Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 417 (2023) 116446
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Quadratic Interpolation Optimization (QIO): A new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems

Weiguo Zhao[a], Liying Wang[a,*], Zhenxing Zhang[b], Seyedali Mirjalili[c,d], Nima Khodadadi[e], Qiang Ge[f]

[a] *School of Water Conservancy and Hydropower, Hebei University of Engineering, Handan, Hebei, 056038, China*
[b] *Prairie Research Institute, University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA*
[c] *Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Fortitude Valley, Brisbane, 4006, QLD, Australia*
[d] *YFL (Yonsei Frontier Lab), Yonsei University, Seoul, Republic of Korea*
[e] *Department of Civil and Architectural Engineering, University of Miami, Coral Gables, FL, USA*
[f] *School of Earth and Space Sciences, Peking University, Beijing, 100871, China*

Received 14 July 2023; received in revised form 22 August 2023; accepted 8 September 2023
Available online xxxx

Abstract

An original math-inspired meta-heuristic algorithm, named quadratic interpolation optimization (QIO), is proposed to address numerical optimization and engineering issues. The main inspiration behind QIO is derived from mathematics, specifically the newly proposed generalized quadratic interpolation (GQI) method. This method overcomes the limitations of the traditional quadratic interpolation method to better find the minimizer of the quadratic function formed by any three points. The QIO utilizes the GQI method as a promising searching mechanism for tackling various types of optimization problems. This searching mechanism delivers exploration and exploitation strategies, in which the minimizer provided by the GQI method assists the QIO algorithm in exploring a promising region in unexplored areas and exploit the optimal solutions in promising regions. To evaluate QIO's effectiveness, it is comprehensively compared with 12 other commonly used optimizers on 23 benchmark test functions and the CEC-2014 test suite. Ten engineering problems are also tested to assess QIO's practicality. Eventually, a real-world application of QIO is presented in the operation management of a microgrid with an energy storage system. The results demonstrate that QIO is a promising alternative for addressing practical challenges. The source code of QIO is publicly available at https://ww2.mathworks.cn/matlabcentral/fileexchange/135627-quadratic-interpolation-optimization-qio.

Keywords: Meta-heuristic; Optimization; Swarm intelligence; Engineering optimization; Microgrid

1. Introduction

Metaheuristics refer to techniques specifically designed to effectively and efficiently solve computationally challenging optimization problems, mainly when dealing with incomplete or imperfect information or limited

* Corresponding author.
E-mail addresses: zhaoweiguo@hebeu.edu.cn (W. Zhao), wangliying@hebeu.edu.cn (L. Wang), zhang538@illinois.edu (Z. Zhang).

https://doi.org/10.1016/j.cma.2023.116446
0045-7825/

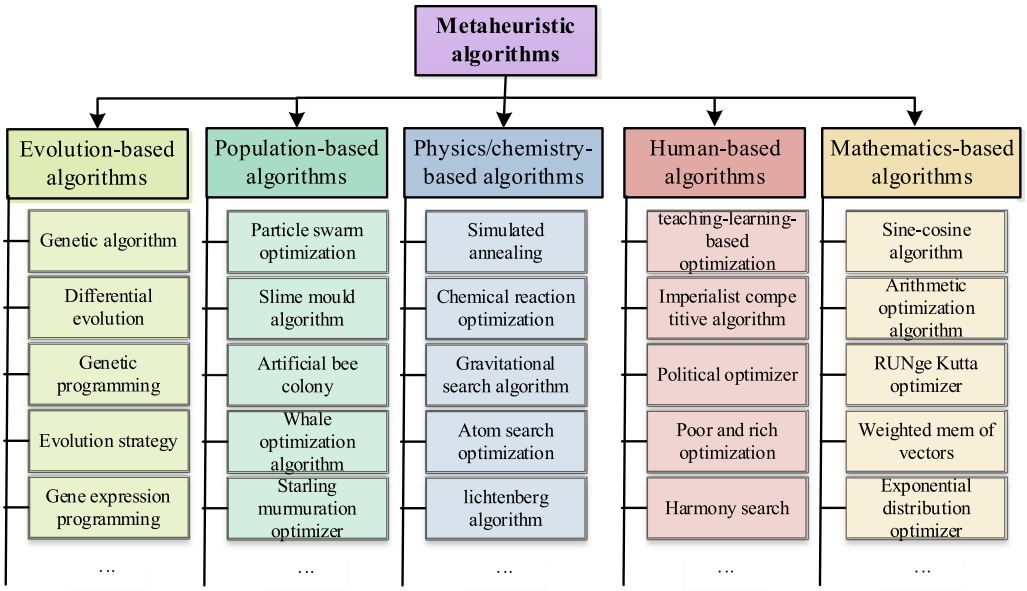

Fig. 1. A classification of metaheuristics.

computational resources. They are high-level procedures or heuristics that can discover, generate, adjust, or select strategies that can yield satisfactory solutions to optimization problems. Inspired by natural phenomena, social behaviors, or physical processes, metaheuristics offer several advantages, such as:

(1) Global search: metaheuristics excel in searching the entire solution space, enabling them to discover the global optimal solution to the problem [1].
(2) Scalability: metaheuristics can tackle various types of problems, including high-dimensional, nonlinear, and discretized problems [2].
(3) Diversity: metaheuristics can explore the search space extensively, resulting in the generation of a diverse set of candidate solutions [3].
(4) Adaptability: metaheuristics can be easily adapted and customized for specific problem domains, allowing for greater flexibility [4].
(5) Computational efficiency: metaheuristics offer notable computational efficiency, often requiring fewer computational resources compared to exact optimization methods, making them practical for tackling large-scale optimization problems [5].

Based on the above advantages, metaheuristics have been successfully applied in a large range of areas such as industry [6,7], agriculture [8,9], energy [10,11], healthcare [12,13], material [14,15], aerospace [16,17], finance [18,19], mechanical engineering [20,21], and communication [22,23].

In this review, we will discuss the development and classification of metaheuristics, with a focus on five categories [24]: evolution-based metaheuristics (EMs), population-based metaheuristics (PMs), physics/chemistry-based metaheuristics (PCMs), and human-based metaheuristics (HMs). Fig. 1 illustrates the classification of metaheuristics.

EMs are the earliest developed metaheuristic algorithms, which are inspired from process of natural selection and genetics principles. These algorithms involve the creation of a population of potential solutions and iteratively applying specific operators to generate new candidate solutions. One prominent EM is the genetic algorithm (GA) [25], which mimics Darwin's natural selection process. In GA, a population of candidate solutions is created, and several genetic operators, including selection, mutation, and recombination, are employed to generate new candidate solutions. The fitness of each candidate solution is evaluated, and the fittest solutions are chosen for the next generation. Another well-known EM is differential evolution (DE), which is based on the concept of mutation and

recombination of individuals in a population. DE and GA differ in their emphasis on operators, with DE focusing on the mutation operator and GA prioritizing the crossover operator. DE is often considered more straightforward and more efficient than GA [26]. In addition, there are several other EMs that have been developed, including genetic programming (GP) [27], evolution strategy (ES) [28], self-organizing migration algorithm (SOMA) [29], gene expression programming (GEP) [30], and biogeography-based optimization (BBO) [31].

PMs, as a major family and branch of metaheuristics, draw inspiration from the collective behavior observed in social animals such as bees, ants, birds, and fish. Such algorithms simulate the behavior of a swarm or group of agents, where each agent represents a candidate solution to an optimization problem. The agents interact with each other and their environment to search for better solutions through group behaviors such as communication, cooperation, and competition.

Particle swarm optimization (PSO) [32] is a classical PM, which simulates the behavior of a swarm of birds or fish abstracted as particles. Each particle represents a candidate solution, and the algorithm searches for the optimum by adjusting the velocity and position of each particle based on its own and its neighbors' best positions. Another popular PM is artificial bee colony (ABC) [33]. ABC simulates the foraging behavior of a bee colony using a population of solution vectors, with three types of bees (employed, onlooker, and scout bees) performing different tasks. The employed bees evaluate their solutions and share information with the onlooker bees, who select the most promising solutions to evaluate. The scout bees randomly search unexplored regions of the search space. As a new and popular PM, the whale optimization algorithm (WOA) is worth mentioning [34]. WOA imitates humpback whales' hunting behaviors to establish search strategies: (1) encircling prey: this behavior involves updating the positions of candidate solutions towards the best solution found so far; (2) bubble-net feeding: this behavior involves modifying positions randomly to explore new regions of the search space; and (3) searching for prey: this behavior involves exploiting the best solution found so far. Some other PMs are slime mould algorithm (SMA) [35], starling murmuration optimizer (SMO) [36], white shark optimizer (WSO) [37], gazelle optimization algorithm (GOA) [38], honey badger algorithm (HBA) [39], coati optimization algorithm (COA) [40], artificial hummingbird algorithm (AHA) [41], butterfly optimization algorithm (BOA) [42], polar bear optimization (PBO) [43], red fox optimization (RFO) [44], and nutcracker optimizer (NO) [45].

The third category of metaheuristics is PCMs, which are inspired by physical phenomena such as attraction, repulsion, and gravity, as well as chemical processes such as chemical reactions and molecular interactions. Simulated Annealing (SA) [46] is a popular PCM, which takes inspiration from the physical process of annealing. It begins with an initial solution and progressively improves it progressively by accepting suboptimal solutions according to a temperature-dependent probability distribution. The algorithm converges to a near-optimal solution by lowering the temperature gradually. Chemical reaction optimization (CRO) is another representative PCM that imitates chemical reactions [47]. In CRO, candidate solutions are represented as molecules, and finding the optimal solution involves simulating chemical reactions between these molecules. The fitness of each molecule corresponds to the objective function value of the corresponding solution, and the algorithm uses various types of chemical reactions such as diffusion, transformation, and degradation to explore and reach the optimal solution. Some other PCMs include gravitational search algorithm (GSA) [48], atom search optimization (ASO), [49], Lichtenberg algorithm (LA) [50], multiverse optimizer (MVO) [51], Young double-slit experiment optimizer (YDSE) [52], artificial physics algorithm (APA) [53], kinetic gas molecules optimization (KGMO) [54], Henry gas solubility optimization (HGSO) [55], optics-inspired optimization (OIO) [56], integrated radiation algorithm (IRA) [57], thermal exchange optimization (TEO) [58], charged system search (CSS) [59], artificial chemical reaction optimization algorithm (ACROA) [60], archimedes optimization algorithm (AOA) [61], and energy valley optimizer (EVO) [62].

The fourth category of metaheuristics is HMs, which are designed to emulate or incorporate human intelligence, experience, and expertise in the optimization process. These algorithms are typically inspired by the way humans approach problem-solving, decision-making, and learning, and they often rely on various cognitive or psychological models to guide the search process. Teaching and learning-based optimization (TLBO) [63] is a well-designed HM, which simulates the teaching and learning processes among students in a classroom setting. The algorithm is inspired by the pedagogical concept of "learning from teachers" and "learning from peers", which involves collaboration and knowledge sharing among individuals with different levels of knowledge and expertise. Other common HMs are imperialist competitive algorithm (ICA) [64], political optimizer (PO) [65], human eye vision algorithm (HEVA) [66], harmony search (HS) [67], poor and rich optimization (PRO) [68], soccer league competition (SLC) [69],

brain storm optimization (BSO) [70], sewing training-based optimization (STBO) [71], exchange market algorithm [72], and league championship algorithm (LCA) [73].

As an emerging branch of metaheuristics, MMs have shown great promise in enhancing the efficiency and effectiveness of optimization approaches. MMs are based on a wide range of mathematical and algorithmic foundations, including certain mathematical functions, rules, formulas, and theories. Sine–cosine algorithm (SCA) [74] is a well-known MM, which draws inspiration from the mathematical properties of the sine and cosine functions. The basic idea behind SCA is to use trigonometric functions to explore search space and converge to optimal solutions. Other examples belonging to this category are: arithmetic optimization algorithm (AOA) [75], Runge–Kutta optimizer (RUN) [76], exponential distribution optimizer (EDO) [77], weighted mem of vectors (INFO) [78], golden sine algorithm (GSA) [79], and circle search algorithm (CSA) [80].

Despite numerous metaheuristics, the question is why it is still necessary to develop new optimization algorithms [81]. While existing optimization algorithms may be effective at dealing with certain types of problems, they may not be suitable for others or may suffer from limitations such as premature convergence and slow convergence speed. Therefore, there is a constant need to develop new optimization algorithms to overcome these limitations and outperform existing algorithms. With the rapid development of various industries, numerous challenging optimization problems are emerging constantly. Existing optimization techniques may not be able to satisfactorily solve these problems, which calls for the development of new optimization technologies to address these issues. A significant number of different strategies have been developed to enhance the performance of the existing algorithms when solving various engineering problems. However, the strategies improving the performance of the existing algorithms are nearing their limits, and the development speed has slowed down. Developing new optimization algorithms can alleviate this stagnation and provide new sources and inspiration for improving existing optimization algorithms. In addition, numerous existing optimization algorithms possess their own set of merits and limitations, which furnish valuable insights and aid in the development of more resilient optimization algorithms. By introducing novel ideas and technologies, scholars can engineer swifter and more efficient optimization algorithms that excel in adapting to diverse and intricate tasks. These are the main motivations of the current study.

In this paper, a new optimization approach called QIO based on mathematics is designed. In the QIO algorithm, the GQI method that can obtain the minimizer of any quadratic function is proposed, and two powerful search strategies are offered. In the exploration strategy, for each dimension, the minimizer of the quadratic interpolation function formed by the positions of two individuals randomly chosen from the population and the current individual is obtained using the GQI method, and the obtained minimizer and the position of another individual randomly chosen from the population are combined to produce a new position for performing a global search. In the exploitation strategy, for each dimension, the minimizer of the quadratic interpolation function formed by the positions of two individuals randomly chosen from the population and the best individual found so far is obtained using the GQI method, and the obtained minimizer and the position of the best individual found so far are combined to produce a new position for performing a local search. Furthermore, the QIO algorithm is tested using 23 benchmark functions and the CEC-2014 test suite, and, ten engineering problems and the application of operation management of a microgrid with an energy storage system are employed to test its ability to solve real-world problems.

The main contributions of this study are summarized as follows, along with an explanation of the optimization performance of QIO.

- A generalized quadratic interpolation (GQI) method is developed for computing the minimizer of the quadratic interpolation function formed by any three points. This method overcomes the positional limitation of the three points in the traditional quadratic interpolation method. The GQI method is analyzed thoroughly and mathematically expressed in detail.
- A new optimizer named quadratic interpolation optimization (QIO) is proposed. It introduces two search strategies based on the GQI method, the exploration strategy and the exploitation strategy. These strategies are designed to improve the overall performance of the algorithm.
- The exploration strategy utilizes the minimizer of the GQI method and the randomly chosen individuals to perform a global search, while the exploitation strategy utilizes the minimizer of the GQI method, and the best individual found so far to perform a local search.
- The effectiveness of QIO has been confirmed through its performance on 23 benchmark problems and the CEC-2014 test suite.

- Three statistical analyses, including the Wilcoxon test, Friedman aligned test, and Quade rank, are used to evaluate the effectiveness of QIO and compare it with 12 established metaheuristic algorithms.
- The practicality of QIO is assessed through its application to 10 classical engineering cases as well as in the operation management of microgrid operation with an energy storage system.

The remaining sections of this study are structured as follows: in Section 2, the GQI method is introduced, along with a detailed description of the QIO algorithm. In Section 3, two sets of experiments are conducted on the 23 benchmark problems and the CEC-2014 test suite. Section 4 presents QIO's efficiency in addressing 10 engineering applications, while Section 5 showcases its practicality in engineering practices through the operation management of microgrid operation with an energy storage system. Lastly, Section 6 summarizes our conclusions and outlines future directions for further research and development.

2. Proposed algorithm

2.1. Quadratic interpolation method

The quadratic interpolation method is a commonly used technique for curve fitting [82,83], which is often used for unary functions to search for the minimum point within a given definite initial interval. In the process of finding the minimum point of a function $f(x)$, a quadratic interpolation polynomial, denoted as $L(x)$, is used to approximate the function $f(x)$. By finding the minimizer of the polynomial $L(x)$, the approximate minimum point of the function $f(x)$ can be obtained.

The primary formulation of quadratic interpolation is described as follows:

$L(x)$, the quadratic interpolant of $f(x)$, can be expressed as:

$$L(x) = \alpha x^2 + \beta x + \delta \qquad \alpha, \beta, \delta \in R \tag{1}$$

where α, β, and δ are undetermined coefficients. Assume $f(x)$ has three points $P_i(x_i, f(x_i))$, $P_j(x_j, f(x_j))$, and $P_k(x_k, f(x_k))$, let $a \le x_i < x_j < x_k \le b$. According to the interpolation conditions, the value of $f(x)$ is equal to that of $L(x)$ at the interpolation points x_i, x_j and x_k, which is expressed as:

$$\begin{cases} L(x_i) = \alpha x_i^2 + \beta x_i + \delta = f(x_i) \\ L(x_j) = \alpha x_j^2 + \beta x_j + \delta = f(x_j) \\ L(x_k) = \alpha x_k^2 + \beta x_k + \delta = f(x_k) \end{cases} \tag{2}$$

To find the minimum of $L(x)$, take the derivative of Eq. (1) and set it to 0. The minimizer x^* can be obtained by:

$$x^* = -\frac{\beta}{2\alpha} \tag{3}$$

Then α and β can be determined by solving Eq. (4):

$$\begin{cases} \alpha = \dfrac{(x_j - x_k) f(x_i) + (x_k - x_i) f(x_j) + (x_i - x_j) f(x_k)}{(x_j - x_k)(x_k - x_i)(x_i - x_j)} \\ \beta = -\dfrac{(x_j^2 - x_k^2) f(x_i) + (x_k^2 - x_i^2) f(x_j) + (x_i^2 - x_j^2) f(x_k)}{(x_j - x_k)(x_k - x_i)(x_i - x_j)} \end{cases} \tag{4}$$

According to Eq. (5), the minimizer of $L(x)$ is:

$$x^* = \frac{(x_j^2 - x_k^2) f(x_i) + (x_k^2 - x_i^2) f(x_j) + (x_i^2 - x_j^2) f(x_k)}{2((x_j - x_k) f(x_i) + (x_k - x_i) f(x_j) + (x_i - x_j) f(x_k))} \tag{5}$$

Meanwhile, the quadratic interpolation polynomial $L(x)$ can be easily obtained as:

$$L(x) = \frac{(x - x_j)(x - x_k)}{(x_i - x_j)(x_i - x_k)} f(x_i) + \frac{(x - x_i)(x - x_k)}{(x_j - x_i)(x_j - x_k)} f(x_j) + \frac{(x - x_i)(x - x_j)}{(x_k - x_i)(x_k - x_j)} f(x_k) \tag{6}$$

The schematic of quadratic interpolation is provided in Fig. 2.

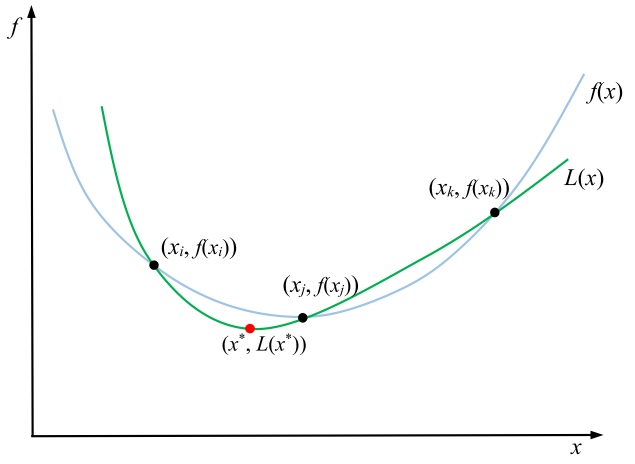

Fig. 2. Schematic of quadratic interpolation.

By narrowing the search interval based on the previously estimated minimum point, x^*, a new search interval is formed with three points. The quadratic interpolation is then applied again using the aforementioned technique until the specified precision requirement is met. This iterative process allows for finding the final approximate minimum point.

2.2. Generalized quadratic interpolation (GQI)

Quadratic interpolation is widely employed to approximate the minimum of $f(x)$. In the field of optimization, it is frequently integrated into an enormous variety of optimization techniques to enhance their search performance [84–87]. However, in practical applications, when dealing with a minimization problem, there are no specific requirements for the selection of three points. However, when the points are chosen improperly, a parabola opening upward may be constructed, leading to a maximizer rather than a minimizer. If there is a specific requirement for the selection of three points to establish a parabola opening downward, this limitation poses challenges in terms of time consumption and effectiveness. To address this issue and meet the demand for a more universal approach, a generalized quadratic interpolation (GQI) method is proposed.

Firstly, a comprehensive analysis is conducted for the quadratic interpolation method, considering all possible cases of three points. For $f(x_i) < f(x_j) < f(x_k)$, there are 6 cases.

(1) When $x_j < x_i < x_k$, if $f(x_j) \ll f(x_k)$, then

$$x^* = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_k)} + (x_k^2 - x_i^2)\frac{f(x_j)}{f(x_k)} + (x_i^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_k)} + (x_k - x_i)\frac{f(x_j)}{f(x_k)} + (x_i - x_j))} = \frac{(x_i^2 - x_j^2)}{2(x_i - x_j)} = \frac{x_i + x_j}{2} \tag{7}$$

If $f(x_j) \to f(x_k)-$, then

$$\begin{aligned} x^* &= \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_i^2) + (x_i^2 - x_j^2)\frac{f(x_k)}{f(x_j)}}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_i) + (x_i - x_j)\frac{f(x_k)}{f(x_j)})} = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_j))} \\ &= \frac{(x_k^2 - x_j^2)(\frac{f(x_i)}{f(x_j)} - 1)}{2(x_k - x_j)(\frac{f(x_i)}{f(x_j)} - 1)} = \frac{x_k^2 - x_j^2}{2(x_k - x_j)} = \frac{x_j + x_k}{2} \end{aligned} \tag{8}$$

So, the minimizers $= \left\{x^* | \frac{x_i + x_j}{2} \le x^* \le \frac{x_j + x_k}{2}\right\}$ (Corresponding to Fig. 3(a)).

(2) When $x_k < x_i < x_j$, if $f(x_j) \ll f(x_k)$, then

$$x^* = \frac{(x_i^2 - x_j^2) + (x_j^2 - x_k^2)\frac{f(x_i)}{f(x_k)} + (x_k^2 - x_i^2)\frac{f(x_j)}{f(x_k)}}{2((x_i - x_j) + (x_j - x_k)\frac{f(x_i)}{f(x_k)} + (x_k - x_i)\frac{f(x_j)}{f(x_k)})} = \frac{(x_i^2 - x_j^2)}{2(x_i - x_j)} = \frac{x_i + x_j}{2} \tag{9}$$

If $f(x_j) \to f(x_k)-$, then

$$x^* = \frac{(x_i^2 - x_j^2)\frac{f(x_k)}{f(x_j)} + (x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_i^2)}{2((x_i - x_j)\frac{f(x_k)}{f(x_j)} + (x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_i))} = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_j))}$$
$$= \frac{(x_j^2 - x_k^2)(\frac{f(x_i)}{f(x_j)} - 1)}{2(x_j - x_k)(\frac{f(x_i)}{f(x_j)} - 1)} = \frac{x_j^2 - x_k^2}{2(x_j - x_k)} = \frac{x_j + x_k}{2} \tag{10}$$

So, the minimizers $= \left\{x^* | \frac{x_j + x_k}{2} \le x^* \le \frac{x_i + x_j}{2}\right\}$ (Corresponding to Fig. 3(b)).

(3) When $x_i < x_j < x_k$, if $f(x_j) \ll f(x_k)$, then

$$x^* = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_k)} + (x_k^2 - x_i^2)\frac{f(x_j)}{f(x_k)} + (x_i^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_k)} + (x_k - x_i)\frac{f(x_j)}{f(x_k)} + (x_i - x_j))} = \frac{x_i^2 - x_j^2}{2(x_i - x_j)} = \frac{x_i + x_j}{2} \tag{11}$$

If $f(x_j) \to f(x_k)-$, then

$$x^* = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_i^2) + (x_i^2 - x_j^2)\frac{f(x_k)}{f(x_j)}}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_i) + (x_i - x_j)\frac{f(x_k)}{f(x_j)})} = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_j))}$$
$$= \frac{(x_j^2 - x_k^2)(\frac{f(x_i)}{f(x_j)} - 1)}{2(x_j - x_k)(\frac{f(x_i)}{f(x_j)} - 1)} = = \frac{x_j^2 - x_k^2}{2(x_j - x_k)} = \frac{x_j + x_k}{2} \tag{12}$$

So, the minimizers $= \left\{x^* | x^* \le \frac{x_i + x_j}{2}\right\}$ and the maximizers $= \left\{x^* | x^* \ge \frac{x_j + x_k}{2}\right\}$ (Corresponding to Fig. 3(c)).

(4) When $x_k < x_j < x_i$, if $f(x_j) \ll f(x_k)$, then

$$x^* = \frac{(x_i^2 - x_j^2) + (x_k^2 - x_i^2)\frac{f(x_j)}{f(x_k)} + (x_j^2 - x_k^2)\frac{f(x_i)}{f(x_k)}}{2((x_i - x_j) + (x_k - x_i)\frac{f(x_j)}{f(x_k)} + (x_j - x_k)\frac{f(x_i)}{f(x_k)})} = \frac{x_i^2 - x_j^2}{2(x_i - x_j)} = \frac{x_i + x_j}{2} \tag{13}$$

If $f(x_j) \to f(x_k)-$, then

$$x^* = \frac{(x_i^2 - x_j^2)\frac{f(x_k)}{f(x_j)} + (x_k^2 - x_i^2) + (x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)}}{2((x_i - x_j)\frac{f(x_k)}{f(x_j)} + (x_k - x_i) + (x_j - x_k)\frac{f(x_i)}{f(x_j)})} = \frac{(x_k^2 - x_j^2)\frac{f(x_i)}{f(x_j)} + (x_j^2 - x_k^2)}{2((x_k - x_j)\frac{f(x_i)}{f(x_j)} + (x_j - x_k))}$$
$$= \frac{(x_j^2 - x_k^2)(\frac{f(x_i)}{f(x_j)} - 1)}{2(x_j - x_k)(\frac{f(x_i)}{f(x_j)} - 1)} = = \frac{x_j^2 - x_k^2}{2(x_j - x_k)} = \frac{x_j + x_k}{2} \tag{14}$$

So, the minimizers $= \left\{x^* | x^* \le \frac{x_j + x_k}{2}\right\}$ and the maximizers $= \left\{x^* | x^* \ge \frac{x_i + x_j}{2}\right\}$. (Corresponding to Fig. 2(d)).

(5) When $x_i < x_k < x_j$, if $f(x_j) \ll f(x_k)$, then

$$x^* = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_k)} + (x_i^2 - x_j^2) + (x_k^2 - x_i^2)\frac{f(x_j)}{f(x_k)}}{2((x_j - x_k)\frac{f(x_i)}{f(x_k)} + (x_i - x_j) + (x_k - x_i)\frac{f(x_j)}{f(x_k)})} = \frac{x_i^2 - x_j^2}{2(x_i - x_j)} = \frac{x_i + x_j}{2} \tag{15}$$

If $f(x_j) \to f(x_k)-$, then

$$x^* = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_i^2 - x_j^2)\frac{f(x_k)}{f(x_j)} + (x_k^2 - x_i^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_i - x_j)\frac{f(x_k)}{f(x_j)} + (x_k - x_i))} = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_j))}$$
$$= \frac{(x_j^2 - x_k^2)(\frac{f(x_i)}{f(x_k)} - 1)}{2(x_j - x_k)(\frac{f(x_i)}{f(x_k)} - 1)} = = \frac{x_j^2 - x_k^2}{2(x_j - x_k)} = \frac{x_j + x_k}{2} \tag{16}$$

So, the maximizers $= \left\{x^* | \frac{x_i + x_j}{2} \le x^* \le \frac{x_j + x_k}{2}\right\}$. (Corresponding to Fig. 2(e)).

