Chapter 10
Evolutionary Decryption of Chaotically Encrypted Information

Ivan Zelinka and Roman Jasek

Abstract. This chapter introduces the concept of decryption of chaotically encrypted information. Five evolutionary algorithms have been used for chaos synchronization here: differential evolution, self-organizing migrating algorithm, genetic algorithm, simulated annealing and evolutionary strategies in a total of 15 versions. The main aim was to ascertain if evolutionary algorithms are able to identify the "key" (control parameter) of the chaotic system, which was used to encrypt information. The proposed scheme is based on the extended map of Clifford strange attractor, where each dimension has a specific role in the encryption process. Investigation consists of one case study. All the algorithms was 100 times repeated in order to show and check robustness of the proposed methods and experiment configurations. All data were processed in order to get summarized results and graphs.

10.1 Introduction

Chaotic systems are extremely sensitive to initial conditions and this feature can be very helpful in the field of cryptography. Various encryption schemes use chaotic systems for encryption key generation and this key is then used for pixel permutation and pixel diffusion. But chaotic systems and their maps can be used directly

Ivan Zelinka
Tomas Bata University in Zlin, Faculty of Applied Informatics, Nad Stranemi 4511, Zlin 76001, Czech Republic
and
VSB-TUO, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: zelinka@fai.utb.cz

Roman Jasek
Tomas Bata University in Zlin, Faculty of Applied Informatics, Nad Stranemi 4511, Zlin 76001, Czech Republic
e-mail: jasek@fai.utb.cz

I. Zelinka et al. (Eds.): Evolutionary Algorithms and Chaotic Systems, SCI 267, pp. 329–343.
springerlink.com

for encryption purpose. The proposed scheme is based on the extended map of the Clifford strange attractor, where each dimension has a specific role in the encryption process. Two dimensions are used for pixel permutation and the third dimension is used for pixel diffusion. The theoretical and simulation results prove many properties of this scheme such as large key space and high security.

Image is a multimedia signal providing the most information to a person. For this reason the question appears as to which way can be signal be secured against unauthorized reading e.g. in medicine or military fields. Position permutation and diffusion of the pixels belongs to basic methods of image encryption. Their combination leads to better security against known attacks and is very often has found practical usage. However, these methods remains open for various encryption algorithms and that is why the knowledge of chaotic systems can be useful. These systems are extremely sensitive to initial conditions and thus they are suitable candidates in the field of cryptography. Many papers have been written on this theme for that very reason.

Chaos-based image encryption is discussed in detail in the previous chapter as well as in [14]. Most of the papers have described the process of the generation of the time series based on a chaotic map. These series are used for the creation of the binary sequence as an encryption key and pixels of plain image are then rearranged and XOR operated with this key. For example in [6] three logistic maps are used in key stream generator and this improved the linear complexity of key stream. Each paper proposed various type of key generator or improvements of chaotic encryptions in terms of security and speed [17], [2], [10] but only a few of them show a different way of encryption, such as using hyper-chaotic system for confusing the relationship between the plain-image and the cipher-image [7], Lorentz system for key-stream generation [5] or S-box algebraic operations [11], [1]. When other methods such as image encryption in wavelet domain proposed in [16] are used with chaos-based encryption scheme, we should expect interesting results. The results for the audio signals are presented in [8], where the wavelet coefficients were modified and the audio signal becomes inaudible.

The aim of this chapter is to show that evolutionary algorithms are capable, at least under strong simplifications, of decrypting information, which has been encrypted by chaotic dynamics. We have used results from [9]. The proposed scheme in this chapter uses the formula of a strange attractor for encryption purposes. It does not create any encryption key but uses attractor map for pixel permutation and diffusion directly. Parameters of attractor map play the role of encryption keys here and key-space is very large due to their non-integer character. Chaos based encryption has been done by means of so called Clifford system (attractor), which is depicted in Fig. 10.1 - 10.3.

$$\begin{aligned} x_n &= \sin(ay_n) + c\cos(ax_n) \\ y_n &= \sin(bx_n) + d\cos(by_n) \end{aligned} \tag{10.1}$$

$$\begin{aligned} x_n &= \sin(ay_n) + c\cos(ax_n) \\ y_n &= \sin(bx_n) + d\cos(by_n) \\ z_n &= \sin(ey_n) + f\cos(ez_n) \end{aligned} \tag{10.2}$$

