Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

Enhancing the Arithmetic Optimization Algorithm Using Fuzzy Parameter Adaptation and Novel Evolutionary Operators

Mohammad Hosseini[1], Adel Ghazikhani[1,2*], Andres Annuk[3], Mohammad Gheibi[4], Kaveh Akbarzadeh-Sherbaf[1], Reza Moezzi[4,5]

[1]Department of Computer Engineering, Imam Reza International University, Mashhad, Iran
[2]Big Data Lab, Imam Reza International University, Mashhad, Iran
[3]Institute of Forestry and Engineering, Estonian University of Life Sciences, Tartu 51006, Estonia
[4]Institute for Nanomaterials, Advanced Technologies and Innovation, Technical University of Liberec, Studentská 1402/2, Liberec 46117, Czech Republic
[5]Association of Talent Under Liberty in Technology (TULTECH), Sõpruse Pst, Tallinn 10615, Estonia
E-mail: a_ghazikhani@yahoo.com

Received: 4 August 2025; **Revised:** 3 November 2025; **Accepted:** 21 November 2025

Abstract: This paper presents an enhanced version of the Arithmetic Optimization Algorithm (AOA), a mathematically inspired metaheuristic optimization method that has received significant attention in recent literature. The proposed Improved AOA (IAOA) integrates three novel mechanisms to address the limitations of the original AOA. First, a new metric-Dispersion of Solutions (DOS)-is introduced to dynamically estimate the algorithm's convergence behavior, enabling adaptive control over key parameters. Second, a fuzzy parameter adaptation scheme is developed to regulate the Math Optimizer Accelerated (MOA) parameter, providing a flexible and intelligent mechanism for balancing exploration and exploitation. Third, chaotic and triangular search operators are incorporated into the population update process to enhance diversity and prevent premature convergence. Theoretical analysis and extensive empirical evaluations on eighteen benchmark test functions and two real-world engineering optimization problems demonstrate that the proposed IAOA outperforms the original AOA in terms of convergence speed. These results confirm the effectiveness and generalizability of the proposed enhancements.

Keywords: metaheuristic algorithms, Arithmetic Optimization Algorithm (AOA), fuzzy parameter adaptation, chaotic movements, triangular movements

MSC: 03B52, 26E50, 46S40, 68W50, 74P10

1. Introduction

Optimization is an important task with many applications in mathematics, engineering, economics, and other quantitative disciplines. In optimization, an objective function that determines a specific behavior in a system is optimized. In machine learning, the objective function is usually a cost function that defines the machine learning model. This cost function is minimized using the data. In design engineering, everything is designed based on different optimization considerations such as high speed, less weight, etc. In economics, the objective function is the profit or gain of a task.

DOI: https://doi.org/10.37256/cm.7320268141

After defining the optimization problem as an objective function, an optimization algorithm is needed to do the optimization process. The optimization algorithm may go through many steps to find the optimal values of the objective function.

Previous literature on optimization algorithms could be divided into two main categories, 1) Derivative-based optimization and 2) Metaheuristic optimization.

In derivative-based optimization, the optimization process is completely mathematical. In these methods, the optimization problem has two parts, an objective function and the constraints. The function should be optimized considering the constraints [1].

The second category of optimization algorithms is metaheuristic algorithms. In these algorithms, an optimization process is defined based on a specific behavior. For example in Particle Swarm Optimization (PSO), the behavior is finding food in birds [2]. This behavior is broken up into some operators. These operators are executed on a population of solutions that are initially generated randomly.

Several recent surveys review metaheuristic optimization algorithms thoroughly [3, 4]. Previous metaheuristic algorithms could be categorized into five groups [3], a) Swarm/evolution-based algorithms such as Barnacles Mating Optimizer (BMO) [5], Bear Smell Search Algorithm (BSSA) [6], Red Deer Algorithm (RDA) [7], Water Strider Algorithm (WSA) [8], Dwarf Mongoose Optimization (DMO) [9], and Mountain Gazelle Particle Swarm Optimization (MGPSO) [10]; b) Sports/game-based algorithms such as Billiards-Inspired Optimization Algorithm (BIOA) [11], Darts Game Optimizer (DGO) [12], Kho-Kho Optimization Algorithm (KKOA) [13] and Basketball Team Optimization Algorithm (BTOA) [14]; c) Math/physics-based algorithms such as Arithmetic Optimization Algorithm (AOA) [15], Balancing composite Motion Optimization (BCMO) [16], Turbulent Flow of Water-based Optimization (TFWO) [17], Newton Metaheuristic Algorithm (NMA) [18] , Quantum-Inspired Algorithm (QIA) [19] and Artificial Satellite Search (ASS) [20]; d) Human social interaction-based algorithms such as Adolescent Identity Search Algorithm (AISA) [21], Forensic-Based Investigation (FBI) [22], Interactive Autodidactic School (IAS) [23], Political Optimizer (PO) [24] , Human Urbanization Algorithm (HUA) [25], Divine Religions Algorithm (DRA) [26] and Candidates Cooperative Competitive Algorithm (CCCA) [27]; e) Concept/process-based algorithms such as Group Optimization (GO) [28], Lévy Flight Distribution (LFD) [29], Color Harmony Algorithm (CHA) [30] , Rain Optimization Algorithm (ROA) [31] and Success Based Optimization Algorithm (SBOA) [32].

In previous literature, metaheuristic optimization algorithms have been optimized with different methods, one of which is fuzzy logic. Valdez et al. surveyed parameter adaptation in PSO, GSA, and ACO using fuzzy logic [33]. Peraza et al. proposed fuzzy logic for parameter adaptation in a harmony search algorithm [34]. Perez et al. improved the parameter adaptation of the bat algorithm using Interval Type-2 fuzzy logic [35]. Olivas et al. used interval type-2 fuzzy systems for parameter adaptation in ACO [36]. Rodríguez et al. proposed a fuzzy hierarchical operator for Grey Wolf Optimizer [37]. This operator is a hierarchical transformation which has a positive effect on GWO. Castillo proposed a generalized type-2 fuzzy logic system for parameter adaptation in the Bee colony algorithm [38]. Nobile et al. proposed fuzzy logic to calculate inertia, cognitive and social factors, and minimum and maximum velocity independently for each particle [39]. In [40], neural networks and a fuzzy inference system are proposed for parameter adaptation in GSA. Olivas et al. proposed interval type-2 fuzzy logic for parameter adaptation in GSA [41]. Valdez et al. did their second review on parameter adaptation in metaheuristic optimization using fuzzy type-2 logic. He investigated the ant colony, particle swarm, bee colony, bat, firefly, and cuckoo search optimization algorithms in this review. Keivanian et al. proposed two fuzzy inferencing methods to address parameterization and computational cost in Imperialist Competitive Algorithm (ICA) [42]. Melin et al. proposed fuzzy dynamic parameter adaptation for the bird swarm algorithm [43]. Ramachandran et al. proposed a hybrid metaheuristic algorithm combining crow search and artificial bee colony algorithms. In this algorithm, one of the contributions is using fuzzy logic for parameter adaptation in the crow search algorithm [44]. Zhou et al. proposed a hybrid PSO fuzzy inference system for parameter adaptation in ACO [45].

AOA is a metaheuristic algorithm that has gained great attention in the literature [15]. Dhal et al. did a complete survey on different improvements in AOA [46]. They identify nine different strategies in the literature that improve AOA. These strategies are: 1) Hybridization: 2) Opposition-Based Learning: 3) Levy Flight: 4) Chaotic: 5) Other Probability Density Functions: 6) Population-based: 7) Information sharing: 8) Arithmetic operators: and 9) Other strategies. Some of these articles have used fuzzy logic in combination with AOA. Tunc et al. proposed a fuzzy weighting approach along with AOA for cash management optimization on automatic teller machines [47]. Maleknasab et al. proposed a fuzzy controller that is optimized with AOA for Sybil attack detection in vehicular ad hoc networks [48]. Similar to [48], other researchers have proposed AOA for optimization of other fuzzy systems [49–51].

As mentioned earlier, fuzzy parameter adaptation is an important approach for improving metaheuristic algorithms. Among the different improvements to AOA, fuzzy parameter adaptation is still not thoroughly investigated [46]. AOA has an important parameter named Math Optimization Accelerated (MOA) which is set manually and has direct impact on the exploration phase. This parameter is a good candidate for fuzzy parameter adaptation.

Building on previous studies on fuzzy parameter adaptation in metaheuristic algorithms, an Building upon prior research on fuzzy parameter adaptation in metaheuristic algorithms, this study aims to develop an improved version of the AOA that enhances both convergence behavior and solution quality. The proposed enhancements are threefold. First, a novel metric termed Dispersion of Solutions (DOS) is introduced to quantify population diversity and guide the adaptive tuning of algorithmic parameters. Second, a fuzzy logic-based adaptation mechanism is incorporated to dynamically adjust the Math Optimizer Accelerated (MOA) parameter, thereby improving the balance between exploration and exploitation. Third, the population evolution process is enriched with chaotic and triangular motion strategies to increase search diversity and prevent premature convergence. Theoretical analysis and extensive experimental evaluations are conducted to assess the effectiveness and robustness of the proposed algorithm.

The main objectives of this paper are to:

1. Introduce and theoretically justify the DOS metric as a convergence indicator;
2. Develop a fuzzy-based parameter control scheme within the AOA framework;
3. Integrate chaotic and triangular dynamics to enhance population evolution; and
4. Demonstrate the proposed method's performance across diverse benchmark optimization problems.

