

# An External Parameter Independent Novel Cost Function for Evolving Bijective Substitution-Boxes

Alejandro Freyre-Echevarría <sup>1</sup>, Ahmad Alanezi <sup>2</sup>, Ismel Martínez-Díaz <sup>1</sup>, Musheer Ahmad <sup>3,\*</sup>, Ahmed A. Abd El-Latif <sup>4</sup>, Hoshang Kolivand <sup>2,\*</sup> and Abdul Razaq <sup>5</sup>

<sup>1</sup> Institute of Cryptography, University of Havana, Havana 10400, Cuba; alefreyre.43@gmail.com (A.F.-E.); diomedes.martnezdaz2@gmail.com (I.M.-D.)

<sup>2</sup> Department of Computer Science, Faculty of Engineering and Technology, Liverpool John Moores University (LJMU), Liverpool L3 3AF, UK; A.D.Alanezi@2019.ljmu.ac.uk

<sup>3</sup> Department of Computer Engineering, Jamia Millia Islamia, New Delhi 110025, India

<sup>4</sup> Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egypt; a.rahiem@gmail.com

<sup>5</sup> Department of Mathematics, Division of Science and Technology, University of Education, Lahore 54770, Pakistan; makenqau@gmail.com

\* Correspondence: musheer.cse@gmail.com (M.A.); H.Kolivand@ljmu.ac.uk (H.K.)

Received: 5 October 2020; Accepted: 13 November 2020; Published: 18 November 2020

**Abstract:** The property of nonlinearity has high importance for the design of strong substitution boxes. Therefore, the development of new techniques to produce substitution boxes with high values of nonlinearity is essential. Many research papers have shown that optimization algorithms are an efficient technique to obtain good solutions. However, there is no reference in the public literature showing that a heuristic method obtains optimal nonlinearity unless seeded with optimal initial solutions. Moreover, the majority of papers with the best nonlinearity reported for pseudo-random seeding of the algorithm(s) often achieve their results with the help of some cost function(s) over the Walsh–Hadamard spectrum of the substitution. In the sense, we proposed to present, in this paper, a novel external parameter independent cost function for evolving bijective s-boxes of high nonlinearity, which is highly correlated to this property. Several heuristic approaches including GaT (genetic and tree), LSA (local search algorithm), and the Hill Climbing algorithm have been investigated to assess the performance of evolved s-boxes. A performance comparison has been done to show the advantages of our new cost function, with respect to cost functions for s-boxes like Clark’s and Picek’s cost functions.

**Keywords:** substitution boxes; cost function; nonlinearity; Walsh–Hadamard spectrum; optimization

## 1. Introduction

Today, the information shared online by users is a highly valuable resource. The integrity of such data rests in the use of cryptographic algorithms, which provide a set of algorithmic tools to maintain the consistency and security of the data. Therefore, the design of cryptographic algorithms with high level of security is a wide area of investigation from researchers tasked to information security and reliability. There exist two large groups of cryptographic algorithms, public-key algorithms and private-key algorithms, but the first group is not in the scope of this paper. We center our attention in private-key algorithms, particularly in block ciphers. Elementary, a block cipher partition the flow of information in  $k$  bit pieces, where each piece of data is matched with a different one. The Data Encryption Standard (DES) [1] and the Advanced Encryption Standard (AES) [2] are,

perhaps, the most representative examples of symmetric block ciphers, although there are other good examples in the literature [3–5], even some that do not rely on nonlinear components—this can reduce the security against linear and differential attacks—because they want to ensure the security of statistical properties in the encryption scheme [6].

Substitution-boxes (s-boxes) are the prominent nonlinear components of modern-day block ciphers, adding confusion to encryption process. Hence, any confusion layer highly depends on the s-boxes to maintain the integrity of shared secrets. Although the primary use of s-boxes is for symmetric block ciphers [2–5], one cannot restrict the use of s-boxes only to these encryption systems. s-boxes are also applied to image encryption schematics with excellent results [7–9]. Therefore, the effectiveness of any symmetric encryption system using s-boxes will rest in the selection of such components.

To warrant the integrity of one s-box, it must satisfy a set of properties which assess the resistance of the s-box towards several cryptanalytic techniques [10,11]. The search for cryptographic sound s-boxes is a well-studied subject in the literature. The construction of s-boxes mainly follows approaches namely: algebraic constructions [2,12,13], pseudo-random generation, chaos-based generation [7,14,15], and heuristic methods [16–18]. Algebraic constructions achieve unsurpassed outcomes with regards of many security properties of s-boxes [19]. Random generation makes it possible to yield a considerable number of s-boxes in short periods. However, any random s-box lack of good values of its properties, therefore it is not suitable for practical applications. Like in random generation, chaos-based s-boxes are quickly integrated into the encryption process, particularly in images encryption, but they mainly ensure a more randomness and statistical sound. The last direction focuses on the use of heuristic methodologies to find s-boxes with strong cryptographic features. Next, we present a representative set of such research findings.

### 1.1. Related Work

The public literature contains an extensive survey of evolutionary computation papers related to design of s-boxes having good cryptographic properties. We present a brief resume of some important results in this area of research. In 2005, Clark et al. proposed a cost function for evolving of s-boxes coupled with simulated annealing to obtain s-boxes with nonlinearity values up to 102 [16]. Later in 2010, Tesař perform an extensive parameter tuning of Clark's cost function which, in combination with a special genetic algorithm named genetic and tree (GaT), makes it possible to obtain 8-bit s-boxes with good nonlinearity [17]. In 2013, Kazymyrov et al. presented a modified gradient descent method to obtain s-boxes with nonlinearity 104 and high algebraic resistance [20]. Ivanov et al. experiment with modified immune algorithm using three different cost functions, included the tuned version of Clark's cost function proposed by Tesař and a function over the differential spectrum of s-boxes to achieve nonlinearity 104 and differential uniformity 6 [21]. They applied genetic algorithms working in reverse mode to evolve high nonlinear bijective s-boxes of sizes from  $8 \times 8$  up to  $16 \times 16$  [22]. In 2014, Picek et al. investigated side-channel analysis resilience of s-boxes considering the confusion coefficient property [23]. Later, in 2016, Picek et al. presented a cost function for evolution of high nonlinear s-boxes [24]. More research was presented by Picek et al. in 2017, making use of cellular automata and genetic programming for design s-boxes with good cryptographic features [25,26]. Isa et al. presented a hybridization of heuristic methods for generating 8-bit permutations with good nonlinearity and differential uniformity results [27]. Menyachikhin (2017) presented the spectral-linear and spectral differential techniques for constructing s-boxes consisting of near-optimal security features of s-boxes [28], wherein, they used the information from linear approximation table and differential spectrum of s-boxes to improve the properties of resulting s-boxes. Lerman et al. are assisted by genetic algorithms in the generation of side-channel attacks robust s-boxes of small dimensions [29]. Martínez-Díaz works with local search algorithms to evolve s-boxes with improved side-channel resistivity having good values of nonlinearity [30]. This line of research was continued by Freyre-Echevarría in 2020, presenting a hybrid heuristic algorithm capable of produce s-boxes with high theoretical resistance to SCA attacks, as well as acceptable nonlinearity and low differential cryptanalysis performance [31].