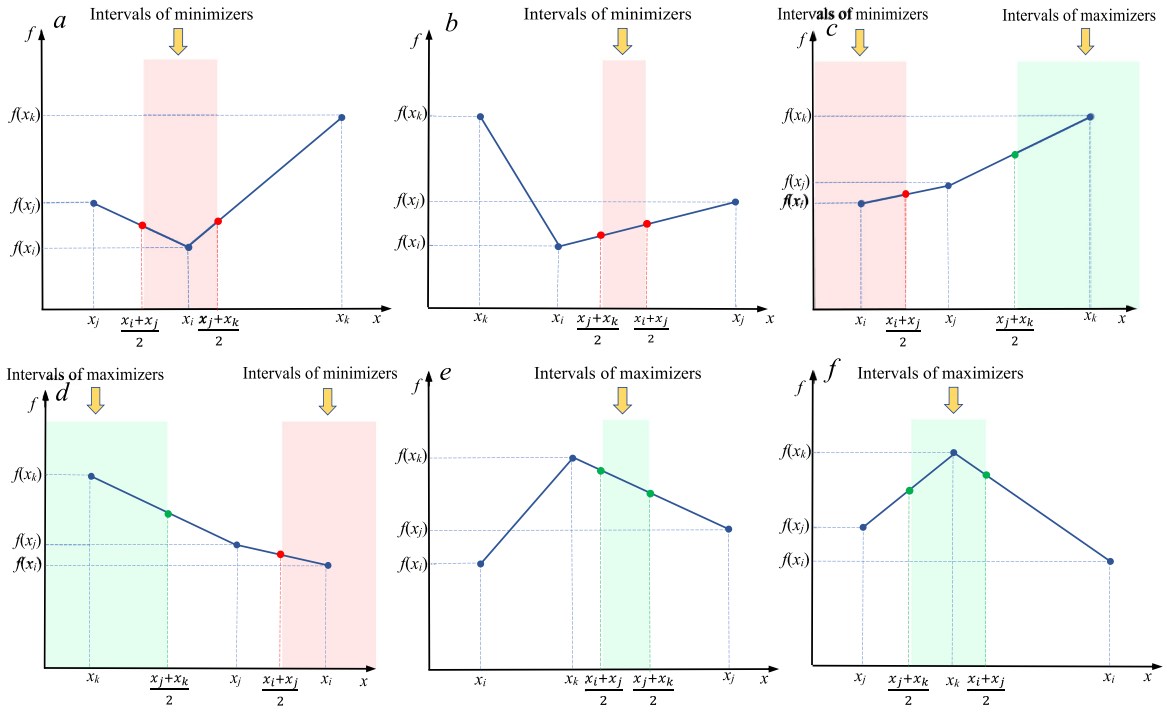

Fig. 3. Minimizer or maximizer intervals of quadratic interpolation for different cases.

(6) When $x_j < x_k < x_i$, if $f(x_j) \ll f(x_k)$, then

$$x^* = \frac{(x_k^2 - x_i^2)\frac{f(x_j)}{f(x_k)} + (x_i^2 - x_j^2) + (x_j^2 - x_k^2)\frac{f(x_i)}{f(x_k)}}{2((x_k - x_i)\frac{f(x_j)}{f(x_k)} + (x_i - x_j) + (x_j - x_k)\frac{f(x_i)}{f(x_k)})} = \frac{x_i^2 - x_j^2}{2(x_i - x_j)} = \frac{x_i + x_j}{2} \tag{17}$$

If $f(x_j) \to f(x_k)-$, then

$$\begin{aligned} x^* &= \frac{(x_k^2 - x_i^2) + (x_i^2 - x_j^2)\frac{f(x_k)}{f(x_j)} + (x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)}}{2((x_k - x_i) + (x_i - x_j)\frac{f(x_k)}{f(x_j)} + (x_j - x_k)\frac{f(x_i)}{f(x_j)})} = \frac{(x_j^2 - x_k^2)\frac{f(x_i)}{f(x_j)} + (x_k^2 - x_j^2)}{2((x_j - x_k)\frac{f(x_i)}{f(x_j)} + (x_k - x_j))} \\ &= \frac{(x_j^2 - x_k^2)(\frac{f(x_i)}{f(x_j)} - 1)}{2(x_j - x_k)(\frac{f(x_i)}{f(x_j)} - 1)} == \frac{x_j^2 - x_k^2}{2(x_j - x_k)} = \frac{x_j + x_k}{2} \end{aligned} \tag{18}$$

So, the maximizers $= \left\{x^* | \frac{x_j + x_k}{2} \le x^* \le \frac{x_i + x_j}{2}\right\}$ (Corresponding to Fig. 2(f)).

Based on the above analysis, Fig. 2 depicts the intervals of the minimizers and maximizers for the quadratic interpolation method in the six cases, in which the intervals of the minimizers or maximizers are highlighted with different colors for each case. From Fig. 3(a) and (b), for $f(x_i) < f(x_j) < f(x_k)$, when $x_j < x_i < x_k$ or $x_k < x_i < x_j$, the minimizer of the corresponding quadratic interpolation function must exist and fall in the intervals of minimizers. From Fig. 3(c) and (d), for $f(x_i) < f(x_j) < f(x_k)$, when $x_i < x_j < x_k$ or $x_k < x_j < x_i$, the corresponding quadratic interpolation function has the minimizer or the maximizer. From Fig. 3(e) and (f), for $f(x_i) < f(x_j) < f(x_k)$, when $x_i < x_k < x_j$ or$x_j < x_k < x_i$, the corresponding quadratic interpolation function only has the maximizer rather than the minimizer. It is evident that the quadratic interpolation method cannot obtain the minimizer for all cases.

Therefore, when the quadratic interpolation method is used in optimization algorithms to address minimization problems, half of the computational time is dedicated to finding the maximizer of the interpolation function instead of the intended minimizer, which greatly increases the computational burden and diminishes the search performance. As a result, the direct application of quadratic interpolation method in optimization algorithms presents certain limitations. To address the limitations and enhance the search efficiency when applied in optimization algorithms,

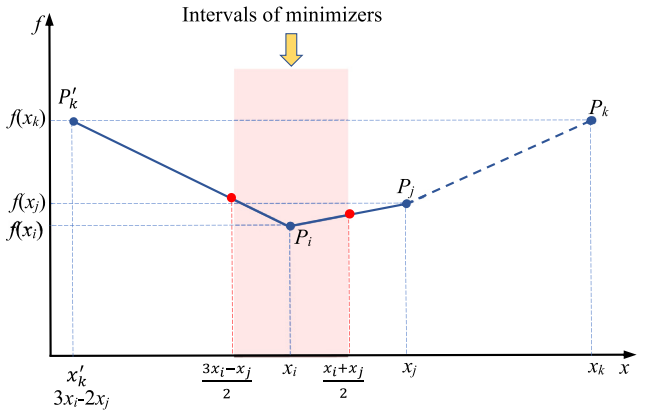

Fig. 4. Schematic for obtaining the minimizer interval in Fig. 3(c).

a generalized quadratic interpolation (GQI) method is developed, in which the interpolation functions of four cases for Fig. 3(c)–(f) are rebuilt.

For $f(x_i) < f(x_j) < f(x_k)$, when $x_i < x_j < x_k$, the minimizer is calculated by Eq. (19). If the value of the minimizer is less than x_j, the minimizer using Eq. (5) is accepted; otherwise, the quadratic interpolation function is rebuilt as shown in Fig. 4. For the case in Fig. 3(c), since the promising region to exploit the optimum solution for $f(x)$ may be around x_i, this region is considered the minimizer interval of the quadratic interpolation function. Thus, to make the minimizer fall on both sides of P_i, P_k is moved to the left of P_i along the x-axis, which makes the minimizer intervals of the quadratic interpolation function formed by P_j, P_i and P'_k, symmetrical about $x = x_i$ with its interval of $(x_j - x_i)$. Therefore, the minimizer interval of the quadratic interpolation function formed by P_i, P_j and P_k is provided by the quadratic interpolation function formed by P_i, P_j and P'_k. As depicted in Fig. 4, the minimizer for the case in Fig. 3(c) can be expressed as:

$$\begin{cases} \left.\begin{aligned} x' &= \frac{(x_j^2 - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \\ x^* &= x' \end{aligned}\right\} \text{if } x' < x_j \\ \left.\begin{aligned} x'_k &= 3x_i - 2x_j \\ x^* &= \frac{(x_j^2 - x_k'^2)f(x_i) + (x_k'^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x'_k)f(x_i) + (x'_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \end{aligned}\right\} \text{else} \end{cases} \tag{19}$$

For $f(x_i) < f(x_j) < f(x_k)$, when $x_k < x_j < x_i$, similar to the above manner, the minimizer interval of the quadratic interpolation function formed by P_i, P_j and P_k in Fig. 3(d) is provided by the quadratic interpolation function formed by P_i, P_j and P'_k in Fig. 5. Thus, the minimizer for the case in Fig. 3(d) can be expressed as:

$$\begin{cases} \left.\begin{aligned} x' &= \frac{(x_j^2 - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \\ x^* &= x' \end{aligned}\right\} \text{if } x' > x_j \\ \left.\begin{aligned} x'_k &= 3x_i - 2x_j \\ x^* &= \frac{(x_j^2 - x_k'^2)f(x_i) + (x_k'^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x'_k)f(x_i) + (x'_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \end{aligned}\right\} \text{else} \end{cases} \tag{20}$$

For $f(x_i) < f(x_j) < f(x_k)$, when $x_i < x_k < x_j$, in the case of Fig. 3(e), since the promising region to exploit the optimum solution for $f(x)$ may be around the left side of x_i, P_j is moved to the left of x_i along the x-axis, which makes the minimizer intervals of the quadratic interpolation function formed by P_i, P'_j and P_k that equals $\frac{x_k - x_i}{2}$.

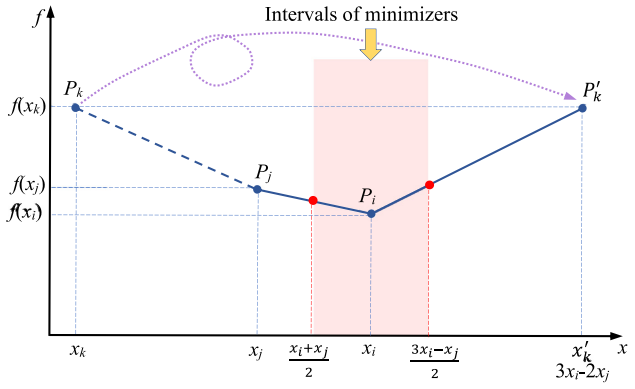

Fig. 5. Schematic for obtaining the minimizer interval in Fig. 3(d).

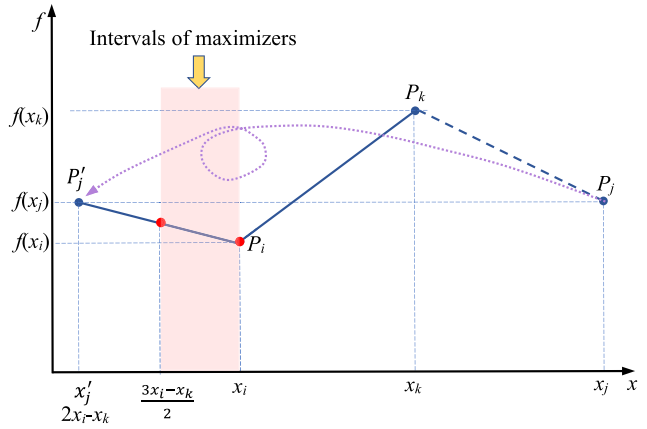

Fig. 6. Schematic for obtaining the minimizer interval in Fig. 3(e).

Therefore, the minimizer interval of the quadratic interpolation function formed by P_i, P_j and P_k in Fig. 3(e) is provided by the quadratic interpolation function formed by P_i, P'_j and P_k in Fig. 6. The minimizer in Fig. 6 for the case in Fig. 3(e) can be expressed as:

$$\begin{cases} x'_j = 2x_i - x_k \\ x^* = \dfrac{(x'^2_j - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - x'^2_j)f(x_k)}{2((x'_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x'_j)f(x_k))} \end{cases} \tag{21}$$

Likewise, for $f(x_i) < f(x_j) < f(x_k)$, when $x_j < x_k < x_i$, the minimizer interval of the quadratic interpolation function formed by P_i, P_j and P_k in Fig. 3(f) is provided by the quadratic interpolation function formed by P_i, P'_j and P_k in Fig. 7. The minimizer in Fig. 7 for the case in Fig. 3(f) can be expressed as:

$$\begin{cases} x'_j = 2x_i - x_k \\ x^* = \dfrac{(x_i^2 - x'^2_j)f(x_k) + (x_k^2 - x_i^2)f(x_j) + (x'^2_j - x_k^2)f(x_i)}{2((x_i - x'_j)f(x_k) + (x_k - x_i)f(x_j) + (x'_j - x_k)f(x_i))} \end{cases} \tag{22}$$

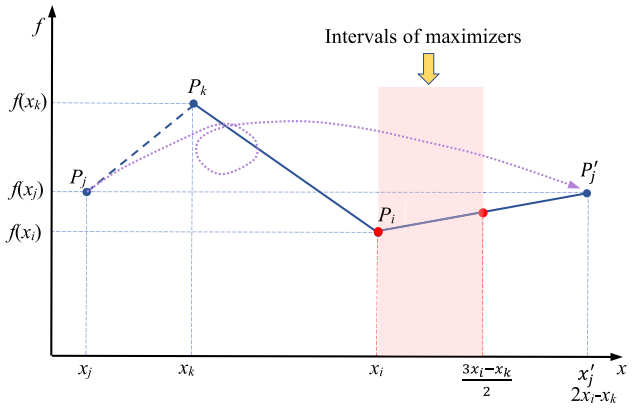

Fig. 7. Schematic for obtaining the minimizer interval in Fig. 3(f).

Therefore, the GQI method for obtaining the minimizers of the interpolation function under all cases is expressed as:

$$
\begin{cases}
x^* = \dfrac{(x_j^2-x_k^2)f(x_i)+(x_k^2-x_i^2)f(x_j)+(x_i^2-x_j^2)f(x_k)}{2((x_j-x_k)f(x_i)+(x_k-x_i)f(x_j)+(x_i-x_j)f(x_k))} & \text{if} \quad x_j < x_i < x_k \text{ or } x_k < x_i < x_j \\
\left.\begin{aligned} x' &= \dfrac{(x_j^2-x_k^2)f(x_i)+(x_k^2-x_i^2)f(x_j)+(x_i^2-x_j^2)f(x_k)}{2((x_j-x_k)f(x_i)+(x_k-x_i)f(x_j)+(x_i-x_j)f(x_k))} \\ x^* &= x' \end{aligned}\right\} \text{if } x' < x_j \\
\left.\begin{aligned} x'_k &= 3x_i - 2x_j \\ x^* &= \dfrac{(x_j^2-x_k'^2)f(x_i)+(x_k'^2-x_i^2)f(x_j)+(x_i^2-x_j^2)f(x_k)}{2((x_j-x_k')f(x_i)+(x_k'-x_i)f(x_j)+(x_i-x_j)f(x_k))} \end{aligned}\right\} \text{else} \\
\left.\begin{aligned} x' &= \dfrac{(x_j^2-x_k^2)f(x_i)+(x_k^2-x_i^2)f(x_j)+(x_i^2-x_j^2)f(x_k)}{2((x_j-x_k)f(x_i)+(x_k-x_i)f(x_j)+(x_i-x_j)f(x_k))} \\ x^* &= x' \end{aligned}\right\} \text{if } x' > x_j \\
\left.\begin{aligned} x'_k &= 3x_i - 2x_j \\ x^* &= \dfrac{(x_j^2-x_k'^2)f(x_i)+(x_k'^2-x_i^2)f(x_j)+(x_i^2-x_j^2)f(x_k)}{2((x_j-x_k')f(x_i)+(x_k'-x_i)f(x_j)+(x_i-x_j)f(x_k))} \end{aligned}\right\} \text{else} \\
\left.\begin{aligned} x'_j &= 2x_i - x_k \\ x^* &= \dfrac{(x_j'^2-x_k^2)f(x_i)+(x_k^2-x_i^2)f(x_j)+(x_i^2-x_j'^2)f(x_k)}{2((x_j'-x_k)f(x_i)+(x_k-x_i)f(x_j)+(x_i-x_j')f(x_k))} \end{aligned}\right\} \text{if} \quad x_i < x_k < x_j \text{ or } x_j < x_k < x_i
\end{cases}
\begin{aligned}
&\text{if} \quad x_i < x_j < x_k \\
&\text{if} \quad x_k < x_j < x_i
\end{aligned}
\tag{23}
$$

The effectiveness of the GQI method in optimization is demonstrated through its application of the minimization function. The function is provided below:

$$f(x) = x^2 - 10\cos(2\pi x) + 10 \quad x \in [0.6, 2] \tag{24}$$

Fig. 8 depicts the search behaviors of the GQI method on $f(x)$ and Table 1 provides a detailed description of the search process using the GQI method. It can be observed that the GQI method adeptly chooses the appropriate interpolation function defined in Eq. (23) based on the specific cases of three points. After 4 iterations, the minimum solution of $f(x)$ is obtained by the GQI method. In contrast, Fig. 9 depicts the search behavior of the quadratic interpolation method on $f(x)$ and Table 2 provides the search process of the quadratic interpolation method in detail. The point P_4 is obtained by the quadratic interpolation method via the points P_1, P_2 and P_3, but the point P_4 is not a smaller solution. Even by constructing different interpolation functions using various combinations of three points chosen from P_1, P_2, P_3, and P_4, it is impossible to obtain a smaller solution.

Table 1
Search process of the GQI method on $f(x)$.

Iterations	Selected three points	Interpolation polynomial	Computing minimizer using the GQI method	Minimizer
1	$P_1(x_1 = 1.16,\ f_{x_1} = 5.987)$, $P_2(x_2 = 1.64,\ f_{x_2} = 19.0638)$, $P_3(x_3 = 1.8,\ f_{x_3} = 10.1498)$	$L_{x_1,x_2,x_3} = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} f_{x_1} + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} f_{x_2} + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} f_{x_3}$	$x_4 = x^*_{x_1,x_2,x_3} = \frac{((2x_1-x_2)^2 - x_2^2) f_{x_1} + (x_2^2 - x_1^2) f_{x_3} + (x_1^2 - (2x_1-x_2)^2) f_{x_2}}{2(((2x_1-x_2)-x_2) f_{x_1} + (x_2-x_1) f_{x_3} + (x_1-(2x_1-x_2)) f_{x_2})}$, $f_{x_4} = f(x_4)$	$P_4(x_4 = 1.0359,\ f_{x_4} = 1.3265)$
2	$P_1(x_1 = 1.16,\ f_{x_1} = 5.987)$, $P_3(x_3 = 1.8,\ f_{x_3} = 10.1498)$, $P_4(x_4 = 1.0359,\ f_{x_4} = 1.3265)$	$L_{x_1,x_3,x_4} = \frac{(x-x_3)(x-x_4)}{(x_1-x_3)(x_1-x_4)} f_{x_1} + \frac{(x-x_1)(x-x_4)}{(x_3-x_1)(x_3-x_4)} f_{x_3} + \frac{(x-x_1)(x-x_3)}{(x_4-x_1)(x_4-x_3)} f_{x_4}$	$x_5 = x^*_{x_1,x_3,x_4} = \frac{(x_1^2 - (3x_4-2x_1)^2) f_{x_4} + ((3x_4-2x_1)^2 - x_4^2) f_{x_1} + (x_4^2 - x_1^2) f_{x_3}}{2((x_1-(3x_4-2x_1)) f_{x_4} + ((3x_4-2x_1)-x_4) f_{x_1} + (x_4-x_1) f_{x_3})}$, $f_{x_5} = f(x_5)$	$P_5(x_5 = 1.0023,\ f_{x_5} = 1.0057)$
3	$P_1(x_1 = 1.16,\ f_{x_1} = 5.987)$, $P_4(x_4 = 1.0359,\ f_{x_4} = 1.3265)$, $P_5(x_5 = 1.0023,\ f_{x_5} = 1.0057)$	$L_{x_1,x_4,x_5} = \frac{(x-x_4)(x-x_5)}{(x_1-x_4)(x_1-x_5)} f_{x_1} + \frac{(x-x_1)(x-x_5)}{(x_4-x_1)(x_4-x_5)} f_{x_4} + \frac{(x-x_1)(x-x_4)}{(x_5-x_1)(x_5-x_4)} f_{x_5}$	$x_6 = x^*_{x_1,x_4,x_5} = \frac{(x_4^2 - x_5^2) f_{x_1} + (x_5^2 - x_1^2) f_{x_4} + (x_1^2 - x_4^2) f_{x_5}}{2((x_4-x_5) f_{x_1} + (x_5-x_1) f_{x_4} + (x_1-x_4) f_{x_5})}$, $f_{x_6} = f(x_6)$	$P_6(x_6 = 0.9922,\ f_{x_6} = 0.9964)$
4	$P_4(x_4 = 1.0359,\ f_{x_4} = 1.3265)$, $P_5(x_5 = 1.0023,\ f_{x_5} = 1.0057)$, $P_6(x_6 = 0.9922,\ f_{x_6} = 0.9964)$	$L_{x_4,x_5,x_6} = \frac{(x-x_5)(x-x_6)}{(x_4-x_5)(x_4-x_6)} f_{x_4} + \frac{(x-x_4)(x-x_6)}{(x_5-x_4)(x_5-x_6)} f_{x_5} + \frac{(x-x_4)(x-x_5)}{(x_6-x_4)(x_6-x_5)} f_{x_6}$	$x_7 = x^*_{x_4,x_5,x_6} = \frac{(x_5^2 - x_6^2) f_{x_4} + (x_6^2 - x_4^2) f_{x_5} + (x_4^2 - x_5^2) f_{x_6}}{2((x_5-x_6) f_{x_4} + (x_6-x_4) f_{x_5} + (x_4-x_5) f_{x_6})}$, $f_{x_7} = f(x_7)$	$P_7(x_7 = 0.9950,\ f_{x_7} = 0.9950)$

Table 2
Search process of the quadratic interpolation method on $f(x)$.

Iterations	Selected three points	Interpolation polynomial	Computing minimizer using the quadratic interpolation method	Minimizer
1	$P_1(x_1 = 1.16,\ f_{x_1} = 5.987)$, $P_2(x_2 = 1.64,\ f_{x_2} = 19.0638)$, $P_3(x_3 = 1.8,\ f_{x_3} = 10.1498)$	$L_{x_1,x_2,x_3} = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} f_{x_1} + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} f_{x_2} + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} f_{x_3}$	$x_4 = x^*_{x_1,x_2,x_3} = \frac{(x_2^2-x_3^2)f(x_1)+(x_3^2-x_1^2)f_{x_2}+(x_1^2-x_2^2)f_{x_3}}{2((x_2-x_3)f_{x_1}+(x_3-x_1)f_{x_2}+(x_1-x_2)f_{x_3})}$ $f_{x_4} = f(x_4)$	$P_4(x_4 = 1.5051,\ f_{x_4} = 22.2602)$
2	$P_1(x_1 = 1.16,\ f_{x_1} = 5.987)$, $P_3(x_3 = 1.8,\ f_{x_3} = 10.1498)$, $P_4(x_4 = 1.5051,\ f_{x_4} = 22.2602)$	$L_{x_1,x_3,x_4} = \frac{(x-x_3)(x-x_4)}{(x_1-x_3)(x_1-x_4)} f_{x_1} + \frac{(x-x_1)(x-x_4)}{(x_3-x_1)(x_3-x_4)} f_{x_3} + \frac{(x-x_1)(x-x_3)}{(x_4-x_1)(x_4-x_3)} f_{x_4}$	$x_5 = x^*_{x_1,x_3,x_4} = \frac{(x_3^2-x_4^2)f(x_1)+(x_4^2-x_1^2)f_{x_3}+(x_1^2-x_3^2)f_{x_4}}{2((x_3-x_4)f_{x_1}+(x_4-x_1)f_{x_3}+(x_1-x_3)f_{x_4})}$ $f_{x_5} = f(x_5)$	$P_5(x_5 = 1.5036,\ f_{x_5} = 22.2582)$
3	$P_2(x_2 = 1.64,\ f_{x_2} = 19.0638)$, $P_3(x_3 = 1.8,\ f_{x_3} = 10.1498)$, $P_4(x_4 = 1.5051,\ f_{x_4} = 22.2602)$	$L_{x_2,x_3,x_4} = \frac{(x-x_3)(x-x_4)}{(x_2-x_3)(x_2-x_4)} f_{x_2} + \frac{(x-x_2)(x-x_4)}{(x_3-x_2)(x_3-x_4)} f_{x_3} + \frac{(x-x_2)(x-x_3)}{(x_4-x_2)(x_4-x_3)} f_{x_4}$	$x_6 = x^*_{x_2,x_3,x_4} = \frac{(x_3^2-x_4^2)f(x_2)+(x_4^2-x_2^2)f_{x_3}+(x_2^2-x_3^2)f_{x_4}}{2((x_3-x_4)f_{x_1}+(x_4-x_2)f_{x_3}+(x_2-x_3)f_{x_4})}$ $f_{x_6} = f(x_6)$	$P_6(x_6 = 1.4634,\ f_{x_6} = 21.8789)$

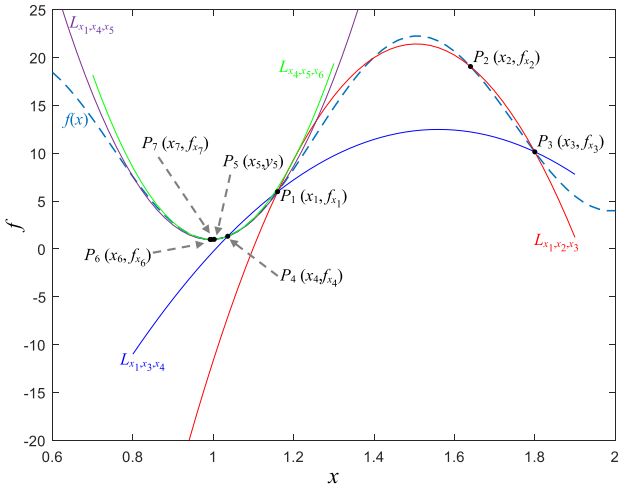

Fig. 8. Search behaviors of the GQI method on $f(x)$.

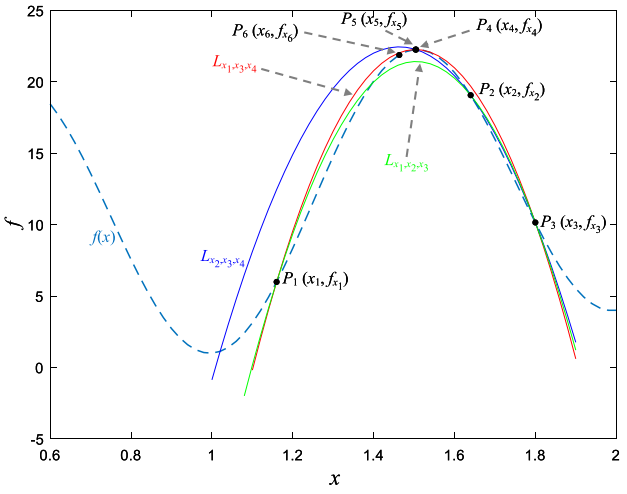

Fig. 9. Search behaviors of the quadratic interpolation method on $f(x)$.

