IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 March 2023, accepted 5 April 2023, date of publication 17 April 2023, date of current version 4 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266290

RESEARCH ARTICLE

Grey Wolf Optimizer and Discrete Chaotic Map for Substitution Boxes Design and Optimization

ALI IBRAHIM LAWAH[1], ABDULLAHI ABDU IBRAHIM[1], SINAN Q. SALIH[2], HUSSAM S. ALHADAWI[3,4], AND POH SOON JOSEPHNG[5]

[1]Department of Electrical and Computer Engineering, Altinbas University, 34217 Istanbul, Turkey
[2]Technical College of Engineering, Al-Bayan University, Baghdad 10010, Iraq
[3]Department of Computer Techniques Engineering, Dijlah University College, Baghdad 10011, Iraq
[4]College of Engineering, University of Warith Al-Anbiyaa, Karbala 56001, Iraq
[5]Faculty of Data Science and Information Technology, INTI International University, Nilai, Negeri Sembilan 71800, Malaysia

Corresponding authors: Sinan Q. Salih (sinan_salih@outlook.com) and Poh Soon Josephng (joseph.ng@newinti.edu.my)

ABSTRACT A metaheuristic approach based on the nature-inspired and well-known Grey Wolf Optimization algorithm (GWO) was employed in this study to design an approach for retrieving strong designs of 8 × 8 substitution boxes (S-boxes). The GWO was developed as a novel metaheuristic based on inspiration from grey wolves and how they hunt. The ability of the GWO to quickly explore the search space for the near/optimal feature subsets that maximize any given fitness function (in consideration of its distinctive hierarchical structure) aids in the construction of strong S-boxes that can satisfy the required criteria. However, when tackling optimization problems, GWO may experience the problem of premature convergence. Therefore, a variant of GWO called Crossover Grey Wolf Optimizer (XGWO) has been proposed in this study. The performance of the proposed novel approach was evaluated using numerous cryptographic performance metrics, including bijective property, bit independence, strict avalanche, linear probability, and I/O XOR distribution and the result was contrasted with a couple of existing S-box creation techniques. Overall, the results of the experiment showed that the suggested S-box design had adequate cryptographic features.

INDEX TERMS Substitution boxes, optimization, nature-inspired algorithms, Grey Wolf Optimizer, cryptology.

I. INTRODUCTION

In modern cryptographic applications, one of the major components is the S-bo; an S-box refers to a ''non-linear substitution mapping function $S(x)$: $GF(2^n) \rightarrow GF(2^m)$, that can be represented as the following Boolean function formulation $f(x) = (f1(x), f2(x), \ldots, fm(x))$. S-boxes are a crucial part of symmetric-key algorithms and are mostly used to hide the relationship between the input key and the output. S-boxes are widely accepted to have a major impact on the algorithm's security attributes [1], [2]. Moreover, S-boxes are essential for establishing the related block cipher's confusion property, which increases its resistance to cryptanalysis [3].

The creation of appropriate S-boxes that will meet the requirements for symmetric-key cryptography has never been easy. This has been the case because the level of security of any encryption process is a function of the S-box quality. In 1977, the Data Encryption Standard (DES) was introduced based on eight 6 × 4 S-boxes [4; it is a good example of this scenario. The known susceptibility of DES to different attacks, such as linear cryptanalysis and parallel brute-force attacks [5], prompted the development of the Advanced Encryption Standard (AES) in the year 2000. The AES relies on the use of S-boxes that have been shown to be robust in the presence of both differential and linear cryptanalysis attacks [7, 6]. Currently, the AES is recognized as a secure standard, but uses a static S-box, meaning that despite changes in key, the S-box remained the same. Hence, it was shown that the key-dependent S-box enhanced

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

the algorithm complexity whereas the static S-box provided ways for cipher text pair attacks [8], [9]. The fundamental problem, which had been resolved in [10] and [11], was the systematic design of dynamic S-boxes, despite the fact that key-dependent S-boxes improved the strength of the algorithm. Many unresolved problems still exist in the design and evaluation of S-boxes for cryptography applications [11], [12]. It is possible to classify the design of S-boxes that can withstand cryptanalysis as an NP-hard problem [13]. Because it is difficult to create an S-box with perfect cryptographic features, there must be some sort of trade-off [13]. Currently, the algebraic, random, and metaheuristic-based approaches are the three generic strategies primarily used in the design of S-boxes. However, each of these methods has certain benefits and drawback; for instance, the random search method, while straightforward, typically results in S-boxes with subpar cryptographic properties [14]. For the algebraic strategy, it is not suitable for large-scale S-box generation despite producing S-boxes with good cryptographic attributes [15]. Thus, the metaheuristic-based technique is a good alternative for constructing S-box due to its versatility and straightforward theoretical foundation. The optimization method in general, and nature-inspired techniques in particular, have been adopted for addressing many optimization issues in the literature, including Engineering & Machine Learning [16], [17], [18], and Power due to their better performance. The literature has investigated the combination of numerous metaheuristics and chaotic maps for the design of 8×8 S-box due to their widespread use in cryptosystems. Some of the recent integrations include the combination of the Jaya algorithm with the Tent map, Logistic map, and Sine map [19], the hybridization of the chaotic map with the globalized Firefly algorithm and its variants [20], the combination of selective chaotic maps with the Tiki-Taka algorithm [21], the hybridization of Tent map with Agent Cowards and Heroes algorithm [22], etc.

Recently, Grey Wolf Optimizer (GWO) [23] was developed as a metaheuristic algorithm that mimics the hunting style of grey wolfs when they search for prey, then attack it in a unique way. The standard version of GWO has been successfully implemented for handling different global optimization problems [24]. In GWO, the position updating mechanism depends mainly on the best three candidate solutions, namely ''Alpha, Beta, and Delta'', which denote the best solution, the second-best solution, and the third-best solution, respectively. All search agents in the swarm follow the best solutions, which in some cases, get stuck in the same position, especially the alpha. This implies that the positions of all search agents exploit the search region of alpha for many more iteration; this could increase the chances of the algorithm being trapped in the local minima. Therefore, the position updating mechanism of GWO needs to be modified to enable the best candidate solutions to explore the search area once they fail to get optimized.

The major objectives of this study are as follows:

i) To design an approach for avoiding local minima entrapment by integrating the GWO with a special crossover operator. The new algorithm that results from this integration is called ''Crossover Grey Wolf Optimizer'' (XGWO). In the XGWO, new solutions are generated based on the already found best solutions (i.e., Alpha, and Beta) which are combined in a specific order. The new crossover step ensures the global search ability of the algorithm and enhances the searching performance of GWO while the stop condition is not satisfied.
ii) To design a new initialization step for XGWO based on the discrete chaotic map instead of the pseudo-random number generator to ensure high nonlinearity and allow more diversification.
iii) To test and validate the performance of XGWO against GWO and other existing metaheuristics. In this study, the utilized test bed (performance metrics) includes bijectivity, strict avalanche criteria (SAC), nonlinearity, bit independence criteria (BIC), linear approximation probability (LP), and differential probability (DP).
iv) XGWO is the first GWO-based crossover variant considered for solving the issues of 8×8 S-boxes generation and optimization.

The structure of this article is as follows: The related works on the use of metaheuristics for S-box design are introduced in Section II of the article. The problem description is detailed in Section III. The original GWO and the suggested variation, XGWO, are presented in Section IV. In Section V, the proposed approach was evaluated, including how well S-box properties are met and how it compares to existing metaheuristics that incorporate chaotic maps. Section VI presented the performance analysis of the proposed S-boxes. The study is concluded in Section VII, along with recommendations for further research.

II. RELATED WORKS

Substitution boxes, as earlier mentioned, are typically built using three general techniques [25] which are the random, algebraic, and metaheuristic methods. It is difficult to execute these methods and produce a large collection of strong S-boxes; hence, only a few studies based on algebraic methods are covered in this review. However, as noted, the algebraic method-based S-boxes tend to exhibit the most superior cryptographic properties [26]. The study by Nyberg [27] designed a strong S-box using the finite field inversion arithmetic method while Daemen and Rijmen [28] proposed an S-box that uses both the inverse mapping method and the affine transformation. S-boxes can also be generated using random computationally based methods; these methods frequently fill S-box entries with random numbers or employ a pseudo-random generator, and they are easy to implement. Being that the S-box generation entries in those methods

rely heavily on chance, these methods frequently produce S-boxes of poor quality [14]. Some works combine the chaotic maps with these methods for S-box generation based on random computing as an upgrade. For instance, a method for creating S-boxes based on the Logistic and exponential maps was presented by Jakimoski and Kocarev [29]. Similarly, Tang [30] developed a technique that produced S-boxes using a discretized, chaotic, two-dimensional Baker map. Khan [31] created a nonlinear fractional chaos-based method that relies on a progressive time Lorenz system for S-box creation. A compositional permutation technique based on a discrete chaotic map was also used by Lambić [32] for S-box creation. In the study by Özkaynak and A. B. Özer, [33], a chaotic Lorenz system-based method of S-box retrieval was presented. Additionally, Khan and Asghar [34] employed the Ginger Breadman Chaotic Map with S8 Permutations as part of their initial S-box generating strategy. The study by Çavuşoğlu et al. [35], offered a novel way of S-box generation based on a scaled Zhongtang chaotic system, whereas Zhang et al. [36] developed S-boxes employing the I-ching operators. A feasible 4-chaotic map-based S-box design method was presented by Alshekly et al. [37]. Considering the chaotic maps' promising performance in S-box design and generation, studies have been focusing on the optimization-based techniques recently.

