A Framework for Fine-Grained Nonlinearity Optimization of Boolean and Vectorial Boolean Functions

MIROSLAV M. DIMITROV, (Graduate Student Member, IEEE)
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia 1113, Bulgaria
e-mail: mirdim@math.bas.bg

This work was supported by the Bulgarian National Science Fund under Grant DH 12/8, 15.12.2017.

ABSTRACT Boolean functions and vectorial Boolean functions (S-boxes) are widely used cryptographic primitives for achieving cryptanalytic resistance of modern block or stream ciphers. In the aspect of information security, one of the most desirable characteristics a given S-box should possess is a high nonlinearity. In this paper, we project the nonlinearity optimization problem to the domain of binary integer programming. Then, we demonstrate how this interconnection could be successfully exploited by SAT solvers. The provided toolbox could serve in cases, where the designer’s goal is to increase (or intentionally decrease) the nonlinearity of a given S-box by applying as minimum changes as possible. For example, we demonstrate how the Skipjack S-box, developed by the U.S. National Security Agency (NSA), and the Kuznyechik S-box, developed by the Russian Federation’s standardization agency, could be optimized to a higher nonlinearity by tweaking, respectively, just 4 and 12 bits (out of 2048). In the end, we show that bijective (8,8) S-boxes, the eight coordinates of which possess the currently known optimal nonlinearity value of 116, do exist.

INDEX TERMS Boolean functions, integer linear programming, cryptography, optimization, S-box, nonlinearity.

I. INTRODUCTION
S-boxes are the main building blocks of the nonlinearity layer of modern block or stream ciphers, like AES [1], SNOW [2], TWOFISH [3], WHIRLPOOL [4], PICARO [5], PRINCE [6]. A weak S-box, in a cryptographic perspective, can be exploited by various attacks such as linear cryptanalysis [7], differential cryptanalysis [8], boomerang attack [9], algebraic attacks [10], and others [11]. One of the most desirable characteristics a given S-box should possess is a high nonlinearity value. An S-box with optimal or near-optimal nonlinearity value can be constructed by using the finite field inversion method as shown in [12]. However, such an S-box would be closely related to an algebraic structure. Thus, as a proactive countermeasure to possible algebraic attacks, new ways of generation or optimization of pseudo-randomly generated S-boxes are proposed. The published methods are manifold. However, heuristic optimization of a given S-box is a resource-consuming task. In general, the size of the search space of bijective S-boxes with dimension \((n, n)\) is \(2^n\). For dimension \((8, 8)\), which is a common choice among the majority of popular S-boxes, the size of the search space is approximately \(2^{1684}\). Furthermore, during the search for a better S-box, each candidate is repeatedly evaluated by some complex cost function. For example, considering state-of-the-art optimization algorithms [13]–[17], the Walsh-Hadamard transform matrix elements are regularly collected during each iteration. However, such calculations add significant overhead to the optimization routine.

In this work, we find an interconnection between the S-box nonlinearity optimization problem and binary integer programming. Thus, we create a lightweight optimization routine, which does not cause any significant computational burden. For example, considering state-of-the-art optimization algorithms [13]–[17], the Walsh-Hadamard transform matrix elements are regularly collected during each iteration. However, such calculations add significant overhead to the optimization routine.

In this work, we find an interconnection between the S-box nonlinearity optimization problem and binary integer programming. Thus, we create a lightweight optimization routine, which does not cause any significant computational burden. Moreover, the proposed toolbox could be used as a proof of infeasibility.

Another major drawback of the state-of-the-art heuristic techniques is their aggressiveness on the initial S-box. Hence, in most cases, it is difficult to link the resulted S-box with the...
initial S-box. In fact, it is difficult to prove that such a link exists in the first place. However, in this work, we present a fine-grained optimization routine, which allow us to optimize the nonlinearity value of a given S-box with minimum changes as possible. From the designer perspective, this property is particularly beneficial, since we could focus the optimization routine on the weak components of a given S-box, without degrading the remaining ones. We demonstrate the effectiveness of the proposed algorithm by increasing the nonlinearity of the Skipjack S-box, developed by NSA, and Kuznyechik S-box, developed by Russian Federation’s standardization agency, by tweaking respectively 4 and 12 (out of 2048) bits only.

During our research, we collected more than 300 papers, which emphasize on the average coordinate nonlinearity value, or ACNV, of bijective S-boxes, instead of the nonlinearity value of the weakest component. In our previous work [18], we show that such constructions should be used with considerable caution. The currently known maximum nonlinearity value for 8-variable balanced Boolean functions is 116 [19]. Furthermore, as shown in [20], the nonlinearity value of 8-variable balanced Boolean functions is upper bounded by 120, which means that the maximum theoretical ACNV of (8,8) bijective S-boxes is less or equal to 118.0. If a bijective S-box with ACNV greater than 116 is found, at least one of its eight coordinates will possess a nonlinearity value of 118, which will finally give an answer to the long-standing problem of the maximum possible nonlinearity value for 8-variable balanced Boolean functions. However, there is an academic skepticism that 8-variable balanced Boolean functions with nonlinearity value 118 really exist. Having this in mind, one open question to be answered is:

- Does bijective (8,8) S-box with an ACNV value of 116 really exist?

In this paper, by using SAT solving techniques, we show that bijective (8,8) S-boxes with ACNV value of 116 really exist. However, despite our attempts, we were not able to find an 8-variable balanced Boolean function with a nonlinearity of 118.

II. PRELIMINARIES

In this section, we present some standard definitions and S-box characteristics. More detailed overview could be found, for example, in [21] and [22].

**Definition 1 (Boolean Function & Truth Tables):** Let us define the set $B = \{0, 1\}$. A Boolean function $f(x)$ of $n$ variables $x_1, \ldots, x_n$ is a mapping $f : B^n \mapsto B$ from $n$ binary inputs $x = (x_1, x_2, \ldots, x_n) \in B^n$ to one binary output $y = f(x) \in B$. The binary truth table (BTT) of an $n$-variable Boolean function $f(x)$ is the vector of all the consecutive outputs of the Boolean function:

$$[f(x)] = [f(0, 0, \ldots, 0), f(0, 0, \ldots, 1), \ldots, f(1, 1, \ldots, 1)]$$

The polarity truth table (PTT) of an $n$-variable Boolean function $f(x)$ is derived from the binary truth table. We define the PTT by $\hat{f}(x) = [1 - 2f(x)]$. By the definition of the polarity truth table, it follows that:

$$f(x) = 0 \Leftrightarrow \hat{f}(x) = 1; f(x) = 1 \Leftrightarrow \hat{f}(x) = -1$$

**Definition 2 (Algebraic Normal Form – ANF):** The algebraic normal form of an $n$-variable Boolean function $f(x)$, denoted by ANF$_f$, is given by the following equation:

$$ANF_f = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_{2^n} \oplus a_1 x_1 x_2 \oplus \cdots \oplus a_{2^n} x_{\bar{1}} x_{\bar{2}} \cdots x_m$$

where the coefficients $a$ belongs to $B$.

**Definition 3 (Hamming Distance):** The Hamming distance between two $n$-variable Boolean functions $f(x)$ and $g(x)$, denoted by $d_H(f, g)$, represents the number of differing elements in the corresponding positions of their truth tables.

**Definition 4 (Linear Boolean Function):** Any $n$-variable Boolean function of the form:

$$l_w(x) = \langle w, x \rangle = w_1 x_1 \oplus w_2 x_2 \oplus \cdots \oplus w_n x_n,$$

where $w, x \in B^n$, is called a linear function.

**Definition 5 (Affine Boolean Function):** Any $n$-variable Boolean function of the form:

$$l_w(x) = \langle w, x \rangle = w_0 \oplus w_1 x_1 \oplus w_2 x_2 \oplus \cdots \oplus w_n x_n,$$

where $w_0 \in B$ and $w, x \in B^n$, is called an affine function.

