Evolutionary approach to S-box generation: Optimizing nonlinear substitutions in symmetric ciphers

Oleksandr Kuznetsov[1,2,3,†], Nikolay Poluyanenko[2,†], Emanuele Frontoni[3,†], Marco Arnesano[1,†] and Oleksii Smirnov[4,*,†]

[1] *eCampus University, 10 Via Isimbardi, 22060 Novedrate, Italy*
[2] *V. N. Karazin Kharkiv National University, 4 Svobody sq., 61022 Kharkiv, Ukraine*
[3] *University of Macerata, 30/32 Via Crescimbeni, 62100 Macerata, Italy*
[4] *Central Ukrainian National Technical University, 8 University ave., 25006 Kropyvnytskyi, Ukraine*

Abstract

This study explores the application of genetic algorithms in generating highly nonlinear substitution boxes (S-boxes) for symmetric key cryptography. We present a novel implementation that combines a genetic algorithm with the Walsh-Hadamard Spectrum (WHS) cost function to produce 8×8 S-boxes with a nonlinearity of 104. Our approach achieves performance parity with the best-known methods, requiring an average of 49,399 iterations with a 100% success rate. The study demonstrates significant improvements over earlier genetic algorithm implementations in this field, reducing iteration counts by orders of magnitude. By achieving equivalent performance through a different algorithmic approach, our work expands the toolkit available to cryptographers and highlights the potential of genetic methods in cryptographic primitive generation. The adaptability and parallelization potential of genetic algorithms suggests promising avenues for future research in S-box generation, potentially leading to more robust, efficient, and innovative cryptographic systems. Our findings contribute to the ongoing evolution of symmetric key cryptography, offering new perspectives on optimizing critical components of secure communication systems.

Keywords

S-box generation, genetic algorithms, nonlinear substitutions, Walsh-Hadamard spectrum, cryptographic primitives, heuristic optimization, cryptographic strength

CEUR-WS.org/Vol-3829/paper1.pdf

1. Introduction

The realm of digital security is in a constant state of evolution, with symmetric key cryptography serving as a fundamental pillar in the architecture of secure communication systems [1–3]. At the core of many symmetric encryption algorithms lie Substitution boxes (S-boxes) [4], which play a pivotal role in establishing the nonlinear components essential for robust encryption [5, 6]. These S-boxes are critical in creating the confusion and diffusion properties that Claude Shannon identified as crucial for secure ciphers [7, 8].

The cryptographic strength of an S-box is multifaceted, encompassing several key indicators [9]. Nonlinearity, which quantifies an S-box's resistance to linear cryptanalysis, stands as a primary measure. For 8×8 S-boxes, commonly employed in modern ciphers, achieving a nonlinearity of 104 represents a significant benchmark [10–12]. However, other properties such as differential uniformity, algebraic degree, and algebraic immunity also play crucial roles in determining an S-box's overall cryptographic efficacy [13, 14].

While algebraically constructed S-boxes, such as the one used in the Advanced Encryption Standard (AES) with its optimal nonlinearity of 112 [15], might seem ideal, they are not without vulnerabilities. The presence of inherent algebraic structures in such S-boxes can create potential weaknesses, making them susceptible to algebraic cryptanalysis [16–18]. This vulnerability underscores the need for randomly generated S-boxes that lack hidden algebraic structures, thereby enhancing resistance against sophisticated cryptanalytic techniques [19–21].

The generation of cryptographically robust S-boxes presents a significant computational challenge. The vast search space of possible configurations for 8×8 S-boxes is estimated at $2^8!$ (approximately 10^{506}), renders exhaustive search methods impractical. This complexity has driven research towards heuristic approaches for S-box generation [22–24]. Methods such as simulated annealing, hill climbing, and genetic algorithms have shown promise in navigating this expansive solution space efficiently [25].

CQPC-2024: Classic, Quantum, and Post-Quantum Cryptography, August 6, 2024, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
oleksandr.kuznetsov@uniecampus.it (O. Kuznetsov); nlfsr01@gmail.com (N. Poluyanenko); emanuele.frontoni@unimc.it (E. Frontoni); marco.arnesano@uniecampus.it (M. Arnesano); dr.smirnovoa@gmail.com (O. Smirnov)

0000-0003-2331-6326 (O. Kuznetsov); 0000-0001-9386-2547 (N. Poluyanenko); 0000-0002-8893-9244 (E. Frontoni); 0000-0003-1700-3075 (M. Arnesano); 0000-0001-9543-874X (O. Smirnov)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Recent advances in heuristic S-box generation have made significant advances. Researchers have studied various cost functions, including the Walsh-Hadamard spectrum (WHS) function [23, 26], the Picek cost function (PCF) [10], improved Walsh-Hadamard spectrum-based cost functions (WCF) [10, 27], and two new extended cost functions (ECF and WCFS) [6, 28, 29] in conjunction with different search algorithms [10, 22]. These efforts have progressively reduced the computational cost of generating highly nonlinear S-boxes, with some methods achieving the target nonlinearity of 104 in fewer than 100,000 iterations.

Despite these advancements, there remains a gap in understanding the full potential of genetic algorithms in this domain. While genetic approaches have been applied to S-box generation, their performance in comparison to other heuristic methods, particularly in terms of consistency and efficiency in generating S-boxes with optimal cryptographic properties, remains an area ripe for exploration.

Our study aims to address this gap by presenting a comprehensive investigation into the application of genetic algorithms for generating 8×8 S-boxes with a nonlinearity of 104. We explore the synergy between genetic algorithms and the WHS cost function, aiming to match or surpass the efficiency of existing methods while leveraging the inherent advantages of evolutionary approaches, such as adaptability and the potential for parallelization.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive review of the literature, detailing the evolution of S-box generation techniques and the current state of the art. Section 3 offers a background on S-boxes, their cryptographic properties, and the theoretical foundations underpinning their design. Section 4 delineates our methodology and experimental setup, including the specifics of our genetic algorithm implementation and evaluation criteria. Section 5 presents our results and a detailed discussion, comparing our findings with existing methods and analyzing their implications. Finally, Section 6 concludes the paper, summarizing our key findings and outlining promising directions for future research in this critical area of cryptographic system design.

