Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Imperialist competition algorithm with quasi-opposition-based learning for function optimization and engineering design problems

Dongge Lei, Lulu Cai & Fei Wu

To cite this article: Dongge Lei, Lulu Cai & Fei Wu (2024) Imperialist competition algorithm with quasi-opposition-based learning for function optimization and engineering design problems, Automatika, 65:4, 1640-1665, DOI: 10.1080/00051144.2024.2420296

To link to this article: https://doi.org/10.1080/00051144.2024.2420296

AUTOMATIKA
2024, VOL. 65, NO. 4, 1640–1665
https://doi.org/10.1080/00051144.2024.2420296

Imperialist competition algorithm with quasi-opposition-based learning for function optimization and engineering design problems

Dongge Lei, Lulu Cai and Fei Wu

College of Electrical and Information Engineering, Quzhou University, Quzhou, People's Republic of China

ABSTRACT
Imperialist competitive algorithm (ICA) is an efficient meta-heuristic algorithm by simulating the competitive behaviour among imperialist countries. However, it still suffers from slow convergence and deficiency in exploration. To address these issues, an improved ICA is proposed by combining ICA with a quasi-opposition-based learning (QOBL) strategy, which is named QOBL-ICA. The improvements include two aspects. First, the QOBL strategy is adopted to generate a population of fitter individuals. Second, a QOBL-assisted assimilation strategy is proposed to enhance the exploration ability of ICA. As a result, the proposed QOBL-ICA has more powerful exploration ability than ICA as well as faster convergence speed. The effectiveness of the proposed QOBL-ICA is verified by testing on 20 benchmark functions and 3 engineering design problems. Experimental results show that the performance of QOBL-ICA is superior to most state-of-the-art meta-heuristic algorithms in terms of global optimum reached and convergence speed.

ARTICLE HISTORY
Received 29 May 2024
Accepted 7 October 2024

KEYWORDS
Imperialist competition algorithm; quasi-opposition-based learning; function optimization; engineering design problem; Wilcoxon test

1. Introduction

1.1. Background

Many problems in engineering practice can finally be formulated as optimization problems. To obtain the ideal solution to these problems, high-performance optimal algorithms are very in demand. Compared to classical derivation-based optimization algorithms, meta-heuristic algorithms are intrinsically global searching, have less computational burden and fewer requirements on objective functions. Due to these advantages, meta-heuristic algorithms have received considerable attention from researchers, and a great number of meta-heuristic algorithms have been developed. For example, prairie dog optimization (PDO) [1], crayfish optimization algorithm (COA) [2,3], hunger games search optimization algorithm (HGSOA) [4], mountain gazelle optimizer (MGO) [5], ship rescue optimization (SRO) [6], artificial rabbits algorithm (ARA) [7], arithmetic optimization algorithm (AOA) [8], marine predators algorithm (MPA) [9], cheetah optimization algorithm (COA) [10], to name but a few.

In addition, meta-heuristic algorithms had also been applied to solve various engineering problems. For example, the flight control tuning [11], feature selection [12], structure performance enhancement of engineering components [13], reliability analysis [14], design optimization [15], conceptual design of fixed wing unmanned aerial vehicle [10], PID controller tuning [16], training of artificial neural networks [17], data transmission optimization [18], search engine [19], etc.

In [20], motivated by the imperialistic competitive phenomenon between countries, Atashpaz-Gargari and Lucas proposed another meta-heuristic algorithm named the ICA. In ICA, each country is an individual. They are divided into imperialists and colonies. Those powerful countries are classified as imperialists while those weaker countries are colonies. Each imperialist along with a certain number of colonies makes up an empire. In the first evolutionary stage, the colonies in an empire are assimilated by the imperialists, and as a result, their power is strengthened. In the second phase, the imperialistic competition between empires occurs. In this phase, the stronger empires try to occupy the colonies of weaker empires. The direct consequence is that the weaker empire becomes weaker and weaker due to gradual losing its colonies, while the stronger empire becomes stronger and stronger. From its intrinsical evolutionary mechanism, ICA can be regarded as a multi-swarm based evolution algorithm. Due to this characteristic, ICA exhibits competitive performance in solving different engineering problems, for example, job shop scheduling problems [21], multi-layer perceptron training [22], node placement problems in wireless sensor networks [23], fuzzy controller coefficient optimization [24], parameter tuning of controllers for automatic generation control (AGC) systems [25,26],

CONTACT Fei Wu wufei@qzc.edu.cn College of Electrical and Information Engineering, Quzhou University, Quzhou, Zhejiang 324000, People's Republic of China

data clustering [27] and so on. In addition, ICA is also combined with other soft computing techniques to solve different engineering problems. For instance, in [28], ICA is combined with the XGBoost model to perform the prediction of the compressive strength of recycled aggregate concrete. In [29], ICA is combined with the adaptive neuro-fuzzy inference system polynomial neural network (ANFIS-PNN) model to predict tunnel boring machine(TBM) performance and in [21], ICA is combined with an artificial neural network to model the deflection of reinforced concrete beams.

1.2. Literature review

After ICA had been proposed, researchers proposed various improved schemes to further enhance the performance of ICA. In the earlier time, researchers tried to design various coefficients changing schemes for ICA. In [30], to make a better trade-off between diversity and convergence, a new assimilation coefficient adjusting scheme was proposed according to the attraction and repulsion principle. A new assimilation coefficient adjusting scheme was proposed based on an interval type-2 fuzzy system (T2FS) in [31]. The input of T2FS is the current iteration number and the output is the assimilation coefficient. Except for the assimilation coefficient, several authors also proposed to adaptively adjust the deviation angle. In [32], the colonies' deviation angle dynamically shrinks or expands according to the countries' probability, which is modelled by a Gaussian model. In [33], the authors proposed using a fuzzy inference system (FIS) to infer an appropriate deviation angle.

As well as changing ICA's coefficients, researchers also have made a lot of efforts to change the assimilation scheme. In [34,35], the authors borrowed the idea from particle swarm optimization (PSO) and designed an assimilation scheme for colonies. In the designed assimilation scheme, the colonies approach the imperialist using the position update formula in PSO. In [36], the moving formula in the firefly algorithm was utilized by ICA as the assimilation strategy. In [37], the Gaussian sampling-based bare-bone assimilation strategy was proposed for ICA.

Except for the above-mentioned ICA variants, people seek to mix ICA with other meta-heuristic algorithms. In [38], the simplex method is hybrid with ICA and severed as a local searcher. In [39], ICA and simulating annealing (SA) are hybridized together. In the hybrid algorithm, when ICA accomplishes its assimilation process, SA starts to perform further searching processes. This idea can also be seen in [40]. In [41], a hybrid of ICA, GA and PSO was proposed. In the hybrid algorithms, GA is responsible for generating high-quality initial countries and PSO is adopted to improve certain imperialist, which is randomly selected from all the imperialist. Another hybrid scheme, such as ICA hybrid with pattern search (PS) [42], and variable neighbourhood search (VNS) [43] has also been proposed.

1.3. Research gap and motivation

The above-mentioned ICA variants indeed outperform the basic ICA to some extent, however, they perform searching or exploration randomly. One distinct shortcoming of random search is that it may visit or revisit unproductive regions of the search space. This ineffective exploration not only leads to a slower convergence rate but also hinders the algorithm to explore more productive regions where more optimal solutions stay. Fortunately, related research results reveal that the position generated by opposition-based learning (OBL) may be closer to the optimal solution than that by a random manner [44]. Due to this appealing fact, the OBL [45] strategy has been adopted by many researchers to improve the optimization ability of meta-heuristic algorithms. In [46], the OBL was incorporated into arithmetic optimization algorithm (AOA) to improve its global search ability. The resulted algorithm is called OBL-AOA. Then the proposed OBL-AOA was applied to obtain appropriate parameters for density-based spatial clustering of applications with noise algorithm (DBSCAN) [47]. In [48], a novel hybrid optimizer was proposed to solve the mechanical engineering optimization problem by combining the flow direction optimization (FDO) with dynamic opposition-based learning (DOBL). In [49], the authors integrated the elite opposition-based learning into the generalized normal distribution algorithm (GNDA) and applied the improved algorithm to the engineering design problem. In [50], the performance of Fick's law optimization algorithm (FLA) was enhanced by incorporating QOBL techniques. In [51], differential evolution with OBL (OBLDE) algorithm was adopted to optimally design of bumper beam and energy absorber design for a passenger car.

Motivated by the above facts, we proposed an improved ICA. The proposed algorithm is called QOBL-ICA. The improvements of the proposed QOBL-ICA consist of two aspects. First, a swarm of N individuals, called countries, are randomly produced in the admitted search range, and then, N quasi-oppositional counterparts are generated. N countries with the best fitness are picked out from the $2N$ countries to make up the final initial population. Second, the quasi-oppositional learning strategy helps ICA explore the best solution in the assimilation step. Specifically, when a colony moves to a new position, a quasi-oppositional position corresponding to the new position is generated. The position with better fitness was adopted as the new position of the colony. The performance of the proposed QOBL-ICA was examined through optimizing benchmark functions and several engineering design problems.

1.4. Paper organization

In the following, the rest part of this paper is arranged. The research significance is presented in Section 2. The basic of ICA is shortly reviewed in Section 3. The proposed QOBL-ICA is illustrated in detail in Section 4. In Section 5, the experimental results and comparative analysis are given. Finally, the conclusive remarks are presented in Section 6.

2. Research significance

As a promising meta-heuristic algorithm, ICA has exhibited great competitive performance in solving various complex optimization problems. Nevertheless, due to its inherent randomness, ICA suffers from certain drawbacks such as singleness of searching way, premature convergence and low ability of local search. Though a multitude of improving schemes mentioned in the Introduction have been proposed for ICA, most of these schemes are still random. Relevant research reveals that the QOBL strategy has a larger chance to approach the optimal solution than random searching for the reason that it hunts for the optimal solution in the opposite direction of the current position. QOBL belongs to a deterministic learning scheme. Observing this fact, we proposed an improved scheme for ICA, which incorporates QOBL into the initial and assimilation stages of ICA. The main contributions or novelties of this paper are summarized as follows.

(1) A new initial population generation scheme for ICA is proposed based on QOBL strategy. Specifically, a group of N countries is first generated in a random way, and then the quasi-opposite countries are produced according to the concept of quasi-opposite number. The fitter ones among the countries and their quasi-opposite countries are selected to comprise the initial population.
(2) A new assimilation strategy that incorporates the QOBL strategy is proposed for ICA. After a colony moves to a new position, the quasi-opposite position is generated. By comparing the fitness of the new position and its quasi-opposite position, the position with better fitness is determined as the final position of the colony. In this way, the colonies can find better candidate solution by exploring informative regions in the searching space.
(3) The incorporation of the QOBL strategy into ICA not only speeds up the convergence of ICA but also enhances the exploration ability, making it effectively avoid falling into local optimum.
(4) The effectiveness of QOBL-ICA is proved by extensive testing on 20 well-known benchmark functions and 3 engineering design problems. The comparison with other state-of-the-art algorithms demonstrates the superiority of QOBL-ICA.

3. A brief review of ICA

Like other evolutionary algorithms (EAs), ICA gradually improves its solution quality via evolutionary operations. In ICA, the main evolutionary operators include assimilation, revolution and imperialistic competition. Before the evolutionary process began, the individuals or countries of ICA were classified into a couple of empires. In each empire, the best country is classified as imperialist and the rest as colonies. In the first evolutionary phase, the colonies moved toward the imperialist to improve their power. This step is called assimilation. After assimilation, the power of the empire will be strengthened. The second evolutionary operation is imperialist competition. The direct consequence of this is that the weakest empire will lose its the weakest colony. The lost colony will be possessed by other stronger empires. In this stage, if a certain weaker empire loses all colonies, the empire collapses. In the end, only the strongest empire survives and the algorithm converges. The whole process will be stated in detail as follows.

3.1. Initialization

Suppose we find the minimum of a d-dimensional optimization problem in a given search range. The d variables are written as a vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$ and each variable has an admitted range, i.e. $l_j \le x_j \le u_j$, $j = 1, 2, \ldots, d$. In ICA, each country represents a candidate solution for an optimization problem to be solved. The objective function value $c = f(\boldsymbol{x})$ is called the country's cost or power. Before the evolution operation starts, an population with N countries is initially produced. Each country is generated according to the following formula:

$$x_{ij} = l_j + (u_j - l_j) \times r, \quad j = 1, 2, \ldots, d; i = 1, 2, \ldots, N, \tag{1}$$

where x_{ij} is the jth component of the ith individual $\boldsymbol{x}_i$, r is a random number lies in $[0, 1]$. After generating N initial countries, one calculates its cost or power as $c_i = f(\boldsymbol{x}_i)$. Then the N countries are sorted in ascending order. N_{imp} countries with the best cost are classified as imperialist, the other $N_{col} = N - N_{imp}$ are colones. Each imperialist possesses one or more colonies. The number of colonies that the nth imperialist can possess is related to its normalized power C_n. C_n is calculated as

$$C_n = c_n - \max_i\{c_i\}, \quad i = 1, 2, \ldots, N_{imp}, \tag{2}$$

where c_n and C_n are the power and normalized power of the nth imperialist. The larger the normalized power

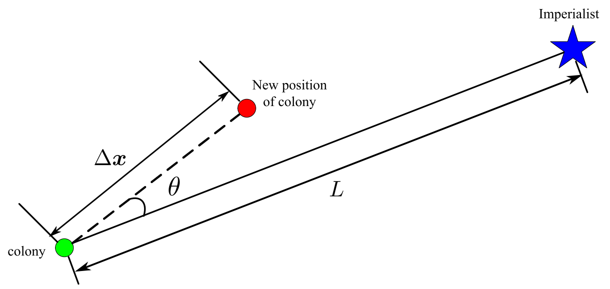

Figure 1. Assimilation process.

is the more colonies it can possess. The number of colonies that the nth imperialist can possess is computed as

$$NC_n = round\{p_n \cdot N_{col}\}, \quad n = 1, 2, \ldots, N_{imp}, \tag{3}$$

where

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right|. \tag{4}$$

After assigning colonies to each imperialist, N_{imp} empires are initially formed colonies.

3.2. Assimilation

After initially generating N_{imp} empires, assimilation begins. In assimilation, each colony moved toward the imperialist to improve its power and thus strengthen the whole power of the empire. Figure 1 illustrates the assimilation process, where a colony moves a distance $\Delta \boldsymbol{x}_i$ from the current position, each component of $\Delta \boldsymbol{x}_i$ is a random number that lies in $[0, \beta \times L]$, that is

$$\Delta x_{ij} \sim U(0, \beta \times L), \tag{5}$$

where the assimilation coefficient β is a positive real number greater than 1 called the assimilation coefficient and L is the distance between the colony and the imperialist. To find a better solution, as well as explore along the direction pointing to the imperialist, the colony also performs searching around the current position with a deviation angle θ. The value of θ is randomly selected from a given interval $[-\gamma, \gamma]$. Mathematically,

$$\theta \sim U(-\gamma, \gamma). \tag{6}$$

In [20], the authors suggest γ to be $\pi/4$.

After assimilating, the colony locates in a new position. At this moment, if the colony's cost is superior to that of the imperialist, then the colony will become imperialist.

3.3. Imperialistic competition

The imperialistic competition process begins after assimilation. The core of imperialistic competition is that the stronger empires become stronger by annexing the colonies of weaker empires. To this end, the weakest colony in the weakest empire is picked up and reassigned to other stronger empires. If an empire is more powerful, then it is more likely to occupy the colony. The power of the nth empire is computed as

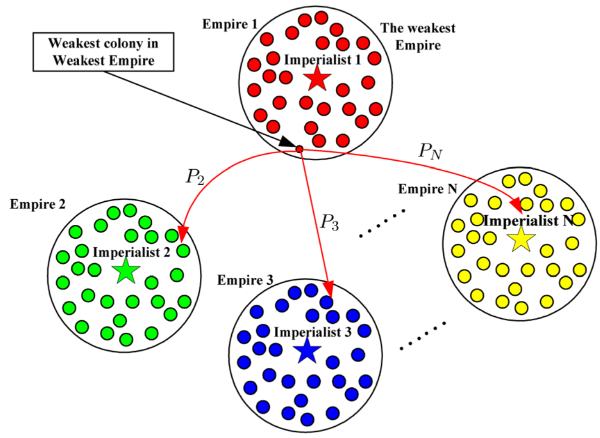

Figure 2. Imperialist competition of ICA.

$$TC_n = c_n + \xi \frac{\sum_{i=1}^{NC_n} c_i}{NC_n}, \tag{7}$$

and the normalized cost is computed as follows:

$$NTC_n = TC_n - \max_i\{TC_i\}, \tag{8}$$

where ξ is a positive number less than 1, called the cost ratio coefficient, c_n is the cost of the nth empire and c_i the ith colony in the empire. In the population, each empire is possible to occupy the lost colony, which is determined by a probability p_n as

$$p_n = \left| \frac{NTC_n}{\sum_{j=1}^{N_{imp}} NTC_j} \right|, \quad n = 1, 2, \ldots, N_{imp}, \tag{9}$$

Denotes

$$\boldsymbol{p} = [p_1, p_2, \ldots, p_{N_{imp}}] \tag{10}$$

be the probability vector. Another random vector $\boldsymbol{r}$ is generated as

$$\boldsymbol{r} = [r_1, r_2, \ldots, r_{N_{imp}}]. \tag{11}$$

Each component r_i of $\boldsymbol{r}$ is a random number in $[0, 1]$. The difference between $\boldsymbol{p}$ and $\boldsymbol{r}$ is

$$\boldsymbol{m} = [p_1 - r_1, p_2 - r_2, \ldots, p_{N_{imp}} - r_{N_{imp}}]. \tag{12}$$

The index corresponding to the maximum value of $\boldsymbol{m}$ is denoted as ν, then the νth empire will occupy the lost colony. Figure 2 gives an explanation of imperialistic competition.

