Breaking LFSR Using Ant Colony Optimization

Hicham GRARI, Ahmed AZOUAOUI, Khalid ZINE-DINE

LAROSERI Lab.

Chouaib Doukkali University, FS

El Jadida, Morocco

grari.hicham@gmail.com, azouaoui.a@ucd.ac.ma, zinedine@ucd.ac.ma

***Abstract*—Ant Colony Optimization is a search meta-heuristic inspired by the behavior of real ant colonies and shown their effectiveness, robustness to solve a wide variety of complex problems. In this paper, we present a novel Ant Colony Optimization (ACO) based attack for cryptanalysis of Linear Feedback Shift Registers (LFSR). A known plaintext attack is used to discover the primitive polynomial used in the LFSR. Moreover, our approach allows us to find a linear equivalence of a given key stream through finding the feedback function. Experimental results prove that ACO can be used as an effective tool to break LFSR.**

Keywords—Cryptanalysis, Ant Colony Optimization, ACO, Linear Feedback Shift Registers, LFSR, Pheromone.

I. INTRODUCTION

In the present world, the need for security is constantly increasing, with the development and evolution of communications networks, especially the cryptology which has become a scientific discipline dealing with Confidentiality, Integrity, and Authentication.

Cryptology is the art and science of secure communication. It consists of two complementary fields: Cryptography and cryptanalysis. Cryptography is the science of building new powerful and efficient encryption and decryption methods. Cryptanalysis is the science and study of method to break cryptographic techniques i.e. ciphers. It is used to find loopholes in the design of ciphers.

Currently, the use of meta-heuristic in this area, is attracting more and more researchers. It may appear an efficient way to break complex ciphers. Ant Colony Optimization (ACO) [1] is a well-known meta-heuristics that was successfully used to produce approximate solutions for a large variety of optimization problems. Ant Colony Optimization is used to attack DES (Data Encryption Standard) by Salabat Khan, A.Armughan and Mehr Y Durrani [2]. Also, H.Grari, A.Azouaoui, K.Zine-Dine[3] proposed Cryptanalysis of Simple Substitution Ciphers Using ACO.

In the previous work, Iwona Polak and Mariusz Boryczka [4] attacked the Linear Feedback shift Register using genetic algorithm, concluding that their method is quite effective and promising to break LFSR. Maiya Din Saibal K. Pal S.K. Muttoo Anjali Jain [5] use a cuckoo search for analysis of LFSR, they found the parameters that are used to find the optimal solution. Furthermore, Ali A. , Hameed A. Younis, and Wasan S. Awad [6] found the shortest LFSR which generates a sequence of key stream knowing part of it, requiring less computational time and information based on genetics algorithm.

A Cryptanalytic Attack of Geffe Generator using genetic algorithm is done by Maiya Din, Ashok K. Bhateja and Ram Ratan [7] showing that divide-and conquer attack have been used for identifying initial states of LFSRs used in Geffe generator. Moreover, cryptanalysis of Nonlinear Stream Cipher Cryptosystem based on Improved Particle Swarm Optimization by Salim A. Abbas Al-Ageelee and Riyam N. J. Kadhum[8]

In this paper we introduce a new evolutionary way to attack Linear Feedback Shift Register (LFSR), using Ant Colony Optimization. We will show that our approach can be used as an effective tool to Break LFSR. The remainder of this paper is organized as follows: In the next section, we introduce Linear Feedback Shift Register. In section III, we present the basic and background of Ant colony optimization meta-heuristic. The full automated attack is given in Section IV, with experimental results in section V. Finally, conclusions are given in section VI.

II. LINEAR FEEDBACK SHIFT REGISTER

A stream cipher produces a pseudo-random sequence of bits which are Xored with the plaintext to produce the ciphertext. Many stream ciphers are based on the Linear Feedback Shift Register (LFSR), a crypto primitive that allow to generate a linear sequence whose properties approximate the properties of sequences of random numbers. LFSR require very less hardware and have high speed of operations.

A Linear Feedback Shift Register (LFSR) is defined by a feedback polynomial of degree L used as feedback function, and the length of the LFSR.

978-1-5386-4609-0/18/$31.00 ©2018 IEEE

The contents of the registers are shifted by one position at each clock. The left-most bit fed to the register is the result of mod-2 addition of bits corresponding to the non-zero coefficients of considered feedback polynomial. The right most bit is used to form the pseudorandom number sequence. All initial states should not be "*0*" because the LFSR would remain locked-up in these states.

