Supplementary materials for

Mengyu ZHANG, Zhenxue HE, Yijin WANG, Xiaojun ZHAO, Xiaodan ZHANG, Limin XIAO, Xiang WANG, 2025. A power optimization approach for mixed polarity Reed–Muller logic circuits based on multi-strategy fusion memetic algorithm. *Front Inform Technol Electron Eng*, 26(3):415-426. https://doi.org/10.1631/FITEE.2400513

1 Introduction

Swarm intelligence optimization algorithms exhibit characteristics such as fast convergence speed and the ability to escape local optima, making them effective for solving combinatorial optimization problems. For example, Rodrigo Olivares et al. (2023) proposed a set of adaptive parameter control methods using reinforcement learning in particle swarm algorithms to integrate Q-learning into the optimization process of parameter control. D. Martínez-Muñoz et al. (2022) proposed a hybrid algorithm combining the unsupervised learning technique of k-means with a continuous swarm intelligent element heuristic algorithm to optimize steel-concrete composite box girder bridges. Additionally, Yinan Shao et al. (2023), introduced an end-to-end multi-objective neuroevolutionary algorithm to evolve neural networks using genetic manipulation and reward signals, resulting in robust models with simple structures. Compared to traditional optimization algorithms, swarm intelligence optimization algorithms, being probabilistic search algorithms, possess strong robustness and operate on simple mechanisms (D. K. Sambariya and R. Prasad, 2016). Consequently, many scholars opt for swarm intelligence optimization algorithms when addressing combinatorial optimization problems due to their effectiveness and simplicity.

2 Preliminaries

2.1 MPRM expressions

In MPRM circuits, the logic function of the n input variables corresponds to 3^n different polarities. For any polarity P, the corresponding MPRM circuit expression is as follows:

$$f_P(x_1, x_2, ..., x_n) = \odot\prod_{k=0}^{2^n-1}(d_k + S_k), \tag{S1}$$

where P denotes the polarity corresponding to the current MPRM circuit, and the subscript k corresponds to the trinary form $(k_1, k_2, ...k_n)$; the value of d_k ranges from $\{0,1\}$, $d_i = 1$, which means that S_k does not appear in the circuit expression, and $d_i = 0$ means that S_k appears in the circuit expression. $\odot\prod$ indicates the XNOR operation, which can be expressed by the following equation:

$$S_k = \bar{x}_1 + \bar{x}_2 + ... + \bar{x}_n. \tag{S2}$$

2.2 Polarity of variables

A set of values for an input variable is called polarity (Liu L et al., 2023). In the mixed polarity

XNOR/OR expression, each variable $\bar{x}_i$ is expressed with respect to the corresponding polarity P, the ternary form $(P_1,P_2,...,P_n)$ is used to represent the corresponding polarity P. $P_j=0$, variable $\bar{x}_i$ is represented as $x_i, P_j=1$, variable $\bar{x}_i$ is represented as $\tilde{x}_i$, $P_j=2$, variable $\bar{x}_i$ appears as both a proto-variable and an inverse-variable. Thus, the relationship of variable $\bar{x}_i$ with P_n and k_n is given as follows:

$$\bar{x}_i=\begin{cases} x_i, & P_n=0, k_n=0. \\ 0, & P_n=0, k_n=1. \\ \tilde{x}_i, & P_n=1, k_n=0. \\ 1, & P_n=1, k_n=1. \\ x_i, & P_n=2, k_n=0. \\ \tilde{x}_i, & P_n=2, k_n=1. \end{cases} \tag{S3}$$

Example 1：Taking the Boolean logic expression Eq. (S4) as an example, it was converted to expressions with polarity "15 $(120)_3$" and "19 $(201)_3$", respectively, as follows:

$$f(x_0,x_1,x_2)=x_1+x_0x_1+x_0x_1x_2, \tag{S4}$$

$$f_{15}(x_0,x_1,x_2)=(\tilde{x}_0+x_1+x_2)\odot(\tilde{x}_0+\tilde{x}_1), \tag{S5}$$

$$f_{19}(x_0,x_1,x_2)=(\tilde{x}_0+\tilde{x}_2)\odot(\tilde{x}_0+x_1+\tilde{x}_2). \tag{S6}$$

2.3 Polarity conversion algorithms

Existing MPRM expressions need to be transformed into Boolean expressions based on the AND/OR/NOT form. This process is referred to as polarity conversion (Xiong S et al., 2023). In the benchmark circuit, Boolean logic expressions are required by using a polarity conversion algorithm in order to obtain MPRM expressions. The pseudocode for the polarity transformation algorithm to convert a given minterm expression to the polarity P of an MPRM expression is as follows:

