# Utilizing Evolutionary Computation Methods for the Design of S-Boxes

[1,3]Elena C. Laskari, [2,3]Gerasimos C. Meletiou and [1,3]Michael N. Vrahatis
[1]Computational Intelligence Laboratory, Department of Mathematics,
University of Patras, GR–26110 Patras, Greece
[2]A.T.E.I. of Epirus, P.O. Box 110, GR–47100 Arta, Greece
[3]University of Patras Artificial Intelligence Research Center (UPAIRC)
University of Patras, GR–26110 Patras, Greece
elena@math.upatras.gr, gmelet@teiep.gr, vrahatis@math.upatras.gr

## Abstract

*Among the most important components of many contemporary ciphers are the substitution boxes (S-boxes) and a great amount of research is devoted to their study. In this contribution, a new methodology for designing strong S-boxes is proposed and two Evolutionary Computation methods, the Particle Swarm Optimization and the Differential Evolution algorithm are employed to tackle the problem at hand. The obtained results are promising and indicate that this novel approach is effective.*

## 1 Introduction

Substitution boxes (S-boxes) are basic components of symmetric key cryptosystems. Essentially, they are nonlinear mappings that take as input a number of bits and transform them into some number of output bits. S-boxes are of major importance in cryptography as they are used to provide the property of confusion to the corresponding cryptosystems and, in some cases, they comprise their only nonlinear part. As the security of the cryptosystems utilizing S-boxes mainly depends on their choice, a great amount of research is devoted into the design of good S-boxes. Recently, the performance of evolutionary heuristics, such as hill climbing [8], Genetic Algorithms [9] and Simulated Annealing [2], on the design of strong regular S-boxes was studied with very promising results.

Evolutionary Computation (EC) methods are inspired from evolutionary mechanisms such as natural selection and social and adaptive behavior. Most commonly used paradigms of EC methods are Genetic Algorithms, Genetic Programming, Evolution Strategies, Differential Evolution, Particle Swarm Optimization and Ant Colony Optimization. A common characteristic of all these algorithms is that they do not require objective functions with good mathematical properties, such as continuity or differentiability. Therefore, they are applicable to hard real-world optimization problems that involve discontinuous objective functions and/or disjoint search spaces [4]. Furthermore, EC methods have tackled effectively and efficiently a number of hard and complex problems in numerous scientific fields [5, 7, 10, 11].

In this contribution, a new methodology for the design of strong regular bijective S-boxes is proposed, and two Evolutionary Computation methods, namely the Particle Swarm Optimization method (PSO) and the Differential Evolution method (DE), are employed to address the problem at hand. The results indicate that the proposed methodology is effective and directions for future work are derived.

## 2 Theory and Problem Formulation

In this section a review of the theoretical background required for the design of S-boxes is provided and the problem formulation as an optimization task is described.

**Theoretical Background:** Let $f : \mathbb{B}^n \mapsto \mathbb{B}^m$ be an S-box mapping $n$ Boolean input values to $m$ Boolean output values. In the case where $m = 1$, $f$ is a single output Boolean function, and if the number of inputs mapping to 0 is equal to the number of inputs mapping to 1, the Boolean function is called *balanced*. Balance is a property of major importance for Boolean functions used in cryptographic applications as it ensures that the function cannot be approximated by any constant function. Generalizing the notion of balance for the multiple output functions, if each possible output value of $m$ binary components appears equally as output of the function, i.e., $2^{n-m}$ times, then the function is called *regular*. In case of S-boxes with $n = m$, the S-boxes are called *bijective* and all possible outputs appear exactly one

1-4244-0605-6/06/$20.00 ©2006 IEEE.

time each. For simplicity reasons, in the rest of the paper we will refer to single output Boolean functions just as Boolean functions and to the multiple output case as S-boxes.

