The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 9 (2023) 619–627

www.elsevier.com/locate/icte

Exploiting an Elitist Barnacles Mating Optimizer implementation for substitution box optimization

Kamal Z. Zamli[a,b], Fakhrud Din[c], Hussam S. Alhadawi[d,e], Shah Khalid[c], Hadeel Alsolai[f], Mohamed K. Nour[g], Fahd N. Al-Wesabi[h,*], Muhammad Assam[i]

[a] *Faculty of Computing, Universiti Malaysia Pahang, Pekan, Malaysia*
[b] *Faculty of Science and Technology, Universitas Airlangga, C Campus JI. Dr. H. Soekamo, Mulyorejo, Surabaya 60115, Indonesia*
[c] *Faculty of Information Technology, University of Malakand, KPK, Pakistan*
[d] *Computers Technologies Engineering Department, Dijlah University College, Baghdad, Iraq*
[e] *College of Engineering, University of Warith Al-Anbiyaa, Karbala, Iraq*
[f] *Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia*
[g] *Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia*
[h] *Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia*
[i] *College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*

Received 21 January 2022; received in revised form 11 October 2022; accepted 27 November 2022
Available online 1 December 2022

Abstract

Barnacles Mating Optimizer (BMO) is a new metaheuristic algorithm that suffers from slow convergence and poor efficiency due to its limited capability in exploiting the search space and exploring new promising regions. Addressing these shortcomings, this paper introduces Elitist Barnacles Mating Optimizer (eBMO). Unlike BMO, eBMO exploits the elite exponential probability (P_{elite}) to decide whether to intensify search process via swap operator or to diversify search by randomly exploring new regions. Furthermore, eBMO uses Chebyshev map instead of random numbers to generate quality S-boxes. Experimental results of eBMO on the generation of 8 × 8 substitution-box are competitive against other existing works.

Keywords: Barnacles Mating Optimizer; Substitution-box; Cryptography

1. Introduction

Cryptography relates to the process of converting secret information into dummy data so that it could reach the desired destination without leakage. As part of computer science, cryptography develops efficient schemes for protecting data over computer networks and communication links from any unauthentic modification and revelation [1]. Broadly, cryptography can be grouped into two categories asymmetric cipher and symmetric cipher. The former applies different keys on a block of data for encryption and decryption. Public key encrypts the plain information into cipher text while private key reverts the whole process. Unlike the former, the latter uses same key for encryption and decryption. Here, the substitution box (S-box) plays very significant role in the current practices involving the Data Encryption Standard (DES), the International Data Encryption Algorithm (IDEA) and the Advanced Encryption Standard (AES). It has been proved that poor S-box design allowed attackers to decode DES [2]. This suggests that failure of cryptosystems increases with weak S-boxes. Therefore, robust S-boxes are essential to develop secure and efficient cryptosystems [3]. Typically, the cryptographic strength of any S-box depends on high value of nonlinearity.

To-date, many metaheuristic algorithms combined with chaotic maps are widely used to create cryptographically strong S-box. Each metaheuristic algorithms and its chaotic map integration has interesting characteristics concerning robustness and noise, and as a result, no single algorithm and chaotic map combination can perform better than all others.

[*] Corresponding author.
E-mail address: falwesabi@kku.edu.sa (F.N. Al-Wesabi).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2022.11.005

For this reason, the application of metaheuristic and its integration with chaotic map for S-box optimization is still an open problem.

Barnacles Mating Optimizer (BMO) is a recently developed metaheuristic algorithm that mimics the mating behaviour barnacles [4]. Although showing promising performance in solving general optimization problems, the original BMO algorithm suffers from slow convergence and poor efficiency due to its limited capability in exploiting the search space and exploring new promising regions. Furthermore, in the absence of any prior knowledge about the global optimal solution, BMO uses a stochastic method to update the population of individuals. In this case, useful information from the search space is not guaranteed to be extracted effectively. As a result, this may affect the overall solution diversity to some extent due to the potential uneven distribution of the individual population within the search space. Tackling these shortcomings, this paper discusses a new variant of BMO, termed Elitist BMO (eBMO). Our contributions can be summarized as follows:

- Unlike its predecessor BMO, eBMO exploits the elite exponential probability (P_{elite}) in order to decide whether to intensify its search process via swap operator from its best (elite) candidate or to diversify its search via exploring a new random search neighbourhood.
- Additionally, eBMO also integrates the Chebyshev map as the replacement of its random number generator to enhance the ergodicity and unpredictability of the updated solution.
- eBMO is the first known Barnacles Mating Optimizer based S-box generator. Performance evaluation of eBMO is promising against other competing algorithms for 8 × 8 S-box generation.

The structure of the remaining paper is as follows. Section 2 is about problem description and evaluation criteria as well as related works along with the general description of BMO. Section 3 presents the detailed design of the proposed eBMO algorithm. Section 4 evaluates the S-box generated by eBMO against S-boxes based on existing algorithms. Finally, Section 5 reflects on the hypothesis of this work along with conclusion and the scope for future work.

2. Preliminaries

2.1. Problem description and evaluation criteria

Mathematically, an $m \times n$ S-box is a one-to-one nonlinear mapping $S: GF(2)^m \rightarrow GF(2)^n$ where the S-box S takes m bits as input and generates n bits as output. It is also represented as a multi-input/multi-output Boolean function expressed as: $S(x) = [f_n(x)\, f_{n-1}(x) \ldots f_1(x)\, f_0(x)]$, where these n Boolean functions each in m-variable are defined as: $f_i(x): GF(2)^m \rightarrow GF(2)^n$.