Algorithm S1 Polarity Conversion Algorithm

Input: Boolean logic functions. Target polarity *P*
Output: Corresponding MPRM logic function in *P* polarity
express the target polarity *P* in trinary form (P_1, P_2, ..., P_n)
for *i*=1: n **do**
  if P_i==0 **then**
    corresponding minimum terms are differentiated from P_i to produce new terms
  else if P_i==0 || P_i==1 **then**
    if x_i==1 **then**
      x_i=0 // Generate a new item
  else if P_i==2 **then**
    maintaining the original subparagraphs
  end if
// perform the following operations on the resulting new dissimilarity term
  if new dissimilarity term==original term
    delete new dissimilarity and original term
  else
    add the new term to the original formula
  end if
end for

3 Multi-strategy fusion memetic algorithm

3.1 Global optimization algorithm-chimp optimization algorithm

M.Khishe et al. (2020) introduced the ChOA, a novel meta-heuristic approach inspired by the hunting behavior of chimpanzee groups. This algorithm is designed to excel in global search processes, offering robust search capabilities while maintaining simplicity with fewer parameters and a streamlined search procedure. Given its strengths, particularly in exploring diverse solution spaces effectively, the ChOA is selected in this paper as the global optimization algorithm. Its integration aims to enhance the overall search capability of the algorithm, contributing to more efficient and effective solutions.

Among chimpanzee populations, individuals exhibit varying abilities and intelligence, leading to the categorization of individuals into four distinct roles: Attacker, Barrier, Chaser, and Driver. Each group is assigned specific tasks based on their capabilities, primarily involving pursuit and prey encirclement behaviors, as well as attacking prey. Let us denote the total number of chimpanzee individuals in the population as N_1, and the position of the m^{th} individual as X_m. The optimal solution for the population is denoted as X_{Attacker}, followed by the second optimal solution X_{Barrier}, the third optimal solution X_{Chaser}, and finally, the fourth optimal solution X_{Driver}.

The mathematical model for the behavior of a chimpanzee population, encompassing driving, encircling, and chasing prey, can be expressed as follows:

$$d = | c * X_{\text{prey}}(t) - m * X_{\text{chimp}}(t) |, \tag{S7}$$

$$X_{\text{chimp}}(t+1) = X_{\text{prey}}(t) - a * d, \tag{S8}$$

$$a = 2f * r_1 - f, c = 2r_2, \tag{S9}$$

where X_{prey} represents the position of the prey, X_{chimp} denotes the position of an individual chimpanzee. The distance between the chimpanzee and the prey is calculated using Eq.(S7), where r_1, r_2 are random numbers between [0,1]. The variable f serves as the convergence factor, gradually reducing from 2.5 to 0 during the iteration process. The coefficient vector a determines the distance between individual chimpanzee and prey; when $|a| < 1$, chimpanzees aggregate towards their prey, while when $|a| > 1$, they move away, expanding their search range. The attendant weights, denoted by c, are provided by the coefficient vector controlling the chimpanzee's expulsion and pursuit behavior towards the prey. Additionally, the chaotic vector, derived from various chaotic mapping, represents the influence of chimpanzee sexual behavior on the hunting process (Wen HS et al., 2024).

Secondly, by evicting, blocking, and chasing the prey, the chimpanzee population can effectively explore the prey's current location and encircle it. This is achieved by modeling the position of the different chimpanzees within the population, as well as the locations of the remaining chimpanzees, using the following mathematical model:

$$\begin{cases} d_{\text{Attacker}} = | c_1 * x_{\text{Attacker}} - m_1 * x |, x_1 = X_{\text{Attacker}} - a_1 * d_{\text{Attacker}}, \\ d_{\text{Barrier}} = | c_2 * x_{\text{Barrier}} - m_2 * x |, x_2 = X_{\text{Barrier}} - a_2 * d_{\text{Barrier}}, \\ d_{\text{Chaser}} = | c_3 * x_{\text{Chaser}} - m_3 * x |, x_3 = X_{\text{Chaser}} - a_3 * d_{\text{Chaser}}, \\ d_{\text{Driver}} = | c_4 * x_{\text{Driver}} - m_4 * x |, x_4 = X_{\text{Driver}} - a_4 * d_{\text{Driver}}, \end{cases} \tag{S10}$$

$$X(t+1)=\frac{x_1+x_2+x_3+x_4}{4}, \tag{S11}$$

where d_{Attacker}、d_{Barrier}、d_{Chaser} and d_{Driver} represent the distance between the four optimal chimpanzee individuals and their prey. $X(t+1)$ denotes the positional information of the remaining chimpanzee individuals other than the optimal ones. Additionally, c_1, c_2, c_3 and c_4 are all random numbers between [0,1], while m_1, m_2, m_3 and m_4 represent chaotic mapping vectors. The flow of the algorithm is outlined below:

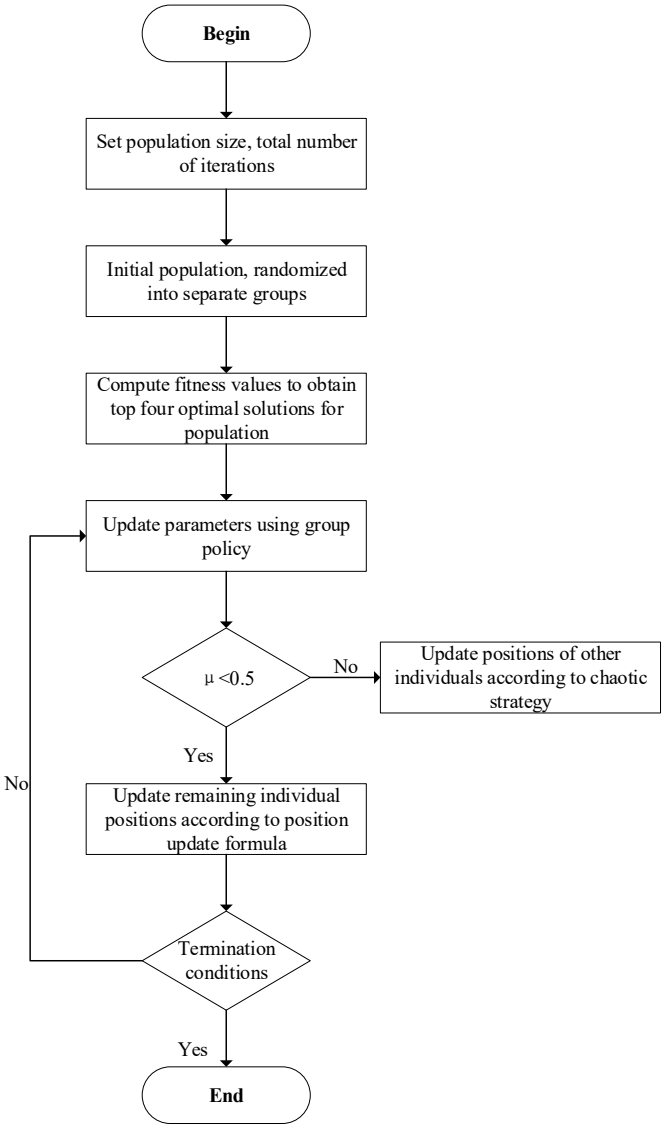

Fig.S1 Flowchart of ChOA

3.2 Local optimization algorithm-Coati Optimization Algorithm

Dehghani Mohammad et al. (2022) introduced the COA, which is tailored to emulate the hunting behavior of coatis. Renowned for its robust evolutionary prowess, rapid convergence, and precise accuracy, this algorithm stands out among other swarm intelligence techniques. Consequently, incorporating the COA as the local optimization method in this paper promises significant enhancements in local search efficiency and overall search accuracy (Shao YN et al., 2023).

Native to the southwestern United States, Mexico, and Central and South America, the coati is a diurnal mammal characterized by a slender head, a flexible and elongated, slightly upturned nose, black paws, and a lengthy, straight tail utilized for signaling and maintaining the balance_features typical of the species (Essam H. Houssein, 2023). As omnivores, coatis primarily consume invertebrates and small vertebrates. Notably, their preferred prey, green iguanas, are frequently located in trees, prompting coatis to hunt them cooperatively in packs.

During their pursuit of prey, coatis must also remain vigilant to evade potential predators, thus dividing their behavior into two primary components: hunting prey, specifically targeting green iguanas concealed in trees, and evading predators by seeking cover from ground-based threats (Chen JN et al., 2024). Assuming a coati population size of N_2 and a total of T iterations, the initial position of individual coatis can be expressed as:

$$X_i : x_{i,j} = lb_j + r(ub_j - lb_j),\ \ i=1,2...N_2,\ j=1,2...m, \tag{S12}$$

where X_i represents the position of the i^{th} individual within the search space, $x_{i,j}$ denotes the value of the j^{th} dimensional decision variable, m indicates the number of decision variables, r is a random number ranging between [0,1], and lb_j and ub_j denote the lower and upper bounds, respectively, of the j^{th} dimensional decision variable.