2.3. Exploration strategy

Exploration refers to the capability of an algorithm to thoroughly search the entire variable space to identify promising areas that may contain the global optimum solution, avoiding local optimal or premature convergence. By searching different regions, exploration facilitates avoidance of neglecting any potential areas and the discovery of diverse and improved solutions. The GQI method is employed to determine the exploration strategy in the proposed QIO. To identify promising regions, two individuals randomly chosen from the current population and the current individual are jointly employed in the GQI method, and the minimizer of the interpolation function formed by three positions is obtained. Meanwhile, the third individual randomly chosen from the current population is utilized to produce the new candidate solution. It can enhance population diversity. When performing exploration using the GQI method, the position of the current individual is updated as:

$$v_i(\mathrm{t}+1) = x^*_{i,rand1,rand2}(t) + w_1 \cdot (x_{rand3}(t) - x^*_{i,rand1,rand2}(t)) + round(0.5 \cdot (0.05 + r_1)) \cdot \log \frac{r_2}{r_3} \tag{25}$$

$$x^*_{i,rand1,rand2}(t) = GQI(x_i(t), x_{rand1}(t), x_{rand2}(t),\ fit(x_i(t)),\ fit(x_{rand1}(t)),\ fit(x_{rand2}(t))) \tag{26}$$

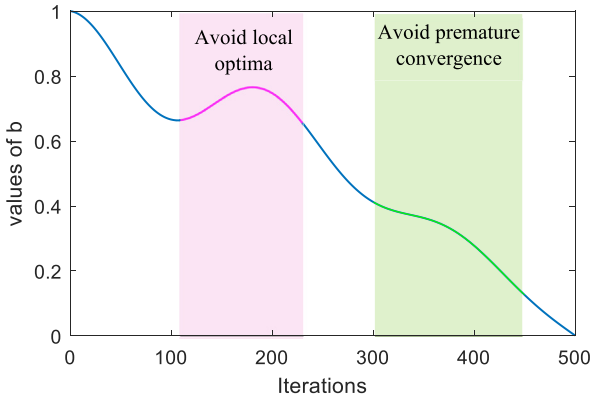

Fig. 10. Variation of adaptive parameter b versus iterations.

where x_{rand1}, x_{rand2} and x_{rand3} are the positions of the different individuals randomly chosen from the current population, r_1, r_2 and r_3 are the random numbers in (0,1), and $fit(\cdot)$ indicates the function fitness value. $GQI(x_i(t), x_{rand1}(t), x_{rand2}(t), fit(x_i(t)), fit(x_{rand1}(t)), fit(x_{rand2}(t)))$ is the GQI function, by which the minimizer of the interpolation function formed by $(x_i, fit(x_i))$, $(x_{rand2}, fit(x_{rand2}))$ and $(x_{rand3}, fit(x_{rand3}))$ is obtained. Three randomly chosen individuals in Eq. (25) can enlarge the search range effectively, and the GQI method is helpful to position the promising regions. To make the algorithm capable of first performing exploration and then balancing exploration and exploitation and finally converging to the optimum solution, an important parameter termed as the exploration weight w_1 is introduced, which can be changed by using an adaptive coefficient b. They are expressed as, respectively:

$$w_1 = 3n_1 b \tag{27}$$

$$b = 0.7a + 0.15a(\cos(\frac{5\pi t}{T}) + 1) \tag{28}$$

$$a = \cos(\frac{t\pi}{2T}) \tag{29}$$

where n_1 follows the standard normal distribution, t is the number of iterations, and T is the maximum number of iterations. Fig. 10 illustrates the variation of the adaptive parameter b versus the number of iterations. The curve in Fig. 10 reveals the overall trend of the parameter b throughout the iterations. During the early stage of iterations, the parameter b takes larger values, which aids the algorithm in exploring promising regions of the search space and promoting population diversity. As the iterations progress, the value of b gradually increases, ensuring a smooth transition from exploration stage to exploitation stage. Notably, there are two changes along the curve. First, there is an increase followed by a decrease in the value of b for the iteration numbers ranging from 120 to 240, which enables the algorithm to avoid the local optima, Second, the value of b exhibits a slower downward trend for the iteration numbers ranging from 300 to 450, which helps prevent premature convergence and enhance search capability of the algorithm.

2.4. Exploitation strategy

Exploitation is often contrasted with exploration, as it focuses on refining the best solution found thus far through local searches. Exploitation aims to converge faster and approach the global optimal solution more closely. In QIO, during the exploitation stage, two randomly chosen individuals from the current population and the best individual found so far are employed in the GQI method to produce a better solution. It is promising for the algorithm to find the global optimal solution in the region surrounding the minimizer. Therefore, when QIO performs exploitation, the position of the current individual is updated as:

$$v_i(\mathrm{t}+1) = x^*_{best,rand1,rand2}(t) + w_2 \cdot (x_{best}(t) - round(1 + \mathrm{rand}) \cdot \frac{Ub - Lb}{Ub^{rD} - Lb^{rD}} \cdot x_i^{rD}(t)) \tag{30}$$

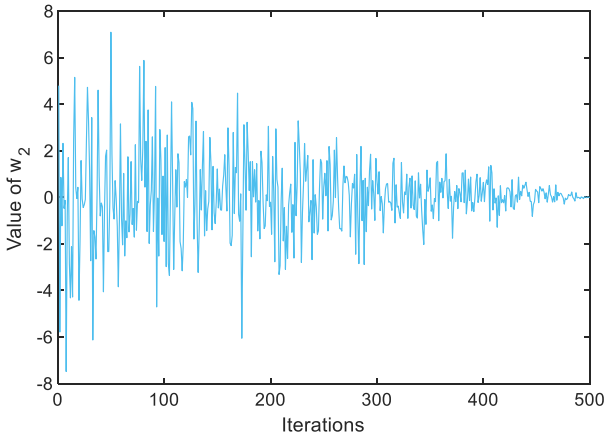

Fig. 11. Variation of exploitation parameter w_2 versus iterations.

$$x^*_{best,rand1,rand2}(t) = GQI(x_{best}(t), x_{rand1}(t), x_{rand2}(t), fit(x_{best}(t)), fit(x_{rand1}(t)), fit(x_{rand2}(t))) \tag{31}$$

$$w_2 = 3(1 - \frac{t-1}{T})n_2 \tag{32}$$

where n_2 follows the standard normal distribution and rD is a random integer from [1, d], Lb_{rD} and Ub_{rD} are the lower and higher boundaries at the (rD)th dimension. And w_2 is as the exploitation weight that is an adaptive coefficient. Fig. 11 depicts the variation of adaptive parameter w_2 versus iterations. It can be observed that the values of w_2 gradually decrease as the iteration number increases, which assists significantly enhance convergence accuracy.

From Eq. (31), the best individual found so far x_{best} and two randomly chosen individuals x_{rand1} and x_{rand2} are employed in the GQI method for producing the minimizer $x^*_{best,rand1,rand2}$, which is generally better than the three individuals. From Eq. (30), the ith individual is updated by searching the neighborhood of $x^*_{best,rand1,rand2}$ with respect to x_{best} with the random perturbation from x_i^{rD}. This search can assist the algorithm to conduct efficient exploitation without stagnation.

After the exploration and the exploitation strategies are achieved, the position of the ith individual is updated according to the following equations:

$$x_i(t+1) = \begin{cases} x_i(t) & fit(x_i(t)) \leq fit(v_i(t+1)) \\ v_i(\mathrm{t}+1) & fit(x_i(t)) > fit(v_i(t+1)) \end{cases} \tag{33}$$

If the fitness value of the candidate position of the ith individual is better than that of the current one, the position of the ith individual is replaced with its candidate one. Otherwise, it will remain unchanged.

The QIO algorithm initiates by generating a set of random solutions. In each iteration, when $rand > 0.5$, the algorithm obtains the minimizer of the GQI function with the current individual and two chosen randomly individuals using Eq. (26) and performs exploration using Eq. (25). When $rand \leq 0.5$, the algorithm obtains the minimizer of the GQI function with the best individual found so far and two chosen randomly individuals using Eq. (31) and performs exploitation using Eq. (30). When exploration or exploitation is achieved, the best solution found so far is updated using Eq. (33). Ultimately, the QIO algorithm stops when a pre-established termination criterion is met, and the best solution found so far is returned.

A flowchart of QIO is depicted in Fig. 12 and its pseudocode is provided in Algorithm 1.

2.5. Computational complexity analysis

All the above-defined equations will be achieved iteratively in the QIO algorithm. Therefore, the computational complexity of the algorithm is relevant to the number of individuals (n), the dimension of the problem (d), and the

Algorithm 1 Pseudo-code of QIO.

Set parameters n and T.
Randomly initialize the population x_i (i=1,...,n) and evaluate their fitness fit_i, and x_{best} is the best solution found so far.
While the stopping condition is not satisfied d**o**
  For each individual i=1:n
    Randomly choose two individuals $rn1 \neq rn2 \neq i$ from the population;
    If $rand$>0.5
      For each position component j=1:d **do**
      Obtain the minimizer $x_j^{i^*}$ of the GQI function with x_j^i, x_{rand1}^j and x_{rand2}^j using Eq. (26);
      End For
      Randomly choose one individual $rand3 \neq rand1$, $rand3 \neq rand2$ and $rand3 \neq i$ from the population;
      Perform exploration strategy for the ith individual using Eq. (25);
    Else
      For each position component j=1:d **do**
      Obtain the minimizer $x_j^{i^*}$ of the GQI function with x_{best}^j, x_{rand1}^j and x_{rand2}^j using Eq. (31);
      End For
    Perform exploitation strategy for the ith individual using Eq. (30);
    Evaluate the fitness fit_i of the ith individual;
    Update the ith individual using Eq. (33);
  End For
  Update the best solution found so far x_{best};
End While
Return the best solution found so far x_{best}.

maximum number of iterations (T). The total computational complexity is:

$$\begin{aligned} &O(\text{QIO}) = O(\text{problem definition}) + O(\text{initialization}) \\ &+O(\text{function evaluation}) + O(\text{position updating in exploration strategy}) \\ &+O(\text{position updating in exploitation strategy}) \\ &= O(1) + O(n) + O(Tn) + O(\frac{1}{2}Tnd) + O(\frac{1}{2}Tnd) \\ &= O(Tnd + Tn + n + 1) \cong O(Tnd) \end{aligned}$$

3. Experimental results and analyses

3.1. Numerical benchmarks and compared algorithms

To verify the effectiveness of the QIO algorithm, two well-known benchmark sets, the classical benchmark function consisting of 23 functions [48,88] and the challenging CEC-2014 suite containing 30 test functions [89], are employed in this study. The first test set comprises 7 unimodal functions (UFs) (F1-F7) and 16 multimodal functions

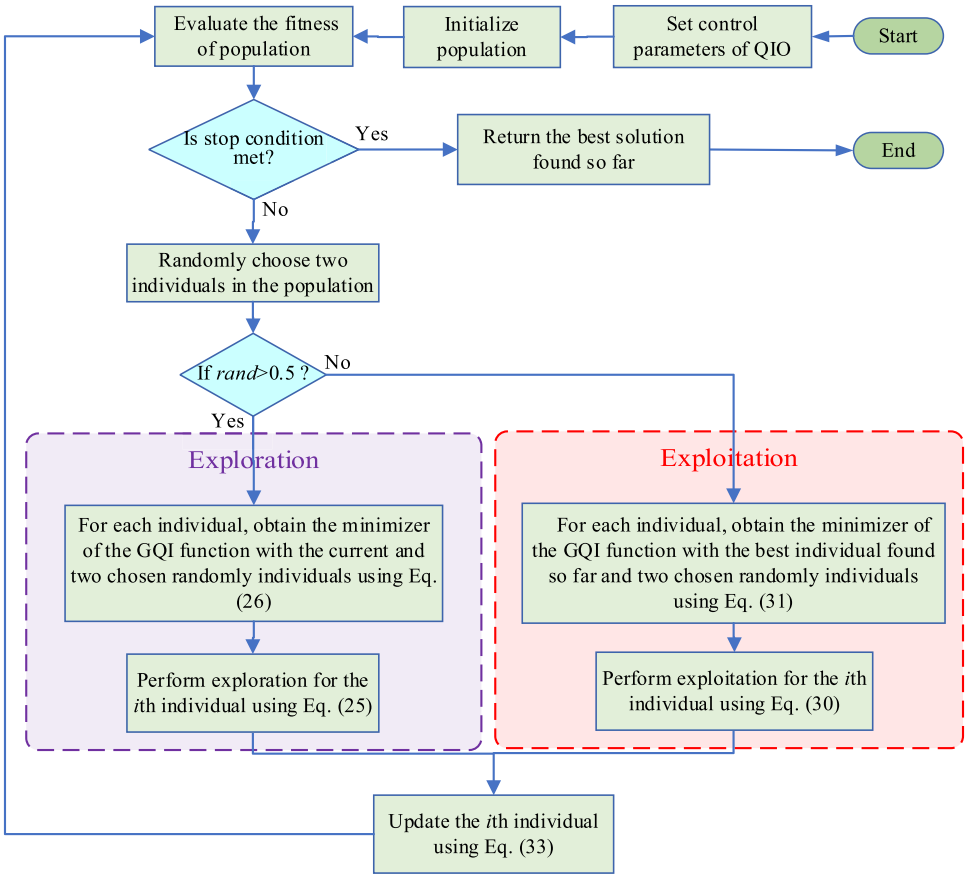

Fig. 12. Flowchart of QIO.

(MFs) (F8–F23), which are used to test the exploration and exploitation capabilities of the algorithm, respectively. Table A.1 describes the mathematical formulae of the first test set. In addition, to highlight the superiority of the QIO algorithm, 12 representative algorithms, including GSA [48], ABC [33], ASO [49], WOA [63], AOA [75], CS [90], MFO [91], SSA [92], HHO [93], LFD [94], GA, and PSO are used for performance comparison. Among them, GSA, ASO, WOA, CS, MFO, SSA, HHO, AOA, and LFD are recently developed optimizers; PSO, GA, and ABC are well-known optimizers. For each algorithm, the population size and the maximum number of fitness evaluations (FEs) are set to 50 and 25000, respectively. 50 independent runs are performed for each function, and the average (Ave) and standard deviation (Std) of the best solutions from each run are taken into account. All the experiments have been carried out on the MATLAB 9.7 (R2019b) and the desktop computer is run as Windows 10 64-bit with an Intel(R) Core(TM) i5-9400F CPU 2.9 GHz processor and 8.00 GB RAM. The parameters for all compared algorithms are from the original literature in the study, which are summarized in Table 3.

3.2. Performance indices

To better quantitatively evaluate the performance of QIO from various aspects, a set of indices are employed in the subsequent analysis.

(1) Average value (Ave): it represents the central value of the fitness values of the function across different runs:

$$Ave = \frac{1}{R}\sum_{i=1}^{R} fit_i \tag{34}$$

where R is the number of runs. The 'Ave' index reflects the central tendency of the fitness function values.

Table 3
Parameter settings of QIO and other competing algorithms.

Algorithms	Parameter settings
GSA	Gravitational constant = 100 and decreasing coefficient = 20
ABC	Limit = $n \cdot d$
ASO	Depth weight = 50 and multiplier weight = 0.2
WOA	Convergence parameter reduces linearly from 2 to 0
AOA	Control parameter $\sigma = 0.499$ and sensitive parameter v = 0.5
CS	Mutation probability = 0.25
MFO	a linearly reduces from -1 to -2, $b = 1$
SSA	Number of leaders = $0.5n$
HHO	E_0 is a random value in $(-1, 1)$
LFD	$Threshold = 2$, $CSV = 0.5$, $\beta = 1.5$, $\alpha_1 = 10$, $\alpha_2 = 0.00005$, $\alpha_3 = 0.005$, $\partial_1 = 0.9$, $\partial_2 = 0.1$
GA	Selection adopts the Roulette wheel method, crossover rate = 0.8, and the mutation rate = 0.2
PSO	$w = 0.6$, $c_1 = 2$, and $c_2 = 2$

(2) Standard deviation (Std): it represents the mean of the distances between the fitness values at each run and the mean of the fitness values at different runs:

$$Std = \sqrt{\frac{1}{R-1}\sum_{i=1}^{R}(fit_i - AV)^2} \tag{35}$$

The 'Std' reflects the fluctuation trend of the fitness function values.

(3) Wilcoxon's rank sum test (WRST)

For the WRST, R+ and R− are calculated as [95,96]:

$$\begin{cases} R^+ = \sum_{d_i>0} rank(d_i) + 0.5\sum_{d_i=0} rank(d_i) \\ R^- = \sum_{d_i<0} rank(d_i) + 0.5\sum_{d_i=0} rank(d_i) \\ T = \min(R^+, R^-) \end{cases} \tag{36}$$

where d_i is the difference between the performance scores of QIO and the compared algorithm on ith problem, $R+$ is the sum of ranks for the given problem in which QIOoutperforms the second, and $R-$ is the sum of ranks for the opposite. When T is less than or equal to the value of the distribution of Wilcoxon for n degrees of freedom, it means QIO outperforms the compared algorithm with p (5%) significance, vice versa.

(4) Friedman aligned test (FAT)

The FAT is expressed as [96]:

$$FAT = \frac{(k-1)\left(\sum_{j=1}^{k} R_j^2 - (kn^2/4)(kn+1)^2\right)}{kn(kn+1)(2kn+1)/6 - 1/k\sum_{i=1}^{n} R_i^2} \tag{37}$$

where R_i is the rank total of ith problem and R_j is the rank total of jth algorithm, k is the number of algorithms and n is the number of problems. The smaller the average of R_i, the better the rank of the corresponding algorithm.

(5) Quade rank (QR)

The QR is expressed as [96]:

$$QR = \frac{(n-1)\frac{1}{n}\sum_{j=1}^{k} k\left(\sum_{i=1}^{n}\sum_{i=1}^{n}(Q_i[r_i^j - \frac{k+1}{2}])\right)^2}{\frac{n(n+1)(2n+1)k(k^2-1)}{72} - \frac{1}{n}\sum_{j=1}^{k} k\left(\sum_{i=1}^{n}\sum_{i=1}^{n}(Q_i[r_i^j - \frac{k+1}{2}])\right)^2} \tag{38}$$

The average ranking for the jth algorithm is given as:

$$T_j = \frac{\sum_{i=1}^{n}(Q_i[r_i^j])}{n(n+1)/2} \tag{39}$$

where Q_i is the rank assigned to the ith problem, r_i^j is the rank of the jth algorithm for the ith problem. The smaller the value of T_j, the better the ranking of the algorithm.

3.3. Quantitative analysis

The experimental results of all compared algorithms on 23 benchmark functions are provided in Table 4. For UFs F1–F7, QIO outperforms other algorithms concerning 'Ave' and 'Std' on F1 and F3–F5. For F6, QIO performs just as well as WOA, AOA, HHO, LFD, and GA. For F2, QIO and AOA can find the true optimal solution at each run. For F7, the results of QIO are ranked second, following the performance of HHO. Undoubtedly, QIO shows superior exploitation when solving UFs. For MFs F8–F23, QIO provides the best results concerning 'Ave' and 'Std' on F8, F12, F13, F15, and F21, and QIO obtains the same best 'Ave' as other algorithms on F9–F11, F14, F16–F19, and F22–F23. Obviously, QIO is the most efficient algorithm for the majority of MFs, indicating its superior exploration in comparison to its competitors.

3.4. Convergence analysis

The convergence curves of all the optimizers on 23 functions are provided in Fig. 13, in which the average of the iterative solutions over 50 runs is plotted. For UFs F1–F6, QIO provides the fastest convergence rate while obtaining the best results. For F7, although the convergence rate of QIO is ranked second, following the performance of HHO, it can rapidly converge to the best solution. For MFs F8–F13 with a number of local optima, QIO can perform a global search across enormous local regions and rapidly position the promising region; then the algorithm switches from exploration to exploitation to perform a local search near the best solution; and finally, the algorithm continuously improves the accuracy of final solution and convergence rate. The typical convergence characteristic observed in the algorithm is crucial in enabling it to avoid getting trapped in local optima. For MFs F15–F23 with less local optima, after short-term exploration, QIO quickly identifies the potential region and converges towards the best solution. Especially for MFs F14, F15, F21 and F23, it is evident that QIO converges to the best solution with the highest speed. In most cases, the convergence rate and the solution accuracy of QIO are the best.

3.5. Statistical analysis

Tables 5–7 present a comparison of QIO with other optimizers on 23 functions using the WRST with a significance level of 5%. In the tables, '+' signifies that QIO offers better results than the competing optimizer, and '−' vice versa. '=' signifies that QIO and the competing optimizer offer similar results. Table 6 summarizes the results of WRST. From Table 8, the WRST results of QIO versus GSA, ABC, ASO, WOA, AOA, CS, MFO, SSA, HHO, LFD, GA, and PSO are 17/2/4, 19/0/4, 14/2/7, 19/4/0, 18/4/1, 19/1/3, 16/3/4, 19/1/3, 18/4/1, 20/2/1, 17/3/3, and 17/1/5, respectively. That is, the numbers of problems that QIO are better than other 12 optimizers are 17, 19, 14, 19, 18, 19, 16, 19, 18, 20, 17 and 17, respectively. It is evident that QIO achieves a significant improvement over each optimizer for solving 23 benchmark functions. These findings highlight the superior performance and effectiveness of QIO in tackling a wide range of optimization problems.

The boxplot is a visual statistic method for evaluating the performance of algorithms using the empirical distribution of the fitness values. The boxplots of various algorithms on 23 functions are plotted in Fig. 14. As shown in the figures, in most cases, the boxplots of QIO are narrower, and the location of boxplots is under the minima of other optimizers, indicating there is a significant difference between QIO and other algorithms in terms of stability and solution quality in tackling these functions.

As an effective statistical evaluation method, the FAT [97] is utilized to assess the rank of QIO among all considered algorithms. The FAT is carried out on the mean best solutions over 50 runs provided by QIO and other competing algorithms, Fig. 15 plots the radar charts of FAT for each algorithm on 23 functions, in which QIO offers a smaller area than other algorithms, indicating its better performance. The average rank of algorithms on all the functions is depicted in Fig. 16. It is obvious that QIO ranks first with an average rank of 104.24 followed by HHO (105.89) and SSA (117.04), and in the last place is CS (224.70). The final ranking of the average rank provided by FAT for each algorithm on all 23 functions is QIO > HHO > SSA > ASO > GA > LFD > GSA > WOA >

Table 4
Comparisons of the results of different algorithms on 23 test functions.

Fun.	Index	QIO	GSA	ABC	ASO	WOA	AOA	CS	MFO	SSA	HHO	LFD	GA	PSO
F_1	Ave	**0**	3.742E−17	1.711E+00	6.510E−19	2.949E−85	6.561E−28	6.681E+02	8.053E+02	2.066E−08	1.027E−226	9.905E−09	1.426E−01	1.911E+00
	Std	0	1.097E−17	5.193E−01	5.392E−19	1.535E−84	4.639E−27	1.404E+02	3.404E+03	5.899E−09	0	5.832E−09	5.453E−02	1.758E+00
F_2	Ave	**0**	3.252E−08	3.394E−02	6.125E−08	1.741E−53	**0**	5.435E+01	3.139E+01	8.117E−01	2.370E−117	9.165E−03	8.144E−02	1.397E−01
	Std	0	6.813E−09	7.372E−03	3.125E−07	6.995E−53	0	1.019E+01	2.279E+01	7.883E−01	8.549E−117	8.376E−03	1.977E−02	8.502E−02
F_3	Ave	**0**	5.695E+02	1.147E+04	1.325E+03	3.047E+04	4.557E−03	9.670E+03	1.624E+04	5.168E+02	2.931E−211	7.695E−05	1.530E+03	7.897E+03
	Std	0	1.896E+02	1.695E+03	5.606E+02	1.140E+04	8.770E−03	1.851E+03	1.136E+04	2.699E+02	0	5.941E−05	5.220E+02	3.380E+03
F_4	Ave	**1.969E−316**	3.426E+00	2.769E+01	2.662E−07	3.644E+01	2.379E−02	2.544E+01	5.749E+01	6.507E+00	9.947E−116	3.760E−03	5.099E+00	2.498E+01
	Std	0	1.507E+00	2.025E+00	2.709E−07	2.875E+01	2.061E−02	2.293E+00	8.990E+00	2.404E+00	4.808E−115	2.709E−03	1.859E+00	5.115E+00
F_5	Ave	**6.51E−07**	4.481E+01	9.732E+03	3.559E+01	2.748E+01	2.818E+01	7.939E+04	4.686E+03	1.516E+02	7.084E−02	2.790E+01	1.986E+02	9.027E+02
	Std	1.72E−07	6.312E+01	5.761E+03	3.066E+01	4.095E−01	3.534E−01	4.025E+04	1.765E+04	2.006E+02	6.074E−02	1.625E−01	1.938E+02	1.028E+03
F_6	Ave	**0**	4.200E−01	4.580E+00	1.000E−01	**0**	**0**	7.877E+02	1010.58	1.218E+01	**0**	**0**	**0**	8.140E+00
	Std	0	9.055E−01	1.785E+00	0.580288457	0	0	176.9125738	3639.549939	6.951E+00	0	0	0	4.463E+00
F_7	Ave	4.46E−04	2.962E−02	1.405E−01	4.712E−02	2.451E−03	4.067E−05	2.614E−01	2.012E+00	8.657E−02	**1.513E−04**	9.134E−03	6.861E−02	1.273E−01
	Std	3.06E−04	1.327E−02	3.563E−02	1.877E−02	2.857E−03	3.501E−05	8.086E−02	4.944E+00	4.228E−02	1.320E−04	8.131E−03	2.758E−02	4.802E−02
F_8	Ave	**−12 569.4866**	−2750.2075	−4738.7503	−7163.5531	−11 373.6890	−5499.1513	−7606.4934	−8768.1058	−7757.2412	−12 569.1467	−6417.4683	−6848.8659	−4857.6514
	Std	1.01E−07	4.936E+02	3.847E+02	5.596E+02	1.636E+03	4.046E+02	2.337E+02	6.990E+02	7.565E+02	6.217E−01	3.837E+03	7.304E+02	5.357E+02
F_9	Ave	**0**	1.620E+01	1.918E+02	2.885E+01	2.274E−15	**0**	1.475E+02	1.537E+02	3.884E+01	**0**	2.579E+00	1.909E+01	4.678E+01
	Std	0	3.814E+00	1.001E+01	7.033E+00	1.125E−14	0	1.498E+01	3.798E+01	1.425E+01	0	7.274E+00	5.054E+00	1.033E+01
F_{10}	Ave	**8.882E−16**	4.755E−09	5.731E−01	5.193E−10	4.370E−15	**8.882E−16**	1.441E+01	1.267E+01	2.108E+00	**8.882E−16**	1.338E−04	9.368E−02	1.111E+00
	Std	0	7.740E−10	1.697E−01	2.933E−10	2.211E−15	0	1.486E+00	8.372E+00	7.518E−01	0	2.823E−05	2.623E−02	6.489E−01
F_{11}	Ave	**0**	1.833E+01	9.357E−01	3.451E−04	1.539E−03	1.093E−01	7.102E+00	1.345E+01	8.741E−03	**0**	2.035E−03	2.018E−01	8.968E−01
	Std	0	3.465E+00	5.845E−02	1.726E−03	1.088E−02	8.040E−02	1.384E+00	3.638E+01	9.787E−03	0	8.668E−03	5.843E−02	1.509E−01
F_{12}	Ave	**3.85E−09**	6.344E−01	4.308E+01	2.629E−03	8.124E−03	4.288E−01	1.732E+01	4.556E+00	4.285E+00	2.650E−05	1.861E−03	7.788E−04	6.597E+00
	Std	1.06E−09	5.668E−01	2.749E+01	1.510E−02	9.062E−03	4.842E−02	5.061E+00	3.707E+00	2.284E+00	1.989E−05	7.620E−04	9.581E−04	5.257E+00
F_{13}	Ave	**1.40E−08**	1.721E+00	5.332E+03	7.702E−04	2.101E−01	2.820E+00	4.717E+03	1.022E+01	1.617E+00	3.216E−04	9.224E−01	1.291E−02	2.904E+01
	Std	4.30E−09	2.159E+00	6.279E+03	2.723E−03	1.496E−01	1.152E−01	6.557E+03	1.162E+01	4.159E+00	2.439E−04	1.249E+00	8.320E−03	5.146E+01
F_{14}	Ave	**0.9980**	5.0178	**0.9980**	1.2960	3.1168	8.8240	**0.9980**	1.5325	1.0179	**0.9980**	1.0774	3.1456	**0.9980**
	Std	4.975E−13	3.108E+00	8.591E−05	5.251E−01	3.422E+00	4.265E+00	8.766E−11	1.184E+00	1.406E−01	1.288E−10	3.380E−01	2.683E+00	3.172E−17
F_{15}	Ave	**3.075E−04**	3.795E−03	5.612E−04	1.020E−03	6.806E−04	1.273E−02	6.603E−04	9.050E−04	6.402E−04	3.325E−04	9.707E−04	1.786E−03	4.534E−04
	Std	6.440E−10	1.950E−03	5.531E−05	2.175E−04	3.579E−04	2.318E−02	9.285E−05	3.146E−04	2.455E−04	1.288E−04	4.641E−04	4.028E−03	1.288E−04
F_{16}	Ave	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**
	Std	7.198E−11	4.062E−16	2.43E−10	2.557E−16	4.388E−10	1.227E−07	3.253E−12	2.243E−16	2.141E−14	1.173E−07	8.163E−11	2.681E−09	2.557E−16
F_{17}	Ave	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**	0.4073	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**
	Std	1.115E−10	3.364E−16	3.38E−09	3.364E−16	3.745E−06	9.070E−03	2.666E−10	3.364E−16	3.939E−14	3.768E−06	7.337E−09	4.971E−07	7.322E−05

(*continued on next page*)

Table 4 (*continued*).