The remainder of this paper is organized as follows: Section 2 reviews the foundational concepts of AOA and related methods. Section 3 presents the proposed algorithm in detail, along with its theoretical underpinnings. Section 4 reports and discusses the experimental results. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Background and definitions

Arithmetic Optimization Algorithm

The AOA is a metaheuristic algorithm inspired by simple arithmetic operators in mathematics [15]. In this algorithm, the exploration and exploitation phases are selected based on a parameter named MOA. This parameter is calculated with the below function.

$$\mathrm{MOA}\left(C_{\_Iter}\right) = \mathrm{Min} + C_{\_Iter} * \left(\frac{\mathrm{Max} - \mathrm{Min}}{M_{\_Iter}}\right). \tag{1}$$

In equation (1), $M_{\_Iter}$ (Max iteration) is the maximum number of iterations and $C_{\_Iter}$ (Current iteration) is the current iteration of the algorithm. Min and Max are the minimum and maximum values of MOA. Three random numbers are generated, namely r_1, r_2 and r_3. If r_1 is bigger than MOA then the exploration phase is triggered with the below equations

$$x_{i.j}\left(C_{\_Iter}+1\right) = \begin{cases} \mathrm{best}\left(x_j\right) \div \left(\mathrm{MOP}+\varepsilon\right) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) & r_2 < 0.5 \\ \mathrm{best}\left(x_j\right) \times \left(\mathrm{MOP}\right) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) & \text{otherwise.} \end{cases} \tag{2}$$

In equation (2), $x_{i.j}$ is the jth position of the ith solution. best (x_j) is the jth position in the best-obtained solution so far. MOP is calculated with the below function.

$$\mathrm{MOP}\left(C\_{Iter}\right) = 1 - \left(\frac{C\_{Iter}^{\frac{1}{\propto}}}{M\_{Iter}^{\frac{1}{\propto}}}\right). \tag{3}$$

MOP is defined in equation (3) using C_{Iter} (Current iteration) and M_{Iter} (Max iteration) and $\propto$ which is parameter set manually. Back to equation (2), UB_j and LB_j are upper and lower bounds for the jth position of the solution and μ is again a parameter between 0 and 1. On the other hand, if r_1 is smaller than MOA the exploitation phase is triggered with the below equations

$$x_{i.j}\left(C\_{Iter}+1\right) = \begin{cases} \mathrm{best}\left(x_j\right) - (\mathrm{MOP}) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) & r_3 < 0.5 \\ \\ \mathrm{best}\left(x_j\right) + (\mathrm{MOP}) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) & \text{otherwise.} \end{cases} \tag{4}$$

The only difference between the equations of the exploration and exploitation phase is the arithmetic operators before the MOP parameter. In the exploitation phase addition and subtraction is used which causes to search for near points whereas in the exploration phase multiplication and division are used which causes to search for a bigger space.

3. Proposed algorithm

The proposed algorithm is an improvement to the AOA. There are three main contributions which are 1) A new parameter named DOS, 2) A fuzzy inference system for adapting the MOA parameter, and 3) Chaotic and triangular movements for the population evolution. These three contributions are justified in sections 3.1, 3.2 and 3.3. After that the fuzzy rule base, time and space complexity and pseudo code is detailed in section 3.4, 3.5 and 3.6 respectively.

3.1 *DOS*

One of the main objectives of meta-heuristic algorithms is convergence to the optimal solution. If the algorithm is converging, our search strategy should be informed. In AOA, the search strategy is controlled by the MOA parameter. If the algorithm is converging, the search strategy should do more exploitation. Therefore, the MOA parameter should have smaller values.

A parameter named DOS is defined so we could have an estimate of the convergence. To calculate DOS first a normalized value of the fitness of each solutions is calculated with the below equation;

$$Nx_i = \frac{\mathrm{fitness}(x_i) - \mathrm{fitness}(W)}{\mathrm{fitness}(T) - \mathrm{fitness}(W)}. \tag{5}$$

In equation (5), x_i is the ith solution. W is the solution with the worst fitness and T is the solution with the best fitness. Nx_i is a normalized value for x_i which has a value in [0, 1]. After that,

DOS is calculated with the below equation

$$\mathrm{DOS} = \sum_{i=1}^{n}\left(\frac{Nx_i}{n}\right). \tag{6}$$

In equation (6), n is the number of solutions.

In Table 1 a small scenario is presented to demonstrate the effectiveness of DOS. Assume the fitness of the optimal solution is 5, the best solution (T) has fitness = 5 and the worst solution (W) has fitness = 0. Three sample solutions x_1, x_2, and x_3 with different fitness values are determined and DOS is calculated.

Table 1. A small scenario to show the effectiveness of DSA (optimal solution is 5, T = 5 and W = 0)

Fitness (x_1)	Fitness (x_2)	Fitness (x_3)	Nx_1	Nx_2	Nx_3	DOS
0	0	0.5	0	0	0.1	0.03
1	2	3	0.2	0.4	0.6	0.4
4	3	3.5	0.8	0.6	0.7	0.7
5	4	4.5	1	0.8	0.9	0.9
5	5	4.5	1	1	0.9	0.96

From Table 1, it could be concluded that when the fitness values of the solutions are near the optimal solution, DOS is near zero and when they are far from the optimal solution DOS is near one. Therefore, DOS could be used as an estimate of the convergence of the solutions to the optimal solution.

DOS is a parameter defined for measuring the convergence of the solutions to the optimal solution. It is defined by finding the distance of each solution form the worst solution normalized with the distance between the best and worst. These distances are then added up together to calculate DOS.

3.2 *Fuzzy inference system for MOA adaptation in AOA*

As mentioned in Section 1, fuzzy inference systems are a well-studied method for parameter adaptation in metaheuristic algorithms [33–36]. A fuzzy inference system is proposed for the MOA parameter in AOA. MOA is the main parameter in AOA, which determines the search strategy in every iteration. In the original AOA, MOA is determined with equation (1). In this research, a Fuzzy Inference System is designed for MOA adaptation (Figure 1).

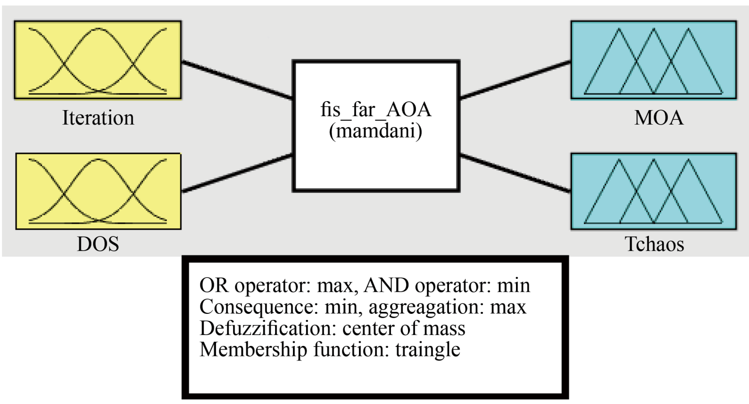


Figure 1. Proposed fuzzy inference system

The proposed Fuzzy Inference System (FIS) has two inputs and two outputs. The first input is the iteration number. The membership function of this input is shown in Figure 2.

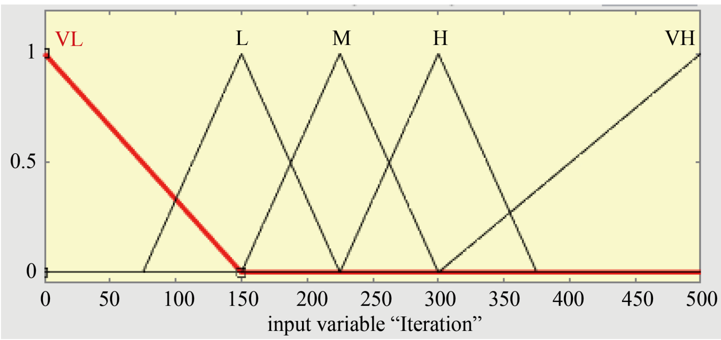


Figure 2. Membership function for the iteration variable

The reason iteration number was selected as input is that in equation 3, MOA is calculated based on it. This variable has five fuzzy values Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH).

The second input is the DOS which was explained in section 3.1. The membership function of DOS is shown in Figure 3.

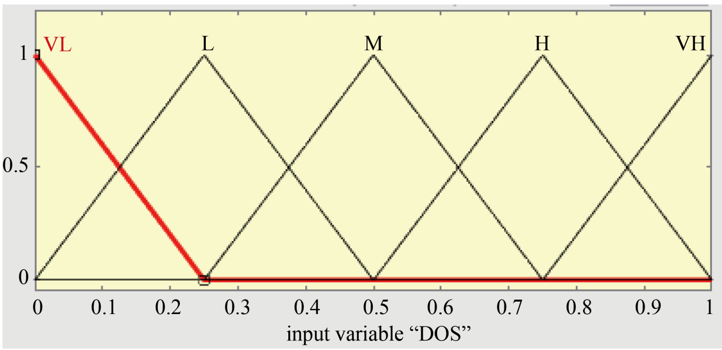


Figure 3. Membership function for the DOS variable

This input was selected due to its ability to estimate the convergence of solutions to the optimal solution. DOS has a direct effect on MOA. This variable has five fuzzy values Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH).

The proposed FIS has two outputs. The first output is MOA. The membership function of MOA is shown in Figure 4. This variable has five fuzzy values Very Small (VS), Small (S), Medium (M), Big (B) and Very Big (VB).

The second output is the number of solutions with chaotic movements. This is detailed in the next section. This variable has four fuzzy values Very Low (VL), Low (L), Medium (M), High (H).

An FIS is defined with DOS and iteration number as the inputs and MOA and number of solution with chaotic movements as the outputs. This FIS helps adapting MOA which was manually configured in AOA.

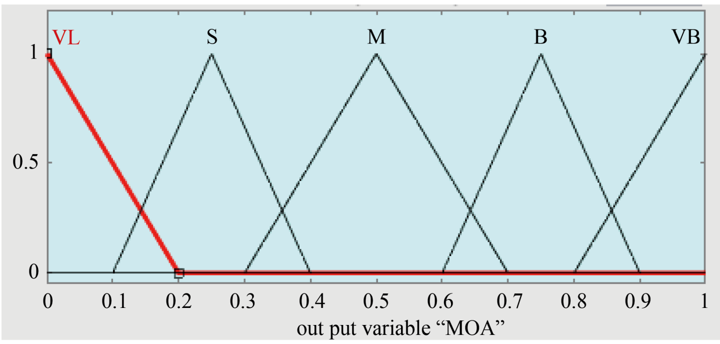


Figure 4. Membership function for the MOA variable

3.3 *Chaotic and triangular movements of the population*

In the original AOA, the population has normal movements. To improve the exploration and exploitation ability of the algorithm, chaotic and triangular movements have been proposed.