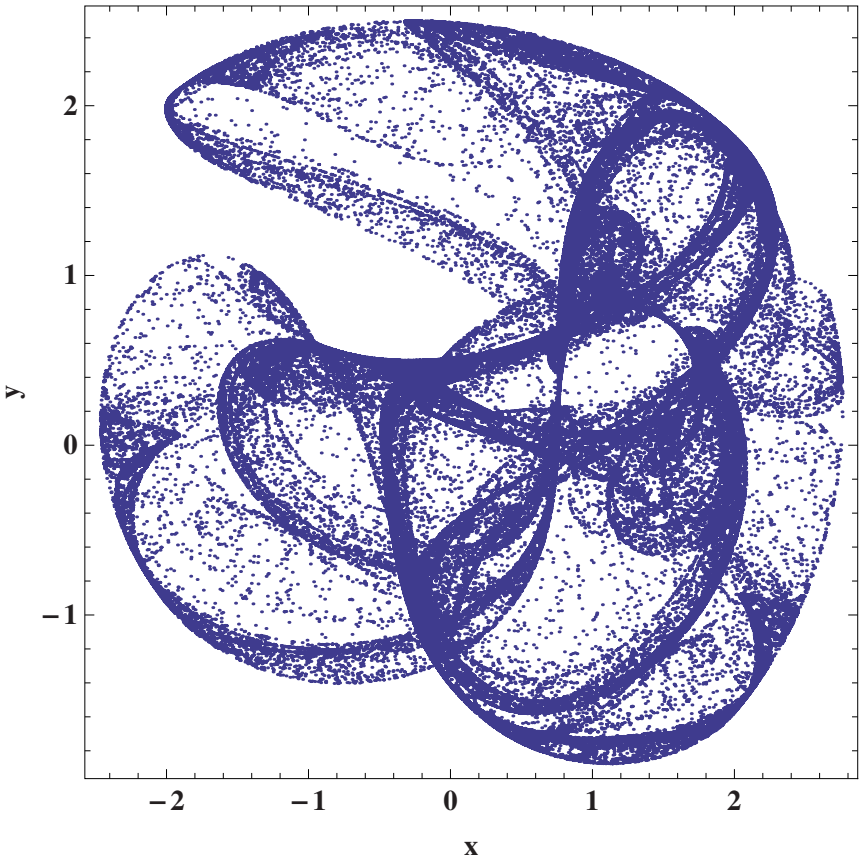

Fig. 10.1 Clifford attractor according to eq. (10.1).

In the research work of [9], encryption and its robustness with Clifford chaotic system use has been tested. The tested "message" for encryption were two pictures, see Fig. 10.4 and Fig. 10.5. In the encryption scheme, the Clifford attractor was used. It belongs to the trigonometric strange attractors and is described by eq. (10.1) and eq. (10.2). Fig. 10.6 shows the flowchart of this encryption scheme. The main parameters (key) are used for the iterative process of the Clifford system. Pixel of image is used as the initial value of Clifford system. New positions and modification value is gained after iterations and quantization. These positions are then used for pixel permutation and the modification value is XOR operated with original pixel value and the value of the previous pixel. Encrypted pixel is gained this way.

When the proposed schema of encryption (for more see [9]) is used, then one can obtain a picture as in Fig. 10.7. Histograms related to the original Lena and its encrypted version are depicted in Fig. 10.8 and Fig. 10.9. Both kind of pictures shows (of course there is also rigorous mathematical background) that the picture is really well encrypted.

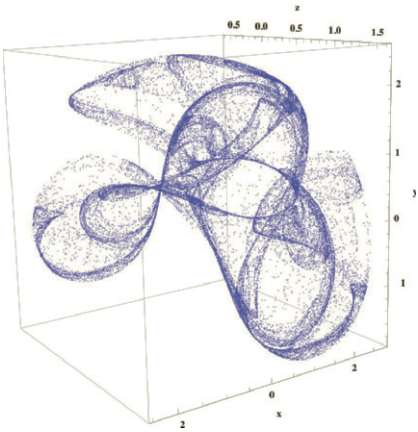

Fig. 10.2 Clifford attractor according to eq. (10.2)...

