Hindawi
Security and Communication Networks
Volume 2023, Article ID 5189651, 7 pages
https://doi.org/10.1155/2023/5189651

Research Article

New Results on Boolean Functions with High Nonlinearity and Low Transparency Order

Yibo Zhang, Yu Wang, Qichun Wang, Yinxia Sun, and Zhipeng Yu

College of Computer Science and Technology, Nanjing Normal University, Nanjing, China

Correspondence should be addressed to Qichun Wang; qcwang@fudan.edu.cn

Received 29 September 2022; Revised 1 February 2023; Accepted 4 March 2023; Published 17 April 2023

Academic Editor: Abdalhossein Rezai

We improve the steepest descent algorithm and increase the double threshold parameter, which significantly improves the algorithm's efficiency. And we design a new cost function so that in terms of search, various characteristics of Boolean functions can be taken into account simultaneously. Applying our algorithm, there are excellent results regarding the 9, 10, 11, and 12 variables. We find a Boolean function with a nonlinearity of 242 in 9 variables and the whole search space. Previously, this result only appeared in the rotational symmetry class. The best-achieved nonlinearity result for permutation (6, 5, 1, 4, 7, 2, 3, 0, 8) and (0, 7, 2, 5, 4, 1, 3, 6, 8) class is 238 and 239 introduced by Kavut in Information and Computation (2010). Still, applying our algorithm, we obtain a balanced Boolean function with a nonlinearity of 240 under the same permutation, indicating that our method is more general. Among the 11 variables, a Boolean function with a higher nonlinearity and a lower transparency level and the absolute value spectrum are maintained at a lower level. The algorithm performs well when considering all aspects of the property. There are similarly promising results in even-numbered variables.

1. Introduction

Boolean functions used in symmetric ciphers should have good cryptographic properties, such as balancedness, correlation immunity, high nonlinearity, high algebraic degree, high algebraic immunity, and low transparency order. However, all such characteristics cannot be optimum simultaneously and trade-offs should be considered. Therefore, constructions of Boolean functions with compromise criteria always challenge open problems [1, 2]. At the same time, many heuristic algorithms are applied to the search for Boolean functions, and many Boolean functions with good properties are obtained. Many heuristic algorithms mainly focus on the cryptographic properties of a Boolean function, and it is not easy to consider other properties simultaneously.

1.1. Related Work. Hill climbing (HC) and genetic algorithm (GA) were first applied to search for highly nonlinear Boolean functions in 1996 [3, 4] by modifying the true table of a Boolean function. Simultaneously, the literature shows numerous cryptographically interesting Boolean functions with more density in RSBFs [5, 6]. It will be helpful for us to capture the desired Boolean functions in this class. In 2007, using a steepest-descent-like algorithm, Maity and Maitra [7] searched Boolean functions in 9 variables with a nonlinearity of 241. In 2010, Kavut and Yücel [8] found a Boolean function with a 9-variable nonlinearity of 242 in the rotational and dihedral symmetry classes. Afterwards, Chakraborty et al. [9] redefined the transparency order in 2017, and Wang and Stănică [10] analyzed theoretically the transparency order constructed two infinite classes of balanced semibent Boolean functions with provably relatively good transparency order in 2019. In addition, Kavut et al. [11] applied the steepest-descent-like iterative search algorithm to build a Boolean function with lower autocorrelation.

1.2. Our Contribution. In our work, we developed an efficient algorithm based on the steepest-descent-like iterative

algorithm. We developed a new steepest-descent-like iterative algorithm. When the number of iterations increases, the cost tends to get stuck in a loop. We have added a double threshold parameter and reset the operation to deal with this situation. And a new cost function is designed to obtain excellent nonlinearity, absolute indicator, and transparency order.

We have found a 9-variable Boolean function with excellent properties, which has a short transparency order and autocorrelation while maintaining a high nonlinearity of 242 in the rotational symmetry class. At the same time, we find the Boolean function with nonlinearity 242 in the whole search space using the randomization seed. This is the first time to find a 9-variable Boolean function with a nonlinearity of 242 without restricting the search range. We also found an 11-variable Boolean function with excellent properties over the results presented in [12]. It has higher nonlinearity and lower transparency order. The results of the article [8] mention that the nonlinearity of the 9-variable Boolean function satisfies the nonlinearity of (6, 5, 1, 4, 7, 2, 3, 0, 8) and (0, 7, 2, 5, 4, 1, 3, 6, 8) permutation, the best, respectively, for 238 and 239. In contrast, this article's steepest-descent dual reset algorithm improves the result to 240.