3.3.1 *Chaotic movements of the population*

In AOA, the exploration phase is handled with division and multiplication operators on the best solution. Chaotic movements could improve the exploration ability of AOA. Chaotic enhancements are usual in metaheuristic optimization [52]. They could balance between exploration and exploitation and help escape local minima. Chaotic movements are usually preferred to other enhancements such as Levy flights and opposition-based learning [53, 54].

Based on previous literature the chaotic enhancement for improved exploration after initialization is not done on all solutions of the population [52]. Here, a maximum of 1/2 of the population may have chaotic movements. The number of solutions with chaotic movements is determined in the proposed FIS.

At the beginning of the algorithm, the number of solutions with chaotic movements is more and they are selected from the worst solutions. As the algorithm goes further and the number of iterations increases, according to the DOS parameter, when the solutions are converging the number of solutions with chaotic movement reduces until they reach zero.

As mentioned in the previous section the number of solutions with chaotic movements is one of the outputs of the FIS. The membership function of this output (NChaos) is shown in Figure 5.

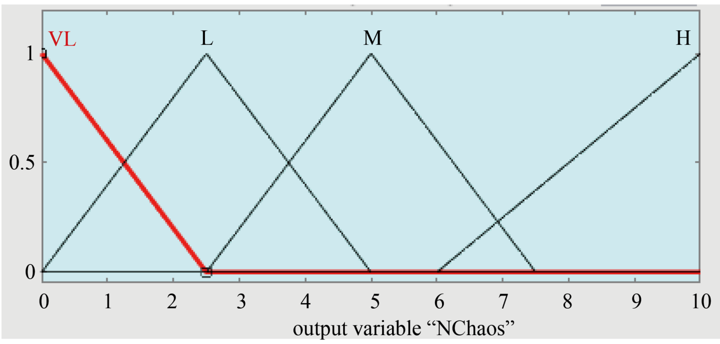


Figure 5. Membership functions for the number of solutions with chaotic movements

The solutions that are selected for chaotic movements are updated using the logistic map below:

$$x_{n+1} = \lambda x_n (1 - x_n) \text{ for } 0 < \lambda \leq 4n. \tag{7}$$

In Equation (8), x_n is a position in one of the solutions [55].

3.3.2 *Triangular movements of the population*

In the proposed algorithm, triangular movements based on the Sine Cosine Algorithm [56] are added to AOA. This type of movement improves both the exploration and exploitation ability of AOA. It could also help escaping local minima. Sine and cosine could also help changing the moving pattern.

In the proposed algorithm sine and cosine are added to both the exploitation and exploration phases in equations (2) and (4).

$$x_{i.j}(C\_{Iter} + 1) = \begin{cases} \text{best}(x_j) \div \sin(r_4) \times (\text{MOP} + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j) & r_2 < 0.5 \\ \text{best}(x_j) \times \cos(r_4) \times (\text{MOP}) \times ((UB_j - LB_j) \times \mu + LB_j) & \text{otherwise.} \end{cases} \tag{8}$$

$$x_{i.j}(C\_{Iter} + 1) = \begin{cases} \text{best}(x_j) - \sin(r_4) \times (\text{MOP}) \times ((UB_j - LB_j) \times \mu + LB_j) & r_3 < 0.5 \\ \text{best}(x_j) + \cos(r_4) \times (\text{MOP}) \times ((UB_j - LB_j) \times \mu + LB_j) & \text{otherwise.} \end{cases} \tag{9}$$

Equations (9) and (10) are the updated exploration and exploitation phases respectively. In the original AOA, there are three random parameters, r_1, r_2, and r_3. In equations (9) and (10) a new random parameter named r_4 is defined which has a value in $[0,\ 2\pi]$. This parameter is used as an input to sin and cos which are responsible for the triangular movements.

Chaotic and triangular movements were added to AOA to improve the exploration and exploitation capabilities. Chaotic movements were performed with a logistic function in Equation (8). The number of solutions with chaotic movements is set by the FIS. The triangular movements were added to Equations (9) and (10).

3.4 *Fuzzy rule base and specifications*

In this section, the rule base of the FIS is detailed. This rule base is proposed based on the behavior of the inputs and outputs influence on each other. These rules are shown in Table 2. These rules are defined based on the influence of the iteration and DOS inputs on the MOA and NChaos parameter. In lower iterations MOA and NChaos has higher values to force the algorithm to go into the exploration phase. But this behavior is changed when DOS has high values. The opposite happens in higher iterations.

Table 2. FIS rule base

Rule No.	Iteration	DOS	MOA	*NChaos*
1	VL	VH	VB	H
2	VL	H	B	H
3	VL	M	M	M
4	VL	L	S	L
5	VL	VL	VS	VL
6	L	VH	VB	H

Table 2. (cont.)

Rule No.	Iteration	DOS	MOA	*NChaos*
7	L	H	VB	H
8	L	M	M	M
9	L	L	S	L
10	L	VL	S	VL
11	M	VH	VB	M
12	M	H	M	M
13	M	M	M	M
14	M	L	M	L
15	M	VL	VS	VL
16	H	VH	S	M
17	H	H	S	M
18	H	M	S	L
19	H	L	S	L
20	H	VL	VS	VL
21	VH	VH	S	L
22	VH	H	S	L
23	VH	M	VS	VL
24	VH	L	VS	VL
25	VH	VL	VS	VL

The proposed FIS has the following specifications (Table 3).

Table 3. FIS specifications

Parameter	Value	Parameter	Value
FIS type	Mamdani	OR operator	Max
Membership functions	Triangle	Consequence	Min
Rule weight	No weight	Aggregation	Max
AND operator	Min	Defuzzification	Center of mass

3.5 *Time and space complexity*

In this section, the time and space complexity of AOA and IAOA is analyzed. In Table 4, some notation is defined for determining the time and space complexity.

Based on the notation defined in Table 4, in AOA the main operation in every iteration is the position update. Therefore the time complexity for all iterations would be $O(I*N*D)$. The space complexity of AOA is again related to the space complexity of population update, which is $O(N*D)$.

For IAOA, the FIS has a time complexity of F, the DOS calculation and chaotic enhancements has a time complexity of D. Therefore the time complexity would be $O(I*N*D+I*F+I*D)$. If $O(I*F)$ doesn't have a high value, the time complexity of AOA and IAOA are the same. The space complexity of AOA and IAOA are the same.

Table 4. Test functions for evaluation

Variable	Notation
Population size	N
Problem dimension	D
Iterations	I
Cost of fitness	C

3.6 *Flowchart and psuedocode of the proposed algorithm*

In this section, the flowchart and pseudo-code of the proposed algorithm is presented. Figure 6 shows the flowchart.

Algorithm 1 Pseudocode of the Improved AOA (IAOA) algorithm

```
Initialize the parameters α, μ
Initialize the solutions randomly.
while (C_Iter < M_Iter) do
    // Calculations before population update
    Calculate the fitness function for the solutions.
    Find the best solution
    Update DOS using Eq. (7).
    Update MOA and NChaos using the FIS. (Figure 1)
    Update MOP using Eq. (3).
    // Usual movements with embedded triangular movements
    for (i = 1 to solution_num) do
        for (j = 1 to solution_dim) do
            Generate three random numbers (r1, r2 and 17 r3) between [0, 1] and one random number
            (r4) between [0, 2π]
            if r1 > MOA then
                if r2 > 0.5 then
                    Update the ith solution's positions using Eq. (9), the multiplication operator
                else
                    Update the ith solution's positions using Eq. (9), the division operator
                end if
            else
                if r3 > 0.5 then
                    Update the ith solution's positions using Eq. (10), the addition operator
                else
                    Update the ith solution's positions using Eq. (10), the subtraction operator
                end if
            end if
        end for
    end for
    C_Iter = C_Iter + 1
    // Chaotic movements
    for (i = 1 to solution_num) do
```

Generate a random number r_5 between [0, 1]
    if $r_5 > 0.5$ **then**
        if (j = 1 to *solution_dim*) **do**
            Update the ith solution's positions using Eq. (8)
        end for
    end if
end for
end while
return the best solution.

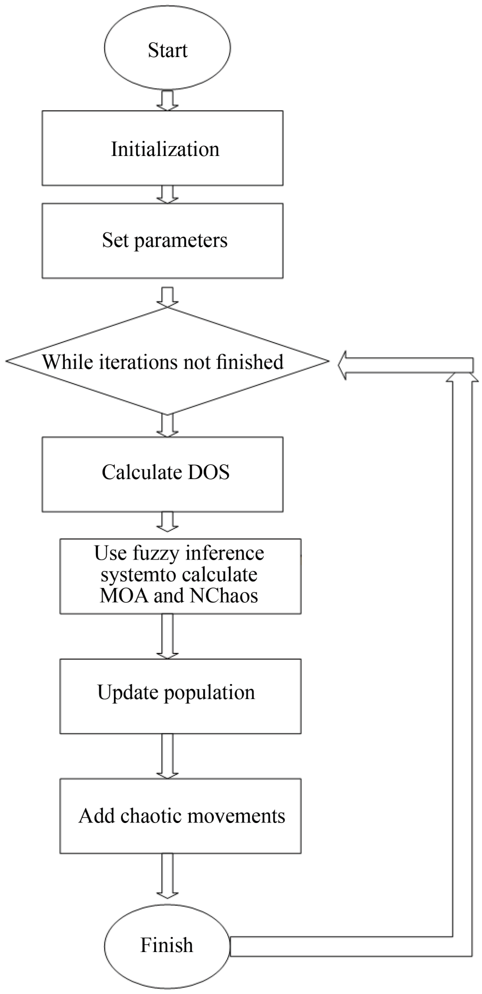


Figure 6. The proposed algorithm

4. Experimental results

In this section, experiments were conducted to assess the effectiveness of the proposed approach. The comparison is done between the AOA [15] and IAOA. Two different sets of experiments were performed. In the first set of experiments, the algorithms were evaluated with 12 diverse test functions from [15]. In the second set of experiments, the algorithms were evaluated on two real-world problems from [15].