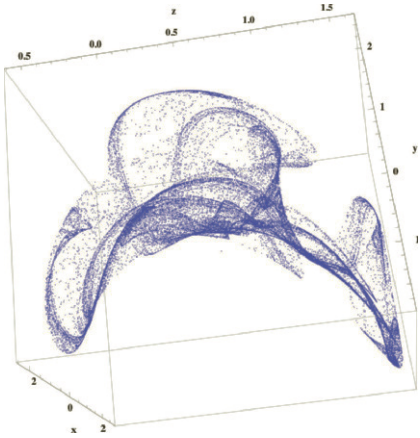

Fig. 10.3 ... and another 3D view.

Fig. 10.4 Lena

Fig. 10.5 Man with camera.

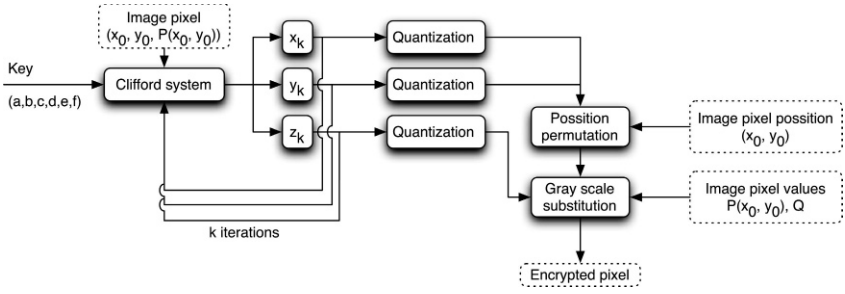

Fig. 10.6 Scheme with encoding.

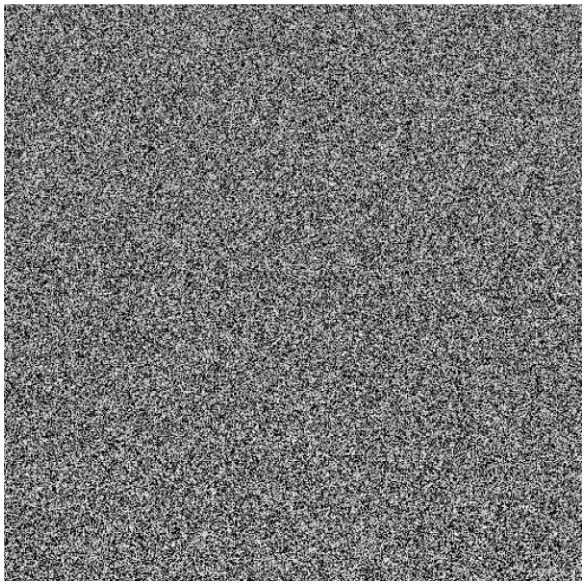

Fig. 10.7 Encrypted Lena.

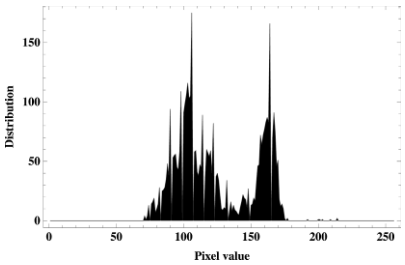

Fig. 10.8 Histogram of original Lena figure...

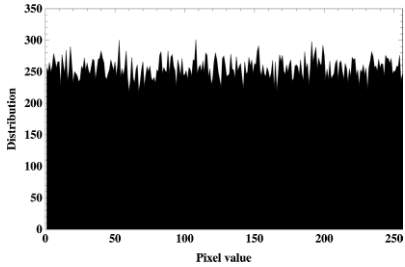

Fig. 10.9 ...and after encryption.

10.2 Motivation

Motivation of this research is very simple. Chaos based encryption is under intensive research attention today and encryption itself is vitally important for various communities, from industrial to government. We would like to ascertain if it is possible to identify key used for encryption by means of evolutionary algorithms.

Good encryption scheme must be resistant against any brute-force attacks, so the key space must be too large. The total precision of a common PC processor is 16 decimal digits, therefore the number of different combinations of one parameter is 10^{16} and it corresponds approximately to 253 size key space. Six attractor parameters are used in the proposed scheme; hence the key space is enlarged to 2^{318}. Also,

the number of iterations k of Clifford system eq. (10.2) and the number of encryption rounds m can be considered as keys. Thus, the key space of the proposed scheme is large enough to make the classical brute-force attack infeasible. Table 10.1 shows comparison of key spaces of various encryption schemes. Our proposed encryption scheme has the largest key space.