Fun.	Index	QIO	GSA	ABC	ASO	WOA	AOA	CS	MFO	SSA	HHO	LFD	GA	PSO
F_{18}	Ave	**3.0000**	**3.0000**	**3.0000**	**3.0000**	**3.0000**	10.5600	**3.0000**	**3.0000**	**3.0000**	**3.0000**	3.0277	3.5400	**3.0000**
	Std	6.013E−10	3.091E−15	1.47E−10	2.096E−15	2.865E−05	1.225E+01	3.781E−11	1.831E−15	1.999E−13	2.407E−05	2.340E−02	3.818E+00	1.813E−15
F_{19}	Ave	**−3.8628**	**−3.8628**	**−3.8628**	**−3.8628**	−3.8600	**−3.8528**	**−3.8628**	**−3.8628**	**−3.8628**	−3.8622	−3.8603	**−3.8628**	**−3.8628**
	Std	1.30E−09	2.824E−15	1.81E−15	3.107E−15	3.177E−03	4.073E−03	2.036E−10	3.140E−15	1.392E−06	1.291E−03	2.405E−03	1.441E−07	3.013E−15
F_{20}	Ave	−3.2935	**−3.3220**	**−3.3220**	**−3.3220**	−3.2292	−3.0861	−3.3217	−3.2242	−3.2434	−3.2681	−3.2010	−3.3054	−3.2588
	Std	5.129E−02	6.344E−17	1.09E−14	0	8.544E−02	7.600E−02	2.564E−04	5.105E−02	7.463E−02	6.656E−02	8.308E−02	4.167E−02	6.136E−02
F_{21}	Ave	**−10.1532**	−6.6447	−10.1503	−7.5652	−9.2907	−3.5029	−10.1518	−7.0863	−9.6480	−10.1520	−7.8199	−7.8089	−5.9493
	Std	8.313E−09	3.696E+00	1.982E−02	3.189E+00	2.197E+00	8.687E−01	1.905E−03	3.215E+00	1.531E+00	1.531E−03	2.836E+00	3.468E+00	2.675E+00
F_{22}	Ave	**−10.4029**	−10.2811	**−10.4029**	−10.3306	−8.4882	−4.1565	−10.3988	−8.3212	−9.6646	−10.4017	−8.3146	−9.2566	−7.7836
	Std	9.162E−09	8.613E−01	1.227E−13	5.115E−01	2.847E+00	1.620E+00	7.840E−03	2.980E+00	1.849E+00	1.569E−03	2.925E+00	2.661E+00	3.146E+00
F_{23}	Ave	**−10.5364**	**−10.5364**	**−10.5364**	−9.7184	−6.7849	−3.8873	−10.5257	−8.6411	−9.9994	−10.5353	−7.9061	−8.8000	−8.1141
	Std	9.044E−09	0	5.963E−13	2.267E+00	3.357E+00	1.335E+00	1.745E−02	3.107E+00	1.627E+00	1.477E−03	3.168E+00	3.144E+00	3.201E+00

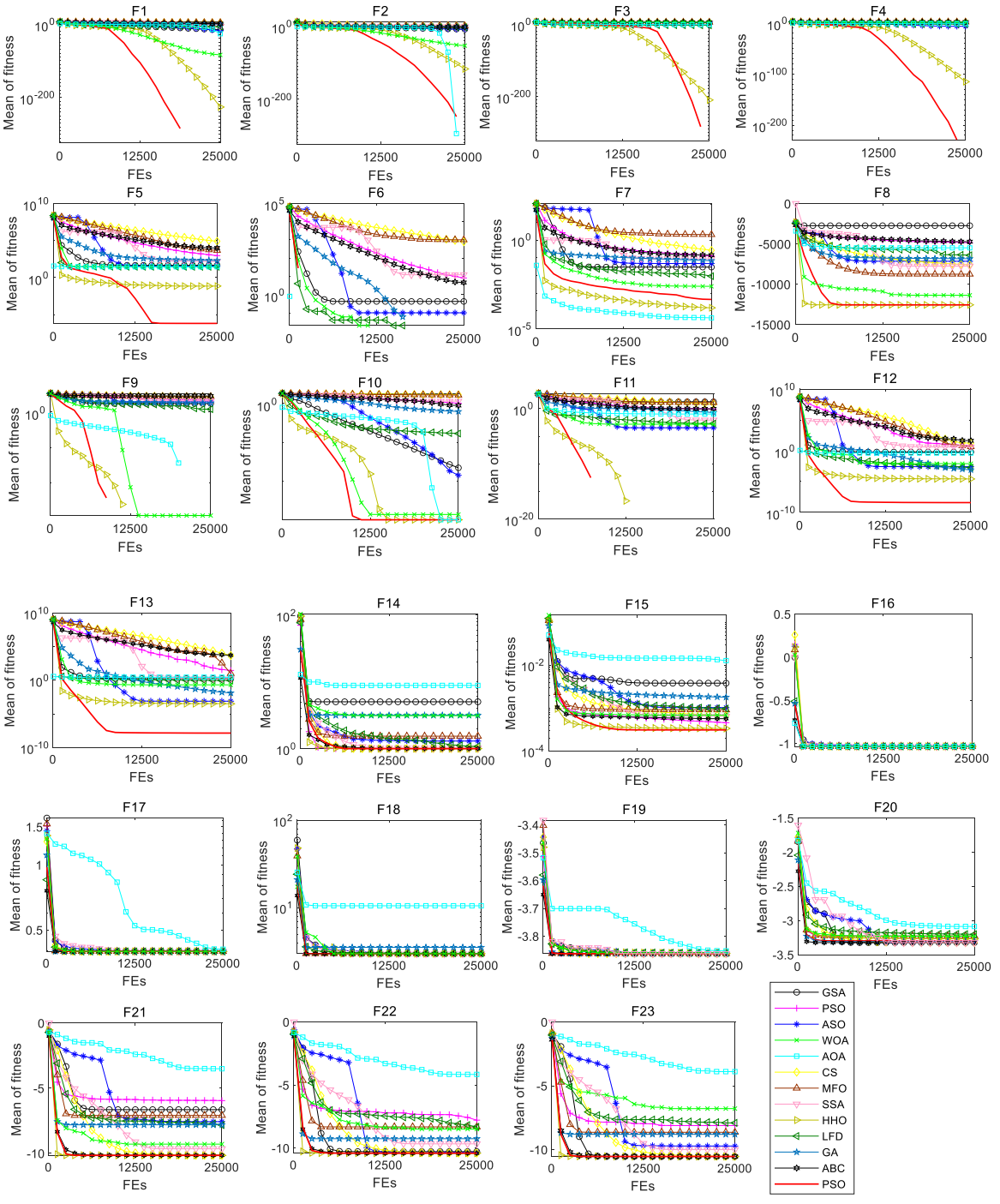

Fig. 13. Convergence curves of QIO and other optimizers on 23 functions.

AOA > PSO > ABC > MFO > CS. The statistical results of FAT explicitly reveal that QIO is the best optimizer among all the considered algorithms.

QR is another important statistical method based on multiple comparisons. Fig. 17 illustrates the radar maps of QR for the rank of each algorithm on all the functions. It can be observed that the radar map of QIO provides a smaller region in comparison to other algorithms, indicating better performance. The average rank of QR for

Table 5
Comparisons of WRST for QIO versus GSA, ABC, ASO, and WOA on 23 functions.

Fun.	GSA vs. QIO				ABC vs. QIO				ASO vs. QIO				WOA vs. QIO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner	p-value	T−	T+	Winner
F_1	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_2	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_3	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_4	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_5	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_6	1.221E−04	0	1275	+	6.291E−10	0	1275	+	5.000E−01	0	1275	≈	1.000E+00	0	1275	≈
F_7	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	6.813E−07	123	1152	+
F_8	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_9	7.487E−10	0	1275	+	7.557E−10	0	1275	+	7.550E−10	0	1275	+	5.000E−01	0	1275	≈
F_{10}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	4.709E−09	0	1275	+
F_{11}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	2.500E−01	0	1275	≈	1.000E+00	0	1275	≈
F_{12}	8.031E−10	1	1274	+	7.557E−10	0	1275	+	2.011E−07	1176	99	−	7.557E−10	0	1275	+
F_{13}	2.661E−09	21	1254	+	7.557E−10	0	1275	+	1.859E−05	1081	194	−	7.557E−10	0	1275	+
F_{14}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.087E−09	6	1269	+	9.634E−10	4	1271	+
F_{15}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{16}	7.557E−10	1275	0	−	1.106E−04	237	1038	+	7.557E−10	1275	0	−	2.294E−01	762	513	≈
F_{17}	7.557E−10	1275	0	−	6.279E−08	77	1198	+	7.557E−10	1275	0	−	2.994E−09	23	1252	+
F_{18}	7.557E−10	1275	0	−	5.339E−08	1201	74	−	7.557E−10	1275	0	−	9.068E−10	3	1272	+
F_{19}	3.177E−01	534	741	≈	7.557E−10	1275	0	−	7.557E−10	1275	0	−	7.557E−10	0	1275	+
F_{20}	7.557E−10	1275	0	−	7.557E−10	1275	0	−	7.557E−10	1275	0	−	4.229E−06	161	1114	+
F_{21}	5.681E−03	351	924	+	2.011E−07	99	1176	+	2.543E−02	406	869	+	7.557E−10	0	1275	+
F_{22}	1.417E−08	50	1225	+	7.557E−10	1275	0	−	1.417E−08	50	1225	+	7.557E−10	0	1275	+
F_{23}	3.177E−01	741	534	≈	7.557E−10	0	1275	+	6.670E−04	285	990	+	7.557E−10	0	1275	+

Table 6
Comparisons of WRST for QIO versus AOA, CS, MFO, and SSA on 23 functions.

Fun.	AOA vs. QIO				CS vs. QIO				MFO vs. QIO				SSA vs. QIO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner	p-value	T−	T+	Winner
F_1	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_2	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_3	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_4	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_5	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_6	1	0	1275	≈	7.554E−10	0	1275	+	1.514E−09	0	1275	+	7.280E−10	0	1275	+
F_7	1.024E−09	1270	5	−	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_8	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_9	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{10}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{11}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{12}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{13}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{14}	7.557E−10	0	1275	+	4.778E−09	31	1244	+	3.177E−01	741	534	≈	1.417E−08	50	1225	+
F_{15}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{16}	7.557E−10	0	1275	+	7.557E−10	1275	0	−	7.557E−10	1275	0	−	7.557E−10	1275	0	−
F_{17}	7.557E−10	0	1275	+	4.415E−02	429	846	+	7.557E−10	1275	0	−	7.557E−10	1275	0	−
F_{18}	1.339E−08	49	1226	+	1.303E−09	1266	9	−	7.557E−10	1275	0	−	7.557E−10	1275	0	−
F_{19}	7.557E−10	0	1275	+	1.757E−09	1261	14	−	7.557E−10	1275	0	−	5.789E−01	695	580	≈
F_{20}	7.557E−10	0	1275	+	3.177E−01	534	741	≈	3.116E−04	264	1011	+	1.623E−03	311	964	+
F_{21}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	2.556E−03	325	950	+	1.245E−04	240	1035	+
F_{22}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	4.602E−01	561	714	≈	2.901E−03	329	946	+
F_{23}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.832E−01	666	609	≈	1.245E−04	240	1035	+

Table 7
Comparisons of WRST for QIO versus HHO, LFD, GA, and PSO on 23 functions.

Fun.	HHO vs. QIO				LFD vs. QIO				GA vs. QIO				PSO vs. QIO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner	p-value	T−	T+	Winner
F_1	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_2	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_3	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_4	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_5	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_6	1	0	1275	≈	1	0	1275	≈	1	0	1275	≈	7.557E−10	0	1275	+
F_7	5.064E−09	1243	32	−	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_8	7.557E−10	0	1275	+	1.417E−08	50	1225	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_9	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{10}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{11}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{12}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{13}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
F_{14}	1.024E−09	5	1270	+	2.191E−06	147	1128	+	2.359E−06	138	1137	+	7.555E−10	1275	0	−
F_{15}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.031E−10	1	1274	+
F_{16}	8.031E−10	1	1274	+	1.519E−02	889	386	−	2.011E−07	1176	99	−	7.557E−10	1275	0	−
F_{17}	7.557E−10	0	1275	+	2.231E−07	101	1174	+	1.245E−04	1035	240	−	1.417E−08	1225	50	−
F_{18}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.245E−04	1035	240	−	7.557E−10	1275	0	−
F_{19}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	5.463E−01	575	700	≈	7.557E−10	1275	0	−
F_{20}	2.51E−05	201	1074	+	5.339E−08	74	1201	+	3.272E−01	536	739	≈	2.723E−03	327	948	+
F_{21}	7.557E−10	0	1275	+	1.159E−02	376	899	+	1.303E−09	9	1266	+	1.324E−07	91	1184	+
F_{22}	7.557E−10	0	1275	+	1.224E−02	378	897	+	1.024E−09	5	1270	+	5.681E−03	351	924	+
F_{23}	7.557E−10	0	1275	+	5.061E−02	435	840	≈	1.303E−09	9	1266	+	1.720E−01	496	779	≈

Table 8
Statistical results of WRST based on Tables 5–7.

Function type	GSA vs. QIO (+/≈/−)	ABC vs. QIO (+/≈/−)	ASO vs. QIO (+/≈/−)	WOA vs. QIO (+/≈/−)	AOA vs. QIO (+/≈/−)	CS vs. QIO (+/≈/−)
UFs	7/0/0	7/0/0	6/1/0	6/1/0	4/2/1	7/0/0
MFs	10/2/4	12/0/4	8/1/7	13/3/0	14/2/0	12/1/3
Total	17/2/4	19/0/4	14/2/7	19/4/0	18/4/1	19/1/3
Function type	MFO vs. QIO (+/≈/−)	SSA vs. QIO (+/≈/−)	HHO vs. QIO (+/≈/−)	LFD vs. QIO (+/≈/−)	GA vs. QIO (+/≈/−)	PSO vs. QIO (+/≈/−)
UFs	7/0/0	7/0/0	5/1/1	6/1/0	6/1/0	7/0/0
MFs	9/3/4	12/1/3	13/3/0	14/1/1	11/2/3	10/1/5
Total	16/3/4	19/1/3	18/4/1	20/2/1	17/3/3	17/1/5

various algorithms on all the functions is depicted in Fig. 18. It shows that QIO is the best-performing algorithm by offering the least average rank of 0.98, followed by HHO (1.41) and ASO (2.83). The final ranking of the average rank provided by QR for each algorithm on all 23 functions is QIO > HHO > ASO > LFD > WOA > AOA > GSA > GA > SSA > PSO > ABC > CS > MFO. It is notable that the final ranks of FAT and QT for the 13 algorithms on the 23 functions are different. This is because FAT primarily focuses on intra-set comparisons and comparability of the problem solutions, while QT takes into account the difficulty and importance of problems and uses the covariate to weight the ranks. Although FAT and QT emphasize different aspects in their ranking approaches, our proposed method consistently achieves the top rank.

3.6. CEC-2014 test function analysis

The CEC-2014 test suite consisting of 30 complex functions is employed to further test the effectiveness of the QIO algorithm. This test suite includes 3 UFs (CF1–CF3), 13 MFs (CF4–CF16), 6 hybrid functions (HFs) (CF17–CF22), and 8 composition functions (CFs) (F23–F30). For each algorithm, the population size and the maximum number of FEs are set to 50 and 100,000, respectively. The problem dimensionality is 30, and 50 independent runs are performed for each problem. Table 9 lists the results of all the algorithms on the test suite. From Table 9, the number of the best results provided by QIO, GSA, ABC, ASO, AOA, CS, SSA, HHO, LFD, GA and PSO is 15, 5, 2, 8, 1, 5, 2, 4, 2, 2, and 2 out of 30 functions; the rest algorithms do not provide the best results for any functions. This result reflects the effectiveness of the search strategies in QIO when tackling challenging functions.

Fig. 19 depicts the convergence curves of all the algorithms for the CEC-2014 suite. From Fig. 19, in most cases, QIO displays better convergence performance, which is helpful for performing exploitation, exploration, switching between both, and avoiding local optima when solving different types of functions. Tables 10–12 present the statistics of WRST with 5% significance for QIO and other optimizers, and Table 13 summarizes the results of WRST from Tables 10–12. From Table 13, the numbers of problems that QIO are better than other 12 optimizers are 22, 24, 17, 25, 26, 19, 28, 21, 23, 27, 25 and 24 out 30 functions, respectively. It is evident that QIO exhibits a significant difference from all other algorithms in over half of the 30 test functions.

Fig. 20 depicts the boxplots of various algorithms on the CEC-2014 functions. Based on Fig. 20, for most functions, QIO exhibits narrower boxplots with lower locations compared to other algorithms, highlighting its stable and effective search performance in solving complex problems.

Fig. 21 plots the radar charts of FAT for each algorithm on the CEC-2014 functions. From Fig. 21, QIO provides a better rank distribution for the CEC-2014 functions compared to other optimizers. The average ranks of FAT for various algorithms on the CEC-2014 functions are illustrated in Fig. 22. Comparing the average rank of each algorithm on all functions, QIO ranks first with an average rank of 127.38. The final ranking of the average rank provided by FAT for each algorithm on all benchmark functions is QIO > ASO > SSA > CS > GA > PSO > ABC > GSA > LFD > WOA > MFO > HHO > AOA.

Fig. 23 depicts the radar charts of QR for each algorithm on the CEC-2014 functions, in which QIO offers the smallest area among all the algorithms, indicating its better ranking. The average rank of QR for each algorithm is provided in Fig. 24, in which the distribution of the average rank of the best solutions for each algorithm on the

Table 9
Results of QIO and other algorithms for CEC-2014 functions.

Fun.	Index	QIO	GSA	ABC	ASO	WOA	AOA	CS	MFO	SSA	HHO	LFD	GA	PSO
CF_1	Ave	**1.078E+06**	2.141E+07	9.312E+07	3.045E+06	6.415E+07	9.646E+08	1.054E+07	7.302E+07	7.288E+06	1.850E+08	1.471E+08	9.739E+06	1.096E+08
	Std	6.901E+05	1.393E+07	1.513E+07	3.429E+06	2.433E+07	3.101E+08	3.246E+06	8.558E+07	3.632E+06	8.922E+07	1.506E+08	3.732E+06	5.267E+07
CF_3	Ave	**1.127E+03**	8.585E+03	5.108E+05	7.534E+03	1.343E+08	6.527E+10	1.000E+10	1.150E+10	1.158E+04	1.535E+10	7.936E+07	1.763E+04	6.356E+07
	Std	2.124E+03	2.450E+03	4.105E+05	5.559E+03	9.320E+07	9.573E+09	0	6.459E+09	9.370E+03	7.609E+09	4.380E+08	5.302E+03	1.435E+08
CF_3	Ave	6.784E+02	6.595E+04	4.980E+04	6.122E+03	6.314E+04	8.091E+04	**5.313E+02**	1.028E+05	2.091E+04	6.342E+04	9.802E+04	1.017E+04	9.120E+03
	Std	6.570E+02	7.759E+03	6.162E+03	6.125E+03	3.991E+04	8.978E+03	6.328E+01	4.481E+04	6.408E+03	6.957E+03	2.086E+04	8.467E+03	7.592E+03
CF_4	Ave	**4.983E+02**	7.256E+02	5.233E+02	5.476E+02	6.483E+02	9.644E+03	4.984E+02	1.156E+03	5.152E+02	1.627E+03	8.134E+02	5.292E+02	1.255E+03
	Std	4.114E+01	6.004E+01	2.491E+01	4.694E+01	5.963E+01	3.190E+03	1.635E+01	6.658E+02	3.664E+01	8.952E+02	3.259E+02	2.934E+01	3.056E+02
CF_5	Ave	5.210E+02	**5.200E+02**	5.210E+02	**5.200E+02**	5.205E+02	5.208E+02	5.209E+02	5.203E+02	**5.200E+02**	5.207E+02	**5.200E+02**	**5.200E+02**	5.209E+02
	Std	6.429E−02	4.406E−04	6.009E−02	1.406E−04	1.389E−01	6.451E−02	5.324E−02	1.640E−01	4.408E−04	1.860E−01	2.905E−02	2.784E−03	7.979E−02
CF_6	Ave	6.158E+02	6.215E+02	6.202E+02	**6.016E+02**	6.362E+02	6.368E+02	6.280E+02	6.228E+02	6.199E+02	6.345E+02	6.351E+02	6.234E+02	6.191E+02
	Std	2.346E+00	2.359E+00	4.693E+00	1.247E+00	3.434E+00	2.265E+00	1.298E+00	3.990E+00	3.997E+00	3.568E+00	3.562E+00	3.328E+00	4.886E+00
CF_7	Ave	**7.000E+02**	7.001E+02	7.001E+02	**7.000E+02**	7.018E+02	1.318E+03	7.002E+02	7.934E+02	7.001E+02	7.697E+02	7.002E+02	7.001E+02	7.202E+02
	Std	1.563E−02	1.046E−03	8.786E−02	2.209E−03	4.548E−01	9.268E+01	6.367E−02	5.185E+01	1.049E−02	4.537E+01	2.162E−01	6.798E−02	1.490E+01
CF_8	Ave	8.455E+02	9.412E+02	9.818E+02	8.426E+02	9.873E+02	1.115E+03	9.022E+02	9.408E+02	9.267E+02	9.832E+02	9.332E+02	8.675E+02	**8.373E+02**
	Std	1.304E+01	1.256E+01	1.041E+01	1.151E+01	3.306E+01	2.511E+01	1.311E+01	3.710E+01	3.533E+01	2.704E+01	4.594E+01	1.105E+01	1.131E+01
CF_9	Ave	9.979E+02	1.066E+03	1.116E+03	**9.374E+02**	1.133E+03	1.189E+03	1.062E+03	1.097E+03	1.033E+03	1.116E+03	1.080E+03	1.014E+03	9.859E+02
	Std	2.157E+01	1.783E+01	1.121E+01	1.154E+01	5.817E+01	2.380E+01	2.000E+01	4.995E+01	3.437E+01	2.801E+01	2.679E+01	2.325E+01	3.996E+01
CF_{10}	Ave	1.674E+03	4.352E+03	6.584E+03	3.198E+03	5.357E+03	6.747E+03	3.665E+03	4.495E+03	4.427E+03	5.364E+03	5.204E+03	2.405E+03	**1.415E+03**
	Std	4.017E+02	4.363E+02	3.537E+02	5.943E+02	8.133E+02	5.277E+02	2.423E+02	9.538E+02	6.180E+02	9.031E+02	1.305E+03	3.816E+02	2.225E+02
CF_{11}	Ave	4.167E+03	5.473E+03	8.072E+03	**4.024E+03**	6.281E+03	7.234E+03	5.024E+03	5.307E+03	4.381E+03	6.468E+03	5.740E+03	4.984E+03	6.058E+03
	Std	5.897E+02	5.978E+02	2.109E+02	7.646E+02	7.817E+02	4.737E+02	2.303E+02	7.549E+02	7.027E+02	7.398E+02	8.605E+02	6.657E+02	1.513E+03
CF_{12}	Ave	**1.200E+03**	**1.200E+03**	1.203E+03	**1.200E+03**	1.202E+03	1.201E+03	1.201E+03	1.200E+03	1.201E+03	1.202E+03	1.201E+03	**1.200E+03**	1.202E+03
	Std	1.013E+00	1.908E−03	2.690E−01	8.054E−03	5.237E−01	4.345E−01	1.443E−01	1.716E−01	3.088E−01	4.563E−01	5.075E−01	1.070E−01	5.345E−01
CF_{13}	Ave	**1.300E+03**	1.301E+03	1.301E+03	1.301E+03	1.301E+03	1.307E+03	1.301E+03	1.302E+03	1.301E+03	1.303E+03	1.301E+03	1.301E+03	1.301E+03
	Std	9.415E−02	5.519E−02	4.746E−02	6.427E−02	1.273E−01	6.730E−01	3.649E−02	1.120E+00	1.145E−01	1.364E+00	1.221E−01	3.714E−02	6.364E−01
CF_{14}	Ave	**1.400E+03**	1.401E+03	1.401E+03	1.401E+03	1.400E+03	1.614E+03	1.401E+03	1.422E+03	1.401E+03	1.442E+03	1.401E+03	1.401E+03	1.414E+03
	Std	5.375E−02	4.495E−02	2.590E−02	5.538E−02	4.930E−02	3.982E+01	2.732E−02	1.656E+01	1.113E−01	2.598E+01	2.440E−01	1.959E−02	1.087E+01
CF_{15}	Ave	1.510E+03	1.507E+03	1.519E+03	**1.504E+03**	1.592E+03	1.830E+05	1.514E+03	1.731E+05	1.509E+03	3.821E+03	1.577E+03	1.532E+03	1.513E+03
	Std	2.981E+00	2.322E+00	1.093E+00	7.715E−01	3.295E+01	7.935E+04	1.590E+00	4.158E+05	3.072E+00	3.139E+03	2.366E+01	8.433E+00	4.417E+00
CF_{16}	Ave	**1.611E+03**	1.614E+03	1.613E+03	1.612E+03	1.613E+03	1.612E+03	1.613E+03	1.612E+03	1.612E+03	1.612E+03	1.613E+03	1.612E+03	1.612E+03
	Std	7.636E−01	2.909E−01	2.460E−01	5.736E−01	4.156E−01	3.915E−01	2.119E−01	5.617E−01	5.438E−01	3.948E−01	3.652E−01	4.780E−01	4.495E−01
CF_{17}	Ave	**7.536E+04**	7.874E+05	1.837E+06	2.614E+05	8.974E+06	4.157E+07	9.659E+04	4.120E+06	2.791E+05	7.103E+06	8.373E+06	1.476E+06	2.111E+06
	Std	6.755E+04	3.128E+05	5.173E+05	1.894E+05	6.279E+06	2.324E+07	4.130E+04	6.234E+06	1.689E+05	7.655E+06	8.404E+06	6.873E+05	2.074E+06
CF_{18}	Ave	3.615E+03	2.308E+03	**2.141E+03**	2.640E+03	2.233E+04	1.780E+08	4.841E+03	3.215E+07	3.704E+03	3.546E+07	5.475E+06	2.515E+03	2.474E+04
	Std	2.100E+03	3.263E+02	4.069E+02	8.483E+02	3.251E+04	3.243E+08	2.180E+03	1.604E+08	1.518E+03	5.761E+07	2.992E+07	7.656E+02	1.034E+05
CF_{19}	Ave	1.915E+03	2.012E+03	**1.907E+03**	1.915E+03	1.965E+03	2.166E+03	1.911E+03	1.947E+03	1.917E+03	2.023E+03	2.031E+03	1.931E+03	1.943E+03
	Std	1.947E+01	3.534E+01	3.134E−01	1.634E+01	4.377E+01	9.771E+01	6.955E−01	4.301E+01	1.278E+01	4.494E+01	7.648E+01	2.638E+01	2.896E+01

(*continued on next page*)

Table 9 (*continued*).