Bolufé and Tamayo use hybrid heuristic methods and machine learning for the evolution of s-boxes, taking in count the properties of nonlinearity and transparency order [32]. Ahmad et al. propose the use of particle swarm optimization and chaotic Renyi's map to obtain  $8 \times 8$  s-boxes with high nonlinearity scores [7].

In most of the aforementioned works, the property of nonlinearity is taken under consideration in the optimization process [7–9,14,16,17,20–22,24,27,28,30,31,33,34–40]. However, heuristic techniques cannot achieve nonlinearity values close to algebraic constructions unless they are seeded with s-boxes having optimal properties [22]. The best value of nonlinearity reported from papers that use random initial s-boxes to seed their algorithms is bounded above by 104 for  $8 \times 8$  s-boxes [16,17,21,24]; and, in most of cases (for not being absolute), these values cannot be achieved without the assistance of cost functions. Cost functions help to describe the behavior of coefficients in the Walsh–Hadamard spectrum of s-boxes. Hence, a well descriptive cost function will undoubtedly help to improve the final nonlinearity of s-boxes. With respect to nonlinearity, the most accurate cost functions are Clark's [16] and Picek's [24] cost functions, each one with its own characteristics described in Section 3.

The rest of the paper is managed as follows. Section 2 contains several definitions with regard to s-boxes and their properties. In Section 3, we briefly describe Clark's and Picek's cost functions and present the contribution of this paper. Section 4 is dedicated to explaining the heuristic methods we have employed and the parametric configuration of the same for each different s-box dimension. Finally, in Section 5, we present and discuss our experimental results and establish some comparison with respect to the results obtained from Clark's and Picek's cost functions.

## 2. Preliminaries

An s-box  $S: F_2^n \rightarrow F_2^m$  is often described as multi-input and multi-output Boolean functions consisting of  $m$  Boolean functions in  $n$  variables known as the coordinates of s-box  $S$ . However, all coordinates functions and their all linear combinations are responsible for deciding the cryptographic strength of the s-box [41].

**Definition 1.** Let  $S: F_2^n \rightarrow F_2^m$  be an s-box. The components of s-box  $S$  are called the  $n$ -variable Boolean functions

$$S_\lambda: x \rightarrow \lambda \cdot S(x)$$

for any  $\lambda \in F_2^m$ . The component corresponding to  $\lambda = 0$  is called zero (or trivial) component (Definition 2.1 from [39]).

**Definition 2.** One s-box  $S: F_2^n \rightarrow F_2^m$  is balanced if every value  $x \in F_2^m$  appears exactly equal to  $2^{n-m}$  times. When  $n = m$ , the s-box  $S$  is known as bijective, i.e., that each input value is uniquely mapped to one output value.

Balanced  $n \times n$  s-boxes are permutations in  $F_2^n$  [13,19]. In particular, the fact that an s-box is invertible (i.e., a permutation) can be characterized by its coordinates [33]. For the sake of simplicity, we restrict ourselves to the study of bijective substitution boxes only.

**Definition 3.** Let  $S: F_2^n \rightarrow F_2^m$  be an s-box. The Walsh–Hadamard transform of  $S$  is computed as [19]:

$$W_S(x, y) = \sum_{z \in F_2^n} (-1)^{y \cdot S(z) \oplus x \cdot z}$$

The linearity (resp. nonlinearity) is the highest (resp. lowest) of any nontrivial component function of one s-box [13,19]. The two can be described in terms of the  $W_S$  transform as:

$$L_S = \max_{x \in F_2^n, y \in F_2^m} |W_S(x, y)| \quad (1)$$

$$N_S = 2^{n-1} - \frac{1}{2} \max_{x \in F_2^n, y \in F_2^{m*}} |W_S(x, y)| \quad (2)$$

This gives the following relation between linearity and nonlinearity of s-boxes.

$$N_S = \frac{2^n - L_S}{2} \quad (3)$$

**Definition 4.** (Parseval's relation) *Given any Boolean function  $f: F_2^n \rightarrow F_2$ , its Walsh–Hadamard transform satisfies that*

$$\sum_{w \in F_2^n} W_f(w)^2 = 2^{2n}$$

A direct result from Parseval's relation is that linearity of Boolean functions (resp. s-boxes) is lower bounded by  $2^{\frac{n}{2}}$ . Notice that equality can only be achieved when  $n$  is even, and such functions are known as the bent functions [42]. Bent functions have the maximum achievable nonlinearity, but they are not balanced. For the case of bijective substitution boxes the maximum achievable nonlinearity cannot be greater than the Sidelnikov–Chabaud–Vaudenay (SCV) bound [43]:

$$N_S \leq 2^{n-1} - 2^{\frac{n-1}{2}} \quad (4)$$

which immediately resolves in

$$L_S \geq 2^{\frac{n+1}{2}} \quad (5)$$

The case of equality denotes the functions which are called the Almost Bent (AB) function. Notice that AB-functions only exists when  $n$  is an odd number [19]. When  $n$  is even, the maximum value of nonlinearity is achieved through power permutations over the finite field  $F_{2^n}$  and equals to [8]:

$$N_S = 2^{n-1} - 2^{\frac{n}{2}} \quad (6)$$

**Definition 5.** *The autocorrelation function of one s-box  $S: F_2^n \rightarrow F_2^m$  is defined as [19]:*

$$AC_S(x, y) = \sum_{z \in F_2^n} (-1)^{y \cdot S(z) \oplus y \cdot S(z \oplus x)}$$

There exists two significant cryptographic parameters: (1) global avalanche characteristics (GAC) [44], which is related to the autocorrelation function, used to measure the level of diffusion ensured by a function; and (2) The absolute value of the autocorrelation function is called an absolute indicator of anticipated s-box. In practice, the absolute indicator is determined as:

$$AC_{max}(S) = \max_{x \in F_2^n, y \in F_2^m} |AC_S(x, y)| \quad (7)$$

**Definition 6.** *Let  $S: F_2^n \rightarrow F_2^m$  be an s-box. For any  $x \in F_2^n, y \in F_2^m$  one can define:*

$$\delta(x, y) = |\{v \in F_2^n : S(v + x) + f(v) = y\}|$$

*The multi-set  $\Delta_S = \{\delta(x, y), x \in F_2^{n*}, y \in F_2^m\}$  represents the I/O differential distribution spectrum of  $S$  and its maximum value is called differential uniformity of  $S$ , denoted  $\delta_S$ .*

For any balanced s-box  $S$  the differential uniformity of s-box satisfies as  $\delta_S \geq 2$  [13]. The functions having  $\delta_S = 2$  are called almost perfect nonlinear (APN) functions. As for nonlinearity property, the APN condition only exists for odd number of variables, and when  $n = 6$  [45]. In the case of  $n$  even, the best-known differential uniformity value is 4 [12,19,33].

### 3. Motivation and Contribution

The outcomes of any optimization algorithm are heavily dependent on the fitness functions used to guide the optimization process. Here, we discuss about some cost functions with proven effectiveness towards nonlinearity of s-boxes obtained by heuristic methods.