2. Preliminaries

A Boolean function on n variables may be viewed as a mapping from $V_n = \{0,1\}^n$ to $\{0,1\}$. The truth table of a Boolean function $f(x_1, \ldots, x_n)$ is a binary string of length 2^n.

$$f = [f(0,0,\ldots,0), f(1,0,\ldots,0), f(0,1,\ldots,0), \ldots, f(1,1,\ldots,1)]. \tag{1}$$

The Hamming weight of a binary string S is the number of 1's in S denoted by wt (S). An n-variable function f is said to be balanced if its truth table contains an equal number of 0's and 1's, i.e., wt $(f) = 2^{n-1}$. Also, the Hamming distance between equidimensional binary strings S_1 and S_2 is defined by $d(S_1, S_2) = \text{wt}(S_1 \oplus S_2)$, where $\oplus$ denotes the addition over GF (2).

An n-variable Boolean function $f(x_1, \ldots, x_n)$ can be considered to be a multivariate polynomial over GF (2). This polynomial can be expressed as a sum of the product representation of all particular kth-order products $(0 \le k \le n)$ of the variables. More precisely, $f(x_1, \ldots, x_n)$ can be written as

$$a_0 \oplus \bigoplus_{1 \le i \le n} a_i x_i \oplus \bigoplus_{1 \le i < j \le n} a_{ij} x_i x_j \oplus \cdots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n, \tag{2}$$

where the coefficients $a_0, a_{ij}, \ldots, a_{12\ldots n} \in \{0,1\}$. This representation of f is called the algebraic normal form (ANF) of f. The number of variables in the highest order product term with a nonzero coefficient is called the algebraic degree, or the degree of f, denoted by $\deg(f)$.

Functions of degree, at most one, are called affine functions. An affine function with a constant term equal to zero is called a linear function. The set of all n-variable affine (respectively, linear) functions is denoted by $A(n)$ (respectively, (n)).

Definition 1. The nonlinearity of an n-variable function f is

$$nl(f) = \min_{g \in A(n)} (d(f, g)), \tag{3}$$

i.e., the minimum distance from the set of all n-variable affine functions.

Definition 2. Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$, both belonging to $\{0,1\}^n$ and $x \cdot \omega = x_1\omega_1 \oplus \cdots \oplus x_n\omega_n$. Let $f(x)$ be a Boolean function on n variables. Then, the Walsh transform of $f(x)$ is a real-valued function over $\{0,1\}^n$ which is defined as

$$W_f(\omega) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus x \cdot \omega}. \tag{4}$$

Definition 3. The autocorrelation function of a Boolean function f at a point α is defined by

$$C_f(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + f(x+\alpha)}. \tag{5}$$

We are interested to find out the point(s) α for which the absolute value of $C_f(\alpha)$ is high.

Definition 4. The absolute indicator is defined as

$$\Delta_f = \max_{a \in \mathbb{F}_2^{n*}} \left| C_f(a) \right|. \tag{6}$$

For cryptographic purposes, our primary motivation is to construct Boolean function(s) f with low values of Δ_f.

Definition 5. For an n-variable Boolean function f, the transparency order in [10] can be viewed as

$$TO(f) = 1 - \frac{1}{2^n(2^n - 1)} \sum_{\alpha \in \mathbb{F}_2^{n*}} \left| C_f(\alpha) \right|. \tag{7}$$

2.1. Rotation Symmetric Boolean Functions. Let $x_i \in \{0,1\}$ for $1 \le i \le n$. For $1 \le k \le n$ [13], we define

$$\begin{aligned}\rho_n^k(x_i) &= x_{i+k}, \text{ if } i+k \le n \\ &= x_{i+k-n}, \text{ if } i+k > n.\end{aligned} \tag{8}$$

Let $(x_1, x_2, \ldots, x_{n-1}, x_n) \in V_n$. We can extend the definition of ρ_n^k to n-tuples as

$$\rho_n^k(x_1, x_2, \ldots, x_n) = \left(\rho_n^k(x_1), \rho_n^k(x_2), \ldots, \rho_n^k(x_n)\right). \tag{9}$$

Definition 6. A Boolean function f is called rotation symmetric if for each input $(x_1, \ldots, x_n) \in \{0,1\}^n$,

$$f\left(\rho_n^k(x_1, \ldots, x_n)\right) = f(x_1, \ldots, x_n), 1 \le k \le n. \tag{10}$$

Definition 7. An orbit $G_n(\Lambda_{n,i})$ is identified by the representative element $\Lambda_{n,i}$ which is the lexicographically first element of the i-th orbit and $i \in \{0, 1, \ldots, g_n - 1\}$ [14]. Accordingly, we can use the simplified truth table to represent an RSBF f. Its form as

$$\left[f\left(\Lambda_{n,0}\right), f\left(\Lambda_{n,1}\right), \ldots, f\left(\Lambda_{n,g_n-1}\right)\right], \tag{11}$$

which is called the rotation symmetric truth table (RSTT). The length of RSTT is expressed as g_n.

