Soft Computing (2022) 26:4863–4881
https://doi.org/10.1007/s00500-022-06873-8

OPTIMIZATION

Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks

Shubham Mahajan[1] · Laith Abualigah[2,3] · Amit Kant Pandit[1] · Maryam Altalhi[4]

Accepted: 23 January 2022 / Published online: 20 February 2022

Abstract

Many population-dependent solutions have recently been suggested. Despite their widespread adoption in many applications, we are still researching using suggested methods to solve real-world problems. As a result, researchers must significantly adjust and refine their procedures based on the main evolutionary processes to ensure faster convergence, consistent equilibrium with high-quality results, and optimization. Thus, a new hybrid method using Aquila optimizer (AO) and arithmetic optimization algorithm (AOA) is proposed in this paper. AO and AOA are both modern meta-heuristic optimization methods. They can be applied to different problems, including image processing, machine learning, wireless networks, power systems, engineering design etc. The proposed approach is examined concerning AO and AOA. To determine results, each procedure is evaluated using the same parameters, such as population size and several iterations. By changing the dimensions, the proposed approach (AO–AOA) is evaluated. The impact of varying dimensions is a standard test that has been used in previous studies to optimize test functions that demonstrate the influence of varying dimensions on the efficiency of AO–AOA. It is clear from this that it fits well with both high- and low-dimensional problems. Population-based methods achieve efficient search results in high-dimensional problems.

Keywords Arithmetic optimization algorithm · Aquila optimizer · Engineering design problems · Metaheuristic · Optimization methods

1 Introduction

In today's world, technologies have less feature space and rely on priorities and financial constraints in information, knowledge-dependent, and expert systems. Researchers must find a feasible solution and adequate details using various algorithms for different problems such as image segmentation (Mahajan et al. 2021), optimization (Liu et al. 2019), target tracking systems (Yan et al. 2020), QoS-aware and social suggestion (Li et al. 2014, 2017; Li and Lin 2020), scheduling problems (Pang et al. 2018; Alawad and Abed-alguni 2021), gold prize prediction (Wen et al. 2017), etc.

Optimization refers to the process of deciding suitable values for variables in a given problem to reduce and optimize the objective function. There are optimization challenges in a variety of fields of study. To solve an optimization problem, several steps must be taken. The parameters of the problem must then be specified. Depending on the form of parameters, issues are classified

✉ Shubham Mahajan
mahajanshubham2232579@gmail.com

Laith Abualigah
laythdyabat@aau.edu.jo

Amit Kant Pandit
amitkantpandit@gmail.com

Maryam Altalhi
marem.m@tu.edu.sa

1 School of Electronics and Communication, Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir 182320, India

2 Faculty of Computer Science and Informatics, Amman Arab University, Amman 11953, Jordan

3 School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia

4 Department of Management Information System, College of Business Administration, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

as either continuous or discrete. Second, it is important to understand the constraints imposed on the parameters (Saremi et al. 2017).

Constraints divide optimization problems into two types restricted and unregulated. The issue's goals should be discussed and answered (Coello 2002; Marler and Arora 2004). To find the best solution, mathematical optimization relies heavily on gradient-based function awareness. However, these techniques are already being used by a variety of researchers and have few disadvantages. Local optima entrapment has an impact on mathematical optimization methods. It refers to an approach that assumes that a local solution is a global solution and therefore fails to achieve the best result.

They are often ineffective in problems involving unknown or computationally expensive derivatives (Mirjalili et al. 2014). Stochastic optimization (Spall 2005) is another form of an optimization algorithm that eliminates these major drawbacks.

To prevent local optimization, stochastic approaches depend on arbitrary operators. It starts the optimization process by generating one or more random solutions to problems. Compared to mathematical optimization methods, it is unnecessary to calculate the gradient of the solution to evaluate the solutions using an objective function (s). Decisions on how to improve results are made based on the practical principles that have been measured. As a result, the problem is considered a black box, and it's a very useful method for dealing with real problems involving unknown search spaces. Because of the benefits mentioned above, stochastic methods are widely used (Michalewicz 2013). Among stochastic optimization methods, nature-inspired population-dependent methods are the most popular (Yang 2010).

These methods approximate standard problem-solving techniques, which are often used by species. The primary goal of all animals is survival. To achieve this purpose, they have grown and adapted in a variety of ways. It is also wise to obtain advice from nature, the world's greatest and oldest optimizing compiler. These algorithms are divided into two categories: single-solution based and multi-solution dependent. The first stage involves developing and improving a single random solution for a specific problem, while the second involves developing and improving multiple solutions for a specific issue.

Methods with multiple solutions are more popular than methods with a single solution (Mirjalili and Lewis 2013). Since multiple solutions are improved during the optimization phase, multi-solution models have a higher inherently higher local optimum-avoidance. Certain methods to leaping from the locally optimal help the solution trapped in a locally optimal.

Current techniques search a larger portion of the search space than single-solution approaches, increasing the global optimum probability. Simulated annealing and hill-climbing are two popular methods that depend on a single solution (Kirkpatrick et al. 1983; Davis 1991). They are both ideal, but the stochastics' cooling factor is strong since they avoid the local optimum of SA. Iterated local search and tabu search are two recent methods that depend on a single solution (Lourenço et al. 2003; Fogel et al. 1966; Glover 1989).

Particle swarm optimization (Eberhat and Kennedy 1995), genetic algorithms (GA) (Holland 1992), differential evolution (DE) (Storn and Price 1997), and ant optimization (ACO) (Colorni et al. 1991) are some common multi-solution-based methods. Darwin's evolutionary theory inspired the GA approach. In this approach, solutions are viewed as entities, and solution parameters replace genes. This approach's main motivation is survival of the fittest species, where the strongest tend to be more interested in improving bad solutions.

Previous research has identified various swarm intelligence optimization approaches, including the firefly algorithm (FA) (Yang 2010; Yang et al. 2010), dolphin echolocation (DEL) (Kaveh and Farhoudi 2013, 2016), grey wolf optimizer (GWO) (Marler and Arora 2004; Abed-alguni and Alawad 2021), and bat algorithm (BA) (Yang 2010). BA and DEL use echolocation like dolphins to locate food and bats to navigate. However, FA mimics the mating behavior of fireflies. The cuckoo's reproductive activity is used in the Cuckoo Search (CS) (Yang 2010; Yang and Deb 2010; Abed-alguni et al. 2021) method. Grey wolf hunting behavior is used in the GWO swarm process technique. Other techniques, such as state of matter search (SMS) (Cuevas et al. 2014, 2013), use various types of matter to solve problems.

The flower pollination algorithm (FPA) (Yang 2012), on the other hand, uses flower pollination and survival behavior for pollination. The question now is why there is a need for new approaches when there are so many that already exist. The solution to this question is in the No Free Lunch (NFL) theorem (Wolpert and Macready 1997), which has scientifically proven that no optimization technique can solve all optimization problems. In other words, when all optimization problems are considered, techniques in this field perform similarly on average. This theorem has influenced the rapidly evolving algorithms proposed in recent decades. It has been one of the motives for writing this article.

Aquila Optimizer (AO) (Abualigah et al. 2021) is a nature-inspired algorithm. The Aquila is among the most common birds of prey in the Northern Hemisphere. Aquila is perhaps the most commonly distributed species of

Aquila. Aquila, like all birds, is a member of the Accipitridae family.

AOA (Abualigah et al. 2020) is a new meta-heuristic method that uses common mathematical operations such as Division (D), Addition (A), Multiplication (M), Subtraction (S), which are applied and modeled to execute optimization in a wide variety of search fields (Boussaïd et al. 2013).

We proposed a hybrid approach in this paper that combines two meta-heuristic approaches, AO and AOA. Their gradient-free and simple structure, black-box nature-inspired methods with higher local optimum avoidance and letting issues are widely used in engineering and other problems (Mahajan et al. 2021; Gogna and Tayal 2013; Zhou et al. 2011; Steenhof et al. 1997). As a result, we are still researching the application of suggested approaches to real-world problems.

The main contributions of the paper are:

1. The application of global optimization utilizing AO–AOA gives better results when experimental results are compared with AOA, AO, GOA, GWO.
2. AO–AOA reduced the computational complexity and it also works efficiently for both high- and low-dimensional problems.

The paper is organized in such a way that Sect. 2 expounds on AO. Section 3 delves into AOA. Section 4 addresses the proposed work. Section 5 discusses the findings. Section 6 discusses the conclusion and future scope.

2 Aquila optimizer

Aquila optimizer (AO) (Abualigah et al. 2021) is a nature-inspired algorithm. The Aquila is among the most common birds of prey in the Northern Hemisphere. Aquila is perhaps the most commonly distributed species of Aquila. Aquila, like all birds, is a member of the Accipitridae family. Aquila is usually dark brown with light Golden-brown plumage on the back of the body. Young Aquilas in this category have a white tail and typically have minor white markings on their wings. Aquila utilizes its strength and agility, as well as its strong feet and wide, sharpened nails, to capture a variety of prey, primarily rabbits, hares, deeps, marmots, squirrels, and other ground animals (Steenhof et al. 1997). Aquila and their distinct behaviors can be seen in the wild.

2.1 Motivation and its behavior while catching prey

Aquila retains territory that can reach up to 200 km. They build large nests in mountains and other high places. Breeding begins in the spring; they are monogamous and live together for several years, if not their entire lives. Females can lay up to four eggs, which they incubate for six weeks. In most cases, one or two newborns survive to fledge in around 12 weeks.

These young Aquila normally gain complete trust in the fall, after which they spread out to create a territory for themselves. Aquila is one of the most studied birds in the world due to its hunting courage.

When male Aquilas hunted alone, they caught substantially more prey. Aquila hunts squirrels, rabbits, and a variety of other species with their speed and sharp talons. They've even been identified as a threat to full-grown deer (Hatch 1968). Land squirrels are the most notable species in Aquila's diet.

The Aquila is known to use four hunting methods, with several distinct variations and the majority of Aquila's ability to cleverly and easily switch back and forth between hunting methods depending on the situation.

Four main points used by Aquila for hunting are

1. Contour flight along with glide attack (Meinertzhagen 1940).
2. High soar along with vertical stoop (Carnie 1954).
3. The slow attack along with low flight (Dekker 1985).
4. Grabbing the prey while walking (Watson 2010).