2. Literature review

The design and generation of cryptographically strong S-boxes have been subjects of intensive research in the field of symmetric key cryptography. This section provides a comprehensive review of the existing literature, focusing on various approaches to S-box generation and their cryptographic properties.

Algebraic constructions of S-boxes, such as those based on finite field inversion used in the Advanced Encryption Standard (AES) [15, 30, 31], have been widely studied. However, as Bard (2009) [16] and Courtois and Bard (2007) [17] point out, these constructions may be vulnerable to algebraic attacks due to their inherent mathematical structure. This vulnerability has led to increased interest in generating S-boxes with more complex algebraic structures [32].

Heuristic approaches have gained significant traction in recent years. Clark et al. (2005) [26] introduced a simulated annealing approach for S-box generation [23], demonstrating its effectiveness in producing S-boxes with high nonlinearity. Building on this work, Souravlias et al. (2017) [33] proposed an algorithm portfolio approach combining simulated annealing and tabu search, showing improved results under limited time budgets.

Genetic algorithms have also been explored for S-box generation [24, 34]. Tesar (2010) [35] combined a genetic algorithm with a tree search method, generating 8×8 S-boxes with nonlinearity up to 104. Picek et al. (2016) [11] presented a novel cost function for evolving S-boxes, achieving significant improvements in both speed and quality of results compared to previous approaches.

Ivanov et al. (2016a, 2016b) [36, 37] introduced an innovative approach using a modified immune algorithm combined with hill climbing, rapidly generating large sets of highly nonlinear bijective S-boxes. Their work demonstrated the potential of hybrid approaches in S-box generation.

Recent advancements have focused on improving specific cryptographic properties. Rodinko et al. (2017) [38] optimized a method for generating high nonlinear S-boxes, achieving nonlinearity of 104, algebraic immunity of 3, and 8-uniformity within reasonable computational time. Freyre Echevarría and Martínez Díaz (2020) [27] proposed a new cost function specifically designed to improve the nonlinearity of bijective S-boxes.

The importance of multiple cryptographic criteria has been emphasized in recent literature. Freyre-Echevarría et al. (2020) [10] introduced an external parameter-independent cost function for evolving bijective S-boxes, considering both nonlinearity and other important properties. Their work highlighted the need for balanced optimization across multiple cryptographic criteria.

More recent studies have explored novel approaches to S-box generation. Artuğer and Özkaynak (2024) [39] proposed a post-processing approach to improve the nonlinearity of chaos-based S-boxes, addressing a longstanding challenge in this area. Haider et al. (2024) [40] introduced an S-box generator based on elliptic curves, offering a balance between randomization and optimization with minimal computation time.

The application of S-boxes in specific cryptographic contexts has also been a focus of recent research. Jamal et al. (2024) [41] developed a region of interest-based medical image encryption technique using chaotic S-boxes, demonstrating the practical applications of advanced S-box designs in specialized domains.

Emerging threats and the need for enhanced security have led to new considerations in S-box design. Fahd et al. (2024) [42] examined the reality of backdoored S-boxes, highlighting the importance of thorough cryptanalysis and the potential vulnerabilities in S-box structures.

In conclusion, the literature reveals a trend towards more sophisticated, multi-criteria optimization approaches in S-box generation. While significant progress has been made in achieving high nonlinearity and other desirable properties, there remains a need for methods that can consistently produce S-boxes with optimal cryptographic characteristics while balancing computational efficiency and resistance to emerging cryptanalytic techniques.

3. Background

Symmetric cryptography forms the backbone of secure communication in the digital age. At the heart of many symmetric ciphers lie Substitution boxes (S-boxes), nonlinear components crucial for ensuring the security and robustness of these cryptographic systems. This section provides a comprehensive overview of S-boxes, their role in symmetric cryptography, and the application of genetic algorithms in their optimization.

3.1. S-boxes in symmetric cryptography

Substitution boxes (S-boxes) are fundamental components in symmetric-key algorithms, serving as the primary source of nonlinearity [7, 8]. An S-box is essentially a vectorial Boolean function that maps a fixed number of input bits to a fixed number of output bits. Formally, an $n \times m$ S-box can be defined as [9]:

$$S : \mathrm{F}_2^n \to \mathrm{F}_2^m,$$

where F_2^n and F_2^m are vector spaces over the Galois field $GF(2)$ with dimensions n and m, respectively.

The cryptographic strength of an S-box is determined by several critical properties [10]:

1) Nonlinearity: A measure of the distance between the S-box and the set of all affine functions. For an $n \times n$ S-box, the nonlinearity is defined as:

$$NL(S) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathrm{F}_2^n, b \in \mathrm{F}_2^n \setminus 0} \left| \sum_{x \in \mathrm{F}_2^n} (-1)^{b \cdot S(x) \oplus a \cdot x} \right|,$$

where $\cdot$ denotes the dot product and $\oplus$ represents bitwise XOR.

2) Differential uniformity: Quantifies the uniformity of output differences when the input is changed. The differential uniformity δ is given by:

$$\delta = \max_{a \neq 0, b} | x \in \mathrm{F}_2^n : S(x) \oplus S(x \oplus a) = b |.$$

3) Algebraic degree: The highest degree among the component Boolean functions of S. For an $n \times m$ S-box, the algebraic degree is:

$$deg(S) = \max_{v \in \mathrm{F}_2^m \setminus 0} deg(v \cdot S).$$

4) Balancedness: An S-box is balanced if each output occurs with equal probability when the input is uniformly distributed.

5) Algebraic Immunity [43]: A measure of resistance against algebraic attacks. For an S-box $S: F_2^n \to F_2^m$, the algebraic immunity is defined as:

$$AI(S) = \min \left\{ \deg(P), P \in I(S) \right\},$$

where $I(S)$ is the ideal generated by the polynomials representing the S-box:

$$I(S) = \begin{pmatrix} y_1 - f_1(x_1, x_2, ..., x_n), \\ y_2 - f_2(x_1, x_2, ..., x_n), \\ ..., \\ y_m - f_m(x_1, x_2, ..., x_n) \end{pmatrix}.$$

The algebraic immunity can be computed by constructing the minimal reduced Gröbner basis of the ideal $I(S)$ using the degree reverse lexicographic (degrevlex) ordering, and finding the polynomial of minimum degree in this basis.

