electronics

MDPI

Article

Optimizing Hill Climbing Algorithm for S-Boxes Generation

Alexandr Kuznetsov [1,2,*], Emanuele Frontoni [1,3], Luca Romeo [3,4], Nikolay Poluyanenko [2], Sergey Kandiy [2], Kateryna Kuznetsova [2] and Eleonóra Beňová [5]

[1] Department of Political Sciences, Communication and International Relations, University of Macerata, Via Crescimbeni, 30/32, 62100 Macerata, Italy
[2] Department of Information and Communication Systems Security, Faculty of Comupter Science, V. N. Karazin Kharkiv National University, 4 Svobody Sq., 61022 Kharkiv, Ukraine
[3] Department of Information Engineering, Marche Polytechnic University, Via Brecce Bianche 12, 60131 Ancona, Italy
[4] Department Economics and Law, University of Macerata, Piazza Strambi 1, 62100 Macerata, Italy
[5] Faculty of Management, Comenius University Bratislava, Odbojárov 10, 820-05 Bratislava, Slovakia
* Correspondence: kuznetsov@karazin.ua

Abstract: Nonlinear substitutions or S-boxes are important cryptographic primitives of modern symmetric ciphers. They are designed to complicate the plaintext-ciphertext dependency. According to modern ideas, the S-box should be bijective, have high nonlinearity and algebraic immunity, low delta uniformity, and linear redundancy. These criteria directly affect the cryptographic strength of ciphers, providing resistance to statistical, linear, algebraic, differential, and other cryptanalysis techniques. Many researchers have used various heuristic search algorithms to generate random S-boxes with high nonlinearity; however, the complexity of this task is still high. For example, the best-known algorithm to generate a random 8-bit bijective S-box with nonlinearity 104 requires high computational effort—more than 65,000 intermediate estimates or search iterations. In this article, we explore a hill-climbing algorithm and optimize the heuristic search parameters. We show that the complexity of generating S-boxes can be significantly reduced. To search for a random bijective S-box with nonlinearity 104, only about 50,000 intermediate search iterations are required. In addition, we generate cryptographically strong S-Boxes for which additional criteria are provided. We present estimates of the complexity of the search and estimates of the probabilities of generating substitutions with various cryptographic indicators. The extracted results demonstrate a significant improvement in our approach compared to the state of the art in terms of providing linear non-redundancy, nonlinearity, algebraic immunity, and delta uniformity.

Keywords: hill-climbing algorithm; heuristic function; mathematical optimization; bijective substitution; local search

Citation: Kuznetsov, A.; Frontoni, E.; Romeo, L.; Poluyanenko, N.; Kandiy, S.; Kuznetsova, K.; Beňová, E. Optimizing Hill Climbing Algorithm for S-Boxes Generation. *Electronics* **2023**, *12*, 2338. https://doi.org/10.3390/electronics12102338

Academic Editor: Hung-Yu Chien

Received: 5 March 2023
Revised: 5 May 2023
Accepted: 18 May 2023
Published: 22 May 2023

1. Introduction

Nonlinear substitutions (S-boxes) have been used for a long time in cryptography. For example, in one of the basic works in cryptography, Claude Shannon pointed out the use of substitutions and permutations [1]. Modern cryptographic algorithms make extensive use of this concept. For instance, the AES standard, established by the US National Institute of Standards and Technology (NIST), uses an 8-bit bijective S-box, which is formed by a simple algebraic construction [2]. This algebraic simplicity is used to criticize AES. It is assumed that there are effective algebraic attacks on AES [3–5], i.e., according to modern ideas, S-boxes in a secure cipher should be random. At the same time, random generation is computationally inefficient. For an 8-bit S-box, the set of possible solutions is huge. It contains $2^8! > 10^{506}$ values. It is really difficult to find a target S-box (with the required cryptographic indicators) [6–8]. For example, the nonlinearity of a randomly chosen substitution rarely $N(S) > 98$ [7].

Heuristic techniques are A good alternative to random generation [6,9–11]; instead of brute force, heuristic algorithms use a limited subset of values. The number of these intermediate estimates (search iterations) is used as a criterion for computational complexity.

Compared to random generation, heuristic techniques enable the finding of 8-bit bijective S-boxes with nonlinearity $N(S) = 104$. However, the computational complexity of the search remains quite high. The performed analysis shows that the best known heuristic algorithm requires more than 65,000 intermediate estimates (search iterations).

It is worth noting that, according to modern concepts, a cryptographically strong S-box must simultaneously satisfy several criteria, including being bijective, and having high nonlinearity and algebraic immunity, low delta uniformity, and linear redundancy. If all these criteria are taken into account, then the complexity of the search increases significantly.

Thus, a relevant task within the cryptography literature is to develop computationally efficient techniques for generating cryptographically strong S-boxes.

In this article, we propose a Hill Climbing (HC) [12,13] optimization strategy to fill in this gap. HC is one of the most effective local optimization algorithms (LSA), which is used to solve various AI problems [14–17].

HC is an iterative algorithm that, at each step, tries to maximize (or minimize) the objective function $f(x)$, where x is a vector of continuous and/or discrete data. At each iteration, HC changes one element in x and determines how it changes the value of intermediate estimates. The iterative process continues until (1) there is no way to find such x, to improve the heuristic value $f(x)$ or (2) the target solution is reached. The found value $f(x)$ is the «local optimum» and the search is terminated.

Thus, the objective function $f(x)$ has to interpret the cryptographic parameters of the S-boxes in some (heuristic) way. Our task is to find a point x (to generate a target S-box), for which there will be optimized heuristics $f(x)$ and cryptographic performance. As target indicators of cryptographically strong 8-bit S-boxes we propose:

- Bijectivity;
- Nonlinearity $N(S) \geq 104$,
- Algebraic immunity $AI(S) \geq 3$,
- Delta uniformity $\delta \leq 8$,
- Linear non-redundancy, i.e., the number of affine non-equivalent component Boolean functions $NnonLE(S) = 255$.

The most difficult task is to generate highly nonlinear random substitutions. Apparently, the value $N(S) = 104$ is the best-known result for random bijective S-boxes and our main goal is this target result.

We consider the cost function based on Walsh–Hadamard Spectra (WHS heuristic function) [18] as one of the first and most studied heuristic functions $f(x)$ used to generate S-boxes [19–21]. Various works provide estimates of the efficiency of local search using WHS [10,19,20]. Although most sources [10,20] point out more efficient heuristics (in combination with other search algorithms), we demonstrated the low generalizability of these approaches. Several experiments show that with the WHS recommend parameters, it is possible to significantly increase the efficiency of local search. We present the results of the comparison and show that HC with WHS function can be more effective than many current techniques. Actually, we manage to obtain the least number of iterations (intermediate estimates) to generate an S-box with $N(S) \geq 104$. We also show that we can generate target S-boxes (with indicators: bijectivity, $N(S) \geq 104$, $AI(S) \geq 3$, $\delta \leq 8$, $NnonLE(S) = 255$) faster (i.e., with less number of search iterations).

The article is structured as follows. Section 2 provides a brief review of the literature sources with the analysis of the obtained results. Section 3 presents the main notation and background of the research. Section 4 describes the proposed heuristic search methodology. Section 5 presents the main results obtained by the proposed approach. Finally, Section 6 provide a discussion about the results and our conclusions. In the list of references, we provide a link to the GitHub repository, where you can find the developed algorithms. The code allows independently reproducing and verifying our results.

2. Related Works

Many modern works are devoted to the problems of finding new ways to improve the efficiency of modern information protection systems, including new ways to generate S-boxes [22–24]. Many of them are based on advanced AI techniques, evolutionary algorithms, chaos, heuristic search, etc. [25–28].

Heuristic algorithms do not use brute force, but select only a subset of solutions (intermediate estimates) [13,29,30]. At the same time, intuitively introduced functions (heuristics) are used to rank decision alternatives [31,32].

Usually, heuristic techniques in AI are used to quickly find a local optimum, i.e., they make it possible to find an acceptable (target) solution to the optimization problem in a short time and at reasonable computational costs. For example, the article [7] studies the HC algorithm. The authors have significantly reduced the difficulty of finding target S-boxes; however, the authors did not obtain a nonlinearity value of $N(S) \geq 100$.

In [6,18,33], the authors present the simulated annealing method (SA). They have reduced the search time and formed an S-box with a nonlinearity value of $N(S) = 102$. In addition, the article [18] proposed a new heuristic function based on Walsh–Hadamard Spectra (the WHS cost function), which is used in many other later works.