The S-box design methods based on metaheuristics make efficient use of the base algorithm to carry out the search process; this improves the outcomes of such random computational-based strategies. Metaheuristic-based systems typically offer internal mechanisms that encourage learning from previous activities [38], in comparison to random-based computational methods, thereby delivering better solutions. This is due to the intensification and diversification operators that are built into these metaheuristic-based approaches. Furthermore, metaheuristic-based systems are generally used to provide internal strategies for overcoming local optima trapping. Because of these factors, the literature contains multiple instances of metaheuristic algorithms for S-box construction being successfully used.

Numerous studies have demonstrated that adding chaotic maps to metaheuristic algorithms enhances the nonlinearity performance of S-boxes. For instance, the study by Chen [39] employed the single population-based metaheuristic based on Simulated Annealing (SA) with chaotic-based swapping. The Wang method [16] was then used in conjunction with chaotic and genetic algorithms (GA) to produce an optimized S-box. The Logistic Map and Bacterial Foraging Optimization (BFO) algorithms were intertwined by Tian and Lu [40]. BFO is a metaheuristic inspired by the foraging pattern of bacteria for the generation of robust 8×8 S-boxes. Authors in [13] have presented a 6-D compound hyper-chaotic map integrated with an artificial bee colony (ABC). The foraging behaviour of honeybees is mimicked in the ABC to obtain the appropriate S-box.

Farah and Belazi [41] proposed an integrated new chaotic map by combining the Jaya algorithm with Sine, Logistic, and Tent maps. In another work by [42], the teaching-learning-based optimization (TLBO) algorithm and the Henon and Logistic maps were used to create a robust S-box. In this method, TLBO finds the best solutions using the teaching and learning phases.

Ahmad et al [43] utilized an Ant Colonization Optimizer-based metaheuristic with a modulated chaotic variable sample to transform the initial S-box into a Traveling Salesman Problem (TSP) through an edge matrix to retrieve a suitable S-box design. In this ACO-based metaheuristic approach, a Logistic map and a Tent map were combined. The Standard Firefly Algorithm (SFA) is used in the work of Ahmed et al. [44] to generate S-boxes using a discrete chaotic map. SFA enhances the search results by utilizing the brightness of biological fireflies. The new approach proposed by Alzaidi et al. [45] relied on the use of an enhanced chaotic map and β-Hill Climbing search algorithm. Furthermore, a chaotic map and the Globalized Firefly Algorithm (GFA) were used by Alhadawi et al. [20] for S-box generation. The chaos-based movement of the best firefly was exploited by the GFA to reach optimal results. In a different study,

Alhadawi et al. [46] combined a chaotic map and the Cuckoo search method for S-box creation. Recently, Soto et al [47] proposed a way of evading premature convergence and improving the non-linearity property during S-box creation; the proposed method is based on human behaviour inspired frameworks and supported by self-organizing maps.

Nafiseh and Sodeif [48] utilized ergodic chaotic maps to enhance the PSO algorithm's ability to create cryptographically robust S-boxes. To arrive at an effective S-box design, Zamli et al. [21] developed the Selective Chaotic Tiki-taka Algorithm (SCMTTA), which hybridized TTA and 5 chaotic maps (Tent map, Logistic map, Chebyshev map, Sine map, and Singer map). TTA searches for the best solutions using the populations of balls and players.

Zamli [22] also created strong S-boxes using a combination of the Tent map and the Adaptive Agent Heroes and Cowards (AAHC). The AAHC dynamically partitions its population into heroes and cowards using exponential functions. An improved multi-swarm PSO (MPSO) method based on the meeting room approach was proposed by Alhadawi et al. [49] for the creation of strong 8×8 S-boxes. Despite the existing number of methods for S-box design, none of these methods could create an S-box that exhibits similar levels of security to AES based on relevant metrics. Hence, it is important to address this issue in both practical and academic settings [13], [43]. Additionally, scholars should endeavour to come up with novel metaheuristic based S-box strategies by investigating the applicability of new metaheuristics for S-box design, given that no single metaheuristic approach could be said to be superior to other strategies.

III. PROBLEM DESCRIPTION

Security professionals and researchers have evaluated S-box strength using a variety of performance metrics [7], [29], [50]. To evaluate the cryptographic qualities of the created S-boxes in this study, six key characteristics were considered; these are nonlinearity, bijectivity, SAC, differential uniformity, BIC, and LP. These well-known performance standards for S-boxes will be briefly explained in the next subsection.

Definition 3.1: Bijectivity: The confirmation of the bijective property of S-boxes using this method was first proposed by [51]. The bijectivity property is said to be satisfied if the Boolean function f_i of an S-box needs to be balanced. This calls for an equal distribution of 0's and 1's in the output of the S-box. Boolean functions are determined with $f_i(1 \leq i \leq n)$ as:

$$wt\left(\sum_{i=1}^{n} a_i f_i\right) = 2^{n-1} \tag{1}$$

where $a_i \in \{0, 1\}$, $(a_1, a_2, \ldots, a_n) \neq (0, 0, \ldots, 0)$, and $wt()$ is the Hamming weight. For an 8×8 S-box, bijectivity is satisfied if the look-up contains different values in the range of [0, 255].

Definition 3.2: Nonlinearity: To prevent linear cryptanalysis, the nonlinearity score is closely related to plaintext confusion and block cipher immunity. The Walsh spectrum (WS) calculates a Boolean function's nonlinearity to estimate its minimum separation from all other Boolean functions.

$$N_f = 2^{n-1}(1 - 2^{-n} \max_{x \in GF(2^n)} |WS \prec f \succ (w)|) \tag{2}$$

The Walsh spectrum of $f_{(x)}$ can be defined as:

$$S_{\prec f \succ}(w) = \sum_{x \in GF(2^n)} (-1)^{f(x) \oplus x.w} \tag{3}$$

where $w \in GF(2^n)$ and $x.w$ signifies the dot product of x and w, which is provided as:

$$x.w = x_1 \oplus w_1 \oplus \ldots \oplus x_n \oplus w_n$$

Definition 3.3: Strict Avalanche Criteria: Webster and Tavares [52] defined SAC by highlighting how output bits might be partially altered by only complementing one input bit. The S-box was then checked to see if it matched the SAC requirements using a successful procedure that was put into place after that. If the dependence matrix served as the representation of the S-box SAC, the S-box would satisfy the SAC if each element's value was close to the optimal value of 0.5. The study by [53] provides additional information regarding this calculation, Meanwhile [54], applies the subsequent equation to obtain the SAC offset amount.

$$S(f) = \frac{1}{n^2} \sum\nolimits_{1 \leq i \leq n} \sum\nolimits_{1 \leq j \leq n} \left|0.5 - P_{i,j}(f)\right| \tag{4}$$

Definition 3.4: Bit Independence Criteria: Webster and Tavares [52] devised the BIC metric as a way of reflecting the pairwise independence of all avalanche vector series generated by a single plaintext complementation. The pairwise independence between such pairings is determined by measuring the correlation between the items in each pair. Consider the S-box with Boolean function $(h_1, h_2, \ldots, h_n)$, an S-box that is said to have met the BIC $h_j \oplus h_k (j \neq k, 1 \leq j, k \leq n)$ should be highly nonlinear [55]. Furthermore, the determination of the BIC of the generated S-box is done by calculating the nonlinearity and SAC of $h_j \oplus h_k (j \neq k)$.

Definition 3.5: Differential Uniformity: This is a measure of the robustness of an S-box to withstand a differential cryptanalysis. To compute DP, Δx_i, an input differential is uniquely mapped to Δy_i, an output differential for each i. The DP is expressed in Eq. 5 as a measure of this differential uniformity for a given S-box.

$$DP(\Delta x - \Delta y) = \left(\frac{\#\{x \in X \,|S(x) \oplus S(x \oplus \Delta x) = \Delta y|}{2^m}\right) \tag{5}$$

where set X contains 2^m possible input values.

Definition 3.6: Linear Probability: The LP is used as a criterion for measuring the maximum imbalance value of a situation because it can serve as a way of determining the maximum imbalance value of the outcome of an event, with a and b representing two masks associated with the parity of the input and output bits, respectively. The LP is also mathematically defined as [5]

$$LP = max_{a,b \neq 0} \left|\frac{\#\{x \in X \,|x.a = f(x).b|}{2^n} - 0.5\right| \tag{6}$$

where set X is composed of all the possible inputs while 2^n is the number of components in X.

IV. THE PROPOSED METHODOLOGY

The suggested modification to GWO is detailed in this section, along with a description of the traditional GWO. Additionally, a discrete chaotic map that is utilized to provide the required randomness for GWO is explained. These algorithms are the basis for the description of the suggested S-box generation methods.

A. DISCRETE CHAOTIC MAP

Let $K = k_0 k_1 \ldots k_{m-2} k_{m-1}$ denote a permutation of the set $\{0, 1, \ldots, m-1\}$. Permutation $k^r = k_{m-1} k_{m-2} \ldots k_1 k_0$ is the reverse permutation of the permutation K.

The composition $h = f \circ g$ of two permutations f and g of the same set A, is the permutation mapping each $z \in A$ into $h(z) = f(g(z))$.