For the design of cryptographically strong Boolean functions and S-boxes two traditional quality measures exist, the *nonlinearity* and the *autocorrelation*. In order to define these two measures, the definitions of *linear* and *affine* Boolean functions, the *Walsh Hadamard Transform*, the *polarity truth table* and the *Parseval's theorem* are needed. A *linear Boolean function* selected by $\omega \in \mathbb{B}^n$ is denoted by $L_\omega(x) = \omega_1 x_1 \oplus \omega_2 x_2 \oplus \cdots \oplus w_n x_n$, where $w_i x_i$, for $i = 1, \ldots, n$, denotes bitwise AND of the $i$th bits of $\omega$ and $x$ and $\oplus$ denotes bitwise XOR. The set of *affine Boolean functions* consists of the set of linear Boolean functions and their complements.

For a Boolean function $f$ a useful representation is the *polarity truth table* which is given by $\hat{f}(x) = (-1)^{f(x)}$. The *Walsh Hadamard Transform* (WHT) of a Boolean function $f$ is defined as $\hat{F}_f(\omega) = \sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{L}_\omega(x)$. The WHT is a measure for the correlation among the Boolean function $f$ and the set of linear Boolean functions. In general two Boolean functions $f, h$ are considered to be *uncorrelated* when $\sum_x \hat{f}(x)\hat{h}(x) = 0$. The maximum absolute value taken by the WHT is denoted as $WH_{\max}(f) = \max_{\omega \in \mathbb{B}^n} \left|\hat{F}_f(\omega)\right|$ and is related to the nonlinearity of $f$. Specifically, the *nonlinearity* $N_f$ of a Boolean function $f$ is defined as $N_f = (2^n - WH_{\max}(f))/2$. Regarding the WHT, as proved by the *Parseval's theorem* it holds that $\sum_{\omega \in \mathbb{B}^n} (\hat{F}_f(\omega))^2 = 2^{2^n}$, which results in $WH_{\max}(f) \geqslant 2^{n/2}$. The set of functions with nonlinearity equal to this lower bound are called *bent* functions but they are never balanced. The set of balanced functions with maximum nonlinearity and the determination of bounds for balanced functions are important open problems [8].

The autocorrelation of a Boolean function is a measure of its self-similarity and is defined by $\hat{r}_f(s) = \sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{f}(x \oplus s)$, where $s \in \mathbb{B}^n$. The maximum absolute value taken by the autocorrelation is denoted as $AC_{\max}(f) = \max_{s \in \mathbb{B}^n \setminus \{0^n\}} \left|\sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{f}(x \oplus s)\right|$.

The nonlinearity and autocorrelation measures for Boolean functions can be extended to S-boxes by defining a set of functions $f_\beta$, that are linear combinations of the outputs of the corresponding S-box. Specifically, for an S-box $f : \mathbb{B}^n \mapsto \mathbb{B}^m$, a function $f_\beta(x)$, for each $\beta \in \mathbb{B}^m$, that is linear combination of the $m$ outputs of $f$, is defined, as $f_\beta(x) = \beta_1 f_1(x) \oplus \cdots \oplus \beta_m f_m(x)$, where $f_j(x)$, for $j = 1, \ldots, m$ denotes the $j$th bit of the S-box's output. There are $2^m - 1$ non trivial functions $f_\beta$, and for each such function the WHT value, denoted as $\hat{F}_\beta(\omega)$, and the autocorrelation value, denoted as $\hat{r}_\beta(s)$, are obtained directly by the former definitions. Thus, the nonlinearity of an S-box is the lowest nonlinearity over all $2^m - 1$ corresponding $f_\beta(x)$ functions, and the autocorrelation of the S-box is the highest autocorrelation over the same $f_\beta(x)$ functions.