Fun.	Index	QIO	GSA	ABC	ASO	WOA	AOA	CS	MFO	SSA	HHO	LFD	GA	PSO
CF_{20}	Ave	2.887E+03	5.786E+04	1.709E+04	2.159E+04	5.788E+04	1.496E+05	**2.332E+03**	6.941E+04	7.439E+03	6.655E+04	6.650E+04	3.309E+04	1.426E+04
	Std	5.863E+02	1.735E+04	4.943E+03	7.471E+03	4.258E+04	7.577E+04	7.016E+01	4.559E+04	3.697E+03	4.422E+04	3.661E+04	1.516E+04	7.241E+03
CF_{21}	Ave	3.923E+04	2.065E+05	3.190E+05	1.397E+05	2.750E+06	6.787E+06	**6.968E+03**	1.326E+06	1.060E+05	5.060E+06	1.894E+06	3.973E+05	5.232E+05
	Std	2.767E+04	7.930E+04	1.110E+05	1.138E+05	3.024E+06	6.123E+06	9.743E+02	4.476E+06	9.222E+04	5.097E+06	2.010E+06	3.341E+05	5.077E+05
CF_{22}	Ave	2.632E+03	3.184E+03	2.694E+03	2.577E+03	3.101E+03	3.444E+03	**2.562E+03**	2.925E+03	2.675E+03	3.036E+03	3.162E+03	2.885E+03	2.751E+03
	Std	1.561E+02	2.245E+02	8.713E+01	1.639E+02	2.121E+02	4.604E+02	1.146E+02	2.139E+02	1.953E+02	3.998E+02	3.185E+02	2.088E+02	2.072E+02
CF_{23}	Ave	**2.500E+03**	2.538E+03	2.615E+03	2.616E+03	2.649E+03	2.511E+03	2.615E+03	2.658E+03	2.620E+03	**2.500E+03**	2.500E+03	2.615E+03	2.633E+03
	Std	0	6.267E+01	6.706E−02	4.285E−01	2.600E+01	5.644E+01	3.096E−03	2.994E+01	3.595E+00	0	5.286E−05	9.377E−02	7.362E+00
CF_{24}	Ave	**2.600E+03**	**2.600E+03**	2.628E+03	2.620E+03	2.610E+03	**2.600E+03**	2.635E+03	2.672E+03	2.635E+03	2.601E+03	**2.600E+03**	2.630E+03	2.629E+03
	Std	3.141E−05	2.210E−02	7.705E−01	8.801E+00	1.707E+01	5.154E−02	3.673E+00	2.171E+01	8.831E+00	0	4.896E−02	2.239E+00	5.864E+00
CF_{25}	Ave	**2.700E+03**	**2.700E+03**	2.723E+03	2.709E+03	2.716E+03	2.701E+03	2.711E+03	2.715E+03	2.714E+03	**2.700E+03**	2.701E+03	2.713E+03	2.719E+03
	Std	0	1.411E+00	2.141E+00	1.528E+00	1.723E+01	5.232E+00	1.468E+00	8.331E+00	3.122E+00	0	7.711E−07	2.957E+00	4.245E+00
CF_{26}	Ave	2.716E+03	2.791E+03	2.701E+03	**2.700E+03**	2.708E+03	2.766E+03	2.700E+03	2.702E+03	**2.700E+03**	2.753E+03	2.701E+03	2.774E+03	2.705E+03
	Std	3.689E+01	2.334E+01	1.493E−01	5.837E−02	2.728E+01	4.295E+01	4.767E−02	1.043E+00	1.259E−01	4.939E+01	1.579E−01	4.417E+01	1.967E+01
CF_{27}	Ave	**2.900E+03**	4.519E+03	3.054E+03	3.103E+03	3.848E+03	3.769E+03	3.132E+03	3.562E+03	3.106E+03	**2.900E+03**	2.925E+03	3.504E+03	3.302E+03
	Std	0	3.360E+02	1.531E+01	2.020E+00	3.098E+02	5.226E+02	1.014E+01	2.363E+02	2.782E+00	0	1.222E+02	3.201E+02	1.972E+02
CF_{28}	Ave	**3.000E+03**	5.280E+03	4.037E+03	4.398E+03	5.100E+03	4.296E+03	3.843E+03	3.848E+03	4.249E+03	**3.000E+03**	3.262E+03	6.607E+03	6.692E+03
	Std	0	8.873E+02	3.441E+01	8.926E+02	6.169E+02	2.355E+03	4.600E+01	1.362E+02	5.987E+02	0	9.090E+02	8.391E+02	7.760E+02
CF_{29}	Ave	4.101E+03	**3.386E+03**	3.422E+04	6.740E+06	6.942E+06	3.930E+08	7.796E+03	2.378E+06	1.235E+04	6.621E+05	2.173E+06	4.037E+03	7.452E+06
	Std	5.961E+02	9.848E+02	1.589E+04	3.026E+07	5.426E+06	2.028E+08	2.299E+03	3.574E+06	5.883E+03	1.954E+06	1.156E+07	1.413E+02	2.186E+07
CF_{30}	Ave	**4.960E+03**	2.760E+04	1.066E+04	7.372E+03	1.482E+05	4.131E+06	7.632E+03	3.980E+04	1.960E+04	1.920E+05	2.885E+05	7.853E+03	4.621E+04
	Std	1.253E+03	1.797E+04	1.268E+03	1.588E+03	1.086E+05	3.077E+06	9.372E+02	3.418E+04	9.241E+03	1.717E+05	2.663E+05	1.099E+03	3.663E+04

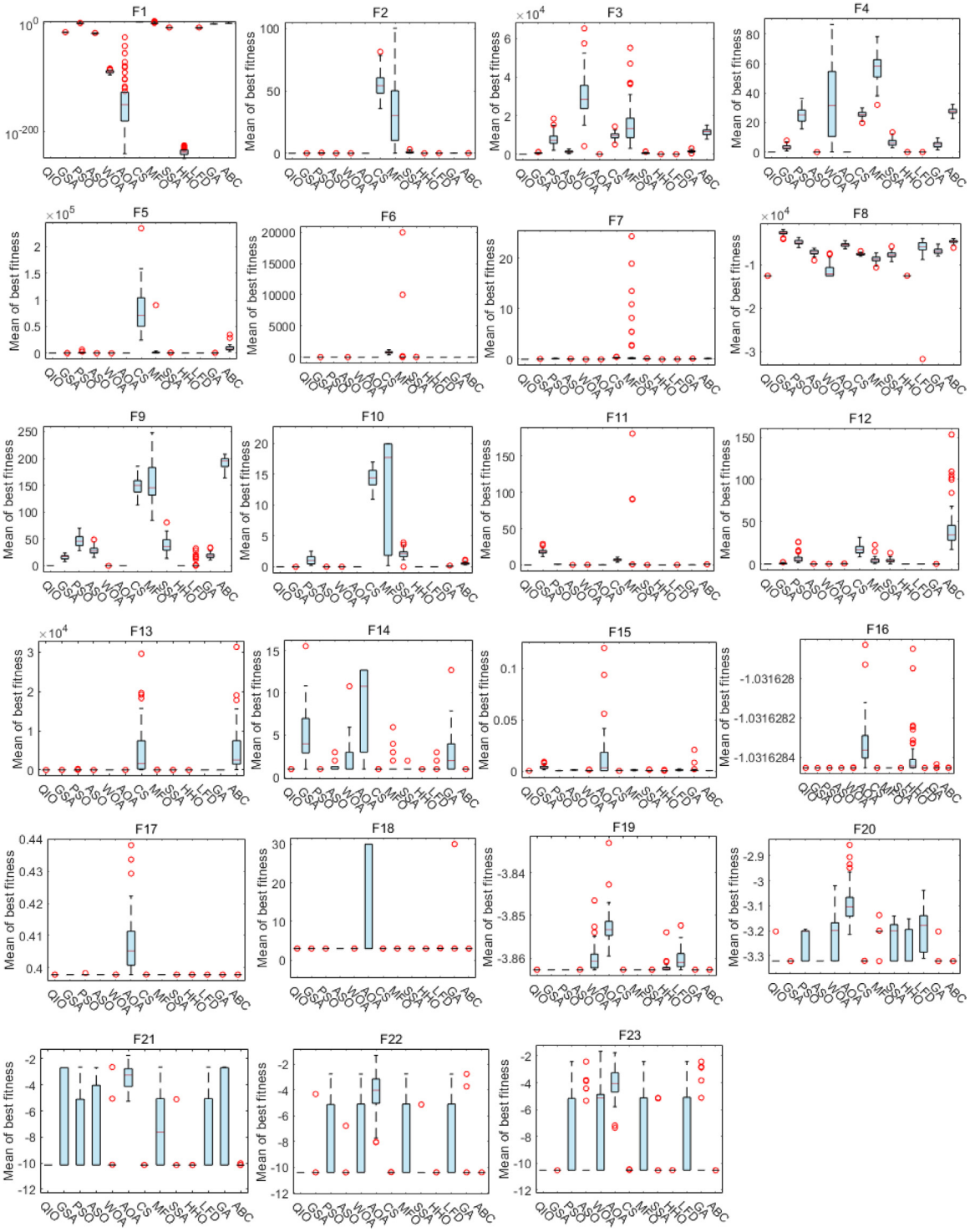

Fig. 14. Boxplots of various algorithms on 23 functions.

CEC-2014 functions is plotted. From Fig. 24, QIO is superior to all other optimizers, with the lowest average rank of 1.23, followed by ASO (1.93) and CS (2.23). In the last place is AOA (6.06). The final ranking of QR offered

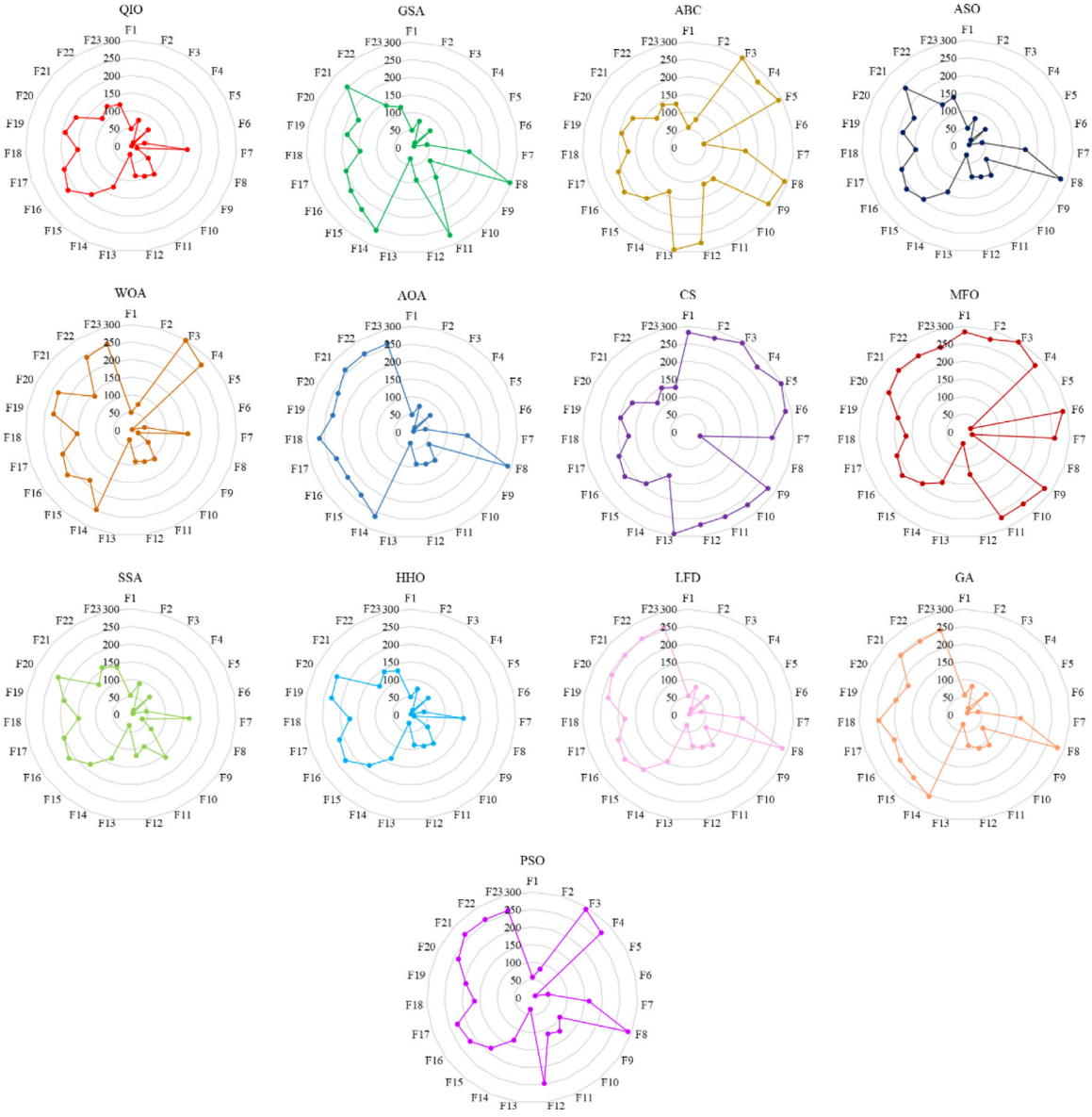

Fig. 15. Radar charts of FAT for QIO and other algorithms on 23 test functions.

by the optimizers on the CEC-2014 functions is QIO > ASO > CS > SSA > GA > GSA > ABC > PSO > LFD > MFO > WOA > HHO > AOA.

In summary, the findings of this section have revealed various characteristics of the proposed QIO algorithm. High exploration ability of QIO is due to the minimizer provided by the GQI method with randomly chosen individuals from the population that contributes to the position updating mechanism in exploration stage. High exploitation ability of QIO is attributed to the minimizer provided by the GQI method with the best individual found so far that is dedicated to the position updating mechanism in exploitation stage. In addition, high local optima avoidance and convergence speed are resulted from the exploration weight. These distinctive search features equally contribute positively in addressing different engineering issues. The next section will assess the performance of QIO on challenging engineering problems.

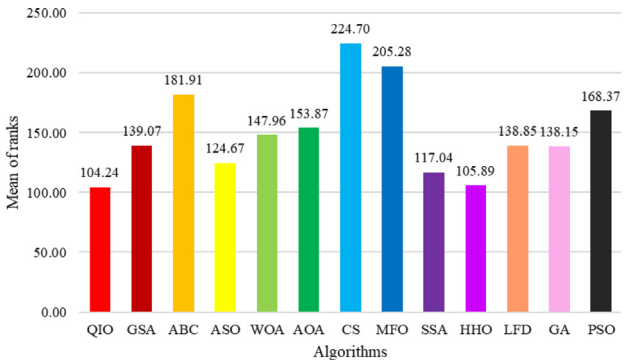

Fig. 16. Average rank of FAT for QIO and other algorithms on 23 test functions.

Table 10
Comparisons of WRST for QIO vs. GSA, ABC, ASO, and WOA.

Fun.	GSA vs. QIO				ABC vs. QIO				ASO vs. QIO				WOA vs. QIO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner	p-value	T−	T+	Winner
CF_1	7.557E−10	0	1275	+	7.557E−10	0	1275	+	3.836E−05	211	1064	+	7.557E−10	0	1275	+
CF_2	1.087E−09	6	1269	+	7.557E−10	0	1275	+	1.227E−09	8	1267	+	7.557E−10	0	1275	+
CF_3	7.557E−10	0	1275	+	7.557E−10	0	1275	+	3.570E−09	26	1249	+	7.557E−10	0	1275	+
CF_4	7.557E−10	0	1275	+	1.851E−03	315	960	+	7.666E−06	174	1101	+	8.031E−10	1	1274	+
CF_5	7.557E−10	1275	0	−	6.206E−03	921	354	−	7.557E−10	1275	0	−	7.557E−10	1275	0	−
CF_6	8.031E−10	1	1274	+	2.298E−06	148	1127	+	7.557E−10	1275	0	−	7.557E−10	0	1275	+
CF_7	7.557E−10	1275	0	−	8.534E−10	2	1273	+	1.586E−08	1223	52	−	7.557E−10	0	1275	+
CF_8	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.844E−01	775	500	≈	7.557E−10	0	1275	+
CF_9	8.031E−10	1	1274	+	7.557E−10	0	1275	+	8.031E−10	1274	1	−	7.557E−10	0	1275	+
CF_{10}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.534E−10	2	1273	+	7.557E−10	0	1275	+
CF_{11}	1.303E−09	9	1266	+	7.557E−10	0	1275	+	3.272E−01	739	536	≈	8.031E−10	1	1274	+
CF_{12}	7.557E−10	1275	0	−	6.048E−05	222	1053	+	7.557E−10	1275	0	−	3.320E−01	738	537	≈
CF_{13}	8.213E−08	1193	82	−	6.259E−01	587	688	≈	9.068E−10	1272	3	−	2.567E−01	520	755	≈
CF_{14}	1.976E−03	958	317	−	3.320E−01	738	537	≈	9.587E−02	465	810	≈	7.179E−03	916	359	−
CF_{15}	2.407E−05	1075	200	−	7.557E−10	0	1275	+	9.068E−10	1272	3	−	7.557E−10	0	1275	+
CF_{16}	7.557E−10	0	1275	+	1.227E−09	8	1267	+	5.366E−09	33	1242	+	7.557E−10	0	1275	+
CF_{17}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.728E−07	128	1147	+	7.557E−10	0	1275	+
CF_{18}	7.157E−04	988	287	−	1.310E−05	1089	186	−	6.881E−02	826	449	≈	8.534E−10	2	1273	+
CF_{19}	7.557E−10	0	1275	+	7.826E−03	913	362	−	2.419E−02	404	871	+	1.130E−08	46	1229	+
CF_{20}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{21}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	5.058E−08	73	1202	+	7.557E−10	0	1275	+
CF_{22}	7.557E−10	0	1275	+	1.258E−02	379	896	+	7.491E−02	822	453	≈	1.155E−09	7	1268	+
CF_{23}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.110E−09	0	1275	+
CF_{24}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{25}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	5.606E−06	0	1275	+
CF_{26}	7.557E−10	0	1275	+	1.573E−01	491	784	≈	7.557E−10	1275	0	−	2.547E−01	498	777	≈
CF_{27}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{28}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.110E−09	0	1275	+
CF_{29}	5.272E−01	703	572	≈	7.557E−10	0	1275	+	1.417E−08	50	1225	+	7.543E−10	0	1275	+
CF_{30}	8.031E−10	1	1274	+	7.557E−10	0	1275	+	5.366E−09	33	1242	+	7.557E−10	0	1275	+

4. Engineering applications of QIO

Engineering optimization problems (EQPs) are characterized by complex and highly constrained conditions, which increase the difficulty of algorithmic solutions. During the solution process, algorithms must consider both the solution precision and the scope of variable space, making their ability to solve a variety of engineering problems an important aspect of algorithm evaluation. To validate QIO's ability to solve real-world problems, 10 classic EQPs are employed in this study.

4.1. 10-bar truss

This design is the minimization of the weight of a 10-bar truss, which has 10 design variables subject to 34 constraints [98]. The cross-sectional areas vary from 0.1 in^2 to 35.0 in^2. The structural diagram of the 10-bar truss is plotted in Fig. 25.

This case is solved by QIO and the aforementioned optimizers, including PSO, ASO, SSA, CS, GA, GSA, ABC, LFD, WOA, MFO, HHO, and AOA, using 50 search individuals and 200 iterations over 30 runs. Table 14 tabulates the statistical comparisons of various optimizers for this case. From Table 14, the results in terms of 'Worst' index provided by QIO are the best, and the 'Best' and 'Ave' indexes rank only second to MFO. Table 15 gives the minimal weight of various optimizers and corresponding design variables. Meanwhile, Fig. 26 offers the

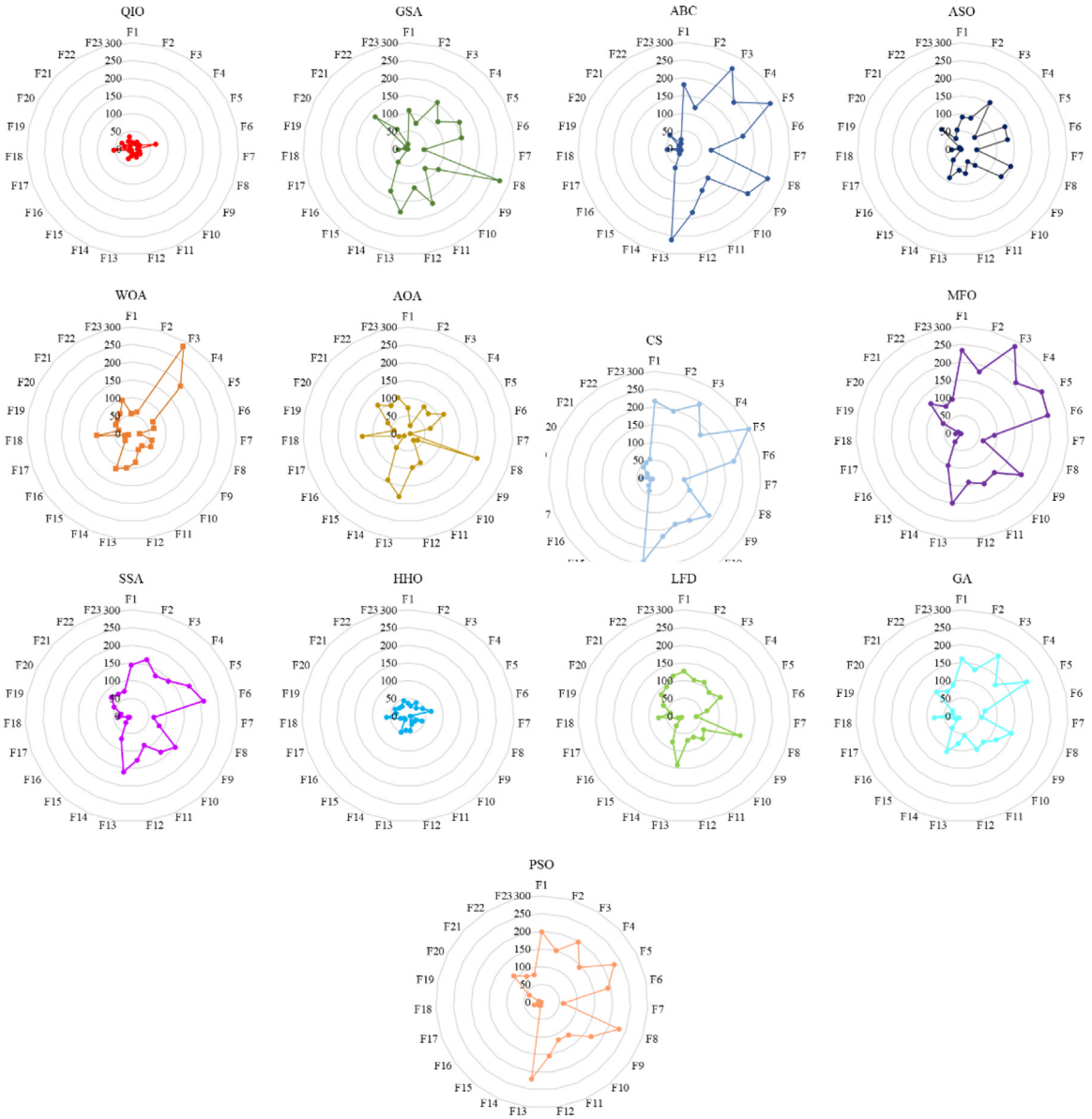

Fig. 17. Radar charts of QR for QIO and other algorithms on 23 test functions.

convergence comparisons of various optimizers. These results demonstrate QIO is very competitive when dealing with this case.

4.2. 25-bar truss design

This design is to minimize the weight of a 25-truss bar [99]. This design has 8 continuous design variables and 80 constraints. The cross-sectional areas of the 25 truss elements are classified into eight groups as shown in Fig. 27, in which different groups are represented by different colors.

This case is tackled by various optimizations using 50 search individuals and 500 iterations over 30 runs. Table 16 provides the statistical comparisons of various optimizers for this case. QIO provides the best results in terms of 'Ave', 'Std', and 'Worst' indexes, showing QIO's superiority over other optimizers in solving this problem.

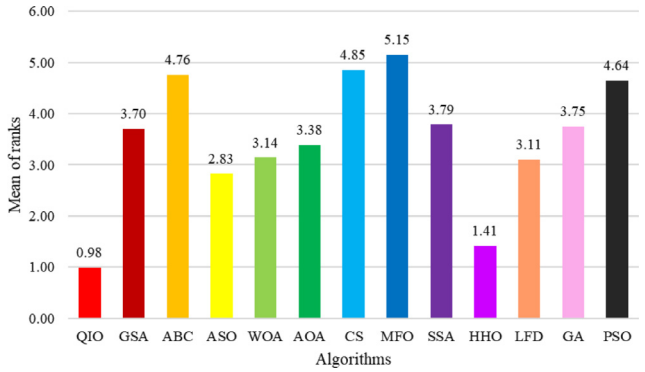

Fig. 18. Average rank of QR for QIO and other algorithms on 23 test functions.

Table 11
Comparisons of WRST for QIO vs. AOA, CS, MFO, and SSA.