The works [19,21,34] present the genetic algorithms (GA) and the immune algorithms (IA) for solving this task. In [19], the authors proposed a new method that combines a special version of the genetic method with the tree search, the name of the method is GaT («Genetic and Tree»). The author obtained an S-box with a nonlinearity value of $N(S) = 104$. The WHS function was also used as a heuristic, for which the author has conducted a large study to select the best parameters. In all subsequent works, the parameters of the WHS heuristic from [21] are mainly used. However, the average number of iterations (intermediate WHS estimates) required to generate a S-box with $N(S) = 104$ using GaT is still big (more than 3 million) [19,35].

Subsequent works have focused on finding the best heuristics. The work [20] proposes a new heuristic PCF (Picek Cost Function). The authors reviewed several heuristic algorithms (GA, GaT, LSA) and compared their performance in combination with the WHS and PCF functions. In almost all their comparisons, algorithms with the new PCF heuristic turned out to be more efficient (it should be noted that WHS was used with parameters from [19]). The best result achieved in [20] was (on average) 167,451 iterations (intermediate PCF estimates) to create an S-box with $N(S) = 104$. The GaT algorithm turned out to be the best again.

The recent papers [10,35] propose a new heuristic WCF (Cost Function of the content of the Walsh–Hadamard spectrum). The authors combined WCF with three algorithms (LSA, GaT, HC) and compared the results with respect to [20]. It turned out that the new WCF heuristic significantly speeds up the search for a S-box with $N(S) = 104$. For instance, for GaT, the average number of iterations (intermediate WCF estimates) decreased to 116,266 [10]. However, the improvement of the HC algorithm is even more interesting. In combination with the new WCF function, it turned out to be the fastest search algorithm, the average number of iterations (number of evaluations) was reduced to 70,596 [10]. In [35], the authors report an even greater improvement. In particular, the average number of iterations (number of estimates) for the HC algorithm is reduced to 65,933.

The logical conclusion is that the choice of the cost function significantly affects the efficiency of the search. This brief analysis of the related works clearly confirms this reasoning. Considered in 1998 [7], the HC algorithm turned out to be more efficient than many modern algorithms [10,35]. In this article, we show that the WHS cost function proposed in [18] is also very efficient.

We assumed that heuristic parameter optimization has to be performed for each heuristic algorithm separately. Till now, WHS has been optimized for SA in [6,18] and for GaT in [19]. Differently from the other state-of-the-art research in this paper, we present our own interpretation of the HC algorithm. In particular, we optimize the WHS parameters

for the HC algorithm and estimate the computational complexity of generating the S-box with $N(S) = 104$.

Our experiments show that the optimized WHS parameters applied to the HC algorithm enable the obtaining of the smallest average number of iterations (number of estimates). We need to perform an average of 50,265 iterations to generate the S-box with $N(S) = 104$. For us, this is the best-known result to date. In addition, we show that our technique makes it possible to generate target S-boxes with additionally introduced cryptographic indicators (bijectivity, $N(S) \geq 104$, $AI(S) \geq 3$, $\delta \leq 8$, $NnonLE(S) = 255$). Such cryptographically strong S-boxes are currently integrated and used in modern symmetric ciphers [36–39].

3. Background

The main object of research in our work is 8-bit substitution. Schematically, it can be represented as a block (S-box) with 8 binary inputs $x_0, x_1, x_2, \ldots, x_7$ and 8 binary outputs $y_0, y_1, y_2, \ldots, y_7$ (Figure 1).

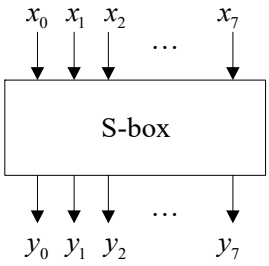

Figure 1. Block diagram of 8-bit substitution.

Thus, an 8-bit S-box implements a Boolean mapping $S : \{0,1\}^8 \to \{0,1\}^8$, which is given by a system of algebraic equations (coordinate Boolean functions) over a binary field:

$$\begin{cases} f_0(x_0, x_1, \ldots, x_7) = y_0, \\ f_2(x_0, x_1, \ldots, x_7) = y_1, \\ \ldots \\ f_7(x_0, x_1, \ldots, x_7) = y_7. \end{cases} \quad (1)$$

For compact notation, we use the definition of $(8,8)$—function $F(x) = (f_1, f_2, \ldots, f_8)$ (Boolean function with several outputs) [40].

A bijective substitution (S-box) can be specified as a vector of 256 non-repeating 8-bit integers, where the element's position is given by $x = (x_0, x_1, x_2, \ldots, x_7)$, and the element's value is given by $y = F(x)$.

The 8-bit S-boxes are widely used in modern cryptography; for example, they are used in symmetric ciphers to complicate the functional dependence of plaintext and ciphertext [41]. Some cryptographic indicators of ciphers are directly determined by indicators of S-boxes [2].

Let us consider the most important S-boxes indicators used in this work.

The nonlinearity $N(S)$ of the substitution is calculated as the minimum nonlinearity over the set of nonlinearities of coordinate Boolean functions and all their linear combinations (see, for example, Definition 1 in [40]):

$$N(S) = \min_{v \in \{0,1\}^8 \setminus \{0\}^8} \{N(v \cdot F(x))\} \quad (2)$$

The nonlinearity of a Boolean function $f(x_0, x_1, \ldots, x_7)$ is defined in terms of the Hamming distance between its outputs and the outputs of all affine functions. This is equivalent to

$$N(f) = \frac{1}{2}(2^8 - WHT_{\max}),$$

where $WHT_{\max}$ is the maximum absolute value in the Walsh–Hadamard spectrum (see, for example, (5) in [40]):

$$WHT_{\max} = \max_{u,v \in \{0,1\}^8 \setminus \{0\}^8} |WHT(v \cdot F(x), u)|,$$

$$WHT(f(x), u) = \sum_{x \in \{0,1\}^8} (-1)^{f(x) \oplus u \cdot x}.$$

The nonlinearity of S-boxes is the most important performance indicator, because it determines the resistance to linear cryptanalysis [42,43]. A safe substitution has to be approximated with equal probability by linear relations of the form:

$$u \cdot x = v \cdot F(x),\ u, v \in \{0,1\}^8 \setminus \{0\}^8. \tag{3}$$

In practice, this means that we can count how many times on the entire set $u, v \in \{0,1\}^8 \setminus \{0\}^8$ the equality (3) is true, i.e., make a so-called linear approximation table (LAT) from values

$$LAT(u, v) = \#\left\{ x \in \{0,1\}^8 \middle| u \cdot x = v \cdot F(x) \right\}.$$

Then, for the resistance to linear cryptanalysis the value

$$\left| LAT(u, v) - 2^7 \right|$$

should be as small as possible for all $u, v \in \{0,1\}^8 \setminus \{0\}^8$.

Maximum value in LAT

$$\max LAT = \max_{u,v \in \{0,1\}^8 \setminus \{0\}^8} \left| LAT(u, v) - 2^7 \right| \tag{4}$$

is related to the nonlinearity (2) by the formula:

$$N(S) = 2^7 - \max LAT. \tag{5}$$

The largest known nonlinearity result for a bijective S-box is given by the algebraic construction of the S-box of the AES cipher $N(S) = 112$ [2]. At the same time, such S-boxes can lead to efficient algebraic cryptanalysis [3–5]. We consider random (non-algebraic) S-boxes, which are formed by heuristic techniques of mathematical optimization.

Most of the works on heuristic methods for generating S-boxes use the indicator $N(S)$ as the main performance criterion. At the same time, many authors proposed different heuristics. For example, the authors of [18] proposes the WHS cost function,

$$WHS = \sum_{v \in \{0,1\}^8 \setminus \{0\}^8} \sum_{u \in \{0,1\}^8} ||WHT(v \cdot F(x), u)| - X|^R, \tag{6}$$

where X and R are parameters with real values.

Many related works use the WHS feature. For example, the article [19] presents a large study on heuristic search optimization using GaT and WHS. The best results are achieved with parameters $X = 21$, $R = 7$. At the same time, the GaT algorithm requires an average of 3.239 million iterations (intermediate WHS estimates) to form an S-box with a nonlinearity value of 104. This result is practically confirmed in a recent paper [35], where the average number of iterations is 3,849,881.

Another example of the PCF heuristic was proposed in [20]. Denote the vector of absolute values $|WHT(v \cdot F(x), u)|$ by $H(S)$, and the i-th position of the vector indicates

the number of coefficients WHT with the value $|4i|$, k is the maximum (last) position in this vector with a non-zero value. Then, the cost function PCF is given by

$$PCF = \sum_{i=1}^{N_p} 2^{-i} H(S)_{k-i}, \tag{7}$$

where N_p is the heuristic parameter (value recommended by the authors is $N_p = 10$).

It is shown in [20] that the PCF function allows a much more efficient implementation of searching target S-boxes. For example, to form an S-box with a nonlinearity of 104, the GaT algorithm needs 167,451 iterations (intermediate PCF estimates), on average. This greatly improves the result from [19] for the WHS function.