**Definition 6 (Walsh-Hadamard Transform – WHT):** For an $n$-variable Boolean function $f(x)$, represented by its polarity table $[\hat{f}(x)]$, the Walsh-Hadamard transform, or WHT, $\hat{F}_f : B^n \mapsto Z$, is defined by:

$$\hat{F}_f(w) = \sum_{x \in B^n} \hat{f}(x)(-1)^{<w, x>},$$

where $w \in B^n$.

**Definition 7 Vectorial Boolean Function – Substitution Table – S-Box: An $n$-binary input to $m$-binary output mapping $S : B^n \mapsto B^m$, which assigns some $y = (y_1, y_2, \ldots, y_m) \in B^m$ by $S(x) = y$ to each $x = (x_1, x_2, \ldots, x_n) \in B^n$, is called an ($n, m$) substitution table (S-box) and is denoted by $S(n,m)$.**

**Definition 8 (Bijective S-box):** An S-box $S(n,m)$ is said to be bijective, if it maps each input $x \in B^n$ to a distinct output $y = S(x) \in B^m$ and all possible $2^m$ outputs are present.

**Definition 9 (S-box Look-up Table – LUT):** The look-up table LUT of an S-box $S(n,m)$ is an $(2^n \times m)$ binary matrix $S$, which rows consist of all outputs of $S(n,m)$, corresponding to all possible $2^n$ inputs ordered lexicographically:

$$S_{LUT} = \begin{bmatrix}
    f_1(0,0, \ldots, 0) & f_2(0,0, \ldots, 0) & \cdots & f_m(0,0, \ldots, 0) \\
    f_1(0,0, \ldots, 1) & f_2(0,0, \ldots, 1) & \cdots & f_m(0,0, \ldots, 1) \\
    \vdots & \vdots & \ddots & \vdots \\
    f_1(1,1, \ldots, 0) & f_2(1,1, \ldots, 0) & \cdots & f_m(1,1, \ldots, 0) \\
    f_1(1,1, \ldots, 1) & f_2(1,1, \ldots, 1) & \cdots & f_m(1,1, \ldots, 1)
\end{bmatrix}$$

**Definition 10 (S-box Coordinates):** We define each column of the $S(n,m)$ LUT as a coordinate of $S(n,m)$. In fact, each column represents the truth table of some function $f_i$ (we are using the notation introduced in the previous definition). If $S(n,m)$ is bijective vectorial Boolean function it follows that $n = m$ and we have exactly $n$ coordinates.
Definition 11 (Polarity Look-up Table – PLUT): The polarity look-up table PLUT of an S-box $S(n, m)$, denoted by $\hat{S}$, is an $(2^n, m)$ matrix with elements in $\{−1, 1\}$, where each element $\hat{S}[j][k]$, for $j = 1, 2, \ldots, 2^n$ and $k = 1, 2, \ldots, m$, is derived from $S[j][k]$ by

$$\hat{S}[j][k] = (-1)^{S[j][k]} = 1 - 2S[j][k]$$

Definition 12 S-Box Extended WHT Spectrum Matrix – EWHTSM: The extended Walsh-Hadamard transform spectrum (EWHTSM) of an S-box $S(n, m)$ is a $(2^n, 2^m)$ matrix $\hat{F}_{Ext}$, which are represented by the Walsh-Hadamard transform spectra $\hat{F}_E(v)$ of the Boolean functions

$$g_v(x) = v f_1(x) \oplus v f_2(x) \oplus \cdots \oplus v f_m(x),$$

where $w$ and $v$ are arranged lexicographically in $B^n$ and respectively $B^m$.

$$\hat{F}_{Ext} = \begin{bmatrix} \hat{F}_{E_0}(0, 0, \ldots, 0) & \cdots & \hat{F}_{E_{2^m-1}}(0, 0, \ldots, 0) \\ \hat{F}_{E_0}(0, 1, \ldots, 1) & \cdots & \hat{F}_{E_{2^m-1}}(0, 1, \ldots, 1) \\ \vdots & \ddots & \vdots \\ \hat{F}_{E_0}(1, 1, \ldots, 0) & \cdots & \hat{F}_{E_{2^m-1}}(1, 1, \ldots, 0) \\ \hat{F}_{E_0}(1, 1, \ldots, 1) & \cdots & \hat{F}_{E_{2^m-1}}(1, 1, \ldots, 1) \end{bmatrix}$$

The idea behind S-box extended Walsh-Hadamard transform matrix is to quantitatively describe the distance with special measure, alike the Hamming distance, between each linear combination of coordinates in the S-box and each possible linear function.

Definition 13 (Linear Approximation Table – LAT): The linear approximation table of an S-box $S(n, m)$, denoted by LAT$_S$ or SLAT, is a table with $2^n$ rows and $2^m$ columns, which entries are given by:

$$S_{LAT}[X][Y] = LAT_S[X][Y] = 2^{n-1} - d_H(X,Y),$$

where $Y$ is a consequent linear combination of coordinates of the current S-box and $X$ is the consequent linear function with length $n$.

Definition 14 (S-box Nonlinearity): The nonlinearity of an S-box $S(n, m)$, denoted by $S_{NL}$, is defined as:

$$S_{NL} = 2^{n-1} - \max(|\text{abs}(w_i)|),$$

where $w_i$ is the set of all elements in LAT, except the uppermost left one.

Definition 15 (S-box ACNV): The average coordinate nonlinearity value, or ACNV, of a given S-box $S$, is the average value of all nonlinearities of coordinates of $S$.

Definition 16 (S-box Decimal Look-up Table – DLUT): Each S-box is uniquely defined by its LUT. Translating each row of the LUT as a decimal number uniquely defines the same S-box as a decimal look-up table (DLUT).

A bijective S-box $S(n, n)$ has exactly $n$ coordinates. Each coordinate is defined by its truth table, which consists of $2^n$ elements from $B$. So, we have a total of $n2^n$ elements from $B$ which uniquely define $S$. Following this observation, we have a total of $2^{n2^n}$ possible choices of S-boxes. But not all of them are bijective. To further restrict our choices to bijective S-boxes only, we need to restrict the set of possible S-boxes with the following observation – all elements in the DLUT of $S$ should be distinct. This reduces the possible choices of S-boxes from $2^{n2^n}$ to $2^n$.

III. DESCENION THROUGH COUPLINGS

In this section, we introduce the concept of couplings, coordinate decomposition, degree of descendibility, S-box coordinate extended linear approximation table (CELAT), as well as some useful properties and inner relationships. For convenience, let us denote as $f(n)$ the integer extracted from $n$, by flipping its $i$-th bit of its binary representation. Obviously, $f(f(n))$ = $n$.

Lemma 1 (The Parity Lemma): Tweaking a bijective S-box $S$ by flipping just one bit in its corresponding Look-up Table (LUT) will convert $S$ to a non-bijective S-box.

Proof: [Proof of Lemma 1] We take an arbitrary bijective S-box $S(n, n)$ and its corresponding Look-up $S_{LAT}$ and Decimal Look-up $S_{DLUT}$ tables. We pick the flipped bit to be somewhere inside the row with index $i$ of $S_{LAT}$. The resulted Look-up Table will be denoted as $S'_{LAT}$. We will prove that the S-box $S'$ is not bijective.

Indeed, if $S_{DLUT} = [d_0, d_1, \ldots, d_{2^n-1}]$, then the resulted Decimal Look-up Table of the S-box $S'$ is equal to $S'_{DLUT} = [d_0, d_1, \ldots, d_{2^n-1}]$, where $d_i$ is the decimal integer, which corresponds to the bit-concatenation of all the bits from the $i$-th row of $S'_{LAT}$. However, from the bijectivity property it follows that $\forall i: j \neq i \Rightarrow j \neq d_i$. Furthermore, by definition (see Lemma 8), $S_{DLUT}$ is in fact a permutation of all the $2^n$ integers in the interval $[0, 2^n - 1]$.