3.4. Elimination and convergence

As the assimilation and imperialistic competition go on, the colonies in the weaker empire will gradually be seized by another empire. As a result, all its colonies will be occupied and thus, the empire will become empty. In this station, the empire will be eliminated from the empire list. When all but one empire exits, the evolution process terminates and the algorithm converges.

4. The proposed QOBL-ICA

Here, the proposed QOBL-ICA is fully illustrated. For the sake of convenience, several concepts related to opposition-based learning and quasi-learning are explained. Then the main procedure of QOBL-ICA is given.

4.1. QOBL

In general, most evolutionary algorithms start to find the optimal solution from an initial group of individuals. The initial individuals are randomly generated in the admitted search space. The individuals try to improve themselves via various evolutionary operators. These operators also work randomly. The initial or current population can be regarded as a guess of the optimal solution. It is proven that, in general, the opposite guess has a larger chance to approach the optimal solution than the random guess. Observing this fact, Tizhoosh proposed the OBL method in [45]. Since OBL has great potential for discovering more optimal solutions, it has attracted increasing interest of researchers to improve the performance of different learning tasks such as reinforcement learning, and back-propagation learning in neural networks, speeding up the convergence rate of different evolutionary algorithms [44,52,53]. In the following, several concepts related to OBL are introduced.

4.1.1. Opposite number

Given a real number x in $[a, b]$, its opposite number is defined as

$$\breve{x} = a + b - x. \tag{13}$$

4.1.2. Opposite point

For D-dimensional Euclid space, a point is denoted as $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$. It's opposite point is denoted as $\breve{\boldsymbol{x}} = [\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d]$, where

$$\breve{x}_i = a_i + b_i - x_i. \tag{14}$$

In Ref. [54], Tizhoosh further proposed the concept of quasi-oppositional based learning (QOBL). Similarly, QOBL is also based on the concept of quasi-oppositional numbers and points. Their definitions are given as follows.

4.1.3. Quasi-opposite number

Let x be a number in $[a, b]$, it's quasi-oppositional number is defined as

$$\bar{x} = \begin{cases} \dfrac{a+b}{2} + r\left(\breve{x} - \dfrac{a+b}{2}\right), & \breve{x} < \dfrac{a+b}{2}; \\ \breve{x} + \left(\dfrac{a+b}{2} - \breve{x}\right), & \breve{x} \geq \dfrac{a+b}{2}. \end{cases} \tag{15}$$

4.1.4. Quasi-opposite point

Let $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$ be a point in R^d Euclid space. It's quasi-oppositional point $\bar{\boldsymbol{x}} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_d]$ is defined as

$$\bar{x}_i = \begin{cases} \dfrac{a+b}{2} + r\left(\breve{x}_i - \dfrac{a+b}{2}\right), & \breve{x}_i < \dfrac{a+b}{2}; \\ \breve{x}_i + \left(\dfrac{a+b}{2} - \breve{x}_i\right), & \breve{x}_i \geq \dfrac{a+b}{2}. \end{cases} \tag{16}$$

4.1.5. Opposition or quasi-opposition-based optimization

Considering a minimization problem with objective function $f(\boldsymbol{x})$. Let the current point is $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$, its oppositional and quasi-oppositional point are denoted as $\breve{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}$, respectively. If $f(\breve{\boldsymbol{x}}) < f(\boldsymbol{x})$ or $f(\bar{\boldsymbol{x}}) < f(\boldsymbol{x})$, then replace $\boldsymbol{x}$ with $\breve{\boldsymbol{x}}$ or $\bar{\boldsymbol{x}}$; otherwise, $\boldsymbol{x}$ is reserved.

4.2. QOBL-ICA

QOBL-ICA also includes initialization, assimilation, and imperialist competition three steps. Different from ICA, QOBL strategy is embedded into initialization and assimilation to increase the quality of initial population and enhance the exploration ability.

4.2.1. The QOBL-based population initialization

The initialization of QOBL-ICA includes the following steps.

(a) Randomly generates an initial population $\boldsymbol{X}_0$
(b) Generates a quasi-oppositional population $\boldsymbol{QX}_0$ from $\boldsymbol{X}_0$
(c) Calculates the cost of each individual in $\boldsymbol{X}_0$ and $\boldsymbol{QX}_0$, which are denoted as $\boldsymbol{F}_0$ and $\boldsymbol{QF}_0$
(d) Sort the cost of in $\boldsymbol{F}_0 \bigcup \boldsymbol{QF}_0$ in ascending order. Correspondingly, the individuals in $\boldsymbol{X}_0 \bigcup \boldsymbol{QX}_0$ are also sorted according to the order as in $\boldsymbol{F}_0 \bigcup \boldsymbol{QF}_0$.
(e) Select the first N individuals from $\boldsymbol{X}_0 \bigcup \boldsymbol{QX}_0$ as the initial population of QOBL-ICA. At the same time, the first N costs from $\boldsymbol{F}_0 \bigcup \boldsymbol{QF}_0$ are the cost of the initial population.

4.2.2. QOBL-based assimilation

In the assimilation stage, QOBL is also used to help the algorithm find better potential solutions. Let $\boldsymbol{x}_i$ denote the current position of the ith colony. The colony

Algorithm 1 QOBL-ICA algorithm

```
Define the object function f(x) and determine related parameters of the objective function.
Determine the algorithm's parameter, such as the maximum number of iterations T_max, the number of empires
  N_imp.
Generate an initial population according to QOBL-based population initialization method.
Sorting the initial countries according to their power.
Generate initial empires according to method described in Section 3.1.
for t = 1 : T_max do
    for n = 1 : N_imp do
        QOBL based Assimilation: change the position of colonies in each empire according to the method
        described in Section 4.2.1.
        Comparing the cost of each colony with that of the imperialist, if the cost of a certain colony is lower than
        that of the imperialist, then the colony is selected as the new imperialist.
        Imperialist competition. Calculating the total power and normalized power of the nth empire according
        to formulas (7) and (8).
        Assigning the weakest colonies in the weakest empire to the empire that wins the competition according
        to (9)–(12).
    end for
    Recording the country's position with the smallest cost.
end for
Return the final country's position with the smallest cost.
```

updates its position according to the way described in Equation (5). The new position is denoted as $\boldsymbol{x}_i'$. Then the quasi-oppositional point of $\boldsymbol{x}_i'$ is generated and is represented by $\boldsymbol{qx}_i'$. If $f(\boldsymbol{qx}_i') < f(\boldsymbol{x}_i')$, then $\boldsymbol{qx}_i'$ is adopted as the final position of the colony, otherwise, $\boldsymbol{x}_i'$ is used as the final position.

For the sake of completeness, the flowchart of proposed QOBL-ICA is given in Algorithm 1.

5. Experiments and discussion

To examine its effectiveness QOBL-ICA, a set of benchmark functions and three engineering design problems are solved using QOBL-ICA. Several ICA variants including Gaussian Bare-bone imperialist competition algorithm (GBB-ICA), AR-ICA, ICA-PSO, opposition-based learning imperialist competition algorithm (OBL-ICA), and four well-developed meta-heuristic algorithms including dragonfly algorithm (DA) [55], gravitational search algorithm (GSA) [56], performance guided JAYA (PGJAYA) [57], marine predators algorithm (MPA) [58] are selected as the compared algorithms. For those ICA variants, their parameters are set to the same values for fair comparison. Specifically, the population size, i.e. the number of countries is set to 20. The initial number of imperialists is set to 8, assimilation coefficient $\beta = 2$, cost ratio coefficient $\xi = 0.02$. For ICA-PSO, $c_1 = c_2 = 2, w = rand$, and AR-ICA, $d_{div} = 0.8, \beta_{div} = 3$. The parameters of DA, GAS, PGJAYA and MPA are set to the suggested values as the cited reference. All the referenced algorithms are independently executed 30 times to eliminate the effect of randomness.

5.1. Effect of QOBL based initialization

In the proposed QOBL-ICA, QOBL is adopted to generate a better initial population nearing the optimal solution. In such a way, the convergence speed of the algorithm can be accelerated. To justify this, an ICA variant that combines chaos initialization and QOBL-based assimilation is implemented, which is named Chaos-QOBL-ICA. The QOBL-ICA and Chaos-QOBL-ICA are tested using the Goldstein–Price function, which is defined as

$$
\begin{aligned}
f(x) = & \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2\right. \\
& \left. -14x_2 + 6x_1x_2 + 3x_2^2\right] \\
& \times \left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2\right. \\
& \left. + 48x_2 - 36x_1x_2 + 27x_2^2\right].
\end{aligned}
$$

The global minimum of Goldstein–Price function locates at $(0, -2)$, which corresponds to the optimal function value 3. QOBL-ICA and Chaos-QOBL-ICA independently run 30 times. The convergence speed as well as the optimal solution obtained by QOBL-ICA and Chaos-QOBL-ICA are compared.

Table 1 lists the statistical results of the optimal solution in terms of the best, the worst, the mean and std. It can be seen that QOBL-ICA and Chaos-QOBL-ICA all can find the global solution of the Goldstein–Price function during the 30 runs. This phenomenon explains that QOBL-ICA and Chaos-QOBL-ICA exhibit superior performance. In addition, the std obtained by QOBL-ICA is smaller than that obtained by Chaos-QOBL-ICA, which show that QOBL-ICA is more stable than Chaos-QOBL-ICA.

Table 1. The optimization results of Goldstein–Price function obtained by QOBL-ICA and Chaos-QOBL-ICA.

	QOBL-ICA	Chaos-QOBL-ICA
Best	3	3
Worst	3	3
Mean	3	3
Std	4.53559E-15	3.62754E-15

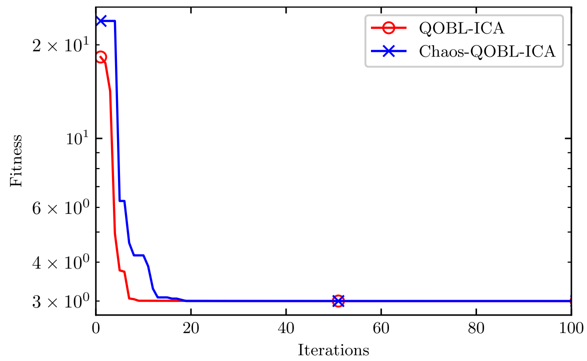

Figure 3. The convergence curve of QOBL-ICA and Chaos-QOBL-ICA on Goldstein–Price function (The first 100 iterations).

Table 2. The fitness value of the first 10 iteration.

Iterations	QOBL-ICA	Chaos-QOBL-ICA
1	18.27278844	23.86540146
2	17.46847934	23.86540146
3	14.20523906	23.86540146
4	4.947204244	23.86540146
5	3.764411689	6.29339613
6	3.727923439	6.29339613
7	3.057786821	4.614080497
8	3.041663013	4.20736139
9	3.006362829	4.20736139
10	3.006362829	4.20736139

To check if QOBL accelerates the convergence speed, Table 2 lists the fitness values in the first 10 iterations. It can be seen that in the first 4 iterations, the fitness obtained by QOBL-ICA is greatly smaller than those by Chaos-QOBL-ICA. This illustrates that QOBL-based initialization can indeed generate better individuals nearer to the optimal solution. To more intuitively see this fact, Figure 3 shows the convergence curve of QOBL-ICA and Chaos-QOBL-ICA in the first 100 iterations. One can easily see that the convergence speed of QOBL-ICA is significantly faster than that of Chaos-QOBL-ICA. In summary, the QOBL-based initialization can bring an increase in convergence speed for ICA.

5.2. Description of benchmark functions

First, the experiment on benchmark functions is carried out. Here, 20 benchmark functions are adopted. Table 3 lists the function name, mathematical expression, search range and the global minimum of those adopted benchmark function. In Table 1, functions F01–F08 are unimodal and F09–F20 are multimodal. Compared with unimodal functions, the multimodal functions are more difficult to optimize because they have many local optimums. Many algorithms are easy to trap into local optimum. Though F08 is unimodal, it usually is treated as multimodal function because it has a narrow valley from the achieved local optima to the global minimum, which makes it hard to find its global minimum.

5.3. Optimization results for 10-D benchmark functions

Tables 4 and 5 show the experimental results on 10-D benchmark functions in terms of the best, the worst, the mean and the std in 30 runs of each algorithm. From Tables 4 and 5, one can see from the whole that our proposed QOBL-ICA exhibits better than other referenced algorithms no matter unimodal or multi-modal functions.

As far as the unimodal functions F01–F07 are concerned, the best, worst, mean and std obtained by the proposed QOBL-ICA algorithm are all equal to 0, which indicates that QOBL-ICA finds their global optimum value in each run. As for OBL-ICA, ICA, GBB-ICA, ICA-PSO, DA, GSA, PGJAYA and MPA, they all cannot get the global optimum value in the given iterations. Relatively speaking, MPA performs better among these algorithms except for the proposed QOBL-ICA on F01–F07. For the Rosenbrock function F08, QOBL-ICA does not show superiority compared to other compared algorithms. Among all the referenced algorithms, PGJAYA gives the best function value, 8.584e-07, which is better than 2.7553E-06 found by ICA-PSO and 9.0226E-05 by MPA.

As far as multimodal functions are concerned, from Tables 4 and 5, one can see that the proposed QOBL-ICA demonstrates absolute superiority. For functions F09–F11, F13–F19, QOBL-ICA has found their global optimum. Most importantly, for functions F09–F11, F13–F16 and F19, the worst, mean and std obtained by QOBL-ICA are all equal to 0. This implies that QOBL-ICA can reach the global optimum every time in the 30 runs. For function F12, i.e. Ackley function, the proposed QOBL-ICA, MPA and ICA-PSO all find the global optimum, however, the mean, worst and std obtained by QOBL-ICA and MPA are equal to 0, which implies they are more reliable than ICA-PSO. For functions F17 and F20, though the global minimum has not been arrived at, QOBL-ICA still outperforms other algorithms.