Let S_m denote all permutations of the set $\{0, 1, \ldots, m-1\}$. Lehmer code [56] is bijective function l: $S_m \rightarrow \{0, 1, 2, \ldots, m! - 1\}$. Function $l(K) = \sum_{0 \leq i < m} c_i \cdot (m - 1 - i)!$ Where $K \in S_m$ and c_i is the number of elements of the set $\{j > i | k_j < k_i\}$. Inverse Lehmer code is bijective function $l^{-1} : \{0, 1, 2, \ldots, m! - 1\} \rightarrow S_m$.

In [57] defined a 1-D discrete chaotic map as follows:

$$Z_{i+1} = Z_i \circ f(Z_i, C) \tag{7}$$

where $Z_i, C \in S_m$ and $f : S_m \to S_m$. If $z_i = l(Z_i)$ and $c = l(C)$, this map can also represent as

$$Z_{i+1} = l[l^{-1}(Z_i) \circ f\left(l^{-1}(Z_i), l^{-1}(c)\right)] \tag{8}$$

where $z_i, c \in \{0, 1, 2, \ldots, m! - 1\}$ and $f : S_m \to S_m$. In [57] and [58], The specific scenario of a one-dimensional discrete chaotic map is considered, in which

$$f(Z_i, C) = l^{-1}(\left|l(C \circ Z_i - l((C \circ Z_i)^r)\right|) \tag{9}$$

Based on (1) and (3) we derive the map $F_m : \{0, 1, 2, \ldots, m! - 1\} \to \{0, 1, 2, \ldots, m! - 1\}$ by:

$$F_m(z) = l(l^{-1}(z) \circ l^{-1}\left(\left|l\left(C \circ l^{-1}(z)\right)\right.\right. \\ \left.\left. - l\left(\left[C \circ l^{-1}(z)\right]^r\right)\right|)) \tag{10}$$

This map can also be presented as

$$Z_{i+1} = Z_i \circ l^{-1}(\left|l(C \circ Z_i) - l\left((C \circ Z_i)^r\right)\right|) \tag{11}$$

B. GREY WOLF OPTIMIZER

The social leadership among grey wolves, as well as their hunting behavior, serve as the inspiration for the GWO. In GWO, three types of leader wolves (α, β, and δ) are taken into consideration; they are taken as the best solutions that will lead the other wolves (referred to as ω) toward finding the optimal solutions while exploring the global best. The hunting strategy of grey wolves is comprised of three key steps - encircling, haunting, and attacking the target.

- Encircling: This is the encircling of the prey by the hunting grey wolves; it can be modeled as follows:

$$D = \left|C \times X_p(t) - X(t)\right| \tag{12}$$
$$X(t+1) = X_p(t) - A \times D \tag{13}$$

where X_p = the position of the target (prey), X = the position vector of the hunting wolf, t = the current iteration. The coefficient vectors, C, and A, can be calculated thus:

$$A = 2 \times A \times r_1 - a(t) \tag{14}$$
$$C = 2 \times r_2 \tag{15}$$

where r_1, r_2 are ''random vectors in the range of [0, 1]; the elements of the vector α decreases linearly from 2 to 0 during the iteration process by:

$$a(t) = 2 - (2 \times t)|MaxIter \tag{16}$$

- Haunting: The haunting behavior of the wolves is mathematically modeled by assuming that a, β, and δ know more about the position of the target (prey); hence, the position of the a, β and δ (best solution) serve as the guide for the rest of the wolves ω. The hunting behavior of the wolves is described using the following Equations:

$$D_a = |C_1 \times X_a - X(t)|, \\ D_\beta = \left|C_2 \times X_\beta - X(t)\right|, \\ D_\delta = |C_3 \times X_\delta - X(t)| \tag{17}$$

where C_1, C_2 and C_3 are calculated by

$$X_{i1}(t) = X_a(t) - A_{i1} \times D_a(t), \\ X_{i2}(t) = X_\beta(t) - A_{i2} \times D_\beta(t), \\ X_{i3}(t) = X_\delta(t) - A_{i3} \times D_\delta(t) \tag{18}$$

where X_a, X_β and X_δ are the first three best solutions at iteration t, A_1, A_2 and A_3 are calculated as in Eq. (14), and D_a, D_β and D_δ are defined as Eq. (17).

$$X(t+1) = \frac{X_{i1}(t) + X_{i2}(t) + X_{i3}(t)}{3} \tag{19}$$

- Attacking: The wolves initiate the attacking step as soon as the hunting step has ended (when the prey no longer moves). The exploration and exploitation processes can be mathematically governed by the value of a, which decreases linearly throughout the iteration process. The value of a is updated after every iteration within the range of 2 to 0 using Eq. (16). Exploitation, as per [59], is dedicated to the other half of iterations after a seamless transition from exploration. Wolves move to any random position between their current position and the position of the prey during this phase.

C. THE PROPOSED XGWO ALGORITHM

The standard version of GWO has been implemented successfully in developing and solving different types of optimization problems [60], [61], [62]. However, the exploitation part of GWO depends mainly on selecting the best three solutions, which are called ''Alpha'', ''Beta'', and ''Delta''. The final best solution found so far is the alpha, which remains the same every iteration until the algorithm finds a new better current best solution to be the new alpha. Even though GWO can be easily implemented for numerous purposes, it still suffers from the problems of lacking population diversity, premature convergence, and exploitation-exploration imbalance [63], [64], [65]. Additionally, while the GWO's position update equation is useful for exploitation, it is insufficient to yield a workable solution.

Scholars have modified numerous algorithms in the literature using evolutionary operators, i.e., mutation and crossover [66], [67]. These operators help the search agents to explore and exploit the search space by updating the solutions based on specific mechanisms. Due to the enhancement in the searching performance of optimization algorithms when utilizing any evolutionary operator, this study proposes a special crossover operator, which helps the algorithm to escape the local minima and explore different search regions. Therefore, a new step is added to the traditional GWO algorithm, which is located at the end of each iteration. In another word, the original version of GWO is executed, then, the current best solution – or alpha is enhanced via a crossover operator. The proposed algorithm is called Crossover Grey Wolf Optimizer (XGWO). Moreover, the initialization step of the XGWO is implemented using a discrete chaotic map which is used as the pseudo-random number generator, replacing the uniform

distribution. The steps of the proposed modification are given as follows:

- **Step 1:** Inputs: Get all the required parameters for solving a specific optimization problem, such as the number of search agents (*Size*), and the number of total iterations (*Itr*). In addition, the number of dimensions (*Dim*) is set to 256, which represents the total number of bits in each S-box, i.e., it is equal to 16×16. Therefore, the range for each S-box is [0, 255], which represents the upper and lower boundaries, respectively.
- **Step 2:** Initialization: Generate each search agent randomly in the search space using the discrete chaotic map using Equation (11) instead of randomizing the agents using a uniform distribution.
- **Step 3:** Evaluation function: The fitness of each generated solution is evaluated based on the targeted objective function. In this study, the fitness function is represented by the nonlinearity function, which is given in Eq. 2.

iv **Step 4:** Determine $\boldsymbol{W_{Alpha}}$, $\boldsymbol{W_{Beta}}$, and $\boldsymbol{W_{Delta}}$. In this step, the best wolfs are determined by sorting the solution ascendingly. If there is a new solution with fitness better than the previous, then, keep the new solution as the new W_{Alpha}, otherwise, go to step 6.

- **Step 5:** Solution updating: In this step, the position of each wolf (including the omegas) is updated using Equations (17-19). Then, the fitness value for each search agent is re-evaluated to determine if there is a new current best solution or alpha.
- **Step 6:** Crossover operator: If the current best solution (W^t_{alpha}) is equal to the previous iteration best solution (W^{t-1}_{alpha}),i.e., not updated, then, the crossover operator is executed. The inputs to the crossover operator are the best two search agents, W^t_{alpha} and W^t_{beta}. The proposed crossover works as follows:

i Set S_1 and S_2 as two parents, where the S_1 denotes the W_{alpha}, while S_2 denotes the W_{beta}.

ii Determine S_1^{max} and S_2^{max} which denote the positions of the values higher than or equal 128 in S_1 and S_2 respectively. In addition, determine S_1^{min} and S_2^{min} denote the values lower than to 128 in S_1 and S_2 respectively.

iii Perform an intersection operator between the previously mentioned sets as follows: $S_{P1} = S_1^{max} \cap S_2^{max}$, and $S_{P2} = S_1^{min} \cap S_2^{min}$.

iv Generate two random solutions as offspring (O_1, O_2), in the range $[R_1, R_2]$ where:
$R_1 = \text{Round}(\min(S_1(S_{P2}),S_2(S_{P2}))-Ia)$, and $R_2 = Round(max(S_1(S_{P1}),S_2(S_{P1}))+Ia)$,
$I = max(S_1(S_{P1}),S_2(S_{P1}))-min(S_1(S_{P2}),S_2(S_{P2}))$, and a is a random number in the range [0, 1].

v Replace the position of the repeated generated values with the original values (if any).

vi Evaluate the new generated offspring (O_1, O_2) using the nonlinearity function.