Since $S$ is with dimensions $(n, n)$, each element of $S_{DLUT}$ is represented by exactly $2^n$ bits. Having this in mind, the number of possible distinct values of $d_i$ is $2^n$ ($d_i$ with first bit flipped, $d_i$ with second bit flipped, $\ldots$, $d_i$ with last bit flipped). Since the binary representations of all those distinct values consist of exactly $n$ bits, their decimal representations are values less or equal than $2^n - 1$. Therefore, no matter which bit of $d_i$ is flipped, $d_i$ will collide with exactly one $d_j$, for some $j \neq i$. Hence, $S'_{DLUT}$ will hold two different elements, $d_i$ and $d_j$, with equal values, and therefore, $S'$ is a non-bijective S-box.

It is not possible to get a bijective S-box by modifying (flipping) a single bit of the $S_{LAT}$ of other bijective S-box. However, as shown in the next Lemma, the minimum count of bits we need to change in the LUT of a random bijective S-box to get a new bijective S-box is 2.
Lemma 2 (Couplings Lemma): The smallest nonzero number of bits from the LUT of a random bijective S-box needed to be modified to obtain another bijective S-box is 2.

Proof: [Proof of Lemma 2] Let us take a bijective S-box \( S(n, n) \) and define the DLUT of \( S \) as an array \( D_{\text{DLUT}} = [d_0, d_1, \ldots, d_{2^n-1}] \). Since \( S \) is bijective, it follows that \( \forall i: i \neq j \rightarrow d_i \neq d_j \). We recall that \( D_{\text{DLUT}} \) is in fact a permutation of all the \( 2^n \) integers in the interval \([0, 2^n - 1]\).

Without loss of generality, we pick the first flipped bit to be somewhere inside the row with index \( i \) of \( D_{\text{DLUT}} \). Let us denote the resulted Look-up Table, when this bit is flipped, as \( D'_{\text{DLUT}} \). As shown in the previous lemma, the S-box \( S' \), which corresponds to the Look-up Table \( D'_{\text{DLUT}} \), is not bijective. However, we will show that we could always flip another distinct (non-trivial) bit, which could transform the S-box \( S' \) to some bijective S-box \( S'' \), where \( S'' \neq S \).

Using the notations introduced throughout the proof of the previous lemma, we have

\[
S_{\text{DLUT}} = [d_0, d_1, \ldots, d_{2^n-1}],
\]

and

\[
S'_{\text{DLUT}} = [d_0, d_1, \ldots, \overline{d_i}, \ldots, d_{2^n-1}],
\]

where \( \overline{d_i} \) is the decimal integer, which corresponds to the bit-concatenation of all the bits from the \( i \)-th row of \( S'_{\text{DLUT}} \). Following the same observation made in Lemma 1, no matter which bit of \( d_i \) is flipped, \( \overline{d_i} \) will collide with exactly one \( d_j \), for some \( j \neq i \), and \( S'_{\text{DLUT}} \) will hold two different elements, \( \overline{d_i} \) and \( d_j \), with equal values

\[
S''_{\text{DLUT}} = [d_0, d_1, \ldots, d_j, \ldots, \overline{d_i}, \ldots, d_{2^n-1}].
\]

However, all the remaining \( 2^n-2 \) elements from \( S'_{\text{DLUT}} \), i.e. all the elements in \( S'_{\text{DLUT}} \) with \( d_i \) and \( \overline{d_i} \) excluded, differ from each other. Since \( d_j = \overline{d_i} \) and \( d_i \neq \overline{d_i} \), we could highlight two reasons for the non-bijectivity of the S-box \( S' \):

- The value \( d_i \) is missing from \( S'_{\text{DLUT}} \).
- There are two identical values in \( S'_{\text{DLUT}} \) - \( d_j \) and \( \overline{d_i} \).

Having this in mind, if we could modify \( d_j \) to \( d_i \), by using just a single flip, we could convert \( S'_{\text{DLUT}} \) to \( S''_{\text{DLUT}} \), where the S-box \( S'' \), which corresponds to the Decimal Look-up Table \( S''_{\text{DLUT}} \), is bijective. It is trivial to be shown that this modification is possible. Let us recall that \( \overline{d_i} \) is created by flipping a single bit in \( d_i \) on position, for example, \( x \). Therefore, since \( d_j = \overline{d_i} \), flipping the bit on position \( x \) in \( d_j \) will convert \( d_j \) back to \( d_i \), and we will have the DLUT \( S''_{\text{DLUT}} \), s.t.

\[
S''_{\text{DLUT}} = [d_0, d_1, \ldots, d_j, \ldots, d_i, \ldots, d_{2^n-1}].
\]

Since the elements in the S-box \( S'' \), which corresponds to the Decimal Look-up Table \( S''_{\text{DLUT}} \), are now a permutation of all the \( 2^n \) integers in the interval \([0, 2^n - 1]\), \( S'' \) is bijective. Furthermore, the permutation \( S''_{\text{DLUT}} \) is exactly one transposition away from the permutation \( D_{\text{DLUT}} \).

Definition 17 (Couplings): Let us take a bijective S-box \( S(n, n) \) and its corresponding DLUT

\[
D_{\text{DLUT}} = [d_0, d_1, \ldots, d_{2^n-1}].
\]

We define as a coupling each set \( \{d_s, f(d_s)\} \), while the set of all couplings in \( S \) as \( \{S \}$$.)

Lemma 3 (Couplings Set Cardinality): Given a bijective S-box \( S(n, n) \):

\[
|C(S)| = n2^{n-1}.
\]

Proof: [Proof of Lemma 3] If we flip a bit on column \( i \) in the LUT of \( S \), the corresponding unique second flip we need to perform to guarantee the bijectivity property of the newly created S-box has to be on column \( i \) as well, i.e. we are flipping two distinct bits sharing the same S-box coordinate \( i \). Thus, we have exactly \( n \) coordinates, each having \( \frac{2^n}{2} \) distinct couplings, or a total of \( n2^{n-1} \) couplings.

Definition 18 (Couplings Pivot Set): We define the set \( \{S \}$$.) as the maximum subset of the coupling set of a bijective S-box \( S(n, n) \), which holds couplings operating only on column \( i \) of the \( S_{\text{DLUT}} \), i.e. couplings of the form \( \{d_i, f(d_i)\} \). We call each such maximum subset \( \{S \}$$.) a couplings pivot set operating on column \( i \) of \( S_{\text{DLUT}} \).

Corollary 1 (Properties of Couplings Pivot Sets): Considering the definitions of the Couplings Pivot Sets on bijective S-box \( S(n, n) \), the following properties hold true:

- \( \forall i \neq j \), \( \{S \}$$.) \cap \{S \}$$.) = \emptyset \)
- \( \forall i \), \( |\{S \}$$.)\} = 2^{n-1} \)
- \( \bigcup_{i=1}^{n} |\{S \}$$.)\} = n2^{n-1} \)

Definition 19 (Coordinate Decomposition): Let \( S \) be an \( (n, n) \) bijective S-box. We take a random element with coordinates \( (x, y) \) of its corresponding linear approximation table \( S_{\text{LAT}} \). We denote the binary representation of \( y \) as:

\[
y(2) = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \cdots + b_12^1 + b_02^0
\]

The coordinate decomposition of an element with coordinates \( (x, y) \), denoted by \( \Delta_{x, y} \), is the set:

\[
\Delta_{x, y} = \bigcup_{i=0}^{n-1} \{b_i(n - i - 1)\}
\]

Definition 20 (Nonlinearity Bottleneck Snapshot – NBS): We define the nonlinearity bottleneck snapshot \( S_{\text{NBS}} \) of a bijective S-box \( S(n, n) \) as a set of tuples holding all coordinates of the elements of \( S_{\text{LAT}} \), which are holding down the nonlinearity value \( S_{\text{NL}} \) of \( S \), i.e.