2.2 Initialization of solution

AO is a population-dependent approach in which the optimization rule begins with the population of candidate solutions (X) as shown in Eq. 1, which is produced stochastically here between the given problem's upper bound (UB) and lower bound (LB). In each iteration, the optimum solution is calculated as the optimal solution (Abualigah et al. 2021).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \vdots & \vdots & x_{1,\mathrm{Dim}-1} & x_{1,\mathrm{Dim}} \\ x_{2,1} & x_{2,2} & \vdots & \vdots & x_{2,\mathrm{Dim}-2} & x_{2,\mathrm{Dim}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \vdots & \vdots & \vdots & \vdots & x_{N-1,\mathrm{Dim}} \\ x_{N,1} & \vdots & \vdots & \vdots & x_{N,\mathrm{Dim}-1} & x_{N,\mathrm{Dim}} \end{bmatrix} \tag{1}$$

where $X =$ solution of current candidate; X is randomly generated by Eq. 2

$$X_{i,j} = \mathrm{rand} \times \left(\mathrm{UB}_j - \mathrm{LB}_j\right) + \mathrm{LB}_j \tag{2}$$

where $X_i =$ value of decision of ith solution; $N =$ No. of candidate solution; Dim = Dimension Size; rand = random no.; $\mathrm{LB}_j =$ lower bound of jth; $\mathrm{UB}_j =$ upper bound of jth; $i = 1, 2..., N$; $j = 1, 2..., \mathrm{Dim}$.

2.3 Mathematically representation of AO

The AO (Abualigah et al. 2021) method represents Aquila's hunting behavior, displaying each phase of the hunt's behaviors. As a result, the proposed AO algorithm's optimization procedures are divided into four methods. AO can be shifted from exploration to exploitation by utilizing various behavioral conditions like

$$\text{If}, \begin{cases} t \leq \left(\frac{2}{3}\right) \times T; \text{ execution of exloration steps} \\ \text{otherwise; execution of exploitation steps} \end{cases}$$

Four steps are involved in a mathematical model of AO (Abualigah et al. 2021).

1. *Expanded Exploration* (X_1) In this step, high soar along with vertical stoop behavior of Aquila is mathematically represented in Eq. 3

$$X_1(t+1) = X_{\text{best}}(t) \times \left(1 - \frac{t}{T}\right) + (X_M(t) - X_{\text{best}}(t) \times \text{rand}) \quad (3)$$

where $X_1(t+1)$ = next iteration's solution for t; $X_{\text{best}}(t)$ = optimum solution till tth iteration; $\left(1 - \frac{t}{T}\right)$ = for controlling exploration; $X_M(t)$ = mean value of solution at tth iteration, evaluated using Eq. 4; rand = random value within 0 – 1; T = max. iterations; t = present iteration

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t), \quad \forall j = 1, 2, \ldots, \text{Dim} \quad (4)$$

Dim = size of dimension; N = no. of candidate.

2. *Narrow Exploration* (X_2) In this contour flight along with glide attack behavior of Aquila is mathematically represented in Eq. 5

$$X_2(t+1) = X_{\text{best}}(t) \times \text{Levy}(D) + X_R(t) + (y - x) \times \text{rand} \quad (5)$$

$X_2(t+1)$ = next iteration's solution for t; D = space of dimension; Levy (D) = flight distribution function of levy, evaluated using Eq. 6; $X_R(t)$ = random solution within [1 N] at ith iteration

$$\text{Levy } (D) = s \times \frac{u \times \sigma}{|\upsilon|^{\frac{1}{\beta}}} \quad (6)$$

where s = constant values (0.01); u and υ = random no. within 0 – 1. σ is evaluated by Eq. 7

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin e\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2\left(\frac{\beta-1}{2}\right)}\right) \quad (7)$$

where β = constant value (1.5); Spiral shape is presented by y and x in Eq. 5, are evaluated as

$$y = r \times \cos(\theta) \quad (8)$$

$$x = r \times \sin(\theta) \quad (9)$$

where

$$r = r_1 + U + D_1 \quad (10)$$

$$\theta = -\omega \times D_1 + \theta_1 \quad (11)$$

$$\theta_1 = \frac{3 \times \pi}{2} \quad (12)$$

r_1 has value range from 1 to 20 for no. of search cycles. U = 0.00565; D_1 = integer no; ω = 0.005.

3. *Expanded Exploitation* (X_3) In this slow attack, along with the low flight behavior of Aquila is mathematically represented in Eq. 13

$$X_3(t+1) = (X_{\text{best}}(t) - X_M(t)) \times \alpha - \text{rand} + ((\text{UB} - \text{LB}) \times \text{rand} + \text{LB}) \times \delta \quad (13)$$

where $X_3(t+1)$ = next iteration's solution for t; $X_{\text{best}}(t)$ = supposed location of prey till ith iteration; $X_M(t)$ = mean value of solution at tth iteration, evaluated using Eq. 4; rand = random no. b/w 0 – 1; α and δ = adjustable exploitation parameters (0.1); UB and LB = upper and lower bound.

4. *Narrowed Exploitation* (X_4) In this grabbing the prey while walking behavior of Aquila is mathematically represented in Eq. 14

$$X_4(t+1) = QF \times X_{\text{best}}(t) - (G_1 \times X(t) \times \text{rand}) - G_2 \times \text{Levy}(D) + \text{rand} \times G_1 \quad (14)$$

where $X_4(t+1)$ = next iteration's solution for t; QF = quality function, evaluated by using Eq. 15; G_1 = different motions of AO evaluated using Eq. 16; G_2 = decreasing value b/w 2 – 0.

When preys escape from 1st place to last place, then slope used by AO to catch the prey is evaluated by Eq. 17

$$\text{QF}(t) = t^{\frac{2 \times \text{rand}() - 1}{(1-T)^2}} \quad (15)$$

$$G_1 = 2 \times \text{rand}() - 1 \tag{16}$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{17}$$

where $\text{QF}(t) =$ quality function at ith iteration; $rand =$ random no. b/w $0 - 1$; $T =$ max. iterations; $t =$ present iteration.

3 Arithmetic optimization algorithm

AOA (Abualigah et al. 2020) is a new meta-heuristic method that uses common mathematical operations such as Division (D), Addition (A), Multiplication (M), Subtraction (S), as shown in Fig. 1, which is applied and modeled to execute optimization in a wide variety of search fields (Abualigah et al. 2020). Commonly, population-based algorithms (PBA) launch their improvement processes by randomly selecting several candidate strategies. This defined solution is enhanced incrementally by a set of optimization standards and analyzed sequentially by a particular objective function, and that's the basis of optimization techniques. Although PBA is stochastically trying to find some efficient strategy to optimization problems, a single-run solution is not guaranteed. However, the chance of an optimum global solution to the problem is enhanced by a large set of possible solutions and optimization simulations (Gogna and Tayal 2013). Considering variations among meta-heuristic methods in PBA approaches, the optimization process comprises two cycles: exploitation vs. exploration. The previous examples for extensive coverage are search fields through search agents to bypass local solutions. Above is an increase in the performance of solutions achieved during the exploration process.

3.1 Motivation

Arithmetic is a key component of mathematics and its most important components of modern math, including analysis, geometry and algebra. Arithmetic operators (AO) (Abualigah et al. 2021) are traditionally used for the study of numbers (Abualigah et al. 2020). These basic math functions are used for optimization for finding ideal elements, particularly with selected solutions. Optimization challenges have appeared in all mathematical fields, such as engineering, economics and computer science to organizational analysis and technology, and the advancement of optimization methods has drawn mathematics attention from time to time. The key motivation of the new AOA is the use of AO to solve problems. The behavior of AO and their effect mostly on existing algorithms, the arrangement of AO and their superiority is shown in Fig. 2. AOA is then proposed based on a statistical model.

3.2 Initial stage

The method of optimization starts with selected sets denoted by A as in Eq. 18. The ideal set in every iteration is

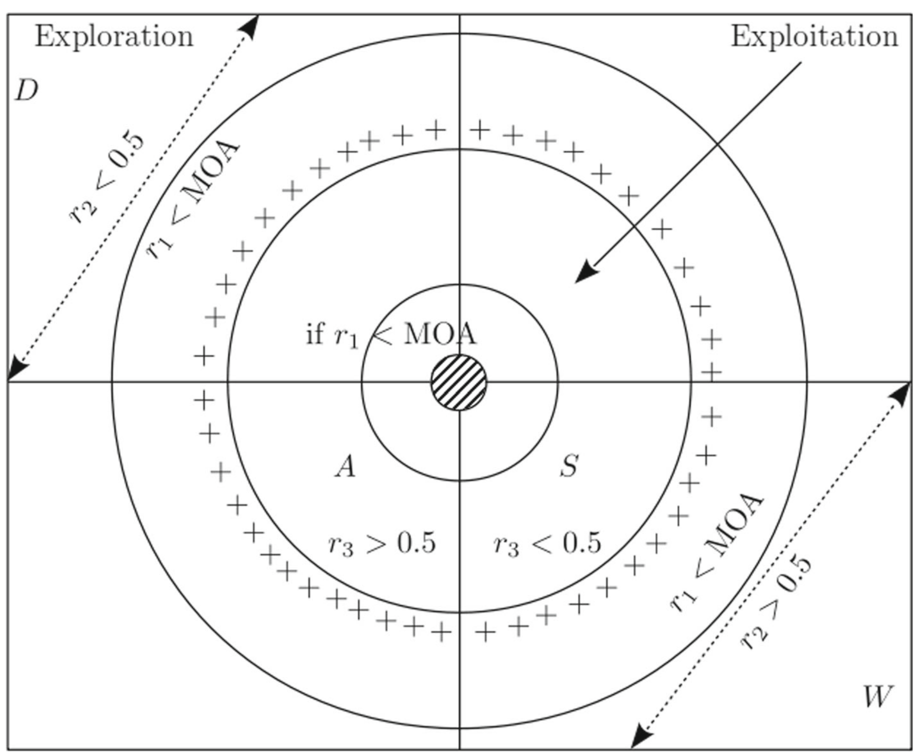


Fig. 1 Shows AOA search phases (Abualigah et al. 2020)

created randomly and is taken as the optimum solution (Abualigah et al. 2020).