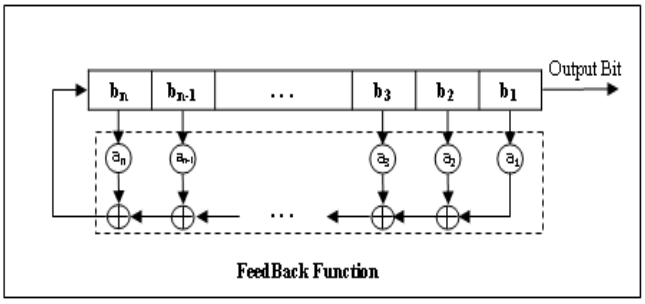

Fig. 1. Linear Feedback Shift Register (LFSR)

The scheme of LFSR is shown in fig. 1. The period of the shift register is the length of the output sequence before it starts repeating. When the feedback polynomial is primitive and of degree L the shift register is known as a maximum length LFSR. The output sequence of a maximum length LFSR is periodic with period 2^L-1.

III. Ant Colony Optimization

Optimization techniques have got a significant importance in determining efficient solutions of different complex and hard problems. In particular, Ant Colony Optimization [1], which represents a class of population based meta-heuristics, inspired by the foraging behavior of real ant colonies, which enables them to find shortest paths between food sources and their nest using a pheromone communication.

Ants initially explore randomly the environment surrounding their nest, when an ant finds food; it walks backs to the colony leaving behind a chemical pheromone trail that may depend on the quantity and the quality of the food. When choosing their way, other ants of the colony are expected to follow the path of greater pheromone trail, left by earlier ants, with higher probability; this indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. This simple idea is implemented by the ACO methods to resolve and address hard combinatorial problems such as traveling salesman problems, quadratic assignment problems, vehicle routing problems, or constraint satisfaction problems. The basic idea of ACO is to model the problem to solve as the search for a minimum cost path in a graph, and to use artificial ants to search for good paths.

The first ACO algorithm, called Ant System (AS) introduced by M. Dorigo, V. Maniezzo, A. Colorni [9] was applied to Travelling Salesmen Problem. Other ACO variants mostly differ in the rule used for the solution construction and the pheromone update, including Ant Colony System (ACS) presented by M. Dorigo, L. Gambardella. [10], and Min Max Ant System (MMAS) given by T. Stutzle and H. Hoos [11].

IV. Proposed Approach

To solve a combinatorial optimization problem via ACO, the main procedure is described as follows: at each cycle, every ant constructs a solution and then pheromone trails are updated. The algorithm stops iterating when a termination condition is met. Generally, the termination condition may be a certain number of iterations or the achieved result is close enough to lower (upper) bound.

The proposed algorithm LFSRACO based on the ACO to attack LFSR follows this standard scheme, whereas several components are dependent on the characteristics of our problem. In what follows, we describe the details about constructing a solution, and then explain how to define the heuristic information and the fitness function. Finally, we present the strategy to update pheromone trails.

A. Solution construction

Let $I = \{i_1, i_2 \ldots i_n\}$ the Initial State of the LFSR to be attacked with the length n, and $O = \{O_1, O_2 \ldots O_{2^n-1}\}$ the output of the LFSR.

In this paper, we adopt a known-plain text method to attack LFSR. In this method, cipher text and part of the plain text are known, which means that the output O is known (deduced from the plaintext and the associated ciphertext). the goal is to find the primitive polynomial used to generate the Output O .

In our approach, the search space is modeled as two layers of n vertices, where n is the length of LFSR. All vertices on the top layer vertices are equal to '1' and the bottom layer vertices are equal to '0'. Plus the start node N_0, and the final node N_{n+1}.

The search space is a grid of two rows and n columns. Every vertex in a column is connected to all the vertices in the next column.