Algorithm 1: S-box generation based XGWO

Inputs: $\#P$ = Size of Population, $\#MaxItr$ = No. of Iterations

1. Initialize the population of the algorithm via Discrete Chaotic Map
2. Evaluate the fitness function for all search agents via nonlinearity function
3. Determine the best three search agents W_{alpha}, W_{beta}, and W_{delta}
4. ***WHILE*** $(Itr < \#MaxItr)$
5. Update the value of a
6. ***For*** w = 1 ***To*** #P // For each wolf in the Pack
8. Update A & C for the three search agents
   $A_1 = 2 \times a \times rand(0,1) - a$ // For W_{alpha}
   $C_1 = 2 \times rand(0,1)$ // For W_{alpha}
   $A_2 = 2 \times a \times rand(0,1) - a$ // For W_{beta}
   $C_2 = 2 \times rand(0,1)$ // For W_{beta}
   $A_3 = 2 \times a \times rand(0,1) - a$ // For W_{delta}
   $C_3 = 2 \times rand(0,1)$ // For W_{delta}
9. Update D for the three search agents
   $D_{alpha} = |C_1 \times W_{alpha} - W_w|$
   $D_{beta} = |C_2 \times W_{beta} - W_w|$
   $D_{delta} = |C_3 \times W_{delta} - W_w|$
10. Calculate X_1, X_2, and X_3
   $X_1 = W_{alpha} - A_1 \times D_{alpha}$
   $X_2 = W_{beta} - A_2 \times D_{beta}$
   $X_3 = W_{delta} - A_3 \times D_{delta}$
11. Update the position of the current search agent W_w
   $W_w = \frac{X_1+X_2+X_3}{3}$
12. ***End For***
13. *W_a = Determine the best search agent so far*
14. *W_b = Determine the second-best search agent so far*
15. ***IF*** W_a=W_{alpha} ***Then***
16. Perform Crossover Operator between W_{alpha} and W_{beta}
   XO (W_{alpha} , W_{beta})
17. ***End IF***
18. Determine the new W_{alpha}, W_{beta}, and W_{delta}
19. ***LOOP***
20. **Outputs :** Return W_{alpha}

FIGURE 1. S-box generation process based on XGWO.

vii Compare the fitness of O_1 and O_2 with the parents S_1 and S_2.

viii Replace the best offspring with the worst parent, if any, as new alpha and beta, otherwise, keep the original W_{alpha} and W_{beta}.

- **Step 7:** Repeating or Stop Criteria: The steps above are repeated until the stop condition is met, which is a fixed number of maximum iterations (*Itr*). If the current iteration t is less than *Itr*, then, go to step 5 and update the positions for the search agents, otherwise, return the position of W_{alpha} and the fitness values as well.

The Pseudo code the XGWO based S-box is presented in Figure 1. While, the complete flowchart of XGWO is presented in Figure 2.

V. EVALUATION OF THE PROPOSED ALGORITHMS

In this study, the analysis focuses on three interrelated objectives:

i To assess the convergence-related performance enhancements of XGWO over GWO for S-box optimization;

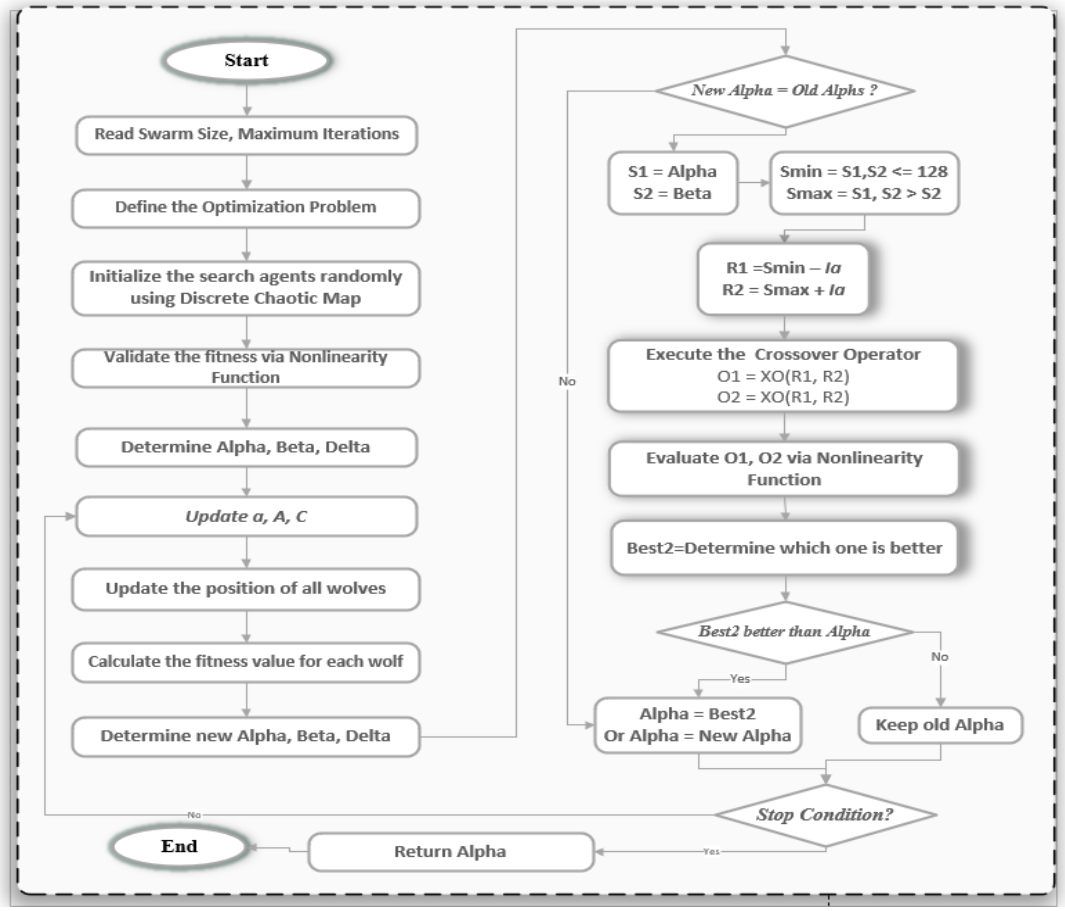


FIGURE 2. The flowchart of XGWO.

ii To analyze the XGWO and GWO-based S-boxes for cryptographic properties in terms of the bijectivity, nonlinearity, SAC, BIC, LP, and differential approximation probability (DP);

iii To compare the generated XGWO and GWO S-boxes with competing metaheuristics that integrate chaotic maps

The proposed enhancement to the GWO in this study was developed and executed on a machine with the following specification: Windows 10, 2.6 GHz Intel Core i7 CPU, 512 GB flash storage and 16 GB 1867 MHz DDR3 RAM. The MATLAB programming language serves as the basis for the implementation of the XGWO and GWO. The parameters for the implementation were set as follows: iterations ($MaxItr$) = 100, and Population size ($PopS$) = 50. Statistical significance was established by running the XGWO and GWO 20 times. Out of the 20 executions, one best S-box configuration was selected to benchmark.

Figure 3 shows the convergence curve for GWO and XGWO, where XGWO achieved better convergence than GWO; furthermore, XGWO has greater convergence and a higher nonlinearity score than GWO. Early in the iteration, where the starting nonlinearity score for XGWO is higher than GWO, the effect of the discrete chaotic map as a component of the population initialization may be observed. Additionally, Table 1 summarize the nonlinearity score for 20 runs of GWO and XGWO, while Table 2 tabulates the appropriate Mann-Whitney U Test results.

Table 2 showed that (H0) is rejected with α < critical value. Hence, XGWO and GWO significantly differed in terms of their average nonlinearity performances.

VI. EVALUATING THE GENERATED S-BOXES

The performance of the predicted S-boxes in terms of security is evaluated and examined in this section. The S-box generated by GWO is shown in Table 3, while the S-box generated by XGWO is shown in Table 4. To understand the cryptographic robustness of the generated S-boxes, security professionals and researchers have used a variety of performance metrics to assess the robustness of S-boxes [7], [29], [50] (see Section III). Six major properties have been taken into consideration to assess the cryptographic properties of the generated S-boxes, such as nonlinearity, bijectivity, strict avalanche criteria (SAC), bit independence criteria (BIC), differential uniformity and linear probability. The next subsection will

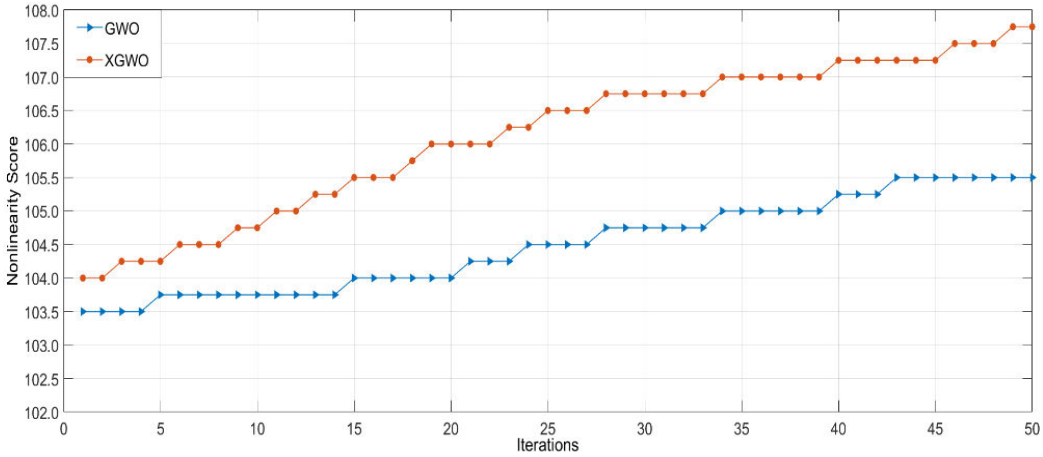


FIGURE 3. Convergence curve for the proposed methods.