5.4. Optimization results for 30-D benchmark functions

Here, we increase the dimension of benchmark functions to 30. As the benchmark function dimension increases, the complexity of optimization problem also

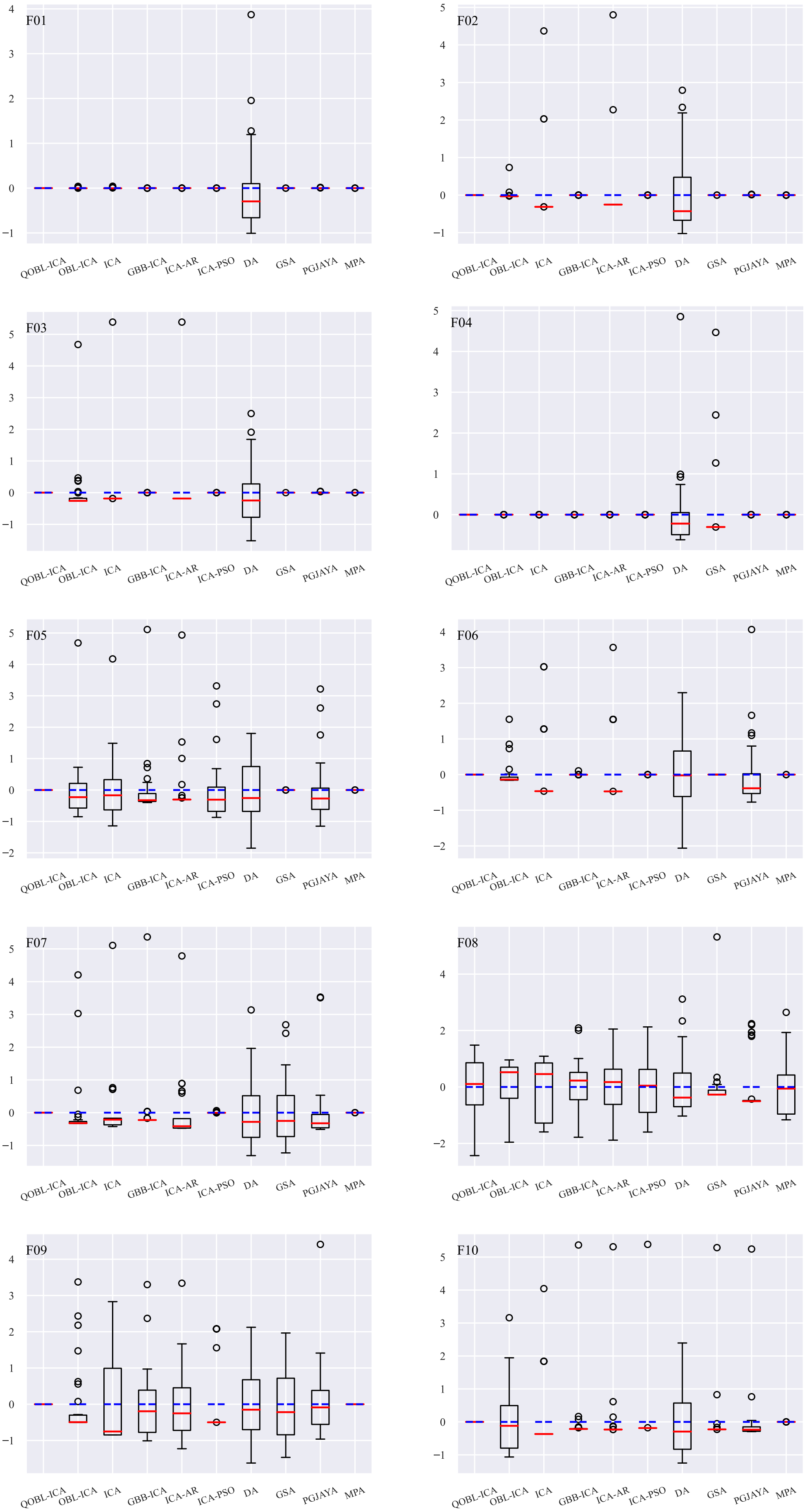

Figure 4. Boxplot of the benchmark functions for 10-D case (F01–F10).

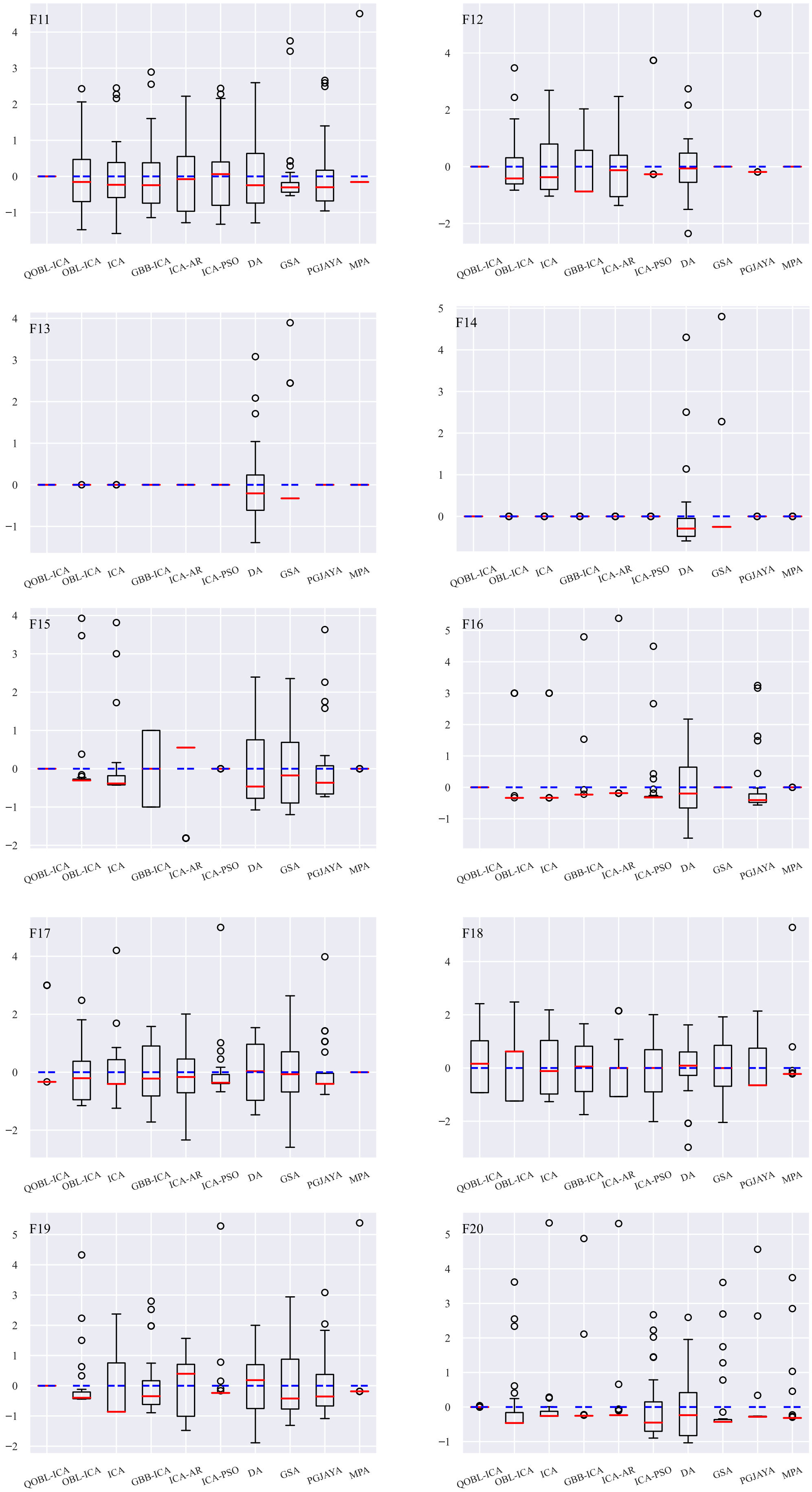

Figure 5. Boxplot of the benchmark functions for 10-D case (F11–F20).

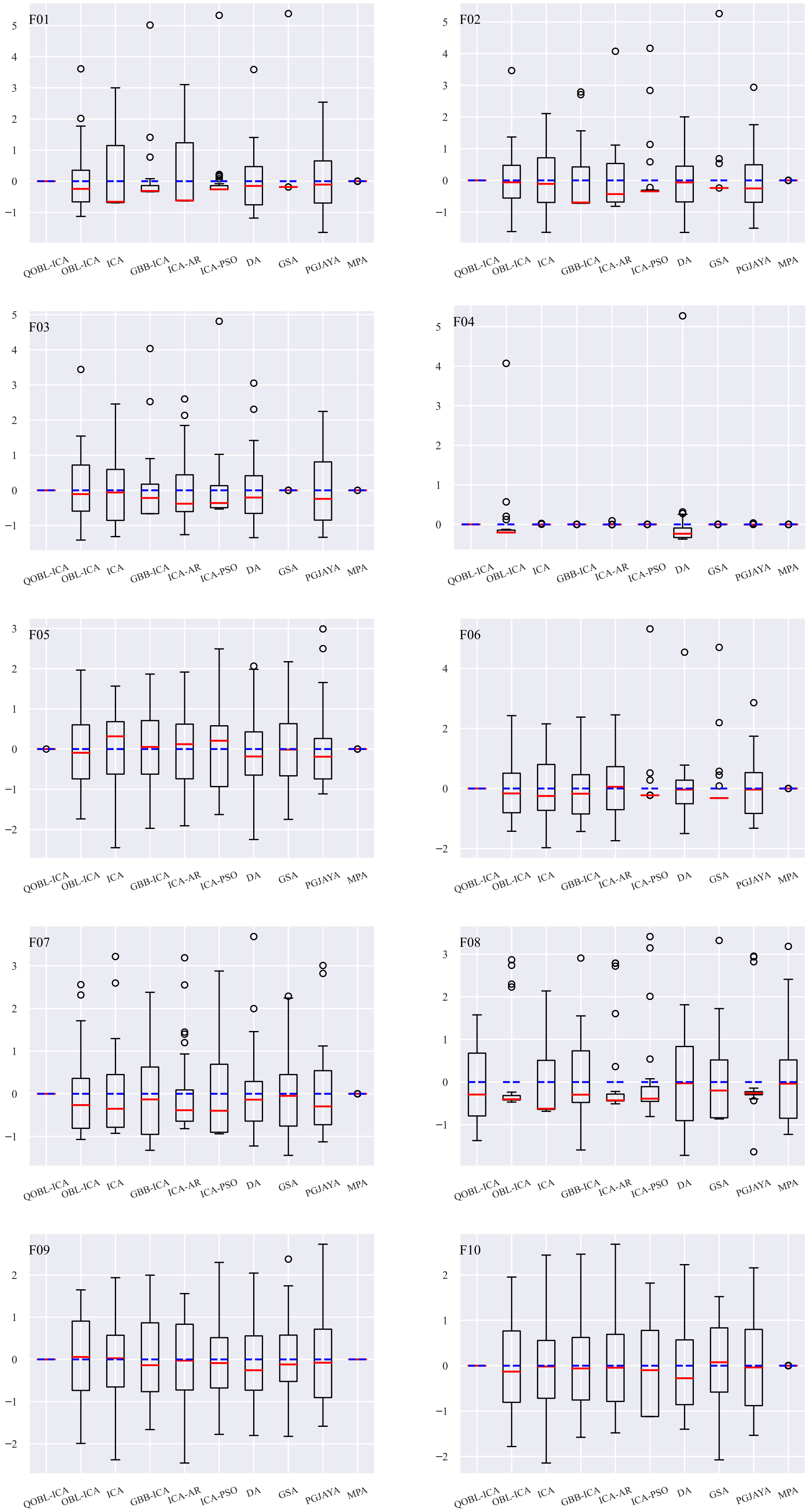

Figure 6. Boxplot of the benchmark functions for 30-D case (F01–F10).

Table 3. Benchmark functions.

Number	Name	Expression	Range	Global Min
F01	Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[−5.12, 5.12]	0
F02	Sum square	$f_2(x) = \sum_{i=1}^{D} ix_i^2$	[−5.12, 5.12]	0
F03	Ellipsoid	$f_3(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2$	[−65.536, 65.536]	0
F04	Sum Power	$f_4(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	[−1, 1]	0
F05	Schwefel 2.21	$f_5(x) = \max\{\lvert x_i \rvert\}$	[−100, 100]	0
F06	Schwefel 2.22	$f_6(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[−10, 10]	0
F07	Elliptic	$f_7(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	[−100, 100]	0
F08	Ronsenbrock	$f_8(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	[−2.048, 2.048]	0
F09	Rastrigin	$f_9(x) = 10D + \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i))$	[−5.12, 5.12]	0
F10	Schwefel1.02	$f_{10}(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	[−100, 100]	0
F11	Griewangk	$f_{11}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[−600, 600]	0
F12	Ackley	$f_{12}(x) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{D} x_i^2}{D}}\right) - \exp\left(\frac{\sum_{i=1}^{D}\cos(2\pi \cdot x_i)}{D}\right) + 20 + e$	[−32,32]	0
F13	Exponential	$f_{13}(x) = -\exp\left(-0.5\sum_{i=1}^{D} x_i^2\right)$	[−1, 1]	0
F14	Quartic	$f_{14}(x) = \sum_{i=1}^{D} ix^4 + rand()$	[−1.28, 1.28]	0
F15	Bent Cigar	$f_{15}(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	[−10,10]	0
F16	Alpine	$f_{16}(x) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	[−10,10]	0
F17	Salomon	$f_{17}(x) = 1 - \cos\left(2\pi \sum_{i=1}^{D} x_i\right) + 0.1\sum_{i=1}^{D} x_i^2$	[−100, 100]	0
F18	Pathologic	$f_{18}(x) = \sum_{i=1}^{D-1}\left(0.5 + \frac{\sin^2(\sqrt{100x_i^2 + x_{i+1}^2}) - 0.5}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)}\right)^2$	[−100,100]	0
F19	NonconRastrigin	$f_{19}(x) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10], y_i = \begin{cases} x_i, \lvert x_i \rvert < 1/2; \\ round(2x_i)/2, \lvert x_i \rvert \geq 1/2. \end{cases}$	[−5.12, 5.12]	0
F20	Schaffer F7	$f_{20}(x) = \left[\frac{1}{D-1}\sum_{i=1}^{D-1}[(x_i^2 + x_{i+1}^2)^{0.25} + (x_i^2 + x_{i+1}^2)^{0.25}\sin(50(x_i^2 + x_{i+1}^2)^{0.1})]\right]^2$	[−100, 100]	0

increases. Thus it would be more effective to examine the performance of the proposed algorithm.

The optimization results are statistically listed in Tables 6 and 7 in terms of the best, worst, mean and std. Comparing Tables 4–7, one can find that, except for the proposed QOBL-ICA, as the dimension increases, the solution quality of all referenced algorithms remarkably decreases. It is surprising that the performance of the proposed QOBL-ICA is still well-preserved. More specifically, for unimodal functions F01–F07, the proposed QOBL-ICA finds the global minimum of all the functions each time among the 30 runs, which shows similar behaviour as the 10-D case. Unfortunately, the other algorithms cannot give satisfactory optimization results and even exhibit failure in finding the optimal solution, for example, ICA, OBL-ICA and DA for F01, F02, F03, and OBL-ICA, ICA, GBB-ICA, AR-ICA, and DA for F05, F06, F07 are failed cases. For Rosenbrock function F08, all the reference algorithms fail to obtain the ideal optimal solution. As multi-modal functions are concerned, the proposed QOBL-ICA successfully finds the global minimum of functions F09–F11, F13–F17 and F19–F20. It should be noted that for functions F09–F11, F13–F16 and F19, the mean, worst and std obtained by QOBL-ICA are equal to 0. From this, one can conclude that QOBL-ICA is more reliable than other algorithms on these benchmark functions. Overall, for 30-D case, MPA shows better performance than other algorithms. It finds the global minimum of functions F09, F11, F13, F19.

To exhibit the optimization results more intuitively, the distribution of the optimal values found by each algorithm among the 30 runs for each function are shown in Figures 4–7 for 10-D and 30-D cases, respectively, via box-and-whisker plots. From Figures 4–7, we can see that the optimal value distribution obtained by QOBL-ICA among 30 runs is more concentrated than other algorithms. This indicates that QOBL-ICA is more reliable than other algorithms.

5.5. Convergence analysis

To further evaluate the performance of QOBL-ICA, Figures 8–11 show the convergence curve of different algorithms when solving 10-D and 30-D benchmark functions, respectively. From Figure 8, we can see that except for function F08, QOBL-ICA converges more rapidly than the other algorithms. Moreover, QOBL-ICA converges closer to the global minimum than other reference algorithms. That is to say, for most of the benchmark functions, there is no distinct stagnation that occurs in the searching procedure

Table 4. Statistical results for 10-D benchmark function (F01-F10).