The evaluation criteria comprising bijectivity, nonlinearity, strict avalanche criterion (SAC), bit independence criterion (BIC), differential approximation probability (DP) and linear approximation probability (LP) determine the strength of S-boxes against security attacks [5].

An 8 × 8 S-box qualifies the bijectivity criterion if each of its Boolean function is to be 0/1 balanced, and all 256 entries of S-box are within the 0–255 range and distinct [6]. The nonlinearity of S-box in block ciphers is essential for mitigating linear cryptanalysis. To compute the nonlinearity for an n-bit Boolean function f, the Walsh spectrum (see Eq. (1)) is used.

$$nl(f) = 2^{n-1} - \frac{1}{2}\left(\max_{z \in GF(2)^n} \left|S_f(z)\right|\right) \tag{1}$$

where the Walsh spectrum $S_f(z)$ of Boolean function f is defined as:

$$S_f(z) = \sum_{x \in GF(2)^n} (-1)^{f(x) \bigoplus x \bullet z}$$

where z is from $GF(2)^n$ and $x \bullet z$ is the x and z bitwise dot product. The purpose is to maximize N_f as it is used as an objective function. In case of an 8 × 8 S-box, the optimal nonlinearity value is 112. The AES S-box is shown in Fig. 1 which has nonlinearity value of 112. The way this S-box works with 8 bits input and 8 bits output is as follows. For an input value $(A9)_{16}$ in hexadecimal which is $(10101001)_2$ in binary, the high four bits denote the row i.e., row number $(1010)_2 = (10)_{10}$ and the low four bits denote the column $(1001)_2 = (9)_{10}$. This results in $(D3)_{16} = (11010011)_2$.

A function is said to satisfy SAC if a single input bit alteration leads to 50% change of all output bits. To compute SAC for an S-box, the use of dependence matrix is suggested in literature [7]. According to this method, an S-box satisfies SAC if the dependence matrix for it results a mean value of 0.5.

Bit independence is also an important criterion in the design of strong S-boxes. For an 8 × 8 S-box to quality BIC, each evaluation $f_j \bigoplus f_k\, (j \neq k, 1 \leq j, k \leq 8)$ from its 8 Boolean functions $f_i\, (1 \leq i \leq 8)$ should be highly nonlinear as well as satisfy the avalanche criterion. For BIC verification, it is necessary to calculate the nonlinearity as well as SAC of the Boolean expression $f_j \bigoplus f_k$.

For an S-box, differential cryptanalysis can be established via an imbalanced XOR distribution table. To have an ideal S-box, differential uniformity is essential. Here, all input bits are analysed with ascertaining their uniform mapping to measure the probability. The differential probability measurement in a map f is performed as follows:

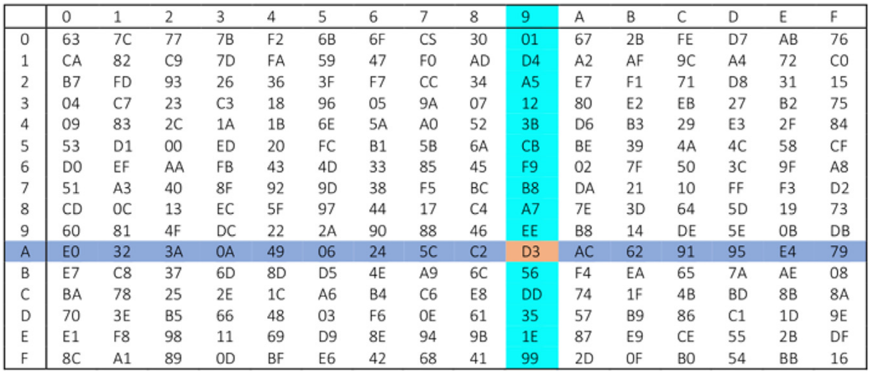
$$DP(f) = \left(\frac{\#\{x \in X | S(x) \bigoplus S\left(x \bigoplus \Delta x\right) = \Delta y\}}{2^m}\right) \tag{2}$$

where X denotes 2^m input combinations.

Linear approximation probability (LP) is largest imbalance value of an event where parities of input bit and output bit selected by masks Γ_x and Γ_y respectively are same. Mathematically, Eq. (3) represents LP as [8]:

$$LP = \max_{\Gamma x, \Gamma y \neq 0} \left|\frac{\#\{x \in X | x \bullet \Gamma x = S(x) \bullet \Gamma y\}ti}{2^m} - \frac{1}{2}\right| \tag{3}$$

where Γ_x masks input, Γ_y masks output and X denotes all 2^m possible inputs. An S-box with lower LP value effectively resists linear cryptanalysis.



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

Fig. 1. AES S-box.