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,j} & a_{1,1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & \cdots & a_{2,j} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N-1,1} & \cdots & \cdots & \cdots & a_{N-1,j} & \cdots & a_{N-1,n} \\ a_{N,1} & \cdots & \cdots & \cdots & a_{N,j} & a_{N,n-1} & a_{N,n} \end{bmatrix} \tag{18}$$

Exploitation/Exploration should be carefully chosen at the start of AOA. The coefficient of math optimizer accelerated (MOA) is defined in Eq. 19.

$$\mathrm{MOA}(C_{\mathrm{iter}}) = \mathrm{Min} + C_{\mathrm{iter}} \times \left(\frac{\mathrm{Max} - \mathrm{Min}}{M_{\mathrm{iter}}}\right) \tag{19}$$

where $\mathrm{MOA}(C_{\mathrm{iter}}) = i$th iteration function value; $M_{\mathrm{iter}} =$ Max. no. of iteration; Max and Min $=$ Accelerated Function of Max. and Min. Values; $C_{\mathrm{iter}} =$ current iteration (within 1 and M_{iter}).

$$a_{i,j}(C_{\mathrm{iter}}+1) = \begin{cases} \mathrm{besta}_j - \mathrm{MOP} \times \left((\mathrm{UB}_j - \mathrm{LB}_j) \times \mu + \mathrm{LB}_j\right), & r_3 < 0.5 \\ \mathrm{besta}_j + \mathrm{MOP} \times \left((\mathrm{UB}_j - \mathrm{LB}_j) \times \mu + \mathrm{LB}_j\right), & \text{otherwise} \end{cases} \tag{22}$$

3.3 Exploration stage

The exploratory nature of AOA is discussed, as per the AO, mathematical calculations whether using Division (D) or Multiplication (M) operator have obtained high distribution values or decisions that contribute to an exploration search method. However, as opposed to other operators, these D and M operators never easily reach the objective due to the high distribution of S and A operators. AOA exploration operators exploit the search field arbitrarily through many regions and seek a better alternative dependent on two key search techniques M and D search techniques as shown in Eq. 20 (Yang et al. 2021).

$$a_{i,j}(C_{\mathrm{iter}}+1) = \begin{cases} \mathrm{besta}_j \div (\mathrm{MOP} \div \varepsilon) \times \left((\mathrm{UB}_j - \mathrm{LB}_j) \times \mu + \mathrm{LB}_j\right), & r_2 < 0.5 \\ \mathrm{besta}_j \times \mathrm{MOP} \times \left((\mathrm{UB}_j - \mathrm{LB}_j) \times \mu + \mathrm{LB}_j\right), & \text{otherwise} \end{cases} \tag{20}$$

where $a_i(C_{\mathrm{iter}}+1) = i$th solution of next iteration; $a_{i,j}(C_{\mathrm{iter}}+1) = j$th position in current iteration; $\mu =$ control parameter ≤ 0.5; LB_j and $\mathrm{UB}_j =$ Lower and Upper bound limit; $\varepsilon =$ smallest integer no.; $\mathrm{besta}_j = j$th position of optimum solution till now

$$\mathrm{MOP}(C_{\mathrm{iter}}) = 1 - \frac{C_{\mathrm{iter}}^{\frac{1}{\alpha}}}{M_{\mathrm{iter}}^{\frac{1}{\alpha}}} \tag{21}$$

where Math Optimizer Probability (MOP) = coefficient; $\mathrm{MOP}(C_{\mathrm{iter}}) = i$th iteration function value; $C_{\mathrm{iter}} =$ current iteration; $M_{\mathrm{iter}} =$ Max. iterations ≤ 5.

3.4 Exploitation stage

The exploitation nature of AOA is discussed, as per AO mathematical formulas, whether using addition (A) or subtraction (S) as they provided high-density results. AOA exploitation operators exploit the search field deeply through many regions and seek a better alternative dependent on two key search techniques A and S search techniques as shown in Eq. 22 (Abualigah et al. 2020).

4 Proposed work

A variety of population-based methods have recently been suggested. Despite their widespread use in various engineering applications, we are still investigating suggested methods for solving real-world problems. As a result, researchers must dramatically change and develop their approaches, often focused on major evolutionary processes, to achieve faster integration, consistent balance with high-quality performance, and optimization. Thus, a new hybrid method using Aquila Optimizer (AO) and Arithmetic Optimization Algorithm (AOA) is proposed in this paper. Aquila Optimizer (AO) (Abualigah et al. 2021) is a nature-inspired algorithm. The Aquila is among the most common birds of prey in the Northern Hemisphere. Aquila is perhaps the most commonly distributed species of Aquila.

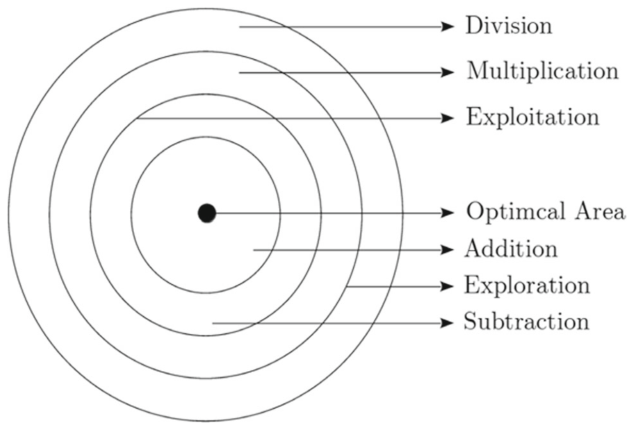


Fig. 2 Shows arithmetic operators according to superiority

Aquila, like all birds, is a member of the Accipitridae family.

Arithmetic is a key component of mathematics and its most important components of modern math, including analysis, geometry and algebra. Arithmetic operators (AO) (Abualigah et al. 2021) are traditionally used for the study of numbers (Gogna and Tayal 2013).

AO and AOA are both modern meta-heuristic optimization methods. They can be applied to different problems, including image processing, machine learning, wireless networks, power systems, engineering design etc. To determine efficiency, each technique is evaluated using the same parameters, such as population size and no. of iterations. The proposed method (AO–AOA) is evaluated by varying the dimensions. The impact of varying dimensions is a standard test utilized in previous studies for optimizing test functions that show the effect of varying dimensions on AO–AOA efficiency. It is clear from this that it fits well with both high- and low-dimensional problems. Population-based methods produce efficient search results in high-dimensional problems. The complete step-by-step overview and working of the proposed model is represented in Fig. 3.

From Fig. 3, the different steps used in the proposed method according to their executions are explained. The 1st step is determining parameters in this different parameter to be used are determined or defined. The 2nd step is to generate the solution using the defined parameters. The 3rd step is to calculate the fitness function followed by selecting the best solution in 4th step. In 5th step, there is a condition, if the value of the random number (rand) is greater than 0.5, then AO is not used, and if the value is less than 0.5, AOA is used. Then in 6th step, if the desired criteria is met, then the best solution is returned in 7th step; otherwise, by feedback lope, it is again fed to 3rd step for calculating the fitness function.

5 Results and discussion

The proposed approach is examined, and the proposed system's efficiency is correlated with the performance of existing methods. The implementation and testing are done in $i5 - 1.70\,\mathrm{GHz}$ processor using MATLAB software. The proposed method's performance (i.e., AO + AOA) is examined on five engineering problems and 23 test functions. The results are further compared with AOA, PSO,