**Objective Functions:** Research on the design of Boolean functions and S-boxes addressing the problem as an optimization task, aims at obtaining functions with high nonlinearity or/and low autocorrelation. The problem of finding the Boolean function $f^*$ with the highest nonlinearity, can be formulated as a maximization one, i.e., maximize subject to $f$ the function $N_f = (2^n - WH_{\max}(f))/2$, or, equivalently, as a minimization task through the maximum absolute value of WHT, i.e., minimize subject to $f$ the function $WH_{\max}(f) = \max_{\omega \in \mathbb{B}^n} \left|\hat{F}_f(\omega)\right|$. For the Boolean function $f^{**}$ with the lowest autocorrelation, the problem is formulated as minimizing subject to $f$ the function $AC_{\max}(f) = \max_{s \in \mathbb{B}^n \setminus \{0^n\}} \left|\sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{f}(x \oplus s)\right|$. For the case of S-boxes, the aforementioned objective functions are generalized as follows. Regarding the nonlinearity of an S-box, the problem is to minimize subject to $f$ the function

$$\max_{\beta \in \mathbb{B}^m, \omega \in \mathbb{B}^n} \left|\hat{F}_\beta(\omega)\right|, \tag{1}$$

and, for the autocorrelation of an S-box, the problem is to minimize subject to $f$ the function

$$\max_{\beta \in \mathbb{B}^m \setminus \{0^m\}, s \in \mathbb{B}^n \setminus \{0^n\}} \left|\hat{r}_\beta(s)\right|. \tag{2}$$

The previously described objective functions can be considered as traditional functions for designing Boolean functions and S-boxes employing optimization methods, as they are commonly used in the relevant literature. Recently, in [1, 2] new spectrum based cost functions were also proposed for the problem of S-boxes design. However, since the proposed methodology of our contribution is new, its performance is studied using the traditional objective functions in order to ensure its effectiveness. We will extend the study of its performance using the spectrum based cost functions in a future correspondence.

## 3 The Proposed Methodology

To tackle the problem of S-boxes design as an optimization task, we employ two Evolutionary Computation methods, namely the Particle Swarm Optimization method and the Differential Evolution method. In the following paragraphs the details of the proposed methodology are described.

For the implementation of the optimization methods to address the problem at hand, the representation of each possible solution needs to be considered. We are interested in finding the S-box $f : \mathbb{B}^n \mapsto \mathbb{B}^m$, with $n$ Boolean inputs and $m$ Boolean outputs, which satisfies the property of highest

possible nonlinearity or the property of lowest possible autocorrelation or both. The representation that we use for each possible solution is the truth table of the corresponding S-box output in decimal form. That is a $2^n$-dimensional vector of integer components, where each component represents the corresponding $m$-bit output of the S-box in decimal form. In this way the optimization problem is transformed into a discrete optimization task. For tackling this problem, we have considered two well-known and widely used Evolutionary Computation methods, namely the Particle Swarm Optimization method [3, 4] and the Differential Evolution algorithm [13]. Both these methods were initially designed for application to real optimization problems, but their performance in handling discrete optimization tasks through the technique of rounding off the real values of the solution to the nearest integer [12] has also proved to be efficient [5, 6, 7].

As we have already mentioned, balance is a very important property for Boolean functions used in cryptographic applications. This property is inherited to S-boxes through regularity, since an S-box is regular if and only if all non zero linear combinations of its output are balanced Boolean functions. Thus, we are interested in finding regular S-boxes with high nonlinearity or/and low autocorrelation. The solution representation used implies that the search space of the problem contains all S-boxes with $f : \mathbb{B}^n \mapsto \mathbb{B}^m$, regular and not. In the relevant literature [2, 8, 9], in order to obtain regular S-boxes as solutions, an initialization with random regular candidate solutions takes place and then the optimization method utilized, is responsible for maintaining the regularity of the produced offsprings. Next, for the construction of regular S-boxes, we propose an alternative technique, named *Regularity Construction Technique*. Specifically, we allow the exploration by the employed method of all the search space, i.e., search among feasible (regular) and unfeasible (non regular) solutions, but for the evaluation of a possible solution, the proposed candidate is transformed to the closest (by means of Hamming distance) regular one, for every wrong component. Thus, the method is allowed to perform better exploration of the search space and moreover its dynamic is retained.