Definition 8. The class of DSBFs [?], a subset of the RSBF class, is invariant under the action of the dihedral group denoted by D_n. In addition to the (left) i-cyclic shift operator ρ_n^i on n-tuples, which is defined previously, the dihedral group D_n also includes the reflection operator $\tau_n(x_0, x_1, \ldots, x_{n-1}) = (x_{n-1}, \ldots, x_1, x_0)$. The $2n$ permutations of D_n are then defined as

$$\left\{\rho_n^1, \rho_n^2, \ldots, \rho_{n-1}^n, \rho_n^n, \tau_n\rho_n^1, \tau_n\rho_n^2, \ldots, \tau_n\rho_{n-1}^n, \tau_n\rho_n^n\right\}. \tag{12}$$

Similar to RSBF, we use d_n to represent the truth table length of DSBF.

2.2. Accelerated Calculation. Furthermore, we introduce an important matrix $\mathscr{A}$ from [7] for analyzing Walsh spectra and accelerated calculation, which will be applied in our search. The matrix is defined as

$$\mathscr{A}_{i,j} = \sum_{x \in G_n\left(\Lambda_{n,i}\right)} (-1)^{x \cdot \Lambda_{n,j}}. \tag{13}$$

Clearly, the size of $\mathscr{A}$ is $g_n \times g_n$ or $d_n \times d_n$, and the matrix element $\mathscr{A}_{i,0}$ is the size of the i-th orbit. Note that the Walsh spectrum of f can be determined by

$$\mathscr{W}_f\left(\Lambda_{n,j}\right) = \sum_{i=0}^{g_n(d_n)-1} (-1)^{f\left(\Lambda_{n,i}\right)} \mathscr{A}_{i,j}. \tag{14}$$

3. Search Strategy

Our search strategy uses the improved steepest-descent iteration algorithm. Each iteration step has an input Boolean function f and an output Boolean function $f_{\min}$. Each iteration step calculates a cost function in the predefined neighborhood, and the Boolean function with the lowest cost is selected as the iteration output. In the algorithm design, the Boolean function has many properties, which are usually challenging to consider. Therefore, we carefully consider the Boolean function with the best comprehensive properties.

The 1-neighborhood of f is obtained by flipping individual elements of its truth table. For an n-variables Boolean function, the 1-neighborhood consists of 2^n many distinct Boolean functions, each being at the Hamming distance 1 to the original Boolean function.

3.1. New Cost Function. We introduced a new cost function that can obtain the Boolean functions with high nonlinearity, low absolute indicator, and low transparency order. We modified C_f and the cost functions. Hence, we use the sum of quartic power errors as the cost function, which is defined as

$$\text{cost}(f) = \sum_{\omega \in \mathbb{F}_2^n} \left(\mathscr{W}_f^2(\omega) - 2^n\right)^4 - \sum_{\alpha \in \mathbb{F}_2^{n*}} \left|\Delta_f(\alpha)\right|. \tag{15}$$

In the search process, if only 1-bit flipping is adopted and the cost is minimized as the target for selection, after several rounds, the cost value will remain unchanged or two adjacent cost values will be repeated. This situation is also called trapped in a locally optimal solution. If no correction is made, the function becomes stable and no new process is created. The same can be said for heuristic algorithms of other classes. In [6], the authors suggest that the second smallest cost value before this algorithm can be selected as an iteration. The disadvantage of this approach is that it involves backtracking. The spatial complexity involved in reversal is challenging to determine in different situations, and the time complexity will also increase. The algorithm is optimized using the steepest-descent-like iterative algorithm based on a greedy algorithm. A novel steepest-descent dual reset algorithm is proposed, that is, speed up convergence. At the same time, it dramatically avoids the problem that the function stays in a suboptimal solution.

