquad (11)$$

8.4 Max-min:

Version proposed by [18]. The idea is to explicitly limit the amount of pheromone on the arcs with two bounds $\boldsymbol{B_{Inf}}$ and $\boldsymbol{B_{sup}}$. At the start of exploration, the arcs of the graph are initialized to the max value.

The update of pheromone is done with a proportionate way; the strongest arcs are the least pheromented. It will be operate by only one ant, who has borrowed the cycle with minimum cost by relation:

$$\tau_{i+1} = \rho \cdot \tau_i + \frac{1}{\sum_{i,j\in Kopt} cout(i,j)} \qquad (12)$$

where $\boldsymbol{K_{opt}}$ being the best path obtained in a cycle of displacement.

The upper limit B_{sup} is variable, it is chosen so that it does not exceed the maximum value placed on the arcs of the trail. This is interesting because it establishes an equilibrium relative importance of arcs and encourages ants to explore other possibilities. The maximum value is expressed by the relation:

$$B_{\sup} = \frac{1}{1-\rho} * \frac{1}{\sum_{i,j\in K-opt} cout(i,j)} \qquad (13)$$

The lower bound is also dynamic. Its value was the subject of several proposals [6] [15], however, and with a lack of rigorous mathematical models, the value based on experiments appear more interesting [9]. We proposed however that this variable is defined by the relation:

$$B_{\inf} = \frac{B_{\sup} \cdot (1 - \sqrt[n]{p_{opt}})}{k \cdot \sqrt[n]{p_{opt}}} \qquad (14)$$

The best results obtained with a text of 90 characters are summarized in the table below:

Ants	parameters	Key	Char corrects
80	α=0.5 β= 0.6 γ=-0.8 τ =1.0 Evap=8 Cvp=32	middle	15
45	α=0.8 β= 9.0 γ=-0.8 τ =0.8 Evap=33 Cvp=120	difficult	13

The characteristic of this version of algorithm is to highlight all the arcs of the graph while keeping a balance of choice quite acceptable. It is a version that is progressing confidently, but slowly towards the optimal solution. Its disadvantage is the high consumption of resources, which slow the onset of the first results. Its advantage is its indifference in the choice of input parameters, because everything will be tested and only the optimal values will be retained for further treatment.

It is interesting to use this technique just for fixing pathways may yield better solutions and use other faster algorithms for further exploration.

9. Contribution

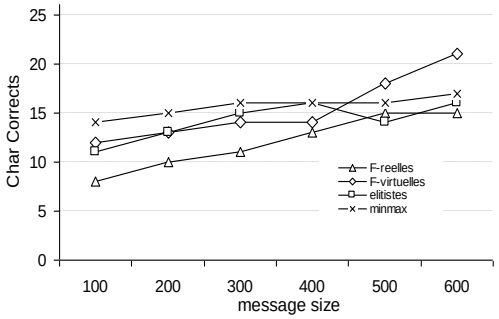
9.1 Works in cryptanalysis :

Following tests realized on the various proposed algorithms, we note the next facts:

- The most research on heuristic techniques are oriented towards the resolution of industrial or economic problems: routing, assignment, scheduling, .. [14] [18] and few studies have been devoted to research in cryptanalysis.
- The most studies addressed the cryptanalysis by the heuristics methods are content simply to use these techniques with studies of synthesis without trying to improve or change their settings.
- The most researchers in cryptanalysis are interested to methods of analysis or differential frequency methods appropriate for each encryption technique that gives up at the most promising results.
- Most cryptanalysis techniques refer to statistical data tables of literary language [5] [10] [19] [20]. These tables were based mainly on observation and differ from one country to another even within the same language.

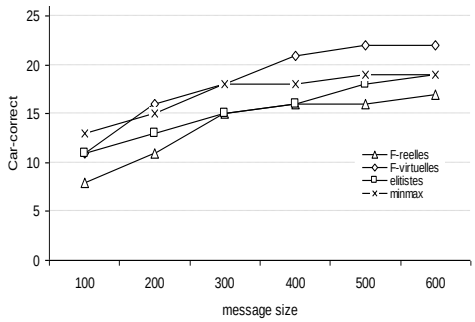
9.2 Synthesis of tests:

With a number of iterations between 500 and 1500, a colony of ants contains from 25 to 120 ants, tests on the various versions of algorithms shown above, using the permutation unigrams and various tables of statistics are illustrated in the following figure:

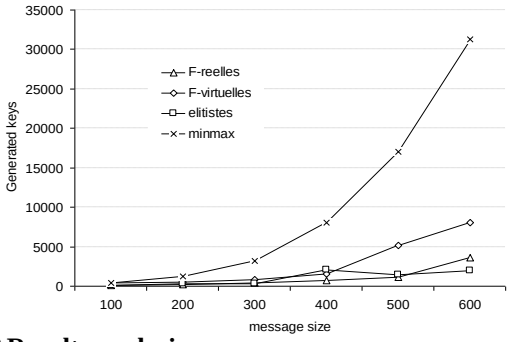


With the same data and using an average between data of unigrams, bigrams and trigrams, the results would be following:

The processing was performed on CPU Dual 2.0. The execution time for the same variants of algorithms was as follows:



Each test starts with a randomly chosen key. During the trip, ants exchange, each in its way, the character of this key to end with a metamorphosed keys closer to the optimum. Some of these will be used by other ants for further processing. If the optimal solution is not reached, other keys will be generated which has the effect of consuming more and more processing resources. The number of keys generated from different methods is shown in the figure below:



9.3 Results analysis :

The analysis of results indicated above show that the method of virtual ants seems the most efficient among the methods tested. The min-max technique gives better results but consumes more processing time because it generates a tree of keys large enough.

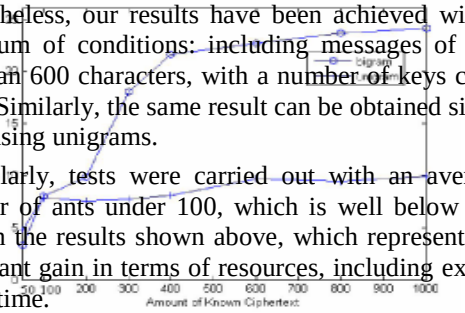
The best way to optimize the solution is to combine the techniques starting with a choice of basic parameters studied using the algorithm of virtual or real ants. A collection of keys will be produced. It will serve as support for the algorithm min-max for further exploration.

All techniques appear similar for small cases.

9.4 Recent works in the same category:

Few results have been found in the same category of research. The most interested are there illustrated in [8] and summarized by the following scheme. They were obtained with a number of ants equal to 1000. The other remaining parameters are defined in an optimal way, and for a text of 100 to 1000 characters. The correct number of characters is taken with an average of 100 keys tested:

Comparing these results with those presented in this paper (§ 9) shows that they are nearly equivalent. Nevertheless, our results have been achieved with a minimum of conditions: including messages of size less than 600 characters, with a number of keys close to 20. Similarly, the same result can be obtained simply by using unigrams.



Similarly, tests were carried out with an average number of ants under 100, which is well below that used in the results shown above, which represents an important gain in terms of resources, including exploration time.

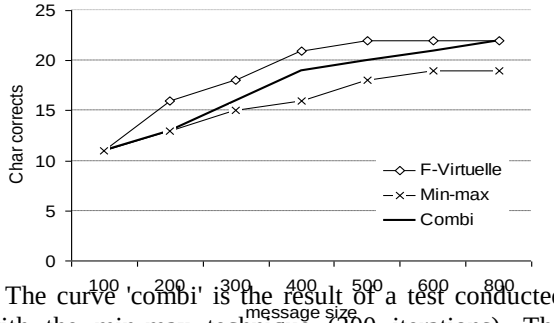
9.5 Proposition :

a. Choice of data :

As it was mentioned, the major drawback of techniques ACO is the large number of basic parameters. And, given the lack of mathematical model to follow in this sense, a study must first be made to compile data that can lead to the optimal solution with minimal effort.

The min-max algorithm works better for small jurisdictions. We propose here a preliminary treatment with this algorithm. An intermediate result will make known including the values of initial data. These parameters are taken as input for the technique of virtual ants to avoid unnecessary tests which appear when taking random values.

In this case, a preliminary test was conducted, the results were as follows:



The curve 'combi' is the result of a test conducted with the min-max technique (300 iterations). The results thus found were included as data for the technique of virtual ants. We notice a slight upward curve but only beyond 800 characters and with 1200 iterations.

b. Smoothing pheromone:

Mechanism discussed in some works [21] [22], smoothing pheromone trails named *shaking*, is to reduce the attraction of overburdened ways and roads leading to it without cancel the effect of pheromones,

which reasserts some local secondary roads considered unimportant ago.

The smoothing, is a dynamic update of pheromone dependant of the differences with the calculate value and the upper bound. It will be extended to all tests according to the relation:

$$\tau(i,j) = \tau(i,j) + \sigma \cdot (\tau_{\max} - \tau(i,j)) \quad (15)$$

where σ is a positive constant less than 1.

Trials preliminaries experiments have been made with this setting techniques for virtual and elitist ants, the results were substantially equivalent to those described above, while the execution time was much lower.