These properties collectively contribute to the S-box's ability to resist various cryptanalytic attacks, including differential, linear, and algebraic cryptanalysis. The concept of algebraic immunity for S-boxes, as introduced by Faugère and Perret, provides a crucial measure of resistance against algebraic attacks, which attempt to express the cipher as a system of low-degree multivariate polynomial equations.

The relationship between the algebraic immunity of an S-box and that of Boolean functions can be established through the following construction. Consider a Boolean function $f_S: F_2^{n+m} \to F_2$ defined as [44, 45]:

$$f_S(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) = \\ = \begin{cases} 1, & \text{if } \forall i, j : f_i(x_1, x_2, ..., x_n) = y_j; \\ 0, & \text{if } \exists i, j : f_i(x_1, x_2, ..., x_n) \neq y_j. \end{cases}$$

The algebraic immunity of the S-box S is then equivalent to the minimum degree of non-zero polynomials in the annihilator of f_S:

$$AI(S) = \min deg(g) \mid g \in Ann(f_S).$$

This formulation provides a bridge between the algebraic immunity of vectorial Boolean functions (S-boxes) and that of single Boolean functions, unifying the concept across different cryptographic primitives.

3.2. Importance of S-boxes in modern ciphers and the need for randomness

S-boxes play a pivotal role in ensuring the security of symmetric ciphers by introducing nonlinearity and complexity into the encryption process [7]. They are employed in widely-used algorithms such as the Advanced Encryption Standard (AES) [15], where the SubBytes operation relies on a carefully designed 8×8 S-box. However, the increasing sophistication of cryptanalytic techniques has necessitated a reevaluation of traditional S-box design methods.

While algebraically constructed S-boxes, such as those used in AES (based on finite field inverses) [30, 31], offer certain advantages in terms of implementation efficiency and some cryptographic properties, they may fall short in terms of algebraic immunity [43]. The structured nature of these S-boxes can potentially lead to vulnerabilities against algebraic attacks, which have gained significant attention in recent years [16, 17].

Algebraic attacks exploit the possibility of expressing the cipher as a system of low-degree multivariate polynomial equations [17, 18]. The complexity of solving such systems is closely related to the algebraic immunity of the S-box [43]. A low algebraic immunity allows for a simpler representation of the cipher, potentially reducing the computational effort required for cryptanalysis [44, 45]. This vulnerability has prompted researchers to explore alternative methods for S-box generation that prioritize high algebraic immunity alongside other critical properties.

To address these concerns, there is growing interest in the cryptographic community in random or pseudo-random S-boxes [24, 46, 47]. These S-boxes, generated through heuristic methods, offer several advantages:

- Higher algebraic immunity: Random S-boxes are less likely to exhibit algebraic structures that can be exploited in attacks, potentially leading to higher algebraic immunity values.
- Resistance to specialized attacks: Algebraically constructed S-boxes might be vulnerable to attacks tailored to their specific structure. Random S-boxes, lacking such predictable structures, can offer better protection against these targeted attacks.
- Flexibility in design: Heuristic methods allow for the optimization of multiple cryptographic criteria simultaneously, enabling a more balanced approach to S-box design.

Adaptability to evolving threat models: As new cryptanalytic techniques emerge, the criteria for S-box generation can be adjusted more easily with heuristic methods compared to algebraic constructions.

Various heuristic approaches have been proposed for generating high-quality random S-boxes, including:

- Simulated Annealing [23, 26, 33, 48]: This method mimics the physical process of annealing in metallurgy, gradually "cooling" the system to find an optimal configuration. It has shown promise in generating S-boxes with good cryptographic properties.
- Hill Climbing [6, 10, 36, 49, 50]: A local search algorithm that iteratively makes small improvements to a candidate solution. This approach can be effective in fine-tuning S-box properties.
- Genetic Algorithms [10, 35, 37, 51]: Evolutionary approaches that mimic natural selection to evolve a population of S-boxes towards desired properties. These algorithms have demonstrated the ability to generate S-boxes with excellent cryptographic characteristics, including high algebraic immunity.

In this work, we focus on genetic algorithms due to their ability to efficiently explore large search spaces and handle multi-objective optimization problems. Genetic algorithms offer a promising approach to generating S-boxes that balance multiple cryptographic criteria, including high algebraic immunity, nonlinearity, and differential uniformity.

3.3. Genetic algorithms for S-box generation

Genetic Algorithms (GAs) are stochastic optimization techniques inspired by the principles of natural selection and evolution [52, 53]. They operate on a population of potential solutions, evolving them over successive generations to improve their fitness concerning a defined objective function. In the context of S-box generation, GAs offer a powerful and flexible approach to optimizing multiple cryptographic properties simultaneously [10, 37, 54].

The fundamental principle of GAs is to emulate the process of natural selection, where the fittest individuals are more likely to survive and reproduce, passing their beneficial traits to future generations [52, 53]. In the case of S-box generation, an "individual" represents a candidate S-box, and its "fitness" is determined by how well it satisfies the desired cryptographic properties.

The basic structure of a GA includes the following components [54, 55]:

- Chromosome representation: Encoding of potential solutions (S-boxes).
- Fitness function: Evaluates the quality of solutions based on cryptographic criteria.
- Selection mechanism: Chooses individuals for reproduction.
- Genetic operators: Crossover and mutation to create new solutions.
- Termination criteria: Conditions for ending the evolutionary process.

A general pseudocode for a Genetic Algorithm applied to S-box generation can be described as follows:

Algorithm: *Genetic algorithm for S-box generation*
Input: Population size N, number of generations G, crossover rate p_c, mutation rate p_m;
Output: Optimized S-box;

1. Initialize population P of N random S-boxes
2. For $g = 1$ to G do
3. Evaluate the fitness of each S-box in P
4. Select parents for reproduction using tournament selection
5. Create new population P' through crossover and mutation:
6. For $i = 1$ to $N/2$ do
7. Select two parents p_1 and p_2 from P
8. If random (0,1) < p_c then
9. (c_1, c_2) = Crossover(p_1, p_2)
10. Else
11. $(c_1, c_2) = (p_1, p_2)$
12. End If
13. Mutate c_1 and c_2 with probability p_m
14. Add c_1 and c_2 to P'
15. End For
16. $P = P'$
17. End For
18. Return the best S-box from P

Key parameters and their roles:

- Population size (N): Determines the diversity of solutions. A larger population allows for broader exploration of the search space but increases computational cost.
- Number of generations (G): Controls the duration of the evolutionary process. More generations

allow for further optimization but may lead to overfitting.