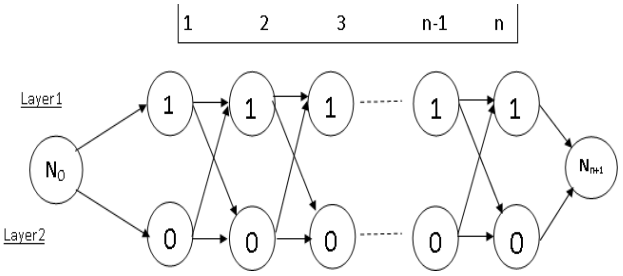

Fig. 2. Search Space for cryptanalysis of LFSR

An ant starts it tour from N_0. And move from left to right, its tour is finished at the last Node N_{n+1}. In a column, an ant can only select a single vertex during a particular tour. At the end, when the tour is completed, it will consist of *n-bit* long binary string. The '*1*' values constitute the taps in the primitive polynomial. Thus, the binary string is a candidate or guessed key that will be used in the LFSR to generate the output. Each ant constructs a solution using the function Probabilistic Stepwise Construction based on a probabilistic move of ants across the nodes. An ant moves from node i to node j with a probability $P(i,j)$ given by equation (*1*).

$$P(i,j) = \begin{cases} \frac{\tau(i,j)^{\alpha}\,\rho(i,j)^{\beta}}{\sum_{j\epsilon N_i}\tau(i,j)^{\alpha}\,\rho(i,j)^{\beta}} & if\ j\epsilon\ N_i \\ 0 & Otherwise \end{cases} \quad (1)$$

Two parameters are used to calculate the probability of moving from a state *i* to another state *j*; first, the amount of pheromone trail $\tau(i,j)$ on the connecting edge. And second the heuristic value $\rho(i,j)$ representing the attractiveness of the choice. The parameters α and β are influencing factors of pheromone and heuristic value, respectively.

B. Heuristic Value

Using an adequate heuristic information ρ is fundamental in making the algorithm find good solutions. Static and dynamic heuristic information are the main types of heuristic information used by ACO algorithms. In the static case, the values of ρ is computed once at initialization time and then remain unchanged throughout the whole algorithm's run. In the dynamic case, the heuristic information depends on the partial solution constructed so far and therefore has to be computed at each step of an ant' s walk.

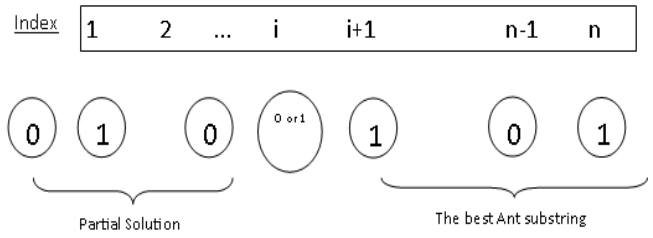

Fig. 3. Heuristic value calculation

The transition probability equation (1) needs a heuristic value calculation method from the problem domain as an efficient search methodology. In our approach, the candidate key with the best fitness value using equation (2) is saved as a global best ant (*BestA*). Now, in the subsequent iterations, at every decision point, ant uses heuristic value which is calculated as follows:

Let '*i'* the vertex in the i-th column (in the search space) as shown in Fig.3, in such position, an ant has to decide which node to move next. The 'j' is the vertex of next column where an ant can move, only two values are possible i.e. either '*0*'or '*1*'. And the binary string $E_{i,j}$ is a concatenation of three binary strings, $P_{(1\ to\ i)}$ is the partial solution constructed, and $BestA_{(i+2\ to\ n)}$ is the best ant binary substring from index '*i+2'* to '*n*'

$$E_{i,j} = \{P_{(1\ to\ i)} \mid j \mid BestA_{(i+2\ to\ n)}\}$$

So, the concatenated binary string $E_{i,j}$ becomes a guessed key which is evaluated using the fitness function ,and the obtained value $F(E_{i,j})$ is used as a heuristic value in (1).

C. Fitness Function

Fitness function (cost function) plays a very important role in a search algorithm to obtain the best solution(s) within a large search space. Fitness function is an indicator of how close the possible solution obtained during search is to the optimal solution. A good fitness function helps the search algorithm in exploring the search space more effectively and efficiently, while bad fitness function makes the search algorithm get trapped in a local optimum point.

Let $d_K(O,G)$ the Hamming's distance between output *O* of attacked LFSR, and the compared individual (*G*) generated using the solution to be assessed '*K'* (only the first 2n bits in output are considered). The fitness function is calculated as shown in (2) .

$$F(K) = \frac{2n - d_k(O,G)}{2n} \quad (2)$$

Where $d_k(O,G) = \sum_{i=0}^{2n-1}(Oi \oplus Gi)$

And Oi and Gi are the ith-bit in *O* and *G* respectively.