\[
(x, y) \in S_{\text{NBS}} \iff |S_{\text{LAT}}[x][y]| = 2^{n-1} - S_{\text{NL}}
\]

Definition 21 (NBS Coordinate Decomposition – NBSD): We define the nonlinearity bottleneck snapshot coordinate
decomposition of a bijective S-box $S(n, n)$, denoted by $\Delta_S$, as a set of all $S_{NBS}$ coordinate decompositions, i.e.:

$$\Delta_S = \bigcup_{(x,y) \in S_{NBS}} \Delta_{x,y}$$

**Definition 22 (Degree of Descendibility – $\Lambda_S$):** For a given bijective S-box $S(n, n)$, we define a family of sets $\Psi_S$, s.t.:

$$E \in \Psi_S \iff \forall Q \in \Delta_S \exists q \in Q : q \in E$$

The degree of descendibility of $S$ is the minimum cardinality of a set in $\Psi_S$, i.e.:

$$\Lambda_S = \min_{\forall \Psi \in \Psi_S} |\Psi|$$

**Corollary 2 (Basic properties of $\Lambda_S$):** For a given bijective S-box $S(n, n)$:

- $\Lambda_S \in \mathbb{N}$
- $\Lambda_S \in [1, n]$
- $\Lambda_S = 1 \iff |\bigcap_{i \in \Delta_S}| \geq 1$
- $\Lambda_S > 1 \iff |\bigcap_{i \in \Delta_S}| = 0$

**Definition 23 (Descendible Coordinate):** For a given bijective S-box $S(n, n)$, we say that coordinate $j$ is descendible, if the following properties hold:

- $\Lambda_S = 1$
- $j \in \bigcap_{i \in \Delta_S}$

**Definition 24 (Couplings Transformation):** For a given bijective S-box $S(n, n)$ and some coupling $c_i$, we denote as $S^{c_i}$ the S-box created by applying coupling $c_i$ on $S$. We define this transform as coupling transform denoting it with the operator $o$, i.e.

$$S^{c_i} = S \circ c_i$$

When we have a list of couplings $\{c_i, c_2, \ldots, c_t\}$, which we want to use for transformation of $S$ in this exact order, we will use the following expression:

$$S^{c_1,c_2,\ldots, c_t} = S \circ c_1 \circ c_2 \circ \ldots \circ c_t$$

**Lemma 4 (Couplings Inverse):** For a given bijective S-box $S$ and any coupling $c$, the following property holds:

$$S = S \circ c \circ c$$

**Proof:** Since $c$ is, in fact, a transposition in the Decimal Look-up Table $S_{DLUT}$ of $S$ (swapping two elements in $S_{DLUT}$ applying the same transposition twice would cancel its effect out.

**Definition 25 (Coupling Transformation Matrix – CTM):** For a given bijective S-box $S(n, n)$ and some coupling $c_i$, we denote as $S^{c_i}_{LAT}$ the transformed LAT of $S$ caused by $c_i$. We define the coupling transformation matrix of $c_i$ on $S$, as:

$$S^{c_i}_{CTM} = S^{c_i}_{LAT} - S_{LAT}$$

**Lemma 5 (Pivot Couplings Commutativity):** For a given bijective S-box $S(n, n)$, for any two couplings $c_a$ and $c_b$, which belongs to the same couplings pivot set $\{S \downarrow^t\}$, we have the following property:

$$S \circ c_a \circ \ c_b = S \circ c_b \circ c_a$$

**Proof:** In case $c_a \equiv c_b$ the theorem follows from lemma 4, i.e.

$$S \circ c_a \circ \ c_b = S \circ c_a \circ c_a = S$$

In the case when $c_a \neq c_b$, since they belong to the same coupling pivot set, it follows that $c_a \cap c_b = \emptyset$, which concludes the proof.

**Corollary 3:** For a given bijective S-box $S(n, n)$, for any couplings $c_j$, which belongs to the same couplings pivot set $\{S \downarrow^t\}$, we have the following properties:

$$S^{c_a,c_b}_{LAT} = S^{c_a}_{LAT} + S^{c_b}_{LAT} + S^{c_a}_{CTM} + S^{c_b}_{CTM}$$

**Lemma 6 (CTM Values):** The value of each element in a CTM is -2, 0, or 2.

**Proof:** For a given bijective S-box $S(n, n)$ and some coupling $c = \{d_i, f(d_i, y)\}$, we denote as $S^{c,i}_{LAT}$ the transformed LAT of $S$ caused by $c$. Let us take some element $e_{x,y}$ from the LAT of $S$ before the coupling transformation. We have:

$$e_{x,y} = 2^{n-1} - d_H(L_q, b_1b_2 \ldots b_{2^n}),$$

for some linear function $L_q$ and some linear combination in binary representation of the coordinates of $S$: $b = b_1b_2 \ldots b_{2^n}$. If $j \notin \Delta_{x,y}$, $e_{x,y}$ is not affected after applying the coupling. However, if $j \in \Delta_{x,y}$, we know that exactly two of the bits of the linear combination $b$ are flipped. We denote them as $b_t$ and $b_{t'}$. Let us denote the element on position $(x, y)$ on the newly created LAT as $e'_{x,y}$.

$$e'_{x,y} = 2^{n-1} - d_H(L_q, b_1b_2 \ldots \bar{b}_t \ldots \bar{b}_t' \ldots b_{2^n})$$

$$= 2^{n-1} - d_H(L_q, b_1b_2 \ldots b_t \ldots b_{t'} \ldots b_{2^n}) \pm 1$$

$$= 2^{n-1} - d_H(L_q, b_1b_2 \ldots b_t \ldots b_t' \ldots b_{2^n}) \pm 1 \pm 1$$

$$= e_{x,y} \pm 1 \pm 1$$

(1)

Since the expression $\pm 1 \pm 1$ is equal to one of the three possible values: -2, 0, and 2, the proof concludes.

**Corollary 4:** For a given bijective S-box $S(n, n)$, let us apply transformations of couplings $c_1, c_2, \ldots, c_k$, which belongs to the same couplings pivot set $\{S \downarrow^t\}$. The elements of the resulting CTM are numbers in the interval $[-2k, -2(k-1), \ldots, -2, 0, 2, \ldots, 2(k-1), 2k]$. 

**Definition 26 (S-box Coordinate Extended LAT – CELAT):** For a given bijective S-box $S(n, n)$, and a given coordinate $i$, we can define the one-dimensional linear approximation table of $S$ as:

$$S^i_{LAT_{1D}}[x] = S_{LAT}[x / 2^n][x \% 2^n]$$

Furthermore, we denote all the couplings in the couplings pivot set $\{S \downarrow^t\}$ as $c_1, c_2, \ldots, c_{2^n-1}$. We have:

$$S_{CTM}^i = S^i_{LAT} - S^i_{LAT}$$
Following the same concept used in the construction of one-dimensional LAT of \( S \), we can define one-dimensional CTM, i.e.:  
\[
S^c_{CTM1D} = S^c_{LAT1D} - S_{LAT1D} \\
S^{c+1}_{CTM} = S^{c+1}_{LAT} - S_{LAT}
\]  
(2)

Finally, we define S-box \( i \)-th Coordinate Extended LAT \( S^i_{CELAT} \) as the following table:

\[
S^i_{CELAT} = \begin{bmatrix}
S_{LAT1D} \\
S^c_{CTM1D} \\
S^{c+1}_{CTM} \\
\vdots \\
S^{2n-1}_{CTM1D} 
\end{bmatrix}
\]

\( S^i_{CELAT} \) has \( 2^{n-1} + 1 \) rows and \( 2^n \) columns. For example, let us consider an S-box \( S(2, 2) \) with \( S_{DLUT} = \{0, 2, 1, 3\} \). For \( n = 2 \) the \( S^2_{CELAT} \) has \( 2^2 - 1 = 3 \) rows and \( 2^2 \times 2 = 16 \) columns. Considering coordinate 2, we have:

\[
\{S^2 \} = \{c_1, c_2\} = \{\{0, 1\}, \{2, 3\}\} \\
S^{c1}_{DLUT} = \{1, 2, 0, 3\} \\
S^{c2}_{DLUT} = \{0, 3, 1, 2\} \\
S^2_{CELAT} = \begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

### IV. TRANSLATING AN S-BOX NL OPTIMIZATION MODEL PROBLEM TO A BINARY INTEGER PROGRAMMING MODEL

In this chapter, the transition from the S-box optimization problem to a system of linear constraints is presented. We are following the same notations as in [23].