3.2. Dual Threshold. In the search process, if only 1-neighbor mode is adopted and the target is selected with the lowest cost, the cost value remains unchanged after several iterations or the cost value of two adjacent iterations is the same. This is also known as falling into a locally optimal solution. If no correction is made, the function becomes stable and no new solution is created. The same is true for other classes of heuristic algorithms. In the document [5], the authors suggest choosing the second smallest generation value before this algorithm as the iteration. The downside to this approach is that it involves backtracking. The spatial complexity involved in inversion is difficult to determine in different situations and the temporal complexity also increases. We propose a dual threshold parameter and optimize the parameter's selection size according to many experiments. At the same time, the problem of the function

staying in the locally optimal solution is significantly avoided.

3.2.1. The First Threshold. CIVT represents the cost-iteration threshold value. The CIVT size is defined as $\lceil e \times n \rceil$, where $e \approx 2.71828$, and n represents the length of the truth table, including g_n and d_n. CIVT is defined as the integer up of its product. At the same time, we refer to a variable $\mathrm{CITV}_{\mathrm{number}} = 0$ to record the degree of $\cos t[k] = \cos t[k+2]$, that is, the number of times that the $\cos t$ of the first k equals the $\cos t$ of the first $k+2$, where k is an integer. When $\cos t[k] = \cos t[k+2]$ is triggered, $\mathrm{CITV}_{\mathrm{number}} + 1$. When $\mathrm{CITV}_{\mathrm{number}}$ reaches our preset value of CIVT, we will trigger the first threshold, randomly select the position, and reset m consecutive bits. m is selected as $m = 2\times$ variables. As for the selection of m here, after repeated attempts on m, we finally determined that the selection method was the best.

3.2.2. The Second Threshold. When the system triggers the threshold for the first time, it will undoubtedly reset the function in the current state, and due to the limitation of m, the current position will only have a small change. Still, the evolution of the function will not be far beyond our expectations. This way, the optimal local situation can be effectively improved, but the change interval is m bits in a row. So if you think of a significant improvement in a property like nonlinearity, it is impossible.

On top of this, we introduce another counter, $\mathrm{CITV}_{\mathrm{count}}$; also, giving an initial value of 0 causes $\mathrm{CITV}_{\mathrm{count}} + 1$ whenever the first double threshold is triggered. When $\mathrm{CITV}_{\mathrm{count}}$ reaches $100 \times m$, we calculate the probability $p_0 = \exp(-k/N)$, where N is the total number of executions, and k is the current number of executions. When $p > p_0$ is true for a given probability p, this moving method is accepted even if the current cost value is not the optimal solution. The idea is similar to the simulated annealing algorithm, which takes weak solutions. The advantage of this is that when the program is just started, there is a high probability that the diluted solution will be accepted so that the function can jump out of the local optimal with a high chance. When the program is about to end, only the weak solution is accepted with a small probability and the variable state of the current function is preserved. And when the second threshold is triggered, we do not select the continuous bit to reset like the first threshold but select the discrete $\log_2 n$ bit (n is the length of the current truth table) to reset. The effect of this is to increase the degree of dispersion of the current function.

3.3. Algorithmic Process. Our main algorithm flow is shown in Algorithm 1.

4. Experimental Results

In this section, we apply the steepest-descent dual reset algorithm and find the optimal Boolean functions of nonlinearity, autocorrelation, and transparency order in 9, 10, 11, and 12 variables, respectively. We give the following table to compare the experimental results of each variable.

In Table 1, we give three representative results for the Boolean function of 9 variables. We demonstrate the excellence of our results and the effectiveness of our algorithm by comparing the nonlinearity, transparency, and balance. While maintaining a higher nonlinearity, our results have a lower transparency order. We find that the new 9-variable Boolean function f_1 is the optimal transparency order result for the Boolean function with a nonlinearity of 240 based on the dihedral symmetry class. Compared with [15], the algorithm in this paper has good convergence in terms of nonlinearity. Before this paper, the result of a 9-variable Boolean function with a nonlinearity of 242 was obtained only by Kavut. It appeared in [8], published in Information and Computation in 2010, and its results were found in the rotationally symmetric class. This paper not only finds a new 9-variable Boolean function f_2 with nonlinearity 242 and excellent transparency order in the rotationally symmetric class but also finds a new 9-variable Boolean function f_3 with nonlinearity 242 and lower transparency order in the whole space 2^{2^n}. No one has ever published such results.

In [8], the author Kavut found the best nonlinearity of 238 and 239 in two kinds of 9-variable Boolean functions with the order of (6, 5, 1, 4, 5, 4, 1, 3, 6, 8) and (0, 7, 2, 5, 4, 1, 3, 6, 8). In this paper, f_4 and f_5 increase the maximum nonlinearity of these two Boolean functions to 240.