TABLE 1. GWO and XGWO nonlinearity score for 20 runs.

Run	Average Nonlinearity Score		Run	Average Nonlinearity Score	
	GWO	XGWO		GWO	XGWO
1	105.75	106.50	**11**	106.75	108.00
2	106.00	107.00	**12**	107.00	108.75
3	106.25	107.25	**13**	107.00	108.50
4	106.25	107.50	**14**	107.25	108.75
5	106.50	107.75	**15**	107.00	108.25
6	106.75	107.50	**16**	106.75	108.50
7	106.00	107.50	**17**	106.50	108.75
8	106.25	107.75	**18**	106.75	108.50
9	106.50	108.00	**19**	107.00	108.75
10	106.25	107.75	**20**	107.25	109.00

briefly explain these well-identified performance criteria for S-boxes.

A. BIJECTIVE CRITERION

Regarding the bijectivity property, we can note that all entries in Table 3 (GWO-generated) and Table 4 (XGWO-generated) for the S-boxes fall within the range of 0 to 255. It is worth noting that each table has distinct and non-repeated values, confirming that both S-boxes satisfy the criteria for bijectivity.

B. NONLINEARITY

Table 5 shows the nonlinearity evaluations of S-boxes produced by GWO and XGWO. The highest, lowest, and mean nonlinearity scores of the generated S-boxes are displayed in Figure 4. For the GWO generated S-box, the nonlinearities were 108, 106, 108, 108, 106, 108, and 106. Given that the lowest nonlinearity value of 106 may be attained using the random approaches, it is clear that premature convergence made it impossible to produce a GWO-based S-box with improved minimal nonlinearity in an acceptable amount of time.

TABLE 2. Mann-whitney U test statistics.

GWO vs XGWO
Confidence level = 95%, critical value = 0.05
Mean Rank GWO = 11.32
Mean Rank XGWO = 29.68
$\alpha = 0.00001$

On the other hand, the nonlinearity scores of the XGWO-generated S-box were 110, 108, 110, 108, 110, 108, 110, and 110, with the lowest nonlinearity score of 108 which was the major aim of the objective function (i.e., Maximize the nonlinearity score). Therefore, the XGWO method that was suggested demonstrated superior performance compared to the traditional GWO method with regards to the maximum, minimum, and average nonlinearity values. It is important to highlight that XGWO has the capability to produce S-boxes that exhibit significant resilience against linear cryptanalysis.

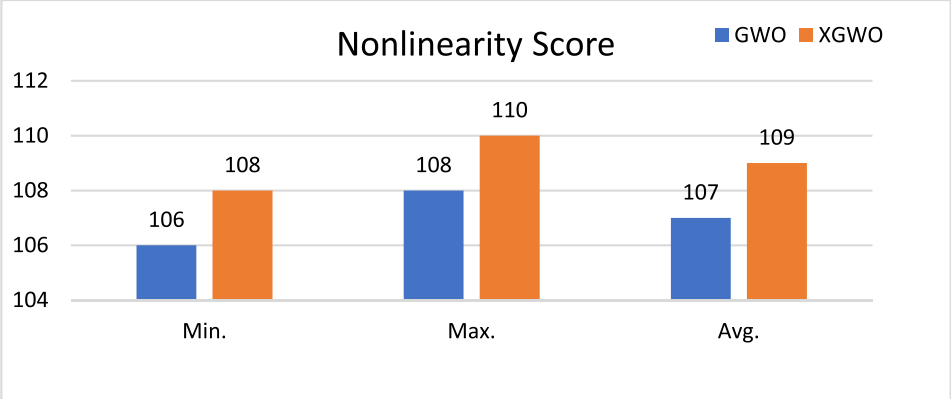


FIGURE 4. Nonlinearity score of the generated S-boxes by GWO and XGWO.

TABLE 3. S-box generated by standard grey wolf optimizer.

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

TABLE 4. S-box generated by crossover grey wolf optimizer algorithm.

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

C. STRICT AVALANCHE CRITERIA

The correlation between input and output bits, known as Strict Avalanche Criterion (SAC), is often used to evaluate S-boxes. The study presented SAC values of the S-boxes generated by the GWO and XGWO methods in Tables 6 and 7. The average SAC values for GWO and XGWO-generated S-boxes

TABLE 5. Nonlinearity comparison of s-boxes.

S-box	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	Min.	Avg.
GWO	108	106	108	108	106	106	108	106	106	107.00
XGWO	110	108	110	108	110	108	110	108	108	109.00

TABLE 6. SAC of the proposed S-box GWO.

0.5938	0.5000	0.5469	0.4219	0.5313	0.5000	0.5000	0.5000
0.5156	0.5156	0.4531	0.5156	0.5156	0.5156	0.5469	0.5313
0.5469	0.5625	0.5313	0.5156	0.5625	0.5313	0.4844	0.4531
0.5000	0.4531	0.5156	0.5781	0.5313	0.5469	0.5000	0.5469
0.5000	0.5625	0.5469	0.5000	0.3906	0.5156	0.4844	0.5156
0.5156	0.4844	0.4375	0.4844	0.5156	0.5313	0.5156	0.5313
0.5000	0.5313	0.5156	0.5000	0.5000	0.5625	0.5313	0.4844
0.4531	0.5313	0.4844	0.5156	0.4375	0.5156	0.5000	0.5313

TABLE 7. SAC of the proposed S-box XGWO.

0.3750	0.5156	0.4688	0.5625	0.4688	0.4688	0.5469	0.4375
0.4844	0.4531	0.5156	0.4531	0.4219	0.5000	0.4531	0.4844
0.4688	0.4531	0.4688	0.4844	0.4375	0.4844	0.5000	0.4844
0.4375	0.5000	0.5469	0.4219	0.5469	0.5000	0.5313	0.5000
0.4375	0.5156	0.5000	0.4688	0.5625	0.5313	0.5313	0.5000
0.5625	0.5156	0.5156	0.5938	0.4688	0.5156	0.4688	0.4844
0.5156	0.4531	0.5313	0.4219	0.5000	0.4844	0.5000	0.5625
0.5938	0.5313	0.5000	0.4844	0.4844	0.4688	0.5156	0.5000

TABLE 8. BIC-nonlinearity of the GWO S-box.

–	102	102	106	104	104	102	98
102	–	106	102	106	104	106	100
102	106	–	106	106	102	104	108
106	102	106	–	102	106	96	106
104	106	106	102	–	102	102	98
104	104	102	106	102	–	102	102
102	106	104	96	102	102	–	106
98	100	108	106	98	102	106	–

were 0.5011 and 0.4936, respectively. Both methods fell just short of the ideal value of 0.5. However, the offset values of the GWO and XGWO-generated S-boxes were 0.02930 and 0.03319, respectively. These values are relatively small, indicating that both methods can produce S-boxes with acceptable SAC properties.

D. BIT INDEPENDENCE CRITERIA

The computation of BIC-nonlinearity values follows a standard procedure. Tables 8 illustrate BIC-nonlinearity values of the Boolean functions for the S-box generated by GWO which has an average value of 103.21 with a minimum value of 96. Meanwhile, the average BIC-SAC value in Table 10 is 0.4995. For XGWO, the BIC-nonlinearity and BIC-SAC values are provided in Table 9 and 11, respectively. The minimum value in Table 9 is 100 whereas the average value is 103.85. Similarly, the average BIC-SAC value in Table 11 for XGWO is 0.505.

The BIC-nonlinearity score, which measures the nonlinearity of S-boxes, was found to be satisfactory for both GWO and XGWO generated S-boxes, with scores of 103.21 and 103.85, respectively. This suggests that the BIC-nonlinearity values for the S-box generated by XGWO were better in terms of the minimum and average values than the S-box generated by GWO. However, the average BIC-SAC value for the S-boxes generated by GWO and XGWO was 0.4995 and 0.505, respectively, indicating that both methods produced S-boxes that closely approached the optimal value of 0.5. As a result, the generated S-boxes met the BIC-SAC criterion.

TABLE 9. BIC-nonlinearity of the XGWO S-box.

–	104	104	104	104	100	104	108
104	–	100	104	104	102	106	104
104	100	–	106	106	104	104	104
104	104	106	–	104	106	100	106
104	104	106	104	–	100	104	106
100	102	104	106	100	–	104	104
104	106	104	100	104	104	–	102
108	104	104	106	106	104	102	–

TABLE 10. BIC-SAC of the proposed S-box by GWO.

–	0.4844	0.4665	0.4821	0.5134	0.5022	0.4933	0.4978
0.529	–	0.4732	0.5022	0.5179	0.4888	0.4911	0.5156
0.4911	0.5156	–	0.4844	0.4777	0.4844	0.4866	0.4978
0.4888	0.5022	0.4777	–	0.5089	0.5201	0.4866	0.5089
0.5446	0.5223	0.5022	0.4955	–	0.4844	0.5022	0.4777
0.5268	0.5112	0.5112	0.4978	0.5268	–	0.4799	0.5022
0.5089	0.5067	0.5000	0.5335	0.5067	0.4933	–	0.5022
0.4955	0.5134	0.4955	0.5045	0.4866	0.4688	0.4866	–

TABLE 11. BIC-SAC of the proposed S-box by XGWO.