- Crossover rate (p_c): Probability of performing crossover. Higher rates promote the exploration of new solution combinations.
- Mutation rate (p_m): Probability of mutating each bit in a chromosome. Higher rates increase diversity but may disrupt good solutions.

The fitness function is crucial in guiding the evolutionary process towards S-boxes with desired cryptographic properties.

The selection mechanism, often implemented as tournament selection, ensures that fitter individuals have a higher chance of being chosen for reproduction. This process mimics natural selection, where more adapted individuals are more likely to pass on their genes.

Crossover operators for S-boxes must be carefully designed to preserve the bijective property. One approach is to use a permutation-based crossover, where segments of the S-box permutation are exchanged between parents. For example, given two parent S-boxes P_1 and P_2, a two-point crossover might produce offspring C_1 and C_2 as follows:

$$P_1 = (a_1, a_2, ..., a_k \mid a_{k+1}, ..., a_l \mid a_{l+1}, ..., a_n);$$

$$P_2 = (b_1, b_2, ..., b_k \mid b_{k+1}, ..., b_l \mid b_{l+1}, ..., b_n);$$

$$C_1 = (a_1, a_2, ..., a_k \mid b_{k+1}, ..., b_l \mid a_{l+1}, ..., a_n);$$

$$C_2 = (b_1, b_2, ..., b_k \mid a_{k+1}, ..., a_l \mid b_{l+1}, ..., b_n).$$

Mutation operators introduce small random changes to maintain genetic diversity and prevent premature convergence. For S-boxes, this might involve swapping two randomly chosen elements or applying a random permutation to a subset of elements.

4. Modified genetic algorithm

Our research focuses on developing and implementing a modified genetic algorithm for generating cryptographically strong S-boxes. This section details our approach, the algorithm's structure, and the experimental setup used to evaluate its performance.

4.1. Modified genetic algorithm overview

We have developed a modified genetic algorithm that incorporates elements of hill climbing, enhancing its ability to navigate the complex search space of S-box configurations. This approach allows for a more targeted exploration of promising regions while maintaining the population-based nature of genetic algorithms.

The core idea of our algorithm is to maintain a population of S-boxes, subject them to controlled mutations, evaluate their cryptographic properties, and selectively propagate the best specimens to subsequent generations. This process is iterated until either an S-box meeting the desired criteria is found or a predefined computational limit is reached. The pseudocode for our modified genetic algorithm is:

Algorithm: *Modified genetic algorithm for S-box generation*

Input: S_{pop}, K_{iter}, K_{pop}, K_{mut}

Output: Optimized S-box or 0 (failure)

```
1. For t = 0 to K_iter - 1 do
2.    S_pop = elite_selection(S_pop)
3.    For p = 0 to K_pop - 1 do
4.       S ⟵ S_pop[p]
5.       For k = 0 to K_mut - 1 do
6.          S' ⟵ S
7.          i ⟵ random(0, 255)
8.          j ⟵ random(0, 255)
9.          swap(S'[i], S'[j])
10.         N_f, F_c ⟵ evaluate(S')
11.         If N_f ≥ 104 then
12.            Return S'
13.         S_pop = S_pop ∪ {S'}
14.      End For
15.   End For
16. End For
17. Return 0
```

Key components and parameters of the algorithm:

- S_{pop}: The population of S-boxes, initially generated using the Fisher-Yates shuffle algorithm to ensure bijectivity.
- K_{iter}: Maximum number of iterations, set to 150,000 in our experiments.
- K_{pop}: Population size, representing the number of elite S-boxes maintained in each generation.
- K_{mut}: Number of mutations applied to each S-box in the population per generation.

The elite selection function performs a crucial role in our algorithm. It ranks the S-boxes based on their nonlinearity and objective function values, prioritizing higher nonlinearity and lower objective function values. This function ensures that only the top K_{pop} S-boxes survive to the next generation, maintaining a high-quality population.

4.2. Mutation operator

Our mutation operator is designed to preserve the bijectivity of the S-box while introducing controlled randomness. It operates by swapping two randomly selected (distinct) elements within the S-box. This approach ensures that the fundamental property of bijectivity is maintained throughout the evolutionary process.

Formally, the mutation can be described as:

$$S'[i] = S[j], S'[j] = S[i],$$

where $i, j \in 0,1,\ldots,255$, $i \neq j$ and all other elements remain unchanged.

4.3. Objective function

The choice of objective function is critical in guiding the evolutionary process towards cryptographically strong S-boxes. We employ the WHS function proposed by Clark et

al. [26], which has shown effectiveness in generating high-quality S-boxes. The WHS function is defined as [26]:

$$WHS = \sum_{b=1}^{255} \sum_{i=0}^{255} \left| \left| WHT[b,i] \right| - X \right|^R ,$$

where $WHT[b,i]$ represents the Walsh-Hadamard transform coefficients; i iterate over all component functions and their linear combinations; b iterates over all linear functions; X and R are real-valued parameters.

Based on empirical studies, we set $R = 12$ and $X = 0$, which has been shown to yield optimal results in generating bijective S-boxes with high nonlinearity [56, 57].

4.4. Evaluation criteria

The primary criteria for evaluating the generated S-boxes are:

- Nonlinearity (NL): We aim for a nonlinearity of at least 104, which is close to the theoretical maximum for 8×8 S-boxes.
- Differential uniformity (δ): Lower values indicate better resistance against differential cryptanalysis.
- Algebraic degree (deg): Higher degrees provide better resistance against algebraic attacks.
- Algebraic immunity (AI): Higher values indicate increased resistance to algebraic cryptanalysis.

The evaluate function in our algorithm computes these properties for each generated S-box, allowing us to assess its cryptographic strength comprehensively.