In Table 2, we compare the results in [2, 12, 15, 16], and we find the Boolean functions of high nonlinearity and low transparency in the class of rotational symmetry, represented by f_6 and f_7. Among the equilibrium 10-element Boolean functions, f_7 has the highest nonlinearity among the known results and maintains a good transparency order. For the previous 10-variable Boolean function with a nonlinearity of 490, the lowest transparency order is 0.9864 given in [12]. However, the result of f_6 presented in this paper has lower transparency order compared with that in [12].

The result of an 11-variable Boolean function is shown in Table 3. We present two representative results. Compared with the results given in [12, 15], our search's nonlinearity of the Boolean function obtained is higher. In addition, when the previously known nonlinearity is 990, the lowest transparency order is 0.9872. Still, our result f_8 has a lower transparency order with the same nonlinearity, which is also due to the high convergence of our designed algorithm. According to the conclusion of [10], Boolean functions with high nonlinearity usually have higher transparency order. However, after applying our improved search algorithm, this paper finds the equilibrium 11-element Boolean function f_9, which has lower transparency order while maintaining higher nonlinearity, compared with the result in [12]. The result f_9 is the best among the general equilibrium 11-variable Boolean functions considering the nonlinearity and transparency order.

Table 4 compares the results of the 12-variable Boolean function with [14, 17]. The result f_{10} is that we obtain

Require: a random Boolean function
Ensure: excellent character Boolean function
$f \leftarrow f_{\text{initial}}$
$i \leftarrow 0$
$j, k \leftarrow 0$
while $k < N$ **do**
  while $j <$ length of f **do**
    Flip the j^{th} bit in f
    //Use the array A_cost[] to record the cost value of each time
  end while
  Select the smallest cost value from A_cost[] and record it as $\cos t_{\text{new}}$
$f_{\text{new}} \leftarrow f_{\min}$//Update Boolean functions
  if $\text{A\_cost}[k]_{\min} = \text{A\_cost}[k+2]_{\min}$, **then**
    CITV_number$\leftarrow$CITV_number + 1
    //If the cost value of the k times is equal to the $k+2$ times, the first double threshold counter plus 1
  end if
  if CITV_number == CITV, **then**
    reset.rando m ()//The first threshold reset
    CITV_count$\leftarrow$CITV_count + 1//Second threshold counter plus 1
  end if
  if CITV_count = $100 \times m$, **then**
    if $p > p_{0,}$ **then**
      reset.count ()//The second threshold reset
    end if
  end if
end while

Algorithm 1: Dual threshold algorithm for search excellent character Boolean function.

Table 1: Results in 9-variable.

	nl	ac	To	Balanced
Kavut and Yücel [15]	236	32	—	Yes
Xu and Wang [12]	240	—	0.9617	Yes
Kavut and Yücel [8]	242	32	—	No
Our result f_1	240	72	0.9611	Yes
Our result f_2	242	32	0.9832	No
Our result f_3	242	48	0.9828	No
Our result f_4	240	160	0.9870	Yes
Our result f_5	240	128	0.9845	Yes

Table 2: Results in 10-variable.

	nl	ac	To	Balanced
Kavut and Yücel [15]	486	56	—	Yes
Behera and Gangopadhyay [16]	488	40	—	Yes
Sarkar and Maitra [2]	488	40	—	Yes
Xu and Wang [12]	490	—	0.9864	Yes
Our result f_6	490	80	0.9846	Yes
Our result f_7	492	40	0.9886	Yes

Table 3: Results in 11-variable.

	nl	ac	To	Balanced
Kavut and Yücel [15]	984	80	—	Yes
Xu and Wang [12]	990	—	0.9872	Yes
Our result f_8	990	88	0.9869	Yes
Our result f_9	992	128	0.9856	Yes

Table 4: Results in 12-variable.

	nl	ac	To	Balanced
Clark et al. [17]	1990	140	—	Yes
Behera and Gangopadhyay [16]	1996	—	—	Yes
Our result f_{10}	2000	136	0.9923	Yes

a balanced Boolean function with a nonlinearity of 2000 and an autocorrelation of 136 and the transparency order is the lowest known.

5. Conclusion

The space of Boolean functions is vast, and the area of Boolean functions of n variables is 2^{2^n}; so, it is not feasible to search exhaustively. An excellent approach is searching in a narrowed space using rotational symmetry classes. Under this circumstance, designing a workable search method for colleges and universities is also crucial. With our improved steepest descent method, good results can be obtained in rotationally symmetric class Boolean functions and entire space Boolean functions.