In 2005, Clark et al. introduce a cost function that consider all values of the Walsh–Hadamard spectrum for Boolean functions, which can be scaled for the multi-output case (s-boxes) [16]:

$$WHS_S = \sum_{y \in F_2^n} \sum_{x \in F_2^n} ||W_f(x, y)| - X|^R$$

where  $X$  and  $R$  are real-valued parameters like  $R = 3$  and  $X \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$  [16]. The maximum nonlinearity obtained by Clark et al. using simulated annealing and WHS as the cost function was 102 for  $8 \times 8$  s-boxes. Later in 2010, Tesař perform an extensive parameter tuning on WHS, obtaining that the best set of parameters has  $R = 7, X = 21$ , after establishing some heuristic ranking over more than one thousand pairs of values for  $X$  and  $R$  [17]. The highest value of nonlinearity reported for the tuned version of WHS, and the genetic and tree algorithm was 4, 10, 22, 48, and 104 for s-boxes of sizes from  $4 \times 4$  up to  $8 \times 8$  respectively.

In 2016, Picek et al. propose the representation of the Walsh–Hadamard spectrum as the histogram of frequencies for all absolute values in the spectrum [24]. The histogram is represented by a vector having in the  $i$ -th component the count of coefficients associated to absolute value  $4i$  from the Walsh–Hadamard spectrum of the s-box. Let  $l$  indicates the last component of the vector with nonzero value. The absolute value of the Walsh–Hadamard spectrum associated with  $l$  is the linearity of the s-box  $S$ , therefore the nonlinearity of  $S$  is related to this value. Then, the function PCF is given by the formula [24]:

$$PCF_S = \sum_{i=0}^{N-1} \frac{H(S)_{l-i}}{2^i}$$

where  $H(S)_k$  is the  $k$ -th component of the zero-indexed vector  $H(S)$ , which represent the aforementioned histogram of frequencies, and  $H(S)_k = 0, \forall k < 0$ . Multiplying by the term  $2^{-i}$ , i.e., dividing  $2^i$  as shown in (14), the authors pretend to give some ranking to the influence of one coefficient over the final value of the function. For example, the maximum absolute value of the Walsh–Hadamard spectrum is divided by  $2^0 = 1$ , implying that it is the most influential on the final result of PCF. Altogether, Picek et al. set the value of parameter  $N$  to 10, given by the fact that if one s-box does not contain ten levels of coefficients ( $N < 10$ ), then all coefficients of the spectrum are considered when PCF is calculated.

One can characterize both functions in terms of the count of coefficients in the Walsh–Hadamard spectrum taken into account to compute their value and the dependence of any external parameter with no relation with such coefficients. In the case of Clark’s cost function, the whole spectrum is analyzed to get the final value of the function. However, the values of parameters  $X$  and  $R$  represent a major disadvantage, since they have to be tuned in order to achieve the best results [18]. In contrast, Picek’s cost function selects a sample of the coefficients in the spectrum depending on the value of parameter  $N$ ; if sufficiently large, all the values in the spectrum are taken into account. Moreover, after the study of results from [18], we notice that selection of parameter  $N$  has influence in the outcome nonlinearity of evolved s-boxes, as well the number of solution evaluations to obtain desired values of this property. We refer the readers to [16,17,24] for more information about Clark’s and Picek’s cost functions.

#### Our Contribution

The main contribution of our paper is the construction of a novel cost function for evolving the performance of high nonlinearity s-boxes without depending on in any external parameter unlike the Clark’s and Picek’s cost functions which are external parameter dependent [16,24]. Moreover, further contributions includes in the form of statistical analysis of correlation between Clark’s,

Picek's, and our function with respect to nonlinearity of s-boxes and the convergence rate of various algorithms for different s-box dimensions using our new cost function.

#### 4. Experimental Setup

Optimization algorithms can be divided in two groups: exact methods and heuristic methods. While exact methods guaranty the optimal solution in a finite time, for more complex problems, like that in our paper, the notion of time exponentially growth with regard to the dimensions of the problem. Hence, the heuristic methods are suitable to challenge these problems. It is well known that heuristic methods do not guaranty to find the optimal solution, but they often achieve a good solution in a reasonable time. Most of heuristic methods are problem dependent. In counterpart, meta-heuristic algorithms establish a high-level algorithmic framework to bring the more accurate solution to the problem.

In Section 1.1, we commented on successful applications of meta-heuristic techniques to improve different parameters related to the security of s-boxes. The effectiveness of these algorithms to obtain desired nonlinearity values differ on the fitness function, the selection, variation and mutation operators and the characteristics of the selected algorithm. Nonetheless, these optimization algorithms ensure substitution boxes applicable to real life encryption systems with the advantage of randomized structure, unlike algebraic constructions, different pool of high nonlinear solutions each time the algorithm is executed and low consumption of time. Moreover, these algorithms can be seeded with optimal s-boxes instead of pseudo random substitutions, and they will manage to produce optimal or almost optimal s-boxes in terms of nonlinearity.

In our experiments, we applied three optimization methods, namely, genetic and tree algorithm (GaT) [17], local search algorithm (LSA) [24], and a hill simple climbing algorithm (HC) explained later in this section. We choose these optimization methods to obtain a high non-linear value, because their exploitation capabilities over the solution search space. The experiments conducted on this paper present the results achieved with proposed cost function for evolving bijective s-boxes of sizes ranging from  $5 \times 5$  to  $8 \times 8$ . We also include pseudo-code of each algorithm in the appendix of this paper (see Appendix B, C, D).

##### 4.1. Common Parameters

We execute different amounts of experiments according to dimensions of the analyzed s-box space. The stopping condition for each algorithm is reached when a fixed number of solution evaluations are done, which differ in relation to the size of the space as shown in Table 1. In all algorithms used, we define the fitness function as the tuple  $(N_s, C_s)$ . Here,  $N_s$  represents the nonlinearity of the solution and  $C_s$  represents the value of our new cost function. We decide not to include small  $4 \times 4$  s-boxes since the best value of nonlinearity can be achieved without the assistance of optimization algorithms. However, for larger sizes, s-boxes having maximal nonlinearity are quite difficult to found by any evolutionary algorithm when it starts from random  $n$ -bit permutations. Thus, our goal is to increase nonlinearity of evolved s-boxes as much as possible.

##### 4.2. Local Search Algorithm

The local search algorithm from [24] receives as input a random s-box from the space. In each iteration, the algorithm generates new solutions with given mutation operators. The mutation operator randomly decides  $k$  different positions in solution and then permutes the element at selected positions. To get the best results, Picek et al. set mutation operators with  $k \in \{2, 3, 4, 5, 6, 7\}$ .

In the LSA algorithm, each mutation operator is defined with two parameters  $k$  and  $l$ , where  $k$  is a number of positions whose elements are to be permuted, and  $l$  defines how many times that mutation operator is to be applied on the current solution (see Table 6 from [24]). The best solution is selected and set as the current solution from generated solutions.

#### 4.3. Genetic and Tree

The genetic and tree algorithm was presented by Tesař in [17]. The algorithm is a combination of a special case of the genetic algorithm and total tree search. We note two important aspects of the method: the criterion to begin the total tree search portion of it and the stopping condition. The algorithm swap between the genetic and the tree part when an s-box with nonlinearity close to desired value is found (see Table 2), then, it executes the tree portion of the algorithm until a s-box with desired non-linearity is obtained or the algorithm depletes all solution evaluations presented in Table 1 for the corresponding s-box space. For more information about the configuration of GaT algorithm, we refer the readers to [17,24].

**Table 1.** Common parameters.

S-box size	Experiments	Number of Evaluations	Maximum Nonlinearity
$5 \times 5$	100	125,000	12
$6 \times 6$	100	250,000	24
$7 \times 7$	30	500,000	56
$8 \times 8$	30	1,000,000	112

**Table 2.** Parameters for GaT algorithm. NT—value of nonlinearity to swap between genetic and tree part of the algorithm. NEL—desired nonlinearity.