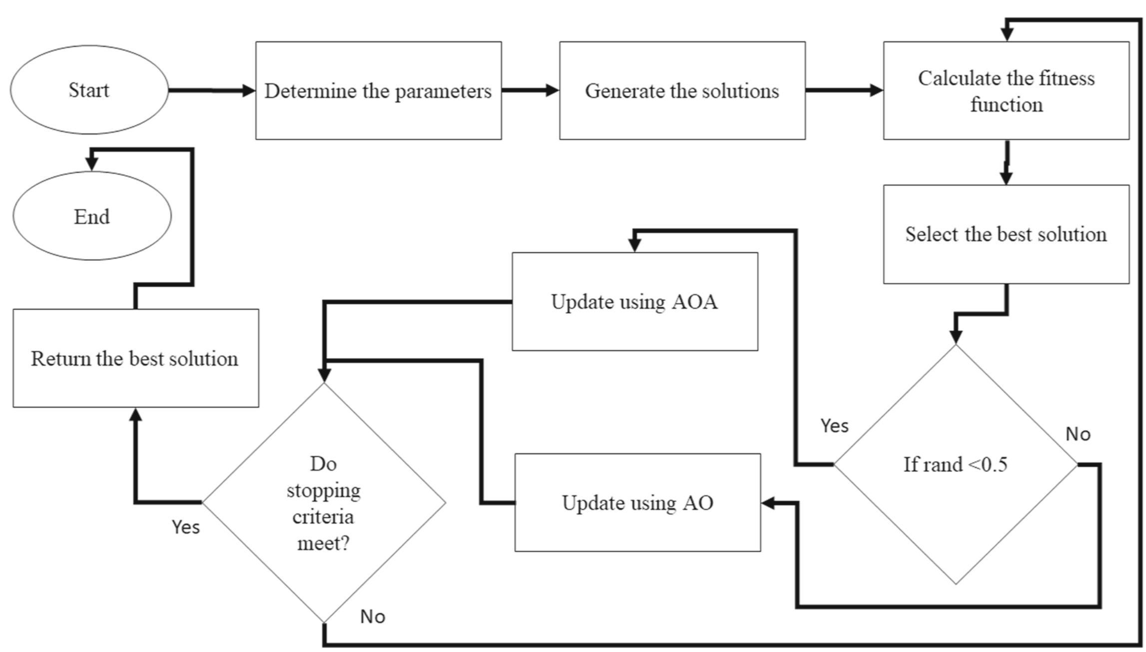


Fig. 3 Shows step-by-step working of the proposed method

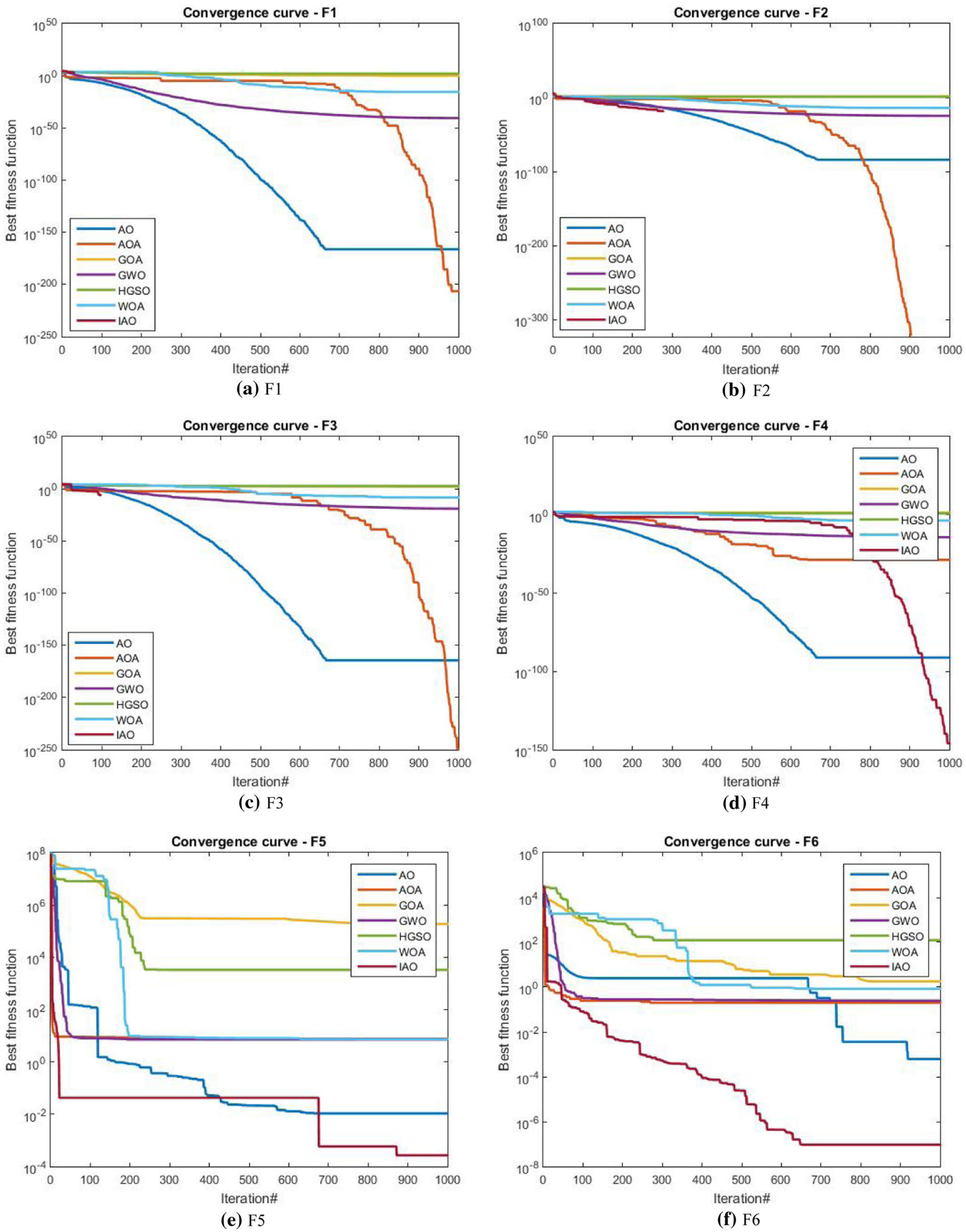
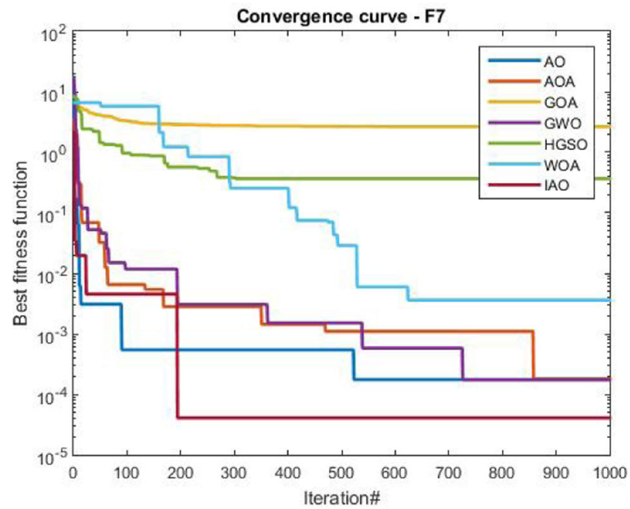
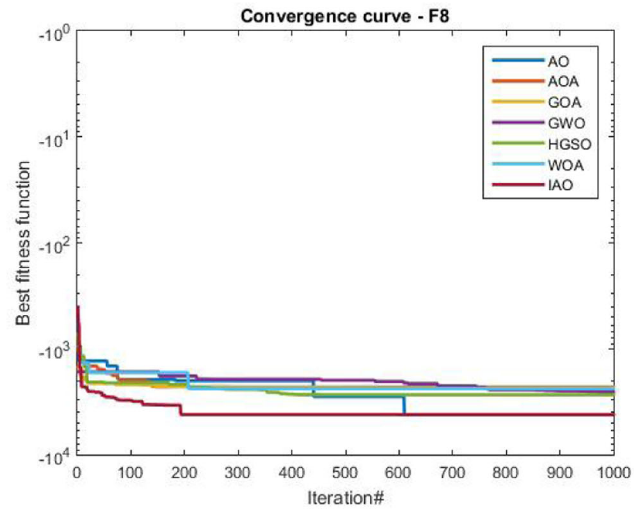


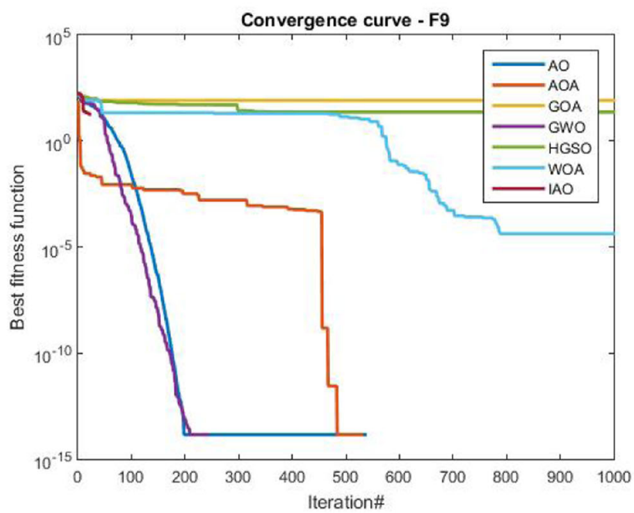
Fig. 4 Convergence behavior of AO–AOA in contrast with other methods