**Definition 27** Integer Programming – Optimization Problem: A pure integer linear program is a problem of the form:

\[
\begin{align*}
\text{max} & \quad cx \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0 \text{ integral}
\end{align*}
\]

where the data consists of the row vector \( c \) with size \( n \), \( (m, n) \) matrix \( A \), and column vectors \( b \) and \( x \) with respective sizes of \( m \) and \( n \). The column vector \( x \) contains the variables to be optimized. We say that the set \( S \) is the set of feasible solutions, i.e.:

\[
S := \{ x \in \mathbb{Z}_{+}^n : Ax \leq b \}
\]

**Definition 28** (Binary Integer Linear Programming – BiILP): A pure binary integer linear program is a problem of the form:

\[
\begin{align*}
\text{max} & \quad cx \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0 \text{ binary}
\end{align*}
\]

where the data consists of \( c \) with size \( n \), \( (m, n) \) matrix \( A \), and column vectors \( b \) and \( x \) with respective sizes of \( m \) and \( n \). The column vector \( x \) contains the binary variables to be optimized. We say that the set \( S \) is the set of feasible solutions, i.e.:

\[
S := \{ x \in \mathbb{B}^n : Ax \leq b \}
\]

In the context of the feasibility problem we are looking for just one element in the set \( S \), not the optimal one.

For an \( (n, n) \) S-box \( S \), we denote \( 2^{n-1} \) by \( r \) and \( 2^n \) by \( m \). Let us construct its CELAT using coordinate \( i \) i.e.

\[
S^i_{CELAT} = \begin{bmatrix}
S_{LAT1D} \\
S^{c1}_{CTM1D} \\
S^{c2}_{CTM1D} \\
\vdots \\
S^{2n-1}_{CTM1D} 
\end{bmatrix} = \begin{bmatrix}
l_1 & l_2 & \cdots & l_m \\
c_{11} & c_{12} & \cdots & c_{1m} \\
c_{21} & c_{22} & \cdots & c_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
c_{r1} & c_{r2} & \cdots & c_{rm} 
\end{bmatrix}
\]

We want to apply some coupling transformations subset \( P = p_1, p_2, \cdots, p_k \) which belongs to the pivot coupling set \( \{S^i \} \). From corollary 3 it follows that:

\[
S^{p_1, p_2, \cdots, p_k}_{LAT1D} = S_{LAT} + \sum_{i=1}^{k} S^{p_i}_{CTM}
\]

We denote

\[
S_{LAT1D} = [q_1, q_2, \cdots, q_m]
\]

Then, we can construct the following system of equations:

\[
q_1 = l_1 + c_{11}x_1 + c_{21}x_2 + \cdots + c_{r1}x_r \\
q_2 = l_2 + c_{12}x_1 + c_{22}x_2 + \cdots + c_{r2}x_r \\
\vdots \\
q_m = l_m + c_{1m}x_1 + c_{2m}x_2 + \cdots + c_{rm}x_r
\]

where \( x = (x_1, x_2, \cdots, x_r) \) \( \in \mathbb{B}^r \), and \( x_i = 1 \) if \( p_i \in P \).

We have \( S_{NL} = 2^{n-1} - \max_{j=1}^m abs(l_j) \). If coordinate \( i \) is
descendable, we can construct the following binary integer programming feasibility problem:

subject to \( < S'_{CELAT}^{T}, x > \leq A \)
subject to \( < S'_{CELAT}^{T}, x > \geq B \)
\( x \geq 0 \) binary

where \( A \) is a column vector with \( 2^{n-1} + 1 \) elements, each equal to \( 2^{n-1} - S_{NL} - 2 \), while \( B \) is a column vector with \( 2^{n-1} + 1 \) elements, each equal to \( S_{NL} - 2^{n-1} + 2 \). Let us denote the SAT problem descending on coordinate \( i \) in equation 5 as \( \Omega_{S,i} \). This is \( NP \)-hard problem with a total of \( 2^{n-1} \) binary variables and \( 2^n + 2 \) restrictions. However, we can further divide the problem to an union of subproblems, i.e.:

\[
\Omega_{S,i} = \bigcup_{d=1}^{n-1} \Omega_{S,i}^d
\]

where each subproblem \( \Omega_{S,i}^d \) is modelled using the following restrictions:

subject to \( < S'_{CELAT}^{T}, x > \leq A \)
subject to \( < S'_{CELAT}^{T}, x > \geq B \)
\( \sum_{j=1}^{r} x_j = d \)
\( x \geq 0 \) binary

Solving any of the subproblems will yield a solution to the original problem.

**Corollary 5:** For subproblems \( \Omega_{S,i}^d \) of a binary integer programming feasibility problem \( \Omega_{S,i} \), the following property holds:

\[
\bigcup_{d=1}^{n-1} \Omega_{S,i}^d = \emptyset
\]

**Corollary 6:** The search space of the subproblem \( \Omega_{S,i}^d \) for the bijective S-box \( S(n, n) \) is \( 2^{(2^n - 1)} \).

**Theorem 1:** For a subproblem \( \Omega_{S,i}^d \), all restrictions with participation of some \( l_j \) for which the following inequalities hold:

\[
l_j \leq 2^{n-1} - S_{NL} - 2d - 2
\]
\[
l_j \geq S_{NL} - 2^{n-1} + 2d + 2
\]

are always satisfied.

**Proof:** For a subproblem \( \Omega_{S,i}^d \) we have the following restrictions:

\[
l_1 + c_{11}x_1 + c_{21}x_2 + \cdots + c_{1r}x_r \leq 2^{n-1} - S_{NL} - 2
\]
\[
l_1 + c_{11}x_1 + c_{21}x_2 + \cdots + c_{1r}x_r \geq S_{NL} - 2^{n-1} + 2
\]
\[
l_2 + c_{12}x_1 + c_{22}x_2 + \cdots + c_{2r}x_r \leq 2^{n-1} - S_{NL} - 2
\]
\[
l_2 + c_{12}x_1 + c_{22}x_2 + \cdots + c_{2r}x_r \geq S_{NL} - 2^{n-1} + 2
\]
\[
\vdots
\]
\[
l_m + c_{1m}x_1 + c_{2m}x_2 + \cdots + c_{rm}x_r \leq 2^{n-1} - S_{NL} - 2
\]
\[
l_m + c_{1m}x_1 + c_{2m}x_2 + \cdots + c_{rm}x_r \geq S_{NL} - 2^{n-1} + 2
\]

From lemma 6, we know that the possible values of the elements \( c_{ij} \) are \(-2, 0 \) or \( 2 \). Hence:

\[
\min_{i,j} c_{ij} = -2
\]
\[
\max_{i,j} c_{ij} = 2
\]

Since \( \sum x_j = d \), we have:

\[
\min_{j} (c_{1j}x_1 + c_{2j}x_2 + \cdots + c_{rj}x_r) = -2d
\]
\[
\max_{j} (c_{1j}x_1 + c_{2j}x_2 + \cdots + c_{rj}x_r) = 2d
\]

If for some \( l_j \) the following inequalities hold:

\[
l_j \leq 2^{n-1} - S_{NL} - 2d - 2
\]
\[
l_j \geq S_{NL} - 2^{n-1} + 2d + 2
\]

then

\[
l_1 + c_{11}x_1 + c_{21}x_2 + \cdots + c_{1r}x_r
\leq l_1 + \max_{j} (c_{1j}x_1 + c_{2j}x_2 + \cdots + c_{rj}x_r)
\leq l_1 + 2d
\leq 2^{n-1} - S_{NL} - 2d + 2d
\leq 2^{n-1} - S_{NL} - 2
\]

and on the other hand

\[
l_1 + c_{11}x_1 + c_{21}x_2 + \cdots + c_{1r}x_r
\geq l_1 - \min_{j} (c_{1j}x_1 + c_{2j}x_2 + \cdots + c_{rj}x_r)
\geq l_1 - 2d
\geq S_{NL} - 2^{n-1} + 2d - 2d
\geq S_{NL} - 2^{n-1} + 2
\]

which completes the proof.