Table 10.1 Key space comparison

Encryption scheme	Key space
Proposed in [9]	2^{318}
[14]	2^{128}
[6]	2^{158}
[7]	2^{232}
[8]	2^{256}

10.3 Selected Evolutionary Algorithm – A Brief Introduction

For the numerical and symbolic experiments described here, stochastic optimization algorithms such as Differential Evolution (DE) [15], Self Organizing Migrating Algorithm (SOMA) [18], Genetic Algorithms (GA) [12], Simulated Annealing (SA) [13], [4] and Evolutionary Strategies (ES) [3] were selected. Description of all selected algorithms can be found in the mentioned references or in Chapter 6.

10.4 Evolutionary Decryption

10.4.1 Used Hardware, Problem Selection and Case Studies

Evolutionary decryption in this case study has been done on a specialized grid computer. This grid computer consist of two special Apple servers (for pictures, see Chapter 6). A total of 78 CPUs were available for computation. This grid has been used for calculations so that each CPU has been used like a single processor and thus a rich set of statistically repeated experiments were possible which are not time dependent. Typical parallel computing has been avoided in experiments described here.

10.4.2 Cost Function

The Lena picture has been encrypted by eq. (10.2) with parameters defined as: a = -1.85, b = 1.48, c = -1.55, d = -1.87, e = -4.32, f = 0.63. In [9] the encrypted picture was successfully tested for key sensitivity. The set of keys in [9] are very similar, only one parameter is different with minimal divergence ($b = 1.4800001$). This small difference is enough to get after the decryption of a noisy picture as in Fig. 10.7. To test key sensitivity, which is based on encryption of the "Lena"

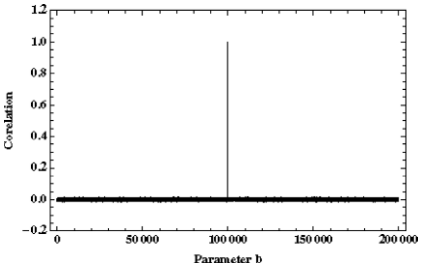

Fig. 10.10 Correlation of Lena picture, sharp peak at position 100000 represent solution - right estimation of the parameter b.

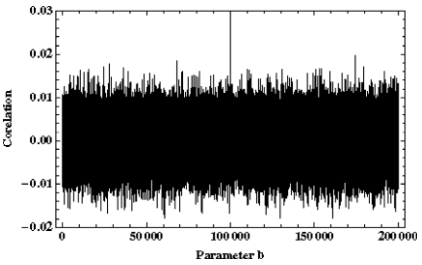

Fig. 10.11 Detail view.

image by mentioned setting, cross-correlation of their encrypted forms was then computed. Fig. 10.10 and Fig. 10.11 shows cross-correlation of images encrypted by these two different set of keys. Correlation value does not exceed 0.02. This implies very low correlation and very low similarity of images and their pixels. In general, adjacent pixels of the most plain-images are highly correlated. One of the requirements of an effective image encryption process is to generate encrypted image with low correlation of adjacent pixels. Correlation between two horizontally, vertically and diagonally adjacent pixels of original and encrypted image was also analyzed in [9].

The fitness (cost function) has been calculated very simply. In fact it was a simple search for such a value of parameter b so that cross-correlation value was equal to 1, i.e. right parameter of b was found. In total, 6 parameters were used like the key. To simplify the situation and make calculation time shorter, only parameter b has been selected for evolutionary estimation. Also, another important note should be mentioned here: numerical accuracy. Parameter b has been estimated with different level on numerical precision. The largest was for $\Delta b = 1 \times 10^{-15}$, which was used to generate in total 200 000 data points around the right value of b. For this type of precision, a tiny region of parameter b has been explored.