4.1 *Experiments on test functions*

In this section the conducted experiments are detailed. In Table 5. The compared algorithms are listed.

In all the experiments the population size was set to 20 and the maximum number of iterations to 500. For evaluating the algorithms, 18 test functions from three different types were selected. The functions are listed in Tables 6–8.

Each test function was tested 30 times in 30, 40 and 50 dimensions and the average convergence speed and accuracy were calculated. For functions the evaluated dimension is mentioned in Table 8.

In Table 9, the average convergence speed is shown based on an average and standard deviation of the number of function evaluations. In all the algorithms the convergence is calculated when the fitness of the best solution in last 50 iterations is changing less than 10^{-6}.

From Table 9, it could be concluded that in the 18 test functions, IAOA, AOA and SMA have the fastest convergence in 4, 4 and 9 functions respectively. Therefore SMA has the best convergence speed and IAOA has the second rank. On the Other hand IAOA has better convergence than AOA in 10 functions out of the 18 test functions.

Table 5. Compared algorithms

Algorithm Name	Reference
Slime Mould Algorithm (SMA)	[57]
Grey Wolf Optimizer (GWO)	[58]
Harris Hawks Optimization (HHO)	[59]
Whale Optimization Algorithm (WOA)	[60]
Arithmetic Optimization Algorithm (AOA)	[15]
Improved Arithmetic Optimization Algorithm (IAOA)	Proposed

Table 6. Unimodal test functions for evaluation

Function	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_j \right)^2$	$[-100, 100]$	0
$F_4(x) = \max_i \{\lvert x_i \rvert .1 \le i \le n\}$	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{n} ([(x_i + 0.5)])^2$	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random(0.1)$	$[-128, 128]$	0

Table 7. Multimodal test functions for evaluation

Function	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$[-5.12, 5.12]$	
$F_9(x) = -20exp\left(-2/0\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0
$F_{10}(x) = \frac{1}{4,000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0
$F_{11}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\mu y_{i+1})\right] + (y_n-1)^2\right\} + \sum_{i=1}^{n}u(x_i = 10.100.4)$ $\mathrm{u}(x_i.a.k.m) = \begin{cases} k(x_i-a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i-a)^m x_i < -a \end{cases}$	$[-50, 50]$	0
$F_{12}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right] + (x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n}u(x_i \cdot 5 \cdot 100 \cdot 4)$	$[-50, 50]$	0

Table 8. Fixed-dimension multimodal test functions for evaluation

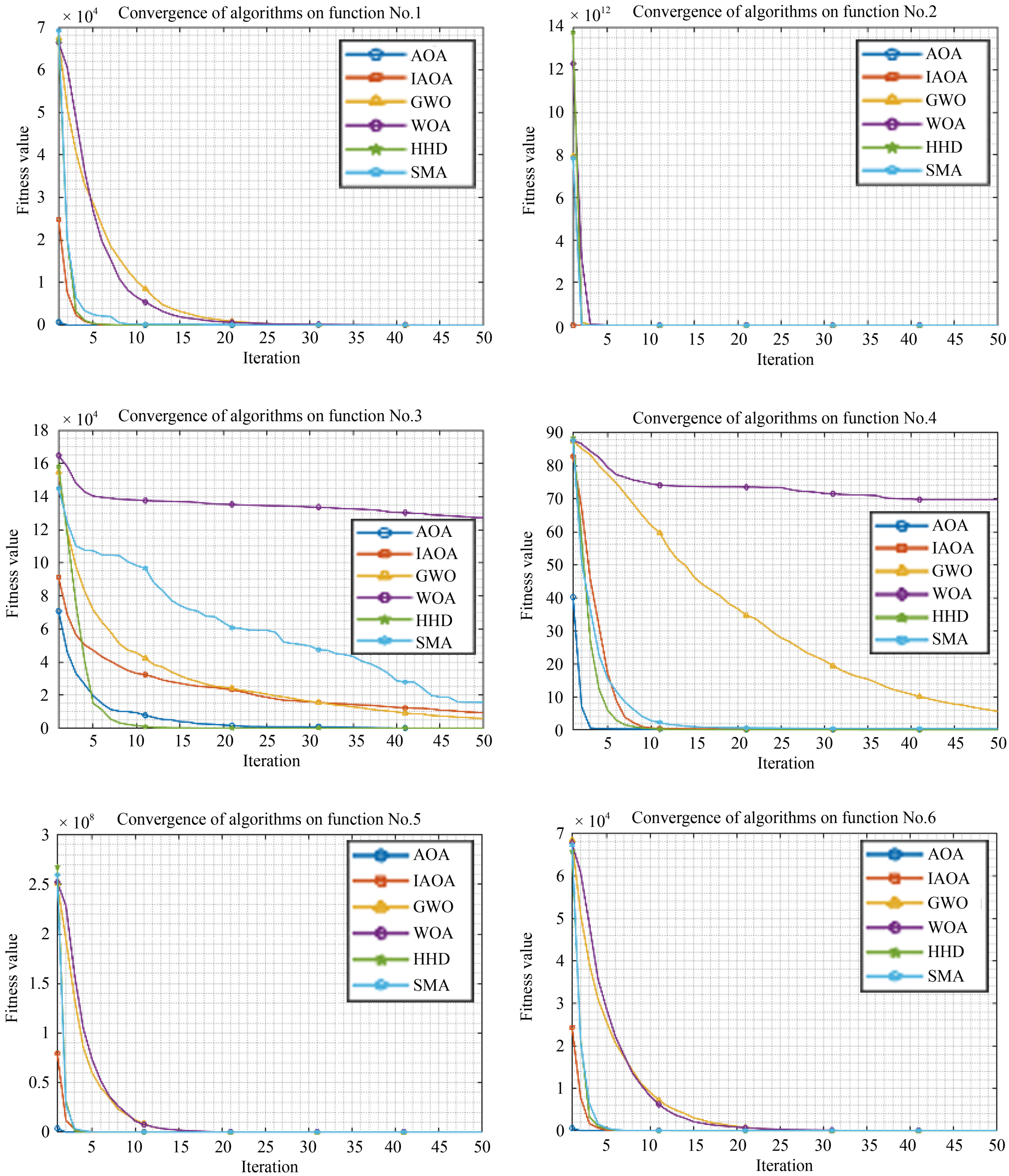
Function	Dim	Range	$f_{\min}$
$F_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}}(x_i - a_{ij})\right)$	2	$[-65, 65]$	1
$F_{14}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]$	4	$[-5, 5]$	0.0003
$F_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 - 4x_2^4$	2	$[-5, 5]$	-1.0316
$F_{16}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 - \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{17}(x) = \left[1 + (x_1+x_2+1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_i + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	$[-2, 2]$	3
$F_{18}(x) = -\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	4	$[0, 1]$	-10.1532

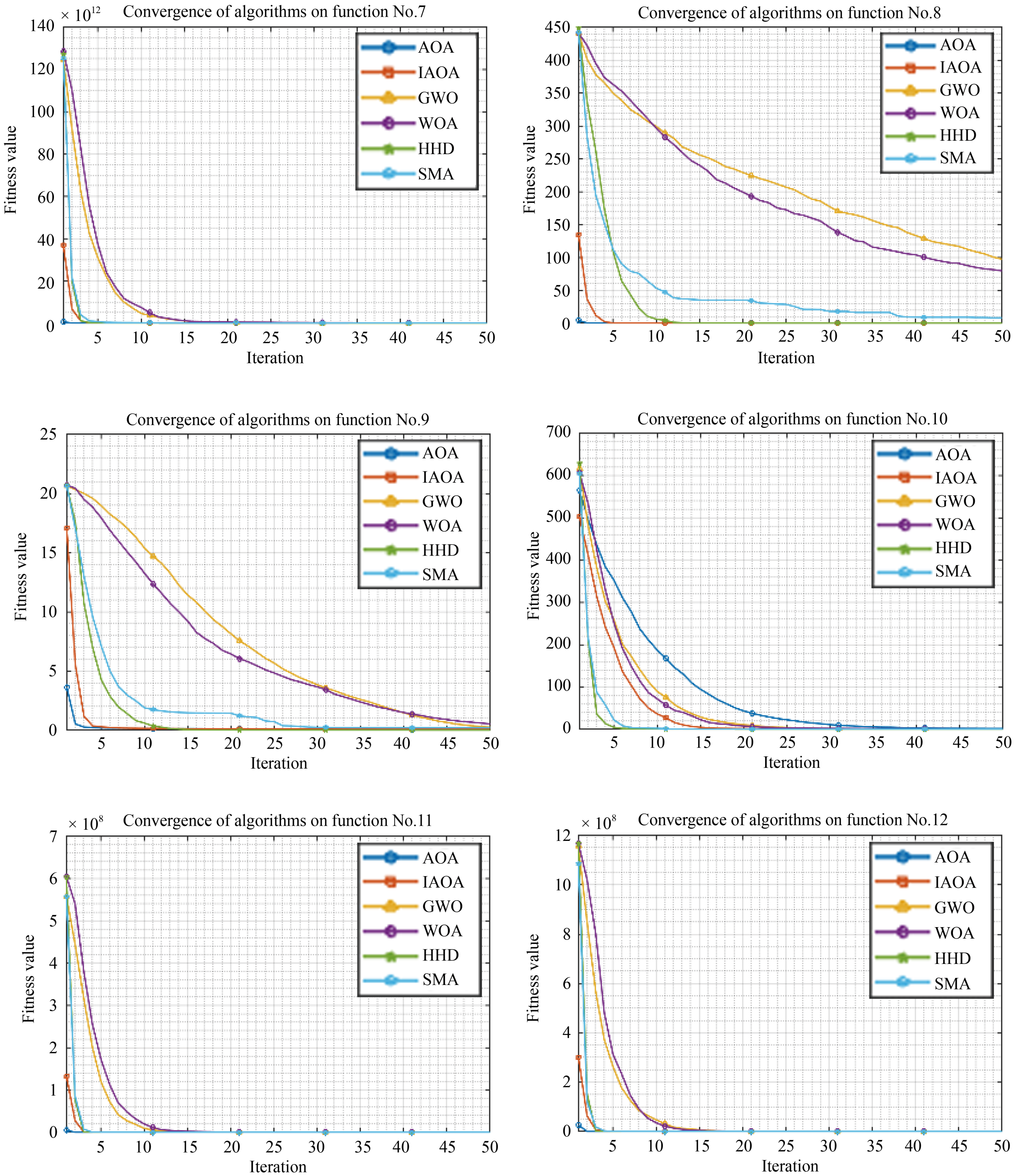
Table 9. Convergence speed of the compared algorithms on each test functions