9.6 Conclusion :

Although they were help the researchers in cryptanalysis, the heuristics methods do not provide often accurate results and require a runtime rather important. Similarly, simple techniques such taboo, simulated annealing and some versions of genetic algorithms are favored over the algorithms of ant colonies because they require setting studied and rough, difficult to model.

Decryption of encrypted text can not be solved by using a special technique. It often requires manual intervention once the plain begins to appear.

Following experiments above, it appears that the ACO algorithms can yield interesting results provided they are combined in an appropriate manner. Similarly, the data must also be well chosen. Other ideas like the index of coincidence can be introduced in the compute of decryption keys cost. Also, other ideas can improve the quality of the solution, including the choice of key sizes greater than 27 or use a special statistical table.

10. References :

[1] S. Peleg and A. Rosenfeld, "Breaking substitution ciphers using a relaxation algorithm," Communications of the ACM, vol. 22(11), 1979.

[2] J. Carrol and S. Martin, "The automated cryptanalysis of substitution ciphers," Cryptologia, vol. 10(4), 1986.

[3] W. S. Forsyth and R. Safavi-Naini, cryptanalysis of substitution ciphers", Cryptologia, vol.17(4), 1993.

[4] R. Spillman, M. Janssen, B. Nelson and M. Kepner, "Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers," Cryptologia, vol.17(1), 1993.

[5] A.J. Clark, "Optimisation Heuristics for Cryp-tology", PhD thesis, Queensland University of Technology, 1998.

[6] M. Dorigo, K. Socha, "An Introduction to Ant Colony Optimization", Technical Report No. TR/IRIDIA/2006-010, April 2006.

[7] D. Bahler and J. King, "An implementation of probabilistic relaxation in the cryptanalysis of simple substitution systems", Cryptologia, vol.16(3),1992.

[8] M.Faisal Uddin, Amr M. Youssef, "Life Technique for the Cryptanalysis of Simple Substitution Ciphers", IEEE CCECE/CCGEI, Ottawa, May 2006.

[9] A. Dimovski, D. Gligoroski, "Alphabetic substitution cipher using a parallel genetic algorithm domain cooperation through SCOPES PROJECT", Ohrid, Maccedonia,2003

[10] Simon Singh , "The Code Book".

[11] Christophe RITZENTHALER, "The cryptology Course", Université de Marseille, 2006.

[12] A.Dimovski, D.Gligoroski, "Attacks on the Transposition Ciphers Using Optimization Heuristics",in Proceedings of ICEST 2003, October 2003, Sofia, Bulgaria

[13] C. Andrea, L. Thé, V. MARILL, "Optimisation par colonies de fourmis", mai 2006

[14] M. Dorigo, M. Birattari,T. Stutzle, "Ant Colony Optimization",TR/IRIDIA/2006-023,9/2006.

[15] F. Olivetti, F. J. VonZuben, "Max Min Ant System and Capacitated *p*-Medians: Extensions and Improved Solutions", Informatica, 163-171-2005.

[16] Dorigo M., V. Maniezzo, A. Colorni , "Ant System:Optimization by a colony of cooperating agents", IEEETransactions on Systems, Man, and Cybernetics-Part B,26(1):29-41

[17] T. Stützle, H. H. Hoos, "MAX-MIN Ant System Future Generation Computer Systems", Vol : 16 No 8,2000.

[18] N. Monmarché, "Algorithmes de fourmis artificielles : applications `a la classification et a l'optimisation", Thèse, université Tours, 2000.

[19] R. Lewand, "Cryptological Mathematics".

[20] Pfleeger, "Security in Computing", ch. 2

[21] Y. Pigné, "Modélisation et traitement décentralisé des graphes dynamiques", Thèse, Université le Havre,2008.

[22]A. Friedman, "Présentation : Variantes des algorithmes de Fourmis", Cours, Université UQAM, 2006.

[23] B. Bullnheimer, R. Hart land C. Strauss, "An improvement ANT system Algorithm for the vehicle routig problem", TR POM-10/97, Institute Of Management, University Of Vienna, 1997.

[24] A. Coloni, M. Dorigo, V. Maniezzo and M. Trubian, "Ant System for Job Scheduling", Belgian Journal Of Operations research, 1994.

[25] V. Maniezzo, "Exact and Approximate Nondeterministic Tree-search Procedures for Quadratic Assignement Problem", TR-CSR 98-1, University Of Bologna, Italy, 1998.

[26] A. K. Verma, Mayank Dave and 3R. C. Joshi, "Genetic Algorithm and Tabu Search Attack on theMono-Alphabetic Subsitution Cipher in Adhoc Networks",Journal Of Computer Science, 3,2007.