A fitness value equal to 1 indicates that the correct solution has been found (Hamming's distance between *O* and *G* is equal to 0).

D. Pheromone Update

Once each ant has constructed a solution. Only best ant solution in a particular Run (R) is allowed to update the pheromone trails. The ants also update the best ant information based on their tours fitness values. Pheromone over the edges constituting the tour of the best ant is updated using (3).

$$\tau_{i,j} = \tau_{i,j} + Q \times F(K_{Best}) \quad (3)$$

F is the fitness function and K_{Best} is the best key in a Run. And Q is some constant.

With the passage of time, the concentration of pheromone decreases due to diffusion affects; a natural phenomenon known as evaporation. It's used to decreases the influence of old pheromone in future decisions. So, with evaporation, chances to get stuck at local minima are minimized. This evaporation can be performed as:

$$\tau_{i,j} = \tau_{i,j} \times \sigma \ \{whith\ \sigma\ will\ be\ between\ 0\ and\ 1\} \quad (4)$$

E. Proposed Algorithm.

Fig.4 shows the LFSR crypto–primitive and its cryptanalysis by LFSRACO. The task of our algorithm is to recover the function feedback to be used in the LFSR to reproduce our plaintext.

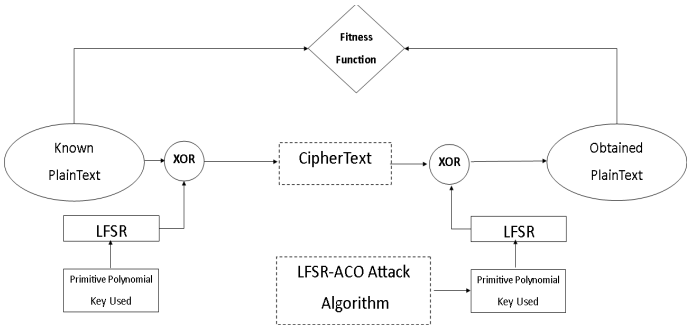

Fig. 4. Layout of LFSRACO attack Algorithm

The main steps of the LFSRACO attack algorithm are described as in Fig.5.

```
1. Perform initialization of pheromone
Repeat
2. Construct Solution of (N) ants by making
the decisions using probability equation (1)
3. Evaluate the Fitness of each ant solution
according to (2)
4. Update best ant Solution.
5. Update pheromone values on edges
constituting the best solution using (3)
6. Perform evaporation using equation (4)
Until (A maximum number of run (R) have been
attained or threshold of Fitness Function is
reached).
```

Fig. 5. LFSRACO Algorithm

V. Experimental Results and Discussions

Values of parameters assumed in this paper such as α, β (weight of pheromone and heuristic value), N (Number of Ants), R (Number of Run), '*Q*' and σ were fine-tuned by a combination of several experiments in order to optimize the cryptanalysis process. The default value of the parameters was *α=1*, *β=1*, *Q=2*, *σ=0.95* and $\tau_0 = 5$ (initial pheromone value). We have implemented our algorithm with C++ language.

However, for the comparison purpose we ran our proposed LFSRACO algorithm on the same example used by Iwona Polak and Mariusz Boryczka [4], based on Genetic Algorithm. For the experiments there were four LFSRs chosen, which differ in length and number of taps – in order to get general result. The studied set consisted of four registers of maximum-length, used in practice:

- ***P1*** **:** $x^{16} + x^5 + x^4 + x^3 + 1$ (USB 3.0 scrambler).
- ***P2*** **:** $x^{22} + x^{21} + 1$ (one register from A5/1 GSM encryption).
- ***P3*** **:** $x^7 + x^3 + 1$ (CRC-7 / telecom systems, MMC).
- ***P4*** **:** $x^{32} + x^{26} + x^{23} + x^{22} + x16^6 + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$ (CRC-32-MPEG2, CRC-32-IEEE 802.3) .

Every LFSR was tested with an initial state generated randomly, for every register.

A. Key space analysis

The first aim of experiments is to determine the number of ants to be used to find de real solution in a minimum search space. The results obtained after carrying experiments are illustrated in Table I. The objective of tests is to find the best values of *N* (Number of Ants used). Table I shows the number of solution searched before locating the reel one, and the average number of run needed for every value of *N*. The average value is presented in Table I (for USB 3.0 scrambler).