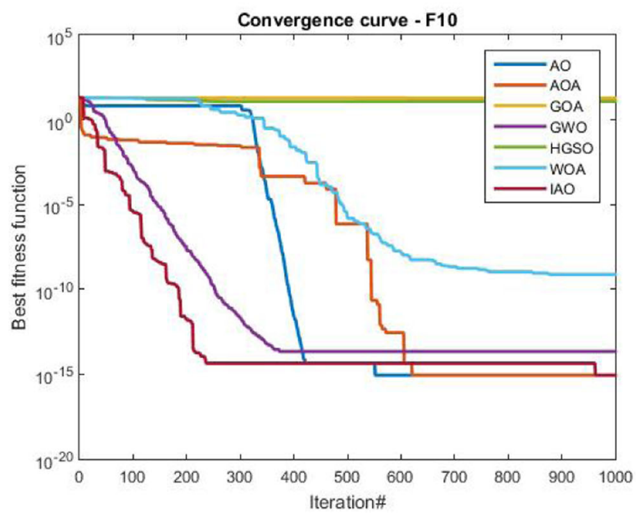
(g) F7



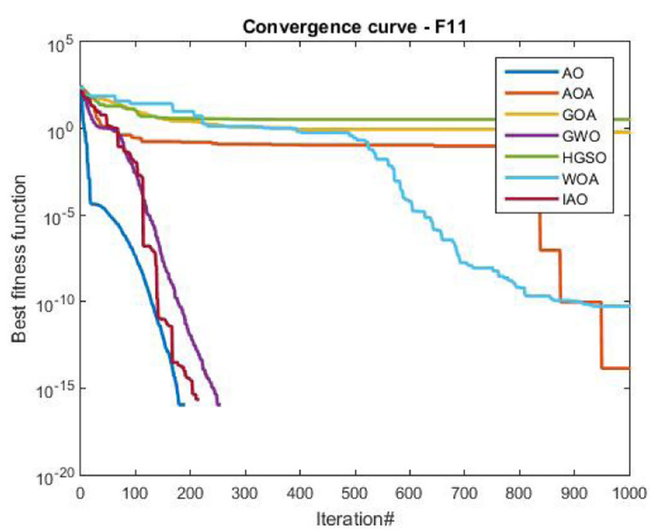
(h) F8



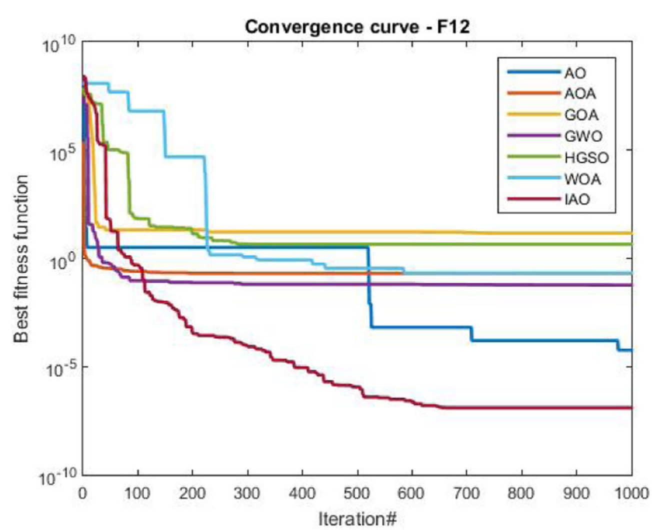
(i) F9



(j) F10

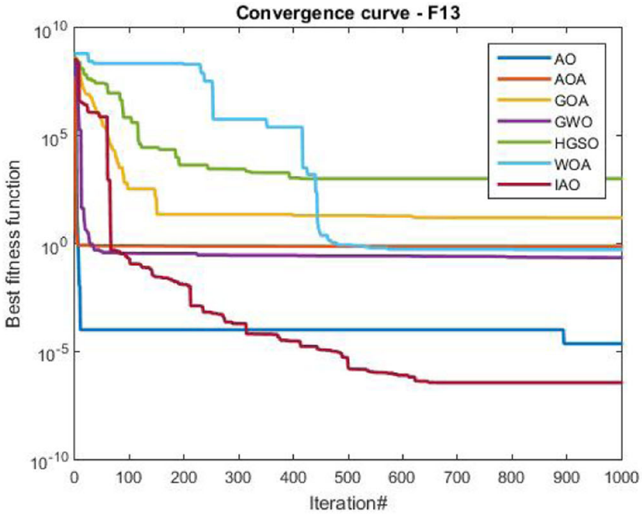


(k) F11

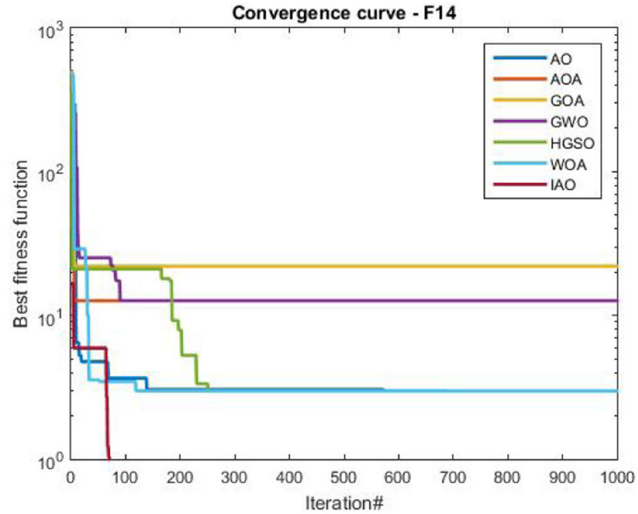


(l) F12

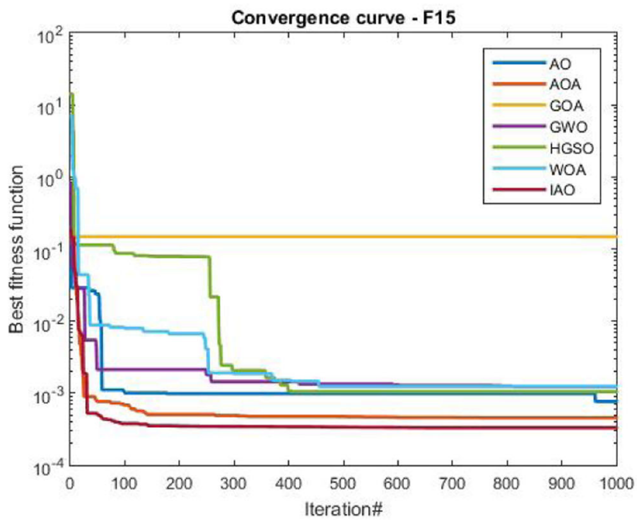
Fig. 4 continued



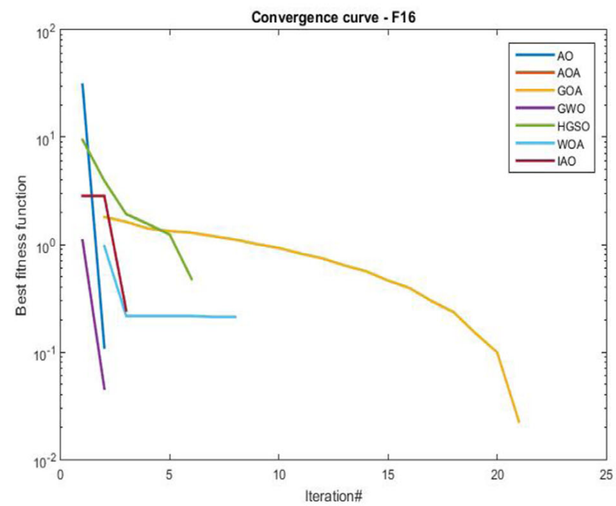
(m) F13



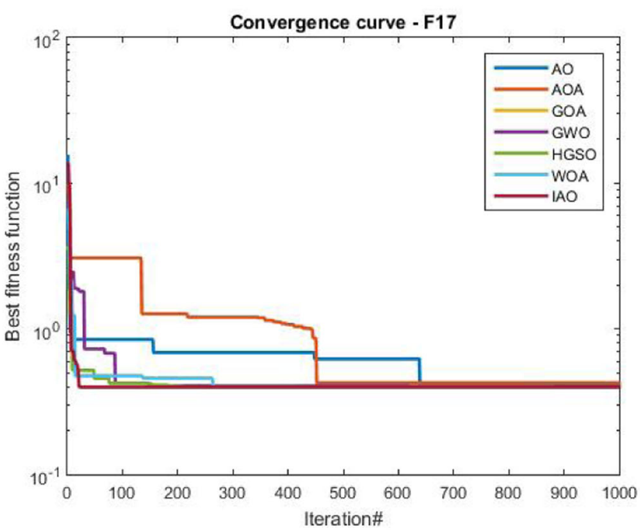
(n) F14



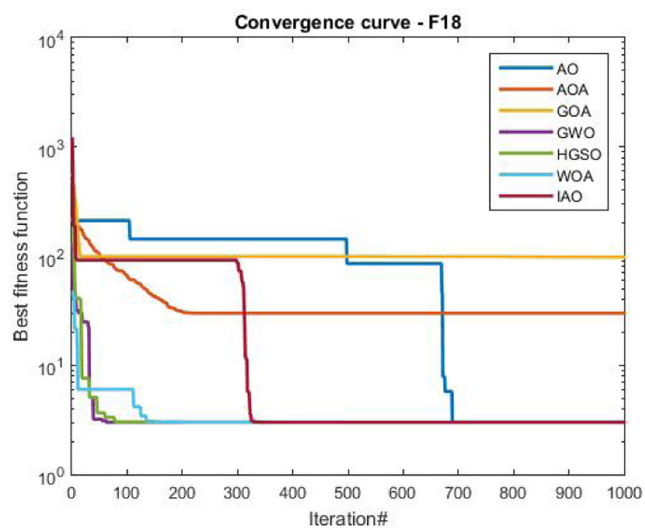
(o) F15



(p) F16

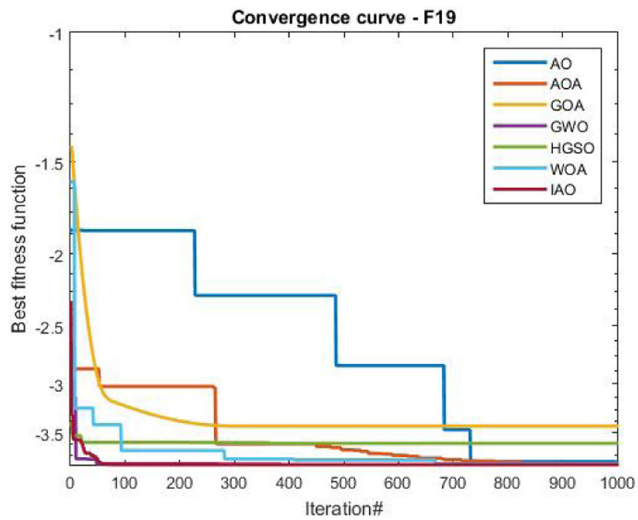


(q) F17

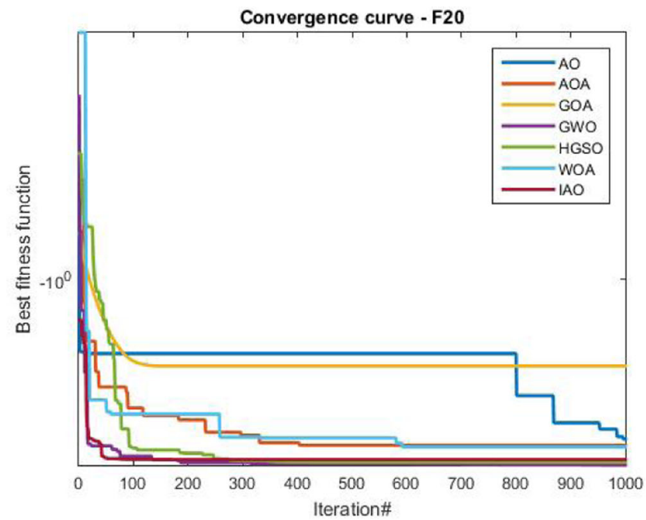


(r) F18

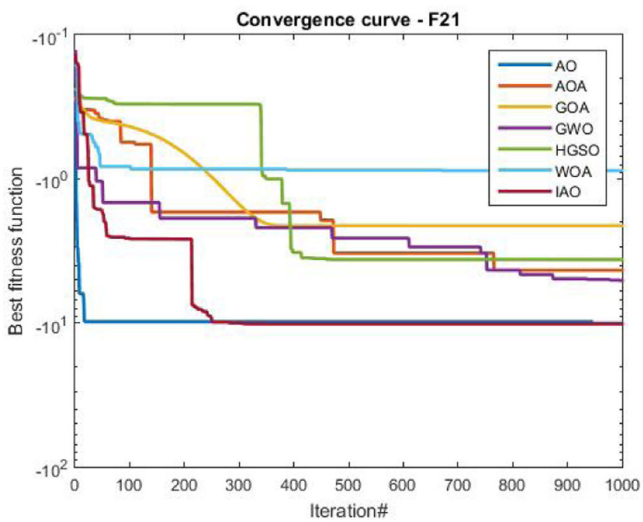
Fig. 4 continued



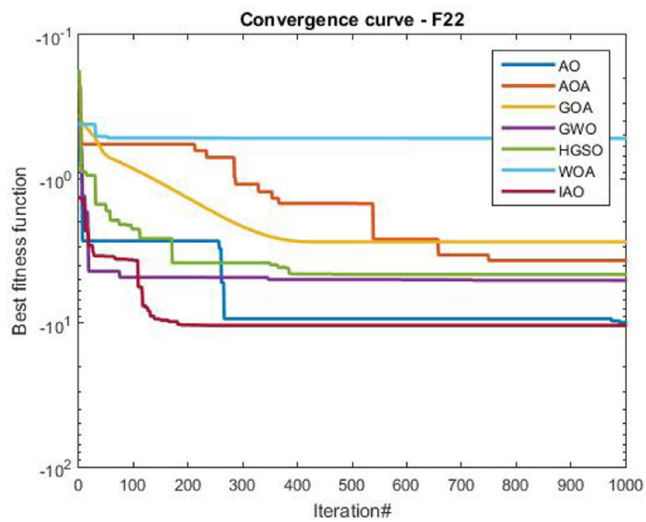
(s) F19



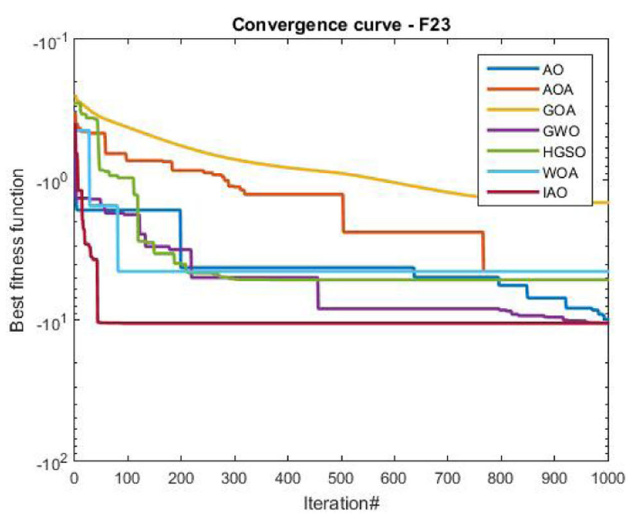
(t) F20



(u) F21



(v) F22



(w) F23

Fig. 4 continued

Table 1 The comparative methods result using thirteen benchmark functions (F1–F13), where the dimension is fixed to 10