–	0.5201	0.4955	0.5134	0.5424	0.5022	0.5089	0.4911
0.5045	–	0.4821	0.4955	0.4866	0.4933	0.4665	0.5223
0.5402	0.5067	–	0.5246	0.529	0.4911	0.5000	0.5201
0.4955	0.5268	0.5000	–	0.4799	0.5134	0.5045	0.4888
0.5089	0.5112	0.5022	0.5268	–	0.5134	0.5402	0.5156
0.5022	0.5179	0.4821	0.5022	0.5156	–	0.4955	0.4888
0.5156	0.5223	0.4933	0.4732	0.5335	0.5022	–	0.4754
0.4978	0.5112	0.5201	0.5112	0.5112	0.5179	0.4821	–

E. DIFFERENTIAL UNIFORMITY

An S-box must have lower differential values in order to resist differential cryptanalysis. The largest differential probability for both GWO and XGWO is 0.03906 or 10/256. (where 10 is the highest value in Table 12 for GWO and Table 13 for XGWO).

In evaluating the Differential Probability (DP) of an S-box using the distribution table, the goal is to minimize the frequency of the maximum value in the table. Based on the results obtained from this evaluation, it can be observed that the GWO method has a frequency of 3 for the maximum entry value, while the XGWO method has a frequency of 9 for the same value. However, both methods proposed in this study have been shown to be effective in producing S-boxes that provide resistance against differential attacks.

F. LINEAR PROBABILITY

For S-boxes to be resistant to linear cryptanalysis, they should have lower LP values. The LP analysis of S-boxes generated by the GWO and XGWO methods is presented in Table 14. The LP score obtained for both S-boxes generated by XGWO and GWO was 0.1172. Therefore, it can be concluded that both LP scores are effective in providing resistance against linear cryptanalysis.

VII. COMPARISON ANALYSIS

To put a current work into perspective, there is a need for a through benchmarking evaluation of the proposed approach against the existing metaheuristic algorithms with chaotic map integration, such as Globalized Firefly Algorithm (GFA) [20], Teaching Learning Based Optimization (TLBO) [42], Standard Firefly Algorithm (SFA) [20], Chaotic Firefly Algorithm (CFA) [44], Genetic Algorithm (GA) [16], Ant Colony Optimization (ACO) [43], Bacterial Foraging Optimization (BFO) [40], Artificial Bee Colony (ABC) [13], Simulated Annealing (SA) [39], Cuckoo Search (CS) [46], Jaya Algorithm (JA) [41], Particle Swarm Optimization (PSO) [49]. The comparative performance of various metaheuristic algorithms with chaotic maps is presented in Table 15, and by examining each major column of the table, several notable observations can be highlighted.

Regarding the column of nonlinearity, it is worth to mention that the proposed XGWO demonstrated superior performance compared to all the competing S-boxes by achieving

TABLE 12. DP For The GWO.

8.0	6.0	8.0	8.0	6.0	6.0	6.0	6.0	6.0	8.0	6.0	8.0	8.0	8.0	6.0	6.0
6.0	6.0	6.0	6.0	6.0	8.0	6.0	6.0	8.0	8.0	8.0	8.0	10.0	6.0	6.0	8.0
4.0	6.0	6.0	8.0	8.0	6.0	8.0	8.0	8.0	6.0	8.0	6.0	6.0	6.0	6.0	8.0
8.0	6.0	6.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0	8.0	8.0	6.0	6.0	6.0	6.0
6.0	6.0	6.0	6.0	10.0	6.0	8.0	8.0	8.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0
6.0	6.0	6.0	8.0	8.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	6.0
6.0	6.0	6.0	6.0	8.0	8.0	8.0	6.0	6.0	8.0	8.0	8.0	6.0	6.0	6.0	8.0
8.0	8.0	6.0	8.0	8.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
6.0	8.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	6.0	6.0	6.0	6.0	4.0	6.0	8.0
6.0	6.0	8.0	6.0	8.0	6.0	8.0	8.0	8.0	4.0	6.0	8.0	8.0	8.0	6.0	8.0
6.0	6.0	6.0	6.0	6.0	8.0	6.0	6.0	8.0	6.0	8.0	8.0	6.0	6.0	6.0	8.0
8.0	8.0	6.0	6.0	8.0	8.0	6.0	8.0	6.0	8.0	6.0	6.0	8.0	6.0	8.0	8.0
6.0	8.0	6.0	8.0	8.0	6.0	8.0	6.0	8.0	8.0	6.0	6.0	4.0	6.0	6.0	6.0
8.0	8.0	8.0	10.0	6.0	6.0	6.0	6.0	8.0	6.0	6.0	6.0	8.0	6.0	8.0	6.0
6.0	8.0	8.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
6.0	6.0	6.0	8.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	6.0	8.0	–

TABLE 13. DP for the XGWO.

6.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	10.0	8.0	6.0	6.0	8.0	8.0	6.0
6.0	8.0	6.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0
8.0	6.0	6.0	6.0	8.0	6.0	6.0	6.0	10.0	8.0	6.0	6.0	6.0	8.0	6.0	8.0
6.0	6.0	8.0	8.0	8.0	8.0	6.0	8.0	6.0	10.0	6.0	6.0	4.0	6.0	6.0	6.0
8.0	6.0	8.0	6.0	6.0	8.0	8.0	6.0	6.0	6.0	6.0	8.0	8.0	10.0	6.0	8.0
6.0	8.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	6.0	6.0
10.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	4.0	6.0	6.0	8.0	8.0	8.0	6.0	6.0
8.0	6.0	6.0	8.0	6.0	8.0	6.0	8.0	6.0	6.0	6.0	6.0	10.0	6.0	8.0	8.0
6.0	6.0	6.0	6.0	8.0	6.0	6.0	6.0	8.0	6.0	8.0	6.0	8.0	8.0	8.0	10.0
6.0	8.0	8.0	8.0	6.0	6.0	6.0	10.0	6.0	6.0	8.0	6.0	6.0	8.0	6.0	6.0
8.0	6.0	6.0	6.0	6.0	10.0	8.0	6.0	6.0	8.0	8.0	6.0	8.0	8.0	6.0	4.0
6.0	8.0	6.0	8.0	6.0	6.0	6.0	6.0	8.0	6.0	6.0	8.0	6.0	6.0	6.0	6.0
6.0	8.0	8.0	6.0	6.0	6.0	8.0	6.0	8.0	6.0	6.0	8.0	6.0	8.0	8.0	6.0
8.0	6.0	4.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	6.0	8.0	6.0	6.0	8.0	6.0
8.0	6.0	6.0	6.0	8.0	8.0	8.0	6.0	8.0	6.0	6.0	6.0	6.0	6.0	8.0	–

the highest average nonlinearity score of 109, this was the main goal of the objective function, which was to maximize the nonlinearity score. The minimum nonlinearity scores for the S-boxes produced by the XGWO, GFA, GA, and PSO approaches were the same. (i.e., the minimum distance value is one way to measure the strength of nonlinearity for an S-box) and higher than all other S-boxes in the comparison. However, the GWO algorithm outperforms the TLBO, SA, and JA algorithms in terms of average nonlinearity. Similarly, GWO, SFA and ACO obtained the same average nonlinearity scores, and lower than the other competing S-boxes. Considering that nonlinearity was used as the objective function for all the methods employed in the comparison. The XGWO approach proves to be efficient in achieving high nonlinearity.

The average values of the all S-boxes generated in the SAC column are reasonably close to the ideal value of 0.5. However, relying solely on the average value can be misleading. The mean of the values in the dependency matrix table

TABLE 14. LP values comparison.

S-box	LP
GWO	0.1172
XGWO	0.1172

can still be around 0.5 even if some of them are not ideal. To obtain a more precise picture, the offset is used, which measures the deviation of each individual value from the ideal value 0.5. Although GWO resulted in a slightly higher offset value than the S-boxes produced by ACO and ABC, which achieved the lowest offset value in the comparison, XGWO still obtained a comparable offset value to the other methods analyzed. Therefore, the SAC properties of the proposed design using GWO-based methods have been shown to be satisfactory.

The BIC-NL and BIC-SAC columns are used to evaluate all S-boxes based on their calculated matrix entries for bit

TABLE 15. comparison of the S-boxes.