Test function no.		Algorithms					
		AOA	IAOA	GWO	WOA	HHO	SMA
1	Avg	8,626.7	6,693.5	4,900.0	5,526	3,056.7	2,155.3
	Std	1,792.0	1,512.4	92.453	343.79	644.23	305.16
2	Avg	8,660.9	5,910.2	5,827.0	6,111.3	4,820.7	2,192
	Std	1,649.7	1,849.2	115.28	284.70	1,658.9	335.12
3	Avg	4,232.0	4,387.1	14,717	15,000	4,732.7	2,472.3
	Std	1,646.2	985.27	460.00	0	1,330.8	907.56
4	Avg	4,228.9	4,072.0	13,986	6,380.3	4,572.6	2,281.3
	Std	1,130.0	1,259.6	529.08	3,557.1	1,463.4	351.31
5	Avg	5,532.0	5,810.6	10,803	9,657.6	7,345.6	2,160.6
	Std	1,294.5	1,391.2	3836.9	1,404.9	2,193.5	460.15
6	Avg	1,860.2	2,372.4	14,443	9,659.6	6,130.0	2,058
	Std	379.11	173.59	1,661.3	1,266.1	2,037.4	468.14
7	Avg	1,811.1	1,940.6	6,484.3	6,308	3,099.0	2,398.6
	Std	726.97	711.55	1,751.0	2,314.8	1,112.4	1,274.3
8	Avg	9,036.2	7,204.4	8,292.3	5,618.6	3,157	2,220.6
	Std	849.15	2,113.2	3,769.3	455.49	684.50	500.86
9	Avg	9,497.3	7,579.3	6,247.3	6,503.6	4,498	2,166
	Std	587.60	2,099.6	129.65	288.72	1,381.7	284.12
10	Avg	4,448.7	2,987.3	6,927.3	5,778.6	3,284.6	2,013.3
	Std	772.64	610.10	3,723.9	951.02	694.47	240.74
11	Avg	2,126.9	2,493.3	12,993	9,175.6	4,432	2,162.3
	Std	385.03	140.28	3,196.2	622.68	1,638.1	653.14
12	Avg	5,286.0	5,360.8	9,188.3	9,727.6	5,668	2,428
	Std	1,264.2	1,565.2	3,389.0	1,559.8	2,139.3	719.73
13	Avg	380.22	364.44	1,138.3	1,020.6	1,116.6	1,368
	Std	73.291	25.209	427.29	345.46	446.88	369.48
14	Avg	1,760.7	2,144.8	1,557.3	2,008	1,694.3	2,297
	Std	951.17	888.58	1,096.1	836.27	918.74	1,292.8
15	Avg	1,553.3	533.33	966.66	1,156	1,593.6	1,667
	Std	1,141.3	89.408	346.23	352.22	469.60	454.57
16	Avg	418.22	655.11	927.33	2,325.3	2,450	1,533
	Std	308.21	217.22	322.57	1,338.5	1,214.3	599.73
17	Avg	1,770.9	932.66	1,225.3	1,546.6	1,771.3	2,187
	Std	1,342.5	661.94	481.36	561.49	627.13	541.23
18	Avg	664.66	615.11	1,039.0	3,030.3	2,660	1,401
	Std	192.39	179.22	464.10	287.49	882.71	936.39

In Figure 7, the convergence of the algorithms in the first 50 iterations is shown.

From Figure 7, it could be concluded that IAOA has a better convergence behavior compared to AOA most of the time. Also the final convergence accuracy of IAOA is almost the same with AOA. It could also be observed that SMA has the best convergence but the accuracy of the convergence is lower than IAOA.


Convergence of algorithms on function No.1
× 10^4
Fitness value
Iteration
AOA
IAOA
GWO
WOA
HHD
SMA
Convergence of algorithms on function No.2
× 10^12
Fitness value
Iteration
AOA
IAOA
GWO
WOA
HHD
SMA
Convergence of algorithms on function No.3
× 10^4
Fitness value
Iteration
AOA
IAOA
GWO
WOA
HHD
SMA
Convergence of algorithms on function No.4
Fitness value
Iteration
AOA
IAOA
GWO
WOA
HHD
SMA
Convergence of algorithms on function No.5
× 10^8
Fitness value
Iteration
AOA
IAOA
GWO
WOA
HHD
SMA
Convergence of algorithms on function No.6
× 10^4
Fitness value
Iteration
AOA
IAOA
GWO
WOA
HHD
SMA


Convergence of algorithms on function No.7
× 10^12
Convergence of algorithms on function No.8
Convergence of algorithms on function No.9
Convergence of algorithms on function No.10
Convergence of algorithms on function No.11
× 10^8
Convergence of algorithms on function No.12
× 10^8
AOA
IAOA
GWO
WOA
HHD
SMA
Fitness value
Iteration

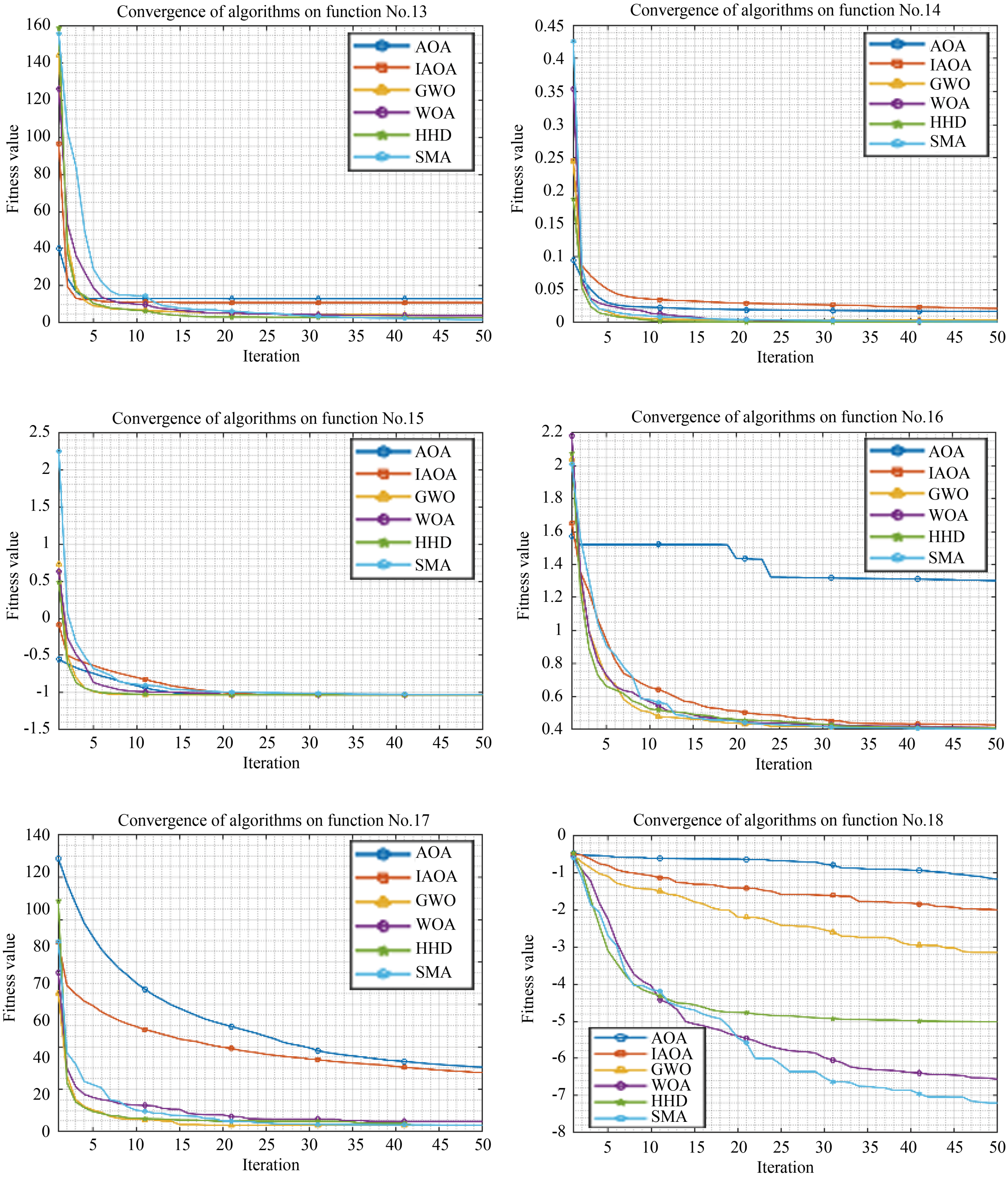


Figure 7. Convergence of the algorithms in Table 4 for the first 50 iterations

The Wilcoxon statistical significance test was performed on the results to show the best algorithms. The result of this test and the box plot could be seen in Figures 8 and 9.