2.2. Related works

S-box design problem being a NP-hard problem is successfully addressed via metaheuristic algorithms in recent literature. Many proposals based on metaheuristic algorithms have been investigated for strong S-box design. Some recent such proposals for 8 × 8 S-boxes optimization are reviewed as follows. Ahmad et al. [9] proposed an S-box generation scheme based on Ant Colony Optimization (ACO) and two chaotic maps namely, logistic map and tent map. In the scheme, these maps generate initial S-boxes followed by their optimization via ACO. The scheme successfully generates S-box with high resistance to linear attacks as compared to other competing S-boxes. Tian and Lu [10] studied the scheme that generates initial solutions as population of S-boxes using a 6-D hyper chaos. Next, Artificial Bee Colony (ABC) is employed with hyperchaotic map to obtain an optimal 8 × 8 S-box. Using similar methodology in their next study, Tian and Lu [11] investigated Bacterial Foraging Optimization (BFO) algorithm for optimization of S-box with intertwining logistic map. Farah et al. [4] proposed an efficient scheme for S-box design using Teaching Learning-based Optimization (TLBO) algorithm and a chaotic map. In another novel scheme by Farah et al. [12], Jaya algorithm with Logistic map, Tent map and Sine Map are combined for S-box generation having reputable evaluation criteria. Alhadawi et al. [13] applied Firefly Algorithm (FA) for an optimal S-box generation from a population of S-boxes obtained initially using a chaotic map with discrete space. The scheme proposed by Zhang et al. [14] generates an optimal S-box via the I-Ching operators inspired by Chinese I-Ching concept. A scheme by Alzaidi et al. [15] applies β-hill climbing algorithm to generate optimized S-box of order 8 × 8. Here, a newly designed discrete-chaotic map obtains an initial S-box as a single candidate solution which undergoes the improved β-hill climbing algorithm for optimization. Recently, Alhadawi et al. [16,17] proposed two schemes based on globalized FA and Cuckoo Search (CS) algorithm and successfully obtained S-boxes with ability to control cryptographic vulnerabilities. A 1-D discrete logistic chaotic map in both the schemes provides an initial population of S-boxes to the optimization algorithms. Moreover, Çavuşoğlu and Kökçam [18] proposed a scheme based on Genetic Algorithm (GA) for generating S-boxes with desired cryptological properties.

Review of some generic optimization-based schemes for creating S-boxes of order $n \times n$ where $(4 \leq n \leq 8)$ is as follows. Millan [19] proposed a scheme based on Hill Climbing (HC) that successfully generated S-box with high nonlinearity. Laskari et al. [20] proposed hybrid scheme based on Differential Evolution (DE) and Particle Swarm Optimization (PSO) applicable to $n \times n$ best S-boxes generation. The scheme by Tesař [21] employed GA with a special cost function and tree searching for optimization of S-boxes. Picek et al. [22] proposed a new cost function as fitness function for standard GA, GA with total tree search (GaT) and Local Search Algorithm (LSA) to generate efficient S-boxes of different dimensions. In a study, Solami et al. [23] investigated a Heuristic Search (HS) technique with hyperchaotic system for generating bijective S-boxes having high quality cryptological features. Recently, Alzaidi et al. [24] studied Sine Cosine Algorithm (SCA) with an improved 1-D chaotic map to optimize S-boxes of order $n \times n$ where $(4 \leq n \leq 8)$. The work by Alhadawi et al. [25] proposed a hybrid technique based on modified PSO, meeting room approach and Tent chaotic map to generate high quality 8 × 8 S-box. In a study by Hematpour and Ahadpour [26], S-box is optimized by new ergodic maps with an enhanced PSO algorithm. Zamli [27] proposed Adaptive Agent Heroes and Cowards (AAHC) algorithm with Tent map to optimize S-box. Zamli et al. [28] also proposed Selective Chaotic Maps with Tiki-Taka Algorithm (SCMTTA) that selects best performing chaotic map from a pool of five chaotic maps to generate optimal S-box. Soto et al. [29] designed optimal S-box by integrating Human Behavior Based Optimization (HBBO) algorithm with Self-organizing Map (SOM). HBBO optimizes S-box whereas SOM solves the premature convergence problem of HBBO. Most recently, Zahid et al. [30] proposed a dynamic S-box design approach based on various modular operations and a heuristic evolution strategy.

Barnacle Mating Optimizer (BMO), proposed by Sulaiman et al. [31], is a new population-based metaheuristic algorithm inspired by how acorn barnacles reproduce in nature. Since its inception, BMO has addressed many real-world problems. For instance, Jia and Sun [32] proposed a novel classification model based on the improved BMO and support vector machine (SVM). Houssein et al. [33] proposed BMO-SVM for gene selection of microarrays cancer classification. Bahasa and Reddy [34] developed a multi-objective opposition-based BMO for optimal configuration of electricity stability. Motivated by these studies and the new features of BMO such as few parameters and low computational cost, this study adapted BMO for S-box optimization.

The genotype sequences of parent barnacles called Dad and Mum are processed to have a genotype for offspring. The three phases of BMO for optimizing a given problem include initialization, selection process and reproduction. In the initialization phase, an array X comprising n solutions simulated as barnacles is created. Mathematically, this array is defined as.

$$X = \begin{bmatrix} X_1^1 & \cdots & X_1^N \\ \vdots & \ddots & \vdots \\ X_n^1 & \cdots & X_n^N \end{bmatrix} \tag{4}$$

where N denotes the number of decision variables and n represents the population size. Each cell i.e., decision variable X^j for $(1 \leq j \leq N)$ of a barnacle X_i for $(1 \leq i \leq n)$ is restricted to upper bound and lower bound expressed as ub and lb, respectively. Finally, the sorting process is applied to place the best barnacle at the top of X.

The second phase of BMO selects parents named Dad and Mum for offspring generation. The main selection criterion for both parents is the size of their penises denoted as pl. Parents with longer pl are selected for matting in this phase. BMO enforces exploitation process via pl-based random selection of an individual barnacle as parent and allows fertilization of a barnacle by only one other barnacle at a time. Exploration in BMO is enforced through sperm cast process which happens when a barnacle selects another barnacle for matting with index greater than its pl. Eq. (5) and Eq. (6) express this selection mathematically.

$$barnacle_D = randperm\,(n) \tag{5}$$

$$barnacle_M = randperm(n) \tag{6}$$

where $barnacle_D$ and $barnacle_M$ are parents that are supposed to mate in population X of size n.