Comparing to another case studies, reported in this book, similarity between two kind of behavior and other parameters was not measured. Only similarity, via cross correlation, has been measured. Due to the chaotic nature of cost function landscape (Fig. 10.11), it was near to random search, thanks to the sophisticated search process. The cost function can be simply described by eq. (10.3). From that viewpoint, it behaves as a blind search, because on cross-correlation graphs, the general trend is completely flat.

$$\begin{aligned} &if \\ &\quad b\ is\ such\ that\ cross-correlation = 1\ then\ stop \\ &else \\ &\quad continue\ in\ evolution \\ &end \end{aligned} \tag{10.3}$$

10.4.3 Parameter Setting

The control parameter settings have been found empirically and are given in Tables 10.2 - 10.7. Number of cost function evaluations was not an objective in this study. Only one objective was there - to successfully estimate part of the encrypting key. We would like to note that settings of all used algorithms here, has been based on our preliminary experiences and certainly can be improved. However, this topic is quite numerically time consuming, so we let this topic open for future research.

Table 10.2 Algorithms abbreviation

Algorithm	Version	Abbreviation
Differential Evolution	DEBest1JIter	D1
	DEBest2Bin	D2
	DELocalToBest	D3
	DERand1Bin	D4
	DERand1DIter	D5
	DERand2Bin	D6
Evolutionary strategies	(μ,λ)	ES1
Evolutionary strategies	(μ+λ)	ES2
Genetic Algorithm		G
Simulated annealing with elitism		SA1
Simulated annealing without elitism		SA2
SOMA	AllToAllAdaptive	S1
	AllToAll	S2
	AllToOne	S3
	AllToOneRandomly	S4

Table 10.3 DE setting.

Parameter	Value
NP	500
F	0.9
CR	0.3
Generations	500
Individual Length	1

Table 10.4 ES setting.

Parameter	Value
μ ,λ	500
σ	1
Iterations	100
Individual Length	1

Table 10.5 GA setting.

Parameter	Value
Population size	500
Mutation	0.4
Generations	561
Individual Length	

Table 10.6 SA setting.

Parameter	Value
No. of particles	500
σ	0.5
$k_{\max}$	66
T_{min}	0.0001
T_{max}	1000
α	0.9
Individual Length	1

Table 10.7 SOMA setting.

Parameter	Value
PathLength	3
Step	.11
PRT	1
PopSize	500
Migrations	10
MinDiv	-0.1
Individual Length	1

All algorithms (SOMA, DE, SA, GA, ES) have been evaluated 100 times in order to find the optimum of both case studies. The primary aim of this comparative study is not to show which algorithm is better and worst, but to show whether evolutionary synchronization can be used for decryption of chaotically encrypted information. Comparing to the other case studies reported in this book, population size is in this application is set to quite a high number (500). This number has not been selected randomly, but was obtained after a very simple set of simulations. Before the population size has been determined, a simple investigation on how dependent successful decryption is on population size was conducted. Fig. 10.12 captures this dependance. Straightforward dependance on population size is clearly visible there. When population size is more than 300, then evolutionary algorithms are capable of finding a larger number of successful decryptions, compared to unsuccessful ones. This is the reason why 500 individuals has been set for each algorithm.

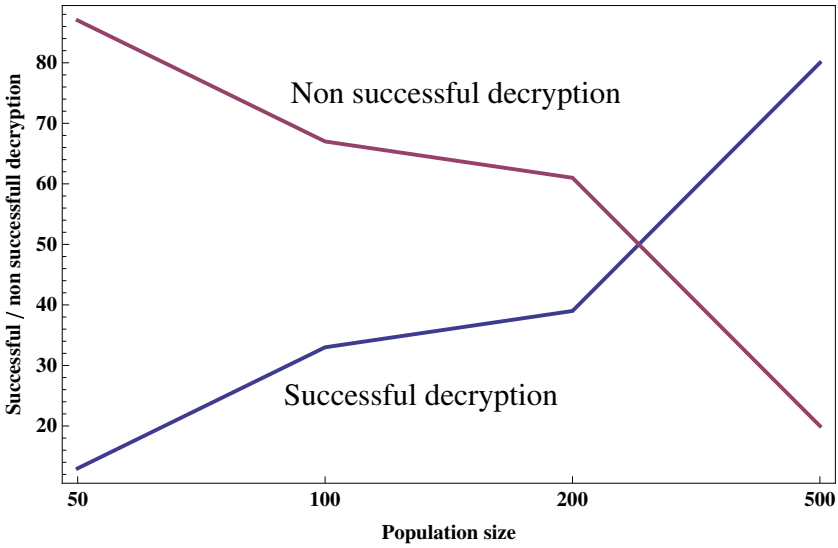

Fig. 10.12 Dependance of the number of successful decryptions on population size.