4.5. Experimental setup

Our experiments were conducted on a high-performance computing cluster to handle the computational intensity of the S-box generation process. The implementation was done in C++ for efficiency, with parallelization to utilize multiple cores.

Given that the calculation of the objective function is the most computationally expensive operation in terms of processor time, the complexity of the entire search algorithm can be considered proportional to the number of times the objective function is calculated. This corresponds to the number of S-boxes that were generated and evaluated. We denote this quantity as K_{Sbox}.

To accelerate the algorithm's performance, we implemented parallel computation of the new population using $N_{thread} = 8$ threads within each iteration. This parallelization significantly reduced the overall execution time of the algorithm.

We conducted a comprehensive parameter sweep to analyze the impact of population size and mutation rate on the quality of the generated S-boxes and the algorithm's convergence rate. Specifically:

- Population size (K_{pop}) was varied from 1 to 21 with a step size of 2.
- The mutation rate (K_{mut}) was varied from 1 to 31 with a step size of 3.

For each combination of K_{pop} and K_{mut}, we performed 100 independent runs of the search algorithm to ensure statistical significance. This resulted in a total of 11×11×100 = 12,100 experimental runs.

The algorithm was set to terminate upon finding an S-box with nonlinearity ≥ 104 or reaching the maximum iteration limit of 150,000. For each run, we recorded the number of S-boxes generated and evaluated (K_{Sbox}), which serves as our primary metric for computational efficiency.

5. Results and discussion

This section presents the results of our comprehensive experimental study on the modified genetic algorithm for S-box generation. We analyze the performance of the algorithm across various parameter configurations and discuss the implications of our findings.

5.1. Overview of experimental results

Our primary metric for evaluating the algorithm's efficiency is K_{Sbox}, which represents the number of S-boxes generated and evaluated before finding an S-box with the desired nonlinearity of 104. Table 1 presents the average K_{Sbox} values for different combinations of population size (K_{pop}) and mutation rate (K_{mut}).

5.2. Analysis of population size impact

One of the most striking observations from our results is the superior performance of the algorithm when $K_{pop} = 1$. This configuration consistently yielded the lowest K_{Sbox} values across all mutation rates, with averages ranging from 49,277 to 58,213. This finding is somewhat counterintuitive, as genetic algorithms typically benefit from larger population sizes that provide greater genetic diversity.

The effectiveness of a single-individual population suggests that our algorithm's behavior in this configuration closely resembles that of a stochastic hill-climbing method. This approach appears to be particularly well-suited to the S-box optimization problem, possibly due to the following factors:

Landscape structure: The fitness landscape of S-box configurations may have numerous local optima that are relatively close in quality to the global optimum. In such a scenario, an aggressive local search can be highly effective.

Mutation operator efficiency: Our swap-based mutation operator appears to be sufficiently powerful to navigate the search space effectively, even without the diversity typically provided by a larger population.

Reduced computational overhead: With $K_{pop} = 1$, the algorithm avoids the computational cost associated with managing and evaluating a large population, allowing for more iterations within the same computational budget.

5.3. Impact of mutation rate

While the population size shows a clear trend, the impact of the mutation rate (K_{mut}) is more nuanced. For $K_{pop} = 1$, we observe that:

The lowest K_{Sbox} (49,277) was achieved with $K_{mut} = 7$. Performance generally degraded with higher mutation rates, with K_{Sbox} increasing to 58,213 at $K_{mut} = 1$.

This pattern suggests that there exists an optimal balance between exploration and exploitation in the search process. Lower mutation rates may lead to premature convergence, while higher rates may disrupt good solutions too frequently.

5.4. Scalability and computational efficiency

As K_{pop} increases, we observe a general trend of increasing K_{Sbox} values, indicating reduced computational efficiency. This scaling behavior can be attributed to:

- Increased evaluation overhead: Larger populations require more objective function evaluations per generation.
- Slower convergence: Diversity maintenance in larger populations may slow down the convergence to high-quality solutions.

However, it's worth noting that larger populations might offer benefits not captured by the K_{Sbox} metric alone, such as increased robustness or the ability to find a more diverse set of high-quality S-boxes.

5.5. Parallelization performance

Our implementation of parallel computation using 8 threads (N_{thread} = 8) has proven to be effective in accelerating the search process. This parallelization strategy is particularly beneficial for configurations with larger K_{pop} and K_{mut} values, where the workload can be more evenly distributed across threads.

Table 1
The average number of S-boxes generated (K_{Sbox}) before finding an S-box with N_f = 104

K_{mut}	K_{pop}										
	1	3	5	7	9	11	13	15	17	19	21
1	58,213	65,942	72,830	86,642	101,726	111,990	112,718	125,113	132,806	140,336	149,339
4	56,067	64,863	75,069	89,598	94,726	105,925	122,364	137,003	136,740	151,874	163,291
7	49,277	67,198	77,848	88,353	103,154	109,618	122,382	130,901	142,463	144,601	165,918
10	54,636	65,723	82,198	92,542	102,797	114,163	129,411	137,442	147,416	161,020	165,672
13	56,042	62,660	83,216	94,538	101,073	117,611	124,466	135,244	152,048	158,696	171,756
16	56,010	68,711	79,645	93,134	107,371	120,567	125,274	140,817	150,494	155,049	169,462
19	56,532	65,910	82,883	92,911	105,144	117,877	129,718	142,017	155,463	164,902	175,531
22	54,775	67,236	77,663	92,874	105,559	120,992	131,029	140,772	156,224	162,808	176,669
25	50,066	70,394	79,596	98,967	115,462	118,406	135,294	144,708	157,321	177,621	183,087
28	54,203	70,453	82,200	91,841	108,783	121,683	133,665	152,984	159,887	176,751	181,781
31	53,709	71,581	91,827	101,536	109,625	126,616	143,233	156,987	160,573	183,069	192,493

5.6. Comparison with existing methods

The best-performing configuration of our algorithm (K_{pop} = 1, K_{mut} = 7) achieves an average K_{Sbox} of 49,277. To contextualize our findings within the broader landscape of S-box generation research, we conducted a comprehensive comparison of our genetic algorithm approach with existing methods. Table 2 presents this comparative analysis, encompassing various techniques and cost functions employed in the field.