Finally, the Dad and the Mum barnacles produce offspring in the reproduction phase. The genotype frequencies of these parent barnacles are considered based on Hardy–Weinberg principle in generation of offspring. Here, the expected genotype frequencies of two alleles D and M from parents expressed as $f\,(DD) = p^2$, $f\,(MM) = q^2$ (homozygotes) and $f\,(DM) = 2pq$ (heterozygotes) are used to compute genotypes for new offspring. Eq. (7) formally expresses the generation of new barnacle $X_i\,(t+1)$.

$$X_i^{t+1} = p \times X^t_{barnacle_D} + q \times X^t_{barnacle_M} \tag{7}$$

where p is randomly selected from interval [0, 1], q is equal to $1-p$. These two values can be considered as the percentage characteristics that the new offspring X_i^{t+1} inherits from variable $barnacle_D$ *of* Dad and variable $barnacle_M$ of Mum. If $p = 0.4$, then the new offspring gets 40% characteristics from Dad while 60% from Mom.

BMO switches to exploration process termed as sperm cast process if indices of both matting barnacles exceed than the set pl value. Mathematically, Eq. (8) defines this casting process.

$$X_i^{t+1} = rand() \times X^t_{barnacle_M} \tag{8}$$

where $rand()$ returns a random number from interval [0, 1].

3. *Proposed elitist barnacle mating* optimizer

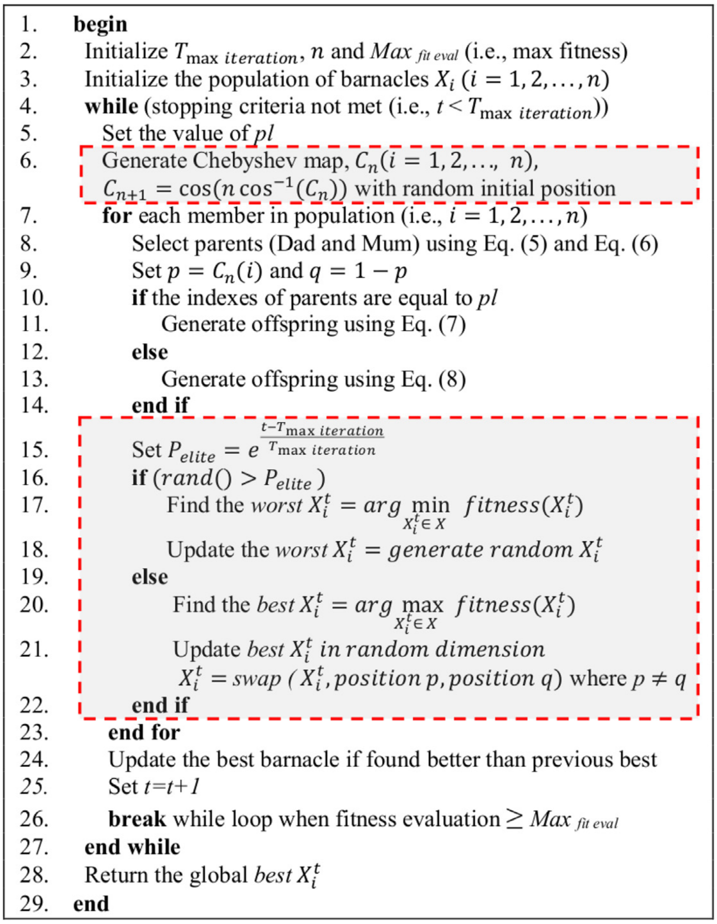
The main additions to the original BMO are highlighted in the dotted line square boxes in Fig. 2. Referring to Fig. 2, the eBMO starts with defining the algorithm's parameters (i.e., $T_{\max\ iteration}$, n, and Max $_{\text{fit eval}}$) and random initialization of the overall population as seen in line 2 till 3. The main iteration starts in line 4. In line 5, eBMO selects the pl value. Meanwhile, in line 6, the Chebyshev map array is generated for C_n $(i = 1, 2, \ldots, n)$ which will be used for the iteration of population. Here, the Chebyshev map will ensure the chaotic values are used for the parameters of p and q. Line 7 until 23 is the start of the iteration of each candidate population. The position update selection is represented in line 8 till 15 based on the current value of pl. In some cases, the position update may cause the current solution's position to be out-of-boundary or duplicated. In the context of S-box, the bijectivity criteria dictates that each item must be uniquely defined within the 0 to 255 range (i.e., with no repetition). Thus, upon each update, the position of each agent (i.e., item is S-box) is checked accordingly. If any item is out-of-boundary or duplicated, the random value from a list of uncovered items will be generated as the replacement. Next, lines 15 till 22 represent the new elitist mechanism introduced in eBMO. This elitist mechanism is controlled by an adaptive and exponential probability called P_{elite} and is given by Eq. (9) as follows:

$$P_{elite} = e^{\frac{t - T_{\max\ iteration}}{T_{\max\ iteration}}} \tag{9}$$

In the early part of the population iteration, the P_{elite} probability is small resulting into eBMO to explore the search space randomly and replacing the current worst population. Towards the end of population iteration, eBMO tends to focus on exploiting the known best candidate solution (i.e., P_{elite} is large) via swapping their respective position in some selected dimensions. Upon successfully undertaking the position update iteration, new $best_{agent}$ is established as depicted in line 24. The iteration continues (see line 25) until Max fit eval has been reached (i.e., in line 26). In the end, the global best agent ($best_{agent}$) will be returned (refer to line 28). Fig. 3 depicts the flowchart of the proposed S-box design scheme.