10.4.4 Experimental Results

Outputs of all simulations is depicted in Fig. 10.13 and Fig. 10.14, which shows results of all 100 simulations. In Fig. 10.13 one can see minimal, average as well as maximal number of cost function evaluations to get successful decryption. We have

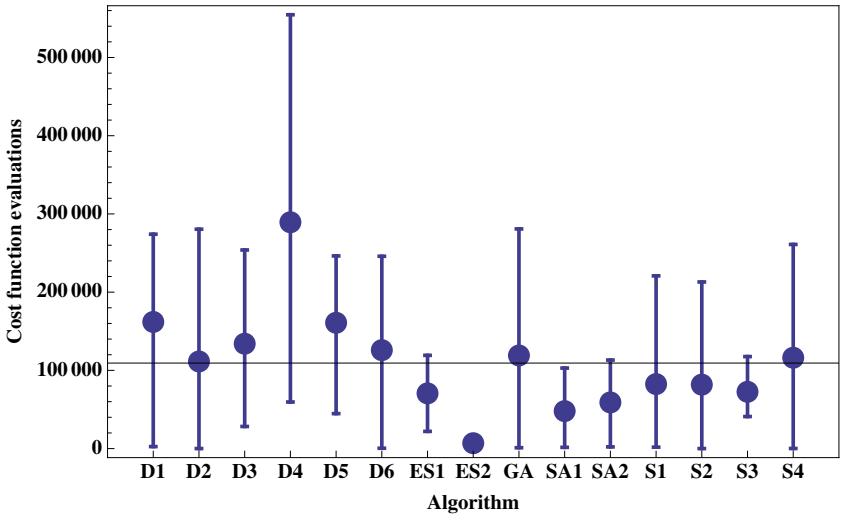

Fig. 10.13 Number of cost function evaluations needed to successfully decrypt figure of Lena. Horizontal line is an average of all.

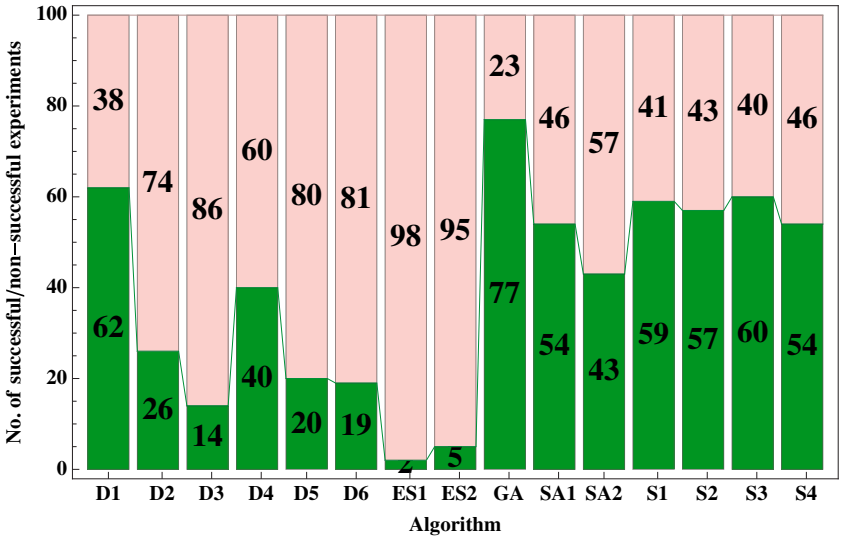

Fig. 10.14 No. of successful/non-successful decryptions. Each bar is divided into two parts. The upper part represent number of non-successful decryption, the lower one successful decryption.

to note that in this figure are reported **only** positive results. All results (positive / negative; successful / non successful;...) are reported in Fig. 10.14. All informations together are summarized in Tables 10.8 - 10.9.

Table 10.8 Experiment summarization, part 1.

Algorithm	D1	D2	D3	D4	D5	D6	ES1	ES2
Cost function evaluations see Fig. 10.13								
Minimum	2552	51	28282	59560	44722	460	22001	192
Average	161918	111398	134070	289378	160836	125863	70601	6904
Maximum	274100	280484	253892	554632	246312	245866	119201	12009
Total for each algorithm	10038904	2896337	670348	10706974	1769198	1636222	141202	55231
Decryption see. Fig 10.14								
Non-successful	38	74	86	60	80	81	98	95
Successful	62	26	14	40	20	19	2	5
Total	100	100	100	100	100	100	100	100

Table 10.9 Experiment summarization, part 2.