**Definition 30 (CELAT with radius R):** For a given bijective S-box \( S(n, n) \), and a given coordinate \( i \), we have:

\[
S'_{CELAT}^i = \begin{bmatrix}
    l_1 & l_2 & \cdots & l_m \\
    c_{11} & c_{12} & \cdots & c_{1m} \\
    c_{21} & c_{22} & \cdots & c_{2m} \\
    \vdots & \vdots & \ddots & \vdots \\
    c_{r1} & c_{r2} & \cdots & c_{rm} \\
\end{bmatrix}
\]

We define as \( S'_{CELAT}^{i,R} \) a matrix constructed of those columns of \( S'_{CELAT}^i \) with first element \( \rho \), for which the following inequalities hold:

\[
\rho > 2^{n-1} - S_{NL} - 2R - 2
\]
\[
\rho < S_{NL} - 2^{n-1} + 2R + 2
\]

Hence, a given subproblem \( \Omega_{S,i}^d \) could be further reduced and launched on \( S'_{CELAT}^{d,R} \), instead of its corresponding full (unreduced) version \( S'_{CELAT}^i \).
FIGURE 1. Automata representation of the optimization process.

V. ALGORITHM
By using automata notation, Figure 1 presents the distinct steps of the optimization process. State $S$ is the initial state of the automata. In this phase, we initialize and process the input. We make some additional checks about the properties of the S-box. For example, we check the bijectivity property of the S-box. We further analyze and extract the descendible coordinates (if such exist).

For example, if we first choose the coordinate $j$ to descent into, we further calculate the corresponding matrix $S^{l}_{CELAT}$. Then, the adjacent state $Re$, by further processing the generated matrix, and by using some radius $R$, generates the matrix $S^{l}_{CELAT}$. Finally, state BIP is translating the problem to a binary integer programming feasibility problem. In case a feasible solution or proof that the problem is infeasible is found, the result is reported back to state Re, the major role of which is to orchestrate the behavior of the optimization routine - increasing the radius, changing the descendible coordinate, or give up.

Few important properties of the automata should be emphasized. For a given S-box $S$, if $\Delta_{S} = 1$, then at least one descendible coordinate, for example $j$, does exist. Thus, if a feasible solution of $\Omega_{S,j}^{R}$ is found, the nonlinearity of $S$ could be increased by activating exactly $R$ couplings. Therefore, we could not only optimize the nonlinearity of $S$, but dictate the impact of our changes to the original S-box as well - increasing the value of $R$ will increase the total count of flipped bits in $S$.

In those cases where $\Delta_{S} > 1$, the aforementioned algorithm, as we will later demonstrate, is still applicable. We just pick a random coordinate $j \in \Delta_{S}$, instead of a descendible one. As a consequence, finding a solution to the $\Omega_{S,j}^{R}$ problem will not increase the nonlinearity of $S$. However, it will decrease the value of $\Delta_{S} = 1$. Thus, by repeating the reduction phase, we would eventually reduce the initial problem to a problem having $\Delta_{S'} = 1$, for some S-box $S'$, yielded by the optimization routine, or the composition of optimization routines, performed on $S$.

VI. EXPERIMENTAL RESULTS
We have implemented the algorithm by using Python, for states S and Re, and the Gurobi SAT Solver [24], for the BIP state itself. We analyzed two famous S-boxes: the Skipjack S-box, developed by the U.S. National Security Agency (NSA) [25], which we will denote as Sk, and the Kuznyechik S-box, standardized by the Russian Federation’s standardization agency [26], which we will denote as Kk.

A. SKIPJACK (CASE $S_{K}$)
The characteristics of $Sk$ are $Sk_{NL} = 100$, $Sk_{NBS} = \{(138, 89), (125, 168), (77, 168)\}$, $\Delta_{Sk} = \{(1, 3, 4, 7), (0, 2, 4)\}$ and $\Lambda_{Sk} = 1$. Since the coordinate with index 4 is descendible, we first try to solve the problem $\Omega_{Sk,4}^{1}$, trying the minimum possible radius value of 1. The translated BIP model consists of 51 rows, 128 columns, and 3420 nonzeros. By using a general-purpose CPU, it took approximately 0.115 seconds to prove that $\Omega_{Sk,4}^{1}$ is infeasible. However, by increasing the radius value by 1, the translated BIP model of $\Omega_{Sk,4}^{2}$, consisting of 189 rows, 128 columns, and 12640 nonzeros, a solution is found. The time required was 0.301 seconds. The found solution coupling set is $\{(130, 138), (183, 191)\}$. Indeed, the resulting S-box does possess a nonlinearity of 102. Furthermore, the found solution required the flipping of only 4 bits, since, by design, each activated coupling modify exactly 2 bits in the S-box it was launched on.

On the other hand, if we require a higher nonlinearity, combined with a greater impact of the structure of $Sk$, we could significantly increase the value of $R$. Indeed, using a radius value of 10, the translated BIP model of $\Omega_{Sk,4}^{10}$ consists of 25125 rows, 128 columns, and 1621980 nonzeros. Despite the greater model, after 14.542 seconds, a solution was found, yielding an S-box with nonlinearity 102, constructed from Sk by flipping exactly 20 bits.

B. KUZNYECHIK (CASE $K_{K}$)
The characteristics of $Kk$ are $Kk_{NL} = 100$, $Kk_{NBS} = \{(90, 47), (184, 105), (55, 165), (102, 103), (222, 151), (62, 105), (72, 85), (237, 98), (110, 15), (246, 28), (65, 106), (135, 171), (76, 167), (251, 54)\}$, $\Delta_{Kk} = \{(2, 3, 5, 6), (1, 2, 5, 6, 7), (0, 2, 5, 7), (1, 3, 5, 7), (1, 2, 4, 7), (0, 2, 5, 6, 7), (0, 3, 5, 6, 7), (3, 4, 5), (0, 2, 4, 6, 7), (1, 2, 6), (1, 2, 4, 6), (2, 4, 5, 6, 7), (4, 5, 6, 7)\}$ and $\Lambda_{Kk} = 2$. Since the degree of descendibility is greater than 1, more precisely $\Delta_{Kk} = \{(2, 5)\}$, we pick a random coordinate from $\Delta_{Kk}$. $\Omega_{Kk,5}^{1}$ and $\Omega_{Kk,5}^{2}$ are reported back as infeasible. However, $\Omega_{Kk,5}^{10}$, consisting of 319 rows, 128 columns and 20904 nonzeros, is feasible. Again, the solver took less than a second to find a solution, i.e. the coupling set $\{(0, 4), (67, 71), (136, 140)\}$. Let’s denote the resulted S-box as $Kk$. The characteristics of $Kk$ are $Kk_{NL} = 100$, $Kk_{NBS} = \{(65, 106), (135, 171), (62, 105), (237, 98), (184, 105)\}$, $\Delta_{Kk} = \{(0, 2, 4, 6, 7), (1, 2, 6), (1, 2, 4, 6), (2, 4, 5, 6, 7), (4, 5, 6, 7)\}$ and $\Lambda_{Kk} = 1$. As expected, the value of $A$ is decreased by 1. Thus, we continue the optimization process by the descendible coordinate with index 2. First, the infeasibility of the two models $\Omega_{Kk,2}^{1}$ and $\Omega_{Kk,2}^{2}$ are proved. However, the BIP model of $\Omega_{Kk,2}^{10}$, consisting of 379 rows, 128 columns, and 25344 nonzeros, yielded a solution for less than a second. Indeed, the found coupling set $\{(95, 127), (207, 239), (108, 157)\}$ increased the nonlinearity of $Kk$ from 100 to 102. Hence, we have shown how Kk could be optimized to an S-box with higher nonlinearity by tweaking just 12 bits.
C. ACNV PROBLEM