In the 9, 10, 11, and 12 variables, we have obtained Boolean functions with excellent properties, many of which are Boolean functions with the best comprehensive properties, such as the comparison results in Section 4.

We also used the cost function given in the experiment in [14] and compared it. After the investigation, we found that the difference in the convergence effect of different costs is

vast when using our same algorithm. Therefore, designing a more rational cost function is also an urgent problem to solve in the heuristic search of Boolean functions. Although our cost has achieved good results in the application, we still hope to optimize the cost function further to make its computational complexity lower and the convergence more accurate.

Appendix

Some results of our experiment are as follows:

Our result 1: 9-variable, nl(f) = 240, ac(f) = 72, To(f) = 0.9611

427f 4c88 5397 f7f6 4128 b109 9c18 cc0f 1635 2ae7 f934 77f1 e586 65a6 9696 62cd 250a 7d54 3afb ce09 cda1 7c57 1818 cc21 9a54 e70b 1b45 ee0b e41b f10b 0f7b 97d5

Our result 2: 9-variable, nl(f) = 242, ac(f) = 32, To(f) = 0.9832

3340 b6a1 1821 f196 42a8 5e2b 7e2f 3c3c b65f a0d9 5ec9 db1e ab2b db36 6618 5ae0 087f 5fe6 e075 7106 212f c918 754c 40e8 a1bc cbfa 7140 32a8 9614 56e0 66e8 a801

Our result 3: 9-variable, nl(f) = 242, ac(f) = 48, To(f) = 0.9828

6da2 cd1d b0b7 47b7 5b3d 90b8 62fe fa29 368e 5ff6 d744 8ec1 7a98 82d0 c40c a201 7acd b4a3 cb61 995e fabc 92f2 e7cc aa40 fe9d daf0 c64c ba08 935b 22cb 5d09 f1df

Our result 4: 9-variable in the permutation of (6, 5, 1, 4, 7, 2, 3, 0, 8), nl(f) = 240, ac(f) = 160, To(f) = 0.9870

e83d 6d1b 7927 5697 3d17 1b92 2786 97a9 3d17 1b92 2786 97a9 17c2 92e4 86d8 a968 6a16 1a53 2647 56c1 1695 53e5 47d9 c1a9 1695 53e5 47d9 c1a9 95e9 e5ac d9b8 a93e

Our result 5: 9-variable in the permutation of (0, 7, 2, 5, 4, 1, 3, 6, 8), nl(f) = 240, ac(f) = 128, To(f) = 0.9845

c200 c23c 0b8d 3b8c bcff a995 f031 9556 23b1 2fb0 01d5 c0c3 cc0d 9556 fd14 566a 0216 546b 19a7 623b 5469 abab 621b acf1 259b 4a2f 6b7f bfbc 4a27 b8cd 97bc bd15

Our result 6: 10-variable, nl(f) = 490, ac(f) = 80, To(f) = 0.9846

abfc ddc7 d180 d249 d530 f326 c46b 46b1 d104 2823 9c2d 7e1b d617 1aad 420a bc75 9520 6346 3af6 7b2d e193 2f94 08df 64f9 c04b 2149 24ff ae84 462a 76ea bc96 1845 b451 7a63 5e78 030f 78ab c94f 4de9 2ac5 da30 e479 3f9c a507 23b3 8588 0a42 cce1 c763 46b9 6b31 13a1 6e07 dc99 bf9a a203 130f 3ffb 094f deab f9d7 b14b 75f3 5755

Our result 7: 10-variable, nl(f) = 492, ac(f) = 40, To(f) = 0.9886

121d 53f6 365a ff38 5b3d 26d8 eaef 4ad0 369f 1be6 4828 e2c5 f8c8 e9ab 21d9 e314 1b6c 96ba 538b a829 3480 4c90 e90d b033 ffd1 e085 b893 cc8f 5847 a282 b95f 1274 47de 6cf4 d72c 8b8c 675a 808b c891 1d96 5a64 d101 60a4 9345 fd83 45a3 cb00 5f4e beae f212 bc01 8576 cad1 825b f1b1 90fa 62c4 613e 8d0d d54c 8a96 33fa 065c 2e20