Recent papers [10,35] reported significant progress in the fast S-box generation of with $N(S) = 104$. The best result in [35] is achieved using the new WCF cost function

$$WCF = \sum_{v \in \{0,1\}^8} \sum_{u \in \{0,1\}^8} \prod_{z \in C} ||WHT(v \cdot F(x), u)| - z|, \tag{8}$$

where $C = \{0, 4, \ldots, 32\}$.

To form an S-box with a nonlinearity value of 104, the GaT algorithm needs 116,266 iterations (intermediate WCF estimates) [10]. The HC algorithm requires 65,933 iterations [35] to form such an S-box (the work [10] reported about 70,596 iterations). For us, this is the best known result, which significantly exceeds the previous one from [19,20].

Consider the system of algebraic Equation (1) as a set of Boolean polynomials

$$y_0 - f_0(x_0, x_1, \ldots, x_7),\ y_1 - f_1(x_0, x_1, \ldots, x_7),\ \ldots,\ y_7 - f_7(x_0, x_1, \ldots, x_7) \tag{9}$$

in a ring $K[x_0, x_1, \ldots, x_7, y_0, y_1, \ldots, y_7]$ of variables $x_0, x_1, \ldots, x_7, y_0, y_1, \ldots, y_7$ with coefficients over a binary field K.

Consider the ideal I, generated by polynomials (9):

$$I(S) = \{(y_0 - f_0) \cdot r_0 + (y_1 - f_1) \cdot r_1 + \ldots + (y_7 - f_7) \cdot r_7\},$$

where

$$r_0, r_1, \ldots, r_7 \in K[x_0, x_1, \ldots, x_7, y_0, y_1, \ldots, y_7].$$

The algebraic immunity of S-box is defined as the minimum degree of a polynomial P from the ideal $I(S)$ [44]:

$$AI(S) = \min\{\deg(P), P \in I(S) \lhd GF(2)[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m]\}, \tag{10}$$

and the minimal reduced Gröbner basis of the ideal $I(S)$ under degree reverse lexicographic order (degrevlex) contains a linear basis of polynomials P from $I(S)$, such that $AI(S) = deg(P)$.

To calculate the algebraic immunity (10), it suffices to construct a minimally reduced Gröbner basis of the ideal $I(S)$, given by Equation (9) and find a polynomial of the minimum degree among the elements of this basis. The value of minimum degree is the desired value of algebraic immunity $AI(S)$. For an 8-bit S-box, you can also use other calculation algorithms, for example, from [45] or [36,46].

The concept of algebraic S-box immunity is closely related to algebraic cryptanalysis of symmetric ciphers [3–5]. For an S-box $S : \{0,1\}^8 \to \{0,1\}^8$; to be safe, the required value is $AI(S) = 3$.

The indicator $AI(S)$ can also be used as a characteristic of S-box randomness. For a randomly generated S-box $S : \{0,1\}^8 \to \{0,1\}^8$ we almost always have a value $AI(S) = 3$. The algebraic structure of such an S-box is described by a complex system of equations, and then the cipher will be resistant to algebraic cryptanalysis. For S-boxes with a simple

algebraic structure, we have a low value $AI(S) = 2$ (for example, S-box AES and some other ciphers [36,46]). Such ciphers are potentially vulnerable to algebraic attacks.

It should be noted that some published results on the generation of random S-boxes do not take into account the indicator $AI(S)$. For example, in [21] there are examples of S-boxes with $N(S) > 104$. A direct check shows that $AI(S) = 2$. In practice, this means that such S-boxes are described by a simple system of equations, i.e., cannot be random. A simple algebraic structure can potentially lead to the vulnerability to algebraic attack [44].

Another important cryptographic indicator of S-boxes is delta uniformity [47,48]. Delta uniformity (the maximum value of the difference table) is used as an indicator of resistance to differential cryptanalysis. It characterizes the maximum probability of the appearance of a difference $\Delta y = F(x) \oplus F(x \oplus \alpha) = \beta$ at the output of the S-box (see Figure 1) with an input difference $\Delta x = x \oplus \alpha$:

$$\delta = \max_{\alpha \in \{0,1\}^8 \setminus \{0\}^8} \max_{\beta \in \{0,1\}^8} | \{ x | F(x) \oplus F(x \oplus \alpha) = \beta \} |. \quad (11)$$

The value δ has to be as small as possible. For example, the best estimate of delta-uniformity for the algebraic S-box cipher AES is $\delta = 4$. In our studies, the target S-boxes must be random and satisfy the condition $\delta \leq 8$.

The concept of linear redundancy (LR) was introduced in [49,50]. An S-box has LR if there are at least two affine equivalent functions on the entire set $v \cdot F(x)$, $v \in \{0,1\}^8 \setminus \{0\}^8$. For example, for the algebraic S-box cipher AES, all 255 component functions $v \cdot F(x)$ are affinely equivalent to each other. Such an LR is called complete [21]. The presence of LR indicates the non-randomness of S-boxes. To quantify the LR S-box, the number of affine non-equivalent component Boolean functions is usually given by $NnonLE(S)$.

To date, there are no cryptographic attacks using the LR property. However, it is assumed that the generated S-boxes does not have to contain affine equivalent Boolean functions $v \cdot F(x)$, i.e., $NnonLE(S) = 255$.

The presence of a fixed point (FP) in the S-box corresponds to the case $F(x) = x$ [21,51]. This case of identity transformation can be supplemented with an inverse fixed point $F(x) = x \oplus \{1\}^8$. For cryptographic applications, the S-box should not contain such points (this condition can be easily ensured by performing an affine transformation [52]). Thus, it is required that the number of fixed points be $NFP(S) = 0$.

4. Methods

We explore the HC algorithm, which is used to solve various mathematical optimization problems [14–17]. For example, in [10] there is the pseudocode of one of the versions of the HC algorithm for generating S-boxes (see Algorithm 1).

The pseudocode uses the nonlinear substitution definition $N(S)$ introduced above.

The value of the cost function (heuristic) is denoted by $CF(S)$.

Random substitution S is used as initial data. The Fisher–Yates method is usually used to form it [53,54].

Our extension and implementation of the HC algorithm formalize three new exit conditions:

- Reaching the limit number of solutions $Ne = 10^6$;
- Formation of the target S-box with the required nonlinearity $N(S) \geq 104$;
- Achievement of the maximum number of consecutive unsuccessful cycles $K = 10^5$. By an unsuccessful cycle we mean the case $CF(S') > CF(S)$.

Thus, the pseudocode of the HC algorithm implemented by us (see Algorithm 2) is different from [10]. In fact, we eliminated the nonlinearity estimate at step 3 of the algorithm, i.e., we accept S' for all $CF(S') \leq CF(S)$. The rationale behind that choice lies to significantly reduce the number of iterations; reduce the computational complexity of generating S-boxes with $N(S) \geq 104$.

Algorithm 1: The Hill Climbing Algorithm [10]

Input: a random substitution S, the number of solution evaluations Ne;
While $Ne > 0$ 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 $CF(S') < CF(S))$ then
$S \leftarrow S'$;
$Ne = Ne - 1$;
Return S.

Algorithm 2: The Implemented Hill Climbing Algorithm

Input:
a random substitution S;
the number of solution evaluations $Ne = 10^6$;
the nonlinearity target S-box $N(S) \geq 104$;
the maximum number of consecutive unsuccessful cycles $Nfr = 10^5$;
$k \leftarrow 0$;
While $(Ne > 0)$ and $(N(S) < 104)$ and $(k < Nfr)$ 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 $CF(S') \leq CF(S)$ then
$S \leftarrow S'$, $k \leftarrow 0$;
else
$k = k + 1$;
$Ne = Ne - 1$;
Return S.

In our research, we have focused on the WHS cost function, i.e., values $CF(S')$ and $CF(S)$ were calculated by Formula (6).

The second part of our research consisted in introducing additional requirements for S-boxes cryptographic indicators. In addition to the requirement of nonlinearity $N(S) \geq 104$ we also introduce restrictions on other indicators, algebraic immunity $AI(S) = 3$, delta uniformity $\delta \leq 8$, linear non-redundancy $(NnonLE(S) = 255)$, and no fixed points $(NFP(S) = 0)$.

Then, we can rewrite the condition for executing the while loop in the Algorithm 2:

$$«(Ne > 0) \text{ and } (N(S) < 104) \text{ and } (AI(S) < 3) \text{ and}$$

$$(\delta > 8) \text{ and } (NnonLE(S) < 255) \text{ and } (k < Nfr)».$$

The condition $(NFP(S) > 0)$ is not used in the search algorithm and we remove fixed points by the affine transformations after finding the target S-box (bijectivity, $N(S) \geq 104$, $AI(S) \geq 3$, $\delta \leq 8$, and $NnonLE(S) = 255$).