Function	Indices	QOBL-ICA	OBL-ICA	ICA	GBB-ICA	AR-ICA	ICA-PSO	DA	GSA	PGJAYA	MPA
F01	Best	**0.0000E+00**	2.2261E-13	3.3483E-13	1.6893E-41	5.8910E-47	8.3221E-45	4.0144E-02	7.6832E-18	3.2113E-11	2.1988E-68
	Worst	**0.0000E+00**	4.2299E-08	4.6868E-08	9.2285E-24	7.8503E-26	5.9367E-21	2.7771E+00	6.1971E-17	1.8320E-08	6.4283E-64
	Mean	**0.0000E+00**	2.6791E-09	4.2274E-09	3.1759E-25	2.6176E-27	3.3716E-22	6.0576E-01	2.4372E-17	1.7908E-09	4.8076E-65
	Std	**0.0000E+00**	8.3302E-09	9.7473E-09	1.6839E-24	1.4332E-26	1.3012E-21	5.7056E-01	1.2449E-17	3.6949E-09	1.3642E-64
F02	Best	**0.0000E+00**	3.4569E-13	2.4843E-11	7.8458E-39	2.3665E-47	4.4929E-36	1.2186E-01	2.9899E-17	1.5763E-10	4.6845E-69
	Worst	**0.0000E+00**	8.8913E-07	5.2429E+01	9.3089E-25	5.2429E+01	1.1127E-19	1.0477E+01	1.8933E-16	2.7670E-08	8.5772E-63
	Mean	**0.0000E+00**	3.6093E-08	3.4953E+00	3.4771E-26	2.6214E+00	4.1187E-21	2.8982E+00	9.4044E-17	5.5336E-09	5.0626E-64
	Std	**0.0000E+00**	1.6278E-07	1.1382E+01	1.7028E-25	1.0553E+01	2.0276E-20	2.7602E+00	3.8016E-17	7.3825E-09	1.6555E-63
F03	Best	**0.0000E+00**	1.7590E-11	3.9391E-09	7.1974E-36	1.9156E-45	4.3585E-39	2.0766E+01	3.5485E-17	7.3475E-11	6.4971E-66
	Worst	**0.0000E+00**	4.5554E-05	4.2950E+03	1.4422E-20	4.2950E+03	1.9356E-12	9.6565E+02	4.9266E-16	5.0901E-08	5.6963E-61
	Mean	**0.0000E+00**	2.4130E-06	1.4317E+02	4.8081E-22	1.4317E+02	6.4521E-14	3.7816E+02	1.0537E-16	9.5560E-09	4.2350E-62
	Std	**0.0000E+00**	8.3652E-06	7.8415E+02	2.6331E-21	7.8415E+02	3.5340E-13	2.3929E+02	8.0263E-17	1.2280E-08	1.1180E-61
F04	Best	**0.0000E+00**	5.5232E-23	1.0722E-27	2.9886E-83	1.9353E-103	9.3787E-58	6.4424E-06	7.3942E-15	2.7352E-24	1.0621E-126
	Worst	**0.0000E+00**	1.4172E-13	2.1607E-20	1.6087E-54	1.4762E-75	1.5520E-29	1.2792E-03	6.9750E-01	9.2432E-17	3.9711E-116
	Mean	**0.0000E+00**	5.2422E-15	1.2742E-21	5.3627E-56	8.5884E-77	5.6688E-31	1.4969E-04	4.4287E-02	5.5439E-18	3.4759E-117
	Std	**0.0000E+00**	2.5907E-14	4.1872E-21	2.9371E-55	3.3051E-76	2.8345E-30	2.3573E-04	1.4874E-01	1.9448E-17	9.9866E-117
F05	Best	**0.0000E+00**	7.6828E-02	2.9252E-02	1.1758E-06	2.5351E-08	1.2542E-14	4.1172E+00	2.2502E-09	4.9839E-05	3.1655E-27
	Worst	**0.0000E+00**	7.2976E+00	1.7386E+00	3.1782E-02	9.9919E+00	5.7821E-01	2.7519E+01	4.8969E-09	7.6606E-04	8.7295E-25
	Mean	**0.0000E+00**	1.1873E+00	3.9650E-01	2.3018E-03	5.7990E-01	1.2035E-01	1.5978E+01	3.5734E-09	2.3862E-04	1.3513E-25
	Std	**0.0000E+00**	1.3271E+00	3.2695E-01	5.8660E-03	1.9399E+00	1.4049E-01	6.5145E+00	6.0319E-10	1.6577E-04	2.3441E-25
F06	Best	**0.0000E+00**	4.3518E-11	2.9406E-06	9.3988E-21	4.8283E-17	5.3445E-21	9.6930E-01	6.3027E-09	5.7538E-06	1.0658E-37
	Worst	**0.0000E+00**	2.7229E-06	2.0000E+01	1.1187E-07	2.0001E+01	4.9346E-11	9.4627E+00	2.1004E-08	1.8294E-04	9.9918E-34
	Mean	**0.0000E+00**	2.5400E-07	2.6667E+00	4.0204E-09	2.3334E+00	2.1915E-12	4.9882E+00	1.3690E-08	3.3984E-05	1.8115E-34
	Std	**0.0000E+00**	6.0197E-07	5.8329E+00	2.0428E-08	5.0401E+00	9.3055E-12	1.9815E+00	3.4806E-09	3.6229E-05	2.7113E-34
F07	Best	**0.0000E+00**	8.1595E-09	1.0000E+04	2.9055E-34	1.0636E-32	1.8493E-29	1.2806E+05	5.2348E+03	1.6775E-07	4.1799E-62
	Worst	**0.0000E+00**	1.1488E-03	2.7458E+07	1.0000E+06	2.2760E+07	6.6789E-08	1.4561E+07	9.4486E+04	2.4405E-04	1.8975E-58
	Mean	**0.0000E+00**	8.3893E-05	2.1153E+06	4.0189E+04	2.0601E+06	3.1448E-09	4.3828E+06	3.3266E+04	3.0988E-05	2.1691E-59
	Std	**0.0000E+00**	2.5651E-04	5.0472E+06	1.8198E+05	4.4002E+06	1.3029E-08	3.3009E+06	2.3210E+04	6.0405E-05	3.9462E-59
F08	Best	4.7872E+00	9.7744E-02	1.2667E-01	2.7143E-02	1.7126E-02	2.7554E-06	9.4170E+00	5.3682E+00	8.5824E-07	9.0226E-05
	Worst	6.0600E+00	7.6003E+00	6.9428E+00	8.7685E+00	6.1374E+00	6.6917E+00	7.5388E+01	2.0751E+03	4.8885E+00	9.5703E-01
	Mean	5.5780E+00	5.1344E+00	4.1737E+00	4.0492E+00	2.9475E+00	2.8696E+00	2.5814E+01	1.0624E+02	8.9755E-01	2.9261E-01
	Std	3.3064E-01	2.6167E+00	2.5833E+00	2.2959E+00	1.5808E+00	1.8269E+00	1.6189E+01	3.7666E+02	1.8093E+00	2.5562E-01
F09	Best	**0.0000E+00**	6.0002E-09	2.7320E-06	5.9697E+00	3.9798E+00	0.0000E+00	2.4000E+01	3.9798E+00	2.9849E+00	0.0000E+00
	Worst	**0.0000E+00**	3.9812E+00	5.7849E+01	4.2783E+01	6.6804E+01	9.9496E-01	7.1881E+01	1.4924E+01	4.5768E+01	0.0000E+00
	Mean	**0.0000E+00**	5.1391E-01	1.3328E+01	1.4593E+01	2.0870E+01	1.9228E-01	4.4740E+01	8.6561E+00	1.0646E+01	0.0000E+00
	Std	**0.0000E+00**	1.0453E+00	1.6001E+01	8.6869E+00	1.3993E+01	3.9262E-01	1.3003E+01	3.2441E+00	8.1006E+00	0.0000E+00
F10	Best	**0.0000E+00**	1.1837E+00	6.1775E-03	6.3297E-07	2.7240E-10	3.0630E-27	1.8303E+02	1.5593E-17	9.4793E-07	3.2517E-41
	Worst	**0.0000E+00**	5.9238E+02	1.0000E+04	1.3623E+00	4.6694E+01	1.0427E-01	3.6527E+03	2.0793E-01	7.9649E-04	8.6704E-29
	Mean	**0.0000E+00**	1.5010E+02	8.3358E+02	5.2827E-02	1.9537E+00	3.5064E-03	1.3720E+03	8.5492E-03	4.3266E-05	4.5733E-30
	Std	**0.0000E+00**	1.4235E+02	2.3056E+03	2.4821E-01	8.5660E+00	1.9032E-02	9.6908E+02	3.8357E-02	1.4506E-04	1.5991E-29

Table 5. Statistical results for 10-D benchmark function (F11-F20).

Function	Indices	QOBL-ICA	OBL-ICA	ICA	GBB-ICA	AR-ICA	ICA-PSO	DA	GSA	PGJAYA	MPA
F11	Best	**0.0000E+00**	8.8673E-02	1.0623E-02	2.2141E-02	1.6653E-15	2.4127E-03	1.2117E+00	0.0000E+00	8.6100E-02	0.0000E+00
	Worst	**0.0000E+00**	3.7279E-01	1.8471E-01	6.2925E-01	4.8962E-01	4.3689E-01	7.2668E+00	1.3589E+00	1.3683E+00	2.8766E-05
	Mean	**0.0000E+00**	1.9614E-01	7.8923E-02	1.9413E-01	1.7926E-01	1.5540E-01	3.2197E+00	1.6910E-01	4.2508E-01	9.5886E-07
	Std	**0.0000E+00**	7.3946E-02	4.3889E-02	1.5306E-01	1.4204E-01	1.1733E-01	1.5850E+00	3.2230E-01	3.6038E-01	5.2519E-06
F12	Best	**4.4409E-16**	6.3339E-07	5.7267E-06	2.5313E-14	2.1760E-14	**4.4409E-16**	1.1913E+00	3.6549E-09	1.2320E-06	**4.4409E-16**
	Worst	**4.4409E-16**	6.7950E-04	1.3950E-03	2.3168E+00	3.5740E+00	2.3168E+00	1.5242E+01	1.0514E-08	1.1551E+00	**4.4409E-16**
	Mean	**4.4409E-16**	1.3114E-04	3.9250E-04	6.9635E-01	1.2719E+00	1.5446E-01	7.6816E+00	6.5711E-09	3.8518E-02	**4.4409E-16**
	Std	**0.0000E+00**	1.5949E-04	3.7838E-04	8.1105E-01	9.4747E-01	5.8780E-01	2.8069E+00	1.8231E-09	2.1090E-01	**0.0000E+00**
F13	Best	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	-9.9855E-01	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**
	Worst	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	-9.6504E-01	**−4.0082E-01**	**−1.0000E+00**	**−1.0000E+00**
	Mean	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	-9.8814E-01	**−9.5380E-01**	**−1.0000E+00**	**−1.0000E+00**
	Std	**0.0000E+00**	2.0407E-10	2.7584E-10	1.4625E-11	9.0920E-15	2.0616E-17	7.6233E-03	1.4439E-01	**0.0000E+00**	**0.0000E+00**
F14	Best	**0.0000E+00**	1.0542E-21	5.1754E-20	3.7773E-72	4.2075E-92	8.0564E-65	6.3049E-06	2.1170E-35	6.9366E-21	7.9243E-131
	Worst	**0.0000E+00**	2.4442E-10	5.0750E-15	4.0353E-52	5.0717E-68	2.9426E-38	3.5095E-02	7.2386E+00	1.6406E-15	4.4430E-118
	Mean	**0.0000E+00**	9.7076E-12	2.6819E-16	1.4614E-53	1.9506E-69	9.8132E-40	4.2406E-03	3.6207E-01	2.1295E-16	1.4865E-119
	Std	**0.0000E+00**	4.4902E-11	9.5292E-16	7.3602E-53	9.3194E-69	5.3723E-39	7.2971E-03	1.4574E+00	3.9214E-16	8.1108E-119
F15	Best	**0.0000E+00**	8.4168E-10	1.0000E+02	3.5680E-31	4.4333E-37	3.0487E-26	2.1401E+05	2.8821E+00	1.1949E-05	9.6079E-62
	Worst	**0.0000E+00**	3.2431E-03	1.0000E+02	1.0000E+02	1.0000E+02	3.0336E-14	2.9631E+06	9.9510E+01	1.2458E-02	6.0660E-58
	Mean	**0.0000E+00**	2.3660E-04	1.0000E+02	5.0000E+01	7.6667E+01	1.1659E-15	1.0666E+06	3.5487E+01	2.0966E-03	3.7162E-59
	Std	**0.0000E+00**	7.7747E-04	8.7553E-04	5.0855E+01	4.3018E+01	5.5732E-15	8.0608E+05	2.7661E+01	2.9005E-03	1.1388E-58
F16	Best	**0.0000E+00**	4.4835E-06	5.9343E-07	5.7229E-15	2.4515E-14	1.6395E-17	1.5040E+00	8.0697E-10	1.0553E-05	3.3848E-37
	Worst	**0.0000E+00**	4.4402E+00	4.4402E+00	9.0746E-05	4.4402E+00	1.8174E-01	7.4807E+00	1.9974E-09	5.2764E-03	5.3914E-32
	Mean	**0.0000E+00**	4.4798E-01	4.4404E-01	4.1883E-06	1.4802E-01	1.2148E-02	4.0550E+00	1.3305E-09	7.9022E-04	2.4837E-33
	Std	**0.0000E+00**	1.3536E+00	1.3548E+00	1.7353E-05	8.1066E-01	3.8391E-02	1.6027E+00	3.3646E-10	1.4052E-03	1.0097E-32
F17	Best	**0.0000E+00**	1.9988E-01	1.9987E-01	2.9987E-01	2.9987E-01	9.9873E-02	9.9987E-01	1.5176E-01	9.9873E-02	9.9873E-02
	Worst	**9.9873E-02**	7.3187E-01	1.4999E+00	1.3999E+00	1.6999E+00	2.1127E+00	3.0999E+00	7.0725E-01	1.3999E+00	**9.9873E-02**
	Mean	**9.9873E-03**	3.6868E-01	4.9654E-01	8.7321E-01	1.0532E+00	3.3929E-01	2.0265E+00	4.2705E-01	3.0987E-01	9.9873E-02
	Std	3.0474E-02	1.4892E-01	2.4280E-01	3.3930E-01	3.2772E-01	3.6105E-01	7.0999E-01	1.0811E-01	2.7836E-01	**3.9421E-17**
F18	Best	**0.0000E+00**	4.3780E-05	3.8916E-05	3.8916E-05	3.8916E-05	1.0886E-01	3.8916E-05	1.8622E-01	3.8916E-05	**0.0000E+00**
	Worst	7.1405E-01	4.7605E-01	7.1405E-01	9.3675E-01	7.1405E-01	4.5993E-01	1.2140E+00	9.4181E-01	4.7604E-01	**2.7142E-03**
	Mean	1.9796E-01	1.5871E-01	2.6184E-01	4.8079E-01	2.3800E-01	2.8483E-01	7.8627E-01	5.7577E-01	1.1082E-01	**1.1038E-04**
	Std	2.1722E-01	1.3011E-01	2.1057E-01	2.7909E-01	2.2534E-01	8.8824E-02	2.6842E-01	1.9377E-01	1.7358E-01	**5.0026E-04**
F19	Best	**0.0000E+00**	1.2273E-07	2.8683E-07	1.0000E+00	2.0000E+00	0.0000E+00	1.4032E+01	3.0000E+00	1.0000E+00	0.0000E+00
	Worst	**0.0000E+00**	6.5240E-02	5.0000E+01	2.8000E+01	4.1000E+01	3.5516E-01	5.6257E+01	5.8000E+01	2.1000E+01	1.0003E+00
	Mean	**0.0000E+00**	6.0528E-03	1.3336E+01	7.5333E+00	2.0933E+01	1.5322E-02	3.4535E+01	1.9966E+01	6.2079E+00	3.3342E-02
	Std	**0.0000E+00**	1.3913E-02	1.5718E+01	7.4405E+00	1.3028E+01	6.5421E-02	1.1050E+01	1.3161E+01	4.8788E+00	1.8262E-01
F20	Best	**7.5201E-24**	0.0000E+00	4.1662E-13	4.5683E-15	4.7748E-25	2.3318E-02	5.4347E-03	4.6603E-34	2.1316E-20	5.5956E-19
	Worst	**4.1619E-08**	9.2507E-02	2.1126E-01	4.0794E-02	6.3335E-02	2.3383E+00	2.4643E+00	8.2235E-04	4.8933E-01	5.7914E-05
	Mean	**1.5916E-09**	1.0565E-02	9.8538E-03	2.0178E-03	2.6944E-03	6.0667E-01	7.0696E-01	8.7808E-05	2.8588E-02	4.5183E-06
	Std	**7.5933E-09**	2.3039E-02	3.8442E-02	8.0847E-03	1.1613E-02	6.5974E-01	6.8898E-01	2.0620E-04	1.0264E-01	1.3484E-05

Table 6. Statistical results of 30-D benchmark function (F01-F10).