S-Box Size	$5 \times 5$	$6 \times 6$	$7 \times 7$	$8 \times 8$
NT	8	20	46	102
NEL	10	22	48	104

#### 4.4. Hill Climbing Algorithm

We propose the use of a simple hill climbing mechanism to produce s-boxes having good nonlinearity in a small amount of solution evaluations. The algorithm receives a random permutation as input. Then, while the number of solution evaluations is not depleted, the algorithm creates a new s-box by swapping a pair of outputs on the target s-box. If the new s-box is better than the current solution of the algorithm, according to the fitness condition of the problem, it replaces the solution, becoming the best solution found by the algorithm.

### 5. Results and Discussion

The current section is entirely dedicated to the analysis of results obtained in our experiments. First, we proposed a novel cost function for evolution of high nonlinearity s-boxes. Then, we present the results achieved by the optimization algorithms described in Section 4 using this new function. Note that we also show the values of differential uniformity and absolute indicator of evolved s-boxes.

#### 5.1. Definition of a New Cost Function

The nonlinearity of one s-box is dependent of the highest absolute value of the Walsh–Hadamard spectrum. Most of evolutionary research papers that only use the value of nonlinearity as fitness function to guide the evolutionary process are not able to reach nonlinearity values greater than 100 in the case of 8-bit permutations. This may happen, since nonlinearity only contains information about the highest score of the Walsh–Hadamard spectrum, without extracting any data of the remaining values in the spectrum. Reviewing of the definition of nonlinearity itself, it is straightforward to notice that reducing the highest absolute values of the spectrum leads to an increase of final nonlinearity of s-boxes. However, if the extreme values of the spectrum are reduced, some other values in the same must be increased, in accordance with Parseval’s relation.

Hence, any function exploiting the Walsh–Hadamard spectrum is tasked to make the spectrum as flat as possible.

Let  $C$  be the set of all absolute coefficients lower or equals to the SCV bound [41]. Then, we have the following:

- $C = \{0, 4, \dots, 2^{\frac{n+1}{2}}\}$  if  $n$  is odd
- $C = \{0, 4, \dots, 2^{\frac{n}{2}+1}\}$  if  $n$  is even

**Proposition 1.** *The nonlinearity of one  $s$ -box  $S$  is maximal if, and only if, all the absolute values of the coefficients in the Walsh–Hadamard spectrum of  $S$  are contained in  $C$ .*

**Proof:** We reduce the demonstration to the case when  $n$  is odd since one must review the same conditions when  $n$  is even. First, let us demonstrate the direct implication, i.e., if the nonlinearity of  $S$  is maximal, then all the absolute values of the coefficients in its Walsh–Hadamard spectrum are contained in  $C$ .  $\square$

For  $n$  odd, if the nonlinearity of  $S$  is maximal, then  $N_S = 2^{n-1} - 2^{\frac{n-1}{2}}$  satisfying the equality in (4). Moreover, the result in (5) implies that equality in (4) results in  $L_S = 2^{\frac{n+1}{2}}$ . Since  $L_S$  is the greatest absolute value of the coefficients in the Walsh–Hadamard spectrum of  $S$ , then  $L_S \geq |X|$  for any arbitrary coefficient  $X$  in spectrum. By definition, one has that  $\max(C) = 2^{\frac{n+1}{2}}$ , thus,  $\max(C) = L_S \geq |X|$ . Hence, all the absolute values of the coefficients in the Walsh–Hadamard spectrum of  $S$  are contained in  $C$ .

Conversely, if all coefficient in the Walsh–Hadamard spectrum of  $S$  are contained in  $C$ , then we have  $L_S \leq 2^{\frac{n+1}{2}}$ . However, the inequality in (5) sustain that  $L_S \geq 2^{\frac{n+1}{2}}$ , hence, we have that  $2^{\frac{n+1}{2}} \leq L_S \leq 2^{\frac{n+1}{2}}$ , which resolves in  $L_S = 2^{\frac{n+1}{2}}$ , the minimal value of linearity achievable for bijective  $s$ -boxes with odd number of variables, i.e., the nonlinearity of  $S$  is maximal. Therefore, the proof is now complete.

**Proposition 2.** *Let  $x$  be an integer and  $K$  a finite set of positive integers such that  $|x| \in K$ . Then the following equality holds*

$$P = \prod_{i \in K} (|x| - i) = 0$$

**Proof:** Let  $M$  be the size of  $K$ . One can decompose the formula of  $P$  in the multiplication of  $M$  subtractions as follows

$$P = (|x| - i_1) \cdot (|x| - i_2) \cdot \dots \cdot (|x| - i_M)$$

where the term  $i_t$  denote the  $t$ -th element of  $K$ . If  $|x| \in K$ , exist some  $i_t$  such that  $|x| = i_t$ , i.e.,  $|x| - i_t = 0$ . Hence, substituting the previous subtraction in the decomposition we have  $P = (|x| - i_1) \cdot (|x| - i_2) \cdot \dots \cdot 0 \cdot \dots \cdot (|x| - i_M)$ , which results in  $P = 0$ . Therefore, the proof is complete. On the basis of propositions 1 and 2 we define our new cost function.  $\square$

**Definition 7.** *Let  $S: F_2^n \rightarrow F_2^m$  be an  $s$ -box. Our new cost function is defined as*

$$C_S = \sum_{y \in F_2^n} \sum_{x \in F_2^n} \prod_{z \in C} ||W_S(x, y) - z|$$

where  $W_S$  is the Walsh–Hadamard transform of  $S$ .

The result from Proposition 2 warranties that any coefficient in the Walsh–Hadamard spectrum whose absolute value is contained in  $C$  does not interfere with the final result of our cost function. Hence, the calculus of  $C_S$  is deduced from the coefficients with absolute values greater than the SCV bound. Moreover, the minimum value of  $C_S$  is achieved when the largest absolute



value in the Walsh–Hadamard spectrum equals to the greatest coefficient in  $C$ , where  $C_S$  is equal to zero, implying that  $S$  has maximum nonlinearity, according to Proposition 1.

### 5.2. Relation to Nonlinearity

We perform a statistical analysis on the relation between nonlinearity property and our new cost function as well as Clark's and Picek's cost functions. For such an analysis, we compute the trajectory of the values of one cost function with respect to nonlinearity as follows:

1. Set up a fixed number  $M$  of cost function upgrades.
2. Execute hill climbing algorithm upgrading the value of the cost function regardless nonlinearity.
3. Save the value of the cost function and the corresponding nonlinearity each time the cost function is upgraded.
4. Repeat Step 3 until there is no available upgrade on the cost function (i.e., the function was updated  $M$  times).

The two vectors obtained through this method contain information about both, the cost function and the nonlinearity each time the cost function is improved. Hence, we can calculate the correlation between the cost function and nonlinearity by means of these vectors.