	S-Boxes	Nonlinearity			SAC		BIC-NL		BIC-SAC	DP	LP
		Min	*Max*	*Avg*	*Avg*	*Offset*	*Min*	*Avg*	*Avg*	*Max DP*	
Proposed	GWO	106	108	107.00	0.5110	0.02930	96	103.21	0.4995	10	0.1172
	XGWO	108	110	109.00	0.4936	0.03319	100	103.85	0.5059	10	0.1172
Existing Metaheuristic-based S-boxes	GFA [20]	108	110	108.50	0.4915	0.03050	98	103.78	0.5048	10	0.1171
	SFA [20]	106	108	107.00	0.4963	0.02855	102	104.64	0.4974	10	0.1250
	CFA [44]	106	108	107.50	0.4944	0.02855	98	104.35	0.4982	10	0.1250
	TLBO [42]	104	110	106.50	0.4995	0.03198	98	104.57	0.4983	10	0.1172
	GA [16]	108	108	108.00	0.5068	0.03221	96	103.35	0.5017	10	0.1250
	ACO [43]	106	110	107.00	0.5015	0.02831	98	104.21	0.5016	10	0.1171
	ABC [13]	106	110	108.00	0.5073	0.02831	100	104.00	0.5029	10	0.1328
	BFO [40]	106	110	107.50	0.5093	0.03173	94	103.07	0.5029	10	0.1015
	SA [39]	102	106	104.00	0.4980	0.03173	100	103.28	0.4969	10	0.1406
	CS [46]	106	110	108.50	0.4995	0.03271	100	103.85	0.5011	10	0.1093
	JA [41]	104	108	106.25	0.5002	0.03247	96	103.64	0.4996	10	0.1171
	PSO [49]	108	110	108.25	0.4914	0.02904	94	103.50	0.5058	10	0.1250

independence criteria. CFA performs the best in the BIC-NL column with a minimum score of 102 and an average score of 104.64, outperforming other S-boxes. XGWO came in second position behind ABC and was on level with SA and CS in terms of the best BIC-NL minimum score of 100.

However, XGWO only ranks fifth with CS regarding the average BIC-NL score of 103.85. Despite a poor minimum BIC-NL score of 98, TLBO has the second-best average BIC-NL of 104.57, indicating that most entries of BIC-NL TLBO matrix are higher than 100. BFO shows the poorest performance in terms of BIC-NL with the best minimum score of 94 and an average score of 103.07. For BIC-SAC average, all entries have values close to the ideal value of 0.5, indicating similar performances. Therefore, Table 15 demonstrates that the S-boxes generated by the proposed methods closely fulfill the BIC criteria.

Referring to the column that displays the Differential Probability (DP) scores for a particular set of S-boxes. All S-boxes have a DP score of 0.0390 and a maximum I/O of 10. Minimizing the frequency of the maximum I/O is crucial for effective defense against differential attacks. After comparing different optimization algorithms, it has been found that GWO and ACO have the best overall performance with a frequency of 3 for the maximum I/O. The other algorithms have been ranked in ascending order of their frequency of maximum I/O as follows: CFA (7), JA, SFA, TLBO (8), XGWO, BFO, CS (9), PSO and GA (10), ABC (12), and GFA, SA (13). Thus, every approach has the potential to create S-boxes that demonstrate strong resilience against differential attacks.

In the LP (linear probability) score column, GWO and XGWO and TLBO are ranked third with a score of 0.1172, while the top-ranked algorithm is BFO with a score of 0.1015. The other algorithms are ranked in ascending order, with JA, GFA, ACO at 0.1171, SFA, CFA, PSO, and GA at 0.1250, ABC at 0.1328, and SA at 0.1406.

VIII. CONCLUSION

Considering the potentially vast search space, random search approaches and even pure metaheuristic approaches often yield unsatisfactory results. Consequently, recent studies have explored the integration of chaotic maps with metaheuristic-based approaches, and significant performance improvements have been reported in the literature due to the properties of chaotic maps, such as their ergodic behavior and sensitivity to initial conditions. In this study, a discrete chaotic map has been utilized with the Grey Wolf Optimizer (GWO) algorithm as the base algorithm. Furthermore, a new crossover operator has been introduced to enhance the diversification aspect of GWO, resulting in a proposed algorithm called "XGWO." The empirical results of the S-box analysis demonstrate that the overall performance of XGWO is promising. Future work in this area will focus on extending the application of the novel XGWO to other optimization problems, specifically exploring the multi-swarm XGWO approach for dynamic optimization problems where the search space changes over time.

REFERENCES

[1] C. Adams and S. Tavares, "The structured design of cryptographically good S-boxes," *J. Cryptol.*, vol. 3, no. 1, pp. 27–41, 1990.
[2] M. Matsui, "On correlation between the order of S-boxes and the strength of DES," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, in Lecture Notes in Computer Science, vol. 950, 1995, pp. 366–375.
[3] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.
[4] E. F. Brickell, J. H. Moore, and M. Purtill, "Structure in the S-boxes of the DES," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1987, pp. 3–8.
[5] M. Matsui, "Linear cryptanalysis method for DES cipher," *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1993, pp. 386–397.
[6] M. Matsui and A. Yamagishi, "A new method for known plaintext attack of FEAL cipher," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1992, pp. 81–91.
[7] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," in *Advances in Cryptology—EUROCRYPT.* Cham, Switzerland: Springer, vol. 90, 1991, pp. 2–21.

[8] G. Manjula and H. Mohan, "Constructing key dependent dynamic S-box for AES block cipher system," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 613–617.

[9] A. U. Rehman, J. S. Khan, J. Ahmad, and S. O. Hwang, "A new image encryption scheme based on dynamic S-boxes and chaotic maps," *3D Res.*, vol. 7, no. 1, p. 7, Mar. 2016.

[10] S. Murphy and M. J. B. Robshaw, "Key-dependent S-boxes and differential cryptanalysis," *Des., Codes Cryptogr.*, vol. 27, no. 3, pp. 229–255, 2002.

[11] J. S. Khan, J. Ahmad, S. S. Ahmed, H. A. Siddiqa, S. F. Abbasi, and S. K. Kayhan, "DNA key based visual chaotic image encryption," *J. Intell. Fuzzy Syst.*, vol. 37, pp. 2549–2561, Sep. 2019.

[12] G. Ivanov, N. Nikolov, and S. Nikova, "Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties," *Cryptogr. Commun.*, vol. 8, no. 2, pp. 247–276, 2016.

[13] Y. Tian and Z. Lu, "S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm," *J. Syst. Eng. Electron.*, vol. 27, no. 1, pp. 232–241, Feb. 2016.

[14] P. C. Van Oorschot, A. J. Menezes, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.

[15] C. Carlet, "On highly nonlinear S-boxes and their inability to thwart DPA attacks," in *Proc. Int. Conf. Cryptol. India*. Berlin, Germany: Springer, 2005, pp. 49–62.

[16] H. Tao, S. Q. Salih, M. K. Saggi, E. Dodangeh, C. Voyant, N. Al-Ansari, Z. M. Yaseen, and S. Shahid, "A newly developed integrative bio-inspired artificial intelligence model for wind speed prediction," *IEEE Access*, vol. 8, pp. 83347–83358, 2020.

[17] S. Q. Salih, A. Sharafati, I. Ebtehaj, H. Sanikhani, R. Siddique, R. C. Deo, H. Bonakdari, S. Shahid, and Z. M. Yaseen, "Integrative stochastic model standardization with genetic algorithm for rainfall pattern forecasting in tropical and semi-arid environments," *Hydrological Sci. J.*, vol. 65, no. 7, pp. 1145–1157, May 2020.

[18] S. Q. Salih, A. A. Alsewari, and Z. M. Yaseen, "Pressure vessel design simulation: Implementing of multi-swarm particle swarm optimization," in *Proc. 8th Int. Conf. Softw. Comput. Appl.*, Feb. 2019, pp. 120–124.

[19] M. B. Farah, A. Farah, and T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dyn.*, vol. 99, no. 4, pp. 3041–3064, 2020.

[20] H. S. Alhadawi, D. Lambić, M. F. Zolkipli, and M. Ahmad, "Globalized firefly algorithm and chaos for designing substitution box," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102671.

[21] K. Z. Zamli, A. Kader, F. Din, and H. S. Alhadawi, "Selective chaotic maps tiki-taka algorithm for the S-box generation and optimization," *Neural Comput. Appl.*, vol. 33, no. 23, pp. 16641–16658, Dec. 2021.

[22] K. Z. Zamli, "Optimizing S-box generation based on the adaptive agent heroes and cowards algorithm," *Exp. Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115305.

[23] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[24] S. Gupta and K. Deep, "Enhanced leadership-inspired grey wolf optimizer for global optimization problems," *Eng. Comput.*, vol. 36, no. 4, pp. 1777–1800, Oct. 2020.

[25] S. Picek, E. Marchiori, L. Batina, and D. Jakobovic, "Combining evolutionary computation and algebraic constructions to find cryptography-relevant Boolean functions," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Cham, Switzerland: Springer, 2014, pp. 822–831.

[26] C. Carlet, "On the higher order nonlinearities of Boolean functions and S-boxes, and their generalizations," in *Proc. Int. Conf. Sequences Appl.* Cham, Switzerland: Springer, 2008, pp. 345–367.

[27] K. Nyberg, "Differentially uniform mappings for cryptography," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1993, pp. 55–64.

[28] J. Daemen and V. Rijmen, "Specification of Rijndael," in *The Design of Rijndael*. Cham, Switzerland: Springer, 2020, pp. 31–51.

[29] 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.

[30] G. Tang, X. Liao, and Y. Chen, "A novel method for designing S-boxes based on chaotic maps," *Chaos, Solitons Fractals*, vol. 23, no. 2, pp. 413–419, 2005.

[31] 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.

[32] D. Lambić, "A novel method of S-box design based on discrete chaotic map," *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2407–2413, 2017.

[33] F. Özkaynak and A. B. Özer, "A method for designing strong S-boxes based on chaotic Lorenz system," *Phys. Lett. A*, vol. 374, no. 36, pp. 3733–3738, 2010.