5. Results

We conducted several trial runs of the algorithm with different WHS parameters. In particular, for the entire range of parameters X and R from [18] we have never been able to form the target S-box with $N(S) \geq 104$. At the same time, S-boxes with $N(S) \geq 102$ are formed fairly quickly. Examples of changes (tracks) of nonlinearity $N(S)$ and values of cost functions $CF(S)$ for different values of the WHS parameters are shown in Figures 2–4:

- The number of iterations N_I is marked along the abscissa;
- On the y-axis (on the left) the value of the WHS cost function $CF(S)$ is marked;
- The value $N(S)$ is marked on the y-axis (on the right).

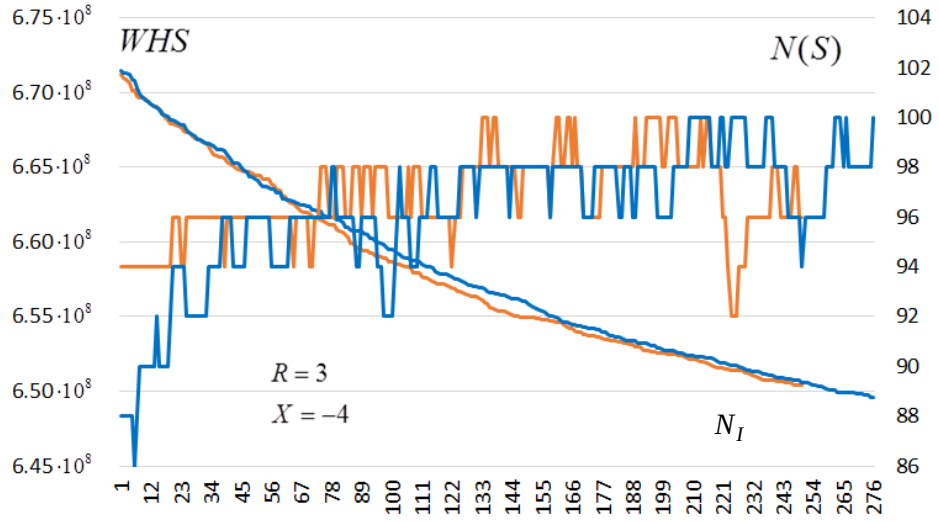

Figure 2. Changes of $N(S)$ and $CF(S)$ for $R = 3$, $X = -4$.

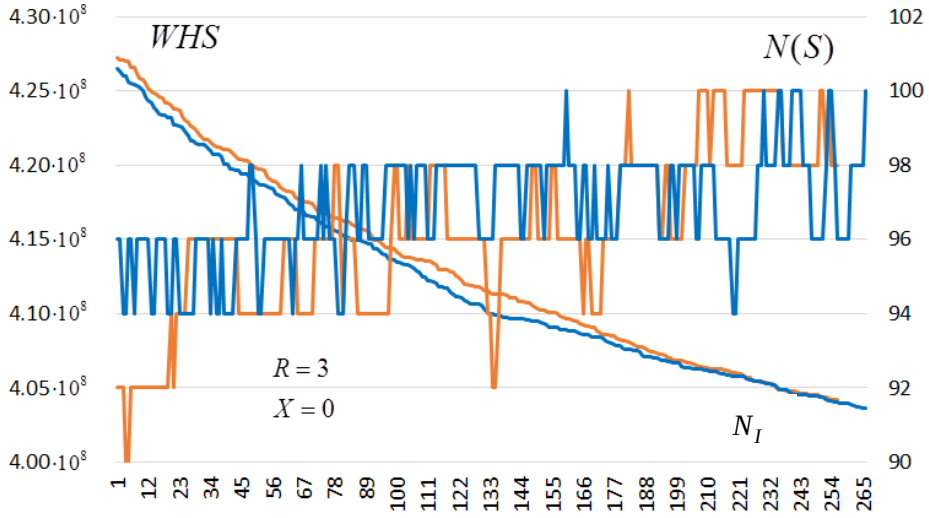

Figure 3. Changes of $N(S)$ and $CF(S)$ for $R = 3$, $X = 0$.

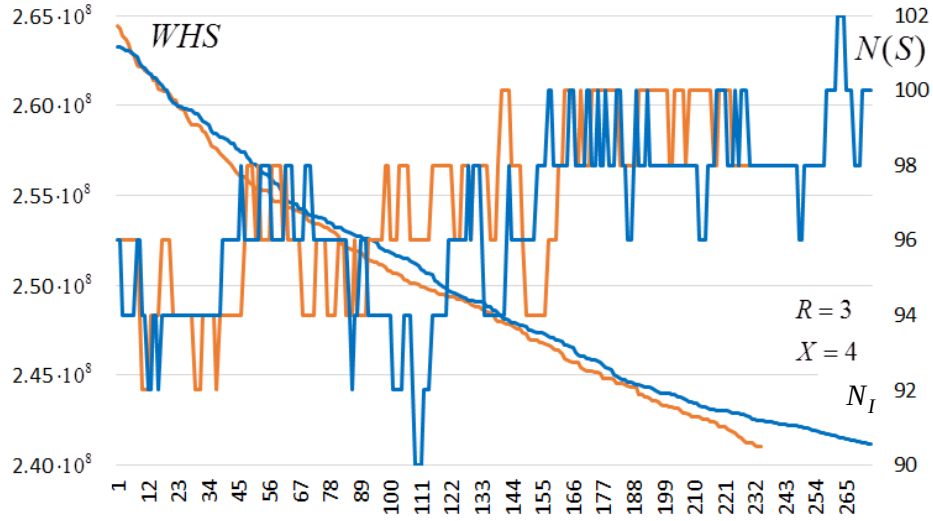

Figure 4. Changes of $N(S)$ and $CF(S)$ for $R = 3$, and $X = 4$.

We review the operation of the HC algorithm (see Algorithm 2) at each iteration and calculate the values of WHS $CF(S)$ and nonlinearity $N(S)$. Each of the figures shows two examples (marked with a different color) of iterative change (track) of the cost function $CF(S)$ and nonlinearity $N(S)$. The values of $CF(S)$ and $N(S)$ at each iteration correspond to the points on the diagram. For convenience of perception, these points are connected by lines.

In [18], there were the values X from the set $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ and $R = 3$. On Figures 2–4 we give tracks for parameters:

- $R = 3$, $X = -4$;
- $R = 3$, $X = 0$;
- $R = 3$, $X = 4$.

Different colors show different tracks.

As can be seen from the examples, the value of the WHS heuristic gradually decreases (on the graphs, only improving positions are marked, i.e., only cases when the condition $CF(S') \leq CF(S)$ is true at step 3 of the algorithm). In this case, the value of the nonlinearity generally increases. Nevertheless, the decrease in the cost function can lead to the nonlinearity decrease in some cases. This is a natural process for local optimization algorithms.

To optimize the parameters X and R in (6), we carried out extensive experimental studies, following the example of [19]. We used the HC algorithm instead of GaT from [19], as the search algorithm (see Algorithm 2).

The research results are shown in Table 1. For each combination of parameters X and R we give the number of iterations of the HC algorithm that were required to generate an S-box with $N(S) = 104$. All values were obtained by averaging over 100 runs of the algorithm. The "–" symbol denotes cases where the target S-box was found in less than 50% of launches.

The results from Table 1 are graphically reflected in Figure 5. For ease of perception, different ranges of values are shown in color.

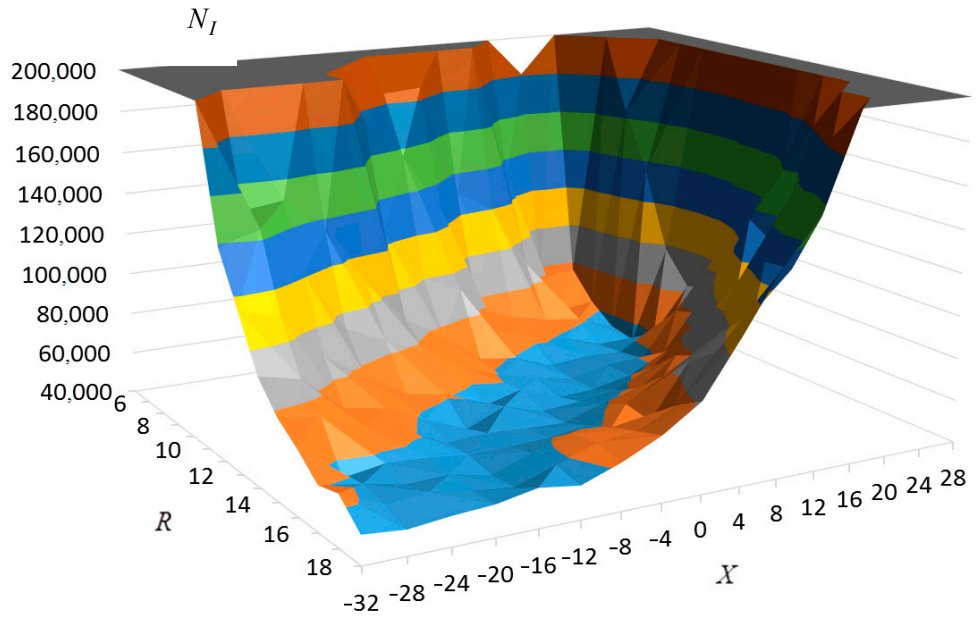

Figure 5. Average number of iterations performed before finding the target S-box, using the WHS function at different parameters X and R.