Our genetic algorithm implementation, utilizing the WHS cost function, achieves results that are on par with the best-known methods in the field. Specifically, our approach generates S-boxes with a nonlinearity of 104 in an average of 49,399 iterations, with a 100% success rate. This performance is comparable to our previous works using hill climbing [6, 28], which required 50,000 iterations on average.

Several key observations emerge from this comparative analysis:

- Parity in Performance: Our genetic algorithm achieves results that are statistically equivalent to the best-known methods, particularly our earlier hill-climbing approach. This parity is significant, as it demonstrates the versatility and potential of genetic algorithms in this domain.
- Algorithmic Diversity: By achieving comparable results through a different algorithmic approach, we have expanded the toolkit available to cryptographers and security researchers. This diversity in high-performing methods enhances the robustness of S-box generation techniques.
- Consistency and Reliability: Like our previous best results, the genetic algorithm maintains a 100% success rate in generating target S-boxes with nonlinearity 104. This level of reliability is crucial for practical applications in cryptographic system design.
- Efficiency Across Methods: The similarity in performance between our genetic algorithm and hill climbing approaches (49,399 vs. 50,000 iterations) suggests that we may be approaching theoretical limits of efficiency for generating S-boxes with these properties using heuristic methods.

Progress from Earlier Genetic Approaches: Compared to earlier genetic algorithm implementations [11, 35], our method shows substantial improvement, reducing the required iterations by orders of magnitude while achieving higher nonlinearity.

The achievement of parity with the best-known results using a genetic algorithm is particularly noteworthy and underscores the potential of evolutionary approaches in cryptographic primitive generation.

5.7. Practical Implications

The superior performance of the $K_{pop} = 1$ configuration has important implications for the practical application of our algorithm:

- Resource efficiency: The algorithm can be effectively run on systems with limited computational resources, as it doesn't require maintaining a large population.
- Simplicity: The simplified population management makes the algorithm easier to implement and tune.
- Adaptability: The algorithm's efficiency makes it suitable for scenarios where S-boxes need to be generated or updated frequently.

Table 2
Comparison of S-box Generation Methods

Method	Cost Function	Algorithm	NL	Success Rate, %	Avg. Iterations
[23,26]	WHS	SA	102	0.5	-
[23]	WHS	SA	104	-	30,000,000
[35]	WHS	HC	100	-	2,500
[35]	WHS	GaT	104	-	3,239,000
[11]	WHS	Ga	102	-	28,200
[11]	WHS	GaT	104	-	3,849,881
[11]	WHS	LSA	102	-	6,701
[11]	PCF	Ga	104	-	741,371
[11]	PCF	GaT	104	-	167,451
[11]	PCF	LSA	104	-	172,280
[27]	WCF	LSA	104	-	89,460
[27]	WCF	HC	104	37	65,933
[58]	WHS	SA	104	56.4	450,000
[48]	WCF	SA	104	100	65,000
[48,59]	ECF	SA	104	100	55,000 … 83,000
[49]	WHS	HC	104	100	50,000
[6,28]	WCFS	HC	104	100	50,000
Our work	WHS	Ga	104	100	49,399

However, it's important to note that while this configuration is optimal for finding a single high-quality S-box, alternative configurations may be more suitable for generating a diverse set of S-boxes or for multi-objective optimization scenarios.

5.8. Limitations and future work

While our results are promising, several avenues for future research remain:

- Extended cryptographic criteria: Incorporate additional criteria such as algebraic immunity and differential uniformity into the objective function.
- Adaptive parameter tuning: Develop methods to dynamically adjust K_{pop} and K_{mut} during the search process.
- Alternative mutation operators: Explore more sophisticated mutation strategies that leverage domain-specific knowledge about S-box structures.
- Multi-objective optimization: Extend the algorithm to simultaneously optimize multiple cryptographic properties, potentially using Pareto-based approaches.

In conclusion, our modified genetic algorithm demonstrates exceptional efficiency in generating cryptographically strong S-boxes, particularly in its hill-climbing-like configuration. These findings contribute valuable insights to the field of cryptographic primitive design and offer a powerful tool for the development of secure symmetric encryption systems.

6. Conclusions

This study presents a significant advancement in the field of S-box generation for symmetric key cryptography, focusing on the application of genetic algorithms to produce highly nonlinear substitutions. Our research demonstrates that genetic algorithms, when properly optimized and combined with the Walsh-Hadamard Spectrum (WHS) cost function, can achieve performance parity with the best-known methods in generating 8×8 S-boxes with a nonlinearity of 104.

Key findings of our work include:

- The genetic algorithm approach achieves an average of 49,399 iterations to generate target S-boxes, comparable to the best results of 50,000 iterations using hill-climbing methods.
- A 100% success rate in producing S-boxes with the desired nonlinearity, matching the reliability of top-performing techniques.
- A significant improvement over earlier genetic algorithm implementations, reducing iteration counts by orders of magnitude.

The achievement of performance parity using a different algorithmic approach expands the toolkit available to cryptographers and highlights the versatility of genetic

methods in cryptographic primitive generation. This diversity in high-performing techniques enhances the robustness of S-box generation methodologies.

Furthermore, our results underscore the potential of genetic algorithms in this domain, particularly their adaptability to evolving cryptographic criteria and their inherent parallelization capabilities. These characteristics position genetic approaches as promising avenues for future research, potentially leading to more efficient, flexible, and innovative S-box generation techniques.

In conclusion, while not surpassing existing methods in raw performance, our genetic algorithm approach offers a valuable alternative that matches the best-known results. This equivalence, coupled with the unique advantages of genetic algorithms, opens new perspectives in cryptographic research and development. Future work should focus on exploiting these advantages, potentially through hybridization with other heuristic methods or by leveraging parallel computing architectures to further enhance S-box generation efficiency.

References

[1] Y. Li, et al., HDLBC: A Lightweight Block Cipher with High Diffusion, Integr. 94 (2024) 102090. doi: 10.1016/j.vlsi.2023. 102090.

[2] A. Tiwari, Chapter 14—Cryptography in Blockchain, Distributed Computing to Blockchain, Academic Press (2023) 251–265. doi: 10.1016/B978-0-323-96146-2.00011-5.