3.2.1 Hounding the prey

For green iguanas concealed in trees, coatis adopt a strategy where a portion of the population ascends the tree to approach the iguana and provoke it until it descends to the ground. Subsequently, the remaining coatis on the ground engage in attacking and hunting the iguana. This strategic approach enables individual coatis to navigate through the search space. If the location of the optimal individual in the population corresponds to the location of the iguana, the mathematical model of a coati individual climbing a tree and subsequently attacking prey on the ground can be expressed as follows:

$$X_i^{P_1} : x_{i,j}^{P_1} = x_{i,j} + r*(\text{Iguana}_j - I*x_{i,j}), \text{for } i=1,2...[\frac{N_2}{2}] \text{ and } j=1,2...m, \tag{S13}$$

$$\text{Iguana}^G : \text{Iguana}_j^G = lb_j + r*(ub_j - lb_j),\ \ j=1,2...m, \tag{S14}$$

where $X_i^{P_1}$ represents the new position of the i^{th} individual after climbing the tree, obtained from Eq. (S13), Iguana denotes the corresponding position of the iguana in the search space, essentially representing the position of the optimal individual in the population. Iguana^G represents the position of the iguana after descending to the ground, which can be derived from Eq. (S14). Following the position update, individual coatis on the ground choose to update their positions in various ways by comparing the fitness values before and after updating, as depicted in Eq. (S15) below:

$$\begin{aligned} X_i^{P_1} : x_{i,j}^{P_1} &= \begin{cases} x_{i,j} + r*(\text{Iguana}_j^G - I*x_{i,j}), & \text{F}_{\text{Iguana}^G} < F_i, \\ x_{i,j} + r*(x_{i,j} - \text{Iguana}_j^G), & \text{else}, \end{cases} \\ for\ \text{i} &= \left\lfloor \frac{N_2}{2} \right\rfloor + 1, \left\lfloor \frac{N_2}{2} \right\rfloor + 2, ..., N_2 \ \text{ and } j=1,2,...,m, \end{aligned} \tag{S15}$$

where $F_{I\text{guana}^G}$ represents the value of the objective function corresponding to an individual iguana, F_i denotes the value of the objective function corresponding to the updated position of the individual coati. $\lfloor . \rfloor$ denotes the largest integer function, rounding down to the nearest integer. I represents an integer selected from the set {1,2}.

After a portion of the coati population climbs a tree to attack prey, the remaining individuals under the tree waited for the prey to fall for predation, while iguana individuals descend from the tree to randomly selected locations. Consequently, individual coatis on the ground must assess the value of the objective function before making a position update, expressed as follows:

$$X_i = \begin{cases} X_i^{P_1}, & F_i^{P_1} < F_i. \\ X_i, & \text{else.} \end{cases} \tag{S16}$$

3.2.2 Escape from the predator

When faced with their natural predator, individual coatis exhibit evasive behavior by selecting a location near their current position for escape, thereby seeking safety. This behavior not only showcases the coati's ability to exploit localized search but also ensures survival in perilous situations. The mathematical model representing this behavior is as follow:

$$lb_j^{\text{local}} = \frac{lb_j}{t}, ub_j^{\text{local}} = \frac{ub_j}{t}, \text{ where } t=1,2,...,T, \tag{S17}$$

$$\begin{aligned} &X_I^{P_2} : x_{i,j}^{P_2} = x_{i,j} + (1-2*r)*(lb_j^{\text{local}} + r*(ub_j^{\text{local}} - lb_j^{\text{local}})) \\ &i = 1,2...,N_2,\ j=1,2,...,m, \end{aligned} \tag{S18}$$

where $X_i^{P_2}$ denotes the new position obtained by the individual coati after escaping form the predator, lb_j^{local} and ub_j^{local} represent the local lower and upper bounds, respectively, of the j^{th} dimension of the individual coati. To determine whether the chosen new position is better or not, the objective function value is evaluated as follows:

$$X_i = \begin{cases} X_i^{P_2}, & \mathrm{F}_i^{P_2} < F_i. \\ X_i, & \text{else.} \end{cases} \tag{S19}$$