TABLE I. Experimental Results for different values of N

Primitive Polynomial	*Number of ant used (N)*	*Number of run needed (R)*	*Number of solution searched*
$x^{16} + x^5 + x^4 + x^3 + 1$	5	132	660
	10	80	800
	15	34	544
	20	31	620
	30	22	660
	40	21	840

As shown in Table I, with a small number of ants (*N*<15), the real solution is founded after examining 660 elements with *N*=5 and 800 elements when *N*=10, the number of Ants is not enough large; thus the best solution found on a Run (which will be used for the pheromone update) is not enough good, which penalizes the convergence of the algorithm to the right solution.

As noted in the Table I, with the values *N* = 15 we need 34 Run de locate the real solution, so it founded in a minimum fraction of the search space. The maximum fitness value is reached after checking 544 elements.

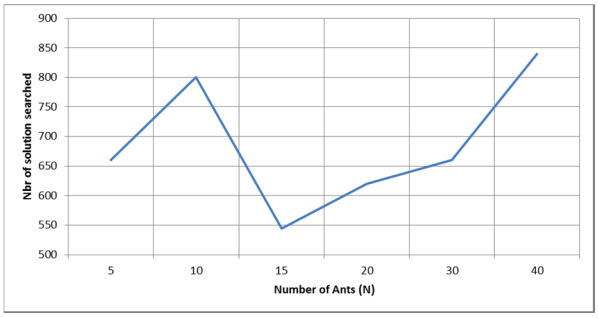

Fig. 6. Number of solution searched evolution

Fig.6 shows the evolution of the number of elements browsed before locating the real one under the number of ants used (*N*), when the number of Ants Exceeds *15*, the search space increases rapidly.

B. Comparaison result with variation in the number of taps

The performance of the LFSRACO Algorithm is measured with different values of *NT* (Number of taps in the primitive polynomial). A feedback tap correspond to a non-zero coefficient of considered primitive polynomial.

TABLE II. EXPERIMENTAL RESULTS FOR DIFFERENT VALUEOF TAPS AMOUT .

Number of Taps (NT)	*Number of Key searched*	*Fitness value*
2	364	1
4	544	1
6	544	1
8	720	1
10	840	0.9
12	1065	0,85

The second part of analysis is to compare with different value of *NT* (Number of Taps) using an LFSR length of 16. Comparison results are illustrated in TABLE II, as we can observe the real Key is located rapidly when using a small taps, the number of key searched increase as the taps amount increase. And the best results were obtained for registers with small amount of taps (Fitness value equal to 1). Primitive polynomials which describe them have a small number of coefficients and therefore they are called low density polynomials. But also for LFSRs with more taps the results were satisfying (90% output bits match for 10 taps and 85% for 12 taps).

C. *Comparaison with GA*

In this part, a comparison with genetics algorithm [4] results is done, using different initials states of LFSR, obtained results are reported in TABLE III. As we can see the maximum fitness function obtained in our approach is better than GA [4] in all case of LFSR. Also we can observe that the value of fitness function decrease as degree of primitive polynomial increases.

TABLE III. COMPARISON OF CRYPTANALYTIC RESULTS OBTAINED WITH GA AND LFSRACO.

LFSR	*Max Fitness Function GA [4]*	*Max Fitness Function ACO*
P1	1	1
P2	0,679	1
P3	1	1
P4	0,73	0,81

For every initial state of the LFSR the results were similar, which shows that these results are independent of initial sequence of LFSR.

D. *Parametric Sensitivity Analysis*

In order to analyze the parametric sensitivity, we chose ***P1*** with 6 taps amount, the number of Ant (*N*) is set to *15* and β is fixed to *1*. With good parameter settings, the long-term effect of the pheromone trails is to progressively reduce the size of the explored search space so that the search concentrates on a small number of promising areas. Yet, this behavior may become undesirable, if the concentration is so strong, as we can see in the Fig.7 with *α=2*, that it results in an early stagnation of the search. In such an undesirable situation the system has ceased to explore new possibilities and no better solution is likely to be found anymore.