[3] M. Milanič, B. Servatius, H. Servatius, Chapter 8 - Codes and cyphers, Discrete Mathematics With Logic, Academic Press (2024) 163–179. doi: 10.1016/B978-0-44-318782-7.00013-7.

[4] Y. Sovyn, et al., Minimization of Bitsliced Representation of 4×4 S-Boxes based on Ternary Logic Instruction, in Cybersecurity Providing in Information and Telecom-munication Systems, vol. 3421 (2023) 12–24.

[5] A. S. Changle, S. P. Metkar, R. K. Patole, Implementation of S-box for Lightweight Block Cipher, 3rd International Conference on Intelligent Technologies (2023) 1–4. doi: 10.1109/CONIT59222. 2023.10205535.

[6] A. Kuznetsov, et al., A New Cost Function for Heuristic Search of Nonlinear Substitutions, Expert Syst. Appls. 237 (2024). doi: 10.1016/j.eswa.2023.121684.

[7] A. J. Menezes, et al., CRC Press (2018). doi: 10.1201/9780429466335.

[8] C. E. Shannon, Communication Theory of Secrecy Systems, Bell Syst. Tech. J. 28 (1949) 656–715. doi: 10.1002/j.1538-7305.1949.tb00928.x.

[9] C. Carlet, Vectorial Boolean Functions for Cryptography, Boolean Models and Methods in Mathematics, Computer Science, and Engineering (2006).

[10] A. Freyre-Echevarría, et al., An External Parameter Independent Novel Cost Function for Evolving Bijective Substitution-Boxes, Symmetry 12 (2020) 1896. doi: 10.3390/sym12111896.

[11] S. Picek, M. Cupic, L. Rotim, A New Cost Function for Evolution of S-Boxes, Evolut. Computation 24 (2016) 695–718. doi: 10.1162/EVCO_a_00191.

[12] J. Álvarez-Cubero, Vector Boolean Functions: Applications in Symmetric Cryptography (2015). doi: 10.13140/RG. 2.2.12540.23685.

[13] K. Lisitskiy, I. Lisitska, A. Kuznetsov, Cryptographically Properties of Random S-Boxes., 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer II, vol. 2732 (2020) 228–241.

[14] I. Gorbenko, et al., Random S-Boxes Generation Methods for Symmetric Cryptography, IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON) (2019) 947–950. doi: 10.1109/UKRCON.2019.8879962.

[15] J. Daemen, V. Rijmen, Specification of Rijndael, The Design of Rijndael: The Advanced Encryption Standard (AES), Springer (2020) 31–51. doi: 10.1007/978-3-662-60769-5_3.

[16] G. V. Bard, Algebraic Cryptanalysis, Springer US (2009). doi: 10.1007/978-0-387-88757-9.

[17] N. T. Courtois, G. V. Bard, Algebraic Cryptanalysis of the Data Encryption Standard, Cryptography and Coding, LNSC 4887 (2007) 152–169. doi: 10.1007/978-3-540-77272-9_10.

[18] N. T. Courtois, J. Pieprzyk, Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, Advances in Cryptology—ASIACRYPT 2002, LNCS 2501 (2002) 267–287. doi: 10.1007/3-540-36178-2_17.

[19] V. Buhas, et al., Using Machine Learning Techniques to Increase the Effectiveness of Cybersecurity, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3188, no. 2 (2021) 273–281.

[20] V. Buriachok, et al., Invasion Detection Model using Two-Stage Criterion of Detection of Network Anomalies, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 2746 (2020) 23–32.

[21] V. Sokolov, P. Skladannyi, H. Hulak, Stability Verification of Self-Organized Wireless Networks with Block Encryption, in: 5th International Workshop on Computer Modeling and Intelligent Systems, vol. 3137 (2022) 227–237.

[22] A. Freyre Echevarría, Evolución Híbrida de S-cajas no Lineales Resistentes a Ataques de Potencia (2020). doi: 10.13140/RG.2.2.17037.77284/1.

[23] J. McLaughlin, Applications of Search Techniques to Cryptanalysis and the Construction of Cipher Components, PhD, University of York (2012). URL: https://etheses.whiterose.ac.uk/3674/

[24] L. D. Burnett, Heuristic Optimization of Boolean Functions and Substitution Boxes for Cryptography, PhD, Queensland University of Technology, (2005). URL: https://eprints.qut.edu.au/16023/

[25] A. Kuznetsov, et al., SBGen: A High-performance Library for Rapid Generation of Cryptographic S-boxes, SoftwareX 27 (2024) 101788. doi: 10.1016/j.softx.2024.101788.

[26] J. A. Clark, J. L. Jacob, S. Stepney, The Design of S-boxes by Simulated Annealing, New Gener. Comput. 23 (2005) 219–231. doi: 10.1007/BF03037 656.

[27] A. Freyre Echevarría, I. Martínez Díaz, A New Cost Function to Improve Nonlinearity of Bijective S-boxes (2020).

[28] O. Kuznetsov, et al., Enhancing Smart Communication Security: A Novel Cost Function for Efficient S-Box Generation in Symmetric Key Cryptography, Cryptography 8 (2024) 17. doi: 10.3390/cryptography8020017.

[29] O. Kuznetsov, et al., Enhancing Cryptographic Primitives through Dynamic Cost Function Optimization in Heuristic Search, Electronics 13 (2024) 1825. doi: 10.3390/electronics13101825.

[30] K. Nyberg, Perfect Nonlinear S-boxes, Advances in Cryptology — EUROCRYPT '91, LNCS 547 (1991) 378–386. doi: 10.1007/3-540-46416-6_32.

[31] K. Nyberg, Differentially uniform mappings for cryptography, Advances in Cryptology—EUROCRYPT '93, LNCS 765 (1994) 55–64. doi: 10.1007/3-540-48285-7_6.

[32] R. La Scala, S. K. Tiwari, Stream/Block Ciphers, Difference Equations and Algebraic Attacks, J. Symbolic Comput. 109 (2022) 177–198. doi: 10.1016/j.jsc.2021.09.001.