4. Evaluation of the eBMO S-box

Performance evaluation of eBMO has three related goals. Firstly, the performance of eBMO is compared to its predecessor BMO in terms of convergence and statistical significance. Secondly, the cryptographic properties of the proposed eBMO's S-box are assessed in terms of nonlinearity, bijectivity, strict avalanche criteria (SAC), linear approximation



```
begin
  Initialize $T_{\max\ iteration}$, $n$ and $Max_{fit\ eval}$ (i.e., max fitness)
  Initialize the population of barnacles $X_i$ $(i = 1, 2, \ldots, n)$
  while (stopping criteria not met (i.e., $t < T_{\max\ iteration}$))
    Set the value of $pl$
    Generate Chebyshev map, $C_n (i = 1, 2, \ldots, n)$,
    $C_{n+1} = \cos(n \cos^{-1}(C_n))$ with random initial position
    for each member in population (i.e., $i = 1, 2, \ldots, n$)
      Select parents (Dad and Mum) using Eq. (5) and Eq. (6)
      Set $p = C_n(i)$ and $q = 1 - p$
      if the indexes of parents are equal to $pl$
        Generate offspring using Eq. (7)
      else
        Generate offspring using Eq. (8)
      end if
      Set $P_{elite} = e^{\frac{t - T_{\max\ iteration}}{T_{\max\ iteration}}}$
      if $(rand() > P_{elite})$
        Find the worst $X_i^t = arg \min_{X_i^t \in X} fitness(X_i^t)$
        Update the worst $X_i^t = generate\ random\ X_i^t$
      else
        Find the best $X_i^t = arg \max_{X_i^t \in X} fitness(X_i^t)$
        Update best $X_i^t$ in random dimension
        $X_i^t = swap\ (X_i^t, position\ p, position\ q)$ where $p \neq q$
      end if
    end for
    Update the best barnacle if found better than previous best
    Set $t=t+1$
    break while loop when fitness evaluation $\geq Max_{fit\ eval}$
  end while
  Return the global best $X_i^t$
end
```

Fig. 2. Pseudocode of Elitist Barnacles Mating Optimizer.

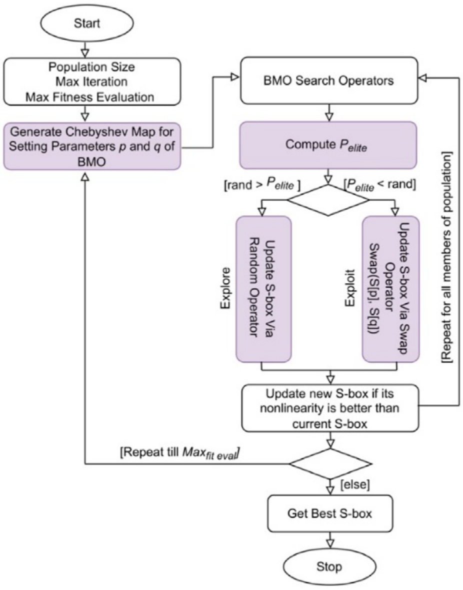

Start
Population Size
Max Iteration
Max Fitness Evaluation
Generate Chebyshev Map for Setting Parameters p and q of BMO
BMO Search Operators
Compute P_{elite}
[rand > P_{elite}]
[P_{elite} < rand]
Explore
Update S-box Via Random Operator
Exploit
Update S-box Via Swap Operator Swap(S[p], S[q])
[Repeat for all members of population]
Update new S-box if its nonlinearity is better than current S-box
[Repeat till $Max_{fit\ eval}$]
[else]
Get Best S-box
Stop

Fig. 3. Flowchart of eBMO for S-box design.

probability (LP), bits independence criteria (BIC) and differential approximation probability (DP). Finally, we benchmark the performance of eBMO against other competing S-box implementations. For implementing eBMO, an experimental setup consisting of a laptop having Windows 10 installed with 16 GB 1867 MHz DDR3 RAM, 2.9 GHz Intel Core i5 CPU and 512 GB flash storage is used. Moreover, eBMO is implemented in Java programming language. The parameters of eBMO are set as: $T_{\max\ iteration} = \infty$, $Max_{pop} = 50$, and $Max_{fit\ eval} = 5000$.

Based on the average nonlinearity score for BMO and eBMO for 20 runs in Table 1, Table 2 highlights the Mann–Whitney statistical analysis for BMO versus eBMO. Statistical result shows that H_0 is rejected with $\alpha <$ critical value. Thus, a significant difference exists between the average nonlinearity performance of eBMO and BMO. Furthermore, eBMO has better convergence than BMO as depicted in Fig. 4. Clearly, eBMO achieves a higher average nonlinearity score than BMO and with faster convergence.

Cryptographic properties of the S-box based on eBMO are highlighted as follows. From Table 3, the eBMO generated S-box fulfils the bijectivity criterion as each entry has unique values from 0 until 255. The nonlinearity score of the eBMO S-box which determines that the S-box fulfils the nonlinearity criterion with a high average score of 109.25. The individual score for each of the 8 Boolean functions within the eBMO S-box are $L_1 = 110$, $L_2 = 108$, $L_3 = 110$, $L_4 = 112$, $L_5 = 110$, $L_6 = 110$, $L_7 = 108$, and $L_8 = 108$. Referring to Table 4,

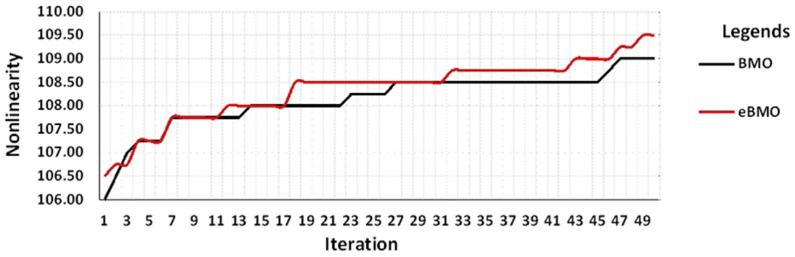


Fig. 4. Flowchart of eBMO for S-box Design.