Function	Measure	AO	AOA	GOA	GWO	PSO	WOA	IAO
F1	Worst	1.87178E−165	5.95285E−86	1.80494E+00	3.88291E−40	8.66936E+01	8.99456E−15	6.34689E−173
	Average	5.03287E−166	1.19057E−86	6.76445E−01	7.76979E−41	5.70671E+01	1.96166E−15	1.27856E−173
	Best	2.22567E−170	0.00000E+00	1.96397E−01	3.72354E−45	2.30723E+01	5.89790E−23	0.00000E+00
	STD	0.00000E+00	2.66220E−86	6.77123E−01	1.73627E−40	2.92968E+01	3.94090E−15	0.00000E+00
	Rank	2	3	6	4	7	5	1
F2	Worst	4.54814E−80	0.00000E+00	4.93240E+01	2.87222E−23	2.30831E+01	6.56867E−12	0.00000E+00
	Average	9.52414E−81	0.00000E+00	3.68587E+01	1.09090E−23	1.42065E+01	1.36462E−12	0.00000E+00
	Best	6.13125E−88	0.00000E+00	9.32233E+00	1.03890E−25	2.01847E+00	6.50645E−16	0.00000E+00
	STD	2.01086E−80	0.00000E+00	1.67932E+01	1.27163E−23	8.43347E+00	2.91117E−12	0.00000E+00
	Rank	3	1	7	4	6	5	1
F3	Worst	8.27862E−74	1.12207E−75	1.16647E+02	1.05462E−15	6.98618E+03	2.05634E+00	1.52887E−157
	Average	1.65572E−74	2.24415E−76	8.10550E+01	2.11919E−16	3.18133E+03	4.33646E−01	3.05803E−158
	Best	0.00000E+00	7.48740E−235	2.86679E+01	1.83222E−19	4.85648E+02	1.73108E−05	2.03329E−191
	STD	3.70231E−74	5.01807E−76	3.73260E+01	4.71089E−16	2.89172E+03	9.08295E−01	6.83715E−158
	Rank	3	2	6	4	7	5	1
F4	Worst	2.05862E−18	5.75941E−19	1.34807E+01	4.86175E−13	2.61709E+01	5.02979E−02	1.99960E−81
	Average	4.11723E−19	1.15188E−19	5.31466E+00	1.07983E−13	1.34657E+01	1.08599E−02	4.00072E−82
	Best	0.00000E+00	3.38838E−127	9.38980E−01	1.35179E−15	3.62088E+00	8.70927E−05	1.35352E−85
	STD	9.20641E−19	2.57568E−19	5.64938E+00	2.11747E−13	8.89794E+00	2.20593E−02	8.94166E−82
	Rank	3	2	6	4	7	5	1
F5	Worst	7.08082E+00	8.28888E+00	9.12364E+05	8.91936E+00	2.13045E+04	8.11556E+00	6.49158E−02
	Average	6.54503E+00	7.90900E+00	1.90504E+05	7.64726E+00	9.74644E+03	7.97902E+00	2.64899E−02
	Best	6.19033E+00	7.41463E+00	2.75657E+03	6.21354E+00	1.48133E+03	7.55149E+00	3.21005E−03
	STD	3.40149E−01	3.21841E−01	4.03634E+05	1.14486E+00	9.14490E+03	2.41048E−01	2.38668E−02
	Rank	2	4	7	3	6	5	1
F6	Worst	2.50003E−01	1.81123E−01	1.54710E+00	7.53494E−01	1.31977E+03	1.02994E+00	7.40253E−04
	Average	5.00010E−02	1.17664E−01	9.46679E−01	2.50407E−01	3.17362E+02	7.78419E−01	3.88728E−04
	Best	8.76702E−08	4.36626E−02	5.18948E−01	3.55586E−06	1.58145E+01	5.29587E−01	1.63773E−05
	STD	1.11804E−01	5.70816E−02	4.27931E−01	3.07609E−01	5.64984E+02	2.08309E−01	2.81340E−04
	Rank	2	3	6	4	7	5	1
F7	Worst	6.82394E−04	4.63798E−04	1.27433E+01	4.83047E−03	1.85919E−01	1.17970E−02	3.76671E−04
	Average	2.79568E−04	1.93034E−04	5.32109E+00	1.57668E−03	7.77608E−02	4.03989E−03	1.84473E−04
	Best	3.01117E−05	8.32353E−06	2.20975E+00	4.19584E−04	3.84239E−03	7.04483E−04	4.19465E−05
	STD	2.45527E−04	1.91012E−04	4.26684E+00	1.83428E−03	6.90218E−02	4.63920E−03	1.40022E−04
	Rank	3	2	7	4	6	5	1

Table 1 (continued)

Function	Measure	AO	AOA	GOA	GWO	PSO	WOA	IAO
F8	Worst	− 1.62546E+03	− 2.14758E+03	− 1.99643E+03	− 1.70811E+03	− 1.56191E+03	− 1.74673E+03	− 3.00544E+03
	Average	− 2.63402E+03	− 2.42020E+03	− 2.22654E+03	− 2.19257E+03	− 1.93418E+03	− 1.93330E+03	− 3.81025E+03
	Best	− 4.18931E+03	− 2.57202E+03	− 2.43115E+03	− 2.56685E+03	− 2.19727E+03	− 2.33646E+03	− 4.18981E+03
	STD	9.66339E+02	1.63440E+02	1.75977E+02	3.71229E+02	2.55106E+02	2.31266E+02	5.45331E+02
	Rank	2	3	4	5	6	7	1
F9	Worst	0.00000E+00	0.00000E+00	1.18821E+02	6.34889E+00	8.19711E+01	2.06339E−08	0.00000E+00
	Average	0.00000E+00	0.00000E+00	7.18626E+01	1.26978E+00	5.45968E+01	4.18099E−09	0.00000E+00
	Best	0.00000E+00	0.00000E+00	3.36272E+01	0.00000E+00	4.23615E+01	2.13163E−13	0.00000E+00
	STD	0.00000E+00	0.00000E+00	3.12172E+01	2.83931E+00	1.68937E+01	9.19820E−09	0.00000E+00
	Rank	1		7	5	6	4	1
F10	Worst	4.35207E−14	8.88178E−16	1.95733E+01	2.22045E−14	1.39229E+01	4.50625E+00	8.88178E−16
	Average	9.41469E−15	8.88178E−16	1.81827E+01	1.79412E−14	7.53756E+00	1.11584E+00	8.88178E−16
	Best	8.88178E−16	8.88178E−16	1.37090E+01	1.50990E−14	4.48869E+00	1.37175E−10	8.88178E−16
	STD	1.90659E−14	0.00000E+00	2.51911E+00	3.89180E−15	3.67824E+00	1.95141E+00	0.00000E+00
	Rank	3		7	4	6	5	1
F11	Worst	0.00000E+00	7.59714E−02	7.27276E−01	6.32900E−02	1.85096E+01	1.99745E−01	0.00000E+00
	Average	0.00000E+00	1.51943E−02	5.21261E−01	2.54921E−02	5.93625E+00	4.00130E−02	0.00000E+00
	Best	0.00000E+00	0.00000E+00	4.24584E−01	0.00000E+00	1.08592E+00	0.00000E+00	0.00000E+00
	STD	0.00000E+00	3.39754E−02	1.25051E−01	2.93593E−02	7.18836E+00	8.92930E−02	0.00000E+00
	Rank	1		6	4	7	5	1
F12	Worst	2.81015E−04	1.66617E−01	2.78509E+01	5.92910E−02	1.02110E+01	3.51549E−01	6.63046E−08
	Average	7.11317E−05	1.06452E−01	2.06176E+01	2.76714E−02	4.98283E+00	2.07096E−01	2.82496E−08
	Best	2.32159E−06	7.16804E−02	1.16056E+01	2.18265E−05	1.41457E+00	6.14444E−02	6.66266E−09
	STD	1.18548E−04	3.76987E−02	6.25560E+00	2.23728E−02	3.50712E+00	1.21778E−01	2.38807E−08
	Rank	2		7	3	6	5	1
F13	Worst	5.15975E−04	9.96230E−01	6.85904E+03	4.32314E−01	2.62748E+02	6.10107E−01	9.37688E−07
	Average	2.14531E−04	9.60893E−01	1.37616E+03	2.87030E−01	7.04794E+01	4.68843E−01	2.32583E−07
	Best	1.94947E−05	8.21551E−01	4.78581E−01	4.24150E−05	6.63077E+00	2.18322E−01	1.79266E−08
	STD	2.03320E−04	7.78956E−02	3.06503E+03	1.70813E−01	1.07868E+02	1.84048E−01	3.96678E−07
	Rank	2		7	3	6	4	1
Summation		29	20	83	51	83	65	13
Mean ranking		2.23	2.50	6.38	3.92	6.38	5.00	1.00
Final ranking		3	2	6	4	6	5	1

Table 2 The comparative methods result using thirteen benchmark functions (F1–F13), where the dimension is fixed to 100

Function	Measure	AO	AOA	GOA	GWO	PSO	WOA	IAO
F1	Worst	1.00916E−83	6.82637E−15	8.84320E+02	1.37630E−21	1.74675E+04	4.65975E+01	4.31344E−158
	Average	2.01832E−84	1.67698E−15	5.08868E+02	3.81367E−22	6.96060E+03	1.30791E+01	8.62688E−159
	Best	0.00000E+00	1.13759E−44	2.86987E+02	1.74736E−24	2.68736E+03	4.21708E−01	1.45360E−178
	STD	4.51310E−84	2.95665E−15	2.22986E+02	5.84969E−22	6.31639E+03	1.90995E+01	1.92903E−158
	Rank	2	4	6	3	7	5	1
F2	Worst	8.55679E−82	1.89604E−141	7.37512E+07	1.58820E−13	9.96710E+01	1.02952E−03	0.00000E+00
	Average	1.78561E−82	3.79216E−142	1.47589E+07	8.27059E−14	6.83324E+01	6.32073E−04	0.00000E+00
	Best	8.80005E−87	9.12273E−181	1.45383E+02	7.35845E−15	2.49651E+01	2.61222E−04	0.00000E+00
	STD	3.78714E−82	8.47931E−142	3.29777E+07	5.82134E−14	3.15898E+01	3.15012E−04	0.00000E+00
	Rank	3		7	4	6	5	1
F3	Worst	2.02316E−40	4.44017E−02	2.31993E+04	9.91480E−02	6.30170E+04	2.47985E+04	3.17092E−168
	Average	4.06112E−41	1.42933E−02	1.45503E+04	2.23799E−02	4.51150E+04	1.93048E+04	8.24766E−169
	Best	0.00000E+00	1.59603E−30	1.06051E+04	1.00236E−04	3.70944E+03	6.16286E+03	1.30205E−175
	STD	9.03963E−41	1.86235E−02	5.07598E+03	4.30377E−02	2.35857E+04	7.56700E+03	0.00000E+00
	Rank	2	3	5	4	7	6	1
F4	Worst	4.13824E−14	5.48822E−02	5.78794E+01	3.03674E−05	4.80257E+01	5.37425E+01	4.68455E−82
	Average	8.27682E−15	4.46541E−02	5.05136E+01	1.36701E−05	2.74560E+01	3.96281E+01	9.41755E−83
	Best	0.00000E+00	3.54891E−02	4.20212E+01	1.64653E−06	1.03937E+01	2.71121E+01	4.10965E−86
	STD	1.85066E−14	6.88397E−03	7.72696E+00	1.34390E−05	1.40062E+01	1.24210E+01	2.09231E−82
	Rank	2	4	7	3	5	6	1
F5	Worst	2.75404E+01	2.87438E+01	1.43276E+07	2.88305E+01	7.71681E+06	1.63072E+06	3.66393E−02
	Average	1.63003E+01	2.85897E+01	8.84974E+06	2.80158E+01	2.42276E+06	5.69703E+05	2.02492E−02
	Best	1.71002E−01	2.82373E+01	2.45215E+06	2.71912E+01	7.20236E+05	3.59478E+03	5.89349E−05
	STD	1.46453E+01	2.06126E−01	4.28050E+06	5.79776E−01	3.00770E+06	6.62559E+05	1.53717E−02
	Rank	2	4	7	3	6	5	1
F6	Worst	2.00453E+00	4.30398E+00	1.22868E+03	3.51384E+00	9.93863E+03	3.18031E+02	1.66633E−03
	Average	4.01645E−01	3.69667E+00	6.75700E+02	2.65763E+00	3.97243E+03	7.41433E+01	5.28743E−04
	Best	5.62535E−06	3.31343E+00	3.18600E+02	1.68145E+00	8.45431E+02	5.46143E+00	4.12683E−05
	STD	8.96042E−01	3.90966E−01	3.57181E+02	7.36150E−01	3.51133E+03	1.36875E+02	7.15011E−04
	Rank	2	4	6	3	7	5	1
F7	Worst	6.35008E−04	8.57393E−04	1.00840E+02	7.59219E−03	9.45597E+00	6.85132E−01	5.33817E−04
	Average	4.10048E−04	3.30177E−04	7.61036E+01	5.39887E−03	3.69371E+00	1.90882E−01	1.77227E−04
	Best	2.48080E−06	1.01731E−05	4.34018E+01	1.39239E−03	1.46457E−01	1.79598E−02	1.70718E−07
	STD	2.66095E−04	3.49278E−04	2.32379E+01	2.55416E−03	3.77766E+00	2.88025E−01	2.29609E−04
	Rank	3	2	7	4	6	5	1