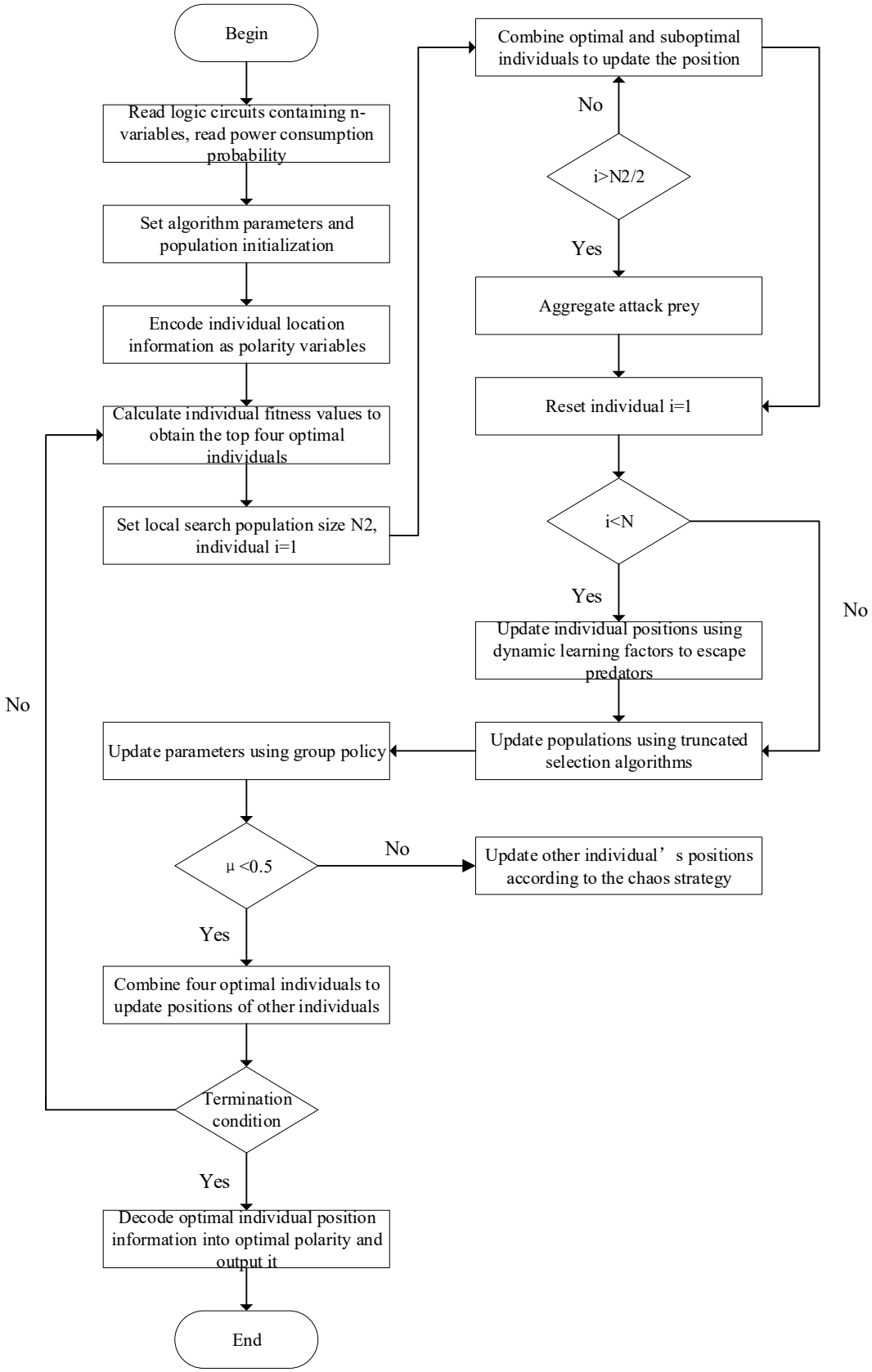

Fig.S2 Flow of MPRM power optimization

4 Experimental results and analysis

4.1 Universal verification

To further assess the universality of the proposed algorithm MFMA, it is compared with six other algorithms: ALO, PSO, WOA, COA, ChOA and DO, using different IEEE CEC function test sets. All experiments are terminated after reaching 1000 iterations. The experiments involve randomly selecting five functions tests for both the IEE CEC 2019 and IEE CEC 2022 test sets. The optimal solutions obtained for different test functions are recorded as the experimental results, as shown in Table S1. The indicators in the table have the same names as those in Table S1, and Std represents the standard deviation calculated based on the experimental results. Analysis of the data reveals that the experimental results obtained by the proposed MFMA on different randomly selected function test sets outperform those of the other four algorithms.