Function	Indices	QOBL-ICA	OBL-ICA	ICA	GBB-ICA	AR-ICA	ICA-PSO	DA	GSA	PGJAYA	MPA
F01	Best	**0.0000E+00**	1.3422E+00	2.2616E-01	2.8956E-06	2.3355E-05	6.8311E-25	3.0935E+00	3.2393E-16	5.1587E-04	2.7924E-56
	Worst	**0.0000E+00**	1.4101E+01	5.2723E+01	7.5304E-01	5.2430E+01	4.0163E-01	3.7722E+01	2.4558E+01	4.9500E-03	1.3742E-51
	Mean	**0.0000E+00**	4.3862E+00	1.0118E+01	4.7658E-02	8.8940E+00	1.8953E-02	1.1695E+01	8.1860E-01	2.2593E-03	8.2673E-53
	Std	**0.0000E+00**	2.7363E+00	1.4445E+01	1.4299E-01	1.4268E+01	7.3027E-02	7.3774E+00	4.4837E+00	1.0765E-03	2.5463E-52
F02	Best	**0.0000E+00**	2.6717E+01	4.2476E+00	6.0610E-04	3.8628E-02	1.2769E-26	2.3291E+01	3.3480E-15	9.3653E-03	6.2611E-56
	Worst	**0.0000E+00**	4.3705E+02	8.6536E+02	1.6117E+02	9.9615E+02	6.4790E+00	2.5136E+02	1.5601E+02	1.1163E-01	1.9333E-50
	Mean	**0.0000E+00**	1.5720E+02	3.8085E+02	3.3099E+01	1.6663E+02	4.9758E-01	1.2600E+02	6.8055E+00	4.4067E-02	1.1475E-51
	Std	**0.0000E+00**	8.2153E+01	2.3376E+02	4.6675E+01	2.0711E+02	1.4596E+00	6.3630E+01	2.8838E+01	2.3395E-02	3.5372E-51
F03	Best	**0.0000E+00**	7.6007E+03	2.7409E+03	2.3028E-03	1.2294E+02	8.5140E-22	2.6801E+03	2.4698E-15	5.3139E-02	1.2502E-51
	Worst	**0.0000E+00**	7.7754E+04	1.2497E+05	9.0195E+04	1.4178E+05	3.7154E+02	6.8821E+04	3.6220E-13	3.5820E-01	4.5634E-48
	Mean	**0.0000E+00**	2.8058E+04	4.5402E+04	1.2768E+04	4.6429E+04	3.7035E+01	2.2942E+04	2.2685E-14	1.6699E-01	3.1382E-49
	Std	**0.0000E+00**	1.4686E+04	3.2917E+04	1.9506E+04	3.7286E+04	7.0666E+01	1.5287E+04	6.4581E-14	8.6557E-02	9.6588E-49
F04	Best	**0.0000E+00**	4.6021E-10	7.2202E-12	9.9833E-23	4.0709E-26	5.3608E-39	1.1854E-05	4.4846E-16	1.5274E-12	9.6035E-125
	Worst	**0.0000E+00**	1.8895E-05	2.7234E-08	3.7280E-10	9.5159E-08	4.1729E-16	4.2563E-03	6.4755E-11	3.9746E-08	4.8413E-115
	Mean	**0.0000E+00**	9.3670E-07	3.4929E-09	1.2428E-11	3.1724E-09	1.6191E-17	2.9135E-04	3.5914E-12	3.6700E-09	1.8073E-116
	Std	**0.0000E+00**	3.4677E-06	6.3724E-09	6.8063E-11	1.7374E-08	7.6605E-17	7.6358E-04	1.2047E-11	8.7033E-09	8.8188E-116
F05	Best	**0.0000E+00**	4.5083E+01	2.1361E+01	1.8760E+01	4.6448E+01	1.3439E+01	1.9575E+01	7.1648E-01	2.8929E-01	2.7524E-20
	Worst	**1.3046E-307**	7.4678E+01	4.6984E+01	4.4481E+01	6.9584E+01	3.3731E+01	5.1802E+01	1.1490E+01	1.9346E+00	7.9258E-19
	Mean	**0.0000E+00**	5.8976E+01	3.7002E+01	3.1979E+01	5.7996E+01	2.1462E+01	3.6389E+01	5.5239E+00	7.3624E-01	1.8564E-19
	Std	**0.0000E+00**	8.1302E+00	6.4821E+00	6.8174E+00	6.1532E+00	5.0085E+00	7.5991E+00	2.7948E+00	4.0781E-01	1.9293E-19
F06	Best	**0.0000E+00**	5.3708E+00	1.4895E+01	1.4004E+00	3.3498E-01	1.0004E-14	9.1693E+00	7.2653E-08	7.1734E-02	3.1432E-31
	Worst	**0.0000E+00**	2.2241E+01	8.0369E+01	6.0006E+01	6.7725E+01	6.6406E-02	7.8516E+01	5.5309E+00	2.7240E-01	5.5042E-27
	Mean	**0.0000E+00**	1.1609E+01	4.6205E+01	2.3419E+01	2.8297E+01	2.7130E-03	2.6398E+01	3.5245E-01	1.3527E-01	4.8092E-28
	Std	**0.0000E+00**	4.4555E+00	1.6142E+01	1.5645E+01	1.6349E+01	1.2184E-02	1.1676E+01	1.1205E+00	4.8802E-02	1.1108E-27
F07	Best	**0.0000E+00**	3.7158E+06	3.3279E+07	1.0987E+06	3.8594E+07	1.0833E-27	1.2166E+07	1.7292E+04	4.2171E+02	5.7841E-51
	Worst	**0.0000E+00**	1.3886E+07	8.3976E+08	3.6454E+07	5.7514E+08	1.3651E+06	2.4970E+08	2.7269E+05	7.4565E+03	1.1289E-45
	Mean	**0.0000E+00**	6.7146E+06	2.1357E+08	1.3744E+07	1.4811E+08	3.3564E+05	7.1377E+07	1.1608E+05	2.3385E+03	2.0042E-46
	Std	**0.0000E+00**	2.8511E+06	1.9794E+08	9.7094E+06	1.3625E+08	3.6392E+05	4.9215E+07	6.9633E+04	1.7310E+03	3.0097E-46
F08	Best	2.6009E+01	1.1548E+02	9.0886E+01	3.1314E+01	2.7598E+01	**2.1959E+01**	1.7769E+02	2.6194E+01	1.7243E+00	2.3628E+01
	Worst	2.8839E+01	3.1379E+03	2.9696E+03	2.1717E+02	2.5772E+03	8.1780E+01	5.7493E+02	7.6127E+03	7.8968E+01	**2.6250E+01**
	Mean	2.7327E+01	5.4124E+02	7.9020E+02	9.7115E+01	4.2191E+02	3.3420E+01	3.7104E+02	1.5948E+03	2.9264E+01	**2.4358E+01**
	Std	9.7618E-01	9.2082E+02	1.0370E+03	4.1989E+01	7.8562E+02	1.4411E+01	1.1436E+02	1.8428E+03	1.7117E+01	**6.0448E-01**
F09	Best	**0.0000E+00**	8.2664E+01	8.2424E+01	8.6642E+01	8.3807E+01	0.0000E+00	1.4411E+02	3.8803E+01	3.6958E+01	**0.0000E+00**
	Worst	**0.0000E+00**	2.2511E+02	2.0939E+02	2.2998E+02	2.4806E+02	2.6331E+01	2.6526E+02	1.7127E+02	1.6691E+02	**0.0000E+00**
	Mean	**0.0000E+00**	1.6060E+02	1.5242E+02	1.5177E+02	1.8426E+02	1.1476E+01	2.0086E+02	9.6315E+01	8.4663E+01	**0.0000E+00**
	Std	**0.0000E+00**	3.9808E+01	2.9919E+01	3.9829E+01	4.1613E+01	6.5713E+00	3.2020E+01	3.2077E+01	3.0636E+01	**0.0000E+00**
F10	Best	**0.0000E+00**	2.6270E+04	4.6671E+03	3.7444E+03	7.9300E+03	1.1823E-24	6.8923E+03	3.1407E+02	1.1838E+02	5.9118E-17
	Worst	**0.0000E+00**	5.5660E+04	4.5145E+04	2.5175E+04	6.1306E+04	6.9997E+02	4.0110E+04	1.0333E+03	1.1134E+03	6.2728E-09
	Mean	**0.0000E+00**	4.0287E+04	2.3611E+04	1.2122E+04	2.6934E+04	2.6668E+02	1.9715E+04	7.2896E+02	5.3203E+02	2.1440E-10
	Std	**0.0000E+00**	8.0053E+03	8.9867E+03	5.4026E+03	1.3059E+04	2.4213E+02	9.3182E+03	2.0340E+02	2.7425E+02	1.1443E-09

Table 7. Statistical results of 30-D benchmark function (F11-F20).

Function	Indices	QOBL-ICA	OBL-ICA	ICA	GBB-ICA	AR-ICA	ICA-PSO	DA	GSA	PGJAYA	MPA
F11	Best	**0.0000E+00**	7.5173E+00	1.3624E+00	4.3141E-02	2.9813E+00	0.0000E+00	1.1064E+01	4.3961E+00	1.4378E-04	**0.0000E+00**
	Worst	**0.0000E+00**	8.9285E+01	1.8156E+02	9.1231E+01	1.8209E+02	1.8397E+00	6.8770E+01	2.7170E+01	1.2462E-01	**0.0000E+00**
	Mean	**0.0000E+00**	1.4790E+01	5.0078E+01	3.6898E+00	6.1492E+01	3.1470E-01	3.7897E+01	1.4415E+01	3.8283E-02	**0.0000E+00**
	Std	**0.0000E+00**	1.4700E+01	5.1305E+01	1.6543E+01	4.7144E+01	4.0271E-01	1.3644E+01	6.3121E+00	3.9304E-02	**0.0000E+00**
F12	Best	**4.4409E-16**	1.1630E+01	2.6644E+00	2.0069E+00	1.1213E+01	4.4409E-16	6.6913E+00	1.4338E-08	2.1071E-02	**4.4409E-16**
	Worst	**4.4409E-16**	1.9963E+01	1.9963E+01	1.9879E+01	1.9963E+01	1.3992E+01	1.5913E+01	3.9379E-08	1.3641E+00	3.9968E-15
	Mean	**4.4409E-16**	1.9509E+01	1.7691E+01	1.2746E+01	1.8233E+01	5.2479E+00	1.2682E+01	2.1485E-08	5.1710E-01	3.4047E-15
	Std	**0.0000E+00**	1.7592E+00	5.3518E+00	6.3962E+00	2.2594E+00	4.7197E+00	2.1615E+00	6.1567E-09	5.2631E-01	1.3467E-15
F13	Best	**−1.0000E+00**	−9.7283E-01	−9.9861E-01	**−1.0000E+00**	**−1.0000E+00**	**−1.0000E+00**	−9.4050E-01	−3.6788E-01	**−1.0000E+00**	**−1.0000E+00**
	Worst	**−1.0000E+00**	−7.2095E-01	−3.4789E-01	−6.0653E-01	−2.2313E-01	−9.8368E-01	−6.5046E-01	−2.2771E-02	**−1.0000E+00**	**−1.0000E+00**
	Mean	**−1.0000E+00**	−9.1404E-01	−7.6104E-01	−9.8654E-01	−8.3495E-01	−9.9929E-01	−7.9621E-01	−8.1898E-02	**−1.0000E+00**	**−1.0000E+00**
	Std	**0.0000E+00**	5.5214E-02	2.3166E-01	7.1777E-02	2.3336E-01	3.0423E-03	7.2706E-02	7.2336E-02	2.5799E-10	**0.0000E+00**
F14	Best	**0.0000E+00**	5.4167E-02	3.6951E-04	9.1095E-11	1.6410E-11	8.7328E-45	9.0759E-02	9.1598E-32	3.2317E-06	1.4153E-102
	Worst	**0.0000E+00**	1.7600E+01	3.2222E+01	7.3512E-04	3.2212E+01	2.7319E-03	4.2462E+00	3.7581E+01	2.8350E-04	4.3851E-96
	Mean	**0.0000E+00**	3.7598E+00	8.4240E+00	4.3403E-05	5.9056E+00	1.9664E-04	1.0584E+00	5.4370E+00	3.7693E-05	1.5952E-97
	Std	**0.0000E+00**	4.4268E+00	9.9328E+00	1.4220E-04	8.6740E+00	6.2227E-04	9.7479E-01	1.0592E+01	5.4589E-05	7.9843E-97
F15	Best	**0.0000E+00**	5.8350E+06	3.5961E+05	6.5794E+01	6.1744E+02	1.5508E-21	1.0294E+07	4.3525E-02	8.9078E+02	3.4786E-48
	Worst	**0.0000E+00**	2.3064E+07	3.0111E+08	3.8193E+05	1.0209E+08	1.6197E+06	1.2362E+08	9.9780E+01	3.7713E+03	2.4481E-45
	Mean	**0.0000E+00**	1.3893E+07	4.5207E+07	4.1020E+04	2.4436E+07	1.4565E+05	3.9538E+07	3.8878E+01	2.1158E+03	2.5447E-46
	Std	**0.0000E+00**	4.8818E+06	6.7907E+07	8.8693E+04	4.2780E+07	3.7346E+05	2.3550E+07	4.0126E+01	7.7438E+02	4.7690E-46
F16	Best	**0.0000E+00**	2.5795E+00	1.7047E+00	1.5348E-03	7.9873E-01	2.8086E-09	1.0204E+01	6.8944E-09	5.9397E-02	2.5327E-31
	Worst	**0.0000E+00**	2.6735E+01	2.6326E+01	1.5347E+01	2.4206E+01	1.7685E+00	3.6889E+01	2.5004E-03	1.0862E+00	1.5910E-27
	Mean	**0.0000E+00**	1.3694E+01	1.4283E+01	2.6223E+00	1.0055E+01	3.8613E-01	2.3206E+01	2.3237E-04	4.6648E-01	1.2989E-28
	Std	**0.0000E+00**	6.7620E+00	6.5891E+00	4.4578E+00	6.0999E+00	5.7401E-01	6.5032E+00	7.1307E-04	2.7054E-01	3.2043E-28
F17	Best	**0.0000E+00**	3.2999E+00	1.9999E+00	5.0999E+00	4.8999E+00	3.0341E-03	3.1999E+00	2.3469E+00	8.9987E-01	9.9873E-02
	Worst	**9.9873E-02**	1.5900E+01	1.4500E+01	1.4900E+01	2.1000E+01	9.8132E+00	1.1100E+01	6.6003E+00	3.0999E+00	1.9987E-01
	Mean	**4.9937E-02**	9.6457E+00	5.8325E+00	8.2232E+00	1.2260E+01	6.6452E+00	7.0399E+00	3.7893E+00	1.6265E+00	1.1654E-01
	Std	**5.0790E-02**	3.7245E+00	3.8587E+00	2.3295E+00	3.7778E+00	2.3430E+00	2.0858E+00	8.9769E-01	4.8419E-01	3.7905E-02
F18	Best	**3.5696E-13**	7.1414E-01	7.1414E-01	2.4974E+00	4.7614E-01	3.3105E+00	2.3802E+00	3.5344E-01	4.7573E-01	8.8337E-18
	Worst	3.0942E+00	2.1422E+00	2.8562E+00	4.3877E+00	2.8562E+00	4.5832E+00	5.2014E+00	1.4932E+00	3.0936E+00	**1.3912E+00**
	Mean	2.1739E+00	1.4599E+00	1.9755E+00	3.5515E+00	1.6697E+00	4.1009E+00	4.5124E+00	9.0021E-01	1.7101E+00	**1.7910E-01**
	Std	6.4270E-01	4.0377E-01	5.1216E-01	4.4035E-01	5.9099E-01	3.1679E-01	7.2056E-01	**2.8744E-01**	7.5572E-01	3.1021E-01
F19	Best	**0.0000E+00**	8.4044E+01	6.4110E+01	4.0061E+01	1.3800E+02	0.0000E+00	9.0883E+01	7.2000E+01	2.9498E+01	**0.0000E+00**
	Worst	**0.0000E+00**	2.4109E+02	2.5903E+02	2.1800E+02	2.6900E+02	2.9113E+01	3.1181E+02	2.8800E+02	1.7500E+02	**0.0000E+00**
	Mean	**0.0000E+00**	1.5812E+02	1.4020E+02	1.1621E+02	1.9910E+02	1.3113E+01	1.9292E+02	1.5050E+02	9.3801E+01	**0.0000E+00**
	Std	**0.0000E+00**	3.4988E+01	4.2281E+01	4.2618E+01	3.3099E+01	6.7533E+00	5.2837E+01	4.5552E+01	3.5807E+01	**0.0000E+00**
F20	Best	**0.0000E+00**	1.4253E-02	1.3431E-02	2.5130E-04	1.0807E-02	9.2098E-05	4.2885E+00	1.1790E-05	2.5720E-04	4.9866E-11
	Worst	**1.1613E-05**	1.7422E+00	1.1085E+00	9.5342E-01	1.8757E+00	9.2805E+00	1.5695E+01	1.9649E-01	2.0112E+00	3.4356E-03
	Mean	**4.2000E-07**	4.6459E-01	2.3481E-01	2.1768E-01	4.2105E-01	3.4270E+00	8.9710E+00	1.8591E-02	3.8869E-01	2.0868E-04
	Std	**2.1202E-06**	5.2530E-01	2.7254E-01	2.3964E-01	4.6115E-01	2.6779E+00	3.2323E+00	3.7361E-02	5.2564E-01	6.7457E-04