Table 2 (continued)

Function	Measure	AO	AOA	GOA	GWO	PSO	WOA	IAO
F8	Worst	− 2.78794E+03	− 4.06457E+03	− 6.42950E+03	− 4.08663E+03	− 3.75012E+03	− 3.03374E+03	− 7.72579E+03
	Average	− 5.45952E+03	− 4.65662E+03	− 7.29553E+03	− 4.77003E+03	− 4.73192E+03	− 3.28899E+03	− 1.08646E+04
	Best	− 1.25120E+04	− 5.48254E+03	− 8.01500E+03	− 5.40216E+03	− 6.08753E+03	− 3.53095E+03	− 1.25681E+04
	STD	4.00809E+03	6.08498E+02	6.47434E+02	5.08751E+02	9.23448E+02	2.00949E+02	2.33870E+03
	Rank	3	6	2	4	5	7	1
F9	Worst	2.59992E−02	0.00000E+00	3.48020E+02	1.49139E+01	3.00793E+02	4.84175E+01	0.00000E+00
	Average	5.19984E−03	0.00000E+00	2.90023E+02	4.68782E+00	2.25716E+02	2.95117E+01	0.00000E+00
	Best	0.00000E+00	0.00000E+00	2.50872E+02	1.70530E−13	1.37319E+02	1.18877E+01	0.00000E+00
	STD	1.16272E−02	0.00000E+00	4.14445E+01	6.10467E+00	7.49862E+01	1.35331E+01	0.00000E+00
	Rank	3	1	7	4	6	5	1
F10	Worst	8.88178E−16	8.88178E−16	2.04081E+01	2.56595E−12	1.82859E+01	2.04944E+01	8.88178E−16
	Average	8.88178E−16	8.88178E−16	2.00413E+01	1.89804E−12	1.54662E+01	9.34103E+00	8.88178E−16
	Best	8.88178E−16	8.88178E−16	1.95756E+01	6.33271E−13	1.09860E+01	2.35216E−01	8.88178E−16
	STD	0.00000E+00	0.00000E+00	3.66843E−01	7.44868E−13	2.76783E+00	1.02359E+01	0.00000E+00
	Rank	1	1	7	4	6	5	1
F11	Worst	0.00000E+00	7.28736E−01	2.02392E+00	0.00000E+00	1.92930E+02	1.09915E+00	0.00000E+00
	Average	0.00000E+00	4.58908E−01	1.49052E+00	0.00000E+00	7.98705E+01	7.55474E−01	0.00000E+00
	Best	0.00000E+00	8.26456E−02	1.24947E+00	0.00000E+00	5.51963E+00	7.59308E−02	0.00000E+00
	STD	0.00000E+00	2.63919E−01	3.06394E−01	0.00000E+00	7.75432E+01	4.02238E−01	0.00000E+00
	Rank	1	4	6	1	7	5	1
F12	Worst	1.43100E−04	7.19391E−01	6.05691E+05	1.71070E−01	1.37493E+06	6.96452E+04	5.47827E−05
	Average	5.17039E−05	6.72788E−01	3.22812E+05	1.41761E−01	3.26428E+05	1.46768E+04	1.80353E−05
	Best	1.25089E−08	6.26844E−01	1.19437E+04	8.63320E−02	2.31749E+01	8.66776E+00	1.44483E−06
	STD	5.40368E−05	4.39489E−02	2.57181E+05	3.37191E−02	5.94440E+05	3.07445E+04	2.11826E−05
	Rank	2	4	6	3	7	5	1
F13	Worst	1.08360E−01	9.94851E−01	2.31891E+02	4.07360E−01	4.61383E+01	5.68316E−01	1.34463E−03
	Average	2.16723E−02	8.96780E−01	6.83334E+01	2.54107E−01	1.29551E+01	3.74361E−01	5.41963E−04
	Best	2.10259E−07	7.61457E−01	6.55719E+00	9.92727E−02	9.17687E−01	1.83653E−01	7.63867E−05
	STD	4.84598E−02	9.22755E−02	9.65830E+01	1.16281E−01	1.90958E+01	1.38524E−01	4.97214E−04
	Rank	2	5	7	3	6	4	1
Summation		28	42	80	43	81	68	13
Mean ranking		2.15	3.50	6.15	3.31	6.23	5.23	1.00
Final ranking		2	3	6	4	7	5	1

Table 3 The results of the comparative methods using ten benchmark functions (F14–F23)

Function	Measure	AO	AOA	GOA	GWO	PSO	WOA	IAO
F14	Worst	1.26705E+01	1.26705E+01	2.10727E+01	1.26705E+01	1.92307E+01	1.07632E+01	1.26705E+01
	Average	6.67091E+00	1.20987E+01	1.55819E+01	7.83894E+00	1.04653E+01	7.65075E+00	4.32495E+00
	Best	9.98004E−01	1.07632E+01	1.99203E+00	2.98211E+00	3.96825E+00	2.98211E+00	9.98004E−01
	STD	5.44697E+00	8.52775E−01	8.21182E+00	4.64043E+00	8.04156E+00	4.26187E+00	4.73724E+00
	Rank	2	6	7	4	5	3	1
F15	Worst	1.30249E−03	1.33126E−02	1.09585E−01	2.03634E−02	6.33375E−02	1.52047E−03	1.08848E−03
	Average	8.87061E−04	4.63828E−03	4.27394E−02	4.36820E−03	1.96312E−02	9.45142E−04	7.59810E−04
	Best	3.11944E−04	4.13806E−04	8.14421E−03	3.08110E−04	5.65059E−04	7.05722E−04	4.76686E−04
	STD	3.71974E−04	5.96477E−03	4.02337E−02	8.94177E−03	2.57494E−02	3.27453E−04	2.57510E−04
	Rank	2	5	7	4	6	3	1
F16	Worst	− 1.02844E+00	− 1.03163E+00	2.10425E+00	− 1.03163E+00	− 1.02991E+00	− 1.03119E+00	− 1.03163E+00
	Average	− 1.03007E+00	− 1.03163E+00	8.52461E−02	− 1.03163E+00	− 1.03124E+00	− 1.03142E+00	− 1.03163E+00
	Best	− 1.03145E+00	− 1.03163E+00	− 1.03163E+00	− 1.03163E+00	− 1.03163E+00	− 1.03162E+00	− 1.03163E+00
	STD	1.47688E−03	1.98318E−07	1.18269E+00	5.05505E−08	7.44855E−04	1.90944E−04	1.11022E−16
	Rank	6		7	2	5	4	1
F17	Worst	0.404803992	0.438568507	0.438568507	0.397898659	0.398733974	0.422821163	0.397887358
	Average	0.399789818	0.409162589	0.409162589	0.397890805	0.398089702	0.406233464	0.397887358
	Best	0.398177978	0.398910351	0.398910351	0.397887596	0.397887382	0.39955111	0.397887358
	STD	0.002817477	0.01655499	0.01655499	4.52388E−06	0.000365006	0.010054555	1.58887E−15
	Rank	4		6	2	3	5	1
F18	Worst	3.12923E+00	1.12019E+02	8.40000E+02	3.00072E+00	3.00071E+00	3.00095E+00	3.00000E+00
	Average	3.07671E+00	5.34484E+01	3.64800E+02	3.00032E+00	3.00046E+00	3.00032E+00	3.00000E+00
	Best	3.05546E+00	3.00000E+00	3.00000E+01	3.00000E+00	3.00021E+00	3.00006E+00	3.00000E+00
	STD	2.98160E−02	4.63105E+01	4.34356E+02	3.53737E−04	1.93184E−04	3.76525E−04	8.53973E−12
	Rank	5		7	3	4	2	1
F19	Worst	− 3.81834E+00	− 3.83850E+00	− 1.93622E+00	− 1.00082E+00	− 3.65549E+00	− 3.08976E+00	− 3.84067E+00
	Average	− 3.84467E+00	− 3.84662E+00	− 2.78208E+00	− 3.28738E+00	− 3.81357E+00	− 3.70818E+00	− 3.84993E+00
	Best	− 3.85501E+00	− 3.85070E+00	− 3.38855E+00	− 3.86277E+00	− 3.86143E+00	− 3.86278E+00	− 3.85448E+00
	STD	1.54193E−02	4.99330E−03	5.64508E−01	1.27823E+00	8.88683E−02	3.45704E−01	5.76333E−03
	Rank	3	2	7	6	4	5	1

Table 3 (continued)