Algorithm	G	SA1	SA2	S1	S2	S3	S4
Cost function evaluations see Fig. 10.13							
Minimum	1112	1586	2169	1828	10	40980	135
Average	119028	47897	59022	82466	81756	72597	116153
Maximum	280837	102941	113166	220822	213012	117624	260978
Total for each algorithm	8808093	2203247	1947736	4865512	4660108	435580	5226899
Decryption see. Fig 10.14							
Non-successful	23	46	57	41	43	40	46
Successful	77	54	43	59	57	60	54
Total	100	100	100	100	100	100	100

10.5 Conclusion

In this chapter, we have studied the possibility of evolutionary decryption of encrypted information, based on chaotic systems. Compared to other case studies, only one "case study" is reported here as given in figures above (Lena decryption). All details about it are discussed below. As a conclusion, summarizing all previous informations, it can be stated that:

- **Usability of evolutionary algorithms.** In experiments reported here, two **very strong** simplifications has been taken into consideration. The first one was that only one parameter b of 6 (a - f from eq. (10.2)) has been estimated. The second one is partially done by restriction based on computer and used software accuracy. Part of decryption (cross-correlations "landscape", where EAs were searching for optimal value of b) has been made in C++ programming language, thus parameter b has been estimated with level of numerical precision of $\Delta b = 1 \times 10^{-15}$. Further, evolutionary search has been restricted to a **tiny** region which was used to generate in total - only 200 000 data points around the right value of b. For this kind of precision, a tiny region of parameter b has been explored. From figures and tables above, it seems that evolutionary search was quite successful, however, it is very logical to expect that if more than one parameter in a wider intervals would be estimated, then evolutionary algorithms would certainly fail.
- **Effectiveness** of used algorithms and proposed methods can be evaluated from two viewpoints. The first one is, that we can evaluate each algorithm separately, according to Fig. 10.14 and Tables 10.8 and 10.9. If we take into consideration the fact that there was 200 000 possible points to search through, it seems that evolutionary algorithms give good performance, because according to Fig. 10.13 all average values (excluding one - DE4) are below 200 000, which is better than a "brute force" method. On the other side, it is important to note that this conclusion is valid only when when brute force (i.e. all

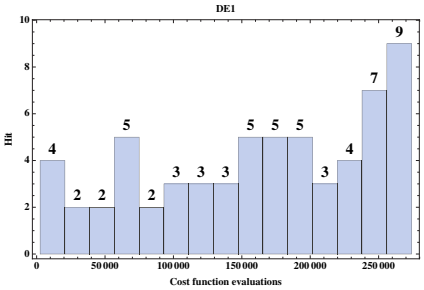

Fig. 10.15 Histogram of successful decryptions for DE1.

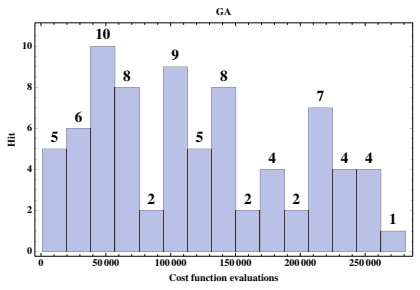

Fig. 10.16 Histogram of successful decryptions for GA.

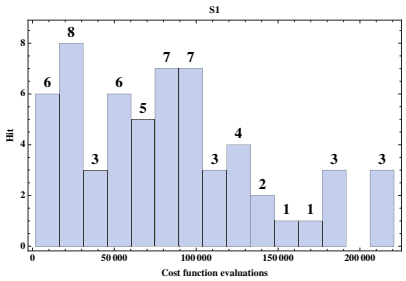

Fig. 10.17 Histogram of successful decryptions for S1.

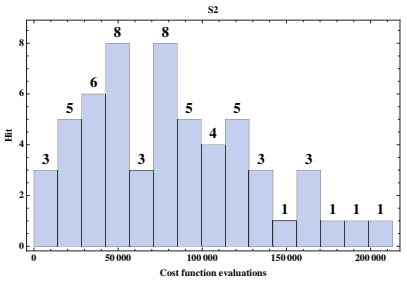

Fig. 10.18 Histogram of successful decryptions for S2.

possible solutions are investigated) is used so that each solution is randomly selected. If each solution would be selected consequently in order (i.e. the first, the second, ...), then the performance of evolution would be overwhelmed on 100 000 (remember, that is the position of the right value of b). If random search is compared, then result would be similar. Algorithms with values below 100 000 are ES1, ES2, SA1-S3.