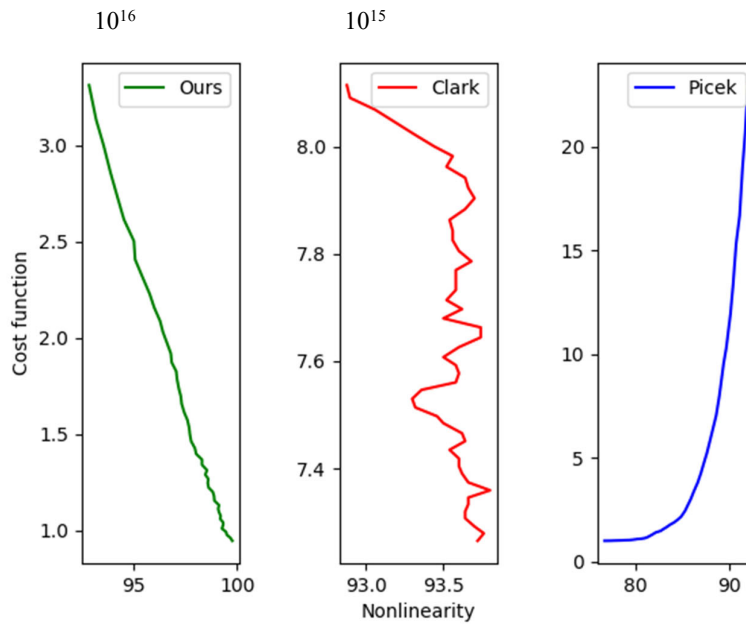
To obtain a more accurate result, we repeat the procedure described above 100 times for each cost function, with  $M = 50$  for s-boxes of dimension eight. Then, we obtain the average trajectories of the cost function and nonlinearity and proceed to calculate the Pearson's correlation coefficient between the two trajectories, i.e., the correlation between the values of the cost function and the value of nonlinearity property. For better understanding of the experiment, Figure 1 present the average trajectory of each cost function w.r.t nonlinearity.

In contradiction with the results presented in [24], the data presented in Table 3 and the curve representing Picek's cost function in Figure 1 (blue) indicates that higher values of the function are better to improve nonlinearity. However, one knows from [24] that for a fixed value of nonlinearity, the minimization of Picek's cost function will lead to maximization of the nonlinearity. Thus, Picek's cost function will not achieve good results if the value of nonlinearity of the s-box is depreciated.

The curve representing the trajectory of Clark's cost function (Figure 1—red) and the corresponding correlation in Table 3 may be helpful to give some explanation of the late convergence of Clark's cost function in [17,24]. Although there is some improvement in the final nonlinearity, one can easily see in the plot that there is no stability in the trajectory. In accordance, the correlation coefficient describes very well this phenomenon. The correlation coefficient indicates an inverse relation between Clark's cost function and nonlinearity, but not enough to ensure a fast convergence of nonlinearity to higher values as the value of Clark's cost function decreases, which agrees with the results obtained by Tesař [17] and Picek et al. [24].

**Table 3.** Correlation of the cost functions w.r.t. nonlinearity. S-box dimension  $8 \times 8$ .

Cost Function	Clark's	Picek's	Ours
Correlation	−0.553	0.824	−0.998
$p$ -value	$3.13 \times 10^{-5}$	$2.04 \times 10^{-13}$	$3.84 \times 10^{-61}$



**Figure 1.** Average trajectory of cost function w.r.t nonlinearity.

Finally, one can observe in Figure 1 (green plot) and Table 3 the results of the analysis for our new cost function. As shown, our proposed cost function is extremely well correlated to nonlinearity. Moreover, Table 4 shows the correlation coefficient between the values of nonlinearity and our new cost function for each s-box dimension we analyze in this paper. Data in Table 4 suggest a strong inverse relation (almost linear) between the values of  $N_S$  and  $C_S$ . Hence, minimizing the results of  $C_S$  will certainly improve the nonlinearity of the s-box. Next, we present the results collected with the proposed cost function.

**Table 4.** Correlation between nonlinearity and our new cost function.

S-Box Size	$5 \times 5$	$6 \times 6$	$7 \times 7$	$8 \times 8$
Correlation	−0.968	−0.998	−0.992	−0.998
p-value	$3.49 \times 10^{-5}$	$2.03 \times 10^{-17}$	$5.01 \times 10^{-22}$	$3.84 \times 10^{-61}$

### 5.3. Results with Our New Cost Function

This section is dedicated to providing and discussing the performance results obtained with our new cost function seeding the optimization algorithms with pseudo-random s-boxes. Together with the values of nonlinearity, differential uniformity, and an absolute indicator of evolved s-boxes, we provide the average number of solution evaluations for each algorithm, to obtain the best value of nonlinearity reported in each s-box space.

In all the experiments conducted on this section, the tuple  $(N_S, C_S)$  determine the fitness conditions of one s-box  $A$  over other s-box  $B$ , as follows:

1.  $N_A > N_B$   
 $N_A = N_B$  and  $C_A < C_B$  if rule 1 is not satisfied

Table 5 shows the results achieved by our new cost function for s-boxes of various sizes. The best nonlinearity values among the results equals the presented by Tesař [17] and Picek et al. [24] for the respective s-box dimension. In addition, we present, in Table 6, the average number of solution evaluations each algorithm needs to obtain the best nonlinearity values reported in Table 5.

**Table 5.** Results with our new cost function for each s-box dimension. Data is given in the format best/average value.

Algorithm	Property	$5 \times 5$	$6 \times 6$	$7 \times 7$	$8 \times 8$
LSA	$N_S$	10 / 9.9	22 / 22	48 / 48	104 / 104
	$\delta_S$	4 / 5.8	6 / 7.08	6 / 8.07	8 / 9.6
	$AC_{max}(S)$	16 / 22.88	32 / 38.24	56 / 59.2	88 / 94.4
GaT	$N_S$	10 / 10	22 / 22	48 / 48	104 / 104
	$\delta_S$	4 / 5.48	6 / 7.04	8 / 8.2	8 / 9.8
	$AC_{max}(S)$	16 / 23.76	32 / 38.08	56 / 58.93	80 / 92.53
HC	$N_S$	10 / 9.8	22 / 22	48 / 48	104 / 104
	$\delta_S$	4 / 5.1	6 / 7.2	6 / 8	8 / 9.53
	$AC_{max}(S)$	16 / 22.3	32 / 37.8	56 / 59.3	88 / 94.93

**Table 6.** Convergence rate of the algorithms to best nonlinearity reported in Table 3 in terms of average solution evaluations.

Algorithm	$5 \times 5$	$6 \times 6$	$7 \times 7$	$8 \times 8$
LSA	10480	2358	7751	149,539
GaT	1260	1563	7007	116,266
HC	9831	1437	5162	70,596

The values of differential uniformity and absolute indicator of evolved s-boxes are not optimal for all dimensions. Notice that all optimization algorithms maintain similar behavior towards differential uniformity and absolute indicator (GaT slightly improves  $AC_{max}(S)$  for  $8 \times 8$  s-boxes), due the fact that these properties were not considered in the optimization process, and any improvement on the same is result of existing relation to nonlinearity.

Since no reasoning in the convergence rate of algorithms for s-boxes of sizes lower than  $8 \times 8$  was presented in [17,24], we cannot establish a fair comparison with the results for such s-box spaces. However, for  $8 \times 8$  s-boxes, we can make a direct comparison with the results of LSA and GaT.