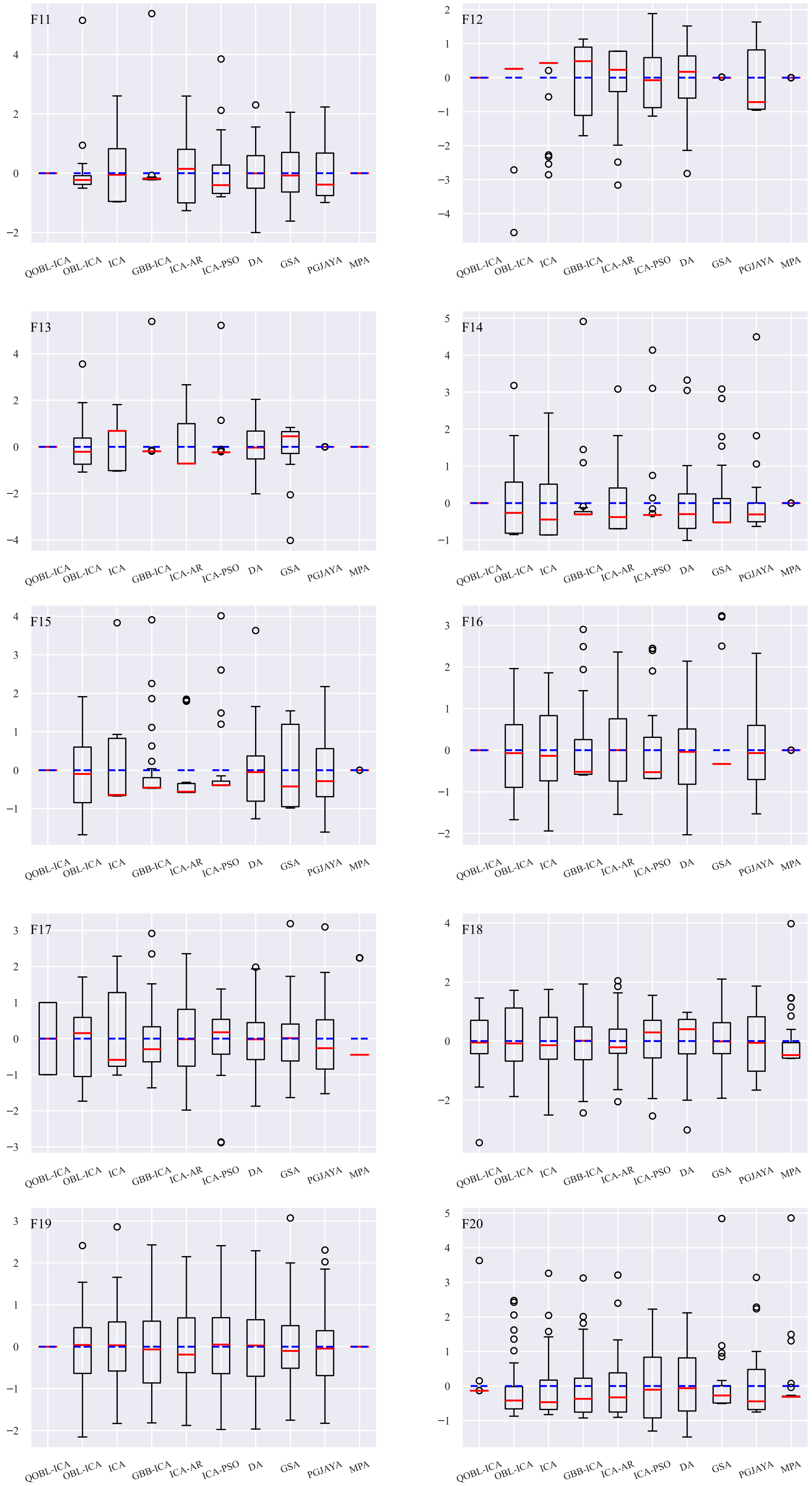

Figure 7. Boxplot of the benchmark functions for 30-D case (F11–F20).

for QOBL-ICA. At the same time, one can see from Figures 8 and 9 that even for multimodal function, QOBL-ICA also shows faster convergence speed and convergence characteristics than other referenced algorithms. However, from Figures 8 and 9, one can see that many algorithms trap into local optimal and cannot escape from it. Thus only local optimal solutions can be found.

Figures 10 and 11 show the convergence curve of all algorithms for 30-D benchmark functions. No matter whether for uni-modal or multimodal functions, most algorithms exhibit stagnation phenomena and converge to a local optimum. For example, for uni-modal functions F01–F07, OBL-ICA, ICA, GBB-ICA, ICA-AR, ICA-PSO, DA, GSA, PGJAYA and MPA all exhibit searching stagnation and converge to local optimum. Only our proposed QOBL-ICA possesses excellent local and global searching ability. Not only is the convergence speed faster but also there is no stagnation occurs. For multi-modal functions, similar phenomena can be observed in the case of unimodal functions. In a word, the convergence characteristic of the proposed QOBL-ICA is better than other referenced algorithms.

5.6. Wilcoxon test

As well as the qualitative analysis presented in Sections 5.3, 5.4 and 5.5, in this section, a Wilcoxon signed rank sum test is carried out to further objectively evaluate the performance of the proposed QOBL-ICA. The significant level of the test is selected as $\alpha = 0.05$. Wilcoxon test compares two group data and determines which is better by testing a statistical hypothesis. The hypothesis is: H_0: group A has no difference to group B; H_1: group A is better than group B. In our test experiments, the best values obtained by QOBL-ICA among the 30 runs are pairwise comparisons with those by other ICAs. Tables 8 and 9 list the test results for 10-D and 30-D benchmark functions, where h equals 1 means one rejects H_0 or accepts H_1. That is to say, $h = 1$ means QOBL-ICA is superior to the compared algorithm. From Tables 4 and 5, we can see that our proposed QOBL-ICA indeed outperforms other algorithms.

5.7. Applications to engineering design problems

Here, our QOBL-ICA is applied to solve several engineering design problems and the obtained optimal solutions are compared with those presented in the literature. Since most engineering design problems have constraints, here, we use the penalty function method to handle constraints. The penalty function method augments the original objective function $f(x)$ as the way shown in (17),

$$F(x) = f(x) + \sum_{j=1}^{m} (P * \max\{0, g_j^2(x)\}), \tag{17}$$

where $g_j(x)$ is the jth inequality constraint, P is a penalty factor. In our paper, $P = 10^5$.

5.7.1. I-beam design problem

The I-beam design problem (IB-DP) focus on finding the minimum of vertical deflection while satisfies the cross-sectional area and stress constraints under given loads. The design variables are the width of flange $b(x_1)$, the height of section $h(x_2)$, the thickness of the web $t_w(x_3)$ and the thickness of the flange $t_f(x_4)$. The maximum vertical deflection of the beam is $f(x) = PL^3/48EI$ when the length of the beam (L) and modulus of elasticity (E) is 5200 cm and 523.104 kN/cm^2, respectively. The I-beam design problem can be formulated as an optimization problem as

$$\begin{aligned}
\text{Consider} \quad & \boldsymbol{x} = [x_1\ x_2\ x_3\ x_4] = [b\ h\ t_w\ t_f] \\
\min \quad & f(\boldsymbol{x}) = \frac{5000}{\frac{x_3(x_2-2x_4)^3}{12} + \frac{x_1x_4^3}{6} + 2x_1x_4\left(\frac{x_2-x_4}{2}\right)^2} \\
\text{s.t.} \quad & g_1(\boldsymbol{x}) = 2x_1x_3 + x_3\,(x_2 - 2x_4) \le 300 \\
& g_2(\boldsymbol{x}) = \frac{18x_2 \times 10^4}{x_3\,(x_2-2x_4)^3 + 2x_1x_3\left(4x_4^2 + 3x_2\,(x_2-2x_4)\right)} \\
& \quad + \frac{15x_1 \times 10^3}{(x_2-2x_4)\,x_3^2 + 2x_3x_1^3} \le 56 \\
\text{with} \quad & 10 \le x_1 \le 50 \\
& 10 \le x_2 \le 80 \\
& 0.9 \le x_3, x_4 \le 5.
\end{aligned}$$

In the literature, the cuckoo search algorithm (CSA) [59], chaos game optimization (CGO) [60], whale optimization algorithm (WOA) [61], etc., had been used to solve this problem. The optimization results obtained by QOBL-ICA and other comparing algorithms are shown in Table 10. From Table 10, one can see that QOBL-ICA outperforms CS, ARSM, CGO and WOA, and is almost the same as that by MFO. That is to say, the QOBL-ICA obtains the state-of-the-art optimization result for I-Beam design problem.

5.7.2. Speed reducer design problem (SRDP)

SRDP pays attention to find the minimum value of the total weight of the speed reducer. This problem is more complex because it involves more design variables, which can be continuous or discrete. The involved variables include the face width $b(x_1)$, module of teeth $m(x_2)$, number of teeth in the pinion $z(x_3)$, length of the first shaft between bearings $l_1(x_4)$, length of the second shaft between bearings $l_2(x_5)$, diameter of first shaft $d_1(x_6)$ and diameter of second shaft $d_2(x_7)$. The

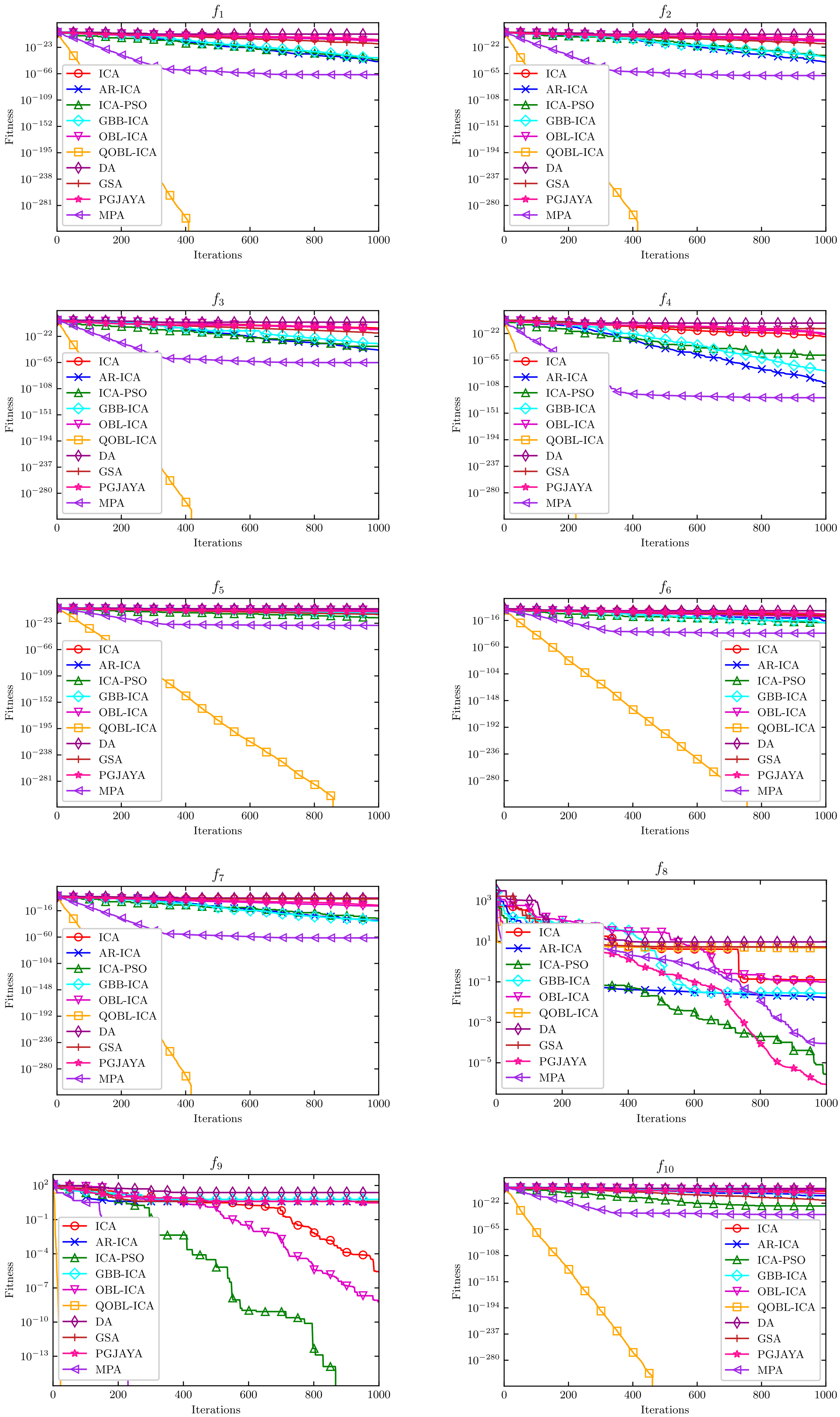

Figure 8. Convergent characteristic of benchmark functions for 10-D case (F01–F10).

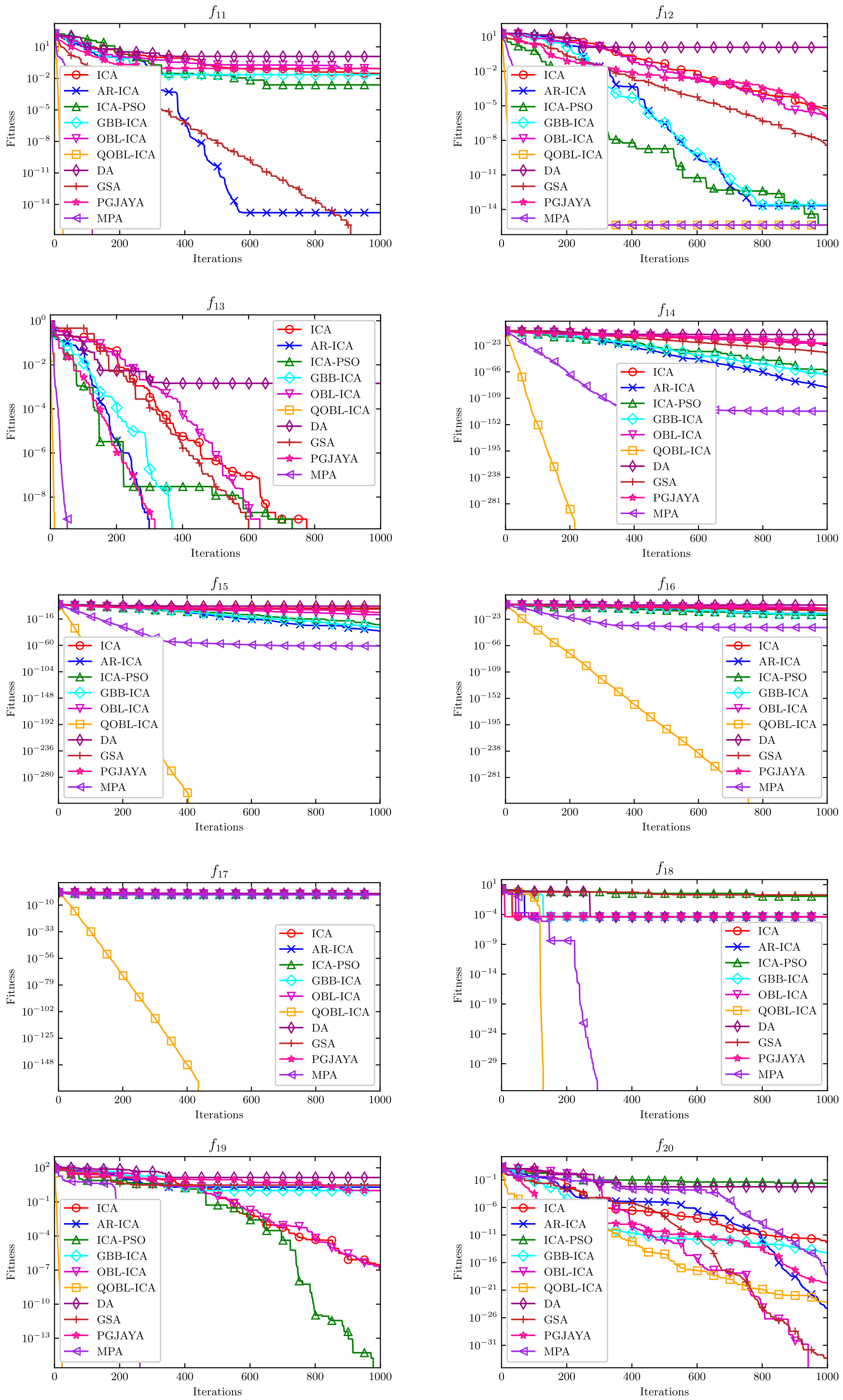

Figure 9. Convergent characteristic of benchmark functions for 10-D case (F11–F20).