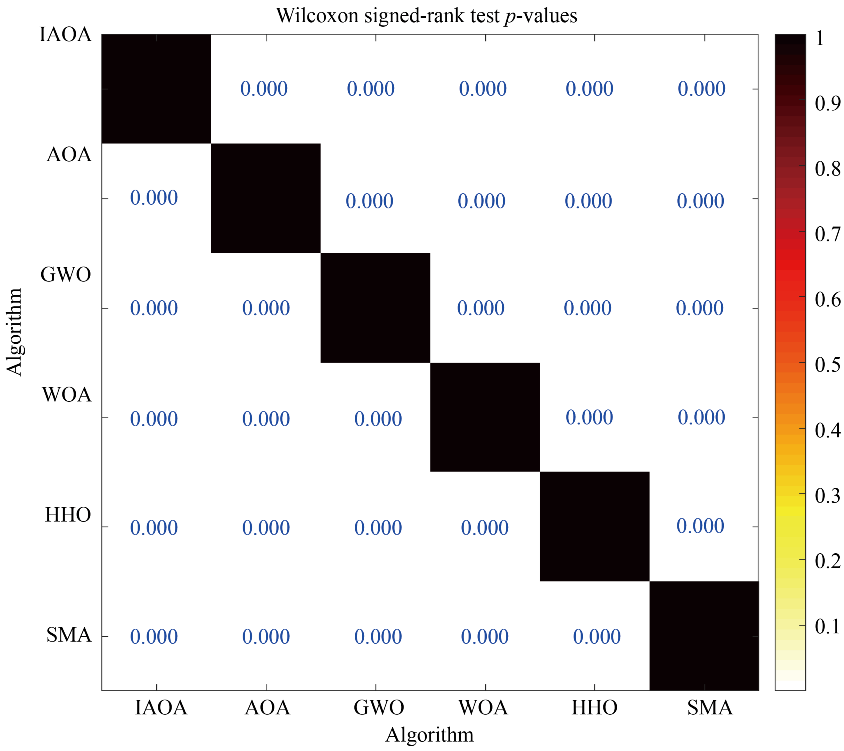


Figure 8. Wilcoxon statistical test on the convergence speed result

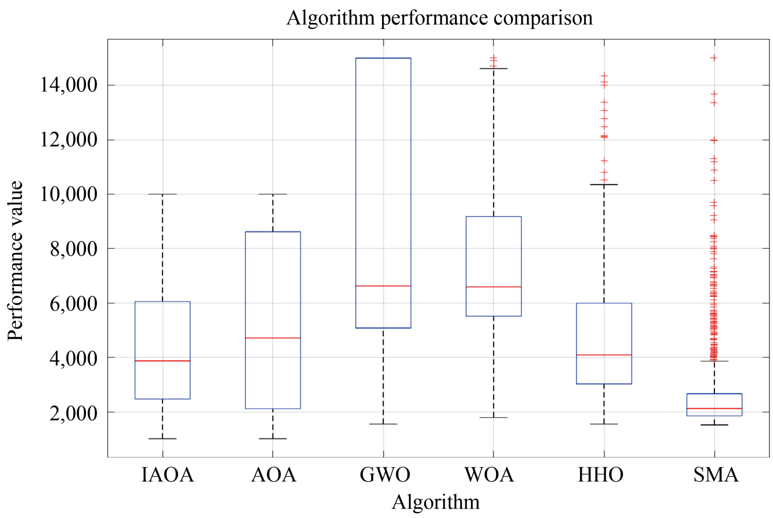


Figure 9. Box plot of convergence speed result

In Table 10, the convergence accuracy is calculated based on an average and standard deviation of the optimal points reached.

From Table 10, it could be concluded that HHO has the best convergence accuracy in 13 test functions. From Table 9, it was concluded that SMA has the best convergence speed, but from Table 10, it could be observed that in 13 test functions IAOA has higher convergence accuracy than SMA. Therefore it could be concluded that IAOA has the best results if we consider convergence speed and accuracy together.

Table 10. Convergence accuracy of the compared algorithms on test functions

Test function no.		Algorithms					
		AOA	IAOA	GWO	WOA	HHO	SMA
1	Avg	0.0025	0.0075	0.0000	0.0000	0.0000	3.0298
	Std	0.0061	0.0042	0.0000	0.0000	0.0000	13.334
2	Avg	0.0039	0.0184	0.0000	0.0000	0.0000	0.3552
	Std	0.0118	0.0177	0.0000	0.0000	0.0001	1.4362
3	Avg	0.1519	807.29	0.2170	11,873	0.0002	6,853.0
	Std	0.1618	3,502.2	0.6220	28,885	0.0015	2,946.3
4	Avg	0.0607	0.0743	0.0000	63.163	0.0000	0.2654
	Std	0.0216	0.0358	0.0001	26.092	0.0000	1.0501
5	Avg	38.69	38.36	37.322	38.156	0.0171	51.275
	Std	0.2479	0.2674	0.6780	0.3846	0.0220	66.351
6	Avg	5.4644	1.8095	1.7362	0.8054	0.0002	14.047
	Std	0.3384	0.3749	0.5374	0.3739	0.0004	19.781
7	Avg	0.0005	0.0022	0.0058	0.0155	0.0015	0.1253
	Std	0.0005	0.0047	0.0055	0.0222	0.0017	0.1034
8	Avg	0.0001	0.0009	15.058	0.0000	0.0000	7.8268
	Std	0.0004	0.0014	40.667	0.0000	0.0000	17.819
9	Avg	0.0006	0.0070	0.0000	0.0000	0.0000	0.2915
	Std	0.0019	0.0091	0.0000	0.0000	0.0000	0.8658
10	Avg	3.3393	0.3043	0.0060	0.0091	0.0000	0.6959
	Std	4.7130	0.0876	0.0121	0.0435	0.0000	2.4929
11	Avg	0.6874	0.1802	0.0848	0.0251	0.0000	0.9708
	Std	0.0406	0.0381	0.0378	0.0123	0.0000	1.0680
12	Avg	3.8629	3.4026	1.6073	0.8744	0.0001	2.3453
	Std	0.0960	0.1140	0.3280	0.3651	0.0001	3.0833
13	Avg	4.2101	3.6124	1.0364	1.1313	0.6839	0.3327
	Std	2.0050	1.1136	1.1248	0.9549	0.6101	0.0000
14	Avg	0.0043	0.0015	0.0011	0.0002	0.0001	0.0002
	Std	0.0088	0.0041	0.0022	0.0001	0.0001	0.0001
15	Avg	−0.3439	−0.3439	−0.3439	−0.3439	−0.3439	−0.3438
	Std	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001
16	Avg	0.4177	0.1377	0.1358	0.1326	0.1326	0.1327
	Std	0.2286	0.0065	0.0097	0.0000	0.0000	0.0001
17	Avg	8.6682	5.2000	1.0001	1.0002	1.0000	
	Std	10.151	7.3738	0.0000	0.0011	0.0000	0.0000
18	Avg	−1.1992	−1.2849	−1.2857	−1.2863	−1.2866	
	Std	0.0516	0.0017	0.0017	0.0024	0.0014	0.0009

4.2 *Experiments on real-world problems*

In this section, 2 real-world problems were selected for the evaluation [61]. The two problems are the welded beam design problem and the speed reducer problem. The problems and the results are detailed next.

4.2.1 *Welded beam design*

The problem is the design of a beam with a uniform rectangular cross-section, which is welded to a base to withstand a force of 6,000 pounds.

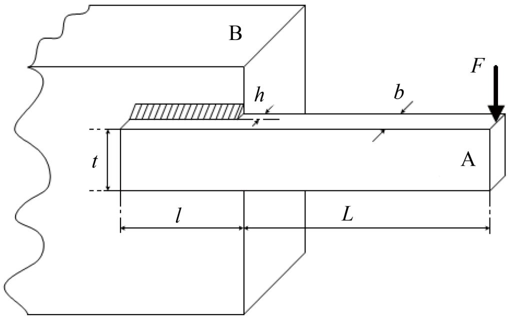


Figure 10. Welded beam design problem [61]

The length of L is equal to 14 inches. The design aims to minimize the construction cost so that an acceptable combination of weld thickness h, weld length l, beam thickness t, and beam width b is found (Figure 10). The objective function is:

$$\text{Min } f = (1 + c_1) h^2 l + c_2 t b (L + I). \tag{10}$$

In equation (11), f is the cost that includes the welding cost and material cost. Parameter c_1 is the cost of welding materials per volume unit (equivalent to 0.10471 dollars per square inch) and c_2 is the cost of the raw material consumed per unit volume (equivalent to 0.04811 dollars per square inch). So f would be:

$$J_{p1}(x) = 1.10471 x_2 x_1^2 - 0.014811 x_3 x_4 (14 + x_2). \tag{11}$$

There are also some constraints, which are:

$$g_1(x) = \tau(x) - 13,600 \leq 0$$

$$g_2(x) = \sigma(x) - 30,000 \leq 0$$

$$g_3(x) = x_1 - x_4 \leq 0$$

$$g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \leq 0$$

$$g_5(x) = 0.125 - x_1 \leq 0$$

$$g_6(x) = \delta(x) - 0.25 \leq 0$$

$$g_7(x) = 6,000 - p_c(x) \leq 0.$$

0, $1 \leq x_1 \leq 20.1 \leq x_2 \leq 10.$ $0.1 \leq x_3 \leq 10.$ $0.1 \leq x_4 \leq 2.$

4.2.2 *Speed reducer*

The speed reducer problem is a design problem, which is a part of the gearbox of a mechanical system. This problem involves seven variables (Figure 11).