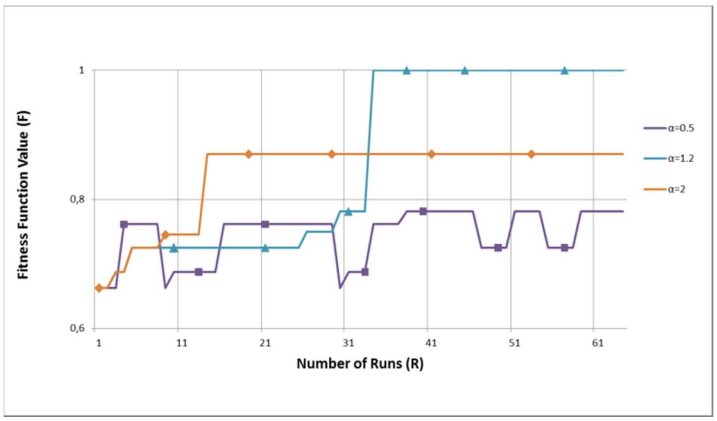

Fig. 7. Fitness function evolution for different values of α

Experimental result indicates that for LFSRACO good parameter settings are those that allowing a reasonable balance between a too emphasis focus of the search process (illustrated in fig7 with *α=2)*, which may lead to stagnation behavior, and a too weak guidance of the search, which can cause excessive exploration, this behavior is illustrated in Fig.7 when α is equal to 0.5.The best results are archived with a value of α=*1.2* in our experiments.

VI. CONCLUSION

This article proposed a new approach for cryptanalysis of LFSR cryptosystem using ant colony optimization. The experimental results show that our approach allows finding a linear equivalence of a given key stream through finding the feedback function. The achieved results are better than GA's results. One main disadvantage of Ant Colony optimization techniques is its large sensitivity to parameter variations. Although a fine tuning of these parameters can be done by experimental trials.

ACO provides a very powerful tool for the cryptanalysis of LFSR Crypto-primitive. Future work includes the use of a local search technique in order to improve the convergence speed and success probability of our algorithm. Another future work might be interesting to applying our approach to break non Linear Feed Back Shift Register or RC4 stream cipher.

I. References

[1] M. Dorigo. Optimization, ―Learning and Natural Algorithms‖. PhD thesis, 1992.

[2] Salabat Khan, Armughan Ali and Mehr Yahya Durrani "Ant-Crypto, a Cryptographer for Data Encryption Standard" IJCSI, Vol. 10, Issue 1, No 1, January 2013.

[3] Hicham Grari,Ahmed Azouaoui,Khalid Zine-Dine "A Novel Ant Colony Optimization Based Cryptanalysis of Substitution Cipher" International Afro-European Conference for Industrial Advancement AECIA 2016.

[4] I. Polak and M. Boryczka, "Breaking LFSR Using Genetic Algorithm," in Computational Collective Intelligence.Technologies and Applications, Berlin Heidelberg, 2013, pp. 731-738.

[5] Din, M., Pal, S.K., Muttoo, S.K., Jain, A.,Applying Cuckoo Search for Analysis of LFSR based Cryptosystem, Perspectives in Science 2016.

[6] Abd, A.A., Younis, H.A., Awad, W.S.: Attacking of stream Cipher Systems Using a Genetic Algorithm. Journal of the University of Thi Qar 6, 1–6 (2011).

[7] M. Pant et al. (eds.),"Cryptanalysis of Geffe Generator Using Genetic Algorithm", Proceedings of the Third International Conference on Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing 259,Springer India 2014.

[8] A. Abbas Al-Ageelee, Salim ,N. J. Kadhum, Riyam "Cryptanalysis of Nonlinear Stream Cipher Cryptosystem based on Improved Particle Swarm Optimization" International Journal of Applied Information Systems,2017.

[9] M. Dorigo, V. Maniezzo, A. Colorni, “The ant system: Optimization by a colony of cooperating agents”. IEEE Transactions on Systems, Man, and Cybernetics-Part B, , 1996, 26(1), 29-41.

[10] M. Dorigo, L. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, 1997, 1(1), 53 -66.

[11] T. Stutzle and H. Hoos, “Improvements on the ant system, introducing the MAX-MIN ant system,” in Proc. ICANNGA97—Third Int. Conf.Artificial Neural Networks and Genetic Algorithms. Wien, Germany: Springer-Verlag, 1997.