Table 1
Average nonlinearity score for BMO and eBMO in 20 runs.

Run #	Average nonlinearity score		Run #	Average nonlinearity score	
	BMO	eBMO		BMO	eBMO
1	107.75	108.00	11	107.00	108.50
2	108.00	108.50	12	106.75	108.00
3	107.00	109.00	13	106.75	108.75
4	108.00	109.00	14	108.00	108.50
5	106.50	109.25	15	107.00	109.50
6	107.25	108.00	16	107.50	109.00
7	106.50	109.50	17	108.00	109.50
8	108.00	108.00	18	107.50	108.50
9	106.75	109.25	19	108.00	108.00
10	107.50	108.25	20	108.00	109.50

Table 2
Mann–Whitney U test statistics.

BMO vs eBMO
Confidence level = 95%, critical value =0.05
Mean Rank BMO = 11.38
Mean Rank eBMO =29.62
$\alpha = 0.00001$

the average SAC value =0.4980 is closed to the required score of 0.5 and with good offset of 0.03271. The given score gives a good indication that our eBMO S-box fulfils the SAC criteria. As far as average BIC SAC and average BIC nonlinearity scores shown in Table 4 are concerned, our eBMO S-box obtains commendable scores of 104.21 and 0.5051, respectively. The high average BIC nonlinearity (i.e., greater than 100) and the near middle average of BIC nonlinearity (i.e., close to 0.5) give a clear indication of the fulfilment of the BIC-SAC and BIC-nonlinearity criteria.

Concerning the I/O XOR distribution, the value 10 is the largest among all obtained values. This results the DP score = 10/256 = 0.0390. Please note that the occurrence of maximum entry value is only 7 times highlighting the fulfilment for imbalance of XOR distribution. Finally, for mitigating linear attack, the linear approximation probability (LP) score of the S-box should be as minimum as possible. Typically, S-box with minimum LP better resists linear attack. The linear LP score for the eBMO S-box is 0.1171 as shown in Table 4. The score is considered sufficient to fulfil the linear approximation probability criterion.

Finally, Table 4 also summarizes all the competing S-boxes produced by the proposed scheme and existing metaheuristics. Based on the comparison, several observations can be highlighted here. Firstly, referring to nonlinearity column in Table 4, the generated S-box based on eBMO has the highest average nonlinearity score than all other S-boxes in comparison with the exception of the work of Sine Cosine Algorithm (SCA) [24] with enhanced dynamic chaotic map. On the other note, eBMO S-box does outperform SCA in terms of the maximum nonlinearity score of 112 (i.e., matching theoretical best) although sharing the same minimal score of 108. Secondly, in term of SAC column presented in Table 4, all the S-boxes in the comparison held a comparable SAC value (i.e., which is mostly close to ideal value of 0.50) and relatively small offset swing from the ideal value. Thirdly, regarding the BIC-nonlinearity column in Table 4, the work of [16] on Globalized Firefly Algorithm (GFA) with discrete chaotic map outperforms all other works in comparison with the minimum score of 102. Our S-box based on eBMO comes in as the overall runner up (i.e., minimum score of 100) with Artificial Bee Colony (ABC) [10], Simulated Annealing (SA) [35] and Cuckoo Search (CS) [17]. The rest of the works have the minimum BIC-NL score of less than 100. In terms of the average BIC nonlinearity score, our proposed work is jointly ranked fifth with the score of 104.21 tying up with the work of [9] on Ant Colony Optimization (ACO) but trailing behind the work of [13] on Firefly Algorithm (FA) with the score of 104.35, the work of [4] on Teaching Learning based Optimization (TLBO) with the score of 104.57 and [16] on Globalized Firefly Algorithm (GFA) with the score of 104.65. Fourthly, concerning the BIC-SAC column, our proposed S-box hold a close and comparable average value of 0.5051 against the rest of other S-boxes in Table 4.

Concerning the DP column, we have observed that almost all S-boxes obtain the best max DP score of 10 including our

Table 3
Generated eBMO S-Box.

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

Table 4
Comparison with existing work.

S-box with chaotic maps	Nonlinearity			SAC		BIC-NL		BIC-SAC	DP	LP
	Min	*Max*	*Ave*	*Ave*	*Offset*	*Min*	*Ave*	*Ave*	*Max DP*	
Elitist Barnacle Mating Optimizer (eBMO)	108	112	109.50	0.4980	0.03271	100	104.21	0.5051	10	0.1171
TLBO [4]	104	110	106.50	0.4995	0.03247	98	104.57	0.4983	10	0.1171
SCA [24]	108	110	109.50	0.4985	0.03467	98	104.07	0.5012	10	0.1328
GFA [16]	106	108	107.00	0.4963	0.02855	102	104.64	0.4974	10	0.1250
FA [13]	106	108	107.50	0.4944	0.03686	98	104.35	0.4982	10	0.1250
GA [36]	108	108	108.00	0.5068	0.03221	96	103.35	0.5017	10	0.1250
ACO [9]	106	108	107.00	0.5015	0.02831	98	104.21	0.5016	10	0.1171
ABC [10]	106	110	108.00	0.5073	0.02831	100	104.00	0.5029	10	0.1328
BFO [11]	106	110	107.50	0.5093	0.03173	94	103.07	0.5029	10	0.1015
SA [35]	102	106	104.00	0.4961	0.02196	100	103.28	0.4969	10	0.1406
CS [17]	106	110	108.50	0.4995	0.03271	100	103.85	0.5011	10	0.1093

eBMO S-box. Finally, regarding the LP column, the proposed eBMO S-box is joint third with the score of 0.1171 along with TLBO [4], and ACO [9] trailing behind ABC [11] with the best LP score of 0.1015 and the second best is CS [17] with the LP score of 0.1093. The rest of the S-boxes have the score higher than 0.1171.