Fun.	AOA vs. QIO				CS vs. QIO				MFO vs. QIO				SSA vs. QIO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner	p-value	T−	T+	Winner
CF_1	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_2	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	3.851E−08	68	1207	+
CF_3	7.557E−10	0	1275	+	9.730E−01	634	641	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_4	7.557E−10	0	1275	+	9.884E−01	639	636	≈	1.383E−09	10	1265	+	4.837E−02	433	842	+
CF_5	1.303E−09	1266	9	−	4.563E−07	1160	115	−	7.557E−10	1275	0	−	7.557E−10	1275	0	−
CF_6	7.557E−10	0	1275	+	7.557E−10	0	1275	+	2.661E−09	21	1254	+	4.124E−07	113	1162	+
CF_7	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.464E−01	604	671	≈
CF_8	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.534E−10	2	1273	+	9.068E−10	3	1272	+
CF_9	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.031E−10	1	1274	+	2.089E−06	146	1129	+
CF_{10}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{11}	7.557E−10	0	1275	+	5.686E−09	34	1241	+	5.949E−08	76	1199	+	8.487E−02	459	816	≈
CF_{12}	1.924E−02	880	395	−	4.430E−06	1113	162	−	4.068E−08	1206	69	−	9.637E−08	1190	85	−
CF_{13}	7.557E−10	0	1275	+	4.013E−09	1247	28	−	8.031E−10	1	1274	+	4.428E−01	558	717	≈
CF_{14}	7.557E−10	0	1275	+	2.979E−05	1070	205	−	7.557E−10	0	1275	+	4.602E−01	561	714	≈
CF_{15}	7.557E−10	0	1275	+	2.823E−09	22	1253	+	7.557E−10	0	1275	+	7.537E−01	670	605	≈
CF_{16}	8.534E−10	2	1273	+	7.557E−10	0	1275	+	1.155E−09	7	1268	+	2.479E−08	60	1215	+
CF_{17}	7.557E−10	0	1275	+	5.272E−01	572	703	≈	1.087E−09	6	1269	+	3.092E−08	64	1211	+
CF_{18}	1.303E−09	9	1266	+	3.888E−04	270	1005	+	2.769E−08	62	1213	+	4.517E−02	430	845	+
CF_{19}	7.557E−10	0	1275	+	2.901E−03	329	946	+	1.230E−06	135	1140	+	1.851E−03	315	960	+
CF_{20}	7.557E−10	0	1275	+	3.541E−07	1165	110	−	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{21}	7.557E−10	0	1275	+	1.469E−09	1264	11	−	1.303E−09	9	1266	+	3.854E−06	159	1116	+
CF_{22}	7.557E−10	0	1275	+	1.974E−02	879	396	−	5.339E−08	74	1201	+	2.861E−01	527	748	≈
CF_{23}	5.000E−01	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{24}	1.043E−03	97	1178	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{25}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{26}	2.015E−07	57	1218	+	2.41E−05	1075	200	−	1.04E−02	372	903	+	2.113E−01	508	767	≈
CF_{27}	8.031E−10	1	1274	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{28}	4.883E−04	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{29}	3.530E−09	6	1269	+	7.557E−10	0	1275	+	7.54E−10	0	1275	+	7.557E−10	0	1275	+
CF_{30}	7.557E−10	0	1275	+	9.068E−10	3	1272	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+

Table 17 gives the minimal weight of various optimizers and corresponding design variables. The good convergence performance of QIO is highlighted in Fig. 28 based on the convergence comparisons of various optimizers.

4.3. 72-bar truss design

This design needs to minimize the weight of the 72-bar truss, which involves 16 design variables subjected to 198 constraints. 72 truss elements are organized into 16 groups, and each group is represented by a different color in Fig. 29, in which the topology structure is illustrated. The cross-section areas are described in the literature [100].

This case is handled with various optimizers using 50 search individuals and 1000 iterations over 30 runs. The statistical comparisons of various optimizers for this case are tabulated in Table 18. From Table 18, QIO provides the best results concerning the 'Ave', 'Std', 'Worst' and 'Best' indexes, showing the competitiveness of QIO for this challenging problem. Table 19 presents the minimal weight of various optimizers and corresponding design variables. Fig. 30 shows the convergence comparisons of various optimizers for this case.

4.4. Robot gripper design

The aim of this design requires minimizing the difference between the minimal and maximal forces generated by the robot gripper as shown in Fig. 31 [101]. This design has seven design variables and six constraints related

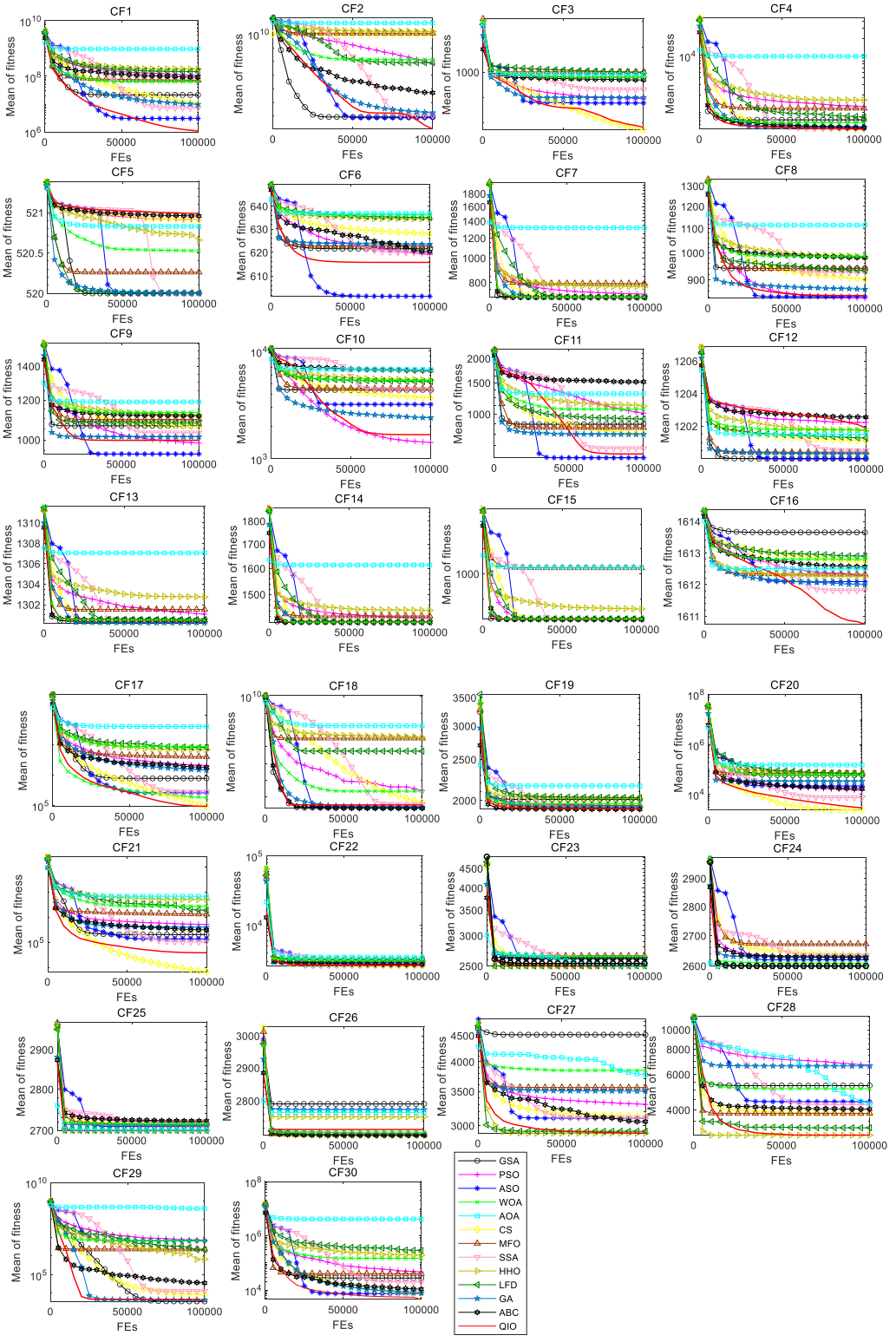

Fig. 19. Convergence curves of QIO and other optimizers on the CEC-2014 suite.

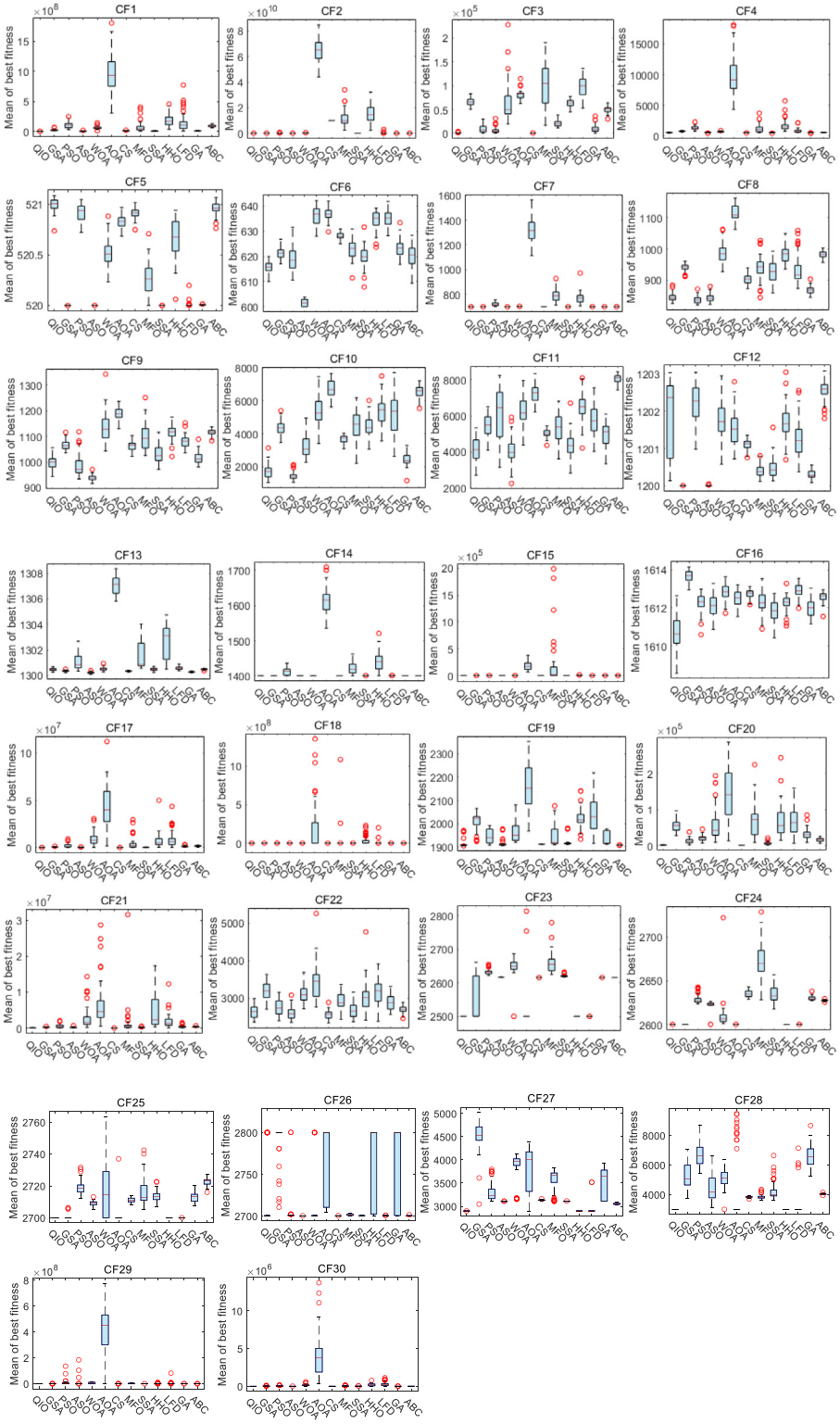

Fig. 20. Boxplots of various algorithms on the CEC-2014 suite.

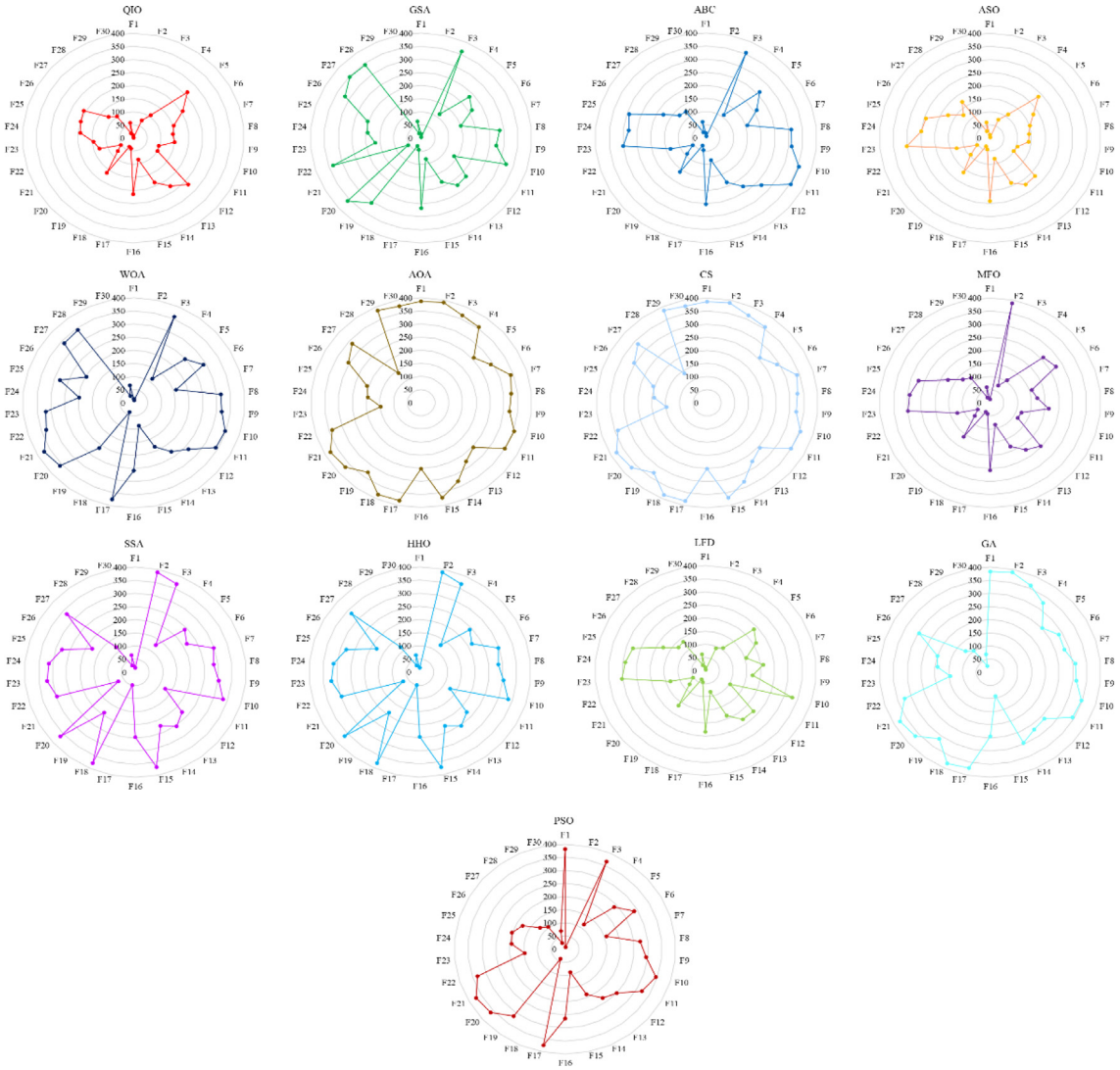

Fig. 21. Radar charts of FAT for QIO and other algorithms on CEC-2014 suite.

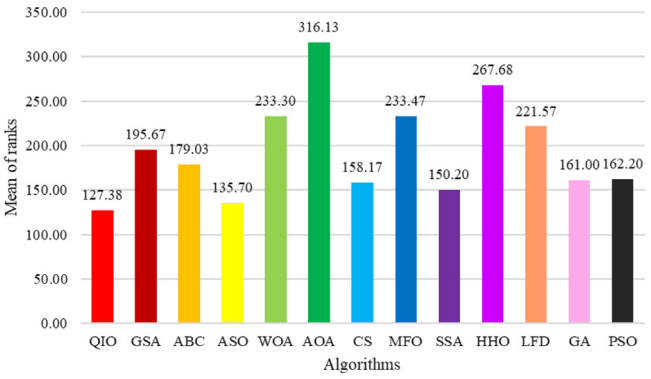

Fig. 22. Average rank of FAT for QIO and other algorithms on CEC-2014 suite.

Table 12
Comparisons of WRST for QIO vs. HHO, LFD, GA, and PSO.

Fun.	HHO vs. QIO				LFD vs. QIO				GA vs. QIO				PSO vs. QIO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner	p-value	T−	T+	Winner
CF_1	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_2	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.031E−10	1	1274	+
CF_3	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.979E−09	16	1259	+	1.087E−09	6	1269	+
CF_4	7.557E−10	0	1275	+	8.031E−10	1	1274	+	7.717E−05	228	1047	+	7.557E−10	0	1275	+
CF_5	7.557E−10	1275	0	−	7.557E−10	1275	0	−	7.557E−10	1275	0	−	1.199E−05	1091	184	−
CF_6	7.557E−10	0	1275	+	7.557E−10	0	1275	+	1.087E−09	6	1269	+	8.035E−05	229	1046	+
CF_7	7.557E−10	0	1275	+	3.785E−09	27	1248	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_8	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.159E−09	38	1237	+	2.556E−03	950	325	−
CF_9	7.557E−10	0	1275	+	7.557E−10	0	1275	+	6.562E−05	224	1051	+	1.689E−02	885	390	−
CF_{10}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	5.686E−09	34	1241	+	3.888E−04	1005	270	−
CF_{11}	7.557E−10	0	1275	+	2.100E−09	17	1258	+	7.161E−07	124	1151	+	1.191E−07	89	1186	+
CF_{12}	2.332E−01	761	514	≈	5.388E−04	996	279	−	1.130E−08	1229	46	−	2.043E−01	506	769	≈
CF_{13}	8.031E−10	1	1274	+	8.232E−04	291	984	+	1.024E−09	1270	5	−	2.926E−08	63	1212	+
CF_{14}	9.068E−10	3	1272	+	5.584E−04	280	995	+	7.557E−10	1275	0	−	6.381E−09	36	1239	+
CF_{15}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.364E−05	230	1045	+
CF_{16}	1.087E−09	6	1269	+	7.557E−10	0	1275	+	6.024E−09	35	1240	+	1.865E−09	15	1260	+
CF_{17}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	8.534E−10	2	1273	+
CF_{18}	7.557E−10	0	1275	+	5.837E−06	168	1107	+	1.924E−02	880	395	−	1.909E−01	502	773	≈
CF_{19}	8.031E−10	1	1274	+	4.013E−09	28	1247	+	3.242E−05	207	1068	+	3.679E−06	158	1117	+
CF_{20}	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{21}	7.557E−10	0	1275	+	9.068E−10	3	1272	+	8.031E−10	1	1274	+	2.661E−09	21	1254	+
CF_{22}	9.138E−08	84	1191	+	5.064E−09	32	1243	+	2.349E−07	102	1173	+	4.749E−03	345	930	+
CF_{23}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{24}	2.925E−04	153	1122	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{25}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{26}	1.553E−06	110	1165	+	3.566E−01	542	733	≈	3.451E−08	66	1209	+	1.038E−02	372	903	+
CF_{27}	1	0	1274	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{28}	1	0	1275	≈	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+
CF_{29}	1.631E−01	153	1122	≈	2.229E−09	18	1257	+	1.253E−05	185	1090	+	5.064E−09	32	1243	+
CF_{30}	9.773E−09	36	1239	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+	7.557E−10	0	1275	+

Table 13
WRST of all the algorithms for CEC-2014 functions.

Function type	GSA vs. QIO (+/≈/−)	ABC vs. QIO (+/≈/−)	ASO vs. QIO (+/≈/−)	WOA vs. QIO (+/≈/−)	AOA vs. QIO (+/≈/−)	CS vs. QIO (+/≈/−)
Unimodal	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	2/1/0
Multimodal	7/0/6	10/2/1	3/3/7	9/2/2	11/0/2	8/1/4
Hybrid	5/0/1	4/0/2	4/2/0	6/0/0	6/0/0	2/1/3
Composition	7/1/0	7/1/0	7/0/1	7/1/0	6/2/0	7/0/1
Total	22/1/7	24/3/3	17/5/8	25/3/2	26/2/2	19/3/8
Function type	MFO vs. QIO (+/≈/−)	SSA vs. QIO (+/≈/−)	HHO vs. QIO (+/≈/−)	LFD vs. QIO (+/≈/−)	GA vs. QIO (+/≈/−)	PSO vs. QIO (+/≈/−)
Unimodal	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0
Multimodal	11/0/2	6/5/2	11/1/1	11/0/2	9/0/4	8/1/4
Hybrid	6/0/0	5/1/0	6/0/0	6/0/0	5/0/1	5/1/0
Composition	8/0/0	7/1/0	3/5/0	7/1/0	8/0/0	8/0/0
Total	28/0/2	21/7/2	23/6/1	27/1/2	25/0/5	24/2/4

to the robot. This design is formulated below.

Consider variables: $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [a,\ b,\ c,\ e,\ f,\ l,\ \delta]$

Minimize: $f_4(\vec{x}) = -\min_z F_k(\vec{x}, z) + \max_z F_k(\vec{x}, z)$

Subject to:

$$g_1(\vec{x}) = -Y_{\min} + y(\vec{x}, Z_{\max}) \le 0,$$

$$g_2(\vec{x}) = -y(\vec{x}, Z_{\max}) \le 0,$$

$$g_3(T) = Y_{\max} - y(T, 0) \le 0,$$

$$g_4(T) = y(T, 0) - Y_G \le 0,$$

$$g_5(\vec{x}) = l^2 + e^2 - (a+b)^2 \le 0,$$

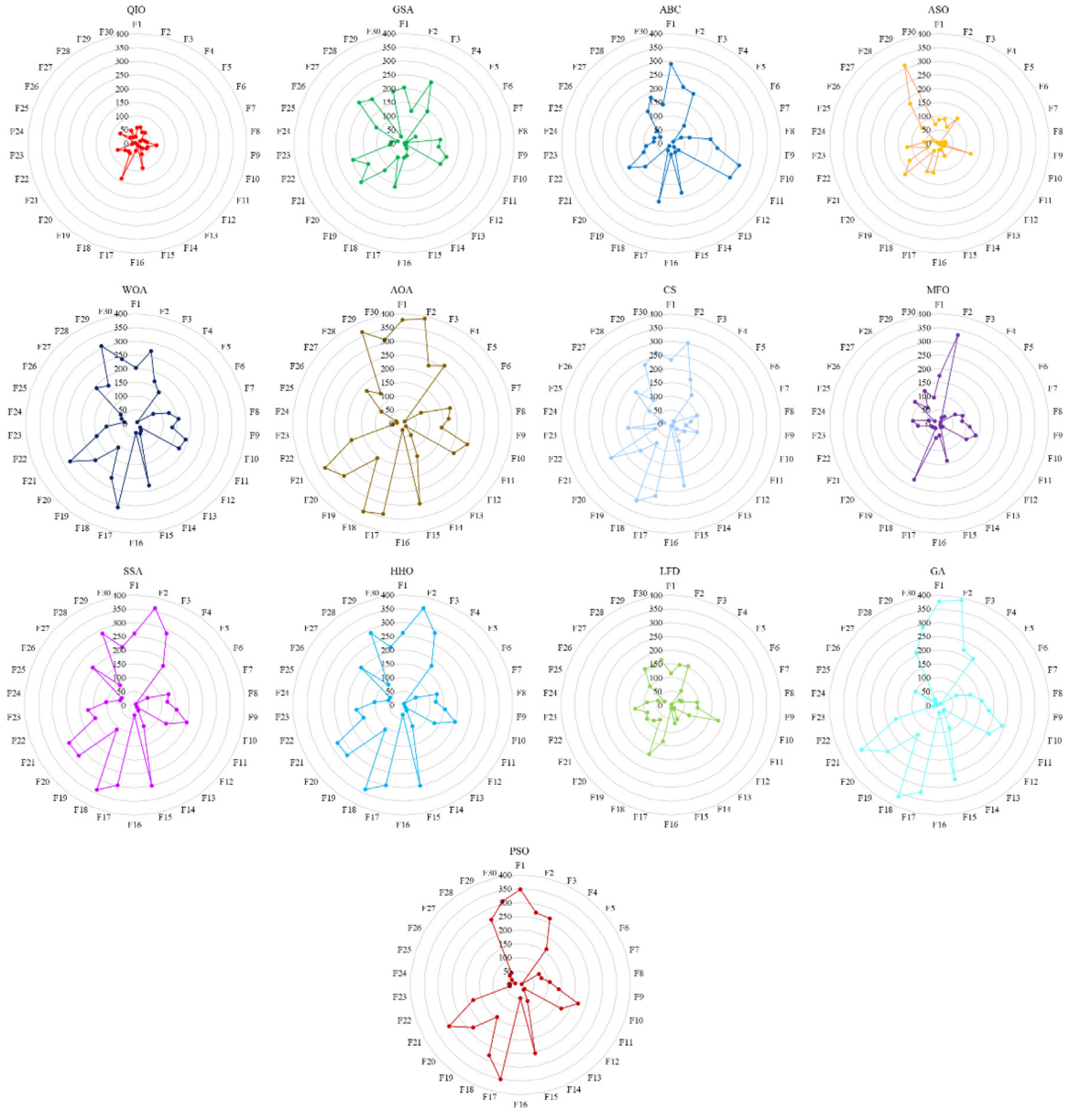

Fig. 23. Radar charts of QR for QIO and other algorithms on CEC-2014 suite.

$$g_6(\vec{x}) = b^2 - (a-e)^2 - (l - Z_{\max})^2 \leq 0,$$

$$g_7(\vec{x}) = Z_{\max} - l \leq 0,$$

where,

$$F_k = \frac{Pb\sin(\alpha+\beta)}{2c\cos(\alpha)},$$

$$\alpha = \cos^{-1}\left(\frac{a^2+g^2-b^2}{2ag}\right) + \phi,\ g = \sqrt{e^2+(z-l)^2},$$

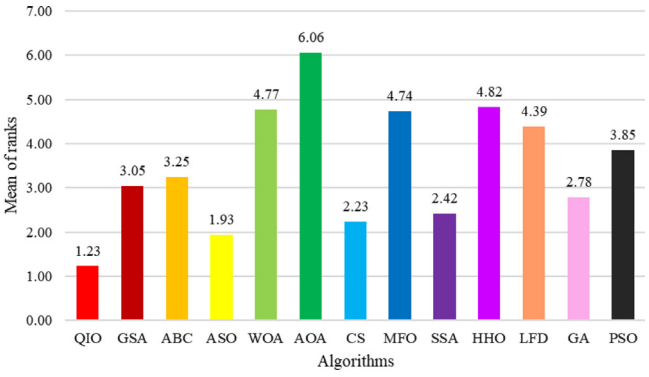

Fig. 24. Average rank of QR for QIO and other algorithms on CEC-2014 suite.