[33] D. Souravlias, K. Parsopoulos, G. Meletiou, Designing Bijective S-boxes Using Algorithm Portfolios with Limited Time Budgets, Appl. Soft Comput. 59 (2017) 475–486. doi: 10.1016/j.asoc.2017.05. 052.

[34] W. Millan, et al., Evolutionary Heuristics for Finding Cryptographically Strong S-Boxes, Information and Communication Security, LNCS 1726 (1999) 263–274. doi: 10.1007/978-3-540-47942-0_22.

[35] P. Tesar, A New Method for Generating High Non-linearity S-Boxes, (2010). URL: http://dspace.lib.vutbr.cz/xmlui/handle/11012/56957

[36] G. Ivanov, N. Nikolov, S. Nikova, Cryptographically Strong S-Boxes Generated by Modified Immune Algorithm, Cryptography and Information Security in the Balkans, LNSC 9540 (2016) 31–42. doi: 10.1007/978-3-319-29172-7_3.

[37] G. Ivanov, N. Nikolov, S. Nikova, Reversed Genetic Algorithms for Generation of Bijective S-boxes with Good Cryptographic Properties, Cryptogr. Commun. 8 (2016) 247–276. doi: 10.1007/s12095-015-0170-5.

[38] M. Rodinko, R. Oliynykov, Y. Gorbenko, Optimization of the High Nonlinear S-Boxes Generation Method, Tatra Mountains Math. Publ. 70 (2017) 93–105. doi: 10.1515/tmmp-2017-0020.

[39] F. Artuğer, F. Özkaynak, A New Post-Processing Approach for Improvement of Nonlinearity Property in Substitution Boxes, Integration 94 (2024) 102105. doi: 10.1016/j.vlsi.2023.102105.

[40] T. Haider, N. A. Azam, U. Hayat, Substitution Box Generator with Enhanced Cryptographic Properties and Minimal Computation Time, Expert Syst. Appl. 241 (2024) 122779. doi: 10.1016/j.eswa.2023.122779.

[41] S. S. Jamal, et al., Region of Interest-Based Medical Image Encryption Technique Based on Chaotic S-boxes, Expert Syst. Appl. 238 (2024) 122030. doi: 10.1016/j.eswa.2023.122030.

[42] S. Fahd, et al., The Reality of Backdoored S-Boxes—An Eye Opener, J. Inf. Secur. Appls. 80 (2024) 103674. doi: 10.1016/j.jisa.2023.103674.

[43] G. Ars, J.-C. Faugère, Algebraic Immunities of Functions Over Finite Fields, INRIA, (2005). URL: https://hal.inria.fr/inria-00070475

[44] O. O. Kuznetsov, et al., Algebraic Immunity of Non-linear Blocks of Symmetric Ciphers, Telecommun. Radio Eng. 77 (2018) 309–325. doi: 10.1615/TelecomRadEng.v77.i4.30.

[45] A. Kuznetsov, et al., Evaluation of Algebraic Immunity of modern block ciphers, IEEE Int. Conf. Dependable Syst., Serv. Technol. (2018) 288–293. doi: 10.1109/DESSERT.2018.8409146.

[46] K. Lisitskiy, I. Lisitska, A. Kuznetsov, Cryptographically Properties of Random S-boxes, in: ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, vol. 2732 (2020) 228–241.

[47] I. Gorbenko, et al., Random S-boxes Generation Methods for Symmetric Cryptography, IEEE 2nd Ukraine Conference on Electrical and Computer Engineering, (2019) 947–950. doi: 10.1109/ukrcon.2019.8879962.

[48] A. Kuznetsov, et al., Optimized Simulated Annealing for Efficient Generation of Highly

Nonlinear S-boxes, Soft. Comput. (2023). doi: 10.1007/s00500-023-09334-y.
[49] A. Kuznetsov, et al., Optimizing Hill Climbing Algorithm for S-Boxes Generation, Electronics 12 (2023) 2338. doi: 10.3390/electronics12102338.
[50] A. Freyre-Echevarría, et al., Evolving Nonlinear S-Boxes With Improved Theoretical Resilience to Power Attacks, IEEE Access 8 (2020) 202728–202737. doi: 10.1109/ACCESS.2020.3035163.
[51] E.C. Laskari, G.C. Meletiou, M.N. Vrahatis, Utilizing Evolutionary Computation Methods for the Design of S-Boxes, International Conference on Computational Intelligence and Security, (2006) 1299–1302. doi: 10.1109/ICCIAS.2006.295267.
[52] A. Ghosh, S. Das, B. Saha, Chapter 6—Nature-inspired Optimization Algorithms, Artificial Intelligence in Textile Engineering, Woodhead Publishing (2024) 171–231. doi: 10.1016/B978-0-443-15395-2.00002-8.
[53] C.-W. Tsai, M.-C. Chiang, Chapter Seven—Genetic algorithm, Handbook of Metaheuristic Algorithms, Academic Press (2023) 111–138. doi: 10.1016/B978-0-44-319108-4.00020-4.
[54] T. Kapuściński, R. K. Nowicki, C. Napoli, Application of Genetic Algorithms in the Construction of Invertible Substitution Boxes, Artificial Intelligence and Soft Computing, LNAI 9692 (2016) 380–391. doi: 10.1007/978-3-319-39378-0_33.
[55] C.-W. Tsai, M.-C. Chiang, Chapter Sixteen—Local Search Algorithm, Handbook of Metaheuristic Algorithms, Academic Press (2023) 351–374. doi: 10.1016/B978-0-44-319108-4.00030-7.
[56] A. Kuznetsov, et al., WHS Cost Function for Generating S-boxes, IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (2021) 434–438. doi: 10.1109/PICST54195.
2021.9772133.
[57] A. Kuznetsov, et al., Opportunities to Minimize Hardware and Software Costs for Implementing Boolean Functions in Stream Ciphers, Int. J. Comput. 18 (2019) 443–452.
[58] A. Kuznetsov, et al., Optimization of a Simulated Annealing Algorithm for S-Boxes Generating, Sensors 22 (2022) 6073. doi: 10.3390/s22166073.
[59] A. Kuznetsov, et al., Generation of Nonlinear Substitutions by Simulated Annealing Algorithm, Inf. 14 (2023) 259. doi: 10.3390/info14050259.