For both the local search algorithm and the genetic and tree algorithm, the obtained results are better than those presented in [24] for the best configuration of Picek's cost function. The notorious difference for both algorithms using our cost function with regard to Picek's cost function is the average solution evaluations to obtain nonlinearity 104. The local search algorithm reduces more than 22,500 solution evaluations from the best result with PCF in the result presented in Table 6. In addition, the GaT algorithm shows greater improvement, reducing more than 50,000 solution evaluations to obtain the aforementioned nonlinearity. However, the best performance is achieved by hill climbing algorithm. Notice that hill climbing only needs approximately 1000 solution evaluations to obtain s-boxes having nonlinearity value 102, equal to the best nonlinearity reached by Clark's cost function when  $R = 3, X = 4$  and the most repeated result for Picek's cost function in its initial version, both, after nine million solution evaluations [24]. To obtain nonlinearity 104, the hill climbing algorithm reduces to half the best performance of the algorithms in [24]. Moreover, the worst performance of HC is better than the best average solution evaluations for an algorithm using Picek's cost function in [24] GaT by approximately 30,000 solution evaluations. Substitution boxes having nonlinearity value equal to 104 were obtained in less than 35,000 solution evaluations through the hill climbing algorithm. Figure 2 shows the convergence rate for each algorithm referred in Table 6.

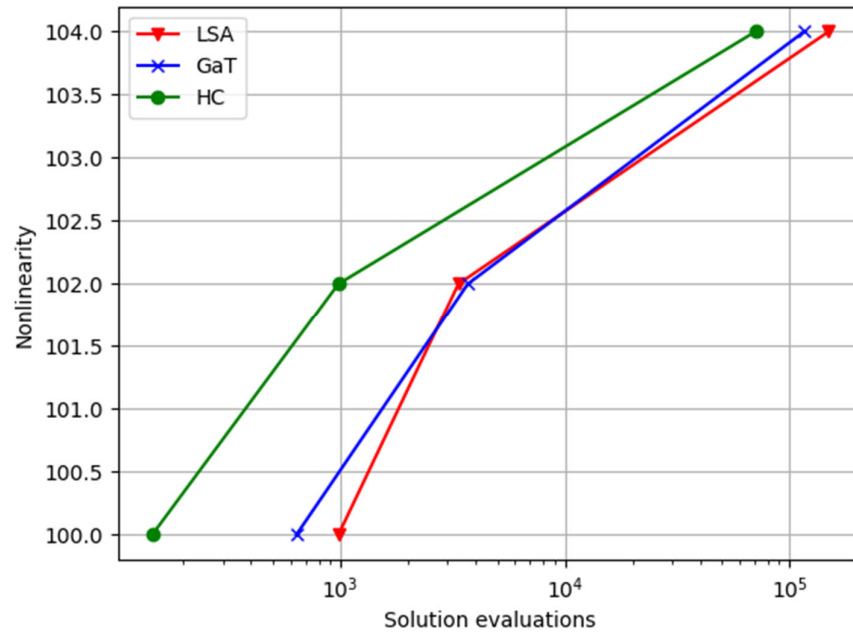


Figure 2. Convergence of algorithms for our new cost function.

We perform larger experiments to search for s-boxes having nonlinearity greater than 104, but no interesting results were found starting from random s-boxes. Hence, other seeding methods or more complex optimization algorithms should be tested with our new cost function, in order to obtain s-boxes with higher nonlinearity values than the achieved in this paper.

## 6. Conclusions

In this paper, we presented an effective novel cost function for evolving highly nonlinear s-boxes based on the existing bounds for the values in the Walsh–Hadamard spectrum. Altogether, we removed the cost function, the intervention of external parameters unrelated to the coefficients in the spectrum that may affect the performance of the older cost functions, and increase the correlation with the non-linearity property. We also showed that our new function is capable of producing the same results as other important cost functions in a lower number of solution evaluations; therefore, it is more effective for evolution of s-boxes. However, it is still an open problem as to how to achieve s-boxes with a nonlinearity higher than presented in Table 5 for an optimization algorithm seeded with random s-boxes in a reasonable amount of solution evaluations.

Future investigations will be directed to the study of the autocorrelation and differential spectrum of s-boxes for the definition of new cost functions that help to improve other properties, such as an absolute indicator and differential uniformity. We also plan to include our cost function in a trade-off multi-objective optimization related to the resistance against other attacks different from classical linear and differential attacks.

**Author Contributions:** This paper is the result of collaborated work of all the authors in all aspects. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Some Substitution Boxes with the Best Nonlinearity Achieved

$S_5 = \{7, 12, 21, 29, 17, 23, 10, 26, 31, 16, 27, 30, 28, 3, 5, 15, 8, 18, 1, 19, 9, 2, 11, 25, 4, 0, 6, 20, 22, 24, 13, 14\}$

$S_6 = \{28, 17, 1, 13, 38, 41, 46, 6, 0, 29, 53, 20, 59, 5, 24, 21, 35, 26, 61, 16, 48, 37, 58, 22, 55, 44, 39, 56, 34, 43, 15, 27, 51, 12, 62, 52, 14, 33, 10, 2, 31, 49, 57, 42, 11, 3, 30, 40, 36, 19, 63, 60, 4, 45, 32, 50, 47, 23, 54, 7, 9, 25, 8, 18\}$

$S_7 = \{14, 73, 38, 4, 71, 99, 108, 19, 61, 39, 82, 55, 60, 95, 125, 106, 23, 117, 119, 76, 120, 46, 7, 34, 96, 86, 28, 31, 81, 41, 42, 57, 87, 0, 79, 116, 13, 52, 74, 3, 33, 43, 25, 47, 68, 53, 16, 121, 85, 15, 17, 32, 63, 37, 26, 93, 29, 107, 6, 126, 22, 66, 124, 77, 89, 102, 30, 59, 75, 24, 40, 127, 88, 70, 98, 54, 109, 1, 48, 35, 112, 11, 90, 111, 92, 8, 114, 2, 94, 122, 80, 56, 100, 78, 27, 72, 103, 84, 45, 115, 101, 113, 20, 9, 44, 36, 123, 62, 64, 69, 83, 51, 104, 58, 105, 49, 12, 67, 21, 110, 10, 65, 5, 91, 97, 18, 50, 118\}$

$S_8 = \{236, 60, 6, 66, 185, 96, 206, 221, 167, 103, 159, 174, 156, 152, 239, 200, 45, 28, 38, 136, 107, 243, 34, 8, 71, 115, 201, 114, 157, 91, 47, 233, 16, 124, 70, 122, 46, 183, 203, 104, 158, 58, 179, 63, 166, 247, 123, 74, 230, 188, 72, 39, 145, 139, 210, 106, 125, 246, 205, 253, 204, 13, 20, 142, 56, 10, 128, 191, 198, 55, 197, 216, 116, 105, 195, 224, 144, 141, 22, 169, 138, 199, 154, 49, 234, 244, 121, 119, 252, 249, 153, 0, 112, 127, 75, 42, 160, 130, 209, 9, 193, 213, 172, 35, 102, 220, 109, 51, 53, 242, 180, 151, 120, 170, 111, 14, 44, 110, 215, 79, 255, 97, 11, 29, 99, 228, 81, 189, 147, 4, 192, 176, 184, 76, 163, 23, 26, 254, 5, 61, 133, 143, 214, 32, 37, 222, 88, 84, 171, 1, 146, 85, 30, 15, 212, 162, 187, 40, 41, 92, 94, 113, 186, 95, 232, 245, 12, 227, 235, 50, 150, 126, 148, 219, 100, 62, 131, 65, 68, 82, 226, 2, 57, 59, 223, 135, 80, 251, 31, 7, 36, 25, 155, 108, 98, 218, 118, 86, 101, 134, 54, 48, 78, 149, 21, 90, 207, 225, 217, 231, 18, 73, 69, 202, 129, 67, 24, 178, 237, 27, 240, 87, 137, 196, 161, 19, 168, 140, 173, 182, 190, 33, 229, 181, 3, 241, 83, 89, 17, 64, 132, 238, 208, 177, 93, 77, 211, 175, 165, 52, 250, 248, 194, 164, 117, 43\}$

## Appendix B. Pseudo-Code of the Local Search Algorithm

**Require:** a random substitution  $S$

**Require:** the number of solution evaluations  $NoE$  (given in Table 1)

**Require:** set of mutation operator given in [18] with parameters  $k, l$

**while**  $NoE > 0$  **do:**

    Generate a population of neighbors of  $S$  applying the mutation operators supplied in the input. The population is denoted as  $N$ .