The second point of view is that when we evaluate all results of all algorithms together, as reported in Fig. 10.14. In that case, unfortunately it appears that average effectiveness is almost random.

- **Performance-** misleading conclusion can also be made when we forget that **only** positive results are repoted in Fig. 10.13. For example, algorithm ES2 seems to be absolutely excellent, however when one takes a closer look in Fig. 10.14, then it is easily visible that values reported in Fig.10.13. are based on **5 successful** results. Cost function evaluations in all 5 cases are low and probably are a matter of "randomness", i.e. from only 5 cases we can hardly deduced any statistics. Another point of view can be obtained when separate histograms are reported for each algorithm, for example in Figs. 10.15 - 10.18. It is clearly visible that some algorithms has found more positive results

below 100 000 and 200 000, and for some of them it is just simply a uniform distribution.

- **Ability to locate extreme on chaotic landscape.** All results plotted and discussed above shows one quite important preliminary fact. More or less, evolutionary algorithms are capable to find an extreme on chaotic landscapes, which does not contain so called trend (general trend, average trend, ...), i.e. such a landscape is completely flat. To get more rigorous conclusion, it is however needed to do more extensive study in various chaotic landscapes.

Based on all results and their analysis, we can conclude, that chaos based encryption is still very safe and is probably not solvable by such techniques as evolutionary algorithms. On the other side, results reported here seems to be an inspiration (at least for us) for more extensive study on how effective evolutionary decryption is, when more individuals and decrypted parameters are taken into account.

Acknowledgements. This work was supported by grant No. MSM 7088352101 of the Ministry of Education of the Czech Republic and by grants of the Grant Agency of the Czech Republic GACR 102/09/1680.

References

1. Asim, M., Jeoti, V.: On Improving an Image Encryption Scheme based on Chaotic Logistic Map. In: ICIAS (2007)
2. Asim, M., Jeoti, V.: Hybrid Chaotic Image Encryption Scheme based on S-box and Ciphertext Feedback. In: ICIAS (2007)
3. Beyer, H.-G.: Theory of Evolution Strategies. Springer, New York (2001)
4. Cerny, V.: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. J. Opt. Theory Appl. 45(1), 41–51 (1985)
5. Fu, C., Zhang, Z., Cao, Y.: An Improved Image Encryption Algorithm Based on Chaotic Maps. In: ICNC (2007)
6. Fu, C., Zhang, Z., Chen, Z., Wang, X.: An Improved Chaos-Based Image Encryption Scheme. In: ICCS 2007. Springer, Berlin (2007)
7. Gao, T., Chen, Z.: A new image encryption algorithm based on hyper-chaos. ScienceDirect (2007)
8. Giesl, J., Vlcek, K.: Audio signal encryption in wavelet domain based on chaotic maps. In: Mendel 2008, Brno (2008)
9. Giesl, J., Vlcek, K.: Image Encryption Based on Strange Attractor. ICGST-GVIP Journal 9(2), 19–26 (2009)
10. Gu, G., Han, G.: An Enhanced Chaos Based Image Encryption Algorithm. In: ICICIC (2006)
11. He, X., Zhu, Q., Gu, P.: A New Chaos-Based Encryption Method for Color Image. Springer, Berlin (2006)
12. Holland, J.: Adaptation in Natural and Artificial Systems. Univ. Michigan Press, Ann Arbor (1975)
13. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)
14. Mao, Y., Chen, G.: Chaos-Based Image Encryption. Springer, Berlin (2003)

15. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999)
16. Seo, Y.-H., Kim, D.-W., Yoo, J.-S., Dey, S., Agrawal, A.: Wavelet Domain Image Encryption by Subband Selection and Data Bit Selection. Springer, Berlin (2003)
17. Wong, K., Kwok, B., Law, W.-S.: A Fast Image Encryption Scheme based on Chaotic Standard Map. Springer, Berlin (2006)
18. Zelinka, I.: SOMA – Self Organizing Migrating Algorithm. In: Babu, B., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, pp. 167–218. Springer, New York (2004)