The obtained results show that when choosing $R = 12$ and $X = 0$ for the heuristic function (6), the best results are achieved in terms of the average number of iterations (intermediate estimates) for the HC algorithm. Compared to the best result published in [35], we observe a reduction in the number of iterations by more than 20%. On average,

the HC algorithm with optimized parameters WHS ($R = 12$ and $X = 0$) needs to perform 50,265 iterations.

Table 1. The average number of iterations N_I that were performed to find the S-box with $N(S) = 104$, using the WHS function.

					X			
R	**28**	**24**	**20**	**16**	**12**	**8**	**4**	**0**
5	–	–	–	–	–	–	–	–
6	–	–	–	76,407	74,028	131,640	200,386	–
7	–	–	–	63,123	61,182	88,677	119,959	130,057
8	–	–	135,248	54,553	61,261	63,599	67,845	75,202
9	–	–	97,632	52,332	52,829	58,811	62,225	68,247
10	–	–	82,720	60,552	53,835	54,412	52,746	59,391
11	–	–	86,567	77,219	55,569	55,230	51,278	54,607
12	–	–	120,302	80,965	64,079	54,808	51,477	50,265
13	–	–	132,661	108,804	74,352	64,675	56,544	54,004
14	–	–	163,762	135,713	97,179	75,484	62,673	58,345
15	–	–	181,318	127,413	113,160	80,836	74,119	63,903
16	–	–	–	164,523	126,476	98,593	86,159	61,689
17	–	–	–	–	127,581	110,609	95,163	74,467
18	–	–	–	–	155,198	134,842	107,553	82,044
R	**−4**	**−8**	**−12**	**−16**	**−20**	**−24**	**−28**	**−32**
5	–	–	–	–	–	–	–	–
6	–	–	–	–	–	–	–	–
7	185,806	184,416	–	–	–	–	–	–
8	129,162	156,697	–	–	–	–	–	–
9	92,074	100,506	124,512	–	–	–	–	–
10	61,968	76,063	87,195	108,244	134,658	151,218	148,705	–
11	59,183	65,964	76,724	84,920	93,532	121,572	144,282	138,456
12	57,436	52,491	61,249	64,744	76,193	88,271	108,969	114,841
13	52,490	57,945	50,326	58,004	66,747	69,984	86,396	90,622
14	52,632	53,320	54,406	53,227	62,222	67,536	74,931	77,978
15	57,090	54,603	54,067	53,737	55,952	56,868	53,962	70,220
16	62,064	62,749	57,505	50,798	54,547	53,964	57,724	60,717
17	66,137	64,422	57,958	51,717	53,741	57,006	55,372	63,607
18	70,148	61,668	55,114	57,795	54,358	53,199	51,289	53,015

It is worth noting that for our version of the HC algorithm, there is a range of values of parameters X and R, for which close values of computational complexity are observed. These values of the X and R parameters are highlighted in Table 1. For each combination of these parameters, we have the smallest average number of iterations compared to the best result known to date from [31]. To compare the computational complexity of different methods for generating 8-bit bijective S-boxes with $N(S) = 104$ we present Table 2.

For our results, Table 2 also shows the confidence intervals calculated for a statistical significance level of $\alpha = 0.05$. Practically, this means that in 95% of cases the number of iterations required to generate S-boxes will be in the specified interval. As can be seen from the above results, our optimization of the HC algorithm compared to the best-known result allows us to reduce the average number of iterations by more than 20%.

In the last few years, many papers in the field of S-boxes generation have been published, e.g., [24–27,55]. In these papers, the authors estimate the average value of nonlinearity over all coordinate Boolean functions (1). However, the nonlinearity of S-boxes calculated by formula (2) is usually not high. For example, in [27], the minimum nonlinearity of the two generated S-boxes is 98 and 100. Two examples in [25] are also given, with minimum nonlinearity of 96 and 100. The paper [55] gives an estimate of the linear approximation probability of 0.125. This means that the maximum value in LAT calculated by Formula (4) is 32, whence, by Formula (5) we have a non-linearity of 96. In

our comparison in Table 2, we present only the results with minimum (for all component Boolean functions) nonlinearity, i.e., for $N(S) = 104$.

Table 2. Comparison of the computational complexity of generating 8-bit bijective S-boxes with $N(S) = 104$.

Literary Source	Generation Method	Cost Function, Parameters	Generation Complexity (Average Number of Iterations)
[19]	GaT	WHS, $X = 21$ and $R = 7$	3,239,000
[20]	GA	PCF, $N_p = 10$	741,371
	LSA	PCF, $N_p = 10$	172,280
	GaT	PCF, $N_p = 10$	167,451
[10]	LSA	WCF	149,539
	GaT	WCF	116,266
	HC	WCF	70,596
[35]	GaT	WHS, $X = 21$ and $R = 7$	3,849,881
	GA	PCF, $N_p = 10$	741,371
	LSA	PCF, $N_p = 10$	172,280
	GaT	PCF, $N_p = 10$	167,451
	LSA	WCF	89,460
	HC	WCF	65,933
Our work	HC algorithm (see Algorithm 2)	WHS, $X = 0$ and $R = 12$	$50{,}265 \pm 4007$
		WHS, $X = 4$ and $R = 11$	$51{,}278 \pm 5565$
		WHS, $X = 4$ and $R = 12$	$51{,}477 \pm 6009$
		WHS, $X = -16$ and $R = 16$	$50{,}798 \pm 4056$
		WHS, $X = -16$ and $R = 17$	$51{,}717 \pm 5062$
		WHS, $X = -28$ and $R = 18$	$51{,}289 \pm 4072$

Table 3 shows the results of generating target S-boxes with additionally introduced cryptographic indicators. In addition to the indicator $N(S)$, there are also the results of forming target S-boxes, that correspond to the additionally introduced criteria. All calculations were carried out with R =12 and $X = 0$. We performed 100 runs for each set of parameters.

Table 3 uses the following notation:

- $k_{\min}$ is the minimum number of iterations performed by the search algorithm out of all 100 runs, provided that the target S-box is found;
- $k_{\max}$ is the maximum number of iterations performed by the search algorithm out of all 100 runs, provided that the target S-box is found;
- k_{aver} is the average number of iterations performed by the search algorithm, provided that the target S-box is found;
- $k_{N=104}$ is the average number of iterations to find a bijective S-boxes with $N(S) = 104$;
- $k_{\text{add}} = k_{\text{aver}} - k_{N=104}$ is the average number of additionally performed iterations for the search algorithm after finding the first bijective S-box with $N(S) = 104$;
- k_{rej} is the average number of the rejected iterations with the additional criteria, i.e., after finding the bijective S-box with $N(S) = 104$, but this S-box does not correspond to the additional criteria.

As we can see from the results shown in Table 3, the probability of finding the target S-box with the required parameters is high. In our experiments, the probability of generating bijective S-boxes (with parameters $N(S) \geq 104$, $AI(S) = 3$, $\delta \leq 8$, $NnonLE(S) = 255$, and $NFP(S) = 0$) is ≈83%.

The most expensive (in terms of number of iterations) additional parameter is the condition for delta uniformity $\delta \leq 8$. This criteria increases the average required number of iterations to find the target S-box on average by $\approx 50\%$. Moreover, the scatter is quite

significant, from immediately found to several hundred thousand additional iterations (in this case, up to several hundred found bijective S-boxes with $N(S) = 104$) are rejected.

Table 3. Results of generating target S-boxes with additionally introduced cryptographic indicators.