**for each** s-box  $S'$  in  $N$  **do**

**if**  $N_{S'} > N_S$  or  $(N_{S'} = N_S$  and  $C_{S'} < C_S)$  **then**

$S \leftarrow S'$

$NoE = NoE - 1$

**Return**  $S$

## Appendix C. Pseudo-Code of the Hill Climbing Algorithm

**Require:** a random substitution  $S$

**Require:** the number of solution evaluations  $NoE$  (given in Table 1)

**while**  $NoE > 0$  **do:**

$S' \leftarrow S$

    Select at random two different positions  $i$  and  $j$  and swap the outputs on  $S'$  corresponding to  $i$  and  $j$

**if**  $N_{S'} > N_S$  or  $(N_{S'} = N_S$  and  $C_{S'} < C_S)$  **then**

$S \leftarrow S'$

$NoE = NoE - 1$

**Return**  $S$

## Appendix D. Procedure of the Genetic and Tree Algorithm

**Require:** the number of solution evaluations  $NoE$  (given in Table 1)

**Require:** the size of the farm  $M$  and the number of successors  $C$  for each for each s-box in the farm

**Require:** parameters  $NT$  and  $NEL$  (see Section 4.3 and Table 2)

**Step 1:** Randomly generate  $M \times C$  s-boxes which is taken as the initial population  $P$ . Save the current best nonlinearity in  $P$  to the value  $CN$ . If  $CN \geq NEL$  return the s-box with nonlinearity  $CN$ . If  $CN \geq NT$ , then proceed to **Step 4**. Otherwise, sort  $P$  according to the value of the function  $CF$  (our cost function) and select the best  $M$  candidates to be the new farm, reduce  $NoE$  in  $M \times C$  and proceed to **Step 2**.

**Step 2:** For each  $S$ , randomly select  $C$  successors from the neighborhood  $N(S)$  resulting of swap the outputs corresponding to a pair of different inputs of  $S$ , where  $S$  can also be chosen, and repetition is permitted. Then, for all  $S$  from the new population, determine the cost function  $CF$  and the nonlinearity. Update  $CN$ , reduce  $NoE$  in  $M \times C$  and proceed to **Step 3**.

**Step 3:** If any s-box in  $P$  present nonlinearity greater or equals  $NEL$  return such s-box. In the case of  $NT \leq N_S \in P \leq NEL$ , then proceed to **Step 4**. Otherwise, sort  $P$  according to the value of the function  $CF$  (and select the best  $M$  to become the new farm and proceed to **Step 2**).

**Step 4:** Let  $S_1$  be the s-box with nonlinearity higher or equal to  $NT$  and with cost  $CF(S_1)$ . Put  $CN = NT$ . Search overall  $N(S_1)$ , until:

- (a) find s-box  $S'$  with nonlinearity equal to  $NEL$ , then return  $S'$
- (b) find s-box with nonlinearity greater than  $CN$ , then proceed to **Step 5**
- (c) find s-box with nonlinearity equal to  $CN$  and a improved cost than  $CF(S_1)$ , then proceed to **Step 5**.
- (d) no s-box from  $N(S_1)$  has values to go to option a, b or c, or the number of solution evaluations is depleted, then **FAIL** (unless the number of solution evaluations is depleted, one can ignore this failure condition to have the algorithm perform more exploration of the space, always returning to **Step 3**).

**Step 5:** Let  $S_2$  be an s-box from option **Step 4** b) or c). Set  $CN = N_{S_2}$ . Save  $S_1$  with ordinal number of  $S_2$  over  $N(S_1)$  into a *LIFO* (Last In First Out) stack and set  $j = 2$  (ordinal number of the diagnostic s-box). Then proceed to **Step 6**.

**Step 6:** Search the overall  $N(S_j)$  until:

- a) find s-box  $S'$  with nonlinearity equal to  $NEL$ , then return  $S'$
- b) find s-box with nonlinearity greater than  $CN$ , then proceed to **Step 7**
- c) find s-box with nonlinearity equal to  $CN$  and a better cost than  $CF(S_j)$ , then proceed to **Step 7**.
- d) the number of solution evaluations is depleted, then **FAIL**
- e) no s-box from  $N(S_j)$  has values to go to option a, b or c, then proceed to **Step 8**

**Step 7:** Set  $j = j + 1$ . Let  $S_j$  be the s-box from the option **Step 6** b) or c). Set  $CN = N_{S_j}$ . Save  $S_{j-1}$  with ordinal number of  $S_j$  over  $N(S_{j-1})$  into the *LIFO* stack. Then proceed to **Step 6**.

**Step 8:** Set  $j = j - 1$ . If  $j=0$ , then **FAIL** (this failure condition can be omitted like option d in **Step 4**), else resume  $S_j$  from *LIFO* stack and proceed to **Step 6**.

## References

1. Coppersmith, D. The Data Encryption Standard (DES) and its strength against attacks. *IBM J. Res. Dev.* **1994**, *38*, 243–250.
2. Daemen, J.; Rijmen, V. *The Design of Rijndael*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2.
3. Barreto, P.; Rijmen, V. The Khazad legacy-level block cipher. *Primit. Submitt. NESSIE* **2000**, *97*, 106.
4. Piret, G.; Roche, T.; Carlet, C. PICARO—A block cipher allowing efficient higher-order side-channel resistance. In Proceedings of the International Conference on Applied Cryptography and Network Security, Singapore, 26–29 June 2012.
5. Vaudenay, S.; Junod, P. Device and Method for Encrypting and Decrypting a Block of Data. U.S. Patent 7,499,542, 3 March 2009.
6. Farah, M.B.; Kachouri, A.; Samet, M. Improvement of cryptosystem based on iterating chaotic map. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 2543–2553.
7. Ahmad, M.; Khaja, I.A.; Baz, A.; Alhakami, H.; Alhakami, W. Particle Swarm Optimization Based Highly Nonlinear Substitution-Boxes Generation for Security Applications. *IEEE Access* **2020**, *8*, 116132–116147.
8. Abd EL-Latif, A.A.; Abd-El-Atty, B.; Venegas-Andraca, S.E. Venegas-Andraca, A novel image steganography technique based on quantum substitution boxes. *Opt. Laser Technol.* **2019**, *116*, 92102.