5. Conclusion and future work

Several points based on the proposed work are elaborated further here. Firstly, we observe that eBMO generated S-box qualifies all the criteria to be a robust S-box. In fact, in terms of nonlinearity, our S-box is ranked the first along with SCA. Considering other criteria, the results of eBMO S-box are also comparable. For this reason, it can be concluded that our enhancement of BMO with elitism mechanism along with the replacement of its random number generator with the Chebyshev map is useful to allow generation of a cryptographically strong S-boxes that are able to resist linear as well as differential attacks.

Secondly, a more subtle observation can also be highlighted here. Generally, referring to the overall performance in Table 8, metaheuristic-based solution (with chaotic maps) performs better than general computational-based solution (with chaotic maps in terms of nonlinearity criterion and max DP scores.

Thirdly, it should be noted that having a large nonlinear score is not the sole criteria for a cryptographically strong S-box (i.e., for 8 × 8 S-box, the best theoretical value is 112). In our case, eBMO is able to generate the nonlinearity score close to 112, however, with poor S-box properties particularly in terms of strict avalanche criteria (SAC) offsets, bits independence criteria (BIC) and linear approximation probability (LP). We foresee that to effectively deal with more than one criterion; a multi-objective-based solution could be explored further as part of eBMO for S-box optimization.

Finally, though the present work has undertaken S-box optimization and generation as the benchmark for eBMO, the proposed algorithm can be equally adopted for other optimization problems too. As the scope for future work, we hope to adopt our eBMO for other NP-hard optimization problems (including the time tabling problems, travelling salesman problems and search-based software engineering problems) to further demonstrate its performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data sharing does not apply to this article as no datasets were generated.

Acknowledgements

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (18/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R303), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4310373DSR44). This work described in this paper is funded in partial by the xxxxxxxxx and the KPT Fundamental Research Grants: (FRGS/1/2019/ICT02/UMP/02/13) (Formulation of Bi-objective Elitist Dragonfly Algorithm for Constructing Prioritized T-Way Test Cases).

References

[1] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, Int. J. Bifurcation Chaos 8 (06) (1998) 1259–1284, http://dx.doi.org/10.1142/S021812749800098X.

[2] R. Bhanot, R. Hans, A review and comparative analysis of various encryption algorithms, Int. J. Secur. Appl. 9 (2015) 289–306, http://dx.doi.org/10.14257/IJSIA.2015.9.4.27.

[3] A. Razaq, S. Akhter, A. Yousaf, U. Shuaib, M. Ahmad, A group theoretic construction of highly nonlinear substitution box and its applications in image encryption, Multimedia Tools Appl. 81 (3) (2022) 4163–4184, http://dx.doi.org/10.1007/s11042-021-11635-z.

[4] T. Farah, R. Rhouma, S. Belghith, A novel method for designing s-box based on chaotic map and teaching–learning-based optimization, Nonlinear Dynam. 88 (2) (2017) 1059–1074, http://dx.doi.org/10.1007/s11071-016-3295-y.

[5] D.K. Branstad, J. Gait, S. Katzke, Report of the workshop on cryptography in support of computer security, 1977.

[6] J. Detombe, S.E. Tavares, Constructing large cryptographically strong s-boxes, in: Advances in Cryptology, Springer, Berlin, Heidelberg, 1992, http://dx.doi.org/10.1007/3-540-57220-1_60.

[7] A. Webster, S.E. Tavares, On the design of s-boxes, in: Advances in Cryptology, Springer, Berlin, Heidelberg, 1986, http://dx.doi.org/10.1007/3-540-39799-X_41.

[8] M. Matsui, Linear cryptanalysis method for des cipher, in: Workshop on the Theory and Application of Cryptographic Techniques, Springer, Berlin, Heidelberg, 1993, http://dx.doi.org/10.1007/3-540-48285-7_33.

[9] M. Ahmad, D. Bhatia, Y. Hassan, A novel ant colony optimization based scheme for substitution box design, Procedia Comput. Sci. 57 (2015) 572–580, http://dx.doi.org/10.1016/j.procs.2015.07.394.

[10] Y. Tian, Z. Lu, S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm, J. Syst. Eng. Electron. 27 (1) (2016) 232–241, http://dx.doi.org/10.1109/JSEE.2016.00023.

[11] Y. Tian, Z. Lu, Chaotic s-box: Intertwining logistic map and bacterial foraging optimization, Math. Probl. Eng. 2017 (2017) 1–12, http://dx.doi.org/10.1155/2017/6969312.

[12] M.A.B. Farah, A. Farah, F. Tarek, An image encryption scheme based on a new hybrid chaotic map and optimized substitution box, Nonlinear Dynam. 99 (2020) 3041–3064, http://dx.doi.org/10.1007/s11071-019-05413-8.

[13] H.S. Alhadawi, M.F. Zolkipli, M. Ahmad, A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map, Neural Comput. Appl. 31 (2018) 7201–7210, http://dx.doi.org/10.1007/s00521-018-3557-3.

[14] T. Zhang, C.L.P. Chen, L. Chen, X. Xu, B. Hu, Design of highly nonlinear substitution boxes based on i-ching operators, IEEE Trans. Cybern. 48 (12) (2018) 3349–3358, http://dx.doi.org/10.1109/TCYB.2018.2846186.

[15] A.A. Alzaidi, M. Ahmad, M.N. Doja, E. Al Solami, M.S. Beg, A new 1d chaotic map and β-hill climbing for generating substitution-boxes, IEEE Access 6 (2018) 55405–55418a, http://dx.doi.org/10.1109/ACCESS.2018.2871557.