Search Criteria	Number of Target S-Boxes	$k_{\min}$	$k_{\max}$	k_{aver}	$k_{N=104}$	k_{add}	k_{rej}
bijectivity, $N(S) \geq 104$, $AI(S) \geq 3$	99	14,916	123,800	52,936	52,928	8	0
bijectivity, $N(S) \geq 104$, $\delta \leq 8$	95	24,715	292,866	80,690	53,385	27,305	19
bijectivity, $N(S) \geq 104$, $NFP(S) = 0$	91 (100 not considering $NFP(S) = 0$)	10,771	103,924	50,864	–	–	–
bijectivity, $N(S) \geq 104$, $NnonLE(S) = 255$, $NFP(S) = 0$	87 (100 not considering $NFP(S) = 0$)	19,676	121,638	54,690	54,688	2	0
bijectivity, $N(S) \geq 104$, $\delta \leq 8$, $AI(S) \geq 3$, $NFP(S) = 0$	80 (94 not considering $NFP(S) = 0$)	26,333	230,516	81,380	51,589	29,791	12
bijectivity, $N(S) \geq 104$, $\delta \leq 8$, $AI(S) \geq 3$, $NnonLE(S) = 255$, $NFP(S) = 0$	83 (98 not considering $NFP(S) = 0$)	15,460	399,229	90,452	54,803	35,649	30

All found target S-boxes correspond to the criteria $AI(S) \geq 3$ and $NnonLE(S) = 255$. Thus, we can conclude that the introduction of these two additional parameters does not affect the performance of the suggested search algorithm.

As it was mentioned earlier, after finding the target S-box, we perform an affine transformation in order to fulfill the criteria $NFP(S) = 0$. About 90% of cases were successful in finding an affine transformation of the target S-box that does not contain fixed points.

It should be clarified that a small number of k_{add} with $k_{rej} = 0$ is formed due to the multi-threading of our software implementation. When calculating the parameters of the target S-box found by one thread, the other thread carries out some more (1–12, depending on the set of checked parameters) iterations, which are also counted in the total number. There are no such anomalies if you run the search algorithm in one thread.

As can be seen from the above results, the proposed optimization of the WHS heuristic allows using the HC algorithm (Algorithm 2) to generate target S-boxes with additionally introduced cryptographic indicators (bijectivity, $N(S) \geq 104$, $AI(S) \geq 3$, $\delta \leq 8$, $NnonLE(S) = 255$, and $NFP(S) = 0$) for 90 452 iterations, on average. The probability of finding a target S-box with only one run of the algorithm is greater than 80%.

6. Conclusions

In this paper, we have studied algorithms for heuristic search for nonlinear substitutions of symmetric ciphers. We proposed the integration of the HC algorithm for solving this task and we studied the cost function WHS based on Walsh–Hadamard Spectra. Our goal was to develop computationally efficient techniques for generating cryptographically strong S-boxes (bijective substitutions with high nonlinearity and algebraic immunity, low delta uniformity, and linear redundancy).

Our research consisted of optimizing the WHS objective function (heuristic) in the context of the local optimization algorithm HC. We managed to reduce the computational complexity of generating the S-box with $N(S) = 104$ more than 20% compared to the best known result from [35]. Most interesting in this result is the combination of search

algorithm (HC) and heuristic function (WHS). These are well-known and thoroughly studied mathematical objects [10,18–21,35]. Until now, it was believed that the optimal parameters for WHS were $X = 21$, $R = 7$. These values were obtained while selecting the parameters during the optimization of the GaT algorithm in [19]. However, as it turned out, these parameters are far from optimal when using the HC algorithm (see Algorithm 2). In our experiments, the parameters ($X = 0$, $R = 12$) are optimal. In other words, we believe that the optimization of each cost function should be carried out separately for each mathematical optimization algorithm; our results clearly confirm it. This opens up significant prospects for the study of various heuristics in combination with the most efficient mathematical optimization algorithms.

The second important result of this work is that we were able to quickly generate target S-boxes with additionally introduced cryptographic indicators. In addition to the nonlinearity ($N(S) \geq 104$), we introduced additional restrictions to algebraic immunity ($AI(S) \geq 3$), delta uniformity ($\delta \leq 8$), linear non-redundancy ($NnonLE(S) = 255$), and the absence of fixed points ($NFP(S) = 0$). The algorithm HC (see Algorithm 2) and the heuristic function WHS successfully coped with the task of generating such nonlinear substitutions [38,39]. These results are relevant because most modern symmetric cryptographic algorithms use such substitutions. We give some examples of the formed S-box in Appendix A.

Our software implementations for generating target S-boxes and reproducing all results are available in the public domain on GitHub repository [56]. We hope this will facilitate the reproducibility and independent verification of our results.

Author Contributions: Conceptualization and methodology, A.K.; formal analysis, investigation, E.F.; resources, L.R.; data curation, N.P.; software and validation, S.K.; writing—review and editing, K.K.; validation, project administration, funding acquisition, E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A Examples of Bijective S-Boxes in Hexadecimal Notation

(With parameters $N(S) = 104$, $AI(S) = 3$, $\delta = 8$, $NnonLE(S) = 255$, $NFP(S) = 0$)

```
S-box1 =
{E5, D3, 17, 74, EA, 0C, 33, 3B, 79, 56, EC, 7B, E3, 51, 59, A2,
16, E0, 76, D8, CD, 6F, E4, DD, 7F, E2, 9F, 73, 67, 01, 61, 5E,
FE, 91, B2, 58, 05, BD, 86, D1, DB, FC, 92, C2, AE, 42, C8, 1A,
46, D0, B3, 57, 2B, AD, 21, C0, B6, 45, 09, A5, AC, 28, 22, 39,
88, 71, 1D, 5D, 0A, 4C, 53, 90, A0, E9, 8B, 8A, 31, 20, 8C, 8D,
26, 35, 06, 27, 49, 13, F6, 40, C3, 70, 68, 9B, 60, C7, CE, 0B,
E1, 64, 77, A6, 52, 98, 4D, 18, 47, 0F, D5, 8E, 94, 5A, 4E, 32,
38, F3, 3C, 5C, 2D, ED, A3, DC, CC, F7, 03, 4F, 54, 8F, 07, 65,
04, 97, 63, 41, EF, 15, 85, 4B, 96, 78, 7A, 75, 00, D2, 10, 48,
95, DF, 3D, F1, F5, 0E, F9, 34, F0, BE, 14, C9, 9E, BF, CB, 5B,
A1, D9, FB, FD, AB, 5F, 80, 6A, 62, 36, B9, A4, C6, B1, 87, 12,
99, 72, 7C, C4, 7E, BA, DE, 19, EE, 0D, AF, F2, D6, E7, D7, 3E,
50, 69, CA, 6E, BC, B8, 93, 4A, A7, 1C, 37, 89, 7D, 6D, 2C, 25,
9D, B7, 3A, 9C, 6C, 29, 08, F4, FF, 30, 2A, A9, 43, 84, CF, FA,
3F, 1B, EB, AA, 82, 24, 81, 44, 23, 2F, C1, B5, 66, 83, DA, D4,
BB, 1E, 1F, B4, E8, 55, 9A, 02, B0, E6, A8, 2E, C5, 11, 6B, F8};
```