9. Zahid, A.H.; Al-Solami, E.; Ahmad, M. A Novel Modular Approach Based Substitution-Box Design for Image Encryption. *IEEE Access* **2020**, doi:10.1109/ACCESS.2020.3016401.
10. Matsui, M. Linear cryptanalysis method for DES cipher. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Lofthus, Norway, 23–27 May 1993.
11. Biham, E.; Shamir, A. *Differential Cryptanalysis of the Data Encryption Standard*; Springer Science & Business Media: New York, NY, USA, 1993.
12. Budaghyan, L.; Carlet, C.; Pott, A. New classes of almost bent and almost perfect nonlinear polynomials. *IEEE Trans. Inf. Theory* **2006**, *52*, 1141–1152.
13. Nyberg, K. On the construction of highly nonlinear permutations. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Balatonfured, Hungary, 24–28 May 1992.
14. Ahmad, M.; Haleem, H.; Khan, P.M. A new chaotic substitution box design for block ciphers. In Proceedings of the 2014 International Conference on Signal Processing and Integrated Networks (SPIN), Delhi, India, 20–21 February 2014; pp. 255–258.
15. Farah, M.B.; Guesmi, R.; Kachouri, A.; Samet, M. A new design of cryptosystem based on S-box and chaotic permutation. *Multimed. Tools Appl.* **2020**, *79*, 19129–19150.
16. Clark, J.A.; Jacob, J.L.; Stepney, S. The design of S-boxes by simulated annealing. *New Gener. Comput.* **2005**, *23*, 219–231.
17. Tesar, P. A new method for generating high non-linearity s-boxes. *Radioengineering* **2010**, *19*, 23–26.
18. Ahmad, M.; Ahmad, Z. Random search based efficient chaotic substitution box design for image encryption. *IJRSDA* **2018**, *5*, 131–147.
19. Carlet, C. *Vectorial Boolean Functions for Cryptography, Encyclopedia of Mathematics and its Applications*; Cambridge University Press: Cambridge, UK, 2010; pp. 398–470.
20. Kazymyrov, O.; Kazymyrova, V.; Oliynykov, R. A Method for Generation of High-Nonlinear S-Boxes Based On Gradient Descent. *IACR Cryptol. ePrint Arch.* **2013**, *2013*, 578.
21. Ivanov, G.; Nikolov, N.; Nikova, S. Cryptographically strong S-boxes generated by modified immune algorithm. In Proceedings of the International Conference on Cryptography and Information Security in the Balkans, Koper, Slovenia, 3–4 September 2015.
22. Ivanov, G.; Nikolov, N.; Nikova, S. Reversed genetic algorithms for generation of bijective s-boxes with good cryptographic properties. *Cryptogr. Commun.* **2016**, *8*, 247–276.
23. Picek, S.; Papagiannopoulos, K.; Ege, B.; Batina, L.; Jakobovic, D. Confused by confusion: Systematic evaluation of DPA resistance of various s-boxes. In Proceedings of the International Conference in Cryptology in India, Dehli, India, 14–17 December 2014.
24. Picek, S.; Cupic, M.; Rotim, L. A new cost function for evolution of s-boxes. *Evol. Comput.* **2016**, *24*, 695–718.
25. Picek, S.; Mariot, L.; Leporati, A.; Jakobovic, D. Evolving S-boxes based on cellular automata with genetic programming. In Proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference Companion, Berlin, Germany, 15–19 July 2017.
26. Picek, S.; Mariot, L.; Yang, B.; Jakobovic, D.; Mentens, N. Design of S-boxes defined with cellular automata rules. In Proceedings of the Computing Frontiers Conference, Siena, Italy, 15–17 May 2017.
27. Isa, H.; Jamil, N.; Z'aba, M. Hybrid heuristic methods in constructing cryptographically strong S-boxes. *Int. J. Cryptol. Res.* **2016**, *6*, 1–15.
28. Menyachikhin, A. Spectral-linear and spectral-differential methods for construction of S-boxes with cryptographic parameters close to optimal values. *Mat. Vopr. Kriptografii*, **2017**, *8*, 97–116.
29. Lerman, L.; Veshchikov, N.; Picek, S.; Markowitch, O. On the construction of side-channel attack resilient s-boxes. In Proceedings of the International Workshop on Constructive Side-Channel Analysis and Secure Design, Paris, France, 13–14 April 2017.
30. Martínez-Díaz, I. Búsqueda Local De S-Cajas Con Alta Varianza Del Coeficiente De Confusión. Master's Thesis. Faculty of Math and Computer Sciences, University of Havana, Havana, Cuba, May 2019.
31. Freyre-Echevarría, A. Evolución Híbrida De S-Cajas No Lineales Resistentes A Ataques De Potencia. Bachelor's Thesis. Faculty of Math and Computer Sciences, University of Havana, Havana, Cuba, July 2020.
32. Bolufé-Röhler, A.; Tamayo-Vera, D. Machine learning based metaheuristic hybrids for S-box optimization. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 5139–5152.

33. Ahmad, M.; Al-Solami, E. Evolving dynamic S-boxes using fractional-order hopfield neural network based scheme. *Entropy* **2020**, *22*, 717.
34. Ahmad, M.; Al-Solami, E.; Alghamdi, A.M.; Yousaf, M.A. Bijective S-Boxes Method Using Improved Chaotic Map-Based Heuristic Search and Algebraic Group Structures. *IEEE Access* **2020**, *8*, 110397–110411.
35. Ahmad, M.; Doja, M.N.; Beg, M.M.S. ABC optimization based construction of strong substitution-box. *Wirel. Pers. Commun.* **2018**, *101*, 1715–1729.
36. Belazi, A.; El-Latif, A.A.A.; Rhouma, R.; Belghith, S. Selective image encryption scheme based on DWT, AES S-box and chaotic permutation. In Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24–28 August 2015; pp. 606–610.
37. Razaq, A.; Alolaiyan, H.; Ahmad, M.; Yousaf, M.A.; Shuaib, U.; Aslam, W.; Alawida, M. A Novel Method for Generation of Strong Substitution-Boxes Based on Coset Graphs and Symmetric Groups. *IEEE Access* **2020**, *8*, 75473–75490.
38. Amin, M.; Abd El-Latif, A.A. Efficient modified RC5 based on chaos adapted to image encryption. *J. Electron. Imaging* **2010**, *19*, 013012.
39. Peng, J.; Abd El-Latif, A.A.; Belazi, A.; Kotulski, Z. Efficient chaotic nonlinear component for secure cryptosystems. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 989–993.
40. Zahid, A.H.; Arshad, M.J.; Ahmad, M. A novel construction of efficient substitution-boxes using cubic fractional transformation. *Entropy* **2019**, *21*, 245.
41. Canteaut, A. Lecture notes on Cryptographic Boolean Functions. Inria Paris France. 2016. Available online: <https://www.rocq.inria.fr/secret/Anne.Canteaut/poly.pdf> (accessed on 1 October 2020).
42. Rothaus, O.S. On “bent” functions. *J. Comb. Theory Ser. A* **1976**, *20*, 300–305.
43. Chabaud, F.; Vaudenay, S. Links between differential and linear cryptanalysis. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, 9–12 May 1994.
44. Zhang, X.-M.; Zheng, Y. GAC—The criterion for global avalanche characteristics of cryptographic functions. *J. Univers. Comput. Sci.* **1996**, *1*, 320–337.
45. Browning, K.; Dillon, J.; McQuistan, M.; Wolfe, A. An APN permutation in dimension six. *Finite Fields Theory Appl.* **2010**, *518*, 33–42.

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).