[34] M. Khan and Z. Asghar, "A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and S8 permutation," *Neural Comput. Appl.*, vol. 29, no. 4, pp. 993–999, Feb. 2018.

[35] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan, and S. Kaçar, "A novel approach for strong S-box generation algorithm design based on chaotic scaled Zhongtang system," *Nonlinear Dyn.*, vol. 87, no. 2, pp. 1081–1094, 2017.

[36] T. Zhang, C. L. P. Chen, L. Chen, X. Xu, and B. Hu, "Design of highly nonlinear substitution boxes based on I-Ching operators," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3349–3358, Dec. 2018.

[37] T. K. Alshekly, E. A. Albahrani, and S. H. Lafta, "4D chaotic system as random substitution-box," *Multimedia Tools Appl.*, vol. 81, no. 11, pp. 15793–15814, May 2022.

[38] P. Mesejo, O. Ibáñez, O. Cordón, and S. Cagnoni, "A survey on image segmentation using metaheuristic-based deformable models: State of the art and critical analysis," *Appl. Soft Comput.*, vol. 44, pp. 1–29, Jul. 2016.

[39] G. Chen, "A novel heuristic method for obtaining S-boxes," *Chaos, Solitons Fractals*, vol. 36, no. 4, pp. 1028–1036, 2008.

[40] Y. Tian and Z. Lu, "Chaotic S-box: Intertwining logistic map and bacterial foraging optimization," *Math. Problems Eng.*, vol. 2017, pp. 1–11, Nov. 2017.

[41] A. Farah and A. Belazi, "A novel chaotic Jaya algorithm for unconstrained numerical optimization," *Nonlinear Dyn.*, vol. 93, no. 3, pp. 1451–1480, 2018.

[42] T. Farah, R. Rhouma, and S. Belghith, "A novel method for designing S-box based on chaotic map and teaching-learning-based optimization," *Nonlinear Dyn.*, vol. 88, no. 2, pp. 1059–1074, 2017.

[43] M. Ahmad, D. Bhatia, and Y. Hassan., "A novel ant colony optimization based scheme for substitution box design," *Proc. Comput. Sci.*, vol. 57, pp. 572–580, Jan. 2015.

[44] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, "A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7201–7210, May 2018.

[45] A. A. Alzaidi, M. Ahmad, M. N. Doja, E. Al Solami, and M. S. Beg, "A new 1D chaotic map and β-hill climbing for generating substitution-boxes," *IEEE Access*, vol. 6, pp. 55405–55418, 2018.

[46] H. S. Alhadawi, M. A. Majid, D. Lambić, and M. Ahmad, "A novel method of S-box design based on discrete chaotic maps and cuckoo search algorithm," *Multimedia Tools Appl.*, vol. 80, pp. 7333–7350, Oct. 2020.

[47] R. Soto, B. Crawford, F. González Molina, and R. Olivares, "Human behaviour based optimization supported with self-organizing maps for solving the S-box design problem," *IEEE Access*, vol. 9, pp. 84605–84618, 2021.

[48] N. Hematpour and S. Ahadpour, "Execution examination of chaotic S-box dependent on improved PSO algorithm," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5111–5133, May 2021.

[49] H. S. Alhadawi, S. Q. Salih, and Y. D. Salman, "Chaotic particle swarm optimization based on meeting room approach for designing bijective S-boxes," in *Proc. Int. Conf. Emerg. Technol. Intell. Syst.* Cham, Switzerland: Springer, 2021, pp. 331–341.

[50] M. Dawson and S. E. Tavares, "An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1991, pp. 352–367.

[51] C. Adams and S. Tavares, "Good S-boxes are easy to find," in *Proc. Conf. Theory Appl. Cryptol.* Cham, Switzerland: Springer, 1989, pp. 612–615.

[52] A. Webster and S. E. Tavares, "On the design of S-boxes," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 1985, pp. 523–534.

[53] J. Pieprzyk and G. Finkelstein, "Towards effective nonlinear cryptosystem design," *IEE Proc. E-Comput. Digit. Techn.*, vol. 135, no. 6, pp. 325–335, Nov. 1988.

[54] D. Lambić, "A novel method of S-box design based on chaotic map and composition method," *Chaos, Solitons Fractals*, vol. 58, pp. 16–21, Jan. 2014.

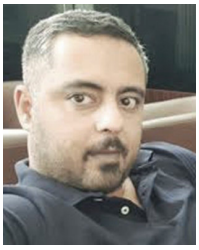
[55] L. Cui and Y. Cao, "A new S-box structure named affine-power-affine," *Int. J. Innov. Comput., Inf. Control*, vol. 3, no. 3, pp. 751–759, Jun. 2007.

[56] D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Proc. Symp. Appl. Math.*, vol. 10, May 1960, pp. 179–193.
[57] D. Lambić, "A new discrete chaotic map based on the composition of permutations," *Chaos, Solitons Fractals*, vol. 78, pp. 245–248, Sep. 2015.
[58] H. S. Alhadawi, M. F. Zolkipli, S. M. Ismail, and D. Lambić, "Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map," *Cryptologia*, vol. 43, no. 3, pp. 190–211, May 2019.
[59] E. Emary, H. M. Zawbaa, and C. Grosan, "Experienced gray wolf optimization through reinforcement learning and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 681–694, Mar. 2018.
[60] M. A. Tawhid and A. M. Ibrahim, "A hybridization of grey wolf optimizer and differential evolution for solving nonlinear systems," *Evolving Syst.*, vol. 11, no. 1, pp. 65–87, Mar. 2020.
[61] A. Bardhan, R. Biswas, N. Kardani, M. Iqbal, P. Samui, M. P. Singh, and P. G. Asteris, "A novel integrated approach of augmented grey wolf optimizer and ANN for estimating axial load carrying-capacity of concrete-filled steel tube columns," *Construction Building Mater.*, vol. 337, Jun. 2022, Art. no. 127454.
[62] J. Adhikary and S. Acharyya, "Randomized balanced grey wolf optimizer (RBGWO) for solving real life optimization problems," *Appl. Soft Comput.*, vol. 117, Mar. 2022, Art. no. 108429.
[63] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Appl. Soft Comput.*, vol. 60, pp. 115–134, Nov. 2017.
[64] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 63–80, Feb. 2018.
[65] Q. Tu, X. Chen, and X. Liu, "Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection," *IEEE Access*, vol. 7, pp. 78012–78028, 2019.
[66] E. H. Houssein, N. Neggaz, M. E. Hosney, W. M. Mohamed, and M. Hassaballah, "Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13601–13618, Oct. 2021.
[67] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "Modified Krill Herd algorithm for global numerical optimization problems," in *Advances in Nature-Inspired Computing and Applications*. Cham, Switzerland: Springer, 2019, pp. 205–221.

ALI IBRAHIM LAWAH received the B.Sc. degree in computer science from the University of Anbar, Iraq, in 2009, and the M.Sc. degree in information and security from the Belarusian State University of Informatics and Radioelectronics, Belarus, in 2019. He is currently pursuing the Ph.D. degree with Altinbas University. His current research interests include designing new optimization algorithms for handling different case studies in information security, such as design strong S-boxes.

ABDULLAHI ABDU IBRAHIM received the bachelor's and master's degrees in computer engineering from Eastern Mediterranean University (North Cyprus) and the Ph.D. degree in electrical and computer engineering from Altinbas University, Turkey. He is currently a Professor with the Department of Electrical and Computer Engineering, Altinbas University.

SINAN Q. SALIH received the B.Sc. degree in information systems from the University of Anbar, in 2010, the M.Sc. degree in computer science from the College of Information Technology, Universiti Tenaga Nasional (UNITEN), in 2012, and the Ph.D. degree in soft computing and intelligent systems from the Faculty of Software Engineering, University of Malaysia Pahang (UMP), in 2019. He is currently a Lecturer with the Technical College of Engineering, Al-Bayan University. He published over 70 research papers (H-index on Google Scholar: 30), and coauthored with more than 50 researchers. His current research interests include optimization algorithms, nature-inspired metaheuristics, machine learning, and feature selection problem for real-world problems.



HUSSAM S. ALHADAWI received the M.Sc. degree in information technology from Universiti Tun Abdul Razak (UNIRAZAK), in 2012, and the Ph.D. degree from Universiti Malaysia Pahang (UMP), Malaysia, in 2018. He is currently a Senior Lecturer and a Senior Researcher in computer engineering with Dijlah University College. His research interests include multimedia security, chaos-based cryptography, cryptanalysis, and optimization techniques. He has served as a reviewer and a technical program committee member for many international conferences. He also served as a referee of some renowned journals, such as *Physica B: Condensed Matter* and IEEE Access.

POH SOON JOSEPHNG received the master's degree in information technology, Australia, the master's degree in business administration, Australia, and the Ph.D. degree in information technology, Australia. He was an Associate Charted Secretary, U.K., with various instructor qualifications, professional certifications, and industry memberships. With his blended technocrat mix of both business senses and technical skills, he has held many multinational corporation senior management positions, global posting, and leads numerous 24 × 7 global mission-critical systems. He has appeared in LIVE television prime time cybersecurity talk show and overseas teaching exposure. His current research interests include strategic IT infrastructure optimization and digital transformation. He was a recipient of the humble young manager nominee twice, five teaching excellence awards, numerous research grants, hundreds of citations, and mentored various students competition awards.