For the initialization of each method's population, the same procedure used in the relevant literature is followed [9], i.e., an array that contains $2^{n-m}$ copies of each of the $2^m$ possible outputs is constructed and its components are randomly swapped to generate the initial random population.

## 4 Experimental Setup and Results

Both the PSO and DE methods were applied considering each component of the possible solution as a real number in the range $[0, 2^m - 1]$ and all populations were constrained to lie within this region. For the evaluation of the suggested solutions, the technique of rounding off the real values of the solution to the nearest integer [12] was applied, followed by the Regularity Construction Technique described in Section 3. Regarding the PSO, we have considered the global variant of the constriction factor version (PSOGC) and local variants with neighborhood size one (PSOLC1) and two (PSOLC2) [4, 6]. For the DE algorithm we have used five variants of the mutation operation (DE1-DE5), described in [6]. For all variants of methods considered here we have used the same parameter setting as in [6].

The proposed approach was tested for bijective S-boxes of size $5 \times 5$, $6 \times 6$, $7 \times 7$ and $8 \times 8$. For each setting, the size of the population was set equal to 20 and the performance of the methods was investigated over 100 independent runs. A threshold of 1000 function evaluations was set for each experiment allowing, thus, 50 iterations for each variant.

In Table 1 the best values for targeting the nonlinearity and autocorrelation of the S-boxes, using the objective functions of Eqs. (1),(2), respectively, are reported.

The proposed methodology obtained the same best nonlinearity values found by the Hill Climbing (HC) algorithm [8] using the same objective function with sample size 10000, with exception the case of $n = 8$, where HC achieved nonlinearity value 100. This, may be due to the small sample size or the small population size used here as the problem dimension increases, and remains under investigation. However, these first results indicate that the new approach can be considered effective in obtaining good regular bijective S-boxes. Furthermore, the proposed approach is simple, it requires only the function values of the proposed solutions and it can be easily combined with alternative regulation construction techniques.

In Table 2 the best joint values of nonlinearity and autocorrelation achieved by all considered methods for both cases of targeting are given. It is important to note that, as indicated by the results, there is a trade-off between the quantities of nonlinearity and autocorrelation. Since, in some cases, we are interested in both high nonlinear and low autocorrelated S-boxes, we intend to study the performance of the proposed approach for tackling the multi-objective problem in a future correspondence.

In Table 3 the frequency of nonlinearity values achieved over 100 independent runs by all the considered methods are reported. All the nonlinearity values obtained are exhibited along with the corresponding times of their appearance shown in parentheses.

Concerning the performance of the different variants of the two methods, they all obtained the same best values, except from DE3, DE4 and DE5, for $n = 7$, where the nonlinearity value 44 was achieved, and PSOLC1, DE1, DE2, DE4 and DE5, for $n = 8$, where the autocorrelation value 88 was obtained.

**Table 1. Best values for S-boxes of size $n$ targeting nonlinearity and autocorrelation.**

| | Nonlinearity | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $n$ | PSOGC | PSOLC1 | PSOLC2 | DE1 | DE2 | DE3 | DE4 | DE5 |
| 5 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 6 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 7 | 46 | 46 | 46 | 46 | 46 | 44 | 44 | 44 |
| 8 | 98 | 98 | 98 | 98 | 98 | 98 | 98 | 98 |
| | Autocorrelation | | | | | | | |
| $n$ | PSOGC | PSOLC1 | PSOLC2 | DE1 | DE2 | DE3 | DE4 | DE5 |
| 5 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 6 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| 7 | 56 | 56 | 56 | 56 | 56 | 56 | 56 | 56 |
| 8 | 80 | 88 | 80 | 88 | 88 | 80 | 88 | 88 |

**Table 2. Best joint values for S-boxes of size $n$ over all methods.**

| | $(N_f, AC_{max})$ | |
|---|---|---|
| $n$ | Targeting $N_f$ | Targeting $AC_{max}$ |
| 5 | $(10, 24)$ | $(8, 16)$ |
| 6 | $(20, 32)$ | $(20, 32)$ |
| 7 | $(46, 64)$ | $(44, 56)$ |
| 8 | $(98, 88)$ | $(96, 80)$ |