Table S1 Test set experimental results

Set	Function	Standard	New	ALO	PSO	WOA	COA	ChOA	DO
2019	F1	Ave	1	1 376 212	48 428 931	4 502 950	1	1	29 953.03
		Std	3.4E-13	801 861.7	31 359 503	7 532 476	1.1E-11	0	48 913.29
	F2	Ave	4.2613	1720.913	18 719.99	7753.769	4.9215	5	439.4143
		Std	0.019155	1119.02	3367.111	1405.114	0.1641	0	202.7996
	F7	Ave	559.8828	1032.842	1090.276	1377.784	1198.362	2776.087	996.1586
		Std	260.8278	320.8832	291.1024	419.655	199.625	244.3567	223.6925
	F8	Ave	3.8811	4.5911	4.0397	4.6323	4.2002	5.4879	4.0455
		Std	0.1968	0.3186	0.4271	0.3222	0.2311	0.0442	0.4531
	F9	Ave	1.1589	1.3269	1.3329	1.3673	2.4395	6.6797	1.2051
		Std	0.04239	0.1557	0.1530	0.1765	0.6376	0.8048	0.1047
2022	F3	Ave	600.0176	630.1828	642.2374	670.5491	662.1575	718.7611	633.4983
		Std	0.0278	9.1182	8.4740	16.3920	6.8365	7.0671	9.4551
	F4	Ave	830.8869	870.8408	871.4732	912.0467	931.3794	1044.296	897.7686
		Std	5.8856	19.5314	12.9988	18.2469	10.8929	1.0842	38.1582
	F7	Ave	2069.283	2156.842	2129.682	2194.259	2153.256	2536.83	2102.154
		Std	42.4618	61.4261	42.1491	52.9266	26.4329	81.8994	36.5655
	F8	Ave	2226.094	2353.458	2328.312	2305.004	2240.561	16 523.65	2248.481
		Std	7.4519	143.4585	103.9879	55.8133	9.1246	21 791.64	36.0890
	F10	Ave	3062.921	3762.052	3831.31	5236.84	4391.231	8447.579	3263.706
		Std	305.9402	931.0268	973.0773	679.5654	1193.923	495.3369	669.422

4.2 Wilcoxon signed-rank test

To assess the significance of the performance disparity between MFMA and each of the compared algorithms, Table S2 presents the statistical outcomes of the Wilcoxon signed rank test for every algorithm at the 95% significance threshold. In Table S2, the nomenclature for all other

metrics remains consistent with Table S1, where α+ denotes the count of experimental outcomes favoring MFMA over the respective algorithm across various test functions, while α- represents the converse scenario. Analyzing the statistics in Table S2 reveals that MFMA exhibits a higher frequency of α+ compared to the other algorithms, indicating its superior performance relative to the competing algorithms in the Wilcoxon signed rank test.

Table S2 Experimental results of Wilcoxon signed-rank test

Set	Function	Vs. ALO	Vs. PSO	Vs. WOA	Vs. COA	Vs. ChOA	Vs. DO
2019	F1	0.001953	0.001953	0.003906	0.742188	0.4375	0.039063
	F2	0.001953	0.001953	0.003906	0.007813	0.007813	0.007813
	F3	0.105469	0.009766	0.300781	0.023438	0.023438	0.25
	F4	0.193359	0.001953	0.019531	0.007813	0.007813	0.007813
	F5	0.001953	0.769531	0.007813	0.007813	0.007813	0.007813
	F6	0.001953	0.001953	0.164063	0.007813	0.015625	0.460938
	F7	0.003906	0.001953	0.003906	0.007813	0.007813	0.007813
	F8	0.003906	0.492188	0.003906	0.007813	0.007813	0.3125
	F9	0.009766	0.013672	0.011719	0.007813	0.007813	0.25
	F10	0.001953	0.001953	0.027344	0.382813	0.109375	0.039063
2022	F1	0.083984	0.001953	0.097656	0.007813	0.007813	0.742188
	F2	0.001953	0.769531	0.003906	0.007813	0.007813	0.742188
	F3	0.001953	0.001953	0.003906	0.007813	0.007813	0.007813
	F4	0.001953	0.001953	0.003906	0.007813	0.007813	0.007813
	F5	0.845703	0.003906	0.003906	0.148438	0.007813	0.742188
	F6	0.001953	0.431641	0.003906	0.007813	0.007813	0.742188
	F7	0.009766	0.027344	0.003906	0.007813	0.015625	0.039063
	F8	0.005859	0.009766	0.007813	0.015625	0.007813	0.054688
	F9	0.083984	0.001953	0.007813	0.007813	0.007813	0.742188
	F10	0.083984	0.048828	0.003906	0.023438	0.007813	0.109375
	F11	0.001953	0.001953	0.003906	0.007813	0.007813	0.007813
	F12	0.009766	0.001953	0.003906	0.109375	0.007813	0.007813
α+/α-	-	16/6	18/4	19/3	18/4	20/2	18/4

4.3 Convergence comparison

To facilitate a comparative analysis of the performance of the proposed algorithms on different function test sets, six test functions from Table S1 were randomly chosen. The performance of each algorithm on these test functions was calculated, with the average value of each algorithm running ten times on each test function calculated. Convergence curves were then plotted based on the calculated data, as depicted in Fig. S3. In these figures, the vertical axis represents the number of iterations of the algorithm, while the horizontal coordinate represents the average of the function results obtained from ten runs of the algorithm. As observed in the figures, the convergence speed of MFMA is notably superior to that of the other four compared algorithms, and the optimal value of the function obtained is also better. This serves to reaffirm the superiority of the proposed algorithm.