S-box2 =
{86, AA, 3C, 6A, 66, A3, C3, 4A, BA, C1, 5E, EC, B4, 4F, 92, A7,
A1, 5D, D4, 90, 54, D7, C6, 06, 85, CD, C9, 64, 20, A4, 15, 1B,
DB, 03, 17, 4E, BD, DF, 23, 19, 8B, 91, 7F, 39, B5, 24, 82, A2,
AC, F1, 49, 41, 63, B1, 33, F6, 4B, E9, 1D, AB, 0E, 01, 60, D3,
CF, C5, 3B, 8E, E4, 6F, 9E, 08, 47, C8, FC, F0, FD, 21, 68, 52,
43, 35, 6D, DE, 1A, E7, B7, D5, 9F, 36, 84, 62, D6, 46, B6, 8F,
22, 2C, 69, F5, B0, 00, 8A, 34, 5A, 75, 72, 81, BC, 5C, 59, F9,
D1, 79, ED, EE, 1F, 77, 38, 7C, F2, 83, 70, C0, 14, D9, 87, C4,
0C, 04, 71, 1E, 11, 16, 61, 2B, 80, 56, A9, 30, C7, 42, 96, B2,
3E, DD, AE, 1C, 8D, 73, E2, F4, 13, 3F, A0, EF, F7, 94, 0B, FE,
89, 55, 7D, 58, E6, 48, 9D, 98, 97, F8, 2F, 37, B8, 32, 05, 3D,
BF, 50, E0, AF, 76, 0F, E1, 2D, DC, 9C, 51, FF, 02, 7E, 10, FB,
CC, 9B, CB, 6C, 26, 65, 5F, 53, D2, 4C, 8C, 45, 12, 07, 3A, 27,
B9, 88, A6, 2A, 25, 0D, 7A, D0, 99, 6E, C2, CE, 6B, 9A, 5B, EB,
CA, FA, A5, B3, 28, 29, BB, 95, 78, 7B, 67, EA, 31, E8, 40, 18,
74, 44, 2E, 09, DA, E3, D8, E5, BE, 57, F3, AD, 0A, 93, 4D, A8}.
S-box3 =
{20, A2, 88, D8, 0D, B7, 74, CD, 8E, 3F, C1, B9, 12, 38, 70, 7B,
78, D7, 07, 4F, AF, 02, 2E, 29, 44, 52, AC, F9, 53, 40, 31, DF,
8A, 36, 43, E2, 95, B3, 25, 30, 65, 0C, 69, 45, 71, 24, B4, 3B,
41, E4, 5B, C5, 1A, D6, 42, B6, AE, 3C, EA, 49, 75, BF, F6, 67,
C9, FE, A0, 35, 51, DD, D3, 60, 80, 10, 0E, 2D, 61, EB, F1, 2A,
8C, FA, 7A, FB, 0F, 11, EE, CA, 8B, FF, 0B, 97, B2, 91, 04, 83,
DC, B5, BD, 87, 1C, F7, 1D, 46, CB, AA, 7F, 5D, E6, 89, 68, EF,
F0, D1, 3E, F3, 57, 9E, DE, E5, C8, 3D, 5E, 76, 08, F5, 5F, A1,
27, FC, E1, 63, FD, 82, 85, 34, 58, C2, CE, B1, 9B, 99, A3, 00,
09, C6, D0, 73, 37, 0A, D4, 98, A5, BB, 2F, C4, 4D, 4A, 32, 7E,
BA, 7C, 5A, 66, 5C, E8, E3, C7, AB, 13, 79, 6D, 47, 7D, 8D, 1B,
21, 6E, 1E, 6F, 06, 05, 64, 39, 90, 14, 6B, 4B, 22, 2B, F2, 16,
19, 03, 86, 6C, 18, 4E, E9, DA, F4, 9C, B8, A8, 77, E0, 56, AD,
DB, 96, 8F, 48, C3, A9, 84, CC, 55, 1F, A6, ED, D5, B0, 50, BC,
9A, F8, BE, EC, 54, 59, CF, 17, 15, D2, 33, E7, 01, 6A, 3A, 26,
62, 9F, 92, 72, 2C, C0, A7, D9, 28, 9D, 4C, 81, 94, 93, 23, A4};
S-box4 =
{7B, 88, E6, E7, B5, 6A, FE, 73, 30, 36, 9A, B9, A3, 43, 7D, E5,
AF, C2, 3C, 4F, 0F, 21, A6, 40, C0, 4D, 9B, CB, 2A, F2, BD, CF,
48, 97, 49, AA, 10, 27, CA, 74, E4, 2D, 1B, 92, B3, 69, BA, 28,
EF, A9, 8F, BF, 8C, 02, A4, 7E, 66, 62, 29, 93, 80, FD, 98, E1,
5A, DA, DF, 6C, 53, 44, 8D, 24, 7C, 68, B8, B0, 57, 09, FF, 39,
07, 00, BE, 6B, 1C, 4C, 13, A0, B2, EB, 2B, F8, D4, D9, A8, 8A,
3B, 55, 6D, A1, 4A, 84, E9, 95, 46, 91, 08, 26, 9C, 77, 81, DB,
5E, 4B, 0E, 96, CD, 20, 8E, 17, 9F, AC, AD, 06, 99, D2, 5F, 89,
D5, C7, 1A, 3A, 12, D3, C8, F4, 61, 15, F5, BB, D6, 23, 33, F1,
0D, C5, 22, AE, 1E, 31, 71, A7, B4, 52, 03, A2, 34, 37, 2F, A5,
EE, 58, CC, 63, DD, CE, D1, 42, 54, 11, C3, 2E, 1F, 90, E2, 79,
5D, C4, D8, FC, 5C, 3F, EA, 6E, E0, 47, 75, 04, 86, C9, 9E, 72,
45, 3D, F6, 65, 05, B1, 85, C1, F9, 9D, DC, E3, 16, 67, C6, F3,
FB, B6, 35, 56, 4E, 51, 87, 59, ED, 64, 7F, 38, D0, AB, F7, 25,
41, 1D, 76, 60, 5B, BC, F0, 19, FA, 94, 14, EC, 2C, D7, 0B, DE,
8B, 18, 0C, 82, 50, 32, 83, 70, 3E, 6F, 7A, 01, E8, 0A, B7, 78}.

References

1. Shannon, C.E. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
2. Daemen, J.; Rijmen, V. Specification of Rijndael. In *The Design of Rijndael: The Advanced Encryption Standard (AES)*; Daemen, J., Rijmen, V., Eds.; Information Security and Cryptography; Springer: Berlin/Heidelberg, Germany, 2020; pp. 31–51. ISBN 978-3-662-60769-5.
3. Bard, G.V. *Algebraic Cryptanalysis*; Springer: Boston, MA, USA, 2009. ISBN 978-0-387-88756-2.
4. Courtois, N.T.; Bard, G.V. Algebraic Cryptanalysis of the Data Encryption Standard. In *Cryptography and Coding, Proceedings of the 11th IMA International Conference, Cirencester, UK, 18–20 December 2007*; Galbraith, S.D., Ed.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 152–169.
5. Courtois, N.T.; Pieprzyk, J. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In *Advances in Cryptology—ASIACRYPT 2002, Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, 1–5 December 2002*; Zheng, Y., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 267–287.
6. Clark, A.J. Optimisation Heuristics for Cryptology. Ph.D. Thesis, Queensland University of Technology, Brisbane City, Australia, 1998.
7. Millan, W. How to Improve the Nonlinearity of Bijective S-Boxes. In *Information Security and Privacy, Proceedings of the Third Australasian Conference, ACISP'98, Brisbane, Australia, 13–15 July 1998*; Boyd, C., Dawson, E., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 181–192.
8. Millan, W.; Clark, A.; Dawson, E. Boolean Function Design Using Hill Climbing Methods. In *Information Security and Privacy, Proceedings of the 4th Australasian Conference, ACISP'99, Wollongong, Australia, 7–9 April 1999*; Pieprzyk, J., Safavi-Naini, R., Seberry, J., Eds.; Springer: Berlin/Heidelberg, Germany, 1999; pp. 1–11.
9. Álvarez-Cubero, J. Vector Boolean Functions: Applications in Symmetric Cryptography. Ph.D. Thesis, Universidad Politécnica de Madrid, Madrid, Spain, 2015.
10. Freyre-Echevarría, A.; Alanezi, A.; Martínez-Díaz, I.; Ahmad, M.; Abd El-Latif, A.A.; Kolivand, H.; Razaq, A. An External Parameter Independent Novel Cost Function for Evolving Bijective Substitution-Boxes. *Symmetry* **2020**, *12*, 1896. [CrossRef]
11. Cusick, T.; Stănică, P. *Cryptographic Boolean Functions and Applications*, 2nd ed.; Academic Press: Cambridge, MA, USA, 2017; p. 275.
12. Hernando, L.; Mendiburu, A.; Lozano, J.A. Hill-Climbing Algorithm: Let's Go for a Walk Before Finding the Optimum. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7.
13. Huang, B.; Zhou, M. Informed Heuristic Search in Reachability Graph. In *Supervisory Control and Scheduling of Resource Allocation Systems: Reachability Graph Perspective*; IEEE: Piscataway, NJ, USA, 2020; pp. 137–156. ISBN 978-1-119-61969-7.
14. Peker, F.; Altun, M. A Fast Hill Climbing Algorithm for Defect and Variation Tolerant Logic Mapping of Nano-Crossbar Arrays. *IEEE Trans. Multi-Scale Comput. Syst.* **2018**, *4*, 522–532. [CrossRef]
15. Karabacak, M.; Fernández-Ramírez, L.M.; Kamal, T.; Kamal, S. A New Hill Climbing Maximum Power Tracking Control for Wind Turbines With Inertial Effect Compensation. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8545–8556. [CrossRef]
16. Dimitrov, M.; Baitcheva, T.; Nikolov, N. Efficient Generation of Low Autocorrelation Binary Sequences. *IEEE Signal Process. Lett.* **2020**, *27*, 341–345. [CrossRef]
17. Ghosh, K.K.; Ahmed, S.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Improved Binary Sailfish Optimizer Based on Adaptive β-Hill Climbing for Feature Selection. *IEEE Access* **2020**, *8*, 83548–83560. [CrossRef]
18. Clark, J.A.; Jacob, J.L.; Stepney, S. The Design of S-Boxes by Simulated Annealing. *New Gener. Comput.* **2005**, *23*, 219–231. [CrossRef]
19. Tesar, P. A New Method for Generating High Non-Linearity S-Boxes. *Radioengineering* **2010**, *19*, 23–26.
20. Picek, S.; Cupic, M.; Rotim, L. A New Cost Function for Evolution of S-Boxes. *Evol. Comput.* **2016**, *24*, 695–718. [CrossRef]
21. 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. [CrossRef]
22. Prathiba, A.; Kanchana Bhaaskaran, V.S. Hardware Footprints of S-Box in Lightweight Symmetric Block Ciphers for IoT and CPS Information Security Systems. *Integration* **2019**, *69*, 266–278. [CrossRef]
23. Wen, H.; Wu, J.; Ma, L.; Liu, Z.; Lin, Y.; Zhou, L.; Jian, H.; Lin, W.; Liu, L.; Zheng, T.; et al. Secure Optical Image Communication Using Double Random Transformation and Memristive Chaos. *IEEE Photonics J.* **2023**, *15*, 1–11. [CrossRef]
24. Zamli, K.Z.; Alhadawi, H.S.; Din, F. Utilizing the Roulette Wheel Based Social Network Search Algorithm for Substitution Box Construction and Optimization. *Neural Comput. Appl.* **2023**, *35*, 4051–4071. [CrossRef]
25. Lawah, A.I.; Ibrahim, A.A.; Salih, S.Q.; Alhadawi, H.S.; JosephNg, P.S. Grey Wolf Optimizer and Discrete Chaotic Map for Substitution Boxes Design and Optimization. *IEEE Access* **2023**, *11*, 42416–42430. [CrossRef]
26. Alhadawi, H.S.; Salih, S.Q.; Salman, Y.D. Chaotic Particle Swarm Optimization Based on Meeting Room Approach for Designing Bijective S-Boxes. In Proceedings of the International Conference on Emerging Technologies and Intelligent Systems; Al-Emran, M., Al-Sharafi, M.A., Al-Kabi, M.N., Shaalan, K., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 331–341.
27. Zamli, K.Z. Optimizing S-Box Generation Based on the Adaptive Agent Heroes and Cowards Algorithm. *Expert Syst. Appl.* **2021**, *182*, 115305. [CrossRef]