**Table 3. Frequency of nonlinearity values achieved by all methods over $100$ runs.**

| $n$ | 5 | 6 | 7 | 8 |
|---|---|---|---|---|
| PSOGC | 8(99) 10(1) | 20(100) | 44(99) 46(1) | 96(28) 98(72) |
| PSOLC1 | 8(99) 10(1) | 20(100) | 44(98) 46(2) | 96(19) 98(81) |
| PSOLC2 | 8(99) 10(1) | 20(100) | 44(99) 46(1) | 96(19) 98(81) |
| DE1 | 8(99) 10(1) | 20(100) | 44(99) 46(1) | 96(28) 98(72) |
| DE2 | 8(99) 10(1) | 20(100) | 44(99) 46(1) | 96(31) 98(69) |
| DE3 | 8(99) 10(1) | 20(100) | 44(100) | 96(17) 98(83) |
| DE4 | 8(99) 10(1) | 20(100) | 44(100) | 96(42) 98(58) |
| DE5 | 8(99) 10(1) | 20(100) | 44(100) | 96(36) 98(64) |

## 5 Conclusions

In this paper, a new methodology for the design of strong regular S-boxes is presented, utilizing two Evolutionary Computation methods, the Particle Swarm Optimization method and the Differential Evolution algorithm. This new approach uses as search space the space of all S-boxes (regular and non regular), allowing, thus, better exploration among the S-boxes, and employs a simple regularity construction technique to provide regular solutions.

The proposed approach has been tested using the traditional measures of nonlinearity and autocorrelation for regular bijective S-boxes and the obtained results are comparable to those of other more complex heuristic methods using the same objective functions. Furthermore, the new methodology required smaller sample size to obtain the same values in almost all cases.

In this contribution the first attempt using the new approach is presented and many ideas for further research are derived. In a future correspondence we intend to study the performance of this methodology using the new spectrum based objective functions for strong regular S-boxes derived in [2], extended also for non bijective S-boxes. Moreover, alternative regulation construction techniques will be considered. Finally, the formulation of the problem as a multiobjective one could provide interesting results.

## References


[1] J. Clark, J. Jacob, and S. Stepney. The design of S-boxes by Simulated Annealing. In *CEC 2004: International Conference on Evolutionary Computation, Portland OR, USA, June 2004*, pages 1533–1537. IEEE, 2004.

[2] J. Clark, J. Jacob, and S. Stepney. Searching for cost functions. In *CEC 2004: International Conference on Evolutionary Computation, Portland OR, USA, June 2004*, pages 1517–1524. IEEE, 2004.

[3] M. Clerc and J. Kennedy. The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[4] J. Kennedy and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.

[5] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, and M. N. Vrahatis. Evolutionary computation based cryptanalysis: A first study. *Nonlinear Analysis: Theory, Methods and Applications*, 63:e823–e830, 2005.

[6] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, and M. N. Vrahatis. Applying evolutionary computation methods for the cryptanalysis of Feistel ciphers. *Applied Mathematics and Computation*, in press, 2006.

[7] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis. Particle swarm optimization for integer programming. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation*, pages 1576–1581, Hawaii, HI, 2002. IEEE Press.

[8] W. Millan. How to improve the nonlinearity of bijective S-boxes. *LUNGS*, 1438:181–192, 1998.

[9] W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson. Evolutionary heuristics for finding cryptographically strong S-boxes. *LNCS*, 1726:263–274, 1999.

[10] K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2–3):235–306, 2002.

[11] V. P. Plagianakos and M. N. Vrahatis. Parallel evolutionary training algorithms for "hardware–friendly" neural networks. *Natural Computing*, 1(2–3):307–322, 2002.

[12] S. Rao. *Engineering Optimization–Theory and Practice*. Wiley Eastern, New Delhi, 1996.

[13] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.