Our result 8: 11-variable, nl(f) = 990, ac(f) = 88, To(f) = 0.9869

aaee cadf 93ab c58c b52c fba8 c601 a2d3 fc05 3bd7 c9f8 aef6 c25b 6675 ef7b c479 d883 3755 3ce8 d00c 93e5 9da3 bb9a c91a 866a 54ec 4e1b 4811 cf99 5cbc 9743 48b4 91a7 f268 790d 0544 2dc3 9ea2 c562 67c3 b03 d ce44 b5d4 fe78 adbc a4ba c6a4 31ea b20b 1fff 1007 8fd3 028a 65f8 13b7 2030 96c8 e5a0 1183 fcc2 f10d 427d 16a3 b803 a025 fb18 d87a 0ba7 49f5 3684 2705 1707 6bd5 c268 f1db fe3e 8240 0f2f 1f4c d32c f962 7990 93ca 1213 ed00 d446 9c8e 08f2 ebd5 e997 aa16 fdae c21a fb42 7925 cbeb a97e 66b9 35c8 99d9 3522 774c a299 d078 727e f2ef 1a40 c8a6 353d ac58 2e72 7933 b40f d2f7 da50 af66 6461 e228 dd86 867f c965 6690 032a 5885 310e ba69 f8e3 633c

Our result 9: 11-variable, nl(f) = 992, ac(f) = 128, To(f) = 0.9856

eda2 9d5c c7f3 37f4 e02b bf1e 0e3e af61 a854 59cb 9fbf 57a8 05fd 5ea8 9cee 3c43 9d81 7265 3287 e19e d7ab 9aaa 326e c9d4 5477 eeb3 32a8 99c5 c6a0 a9fc 5ee1 701b c7e7 8556 3a5c 7c76 4b19 c03b fc03 c3e8 f73f 88ca d29d cc8c 5e08 6ca8 e183 e720 3734 6e6a a9a9 da1b 1f58 cc80 c3c3 f032 f578 9914 c983 aae4 66e9 bd57 2a41 128b b07e bd7e 8176 2639 0ed9 77f0 3fe0 6a69 71cb 4396 f140 4a9e efa5 405b a01e e9c5 bb3e 4eee 9585 e198 f708 c7a2 e0a0 91b0 76a8 11c5 7ca0 99c4 fc56 910a a97f 0941 4e7e 1e25 2dfc 7889 d996 d886 b38 d 069b 56aa 7394 f0b0 8054 e05b e05a ee50 1f19 ae63 3ec1 d6c2 5325 e0d6 c40e 8d8c ac75 2969 e8d3 8ea2 322f 198d 2043 5309 d1de

Our result 10: 12-variable, nl(f) = 2000, ac(f) = 136, To(f) = 0.9923

2ede 819e a331 b5de 00d7 92db 06ee 2a65 99d9 7bb6 1b80 2a56 221f 9e8a 2baf 1f44 a5e5 81b4 5da8 bc4e 8b12 5c88 8044 aaa4 95d4 8a77 1b34 5c14 2ea9 bbd9 31d9 4352 eb00 cb54 e371 f842 ea7e 114d 1769 bc34 0953 8f85 af6d 0c18 f676 4307 bbef ee43 b140 9113 b2af 491c da56 97a8 af29 cae8 d474 440b 5602 3b5e 7920 81f5 426d 552f dbad 6726 83a9 4142 2492 a69b 2648 e880 2100 e630 8fde ac6b 754c 1ee0 a996 2917 37a0 113d a6dc a344 1426 f17f cd2d 9e1c 73b4 f3b1 bcd6 d8a2 a9fd 8e99 dbdf 0779 fd25 5337 b035 7568 43c1 1462 ec4a 9e38 2a05 bef0 1ee6 5408 eb89 6fb4 d6fe 9ee7 d546 4d02 4343 73b8 aaf1 8c91 8256 ab64 f21a c4c8 598a 73fb 437b 4f85 5514 2adc 84bc bfd4 5a18 7f5e 0883 055b adca b9d0 91bc 5ac0 41b5 5f57 2b5a 03b3 8bf7 f736 3a34 3762 db0e 2c33 4866 6e71 4439 f103 f2ff e93c 9b25 64dd bbf4 b50b 7ab5 2159 7918 bb32 7170 7dd5 45f1 6b7c 50c2 bcbd dafd 84b0 378a e633 8380 3b95 b58f 7597 597d ed52 cc3d fd31 5639 3fb5 3b1c 44c1 501a 337e 5870 5b4f 91fd 8588 3719 5ca4 8cd4 6e54 5468 6d09 039d c2aa b6bf b55c ecd2 788f dff8 f084 9ac3 06fa f4db 6db7 7efe 7655 e8cb 9d32 9a71 25f0 bafd d95d 7516 0d5f b022 03fc

910b 998e f59a df48 8404 421b 42d5 322b f996 a997 e386 471d 4445 23de 1869 1a8a a36a 154b fbf8 5e52 882f 25eb 8342 c7e2 fe4b 1c15 b2d3 7646 a993 b212 bc32 18bf 4011 3112 2ffe 94c2.