28. Norvig, P.; Russell, S. *Artificial Intelligence: A Modern Approach, Global Edition*, 4th ed.; Pearson: Harlow, UK, 2021. ISBN 978-1-292-40113-3.
29. Miao, H.; Chen, G.; Li, C.; Dong, Z.Y.; Wong, K.P. Operating Expense Optimization for EVs in Multiple Depots and Charge Stations Environment Using Evolutionary Heuristic Method. *IEEE Trans. Smart Grid* **2018**, *9*, 6599–6611. [CrossRef]
30. Li, Z.; Tam, V.; Yeung, L.K. An Adaptive Multi-Population Optimization Algorithm for Global Continuous Optimization. *IEEE Access* **2021**, *9*, 19960–19989. [CrossRef]
31. Battiti, R.; Brunato, M.; Mascia, F. *Reactive Search and Intelligent Optimization*; Operations Research/Computer Science Interfaces Series; Springer: Boston, MA, USA, 2009; Volume 45. ISBN 978-0-387-09623-0.
32. Huang, B.; Zhou, M. Controllable Heuristic Search. In *Supervisory Control and Scheduling of Resource Allocation Systems: Reachability Graph Perspective*; IEEE: Piscataway, NJ, USA, 2020; pp. 157–179. ISBN 978-1-119-61969-7.
33. Souravlias, D.; Parsopoulos, K.E.; Meletiou, G.C. Designing Bijective S-Boxes Using Algorithm Portfolios with Limited Time Budgets. *Appl. Soft Comput.* **2017**, *59*, 475–486. [CrossRef]
34. Ivanov, G.; Nikolov, N.; Nikova, S. Cryptographically Strong S-Boxes Generated by Modified Immune Algorithm. In Proceedings of the Cryptography and Information Security in the Balkans; Pasalic, E., Knudsen, L.R., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 31–42.
35. Freyre Echevarría, A.; Martínez Díaz, I. A New Cost Function to Improve Nonlinearity of Bijective S-Boxes. 2020. Available online: https://www.researchgate.net/publication/343699912_A_new_cost_function_to_improve_nonlinearity_of_bijective_S-boxes (accessed on 17 May 2023).
36. Kuznetsov, A.; Serhiienko, R.; Prokopovych-Tkachenko, D.; Tarasenko, Y. Evaluation of Algebraic Immunity of Modern Block Ciphers. In Proceedings of the 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2–27 May 2018; pp. 288–293.
37. Rodinko, M.; Oliynykov, R.; Gorbenko, Y. Optimization of the High Nonlinear S-Boxes Generation Method. *Tatra Mt. Math. Publ.* **2017**, *70*, 93–105. [CrossRef]
38. Kuznetsov, A.A.; Potii, O.V.; Poluyanenko, N.A.; Gorbenko, Y.I.; Kryvinska, N. *Stream Ciphers in Modern Real-Time IT Systems*; Studies in Systems, Decision and Control; Springer Nature: Cham, Switzerland, 2022. ISBN 978-3-030-79770-6.
39. Oliynykov, R.; Gorbenko, I.; Kazymyrov, O.; Ruzhentsev, V.; Kuznetsov, O.; Gorbenko, Y.; Dyrda, O.; Dolgov, V.; Pushkaryov, A.; Mordvinov, R.; et al. A New Encryption Standard of Ukraine: The Kalyna Block Cipher. 2015. Available online: https://eprint.iacr.org/2015/650 (accessed on 10 May 2022).
40. Carlet, C. Vectorial Boolean Functions for Cryptography. In *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*; Cambridge University Press: Cambridge, UK, 2006.
41. Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 2018. ISBN 978-0-429-46633-5.
42. Nyberg, K. Linear Approximation of Block Ciphers. In Proceedings of the EUROCRYPT, Perugia, Italy, 9–12 May 1994.
43. Matsui, M. Linear Cryptanalysis Method for DES Cipher. In Proceedings of the Advances in Cryptology—EUROCRYPT '93, Lofthus, Norway, 23–27 May 1993; Helleseth, T., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 386–397.
44. Ars, G.; Faugère, J.-C. Algebraic Immunities of Functions over Finite Fields; INRIA. 2005, p. 17. Available online: https://inria.hal.science/inria-00070475 (accessed on 17 May 2023).
45. Biryukov, A.; De Cannière, C. Block Ciphers and Systems of Quadratic Equations. In Proceedings of the Fast Software Encryption, Lund, Sweden, 24–26 February 2003; Johansson, T., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 274–289.
46. Kuznetsov, O.O.; Gorbenko, Y.I.; Bilozertsev, I.M.; Andrushkevych, A.V.; Narizhnyi, O.P. Algebraic immunity of non-linear blocks of symmetric ciphers. *Telecommun. Radio Eng.* **2018**, *77*, 309–325. [CrossRef]
47. Nyberg, K. Differentially Uniform Mappings for Cryptography. In Proceedings of the Advances in Cryptology—EUROCRYPT '93, Lofthus, Norway, 23–27 May 1993; Helleseth, T., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 55–64.
48. Biham, E.; Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72. [CrossRef]
49. Fuller, J.E. Analysis of Affine Equivalent Boolean Functions for Cryptography. Ph.D. Thesis, Queensland University of Technology, Brisbane City, Australia, 2003.
50. Fuller, J.; Millan, W. Linear Redundancy in S-Boxes. In *Fast Software Encryption*; Johansson, T., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2887, pp. 74–86. ISBN 978-3-540-20449-7.
51. Campbell, K.W.; Wiener, M.J. DES Is Not a Group. In Proceedings of the Advances in Cryptology—CRYPTO' 92, Santa Barbara, CA, USA, 16–20 August 1992; Brickell, E.F., Ed.; Springer: Berlin/Heidelberg, Germany, 1993; pp. 512–520.
52. Özbudak, F.; Yayla, O. On Verification of Restricted Extended Affine Equivalence of Vectorial Boolean Functions. In *Arithmetic of Finite Fields, Proceedings of the 5th International Workshop, WAIFI 2014, Gebze, Turkey, 27–28 September 2014*; Springer: Cham, Switzerland, 2015. [CrossRef]
53. Eastlake, D., 3rd; Schiller, J.; Crocker, S. Randomness Requirements for Security. 2005. Available online: https://www.rfc-editor.org/info/rfc4086 (accessed on 25 July 2020).
54. Knuth, D. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed.; Addison-Wesley Professional: Reading, MA, USA, 1997. ISBN 978-0-201-89684-8.

55. Liu, X.; Tong, X.; Wang, Z.; Zhang, M. Efficient High Nonlinearity S-Box Generating Algorithm Based on Third-Order Nonlinear Digital Filter. *Chaos Solitons Fractals* **2021**, *150*, 111109. [CrossRef]
56. KandiyIIT WHS Hill Climbing S-Box Generator 2022. Available online: https://github.com/KandiyIIT/Hill-Climbing-S-Box-Generator (accessed on 12 May 2022).