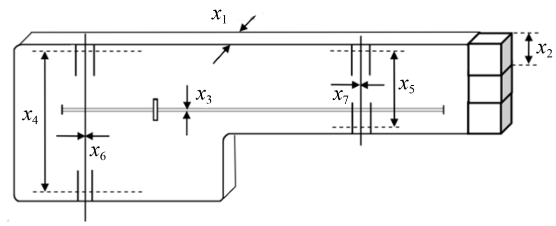


Figure 11. Speed reducer problem [61]

The objective function and constraints are shown below.

$$J_{p2}(x) = 0.7854x_1x_2^2 - \left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right)$$
$$+ 7.477\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \tag{12}$$

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0$$

$$g_3(x) = \frac{1.93}{x_2x_3x_4^3x_6^4} - 1 \leq 0$$

$$g_4(x) = \frac{1.93}{x_2x_3x_5^3x_7^4} - 1 \leq 0$$

$$g_5(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69 * 10^6}}{110x_6^3} - 1 \leq 0$$

$$g_6(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5 * 10^6}}{85x_7^3} - 1 \leq 0$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0$$

$$g_8(x) = \frac{5x_2}{x_1 - 1} - 1 \leq 0$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0$$

$$6 \leq x_1 \leq 3 \_ 0.7 \leq x_2 \leq 0.8 \_ 7 \leq x_3 \leq 28 \_ 7.3 \leq x_4 \leq 8.37$$

$$3 \leq x_5 \leq 8.32 \_ 3 \leq x_6 \leq 9 \_ 5.5 \leq x_7 \leq 9.5.$$

4.2.3 *Results*

To evaluate the performance of the proposed IAOA, two real-world optimization problems-denoted as Jp1 and Jp2-were used. The performance of IAOA was compared against the original AOA, as well as four well-established metaheuristic algorithms: PSO, Artificial Bee Colony (ABC), Firefly Optimization (FFO), and Social Spider Optimization (SSO) [48]. Each algorithm was executed independently over 30 runs, and the average and Standard deviation (Std) of the objective function values were recorded as shown in Table 11.

Table 11. Results on real world problems

Algorithm	Jp1 Avg	Jp1 Std	Jp2 Avg	Jp2 Std
IAOA	1.7250	0.0002	2,994.7831	1.0037
AOA	1.7525	0.1890	2,995.5402	1.0456
SSO	1.7464	0.0257	2,996.1132	1.3351
FFO	2.1974	0.1952	3,000.0054	8.3565
ABC	2.1673	0.2542	2,998.0628	6.3545
PSO	2.0111	0.1085	3,079.2623	26.2173

The results clearly demonstrate that IAOA achieves the best performance among the tested algorithms in both problem instances. In Jp1, IAOA yielded the lowest average objective value (1.7250) with an extremely low standard deviation (0.0002), indicating both high accuracy and exceptional stability. Similarly, in Jp2, IAOA again outperformed the other methods, attaining the lowest average value (2,994.7831) and demonstrating robustness with a standard deviation of only 1.0037. These findings confirm that the proposed enhancements-including the DOS metric, fuzzy adaptation mechanism, and enriched movement strategies-significantly improve the optimization capability of the base AOA.

4.3 *Ablation study*

In this section, an ablation study is performed to see the effect of the three main contributions of IAOA on the results. The three main contributions where the fuzzy inference system, the chaotic movements and the triangular movements. These contributions are defined as three states in the ablation study and the effect of these parameters on the average convergence speed is reported in Table 12.

Table 12. Ablation study result on average convergence speed

Test function No.	State1 (NO FIS)	State2 (NO CHAOS)	State3 (NO TRIANGULAR)
1	6,391.3	6,400	7,080.6
2	9,756.6	6,261.3	5,577.3
3	3,566	4,560	4,501.3
4	4,012	4,236	4,433.3
5	5,460	6,620.6	6,189.3
6	2,051.3	2,220	2,108
7	1,894.6	1,880.6	1,846
8	9,376	8,106	8,624
9	2,048	8,540.6	8,040
10	4,000.6	3,055.3	3,252.6
11	2,884	2,498	2,394.6
12	5,195.3	5,320	5,202
13	1,106.6	1,082.6	1,115.3
14	6,448	7,016.6	5,689.3
15	1,753.3	1,574.6	3,125.3
16	2,016.6	2,272.6	1,652
17	2,646.6	2,525.3	3,174.6
18	2,092.6	1,796	1,735.3

From Table 12, it could be concluded that the fuzzy inference system has the lowest effect and the chaotic movements has the highest effect on the average convergence speed.

4.4 *Parameter sensitivity*

In this section, the sensitivity of IAOA to the function dimension and the λ parameter of chaotic logistic map are analyzed. In Table 13, the parameter sensitivity of the convergence speed to function dimension is calculated when $\lambda = 4$.

Table 13. Parameter sensitivity result for function dimension

Test function No.	Dim = 50	Dim = 100	Dim = 500	Dim = 1,000
1	5,938.6	5,277.3	3,396.6	2,588.6
2	7,111.3	6,910.6	7,611.3	3,275.3
3	4,596.6	3,343.3	2,228	2,060
4	4,033.3	3,558	2,209.3	1,020
5	5,999.3	5,635.3	6,472	5,664.6
6	2,337.3	2,243.3	2,346	2,399.3
7	2,096.6	1,881.3	1,987.3	2,006
8	8,039.3	9,088	8,954.6	8,747.3
9	6,972	8,366.6	6,362.6	5,532
10	2,860.6	2,708.6	2,490.6	2,378
11	2,498.6	2,272.6	2,080.6	2,162.6
12	5,145.3	5,859.3	5,409.3	5,259.3

From Table 13 it could be concluded that IAOA has better convergence speed in higher dimensions. In this experiment only the first 12 test functions were evaluated because function 13–18 are not defined for high dimensions (see Table 8).

In Table 14, the parameter sensitivity of the convergence speed to λ (logistic map parameter) is calculated when function dimension is set to 50 for functions 1–12 and to the specific dimension mentioned in Table 8 for functions 13–18.

Table 14. Parameter sensitivity result for λ parameter in logistic map

Test function No.	$\lambda = 2$	$\lambda = 4$	$\lambda = 6$
1	5,781.3	5,938.6	6,136.6
2	6,084	7,111.3	6,530.6
3	4,670.6	4,596.6	5,180.6
4	3,840.6	4,033.3	3,954
5	6,317.3	5,999.3	5,948.6
6	2,264	2,337.3	2,289.3
7	2,064.6	2,096.6	2,138
8	8,466.6	8,039.3	7,654
9	7,960	6,972	8,791.3
10	2,753.3	2,860.6	2,772.6
11	2,453.3	2,498.6	2,477.3
12	5,166.6	5,145.3	4,956.6
13	1,110.6	1,126.6	1,150
14	7,696.6	7,546	7,220.6
15	1,613.3	1,556	1,509.3
16	1,944	2,168.6	1,828
17	1,969.3	2,565.3	2,597.3
18	1,904	1,877.3	1,964

From Table 14 it could be concluded that IAOA has better convergence speed when $\lambda = 2$ for the chaotic logistic map.

5. Conclusion

Metaheuristic algorithms, particularly those inspired by natural and mathematical principles, have become essential tools for solving complex optimization problems. The AOA is one such method that has gained widespread use due to its mathematical formulation and competitive performance. In this study, we proposed an IAOA that incorporates three key innovations: (1) the DOS metric, which estimates the distance of solutions from the best solution (2) a fuzzy logic-based adaptation mechanism for the MOA parameter, enhancing the algorithm's ability to balance exploration and exploitation; and (3) the use of chaotic and triangular perturbation strategies in the evolutionary process to improve population diversity and convergence behavior.

Comparative experiments on standard benchmark functions and real-world applications indicate that IAOA has good capability for improving convergence speed. From the 18 benchmark functions IAOA had better results in 10 functions compared to AOA and the convergence were almost the same. In the experiments SMA has better convergence speed compared to IAOA but the convergence accuracy of SMA is lower than IAOA most of the time.

An ablation study was performed to see the effect of each of the three components, 1) Fuzzy inference system 2) Chaotic movements and 3) Triangular movements on the convergence speed. The study showed that the chaotic movements and fuzzy inference system had the highest and lowest effect on the convergence speed of IAOA respectively.

Finally a parameter sensitivity experiment was performed. Two of the main parameters, namely function dimension and λ were chosen for this experiment. The experiment clarified that IAOA has better convergence speed in higher dimensions and $\lambda = 2$ was the best setting.

Future work will explore the integration of these mechanisms into other optimization algorithms and the application of IAOA to large-scale and multi-objective optimization problems.

Acknowledgement

The authors gratefully acknowledge the support of the Estonian University of Life Sciences, Institute of Forestry and Engineering, Chair of Energy Application Engineering. This research was made possible through the Energy Efficiency and Renewable Energy Research Infrastructure project, funded by the Estonian Research Council under Grant TARISTU24-TK12.

Conflict of interest

The authors declare no conflicts of interest.

References

[1] Boyd S, Vandenberghe L. *Convex Optimization*. London: Cambridge University Press; 2004. p.727.

[2] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*. California: IEEE; 1995. p.1942–1948.

[3] Alorf A. A survey of recently developed metaheuristics and their comparative analysis. *Engineering Applications of Artificial Intelligence*. 2023; 117: 105622.

[4] Rajwar K, Deep K, Das S. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review*. 2023; 56(11): 13187–13257.

[5] Sulaiman MH, Mustaffa Z, Saari MM, Daniyal H. Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*. 2020; 87: 103330.

[6] Ghasemi-Marzbali A. A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm. *Soft Computing*. 2020; 24(17): 13003–13035.

[7] Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R. Red deer algorithm (RDA): a new nature-inspired meta-heuristic. *Soft Computing*. 2020; 24(19): 14637–14665.

[8] Kaveh A, Dadras Eslamlou A. Water strider algorithm: a new metaheuristic and applications. *Structures*. 2020; 25: 520–541.

[9] Agushaka JO, Ezugwu AE, Abualigah L. Dwarf mongoose optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*. 2022; 391: 114570.