Data Availability

The data that support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 62172230 and 62262018) and the National Natural Science Foundation of Jiangsu Province (no. BK20201369).

References

[1] T. W. Cusick and P. Stanica, *Cryptographic Boolean Functions and Applications*, Academic Press, Cambridge, MA, USA, 2017.

[2] P. Sarkar and S. Maitra, "Nonlinearity bounds and constructions of resilient boolean functions," in *Proceedings of the Advances in Cryptology—CRYPTO 2000: 20th Annual International Cryptology Conference*, pp. 515–532, Santa Barbara, CA, USA, August 2000.

[3] P. K. Behera and S. Gangopadhyay, "An improved hybrid genetic algorithm to construct balanced boolean function with optimal cryptographic properties," *Evolutionary Intelligence*, vol. 15, no. 1, pp. 639–653, 2022.

[4] Y. Wang, G. Gao, and Q. Yuan, "Searching for cryptographically significant rotation symmetric boolean functions by designing heuristic algorithms," *Security and Communication Networks*, vol. 2022, Article ID 8188533, 6 pages, 2022.

[5] J. Du, Q. Wen, J. Zhang, and S. Pang, "Constructions of resilient rotation symmetric boolean functions on given number of variables," *IET Information Security*, vol. 8, no. 5, pp. 265–272, 2014.

[6] S. Kavut, S. Maitra, and M. D. Yucel, "Search for boolean functions with excellent profiles in the rotation symmetric class," *IEEE Transactions on Information Theory*, vol. 53, no. 5, pp. 1743–1751, 2007.

[7] S. Maity and S. Maitra, "Minimum distance between bent and 1-resilient boolean functions," in *Proceedings of the International Workshop on Fast Software Encryption*, pp. 143–160, Springer, Istanbul, Turkey, January 2004.

[8] S. Kavut and M. D. Yücel, "9-Variable boolean functions with nonlinearity 242 in the generalized rotation symmetric class," *Information and Computation*, vol. 208, no. 4, pp. 341–350, 2010.

[9] K. Chakraborty, S. Sarkar, S. Maitra, B. Mazumdar, D. Mukhopadhyay, and E. Prouff, "Redefining the transparency order," *Designs, Codes and Cryptography*, vol. 82, no. 1-2, pp. 95–115, 2017.

[10] Q. Wang and P. Stănică, "Transparency order for boolean functions: analysis and construction," *Designs, Codes and Cryptography*, vol. 87, no. 9, pp. 2043–2059, 2019.

[11] S. Kavut, S. Maitra, and D. Tang, "Construction and search of balanced boolean functions on even number of variables towards excellent autocorrelation profile," *Designs, Codes and Cryptography*, vol. 87, no. 2-3, pp. 261–276, 2019.

[12] Y. Xu and Q. Wang, "Searching for highly nonlinear dpa-resistant balanced boolean functions in the rotation symmetric class," in *Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1212–1216, Paris, France, July 2019.

[13] S. Sarkar and S. Maitra, "Idempotents in the neighbourhood of patterson-wiedemann functions having walsh spectra zeros," *Designs Codes and Cryptography*, vol. 49, pp. 95–103, 2008.

[14] S. Kavut, S. Maitra, and M. D. Yücel, "There Exist Boolean Functions on N (Odd) Variables Having Nonlinearity >rbin2n-1-2(n-1/2)if and Only if N>rbin7," Cryptology ePrint Archive, 2006, https://eprint.iacr.org/2006/181.

[15] S. Kavut and M. D. Yücel, "Improved cost function in the design of boolean functions satisfying multiple criteria," in *Progress in Cryptology - INDOCRYPT 2003*, T. Johansson and S. Maitra, Eds., pp. 121–134, Springer, Berlin, Germany, 2003.

[16] P. K. Behera and S. Gangopadhyay, "Analysis of cost function using genetic algorithm to construct balanced boolean function," in *Proceedings of the TENCON 2018 - 2018 IEEE Region 10 Conference*, pp. 1445–1450, Jeju, Republic of Korea, October 2018.

[17] J. A. Clark, J. L. Jacob, S. Stepney, S. Maitra, and W. Millan, "Evolving boolean functions satisfying multiple criteria," in *Progress in Cryptology — INDOCRYPT 2002*, A. Menezes and P. Sarkar, Eds., pp. 246–259, Springer, Berlin, Germany, 2002.