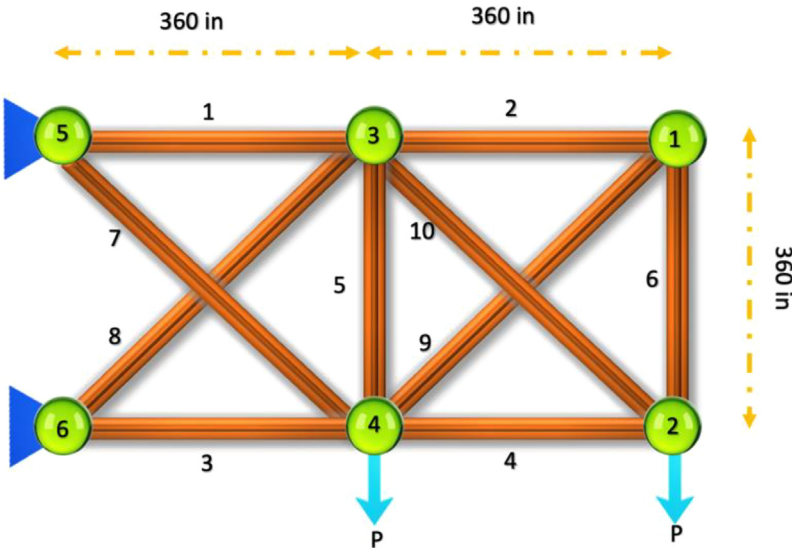

Fig. 25. 10-bar truss design.

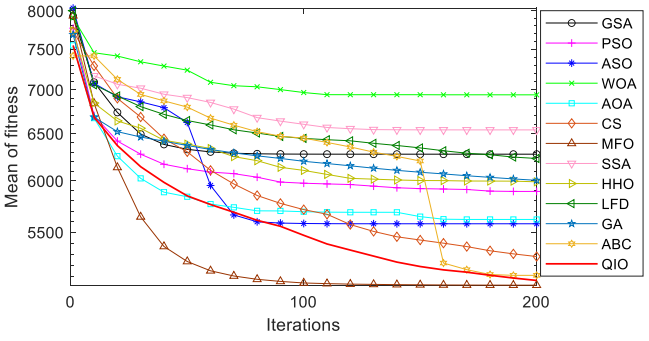

Fig. 26. Convergence comparisons of various optimizers for 10-bar truss design.

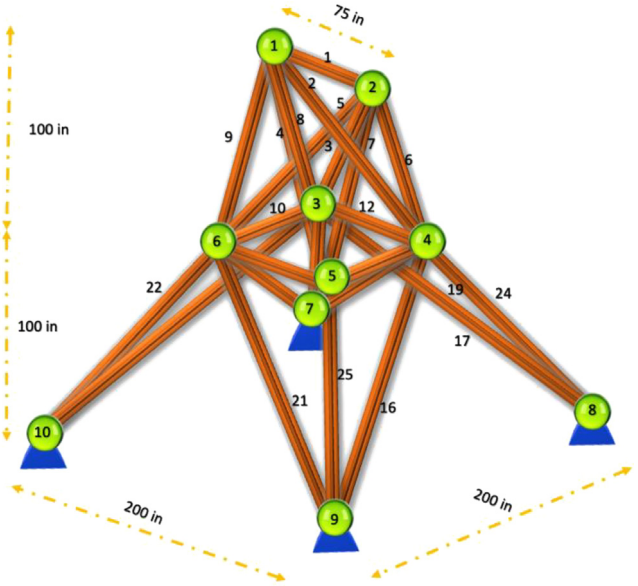

Fig. 27. 25-bar truss design. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 14
Statistical comparisons of various optimizers for 10-bar truss design.

Algorithms	Ave	Std	Worst	Best
QIO	5.0686E+03	2.5798E+01	**5.1514E+03**	5.0152E+03
GSA	6.2772E+03	1.7662E+02	6.6504E+03	5.8412E+03
ABC	5.1126E+03	**2.3980E+01**	5.1845E+03	5.0714E+03
ASO	5.5793E+03	1.6885E+02	5.9265E+03	5.2807E+03
WOA	6.9410E+03	6.0670E+02	7.9332E+03	5.4621E+03
AOA	5.6202E+03	3.0165E+02	6.8219E+03	5.2770E+03
CS	5.2779E+03	1.0604E+02	5.6440E+03	5.1006E+03
MFO	**5.0281E+03**	3.3545E+01	5.1721E+03	**5.0031E+03**
SSA	6.5408E+03	4.1558E+02	7.1562E+03	5.5711E+03
HHO	5.9904E+03	3.9208E+02	6.6127E+03	5.0969E+03
LFD	6.2319E+03	3.6380E+02	6.9640E+03	5.3981E+03
GA	6.0054E+03	1.1541E+02	6.2347E+03	5.7747E+03
PSO	5.8935E+03	1.5146E+02	6.2489E+03	5.5952E+03

$$\beta = \cos^{-1}\left(\frac{b^2+g^2-a^2}{2bg}\right) - \phi, \phi = \tan^{-1}\left(\frac{e}{l-z}\right),$$

$$y(\vec{x}, z) = 2(f + e + c\sin(\beta + \delta)),$$

$$Y_{\min} = 50, Y_{\max} = 100, YG = 150, Z_{\max} = 100, P = 100.$$

Variable range:

$$0 \le e \le 50,\ 100 \le c \le 200,\ 10 \le f, a, b \le 150, 1 \le \delta \le 3.14,\ 100 \le l \le 300.$$

This design is solved by different optimizers using 50 search individuals and 500 iterations over 30 runs. The statistical comparisons of various optimizers for this case are tabulated in Table 20. Among all the algorithms, QIO provides the best results in terms of the 'Ave', 'Std', 'Best' and 'Worst' indexes. Table 21 demonstrates the best values

Table 15
The best results of various optimizers for 10-bar truss design.

Algorithms	The optimal variables										
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Minimum weight
QIO	29.1886965	0.0330574	23.0034491	14.6344968	0.0261975	0.2281041	21.0461689	7.6668043	0.0135288	22.3013911	5.0152E+03
GSA	26.2981273	6.6646761	27.3222181	12.2350523	0.1363922	5.5964092	16.2317908	17.9717367	8.3850376	16.8102162	5.8412E+03
ABC	30.7412409	0.2099113	21.1712350	15.4644322	0.0254516	0.6264677	21.0797512	8.2924199	0.0580408	21.6326338	5.0714E+03
ASO	29.2363019	1.0921368	23.4793756	15.9840211	0.0100003	1.2970192	20.0052728	9.9153520	2.2426106	21.2858225	5.2807E+03
WOA	22.7824716	0.0100000	24.9521233	17.4255136	0.1699231	0.0100000	28.2336730	12.3357698	0.2371570	20.2694394	5.4621E+03
AOA	32.6277058	1.3559157	23.2485266	10.9514860	0.0100000	0.0100000	27.0880065	8.5410995	0.0100000	19.6383165	5.2770E+03
CS	30.6662151	0.0100000	25.3880913	15.8390285	0.0100000	0.0100000	18.8516122	8.3419831	0.0100000	22.1241623	5.1006E+03
MFO	30.3193964	0.0100000	23.1971330	15.1824033	0.0101436	0.1435826	20.6975972	7.6875731	0.0100000	21.1810263	5.0031E+03
SSA	22.7249315	1.3594022	22.8029057	22.5976643	0.7259090	2.6142346	29.1365647	6.0728725	0.0155437	22.7064052	5.5711E+03
HHO	32.8603939	0.0554133	24.0371031	16.6224218	0.0122586	0.0219708	17.8073484	8.6419021	0.0519482	21.5495821	5.0969E+03
LFD	30.2394887	5.1768975	22.7878277	22.9827846	0.1060919	0.0166191	20.2101255	7.3580866	0.0100000	20.9563707	5.3981E+03
GA	27.8052437	4.7623401	24.4228792	14.4013393	0.8168436	4.4596531	21.0677047	15.2159998	6.0278197	16.9024222	5.7747E+03
PSO	26.3334837	2.0757057	24.0080037	19.0031288	0.1025524	5.1308804	21.8659439	10.6023351	2.5249518	18.7987737	5.5952E+03

Table 16
Statistical results of different algorithms for 25-bar truss problem.

Algorithms	Ave	Std	Worst	Best
QIO	**545.6074586**	**0.3538586**	**546.7043446**	545.1934428
GSA	583.9984267	8.7473712	595.1637156	565.8716533
ABC	547.2910112	0.4343070	548.1843348	546.4546865
ASO	549.5928363	3.2282366	557.6796099	546.0313307
WOA	599.7192087	16.7894846	625.7325571	565.9045543
AOA	573.4589790	13.6577708	596.7095359	556.5084140
CS	546.3386957	0.4554707	546.9210279	545.6943935
MFO	546.8955263	1.0925511	549.3590374	**545.1928994**
SSA	600.3560921	13.9249328	634.5546314	582.4055717
HHO	578.5423773	16.0026121	609.6549583	559.4158482
LFD	576.0569446	12.6293456	604.6190521	552.8448660
GA	569.4516503	11.5236906	591.9381408	547.4615456
PSO	558.8448076	6.7657568	583.5699198	551.4816365

of various optimizers and corresponding design variables. These results verify that QIO is the most effective and stable for this problem. Fig. 32 shows the convergence comparisons of various optimizers for this case.

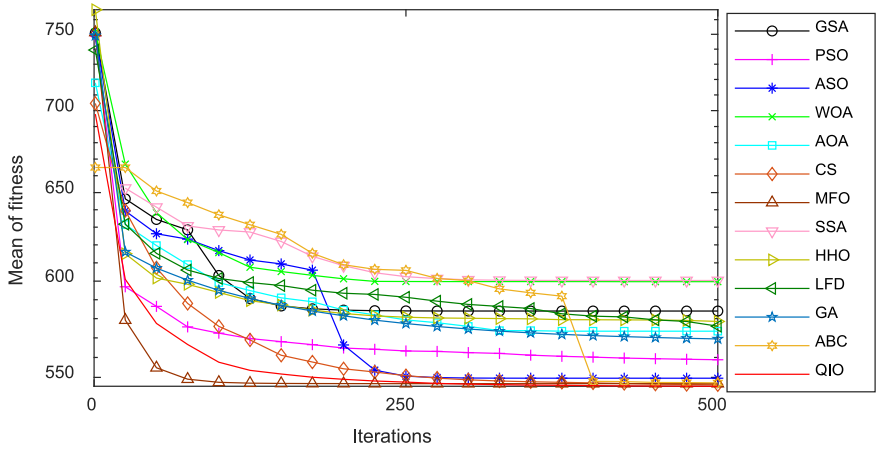

Fig. 28. Convergence comparisons of various optimizers for 25-bar truss design.

Table 17
The best results of various optimizers for 25-bar truss.

Algorithms	Optimum variables								Optimal weight
	A_1	A_2–A_5	A_6–A_9	A_{10}–A_{11}	A_{12}–A_{13}	A_{14}–A_{17}	A_{18}–A_{21}	A_{22}–A_{25}	
QIO	1.0861E−02	2.0066012	2.9669368	1.0149E−02	1.0182E−02	0.6749451	1.6733035	2.6817259	545.1934428
GSA	0.6125294	2.1502008	2.7907194	0.3358992	0.7645674	0.7036458	1.6407087	2.6869280	565.8716533
ABC	0.0185169	1.9571372	2.9739075	0.0217181	2.0552E−02	0.7413109	1.7150647	2.5941458	546.4546865
ASO	0.0839581	1.9010515	3.1555759	1.0074E−02	1.5566E−02	0.6839097	1.6785826	2.6185843	546.0313307
WOA	0.4597509	2.1898804	2.9495375	1.0000E−02	2.8737E−02	0.9939359	1.6988080	2.3685903	565.9045543
AOA	0.3477283	2.0971322	2.8369685	2.7609E−02	4.6527E−02	0.6241943	1.6604082	2.9004917	556.5084140
CS	1.0041E−02	1.9324527	3.0036940	1.0000E−02	1.0000E−02	0.6981749	1.7264973	2.6306830	545.6943935
MFO	0.0100000	2.0133776	2.9549169	1.0000E−02	1.0000E−02	0.6765247	1.6732164	2.6828964	545.1928994
SSA	1.4450017	1.9312483	2.7883411	0.9774755	0.5729667	0.7721428	1.8558284	2.5843159	582.4055717
HHO	0.2348522	2.0905915	2.7903722	0.4783327	0.3085292	0.6305373	1.6977809	2.7875044	559.4158482
LFD	0.0511892	2.1306547	2.7430600	0.4157754	1.0000E−02	0.7249780	1.6862577	2.6778238	552.8448660
GA	0.0590838	2.1572682	2.8048955	2.2289E−02	6.2974E−02	0.7358137	1.6325822	2.6541271	547.4615456
PSO	0.2740680	2.0200304	2.8980314	2.8986E−02	6.2799E−02	0.6153333	1.7279973	2.7911433	551.4816365

Table 18
Statistical results of different algorithms for 72-bar truss problem.

Algorithms	Ave	Std	Worst	Best
QIO	**380.2559829**	**0.4357897**	**381.7138235**	**379.9070326**
GSA	430.9253222	19.2627048	456.0529148	391.3078022
ABC	384.3812264	0.7834576	385.4930617	382.3171368
ASO	388.8457997	6.2728934	409.8959893	383.4151068
WOA	537.4171846	50.7373861	635.0556741	456.2465289
AOA	523.5316526	27.6680684	566.0815431	473.6146637
CS	387.1670368	1.8740521	391.1869256	384.6434380
MFO	383.3871404	8.7278480	412.5305493	379.9277288
SSA	497.3095200	28.0770190	555.9773179	453.0614132
HHO	414.0763509	10.1569264	433.8282000	395.1887632
LFD	621.3831747	61.3624102	724.9603494	521.5035972
GA	411.1723462	9.3028744	434.2426066	397.4199674
PSO	549.6125779	22.1859184	597.3092202	503.9577610

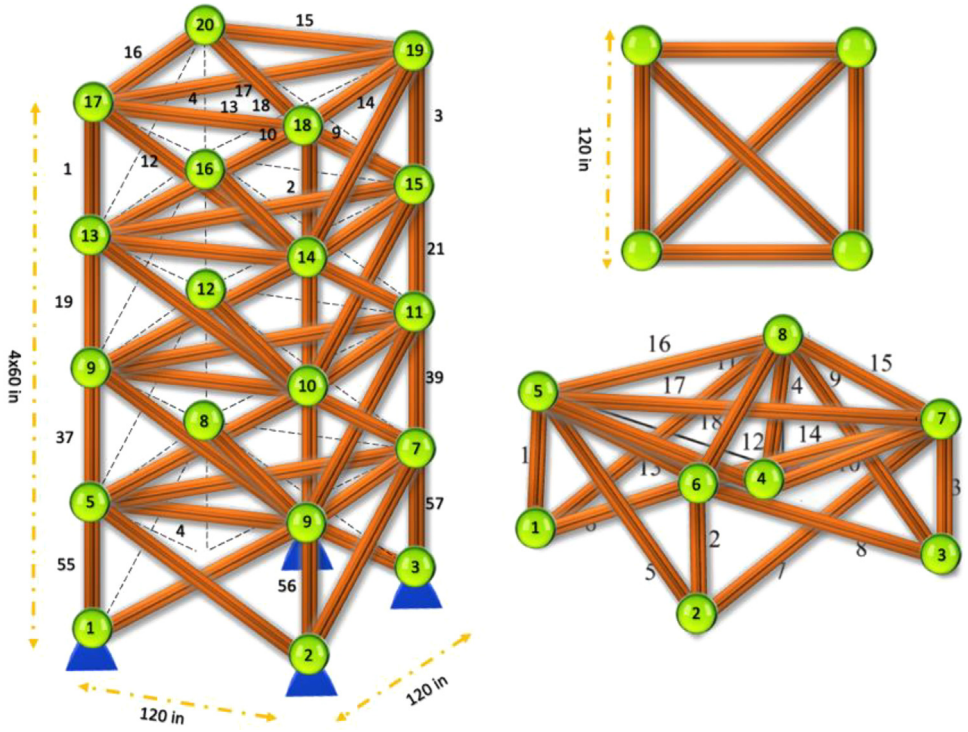

Fig. 29. 72-bar truss design. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

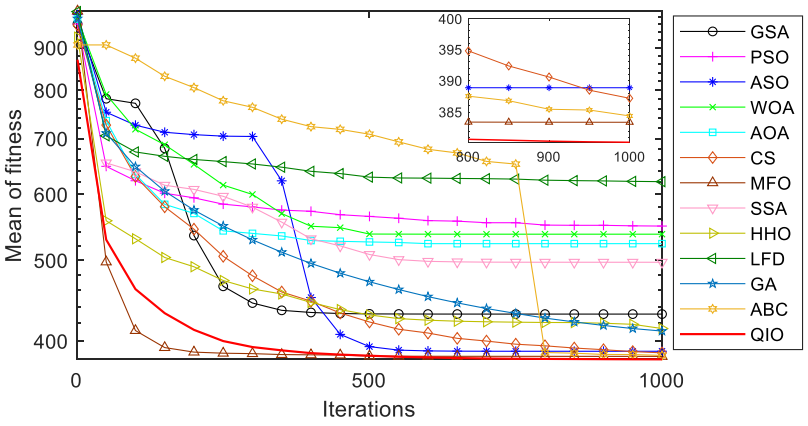

Fig. 30. Convergence comparisons of various optimizers for 72-bar truss design.

4.5. Wind farm layout design

This design is the maximization of the total power output by finding the best arrangement of wind turbines. This problem has 30 design variables and 91 constraints. The schematic diagram of this design is plotted in Fig. 33 [102]. This design is formulated below.

Consider variable: $\vec{x} = [x_1, y_1, x_2, y_2, \ldots, x_{15}, y_{15}]$

Minimize: $f_5 = \sum_{k=1}^{Num} E(P_i)$

Table 19
The best results of various optimizers for 72-bar truss design.

Algorithms	The optimal variables							
	A_1–A_4	A_5–A_{12}	A_{13}–A_{16}	A_{17}–A_{18}	A_{19}–A_{22}	A_{23}–A_{30}	A_{31}–A_{34}	A_{35}–A_{36}
QIO	1.9401862	0.5178724	0.1002806	0.1001750	1.2486741	0.5177549	0.1002704	0.1001072
GSA	1.4237437	0.5422449	0.1335026	0.1802400	1.1600634	0.5230415	0.1005232	0.1011894
ABC	1.8138396	0.5095561	0.1005103	0.1020865	1.3231825	0.5105348	0.1053215	0.1159057
ASO	1.8319118	0.5051381	0.1000000	0.1030277	1.2822887	0.5337935	0.1001059	0.1368918
WOA	1.8367769	0.3596404	0.1000000	0.4261001	1.4566853	0.8357806	0.5388205	0.1000000
AOA	1.5860520	0.7287457	0.1000000	0.1000000	2.4344585	0.6679475	0.1394153	0.4746174
CS	1.9043883	0.4573756	0.1001072	0.1045353	1.4801137	0.4880890	0.1013418	0.1214117
MFO	1.8056575	0.5232580	0.1000000	0.1000000	1.3232788	0.5017512	0.1000000	0.1000000
SSA	1.2848525	0.5579596	0.5720904	0.2543010	1.0370116	0.5186680	0.1494662	0.4683295
HHO	1.9689201	0.4967143	0.1025647	0.1017821	1.3203611	0.5570487	0.1033731	0.1488057
LFD	2.8485391	0.7807121	0.3387012	0.3832805	1.0777835	1.0828232	0.2198574	0.1000000
GA	1.5674284	0.5635221	0.1399762	0.1653959	1.1135965	0.5625433	0.1353816	0.1579601
PSO	1.6567755	0.4889987	0.5740014	0.6747004	1.7816748	0.6505021	0.2451366	0.5457956

The optimal variables								Optimal weight
A_{37}–A_{40}	A_{41}–A_{48}	A_{49}–A_{52}	A_{53}–A_{54}	A_{55}–A_{58}	A_{59}–A_{66}	A_{67}–A_{70}	A_{71}–A_{72}	
0.5418515	0.4965517	0.1023624	0.1002696	0.1562280	0.5390976	0.4179843	0.5746300	379.9070326
0.6333714	0.5517246	0.1045849	0.2502410	0.1459776	0.5804760	0.4307890	0.5820763	391.3085323
0.5175097	0.5221645	0.1011041	0.1297197	0.1567350	0.5521023	0.4575992	0.5172855	382.3171368
0.6692460	0.5313284	0.1000036	0.1821922	0.1522834	0.5331005	0.3623242	0.5031504	383.4151068
0.4409186	0.5579720	0.1000000	0.1000000	0.1613070	0.7523353	0.1000000	0.9514446	456.2465289
0.3053626	0.5882331	0.6389234	0.1000000	0.4076947	0.3642331	0.4746174	0.5901870	473.6146637
0.5529601	0.5227659	0.1112567	0.1036826	0.1612295	0.5359782	0.4376834	0.7057489	384.6434380
0.5214697	0.5331915	0.1000000	0.1000000	0.1564675	0.5532274	0.3987278	0.5367898	379.9277288
0.7306978	0.4914872	0.2271047	0.4408219	0.1331255	0.5754519	0.8229281	0.6484178	453.0614132
0.5103138	0.5403428	0.3392767	0.1017290	0.1450639	0.6015884	0.2531290	0.4293259	395.1887632
0.5597128	0.5382016	0.1471948	0.8359917	0.4454910	0.4549304	0.2631097	0.1548045	521.5035972
0.5908421	0.4909744	0.1453175	0.1620759	0.1702129	0.5641258	0.3720013	0.7712830	397.4199674
0.9109504	0.4221485	0.2033188	0.1455643	0.9216124	0.5425457	0.6075384	0.7955581	503.9577610

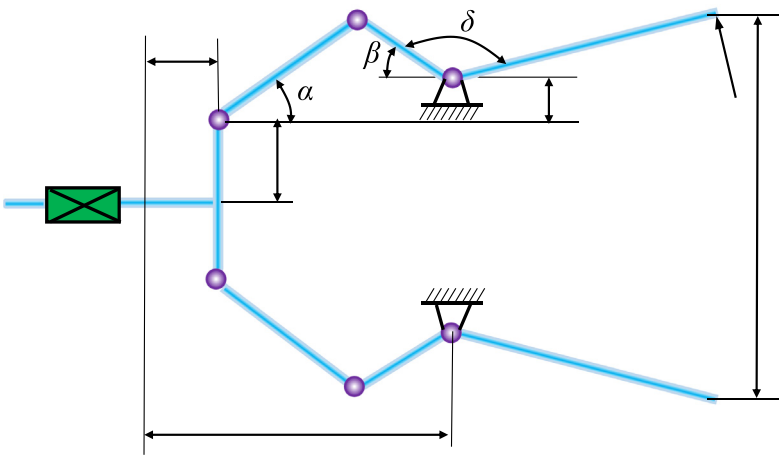

Fig. 31. Robot gripper design.

Table 20
Statistical comparisons of various optimizers for the robot gripper problem.

Algorithms	Ave	Std	Worst	Best
QIO	**3.4929426**	**0.3569469**	**4.5543675**	**2.7828561**
GSA	3.5505E+15	1.0390E+16	5.2151E+16	9.2130627
ABC	4.8727387	0.2926673	5.4941513	4.0395813
ASO	6.3402763	0.8216475	8.3466083	4.8659641
WOA	9.7248666	14.9861127	65.0257800	3.8081790
AOA	4.8968986	1.2715207	8.3509532	3.3503232
CS	4.2019486	0.3753542	4.9222962	3.5516990
MFO	4.4971994	1.0378875	7.8746786	3.2546976
SSA	6.1378197	2.2693935	17.5707749	4.7880091
HHO	5.3983024	1.0414580	7.3786107	3.4196237
LFD	8.3891681	1.9296107	13.3156789	3.9101673
GA	9.3873953	16.4572219	95.8106032	4.0901750
PSO	5.4879817	0.7675585	7.2841848	4.0930221

Table 21
The best results of various optimizers for robot gripper design.

Algorithms	Optimal variables							Optimal difference
	a	b	c	e	f	l	δ	
QIO	147.5413658	134.8737966	199.3192382	12.4714172	128.9446800	105.2004161	2.3079740	2.7828561
GSA	119.3464919	90.4369005	124.1457146	21.8548050	85.2610796	151.3567705	2.6651720	9.2130627
ABC	145.4387272	129.0542607	182.6825732	15.0137100	137.9261705	137.3948114	2.5696253	4.0395813
ASO	132.7789170	117.7366634	147.8799999	14.4450835	190.7089461	117.2790806	2.3953939	4.8659641
WOA	139.2983204	137.0534597	185.7221159	0	128.5680722	144.7537830	2.4483256	3.8081790
AOA	150.0000000	148.4608994	200.0000000	0	150	143.7198554	2.4513675	3.3503232
CS	150.0000000	148.2718646	200.0000000	0	150	147.5286370	2.5193020	3.5516990
MFO	150.0000000	123.3208762	200.0000000	25.6936652	132.7845553	128.3961569	2.5382980	3.2546976
SSA	147.2865013	131.1766455	168.1459727	6.9917053	125.8161345	179.3703342	2.6584208	4.7880091
HHO	149.9286650	143.9660246	173.6394530	5.3438261	147.7608158	124.0513904	2.6036005	3.4196237
LFD	146.8881225	143.4692235	144.5622880	3.1400000	35.2591765	111.1646593	1.7507575	3.9101673
GA	150.6680962	127.9517919	209.6817880	6.7103667	182.6199921	194.9682663	2.8799255	4.0901750
PSO	136.6283873	128.7195806	156.3550005	7.3693226	142.3238800	116.5521740	2.5336786	4.0930221

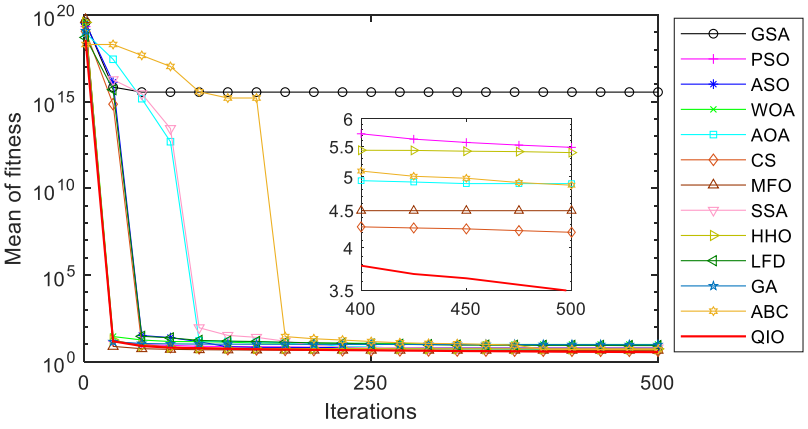

Fig. 32. Convergence comparisons of various optimizers for robot gripper design.