[10] Rani R, Garg V, Jain S, Garg H. A hybrid mountain gazelle particle swarm-based algorithm for constrained optimization problems. *Evolving Systems*. 2025; 16(1): 35.

[11] Kaveh A, Khanzadi M, Rastegar Moghaddam M. Billiards-inspired optimization algorithm; a new meta-heuristic method. *Structures*. 2020; 27: 1722–1739.

[12] Dehghani MJ, Montazeri Z, Givi H, Guerrero JM, Dhiman G. Darts game optimizer: a new optimization technique based on darts game. *International Journal of Intelligent Engineering and Systems*. 2020; 13(5): 286–294.

[13] Srivastava A, Das DK. A new Kho-Kho optimization algorithm: an application to solve combined emission economic dispatch and combined heat and power economic dispatch problem. *Engineering Applications of Artificial Intelligence*. 2020; 94: 103763.

[14] Chen Y, Wang G, Yin B, Ma C, Wu Z, Gao M. Basketball team optimization algorithm (BTOA): a novel sport-inspired meta-heuristic optimizer for engineering applications. *Scientific Reports*. 2025; 15(1): 21629.

[15] Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*. 2021; 376: 113609.

[16] Le-Duc T, Nguyen Q-H, Nguyen-Xuan H. Balancing composite motion optimization. *Information Sciences*. 2020; 520: 250–270.

[17] Ghasemi M, Davoudkhani IF, Akbari E, Rahimnejad A, Ghavidel S, Li L. A novel and effective optimization algorithm for global optimization and its engineering applications: turbulent flow of water-based optimization (TFWO). *Engineering Applications of Artificial Intelligence*. 2020; 92: 103666.
[18] Gholizadeh S, Danesh M, Gheyratmand C. A new Newton metaheuristic algorithm for discrete performance-based design optimization of steel moment frames. *Computers & Structures*. 2020; 234: 106250.
[19] Mu L, Wang P, Xin G. Quantum-inspired algorithm with fitness landscape approximation in reduced dimensional spaces for numerical function optimization. *Information Sciences*. 2020; 527: 253–278.
[20] Cheng M-Y, Sholeh MN. Artificial satellite search: a new metaheuristic algorithm for optimizing truss structure design and project scheduling. *Applied Mathematical Modelling*. 2025; 143: 116008.
[21] Bogar E, Beyhan S. Adolescent identity search algorithm (AISA): a novel metaheuristic approach for solving optimization problems. *Applied Soft Computing*. 2020; 95: 106503.
[22] Chou J-S, Nguyen N-M. FBI inspired meta-optimization. *Applied Soft Computing*. 2020; 93: 106339.
[23] Jahangiri M, Hadianfard MA, Najafgholipour MA, Jahangiri M, Gerami MR. Interactive autodidactic school: a new metaheuristic optimization algorithm for solving mathematical and structural design optimization problems. *Computers & Structures*. 2020; 235: 106268.
[24] Askari Q, Younas I, Saeed M. Political optimizer: a novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*. 2020; 195: 105709.
[25] Ghasemian H, Ghasemian F, Vahdat-Nejad H. Human urbanization algorithm: a novel metaheuristic approach. *Mathematics and Computers in Simulation*. 2020; 178: 1–15.
[26] Mozhdehi AT, Khodadadi N, Aboutalebi M, El-kenawy EM, Hussien AG, Zhao W, et al. Divine religions algorithm: a novel social-inspired metaheuristic algorithm for engineering and continuous optimization problems. *Cluster Computing*. 2025; 28(4): 253.
[27] Cong Y, Yang B, Wei J. A novel meta-heuristic algorithm based on candidate cooperation and competition. *Scientific Reports*. 2025; 15(1): 24971.
[28] Dehghani M, Montazeri Z, Dehghani A, Malik OP. GO: group optimization. *Gazi University Journal of Science*. 2020; 33(2): 381–392.
[29] Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M. Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*. 2020; 94: 103731.
[30] Zaeimi M, Ghoddosian A. Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization. *Soft Computing*. 2020; 24(16): 12027–12066.
[31] Moazzeni AR, Khamehchi E. Rain optimization algorithm (ROA): a new metaheuristic method for drilling optimization solutions. *Journal of Petroleum Science and Engineering*. 2020; 195: 107512.
[32] Lara-Montaño OD, Gómez-Castro FI, Gutiérrez-Antonio C, Dragoi EN. Success-based optimization algorithm (SBOA): development and enhancement of a metaheuristic optimizer. *Computers & Chemical Engineering*. 2025; 194: 108987.
[33] Valdez F, Melin P, Castillo O. A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation. *Expert Systems with Applications*. 2014; 41(14): 6459–6466.
[34] Peraza C, Valdez F, Garcia M, Melin P, Castillo O. A new fuzzy harmony search algorithm using fuzzy logic for dynamic parameter adaptation. *Algorithms*. 2016; 9(4): 69.
[35] Perez J, Valdez F, Castillo O, Roeva O. Bat algorithm with parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions. In: *2016 IEEE 8th International Conference on Intelligent Systems (IS)*. Sofia, Bulgaria: IEEE; 2016. p.120–127.
[36] Olivas F, Valdez F, Castillo O, Gonzalez CI, Martinez G, Melin P. Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Applied Soft Computing*. 2017; 53: 74-87.
[37] Rodríguez L, Castillo O, Soria J, Melin P, Valdez F, Gonzalez CI, et al. A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Computing*. 2017; 57: 315–328.
[38] Castillo O, Amador-Angulo L. A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. *Information Sciences*. 2018; 460–461: 476–496.
[39] Nobile MS, Cazzaniga P, Besozzi D, Colombo R, Mauri G, Pasi G. Fuzzy self-tuning PSO: a settings-free algorithm for global optimization. *Swarm and Evolutionary Computation*. 2018; 39: 70–85.

[40] Pelusi D, Mascella R, Tallini L, Nayak J, Naik B, Abraham A. Neural network and fuzzy system for the tuning of gravitational search algorithm parameters. *Expert Systems with Applications*. 2018; 102: 234–244.

[41] Olivas F, Valdez F, Melin P, Sombra A, Castillo O. Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. *Information Sciences*. 2019; 476: 159–175.

[42] Keivanian F, Chiong R. A novel hybrid fuzzy-metaheuristic approach for multimodal single and multi-objective optimization problems. *Expert Systems with Applications*. 2022; 195: 116199.

[43] Melin P, Miramontes I, Carvajal O, Prado-Arechiga G. Fuzzy dynamic parameter adaptation in the bird swarm algorithm for neural network optimization. *Soft Computing*. 2022; 26(18): 9497–9514.

[44] Ramachandran M, Mirjalili S, Malli Ramalingam M, Gnanakkan CAR, Parvathysankar DS, Sundaram A. A ranking-based fuzzy adaptive hybrid crow search algorithm for combined heat and power economic dispatch. *Expert Systems with Applications*. 2022; 197: 116625.

[45] Zhou X, Ma H, Gu J, Chen H, Deng W. Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism. *Engineering Applications of Artificial Intelligence*. 2022; 114: 105139.

[46] Dhal KG, Sasmal B, Das A, Ray S, Rai R. A comprehensive survey on arithmetic optimization algorithm. *Archives of Computational Methods in Engineering*. 2023; 30(5): 3379–3404.

[47] Tunç A, Taşdemir S. Fuzzy based weighted, arithmetic optimization algorithm (AOA) for cash management optimization on automatic teller machines (ATM). In: *Intelligent and Fuzzy Systems*. Germany: Springer International Publishing; 2022. p.287–294.

[48] Maleknasab Ardakani M, Tabarzad MA, Shayegan MA. Detecting sybil attacks in vehicular ad hoc networks using fuzzy logic and arithmetic optimization algorithm. *The Journal of Supercomputing*. 2022; 78(14): 16303–16335.

[49] Azizi M, Talatahari S. Improved arithmetic optimization algorithm for design optimization of fuzzy controllers in steel building structures with nonlinear behavior considering near fault ground motion effects. *Artificial Intelligence Review*. 2022; 55(5): 4041–4075.

[50] Talpur N, Abdulkadir SJ, Alhussian H, Hasan MH, Abdullah MHA. Optimizing deep neuro-fuzzy classifier with a novel evolutionary arithmetic optimization algorithm. *Journal of Computational Science*. 2022; 64: 101867.

[51] Elkasem AHA, Khamies M, Magdy G, Taha IBM, Kamel S. Frequency stability of AC/DC interconnected power systems with wind energy using arithmetic optimization algorithm-based fuzzy-PID controller. *Sustainability*. 2021; 13(21): 12095.

[52] Limane A, Zitouni F, Harous S, Lakbich R, Ferhat A, Almazyad AS, et al. Chaos-enhanced metaheuristics: classification, comparison, and convergence analysis. *Complex & Intelligent Systems*. 2025; 11(3): 177.

[53] Wang W, Tian J. An improved nonlinear tuna swarm optimization algorithm based on circle chaos map and Levy flight operator. *Electronics*. 2022; 11(22): 3678.

[54] Li Y, Li L, Lian Z, Zhou K, Dai Y. A quasi-opposition learning and chaos local search based on walrus optimization for global optimization problems. *Scientific Reports*. 2025; 15(1): 2881.

[55] Yu H, Zhao N, Wang P, Chen H, Li C. Chaos-enhanced synchronized bat optimizer. *Applied Mathematical Modelling*. 2020; 77: 1201–1215.

[56] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*. 2016; 96: 120–133.

[57] Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. *Future Generation Computer Systems*. 2020; 111: 300–323.

[58] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*. 2014; 69: 46–61.

[59] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. *Future Generation Computer Systems*. 2019; 97: 849–872.

[60] Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*. 2016; 95: 51–67.

[61] Cuevas E, Cienfuegos M. A new algorithm inspired in the behavior of the social-spider for constrained optimization. *Expert Systems with Applications*. 2014; 41(2): 412–425.