SRDP can be mathematically expressed as the following optimization problem:

$$
\begin{aligned}
\text{Consider}\quad & \boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7] \\
& \quad = [b\ m\ z\ l_1\ l_2\ d_1\ d_2] \\
\min\quad & f(\boldsymbol{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 \\
& \quad - 43.0934) \\
& \quad - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\
& \quad + 0.7854(x_4x_6^2 + x_5x_7^2) \\
\text{s.t.}\quad & g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0 \\
& g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0 \\
& g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0
\end{aligned}
$$

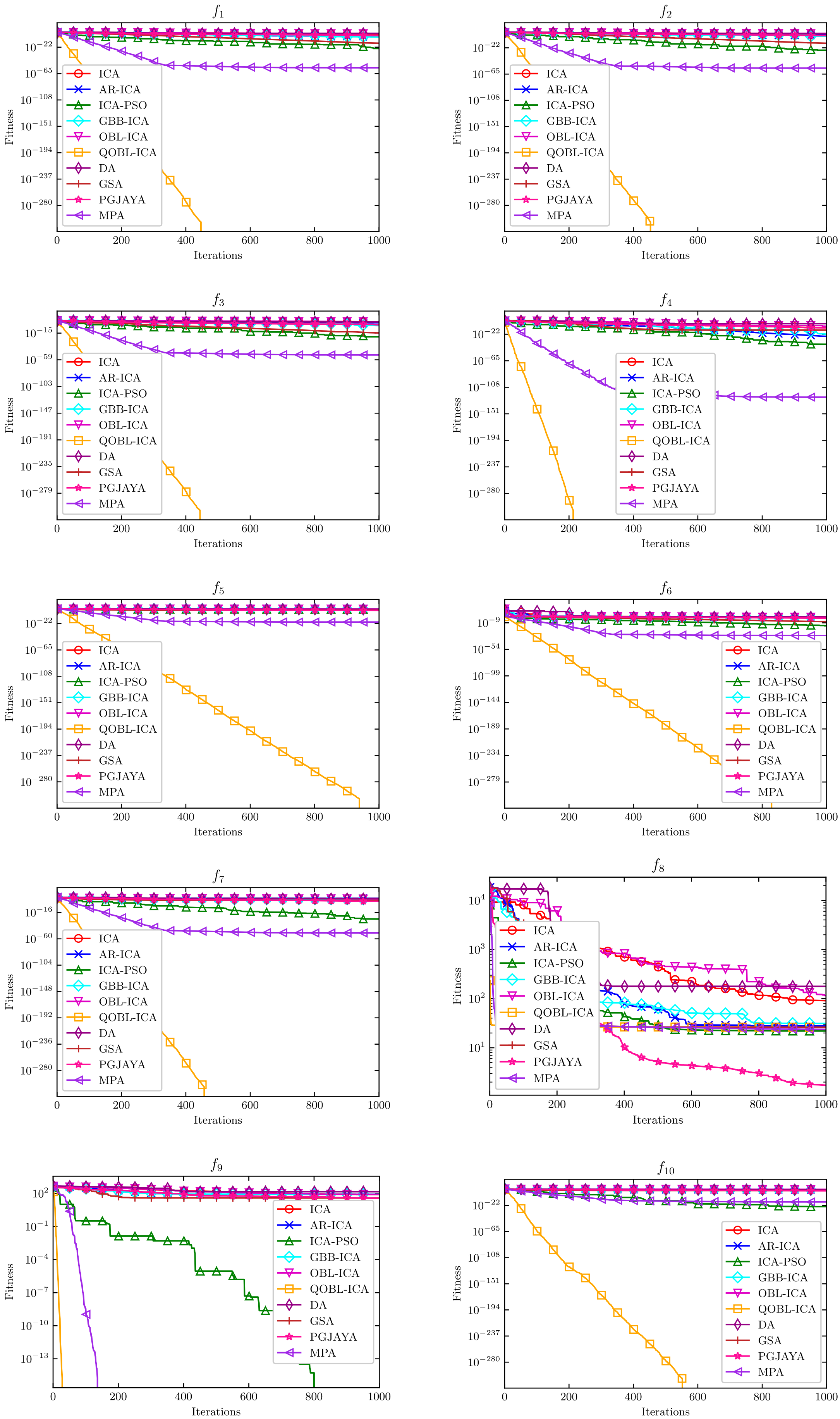

Figure 10. Convergent characteristic of benchmark functions for 30-D case (F01–F10).

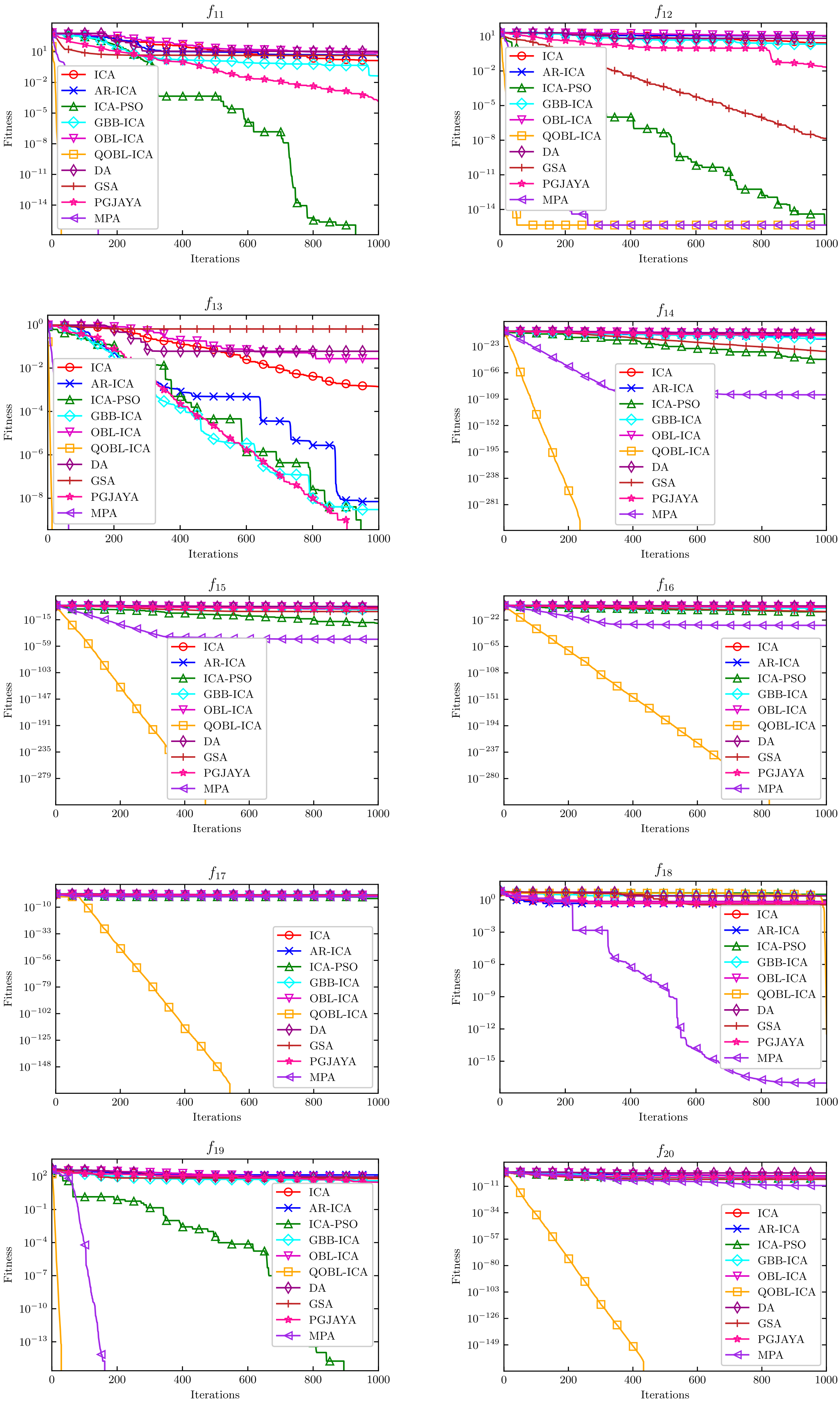

Figure 11. Convergent characteristic of benchmark functions for 30-D case (F11-F20).

Table 8. *h*-value of Wilcoxon signed rank test for 10-D case.

Functions	ICA	GBB-ICA	OBL-ICA	AR-ICA	ICA-PSO	DA	GSA	PGJAYA	MPA
F01	1	1	1	1	1	1	1	1	1
F02	1	1	1	1	1	1	1	1	1
F03	1	1	1	1	1	1	1	1	1
F04	1	1	1	1	1	1	1	1	1
F05	1	1	1	1	1	1	1	1	1
F06	1	1	1	1	1	1	1	1	1
F07	1	1	1	1	1	1	1	1	1
F08	0	0	0	0	0	1	1	0	0
F09	1	1	1	1	1	1	1	1	0
F10	1	1	1	1	1	1	1	1	1
F11	1	1	1	1	1	1	1	1	0
F12	1	1	1	1	1	1	1	1	0
F13	1	1	1	1	0	1	0	0	0
F14	1	1	1	1	1	1	1	1	1
F15	1	1	1	1	1	1	1	1	1
F16	1	1	1	1	1	1	1	1	1
F17	1	1	1	1	1	1	1	1	1
F18	1	1	0	0	1	1	1	0	0
F19	1	1	1	1	1	1	1	1	1
F20	1	1	1	1	1	1	1	1	1

Table 9. *h*-value of Wilcoxon signed rank test for 30-D case.

Functions	ICA	GBB-ICA	OBL-ICA	AR-ICA	ICA-PSO	DA	GSA	PGJAYA	MPA
F01	1	1	1	1	1	1	1	1	1
F02	1	1	1	1	1	1	1	1	1
F03	1	1	1	1	1	1	1	1	1
F04	1	1	1	1	1	1	1	1	1
F05	1	1	1	1	1	1	1	1	1
F06	1	1	1	1	1	1	1	1	1
F07	1	1	1	1	1	1	1	1	1
F08	1	1	1	1	1	1	1	0	0
F09	1	1	1	1	1	1	1	1	0
F10	1	1	1	1	1	1	1	1	1
F11	1	1	1	1	1	1	1	1	0
F12	1	1	1	1	1	1	1	1	1
F13	1	1	1	1	1	1	1	1	0
F14	1	1	1	1	1	1	1	1	1
F15	1	1	1	1	1	1	1	1	1
F16	1	1	1	1	1	1	1	1	1
F17	1	1	1	1	1	1	1	1	1
F18	0	1	0	0	1	1	0	0	0
F19	1	1	1	1	1	1	1	1	0
F20	1	1	1	1	1	1	1	1	1

Table 10. Best results of I-beam design problem.

		Best variable value			
Algorithms	Best cost	b	h	t_w	t_f
QOBL-ICA	0.0066259	50	80	1.764706	5
CS [59]	0.0130747	20	80	0.9	2.321672
ARSM [59]	0.0157	37.05	80	1.71	2.31
CGO [60]	0.01307412	50	80	0.9	2.321792
MFO [61]	0.0066259	50	80	1.7647	5
WOA [61]	0.00662619	49.99799	80	1.764748	5

$$g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0$$

$$g_5(x) = \frac{\left(\left(\frac{754x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6\right)^{\frac{1}{2}}}{110x_6^3} - 1 \leq 0$$

$$g_6(x) = \frac{\left(\left(\frac{754x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6\right)^{\frac{1}{2}}}{85x_7^3} - 1 \leq 0$$

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

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

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

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

with
$$2.6 \leq x_1 \leq 3.6$$
$$0.7 \leq x_2 \leq 0.8$$
$$17 \leq x_3 \leq 28$$
$$7.3 \leq x_4, x_5 \leq 8.3$$
$$2.9 \leq x_6 \leq 3.9$$
$$5.0 \leq x_7 \leq 5.5$$

Several meta-heuristic algorithms including ASOINU [62], beetle swarm optimization algorithm (BA) [62],

Table 11. Best results of speed reducer problem.

Algorithm	Best cost	Best variable value						
		x_1	x_2	x_3	x_4	x_5	x_6	x_7
QOBL-ICA	**2994.3554**	3.4980	0.7000	17.0000	7.3000	7.7152	3.3512	5.2867
BA [62]	2994.4671	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2875
NDE [62]	2994.4711	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2866
IPSO [62]	2994.4711	3.5000	0.6999	17.0000	7.2999	7.7153	3.3502	5.2866
ASOINU [62]	2996.2448	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2865
CSA [63]	2994.4710	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867
EJAYA [64]	2994.4711	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867

novel differential evolution (NDE) algorithm [62], interval particle swarm optimization (IPSO) [62], chameleon swarm algorithm (CAS) [63], Enhanced jaya algorithm (EJAYA) [64] had been used to solve SRDP. Table 11 shows the best optimization result obtained by QOBL-ICA and the comparing algorithms. It can be seen from Table 11 that the result of our QOBL-ICA is better than that of other comparing algorithms.

5.7.3. Car side impact design problem (CSIDP)

This problem aims to find the minimum value of the total weight of the door to avoid side impact. The involved design variables include the thickness of B-Pillar inner (x_1), the thickness of B-Pillar reinforcement (x_2), the thickness of floor side inner (x_3), the thickness of cross members (x_4), the thickness of door beam (x_5), the thickness of door beltline reinforcement (x_6), the thickness of roof rail (x_7), materials of B-Pillar inner (x_8), materials of floor side inner (x_9), barrier height (x_{10}) and hitting position (x_{11}). This problem is mathematically can be expressed an optimization problem as

$$
\begin{aligned}
\text{Consider}\quad & \boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}] \\
\min\quad & f(\boldsymbol{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 \\
& \quad + 4.01x_4 + 1.78x_5 + 2.73x_7 \\
\text{s.t}\quad & g_1(\boldsymbol{x}) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} \\
& \quad - 0.484x_3x_9 + 0.01343x_6x_{10} - 1 \le 0 \\
& g_2(\boldsymbol{x}) = 0.261 - 0.0159x_1x_2 - 0.0188x_1x_8 \\
& \quad - 0.0191x_2x_7 + 0.0144x_3x_5 \\
& \quad + 0.0008757x_5x_{10} \\
& \quad + 0.08045x_6x_9 + 0.00139x_8x_{11} \\
& \quad + 0.00001575x_{10}x_{11} - 0.32 \le 0 \\
& g_3(\boldsymbol{x}) = 0.214 + 0.00817x_5 - 0.131x_1x_8 \\
& \quad - 0.0704x_1x_9 + 0.03099x_2x_6 \\
& \quad - 0.018x_2x_7 + 0.0208x_3x_8 \\
& \quad + 0.121x_3x_9 - 0.00364x_5x_6 \\
& \quad + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} \\
& \quad + 0.00121x_8x_{11} + 0.00184x_9x_{10} \\
& \quad - 0.02x_2^2 \le 0.32 \\
& g_4(\boldsymbol{x}) = 0.74 - 0.61x_2 - 0.163x_3x_8 \\
& \quad + 0.001232x_3x_{10} - 0.166x_7x_9 \\
& \quad + 0.227x_2^2 - 0.32 \le 0 \\
& g_5(\boldsymbol{x}) = 28.98 + 3.818x_3 - 4.2x_1x_2 \\
& \quad + 0.0207x_5x_{10} + 6.63x_6x_9 \\
& \quad - 7.7x_7x_8 + 0.32x_9x_{10} \le 32 \\
& g_6(\boldsymbol{x}) = 33.86 + 2.95x_3 + 0.1792x_3 \\
& \quad - 5.057x_1x_2 - 11.0x_2x_8 \\
& \quad - 0.0215x_5x_{10} - 9.98x_7x_8 \\
& \quad + 22.0x_8x_9 - 32 \le 0 \\
& g_7(\boldsymbol{x}) = 46.36 - 9.9x_2 - 12.9x_1x_2 \\
& \quad + 0.1107x_3x_{10} - 32 \le 0 \\
& g_8(\boldsymbol{x}) = 4.72 - 0.5x_4 - 0.19x_2x_3 \\
& \quad - 0.0122x_4x_{10} + 0.009325x_6x_{10} \\
& \quad + 0.000191x_{11}^2 - 4 \le 0 \\
& g_9(\boldsymbol{x}) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 \\
& \quad + 0.02054x_3x_{10} - 0.0198x_4x_{10} \\
& \quad + 0.028x_6x_{10} - 9.9 \le 0 \\
& g_{10}(\boldsymbol{x}) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 \\
& \quad + 0.0432x_9x_{10} - 0.0556x_9x_{11} \\
& \quad - 0.000786x_{11}^2 - 15.7 \le 0 \\
\text{with}\quad & 0.5 \le x_i \le 1.5,\ i = 1,2,3,4,5,6,7 \\
& x_i \in \{0.192, 0.345\},\ i = 8,9 \\
& -30 \le x_i \le 30,\ i = 10,11
\end{aligned}
$$

This problem had also been solved by several algorithms. Table 12 shows the optimization results of QOBL-ICA and other comparing algorithms including PSO, DE, FA, CS [59] and social network search (SNS) algorithm [65]. One can see from Table 12 that our QOBL-ICA obtains the minimum cost 21.1935, which is smaller than the number of 22.84 obtained by the other comparing algorithms. This implies that our QOBL-ICA is superior to

6. Conclusion

To alleviate the disadvantage of the original ICA, an improved ICA is proposed by integrating the QOBL

Table 12. Best results of Car side impact.