The ACNV optimization problem could be represented as a special, and significantly lighter, in terms of computational burden, case of $S^\ell_{CELAT}$, where $S$ denotes the initial S-box and $\ell$ denotes the coordinate of $S$ to be optimized. Since our goal is ACNV optimization only, we could significantly reduce the size of $S^\ell_{CELAT}$. Let us denote as $S^\ell_{ACNV}$ the matrix formed by the matrix $S^\ell_{CELAT}$, with columns corresponding to linear combinations of coordinates of $S$ removed. Indeed, this is a significant reduction of the feasibility model. For example, if $S$ is of dimension $(n, n)$, $S^\ell_{CELAT}$ has $2^{n-1} + 1$ rows and $2^{2n}$ columns, while $S^\ell_{ACNV}$ has $2^{n-1} + 1$ rows and $2^n$ columns. We further denote the corresponding feasibility problem corresponding to $S^\ell_{ACNV}$ as $\Psi_{S,\ell}$. As usual, we could divide the problem $\Psi_{S,\ell}$ to a union of subproblems $\Psi_{S,i}^d$.

We want to emphasize, that collecting all published results regarding the ACNV problem is a rigid and tedious task. However, during our extensive research, in Table 1, we have collected the most significant of them. It appears, that the majority of papers focused on optimizing the ACNV value of (8,8) S-boxes, were skeptical about the existence of (8,8) S-box with ACNV greater than 112. Indeed, as stated in [70], “It is seen that the average value obtained from the proposed method is close to 112 (theoretical optimal value for an 8 × 8 S-box)”. However, as we have already shown in [18], bijective (8,8) S-boxes with ACNV up to 114 are easy to construct.

We have initiated the optimization routine on a bijective S-box, for simplicity denoted as $S$, from [18], possessing the highest, currently known, ACNV of 114.5. It is composed of 6 coordinates with a nonlinearity value of 114 and 2 coordinates with a nonlinearity value of 116. We will outline a possible trace of improvement, which led to an S-box with an ACNV of 116.0.

- We launched $\Psi_{S,4}^{\leq 9}$. After around 153 seconds a feasible solution, with exactly 9 couplings, was found. We activated the couplings to get $S_1$. ACNV was lifted to 114.75.
- We launched $\Psi_{S,1}^{\leq 8}$ After around 16 seconds a feasible solution, with exactly 8 couplings, was found. We activated the couplings to get $S_2$. ACNV was lifted to 115.
- We consequently launched $\Psi_{S,2}^{\leq 9}$, $\Psi_{S,3}^{\leq 9}$ and $\Psi_{S,5}^{\leq 9}$, to prove their infeasibility for respectively 181, 274 and 173 seconds. However, by launching $\Psi_{S,7}^{\leq 9}$, after 257 seconds, a feasible solution with exactly 9 couplings was found. We activated the couplings to get $S_3$. ACNV was lifted to 115.25.
- We consequently launched $\Psi_{S,2}^{\leq 9}$, $\Psi_{S,2}^{\leq 10}$, $\Psi_{S,2}^{\leq 11}$, to prove their infeasibility for respectively 151, 698 and 3457 seconds. Then, we continued with $\Psi_{S,3}^{\leq 9}$ and $\Psi_{S,3}^{\leq 10}$, to prove their infeasibility respectively for 240 and 1015 seconds. However, $\Psi_{S,3}^{\leq 11}$ yielded a solution after 5171 seconds. We activated the 11 couplings to get $S_4$. ACNV was lifted to 115.5.

- We continued with $\Psi_{S,2}^{\leq 9}$ and $\Psi_{S,2}^{\leq 10}$, to prove their infeasibility for respectively 170 and 715 seconds. However, $\Psi_{S,2}^{\leq 11}$ yielded a solution after 1145 seconds. We activated the 11 couplings to get $S_5$. ACNV was lifted to 115.75.
- Finally, we launched $\Psi_{S,5}^{\leq 9}$ and a feasible solution was found after 69 seconds. We activated the 9 couplings to get $S_6$. ACNV was lifted to 116.

We present $S_6$ as a hexadecimal format:

| Method | Min NL | Max NL | ACNV |
|--------|--------|--------|------|
| [27][28] | 84     | 106    | 100.0 |
| [29]   | 98     | 108    | 102.3 |
| [30]   | 96     | 106    | 102.5 |
| [31]   | 100    | 106    | 103.0 |
| [32]   | 96     | 106    | 103.0 |
| [33]   | 98     | 108    | 103.0 |
| [34]   | 98     | 108    | 103.2 |
| [35]   | 100    | 106    | 103.2 |
| [36]   | 99     | 106    | 103.3 |
| [37]   | 101    | 108    | 103.8 |
| [38]   | 101    | 108    | 103.8 |
| [39]   | 102    | 106    | 104.0 |
| [40]   | 98     | 108    | 104.0 |
| [41]   | 100    | 106    | 104.0 |
| [42]   | 102    | 106    | 104.0 |
| [43]   | 98     | 108    | 104.0 |
| [44]   | 102    | 108    | 104.5 |
| [45][46] | 100   | 106    | 104.7 |
| [47][48] | 102  | 108    | 104.7 |
| [49]   | 100    | 108    | 104.75 |
| [50]   | 102    | 108    | 105.2 |
| [51]   | 102    | 108    | 105.3 |
| [52]   | 100    | 110    | 105.5 |
| [53]   | 98     | 110    | 105.5 |
| [54]   | 102    | 110    | 105.5 |
| [55]   | 104    | 108    | 105.7 |
| [56]   | 102    | 108    | 106.0 |
| [57]   | 104    | 110    | 106.0 |
| [58]   | 104    | 110    | 106.2 |
| [59]   | 104    | 110    | 106.5 |
| [60]   | 106    | 108    | 106.5 |
| [61][62] | 106  | 108    | 106.7 |
| [63]   | 104    | 108    | 106.7 |
| [64]   | 106    | 110    | 107.0 |
| [65]   | 106    | 108    | 107.5 |
| [66]   | 106    | 110    | 107.75 |
| [67]   | 108    | 108    | 108.0 |
| [68]   | 104    | 110    | 108.0 |
| [69]   | 106    | 110    | 108.5 |
| [70]   | 108    | 110    | 108.75 |
| [71]   | 108    | 110    | 109.0 |
| [72][73][74][75] | 112 | 112    | 112.0 |
| [18]   | 114    | 116    | 114.5 |
| this work | 114  | 116    | 114.75 |
| this work | 114  | 116    | 115.0 |
| this work | 114  | 116    | 115.25 |
| this work | 114  | 116    | 115.5 |
| this work | 116  | 116    | 115.75 |
| this work | 116  | 116    | 116.0 |

TABLE 1. A comparison of S-boxes, yielded by various methods to be found in the literature.
The overall nonlinearity of \( S_6 \) is 92.