Function	Measure	AO	AOA	GOA	GWO	PSO	WOA	IAO
F20	Worst	− 2.81984E+00	− 2.23002E+00	− 1.49791E+00	− 3.13764E+00	− 1.61543E+00	− 1.15134E+00	− 3.20029E+00
	Average	− 2.99441E+00	− 2.84867E+00	− 2.20908E+00	− 3.23642E+00	− 2.84091E+00	− 2.25639E+00	− 3.29760E+00
	Best	− 3.13249E+00	− 3.12202E+00	− 2.94811E+00	− 3.32200E+00	− 3.31665E+00	− 3.00457E+00	− 3.32198E+00
	STD	1.24736E−01	3.62606E−01	6.14225E−01	8.22152E−02	7.11709E−01	1.00808E+00	5.44000E−02
	Rank	3	4	7	2	5	6	1
F21	Worst	− 5.05520E+00	− 1.66896E+00	− 6.86756E−01	− 9.81135E+00	− 2.46095E+00	− 3.51363E−01	− 1.01474E+01
	Average	− 8.11399E+00	− 2.20663E+00	− 5.50213E+00	− 1.00241E+01	− 4.99751E+00	− 1.40889E+00	− 1.01509E+01
	Best	− 1.01532E+01	− 2.73352E+00	− 1.01532E+01	− 1.01494E+01	− 9.90558E+00	− 4.58734E+00	− 1.01527E+01
	STD	2.79228E+00	4.75846E−01	4.56208E+00	1.45680E−01	2.99717E+00	1.79615E+00	2.26902E−03
	Rank	3	6	4	2	5	7	1
F22	Worst	− 5.08767E+00	− 2.05382E+00	− 7.43142E−01	− 1.03982E+01	− 2.73464E+00	− 5.23954E−01	− 9.31367E+00
	Average	− 8.28506E+00	− 4.09678E+00	− 5.07989E+00	− 1.04000E+01	− 3.59701E+00	− 1.52140E+00	− 1.01677E+01
	Best	− 1.04029E+01	− 7.35088E+00	− 1.04029E+01	− 1.04015E+01	− 5.08767E+00	− 4.16565E+00	− 1.03895E+01
	STD	2.90006E+00	1.98310E+00	4.91921E+00	1.39777E−03	9.59214E−01	1.53051E+00	4.77518E−01
	Rank	3	5	4	1	6	7	2
F23	Worst	− 1.03626E+01	− 1.57856E+00	− 9.69882E−01	− 9.39793E−01	− 1.85944E+00	− 5.58474E−01	− 1.05349E+01
	Average	− 1.04911E+01	− 2.44959E+00	− 2.28743E+00	− 7.53489E+00	− 3.83833E+00	− 2.10985E+00	− 1.05352E+01
	Best	− 1.05351E+01	− 3.71332E+00	− 3.82643E+00	− 1.05364E+01	− 1.00982E+01	− 3.19356E+00	− 1.05358E+01
	STD	7.31055E−02	9.39359E−01	1.12092E+00	4.36735E+00	3.50752E+00	1.25883E+00	3.71949E−04
	Rank	2	5	6	3	4	7	1
Summation		33	33	62	29	47	49	11
Mean ranking		3.30	4.71	6.20	2.90	4.70	4.90	1.10
Final ranking		3	3	7	2	5	6	1

WOA, GOA, AO, GWO, HGSO, IAO (improved Aquila Optimizer, AO–AOA).

To evaluate the performance, each method is tested on the same parameters like population size and no. of iteration. The proposed AO–AOA is evaluated by varying the dimensions. The impact of varying dimensions is a standard test used in previous studies for optimizing test functions that show the effect of varying dimensions on AO–AOA efficiency. From this, it is noted that it works efficiently for both high- and low-dimensional problems. In high-dimensional problem population, dependent methods give efficient search results. In this work, AO–AOA is used to test scalable multi, and unimodal test functions (F1–F13) with two different dimensions (10 and 100) and results of comparative methods are tested using ten benchmark functions (F14–F23). The AO–AOA is dealing with 13 functions (F1–F13) with two different dimensions is compared using standard deviation (SD) and average fitness value (Avg).

AO–AOA gives the most optimum results in 10, which shows that any optimization methods give optimum results in low dimensions. Now. AO–AOA is testing using high dimensions (100), which also gave the optimum results. To verify, the proposed AO–AOA is compared with previously state-of-art algorithms using the same dimensions (10 and 100) and test function. When the results were analyzed, it showed that AO–AOA is the most optimum in different cases.

It is compared with other methods to evaluate the convergence behavior of AO–AOA, as shown in Fig. 4. The curves of convergence with test functions (F1–F13) can be seen in Fig. 4. After observing Fig. 4, it becomes clear that AO–AOA gives low and stable convergence compared with other methods. AO–AOA has more optimum global search capability and fast convergence and attained optimum results than other methods on the same test functions concerning convergence speed and global search capability.

The average runtime of the proposed AO–AOA algorithm is compared with other pre-existing methods with 10 dimensions is shown in Table 1, and with 100 dimensions is shown in Table 2. As AO–AOA depends on the population method, there is no need for optimization so, the running time needed by AO–AOA is less in terms of seconds when compared with other methods. So, the computational efficiency of the proposed AO–AOA is much optimum than other methods. Observing the results shows that the AO–AOA is on top, followed by other methods.

If we notice the values in Table 3, the AO–AOA is very competitive and superior compared with others on test functions (F14–F23). No. of optimization methods obtained optimum results, but AO–AOA has the best when compared with all. So, AO–AOA is capable of obtaining optimum results.

6 Conclusion

This paper proposes a hybrid approach (AO–AOA) based on population-based models to solve optimization problems. AO and AOA are both modern meta-heuristic optimization methods. They can be applied to different problems, including image processing, machine learning, wireless networks, power systems, engineering design etc. The proposed AO–AOA system has been validated on a detailed set of 23 functions (F1–F23). The obtained findings were compared to other cutting-edge methods such as AOA, PSO, WOA, GOA, AO, GWO, HGSO. Observing the experimental results, it became apparent that AO–AOA has faster convergence with optimal global search capability and generates better results.

Furthermore, AO and AOA can be combined with other existing cutting-edge algorithms, improving the algorithm and providing more accurate results with less computational time, which is needed in real-time applications and problems.

Acknowledgements This study was financially supported via a funding grant by Deanship of Scientific Research, Taif University Researchers Supporting Project number (TURSP-2020/300), Taif University, Taif, Saudi Arabia.

Funding The authors received no specific funding for this study. Taif University, TURSP-2020/300, Maryam Altalhi.

Data and Code Availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest to report regarding the present study.

References

Abed-alguni, BH., Noor AA (2021) Distributed grey wolf optimizer for scheduling of workflow applications in cloud environments. Appl Soft Comput 102(2021):107113

Abed-alguni BH et al (2021) Exploratory cuckoo search for solving single-objective optimization problems. Soft Comput (2021):1–14

Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2020) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609

Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Indus Eng 107250

Alawad NA, Abed-alguni BH (2021) Discrete Jaya with refraction learning and three mutation methods for the permutation flow shop scheduling problem. J Supercomput 1–22

Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 10(237):82–117
Carnie SK (1954) Food habits of nesting golden eagles in the coast ranges of California. Condor 56(1):3–12
Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 191(11–12):1245–1287
Colorni A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life, vol 142, pp 134–142
Cuevas E, Echavarría A, Zaldívar D, Pérez-Cisneros M (2013) A novel evolutionary algorithm inspired by the states of matter for template matching. Exp Syst Appl 40(16):6359–6373
Cuevas E, Echavarría A, Ramírez-Ortegón MA (2014) An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. Appl Intell 40(2):256–272
Davis L (1991) Bit-climbing, representational bias, and test suit design. InProc Intl Conf Genetic Algo 1991:18–23
Dekker D (1985) Hunting behavior of golden eagles, aquila-chrysaetos, migrating in southwestern alberta. Can Field-Nat 99(3):383–385
Eberhat R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Sixth international symposium on micro machine and human science. Piscataway, pp 39–43
Fogel LJ, Owens AJ, Walsh MJ (1966) Artificial intelligence through simulated evolution
Glover F (1989) Tabu search—part I. ORSA J Comput 1(3):190–206
Gogna A, Tayal A (2013) Metaheuristics: review and application. J Exp Theor Artif Intell 25(4):503–526
Hatch DR (1968) Golden eagle hunting tactics. Blue Jay 26(2)
Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
Kaveh A, Farhoudi N (2013) A new optimization method: Dolphin echolocation. Adv Eng Softw 1(59):53–70
Kaveh A, Farhoudi N (2016) Dolphin monitoring for enhancing metaheuristic algorithms: Layout optimization of braced frames. Comput Struct 1(165):1–9
Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
Li J, Lin J (2020) A probability distribution detection based hybrid ensemble QoS prediction approach. Inf Sci 1(519):289–305
Li J, Zheng XL, Chen ST, Song WW, Chen DR (2014) An efficient and reliable approach for quality-of-service-aware service composition. Inf Sci 10(269):238–254
Li J, Chen C, Chen H, Tong C (2017) Towards context-aware social recommendation via individual trust. Knowl Based Syst 1(127):58–66
Liu E, Lv L, Yi Y, Xie P (2019) Research on the steady operation optimization model of natural gas pipeline considering the combined operation of air coolers and compressors. IEEE Access 24(7):83251–83265
Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. In: Handbook of metaheuristics. Springer, Boston, MA, pp 320–353
Mahajan S, Mittal N, Pandit AK (2021) Image segmentation using multilevel thresholding based on type II fuzzy entropy and marine predators algorithm. Multim Tools Appl 26:1–25
Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidisc Optim 26(6):369–395
Meinertzhagen R (1940) How do larger raptorial birds hunt their prey. Ibis 4:530–535
Michalewicz Z (2013) Genetic algorithms+ data structures= evolution programs. Springer Science & Business Media
Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evol Comput 1(9):1–4
Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 1(69):46–61
Pang J, Zhou H, Tsai YC, Chou FD (2018) A scatter simulated annealing algorithm for the bi-objective scheduling problem for the wet station of semiconductor manufacturing. Comput Ind Eng 1(123):54–66
Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 1(105):30–47
Spall JC (2005) Introduction to stochastic search and optimization: estimation, simulation, and control. John Wiley & Sons
Steenhof K, Kochert MN, Mcdonald TL (1997) Interactive effects of prey and weather on golden eagle reproduction. J Anim Ecol 1:350–362
Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
Watson J (2010) The golden eagle. Bloomsbury Publishing
Wen F, Yang X, Gong X, Lai KK (2017) Multi-scale volatility feature analysis and prediction of gold price. Int J Inf Technol Decis Mak 16(01):205–223
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
Yan J, Pu W, Zhou S, Liu H, Bao Z (2020) Collaborative detection and power allocation framework for target tracking in multiple radar system. Inform Fusion 1(55):173–183
Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2(2):78–84
Yang XS, Deb S (2010) Engineering optimisation by cuckoo search. Int J Math Modell Numer Optim 1(4):330–343
Yang XS, Bramer M, Ellis R, Petridis M (2010) Research and development in intelligent systems XXVI. Springer, Development
Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver press
Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, Heidelberg, pp 65–74
Yang XS (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. Springer, Berlin, Heidelberg, pp 240–249
Yang Y, Chen H, Heidari AA, Gandomi AH (2021) Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. Exp Syst Appl 10:114864
Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1(1):32–49