Based on the results of each algorithm run ten times on different test functions, corresponding

box-and-line plots are generated, as shown in Fig. S4. The horizontal axis represents the five algorithms selected for the experiment, while the vertical axis represents the results of each algorithm on the test functions. From the figure, it is evident that MFMA outperforms the other four algorithms on different test functions, achieving more focused and stable results. This further validates the effectiveness of the proposed algorithm.

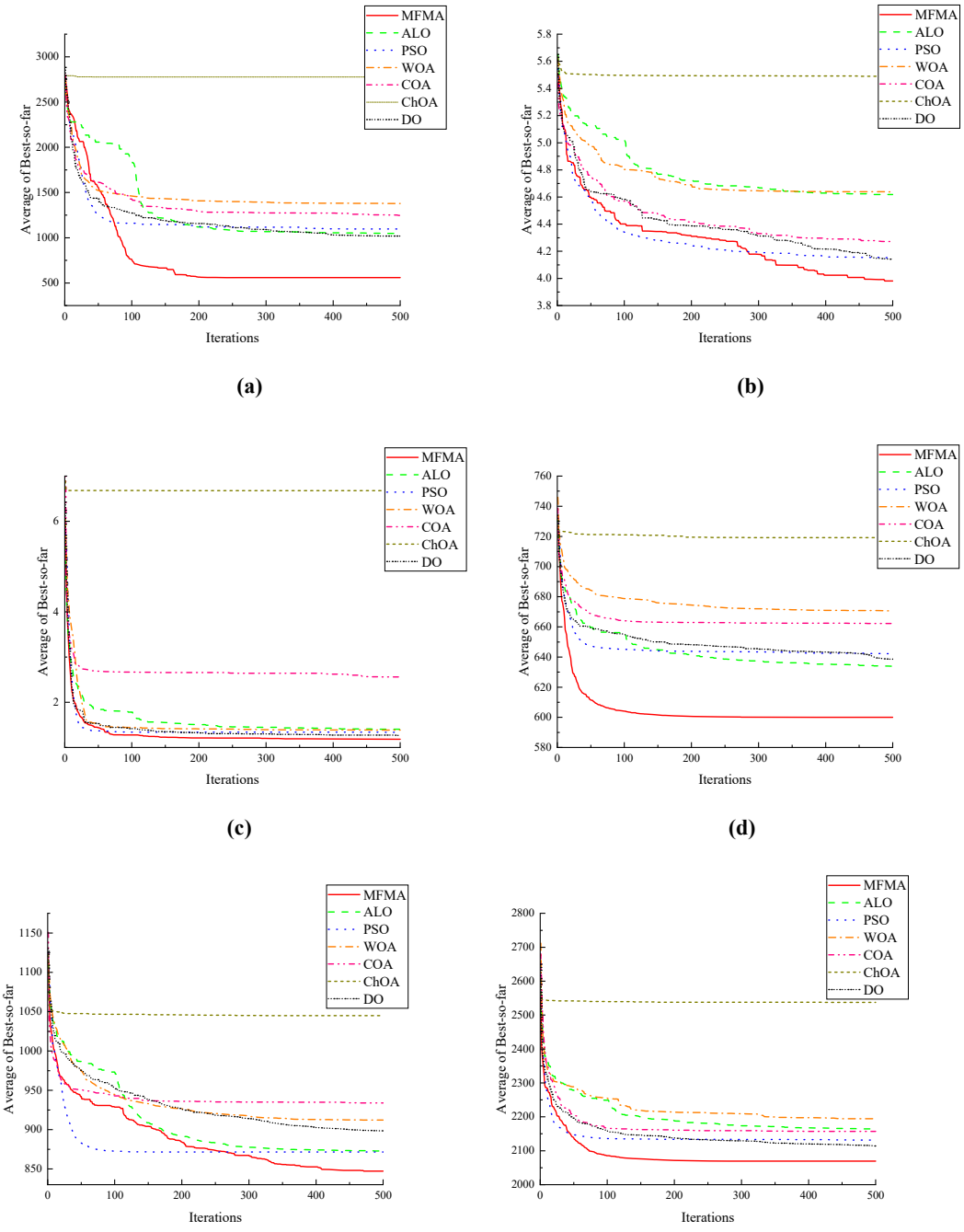

Fig. S3 Convergence performance of optimal value on different test functions: (a) CEC 2019-F7; (b) CEC 2019-F8; (c) CEC 2019-F9; (d) CEC 2022-F3; (e) CEC 2022-F4; (f) CEC 2022-F7