[16] H.S. Alhadawi, D. Lambić, M.F. Zolkipli, M. Ahmad, Globalized firefly algorithm and chaos for designing substitution box, J. Inform. Secur. Appl. 55 (2020) 1–13a, http://dx.doi.org/10.1016/j.jisa.2020.102671.

[17] H.S. Alhadawi, M.A. Majid, D. Lambić, M. Ahmad, A novel method of s-box design based on discrete chaotic maps and cuckoo search algorithm, Multimedia Tools Appl. 80 (2020) 7333–7350b, http://dx.doi.org/10.1007/s11042-020-10048-8.

[18] Ü. Çavuşoğlu, A.H. Kökçam, A new approach to design s-box generation algorithm based on genetic algorithm, Int. J. Bio-Inspired Comput. 17 (1) (2021) 52–62, http://dx.doi.org/10.1504/IJBIC.2021.113360.

[19] W. Millan, How to improve the nonlinearity of bijective s-boxes, in: Australasian Conference on Information Security and Privacy, Springer, 1998, http://dx.doi.org/10.1007/BFb0053732.

[20] E.C. Laskari, G.C. Meletiou, M.N. Vrahatis, Utilizing evolutionary computation methods for the design of s-boxes, in: 2006 International Conference on Computational Intelligence and Security, IEEE, 2006, http://dx.doi.org/10.1109/ICCIAS.2006.295267.

[21] P. Tesař, A new method for generating high non-linearity s-boxes, Radioengineering 19 (1) (2010) 23–26, http://hdl.handle.net/11012/56957.

[22] S. Picek, M. Cupic, L. Rotim, A new cost function for evolution of s-boxes, Evol. Comput. 24 (4) (2016) 695–718, http://dx.doi.org/10.1162/EVCO_a_00191.

[23] E. Al Solami, M. Ahmad, C. Volos, M.N. Doja, M.M.S. Beg, A new hyperchaotic system-based design for efficient bijective substitution-boxes, Entropy 20 (7) (2018) 525, http://dx.doi.org/10.3390/e20070525.

[24] A.A. Alzaidi, M. Ahmad, H.S. Alhadawi, E. Al Solami, Sine cosine optimization-based bijective substitution-boxes construction using enhanced dynamics of chaotic map, Complexity 2018 (2018) 1–16b, http://dx.doi.org/10.1155/2018/9389065.

[25] H.S. Alhadawi, S.Q. Salih, Y.D. Salman, Chaotic Particle Swarm Optimization Based on Meeting Room Approach for Designing Bijective S-Boxes, Springer International Publishing, Cham, 2022, http://dx.doi.org/10.1007/978-3-030-85990-9_28.

[26] N. Hematpour, S. Ahadpour, Execution examination of chaotic s-box dependent on improved pso algorithm, Neural Comput. Appl. 33 (2021) 5111–5133, http://dx.doi.org/10.1007/s00521-020-05304-9.

[27] K.Z. Zamli, Optimizing s-box generation based on the adaptive agent heroes and cowards algorithm, Expert Syst. Appl. 182 (2021) http://dx.doi.org/10.1016/j.eswa.2021.115305.

[28] K.Z. Zamli, A. Kader, F. Din, H.S. Alhadawi, Selective chaotic maps tiki-taka algorithm for the s-box generation and optimization, Neural Comput. Appl. (2021) http://dx.doi.org/10.1007/s00521-021-06260-8.

[29] R. Soto, B. Crawford, F.G. Molina, R. Olivares, Human behaviour based optimization supported with self-organizing maps for solving

the s-box design problem, IEEE Access 9 (2021) 84605–84618, http://dx.doi.org/10.1109/ACCESS.2021.3087139.

[30] A.H. Zahid, A.M. Iliyasu, M. Ahmad, M.M.U. Shaban, M.J. Arshad, H.S. Alhadawi, A.A.A. El-Latif, A novel construction of dynamic s-box with high nonlinearity using heuristic evolution, IEEE Access 9 (2021) 67797–67812, http://dx.doi.org/10.1109/ACCESS.2021.3077194.

[31] M.H. Sulaiman, Z. Mustaffa, M.M. Saari, H. Daniyal, Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020) 103330, http://dx.doi.org/10.1016/j.engappai.2019.103330.

[32] H. Jia, K. Sun, Improved barnacles mating optimizer algorithm for feature selection and support vector machine optimization, Pattern Anal. Appl. 24 (3) (2021) 1249–1274, http://dx.doi.org/10.1007/s10044-021-00985-x.

[33] E.H. Houssein, D.S. Abdelminaam, H.N. Hassan, M.M. Al-Sayed, E. Nabil, A hybrid barnacles mating optimizer algorithm with support vector machines for gene selection of microarray cancer classification, IEEE Access 9 (2021) 64895–64905, http://dx.doi.org/10.1109/ACCESS.2021.3075942.

[34] A.V. Bhasha, B.D.V. Reddy, A multi-objective opposition-based barnacles mating optimization for image super resolution using hyperspectral images, J. Eng. Des. Technol. ahead-of-print (ahead-of-print) (2021) http://dx.doi.org/10.1108/JEDT-01-2021-0030.

[35] G. Chen, A novel heuristic method for obtaining s-boxes, Chaos Solitons Fractals 36 (4) (2008) 1028–1036, http://dx.doi.org/10.1016/j.chaos.2006.08.003.

[36] Y. Wang, K.-W. Wong, C. Li, Y. Li, A novel method to design s-box based on chaotic map and genetic algorithm, Phys. Lett. A 376 (6–7) (2012) 827–833, http://dx.doi.org/10.1016/j.physleta.2012.01.009s.