		Best variables										
Algorithm	Best cost	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
QOBL-ICA	21.1935	0.5	0.8724232	0.5	1.296124	0.5	1.5	0.5	1	1	−6.01976688	−0.004596
PSO [59]	22.84474	0.5	1.1167	0.5	1.30208	0.5	1.5	0.5	0.345	0.192	−19.54935	−0.00431
DE [59]	22.84298	0.5	1.1167	0.5	1.30208	0.5	1.5	0.5	0.345	0.192	−19.5494	−0.00431
FA [59]	22.84298	0.5	1.36	0.5	1.202	0.5	1.12	0.5	0.345	0.192	8.87307	−18.99808
CS [59]	22.84294	0.5	1.11643	0.5	1.302	0.5	1.5	0.5	0.345	0.192	−19.54935	−0.00431
SNS [65]	22.8429	0.5	1.1159332	0.5	1.302919	0.5	1.5	0.5	0.345	0.192	−19.6388662	1.49E-06

into ICA, named QOBL-ICA. To be specific, an QOBL-based population initialization is presented to produce a group of high-quality initial individuals more near to the optimal solution. In addition, a QOBL-based assimilation strategy is proposed to enhance the global exploration ability of ICA. The introduced QOBL not only speeds up the convergence of the algorithm but also increases the chance of jumping out local optimum. The proposed QOBL-ICA shows great superiority to most other advanced meta-heuristic algorithms, which is verified by extensive comparison in terms of optimization results and convergence curves.

Since the QOL strategy can provide improved performance for ICA to solve complex optimization problems, in the future research works, we will attempt to extend QOBL-ICA to multi-objective case. That is to say, a multi-object version of QOBL-ICA will be studied to solve multi-object optimization problems.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was sponsored in part by Public Welfare Project of Zhejiang Province of China under Grant numbers LGN22C140007 and LGG19F050003.

Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article.

ORCID

Fei Wu http://orcid.org/0000-0003-2624-3134

References

[1] Gürses D, Mehta P, Sait SM, et al. A multi-strategy boosted prairie dog optimization algorithm for global optimization of heat exchangers. Mater Test. 2023;65(9):1396–1404. doi: 10.1515/mt-2023-0082

[2] Jia H, Zhou X, Zhang J, et al. Modified crayfish optimization algorithm for solving multiple engineering application problems. Artif Intell Rev. 2024;57(5):127–184. doi: 10.1007/s10462-024-10738-x

[3] Sait SM, Mehta P, Yıldız AR. Optimal design of structural engineering components using artificial neural network-assisted crayfish algorithm. Mater Test. 2024;66(9):1439–1448. doi: 10.1515/mt-2024-0075

[4] Kopar M, Yildiz AR. Composite disc optimization using hunger games search optimization algorithm. Mater Test. 2023;65(8):1222–1229. doi: 10.1515/mt-2023-0067

[5] Mehta P, Sait SM, Yıldız BS, et al. A new enhanced mountain gazelle optimizer and artificial neural network for global optimization of mechanical design problems. Mater Test. 2024;66(4):544–552. doi: 10.1515/mt-2023-0332

[6] Shu-Chuan Chu SCC, Shu-Chuan Chu TTW, Ting-Ting Wang ARY, et al. Ship rescue optimization: a new metaheuristic algorithm for solving engineering problems. J Internet Technol. 2024;25(1):61–77. doi: 10.53106/160792642024012501006

[7] Duzgun E, Acar E, Yildiz AR. A novel chaotic artificial rabbits algorithm for optimization of constrained engineering problems. Mater Test. 2024;65(8):1222–1229. doi: 10.1515/mt-2023-0067

[8] Yıldız BS, Kumar S, Panagant N, et al. A novel hybrid arithmetic optimization algorithm for solving constrained optimization problems. Knowl Based Syst. 2023;271. doi: 10.1016/j.knosys.2023.110554

[9] Kumar S, Yildiz BS, Mehta P, et al. Chaotic marine predators algorithm for global optimization of real-world engineering problems. Knowl Based Syst. 2023; 261:Article ID 110192. doi: 10.1016/j.knosys.2022.110192

[10] Anosri S, Panagant N, Champasak P, et al. A comparative study of state-of-the-art metaheuristics for solving many-objective optimization problems of fixed wing unmanned aerial vehicle conceptual design. Arch Comput Method E. 2023;30(6):3657–3671. doi: 10.1007/s11831-023-09914-z

[11] Kanokmedhakul Y, Bureerat S, Panagant N, et al. Metaheuristic-assisted complex h-infinity flight control tuning for the hawkeye unmanned aerial vehicle: a comparative study. Expert Syst Appl. 2024;248:Article ID 123428. doi: 10.1016/j.eswa.2024.123428

[12] Zhi-Chao Dou ZCD, Zhi-Chao Dou SCC, Shu-Chuan Chu ZZ, et al. Gbrun: a gradient search-based binary Runge Kutta optimizer for feature selection. J Internet Technol. 2024;25(3):341–353. doi: 10.53106/160792642024052503001

[13] Mehta P, Yildiz AR, Sait SM, et al. Enhancing the structural performance of engineering components using the geometric mean optimizer. Mater Test. 2024;66(7):1063–1073. doi: 10.1515/mt-2024-0005

[14] Meng Z, Qian Q, Xu M, et al. Pinn-form: a new physics-informed neural network for reliability analysis with partial differential equation. Comput Method Appl M. 2023;414:Article ID 116172. doi: 10.1016/j.cma.2023.116172

[15] Meng Z, Yıldız BS, Li G, et al. Application of state-of-the-art multiobjective metaheuristic algorithms in reliability-based design optimization: a comparative study. Struct Multidiscip O. 2023;66(8):191–218. doi: 10.1007/s00158-023-03639-0

[16] Yang Y, Gao Y, Wu J, et al. Improving PID controller performance in nonlinear oscillatory automatic generation control systems using a multi-objective marine predator algorithm with enhanced diversity. J Bionic Eng. 2024.
[17] Movassagh AA, Alzubi JA, Gheisari M, et al. Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model. J Ambient Intell Humaniz Comput. 2023;14(5):6017–6025. doi: 10.1007/s12652-020-02623-6
[18] Alzubi JA, Manikandan R, Alzubi OA, et al. Hashed Needham Schroeder industrial IOT based cost optimized deep secured data transmission in cloud. Measurement. 2020;150:Article ID 107077. doi: 10.1016/j.measurement.2019.107077
[19] Sethuraman J, Alzubi JA, Manikandan R, et al. Eccentric methodology with optimization to unearth hidden facts of search engine result pages. Recent Patents Comput Sci. 2019;12(2):110–119. doi: 10.2174/2213275911666181115093050
[20] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE Congress on Evolutionary Computation. IEEE, p. 4661–4667.
[21] Li Y, Yang Z, Wang L. A hybrid imperialist competitive algorithm for energy-efficient flexible job shop scheduling problem with variable-size sublots. Comput Ind Eng. 2022;172:Article ID 108641.
[22] Moayedi H, Gör M, Kok Foong L, et al. Imperialist competitive algorithm hybridized with multilayer perceptron to predict the load-settlement of square footing on layered soils. Measurement. 2021;172:Article ID 108837. doi: 10.1016/j.measurement.2020.108837
[23] Barkhoda W, Sheikhi H. Immigrant imperialist competitive algorithm to solve the multi-constraint node placement problem in target-based wireless sensor networks. Ad Hoc Netw. 2020;106:Article ID 102183. doi: 10.1016/j.adhoc.2020.102183
[24] Reisi N, Hadipour Lakmesari S, Mahmoodabadi M, et al. Optimum fuzzy control of human immunodeficiency virus type1 using an imperialist competitive algorithm. Inform Med Unlocked. 2019;16:Article ID 100241. doi: 10.1016/j.imu.2019.100241
[25] Arya Y. ICA assisted FTI$^{\lambda}$DN controller for AGC performance enrichment of interconnected reheat thermal power systems. J Ambient Intell Humaniz Comput. 2021;14(3):1919–1935. doi: 10.1007/s12652-021-03403-6
[26] Arya Y, Dahiya P, Çelik E, et al. AGC performance amelioration in multi-area interconnected thermal and thermal-hydro-gas power systems using a novel controller. Eng Sci Technol Int J. 2021;24(2):384–396.
[27] Aliniya Z, Mirroshandel SA, et al. A novel combinatorial merge-split approach for automatic clustering using imperialist competitive algorithm. Expert Syst Appl. 2019;117:243–266. doi: 10.1016/j.eswa.2018.09.050
[28] Duan J, Asteris PG, Nguyen H, et al. A novel artificial intelligence technique to predict compressive strength of recycled aggregate concrete using ICA-XGBoost model. Eng Comput-Germany. 2020;37(4):3329–3346. doi: 10.1007/s00366-020-01003-0
[29] Harandizadeh H, Armaghani DJ, Asteris PG, et al. TBM performance prediction developing a hybrid ANFIS-PNN predictive model optimized by imperialism competitive algorithm. Neural Comput Appl. 2021;33(23):16149–16179. doi: 10.1007/s00521-021-06217-x
[30] Afonso LD, Mariani VC, Dos Santos Coelho L. Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability–redundancy optimization. Expert Syst Appl. 2013;40(9):3794–3802. doi: 10.1016/j.eswa.2012.12.093
[31] Bernal E, Castillo O, Soria J, et al. Interval type-2 fuzzy logic for dynamic parameter adjustment in the imperialist competitive algorithm. In: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), p. 1–5.
[32] Abdechiri M, Faez K, Bahrami H. Adaptive imperialist competitive algorithm (aica). In: 9th IEEE International Conference on Cognitive Informatics (ICCI'10), IEEE, p. 940–945.
[33] Khaled AA, Hosseini S. Fuzzy adaptive imperialist competitive algorithm for global optimization. Neural Comput Appl. 2015;26(4):813–825. doi: 10.1007/s00521-014-1752-4
[34] Rabiee A, Sadeghi M, Aghaei J. Modified imperialist competitive algorithm for environmental constrained energy management of microgrids. J Clean Prod. 2018;202:273–292. doi: 10.1016/j.jclepro.2018.08.129
[35] Idoumghar L, Chérin N, Siarry P, et al. Hybrid ICA-PSO algorithm for continuous optimization. Appl Math Comput. 2013;219(24):11149–11170.
[36] Sadhu AK, Rakshit P, Konar A. A modified imperialist competitive algorithm for multi-robot stick-carrying application. Robot Auton Syst. 2016;76:15–35. doi: 10.1016/j.robot.2015.11.010
[37] Chen CH, Chen WH. Bare-bones imperialist competitive algorithm for a compensatory neural fuzzy controller. Neurocomputing. 2016;173:1519–1528. doi: 10.1016/j.neucom.2015.09.025
[38] Peri D. Hybridization of the imperialist competitive algorithm and local search with application to ship design optimization. Comput Ind Eng. 2019;137:Article ID 106069. doi: 10.1016/j.cie.2019.106069
[39] Zandieh M, Khatami A, Rahmati SHA. Flexible job shop scheduling under condition-based maintenance: improved version of imperialist competitive algorithm. Appl Soft Comput. 2017;58:449–464. doi: 10.1016/j.asoc.2017.04.060
[40] Karimi S, Ardalan Z, Naderi B, et al. Scheduling flexible job-shops with transportation times: mathematical models and a hybrid imperialist competitive algorithm. Appl Math Model. 2017;41:667–682. doi: 10.1016/j.apm.2016.09.022
[41] Sadeghi J, Mousavi SM, Niaki STA. Optimizing an inventory model with fuzzy demand, backordering, and discount using a hybrid imperialist competitive algorithm. Appl Math Model. 2016;40(15-16):7318–7335. doi: 10.1016/j.apm.2016.03.013
[42] Moazzami M, Morshed MJ, Fekih A. A new optimal unified power flow controller placement and load shedding coordination approach using the hybrid imperialist competitive algorithm-pattern search method for voltage collapse prevention in power system. Int J Elec Power. 2016;79:263–274. doi: 10.1016/j.ijepes.2016.01.022
[43] Zhou T, Zhang Q, Wang X, et al. Imperialist competitive algorithm based on vnsobl optimization for distributed parallel machine scheduling problem. In: 2019 Chinese Automation Congress (CAC), p. 5717–5723.
[44] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition-based differential evolution. IEEE Trans Evol Comput. 2008;12(1):64–79. doi: 10.1109/TEVC.2007.894200

[45] Tizhoosh H. Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vol. 1. p. 695–701.
[46] Yang Y, Gao Y, Tan S, et al. An opposition learning and spiral modelling based arithmetic optimization algorithm for global continuous optimization problems. Eng Appl Artif Intel. 2022;113:Article ID 104981.
[47] Yang Y, Qian C, Li H, et al. An efficient dbscan optimized by arithmetic optimization algorithm with opposition-based learning. J Supercomput. 2022;78(18): 19566–19604. doi: 10.1007/s11227-022-04634-w
[48] Yildiz BS, Pholdee N, Mehta P, et al. A novel hybrid flow direction optimizer-dynamic oppositional based learning algorithm for solving complex constrained mechanical design problems. Mater Test. 2023;65(1):134–143. doi: 10.1515/mt-2022-0183
[49] Mehta P, Sultan Yıldız B, Pholdee N, et al. A novel generalized normal distribution optimizer with elite oppositional based learning for optimization of mechanical engineering problems. Mater Test. 2023;65(2):210–223. doi: 10.1515/mt-2022-0259
[50] Mehta P, Yildiz BS, Sait SM, et al. A novel hybrid Fick's law algorithm-quasi oppositional-based learning algorithm for solving constrained mechanical design problems. Mater Test. 2023;65(12):1817–1825. doi: 10.1515/mt-2023-0235
[51] Yildirim A, Demirci E, Karagöz S, et al. Experimental and numerical investigation of crashworthiness performance for optimal automobile structures using response surface methodology and oppositional based learning differential evolution algorithm. Mater Test. 2023;65(3):346–363. doi: 10.1515/mt-2022-0304
[52] Cao Z, Chen H. Opposition-based grey wolf optimization approaches for feature selection. In: 2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), p. 431–438.
[53] Houssein EH, Saber E, Ali AA, et al. Opposition-based learning tunicate swarm algorithm for biomedical classification. In: 2021 17th International Computer Engineering Conference (ICENCO), p. 1–6.
[54] Rahnamayan S, Tizhoosh HR, Salama MM. Quasi-oppositional differential evolution. In: 2007 IEEE Congress on Evolutionary Computation. p. 2229–2236.
[55] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl. 2016;27(4):1053–1073. doi: 10.1007/s00521-015-1920-1
[56] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: A gravitational search algorithm. Inform Sci. 2009;179(13): 2232–2248. doi: 10.1016/j.ins.2009.03.004
[57] Yu K, Qu B, Yue C, et al. A performance-guided JAYA algorithm for parameters identification of photovoltaic cell and module. Appl Energ. 2019;237:241–257. doi: 10.1016/j.apenergy.2019.01.008
[58] Faramarzi A, Heidarinejad M, Mirjalili S, et al. Marine predators algorithm: A nature-inspired metaheuristic. Expert Syst Appl. 2020;152:Article ID 113377. doi: 10.1016/j.eswa.2020.113377
[59] Gandomi AH, Yang XS, Talatahari S, et al. Firefly algorithm with chaos. Commun Nonlinear Sci. 2013;18(1):89–98. doi: 10.1016/j.cnsns.2012.06.009
[60] Talatahari S, Azizi M. Optimization of constrained mathematical and engineering design problems using chaos game optimization. Comput Ind Eng. 2020;145: Article ID 106560. doi: 10.1016/j.cie.2020.106560
[61] Chen H, Xu Y, Wang M, et al. A balanced whale optimization algorithm for constrained engineering design problems. Appl Math Model. 2019;71:45–59. doi: 10.1016/j.apm.2019.02.004
[62] Pu SA, Hao LB, Yong ZA, et al. An intensify atom search optimization for engineering design problems. Appl Math Model. 2021;89:837–859. doi: 10.1016/j.apm.2020.07.052
[63] Braik MS. Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. Expert Syst Appl. 2021;174(1):Article ID 114685.
[64] Zhang Y, Chi A, Mirjalili S. Enhanced Jaya algorithm: a simple but efficient optimization method for constrained engineering design problems. Knowl-based Syst. 2021;233:107555). doi: 10.1016/j.knosys.2021.107555
[65] Bayzidi H, Talatahari S, Saraee M, et al. Social network search for solving engineering optimization problems. Comput Intell Neurosci. 2021:Article ID 8548639. doi: 10.1155/2021/8548639