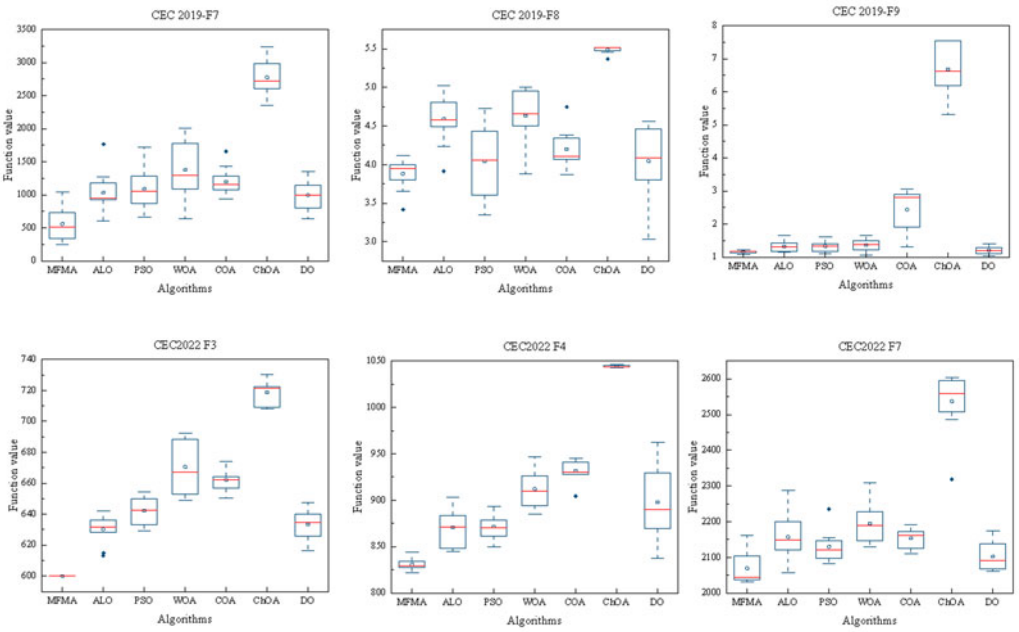

Fig. S4 Experimental data for box plot

References

Rodrigo Olivares, Ricardo Soto, Broderick Crawford, et al., 2023. A Learning—Based Particle Swarm Optimizer for Solving Mathematical Combinatorial Problems[J]. *Axioms*, Vol.12(643): 643

D. Martínez-Muñoz, J. García, J.V. Martí, et al., 2022. Discrete swarm intelligence optimization algorithms applied to steel–concrete composite bridges[J]. *Engineering Structures*, Vol.266: 114607

Yinan Shao, Jerry Chun-Wei Lin, Gautam Srivastava, et al., 2023. Multi-Objective Neural Evolutionary Algorithm for Combinatorial Optimization Problems[J]. *IEEE transactions on neural networks and learning systems*, Vol.34(4): 2133-2143

D. K. Sambariya, R. Prasad, 2016. Application of Bat Algorithm to Optimize Scaling Factors of Fuzzy Logic-Based Power System Stabilizer for Multimachine Power System[J]. *international journal of nonlinear sciences and numerical simulation*, Vol.17(1): 41-53

Liu L, Yuan WH, Liang ZP, et al., 2023. Construction of Polar Codes Based on Memetic Algorithm[J]. *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol.7(5): 1-15

Xiong S, Chen ZP, Jiang N, et al., 2023. Performance Optimization of Multipair Massive MIMO Polarized Relay Systems[J]. *Electronics*, Vol.12(3184): 3184

Khishe, M; Mosavi, M.R, 2020. Chimp optimization algorithm. [J]. *Expert Systems with Applications*, Vol.149: 113338

Wen HS, He X, Huang TW, et al., 2024. Neurodynamic Algorithms with Finite/Fixed-Time Convergence for Sparse Optimization via ℓ1 Regularization[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol.54(1): 131-142

Mohammad Dehghani, 2022. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems[J]. *Knowledge-Based Systems*, Vol.259: 110011

Shao YN, Jerry Chun-Wei Lin, Gautam Srivastava, et al., 2023. Multi-Objective Neural Evolutionary Algorithm for Combinatorial Optimization Problems[J]. *IEEE transactions on neural networks and learning systems*, Vol.34(4): 2133-2143

Essam H. Houssein, 2023. Dynamic Coati Optimization Algorithm for biomedical classification tasks[J]. *Computers in Biology and Medicine*, Vol.164: 107237

Chen JN, Yang YH, Qin ST, 2024. A Distributed Optimization Algorithm for Fixed-Time Flocking of Second-Order Multiagent Systems[J]. *IEEE Transactions on Network Science and Engineering*, Vol.11(1): 152-162