Significant efforts were made to reach higher ACNV - reaching higher ACNV would reveal a balanced Boolean function having a nonlinearity of 118. Unfortunately, all instances \( \Psi_{S_6}^{x,y} \), for \( \forall x, y : x \leq 21, y \in [1, 8] \), were proved infeasible. We want to emphasize that the searching routine is highly efficient. For example, \( \Psi_{S_6}^{17} \), for some \( y \), was proved infeasible for 1065 seconds or approximately 18 minutes. Since the given search space size is equal to \( 128 \) or approximately \( 2^{25} \), this results in checking simultaneously roughly \( 2^{25} \) distinct cases per second.

VII. CONCLUSION

In this work, we have projected the S-box nonlinearity optimization problem to the domain of integer binary programming. The provided framework is not only capable of increasing the nonlinearity value of a given S-box, but also achieving that with a minimum impact on the initial structure. We have demonstrated how the Skipjack S-box and the Kuznyechik S-box could be both optimized to a higher nonlinearity by tweaking, respectively, just 4 and 12 bits. We further revealed that bijective (8,8) S-boxes, the eight coordinates of which possess the currently known optimal nonlinearity value of 112, do exist.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable feedback on the manuscript. The comments and suggestions significantly improved the quality of this work.

REFERENCES

[1] J. Daemen and V. Rijmen, The Design of Rijndael: AES—The Advanced Encryption Standard, vol. 2. New York, NY, USA: Springer-Verlag, 2002.

[2] G. Orhansoy, S. E. Hajji, and Y. Bentaleb, “SNOW 3G stream cipher operation and complexity study,” Contemp. Eng. Sci.-Hikari Ltd, vol. 3, no. 3, pp. 97–111, 2010.

[3] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, “TwoFish: A 128-bit block cipher,” NIST AES Proposal, vol. 15, no. 1, pp. 23–91, 1998.

[4] P. Barreto and V. Rijmen, “The Whirlpool hashing function,” in Proc. 1st Open NESSIE Workshop, vol. 13. Leuven, Belgium: Citeseer, 2000, p. 14.

[5] G. Piret, T. Roche, and C. Carlet, “PICARO—A block cipher allowing efficient higher-order side-channel resistance,” in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur. Berlin: Germany: Springer, 2012, pp. 312–328.

[6] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, and G. Leander, “Prince—A low-latency block cipher for pervasive computing applications,” in Proc. Int. Conf. Appl. Cryptol. Inf. Secur. Berlin: Germany: Springer, 2012, pp. 208–225.

[7] E. Biham, “On Matsui’s linear cryptanalysis,” in Proc. Workshop Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 1994, pp. 341–355.

[8] E. Biham and A. Shamir, “Differential cryptanalysis of DES-like cryptosystems,” J. Cryptol., vol. 4, no. 1, pp. 3–72, 1991.

[9] D. Wagner, “The boomerang attack,” in Proc. Int. Workshop Fast Softw. Encryption. Berlin: Germany: Springer, 1999, pp. 156–170.

[10] N. T. Courtois and J. Pieprzyk, “Cryptanalysis of block ciphers with overdefined systems of equations,” in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. Berlin: Germany: Springer, 2002, pp. 267–287.

[11] H. Gilbert and T. Peyrin, “Super-Box cryptanalysis: Improved attacks for AES-like permutations,” in Proc. Int. Workshop Fast Softw. Encryption. Berlin: Germany: Springer, 2010, pp. 365–383.

[12] K. Nyberg, “Perfect nonlinear S-boxes,” in Proc. Workshop Theory Appl. Cryptograph. Techn. Berlin: Germany: Springer, 1991, pp. 378–386.

[13] W. Millan, “How to improve the nonlinearity of bijective S-boxes,” in Proc. Australas. Conf. Inf. Secur. Privacy. Berlin, Germany: Springer, 1998, pp. 181–192.

[14] J. A. Clark, J. L. Jacob, and S. Stepney, “The design of S-boxes by simulated annealing,” New Gener. Comput., vol. 23, no. 3, pp. 219–231, 2005.

[15] W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson, “Evolutionary heuristics for finding cryptographically strong S-boxes,” in Proc. Int. Conf. Inf. Commun. Secur. Berlin: Germany: Springer, 1999, pp. 263–274.

[16] O. Kazymyrov, V. Kazymyrova, and R. Oliynykov, “A method for generation of high-nonlinear S-boxes based on gradient descent,” IACR Cryptol. ePrint Arch., vol. 2013, p. 578, Jun. 2013.

[17] P. Tesaf, “A new method for generating high non-linearity S-boxes,” Radioengineering, vol. 19, no. 1, pp. 23–26, 2010.

[18] M. M. Dimitrov, “On the design of chaos-based S-boxes,” IEEE Access, vol. 8, pp. 117173–117181, 2020.

[19] S. Picek, R. Santana, and D. Jakobovic, “Maximal nonlinearity in balanced Boolean functions with even number of inputs, revisited,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jun. 2016, pp. 3222–3229.

[20] P. Sarkar and S. Maitra, “Nonlinearity bounds and constructions of resilient Boolean functions,” in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 2000, pp. 515–532.

[21] C. Carlet, Y. Crama, and P. L. Hammer, Boolean Functions for Cryptography and Error-Correcting Codes. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[22] C. Carlet, “Vectorial Boolean functions for cryptography,” in Boolean Models and Methods in Mathematics, Computer Science, and Engineering, vol. 134. Cambridge, U.K.: Cambridge Univ. Press, 2010, pp. 398–469.

[23] M. Conforti, G. Cornuéjols, and G. Zambelli, Integer Programming, vol. 271. Cham, Switzerland: Springer, 2014.

[24] I. Gurobi Optimization. (2018). Gurobi Optimizer Reference Manual. Available: http://www.gurobi.com.

[25] N. Skipjack. (1998). KEA Algorithm Specifications. [Online]. Available: https://csrc.nist.org/encryption/skipjack/skipjack.pdf.

[26] V. Dolmatov, “GOST R 34.12-2015: Block cipher ‘Kuznyechik,’” Transformation, vol. 50, p. 10, Mar. 2016.

[27] M. Khan, “A novel image encryption scheme based on multiple chaotic S-boxes,” Nonlinear Dyn., vol. 82, no. 1, pp. 527–533, 2015.

[28] M. Khan, T. Shah, and S. Batool, “Construction of S-box based on chaotic Boolean functions and its application in image encryption,” Neural Comput. Appl., vol. 27, no. 3, pp. 677–685, 2016.

[29] S. Jamal, M. Khan, and T. Shah, “A watermarking technique with chaotic fractional S-box transformation,” Wireless Pers. Commun., vol. 90, no. 4, pp. 2033–2049, 2016.

[30] M. Khan and Z. Asghar, “A novel construction of substitution box for image encryption with applications with gingerbreadman chaotic map and Sa permutation,” Neural Comput. Appl., vol. 29, no. 4, pp. 993–999, Feb. 2018.

[31] G. Chen, Y. Chen, and X. Liao, “An extended method for obtaining S-boxes based on three-dimensional chaotic baker maps,” Chaos, Solitons Fractals, vol. 31, no. 3, pp. 571–579, Feb. 2007.

[32] M. Khan, T. Shah, H. Mahmood, M. Gondal, and I. Hussain, “A novel technique for the construction of strong S-boxes based on chaotic Lorenz systems,” Nonlinear Dyn., vol. 70, no. 3, pp. 2303–2311, 2012.

[33] M. Khan, T. Shah, H. Mahmood, and M. Gondal, “An efficient method for the construction of block cipher with multi-chaotic systems,” Nonlinear Dyn., vol. 71, no. 3, pp. 489–492, 2013.

[34] G. Jakimoski and L. Kocarev, “Chaos and cryptography: Block encryption ciphers based on chaotic maps,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 48, no. 2, pp. 163–169, Feb. 2001.