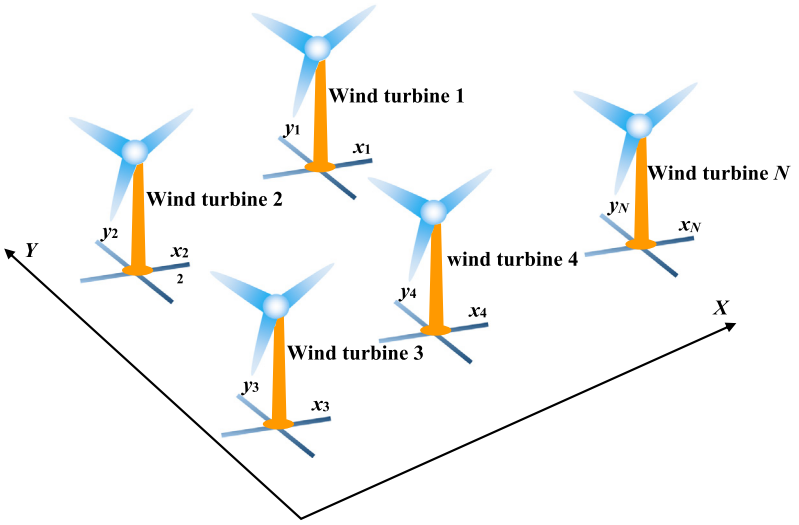

Fig. 33. Wind farm layout design.

$$E(P_i) = \sum_{t=1}^{H} \xi_t \left\{ P_r \left(e^{-(v_r/c_i'((\theta_{t-1}+\theta_t)/2))^{k_i((\theta_{t-1}+\theta_t)/2)}} - e^{-(v_{co}/c_i'((\theta_{t-1}+\theta_t)/2))^{k_i((\theta_{t-1}+\theta_t)/2)}} \right) \right.$$
$$\left. + \sum_{j=1}^{S} \left(e^{-(v_{j-1}/c_i'((\theta_{t-1}+\theta_t)/2))^{k_i((\theta_{t-1}+\theta_t)/2)}} - e^{-(v_j/c_i'((\theta_{t-1}+\theta_t)/2))^{k_i((\theta_{t-1}+\theta_t)/2)}} \right) \frac{e^{(v_{j-1}+v_j)/2}}{5 + e^{(v_{j-1}+v_j)/2}} \right\},$$

Subject to:

$$40 \le x_i \le 1960,$$

$$40 \le y_i \le 1960,$$

$$\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \ge 200, \quad j = 1, 2, \ldots, Num \text{ and } j \ne i$$

where $j = 1, 2, \ldots, Num$ and $j \neq i$, and ξ_n implies the spectrum of frequencies in the range $[\theta_{n-1}, \theta_n)$, $Num = H = 15$, $v_r = 14$, $v_{co} = 3.5$, $S = (v_r - v_{co})/0.3$.

This problem is tackled by various algorithms using 50 search individuals and 1000 iterations over 30 runs, and the statistical comparisons of various optimizers for this design are shown in Table 22. For QIO, the 'Ave' and 'Best' rank second, following MFO. The optimal results of all the algorithms and the corresponding design variables are provided in Tables 23 and 24. Fig. 34 shows the convergence comparisons of various optimizers for this case.

4.6. Four-stage gearbox design

This design is to minimize the weight of the four-stage gearbox as shown in Fig. 35 [103]. This design consists of 22 discrete structure variables and 86 constraints. There are numerous local solutions in feasible search space less than 0.0001 in ratio. This design is formulated below.

Consider variable:

$$\bar{x} = \left\{ N_{p1}, N_{g1}, N_{p2}, N_{g2} \ldots . b_1, b_2, \ldots . x_{p1}, x_{g1}, x_{g2} \ldots . y_{p1}, y_{g1}, y_{g2} \ldots . y_{g4} \right\},$$

Minimize:

$$f_6(\bar{x}) = \left(\frac{\pi}{1000} \right) \sum_{i=1}^{4} \frac{b_i c_i^2 \left(N_{pi}^2 + N_{gi}^2 \right)}{\left(N_{pi} + N_{gi} \right)^2}, i = (1, 2, 3, 4)$$

Table 22
Statistical comparisons of various optimizers for the wind farm layout design.

Algorithms	Ave	Std	Worst	Best
QIO	5784.1767007	139.1781652	5431.5130494	5952.2286732
GSA	4506.3155322	207.7719007	4013.7734701	4842.7908606
ABC	5007.9728887	**52.7857607**	4911.3205291	5128.1924258
ASO	5240.3780093	155.3494607	4941.3552555	5589.1687564
WOA	5432.6672684	170.3883314	5199.3927889	5785.0207622
AOA	5221.9850460	116.8724761	5010.7766317	5450.3318442
CS	5660.8115341	53.2731994	5544.0474980	5736.4223724
MFO	**5955.2134388**	119.4125644	**5703.7851251**	**6136.0774220**
SSA	5398.1496920	147.4021654	5104.0073363	5651.8545966
HHO	5525.6549775	132.3219113	5217.8458130	5677.6836593
LFD	5394.8613765	148.5813023	5160.8242772	5671.4159012
GA	5505.0493307	137.1118256	5185.3157544	5687.6777626
PSO	4929.8813135	58.7965947	4832.4687516	5051.5040181

Table 23
The best results of various optimizers for wind farm layout design in the x-axis.

Algorithms	Optimal variables						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
QIO	1859.4435755	1959.4721015	276.2439430	1169.7600589	757.7900846	483.3876732	44.6221697
GSA	1417.0176302	1422.8053790	528.5238715	956.4402196	1897.6717947	47.4506707	1743.8662667
ABC	476.5709449	689.3539843	1033.7392876	1685.9228878	629.6517033	1127.9853226	1804.3070387
ASO	287.7474801	73.3590392	1526.1968873	1252.5262709	299.7353704	1601.9717056	351.5648555
WOA	1122.8832022	1960.0000000	1318.0079499	1640.1927049	1198.6118468	44.5833089	170.5856594
AOA	1960.0000000	1438.2405367	1960.0000000	579.6171652	1130.0455699	1422.7501816	1960.0000000
CS	742.9468172	661.7038097	192.0321880	82.5495883	1023.1357246	1959.8441953	991.0752656
MFO	1201.2569664	1960.0000000	40.0000000	40.0000000	826.6120969	40.0000000	40.0000000
SSA	1684.5878478	914.4197450	218.4008775	329.4792953	267.7248939	1412.3071034	1191.8134244
HHO	1960.0000000	1112.4638616	1959.9683477	686.7436648	62.5107155	40.0000000	814.9822184
LFD	1526.1232638	909.7502008	157.5778601	229.6485309	750.0890767	1959.7974001	1930.8446934
GA	763.6749074	1681.2586695	174.3289883	621.2547173	1586.8714703	1202.0507150	388.4334638
PSO	1937.5179293	1519.0729792	227.9247836	1285.1533775	1074.7686355	1903.0623817	1368.4588197
Optimal variables							
x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
287.5807587	560.4026021	804.6388409	1169.5005740	44.3854701	1936.6053410	1217.1545839	1601.3380701
1777.4102555	1282.8506129	1050.2502513	41.7534887	75.2596793	451.8088469	658.3171177	918.2850736
1587.2325752	585.8257705	435.7855638	397.5290176	993.8697575	1478.4068942	695.6804561	1739.9698619
1839.8488826	1379.8935035	312.2927320	414.0889033	644.8176740	835.7676208	1939.2506115	798.6995461
427.7763331	1960.0000000	44.5833089	236.2980789	957.9193690	528.7340375	44.5833089	1450.4579361
40.0000000	430.1361722	623.2252717	1886.4431769	317.1348012	1960.0000000	913.0817331	1960.0000000
70.8654265	593.8466609	340.6687647	1240.2101669	1100.9822227	1957.1072230	40.0391374	1925.0955738
1606.7066046	40.0000000	1278.9781337	659.4838276	399.9120344	1960.0000000	40.0000000	1960.0000000
49.5970196	638.4138023	663.4864355	246.1802716	1898.7351884	240.0905918	824.9806009	1107.3822619
40.0000000	1085.5992214	645.9819897	73.1872795	1297.9112727	1957.6882395	1960.0000000	365.4541327
1630.9392420	1177.5680927	1193.4341700	317.1347787	1175.2393626	777.2850118	100.0151659	337.2097752
1213.4099664	593.4000403	787.0581104	321.9622405	1899.7352970	1490.5212405	544.7533845	646.2210424
1135.1831775	1935.8745860	1864.8933202	830.0448789	562.2925200	1843.1581022	623.4131528	947.9011541

Subject to:

$$g_1(\bar{x}) = \left(\frac{366000}{\pi\omega_1} + \frac{2c_1N_{p1}}{N_{pi}+N_{g1}}\right)\left(\frac{\left(N_{p1}+N_{g1}\right)^2}{4b_1c_1^2N_{p1}}\right) - \frac{\sigma_N J_R}{0.0167WK_0K_m} \le 0,$$

Table 24
The best results from various optimizers for wind farm layout design in the y-axis.

Algorithms	Optimal variables							
	y_1	Y_2	y_3	y_4	y_5	y_6	y_7	Y_8
QIO	1476.2977267	1927.4666271	854.1188247	753.1931090	1958.3097191	1882.5042386	1684.0366142	449.7102785
GSA	473.2089925	1201.5888178	1618.6339000	1841.7547902	716.0532248	116.4691619	370.0151632	1814.3383470
ABC	1835.8080897	1628.9085103	301.3963071	1865.9154768	1390.0217938	576.5202673	193.5687478	1428.7326426
ASO	341.5720940	184.6151594	979.3875923	221.5702879	1345.4742643	1531.6112120	535.3743735	1240.9947735
WOA	1280.1898928	44.5833089	44.5833089	159.4662400	1960.0000000	44.5833089	1354.6789293	1960.0000000
AOA	1960.0000000	913.0817331	1960.0000000	245.8107596	1886.4431769	113.5568231	488.9135098	1960.0000000
CS	646.4878878	1946.1443498	1960.0000000	1910.2625720	1729.4352373	1793.6023736	398.2458862	1428.0555502
MFO	393.1285159	652.6896693	686.2784036	1233.0846305	1542.3400163	1960.0000000	1126.2969257	1960.0000000
SSA	356.1620745	1940.3927288	1789.4334326	1812.2102614	1144.6365627	1811.2675610	360.2705180	849.2065037
HHO	326.1984035	656.3186255	1085.0254077	1039.3435735	1526.9685382	1892.3576414	1764.4385474	236.7330695
LFD	709.3616719	1818.9424738	384.9364875	57.6919131	1775.0291113	781.3405783	408.0111965	859.8143002
GA	319.7554979	994.5278341	633.9938577	716.0759776	89.2707922	1664.4811876	1890.7910156	1837.8918894
PSO	85.7181302	147.4339327	983.0456261	1615.7372343	1009.2123511	156.4893881	1877.2730417	731.9559003

Optimal variables							Optimal power output
y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	
1617.4534743	1164.8208982	280.8028684	1953.3754913	56.8328903	1908.0906480	732.6546401	5952.2286732
1241.7029837	334.3444260	168.5740233	505.7360547	1914.3788443	1938.2928789	1589.0231362	4842.7908606
186.4822751	1060.0832817	42.8454836	224.2233750	1684.4172200	1495.5741290	944.7770966	5128.1924258
1577.8099335	574.4085795	525.0563290	1718.4166748	71.3203265	1809.4928446	928.6797625	5589.1687564
1505.1346557	1960.0000000	398.9657177	1319.7575796	720.2739692	1814.4905810	1960.0000000	5785.0207622
1705.1657757	474.3695652	1372.9361509	40.0000000	1960.0000000	1960.0000000	1960.0000000	5450.3318442
1895.1366298	76.1163548	1939.0548594	1075.8464461	1499.7188724	1954.0686230	1520.8167400	5736.4223724
1960.0000000	40.0000000	1034.2223002	226.1921409	1960.0000000	1960.0000000	40.0000000	6136.0774220
1436.4154089	676.6738002	1834.2702506	1900.6287848	1445.3371644	721.4910308	1132.0605778	5651.8545966
632.2882796	627.5001065	1902.7405783	1624.3558095	40.0000000	1189.7615439	1955.7780181	5677.6836593
1763.8731418	1883.6080617	149.6729645	57.1500136	1544.4767493	1958.8007872	733.2510619	5671.4159012
1500.7346597	389.7176695	1509.2134956	1429.7118716	1544.9558721	1297.2772521	974.2934518	5687.6777626
1897.8541465	1048.2829593	389.7845831	1845.3172972	1324.8830353	720.4424976	751.8794863	5051.5040181

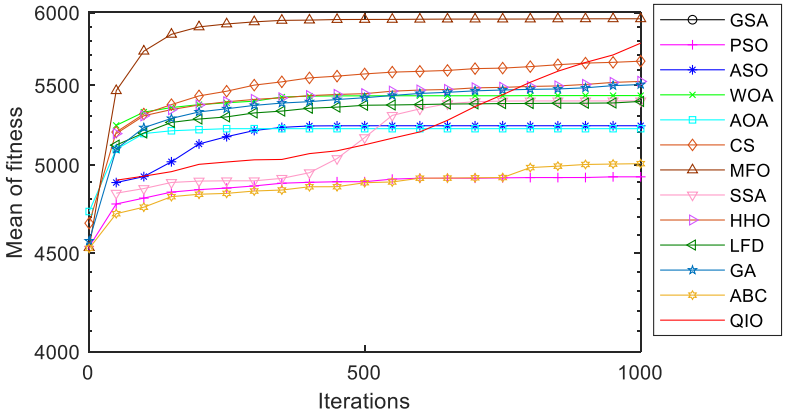

Fig. 34. Convergence comparisons of various optimizers for wind farm layout design.

$$g_2(\bar{x}) = \left(\frac{366000N_{g1}}{\pi\omega_1 N_{p1}} + \frac{2c_2N_{p2}}{N_{p2}+N_{g2}}\right)\left(\frac{(N_{p2}+N_{g2})^2}{4b_2c_2^2N_{p2}}\right) - \frac{\sigma_N J_R}{0.0167WK_0K_m} \le 0,$$

$$g_3(\bar{x}) = \left(\frac{366000N_{g1}N_{g2}}{\pi\omega_1 N_{p1}N_{p2}} + \frac{2c_3N_{p3}}{N_{p3}+N_{g3}}\right)\left(\frac{(N_{p3}+N_{g3})^2}{4b_3c_3^2N_{p3}}\right) - \frac{\sigma_N J_R}{0.0167WK_0K_m} \le 0,$$

$$g_4(\bar{x}) = \left(\frac{366000N_{g1}N_{g2}N_{g3}}{\pi\omega_1 N_{p1}N_{p2}N_{p3}} + \frac{2c_4N_{p4}}{N_{p4}+N_{g4}}\right)\left(\frac{\left(N_{p4}+N_{g4}\right)^2}{4b_4c_4^2N_{p4}}\right) - \frac{\sigma_N J_R}{0.0167WK_0K_m} \le 0,$$

$$g_5(\bar{x}) = \left(\frac{366000}{\pi\omega_1} + \frac{2c_1N_{p1}}{N_{p1}+N_{g1}}\right)\left(\frac{\left(N_{p1}+N_{g1}\right)^3}{4b_1c_1^2N_{g1}N_{p1}^2}\right) - \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{\sin(\phi)\cos(\phi)}{0.0334WK_0K_m}\right) \le 0,$$

$$g_6(\bar{x}) = \left(\frac{366000N_{g1}}{\pi\omega_1 N_{p1}} + \frac{2c_2N_{p2}}{N_{p2}+N_{g2}}\right)\left(\frac{\left(N_{p2}+N_{g2}\right)^3}{4b_2c_2^2N_{g2}N_{p2}^2}\right) - \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{\sin(\phi)\cos(\phi)}{0.0334WK_0K_m}\right) \le 0,$$

$$g_7(\bar{x}) = \left(\frac{366000N_{g1}N_{g2}}{\pi\omega_1 N_{p1}N_{p2}} + \frac{2c_3N_{p3}}{N_{p3}+N_{g3}}\right)\left(\frac{\left(N_{p3}+N_{g3}\right)^3}{4b_3c_3^2N_{g3}N_{p3}^2}\right) - \left(\frac{\sigma_H}{C_P}\right)^2\left(\frac{\sin(\phi)\cos(\phi)}{0.0334WK_0K_m}\right) \le 0,$$

$$g_8(\bar{x}) = \left(\frac{366000N_{g1}N_{g2}N_{g3}}{\pi\omega_1 N_{p1}N_{p2}N_{p3}} + \frac{2c_4N_{p4}}{N_{p4}+N_{g4}}\right)\left(\frac{\left(N_{p4}+N_{g4}\right)^3}{4b_4c_4^2N_{g4}N_{p4}^2}\right) - \left(\frac{\sigma_H}{C_p}\right)^2\left(\frac{\sin(\phi)\cos(\phi)}{0.0334WK_0K_m}\right) \le 0,$$

$$g_{9-12}(\bar{x}) = -N_{pi}\sqrt{\frac{\sin^2(\phi)}{4} - \frac{1}{N_{pi}} + \left(\frac{1}{N_{pi}}\right)^2} + N_{gi}\sqrt{\frac{\sin^2(\phi)}{4} + \frac{1}{N_{gi}}\left(\frac{1}{N_{gi}}\right)^2} + \frac{\sin(\phi)\left(N_{pi}+N_{gi}\right)}{2} + CR_{\min}\pi\cos(\phi) \le 0,$$

$$g_{13-16}(\bar{x}) = d_{\min} - \frac{2c_iN_{pi}}{N_{pi}+N_{gi}} \le 0,$$

$$g_{17-20}(\bar{x}) = d_{\min} - \frac{2c_iN_{gi}}{N_{pi}+N_{gi}} \le 0,$$

$$g_{21}(\bar{x}) = x_{p1} + \left(\frac{\left(N_{p1}+2\right)c_1}{N_{p1}+N_{g1}}\right) - L_{\max} \le 0,$$

$$g_{22-24}(\bar{x}) = -L_{\max} + \left(\frac{\left(N_{pi}+2\right)c_i}{N_{gi}+N_{pi}}\right)_{i=2,3,4} - x_{g(i-1)} \le 0,$$

$$g_{25}(\bar{x}) = -x_{p1} + \left(\frac{\left(N_{p1}+2\right)c_1}{N_{p1}+N_{g1}}\right) \le 0,$$

$$g_{26-28}(\bar{x}) = \left(\frac{\left(N_{pi}+2\right)c_i}{N_{pi}+N_{gi}} - x_{g(i-1)}\right)_{i=2,3,4} \le 0,$$

$$g_{29}(\bar{x}) = y_{p1} + \frac{\left(N_{p1}+2\right)c_1}{N_{p1}+N_{g1}} - L_{\max} \le 0,$$

$$g_{30-32}(\bar{x}) = -L_{\max} + \left(\frac{c_i\left(2+N_{pi}\right)}{N_{pi}+N_{gi}} + y_{g(i-1)}\right)_{i=2,3,4} \le 0,$$

$$g_{33}(\bar{x}) = \frac{\left(2+N_{p1}\right)c_1}{N_{p1}+N_{g1}} - y_{p1} \le 0,$$

$$g_{34-36}(\bar{x}) = \left(\frac{c_i\left(2+N_{pi}\right)}{N_{pi}+N_{gi}} - y_{g(i-1)}\right)_{i=2,3,4} \le 0,$$

$$g_{37-40}(\bar{x}) = -L_{\max} + \frac{c_i\left(2+N_{gi}\right)}{N_{pi}+N_{gi}} + x_{gi} \le 0,$$

$$g_{41-44}(\bar{x}) = -x_{gi} + \left(\frac{\left(N_{gi}+2\right)c_i}{N_{pi}+N_{gi}}\right) \le 0,$$

$$g_{45-48}(\bar{x}) = y_{gi} + \left(\frac{\left(N_{gi}+2\right)c_i}{N_{pi}+N_{gi}}\right) - L_{\max} \le 0,$$

$$g_{49-52}(\bar{x}) = -y_{gi} + \left(\frac{(N_{gi}+2)\,c_i}{N_{pi}+N_{gi}}\right) \le 0,$$

$$g_{53-56}(\bar{x}) = (b_i - 8.255)(b_i - 5.715)(b_i - 12.70)\left(-N_{pi} + 0.945c_i - N_{gi}\right)(-1) \le 0,$$

$$g_{57-60}(\bar{x}) = (b_i - 8.255)(b_i - 3.175)(b_i - 12.70)\left(-N_{pi} + 0.646c_i - N_{gi}\right) \le 0,$$
$$g_{61-64}(\bar{x}) = (b_i - 5.715)(b_i - 3.175)(b_i - 12.70)\left(-N_{pi} + 0.504c_i - N_{gi}\right) \le 0,$$
$$g_{65-68}(\bar{x}) = (b_i - 5.715)(b_i - 3.175)(b_i - 8.255)\left(0c_i - N_{gi} - N_{pi}\right) \le 0,$$
$$g_{69-72}(\bar{x}) = (b_i - 8.255)(b_i - 5.715)(b_i - 12.70)\left(N_{gi} + N_{pi} - 1.812c_i\right)(-1) \le 0,$$
$$g_{73-76}(\bar{x}) = (b_i - 8.255)(b_i - 3.175)(b_i - 12.70)\left(-0.945c_i + N_{pi} + N_{gi}\right) \le 0,$$
$$g_{77-80}(\bar{x}) = (b_i - 5.715)(b_i - 3.175)(b_i - 12.70)\left(-0.646c_i + N_{pi} + N_{gi}\right)(-1) \le 0,$$
$$g_{81-84}(\bar{x}) = (b_i - 5.715)(b_i - 3.175)(b_i - 8.255)\left(N_{pi} + N_{gi} - 0.504c_i\right) \le 0,$$

$$g_{85}(\bar{x}) = \omega_{\min} - \frac{\omega_1\left(N_{p1}N_{p2}N_{p3}N_{p4}\right)}{\left(N_{g1}N_{g2}N_{g3}N_{g4}\right)} \le 0,$$
$$g_{86}(\bar{x}) = \frac{\omega_1\left(N_{p1}N_{p2}N_{p3}N_{p4}\right)}{\left(N_{g1}N_{g2}N_{g3}N_{g4}\right)} - \omega_{\max} \le 0,$$

where,

$c_i = \sqrt{\left(y_{gi} - y_{pi}\right)^2 + \left(x_{gi} - x_{pi}\right)^2}$, $K_0 = 1.5$, $d_{\min} = 25$, $J_R = 0.2$, $\phi = 2\pi/3W = 55.9$, $K_M = 1.6$, $CR_{\min} = 1.4$, $L_{\max} = 127$, $C_p = 464$, $\sigma_H = 3290$, $\omega_{\max} = 255$, $\omega_1 = 5000$, $\sigma_N = 2090$, $\omega_{\min} = 245$.

Variable range:

$$b_i \in \{3.175, 12.7, 8.255, 5.715\},$$

$$y_{p1}, x_{p1}, y_{gi}, x_{gi} \in \{12.7, 38.1, 25.4, 50.8, 76.2, 63.5, 88.9, 114.3, 101.6\},$$

$$7 \le N_{gi}, N_{pi} \le 76 \in \text{int}\,eger.$$

This design is solved by various algorithms using 50 search individuals and 1000 iterations over 30 independent 30 runs. Table 25 shows the statistical comparisons of various optimizers for this design. It can be seen that QIO outperforms other algorithms in terms of the 'Ave', 'Std', 'Worst', and 'Best' indexes, and it successfully transfers from the infeasible area to the feasible area and finds the best solution over 30 runs. In contrast, the other algorithms all fail to jump out of the infeasible area over 30 runs. This large gap demonstrates that QIO has a significant advantage over other algorithms in solving this challenging problem. Table 26 presents the minimal weight of various optimizers and corresponding design variables. Fig. 36 graphically shows the convergence comparisons of various optimizers for this case.

4.7. Rolling element bearing design

This design aims to maximize the dynamic load-carrying capacity of the rolling element bearing as shown in Fig. 37 [104]. It has ten design variables and nine constraints. This design is formulated below.

Consider variable: $\vec{x} = [D_m, D_b, Z, f_i, f_o, K_{D\min}, K_{D\max}, \varepsilon, e, \zeta]$.

$$\text{Maximize: } f_7(\vec{x}) = \begin{cases} f_c Z^{2/3} D_b^{1.8} & if\ D_b \le 25.4\ \text{mm} \\ 3.647 f_c Z^{2/3} D_b^{1.4} & if\ D_b > 25.4\ \text{mm} \end{cases}$$

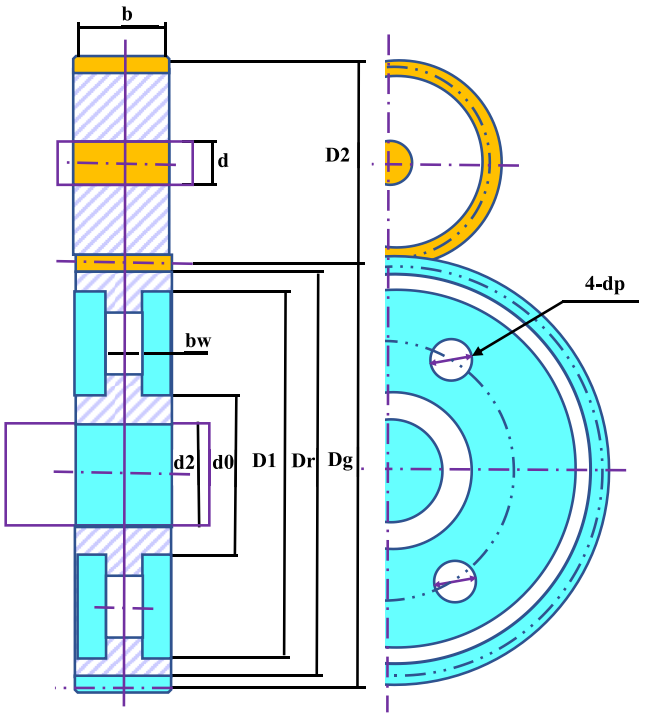

Fig. 35. Four-stage gearbox design.

Table 25
Statistical comparisons of various optimizers for four-stage gearbox design.

Algorithms	Ave	Std	Worst	Best
QIO	**1.2886E+16**	**3.2062E+16**	**1.4954E+17**	**42.8956429**
GSA	1.3615E+19	1.5058E+19	6.1493E+19	1.4837E+18
ABC	1.3370E+17	1.1724E+17	4.7149E+17	6.5699E+14
ASO	1.2200E+19	5.9447E+18	2.2697E+19	2.2373E+18
WOA	4.7992E+17	2.9418E+17	1.2814E+18	1.2852E+16
AOA	7.4159E+18	2.5362E+18	1.2518E+19	2.6601E+18
CS	3.2128E+17	1.5544E+17	8.0736E+17	2.5013E+16
MFO	2.2785E+18	1.5558E+18	5.4040E+18	9.1635E+16
SSA	1.4716E+18	2.7804E+18	1.1778E+19	1.3757E+16
HHO	2.5298E+17	2.1089E+17	7.7645E+17	3.4287E+15
LFD	2.9347E+19	3.3197E+19	1.4144E+20	4.2545E+18
GA	7.7158E+17	1.0692E+18	5.3291E+18	7.4368E+15
PSO	1.2732E+18	1.0256E+18	3.8052E+18	5.2654E+15

Subject to:

$$g_1(\vec{x}) = \frac{\phi_o}{2\sin^{-1}(D_b/D_m)} - Z + 1 \geq 0,\ g_2(\vec{x}) = 2D_b - K_{D\min}(D-d) \geq 0,$$

$$g_3(\vec{x}) = K_{D\max}(D-d) - 2D_b \geq 0,\ g_4(\vec{x}) = D_m - (0.5-e)(D+d) \geq 0,$$

$$g_5(\vec{x}) = (0.5+e)(D+d) - D_m \geq 0,\ g_6(\vec{x}) = D_m - 0.5(D+d) \geq 0,$$
$$g_7(\vec{x}) = 0.5(D - D_m - D_b) - \varepsilon D_b \geq 0,\ g_8(\vec{x}) = \zeta B_w - D_b \leq 0,$$

$$g_9(\vec{x}) = f_i \geq 0.515,\ g_{10}(\vec{x}) = f_o \geq 0.515,$$

Table 26
The results of various optimizers for four-stage gearbox problem.

Algorithms	Optimal variables						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7
QIO	13.3770165	23.4409500	19.2869177	42.1184965	17.7501987	39.6151822	14.2972655
GSA	9.6581192	21.7283366	25.8063005	50.7109146	42.5209446	63.2628942	23.6672846
ABC	17.3923463	49.5928788	12.6961797	34.2281887	13.0873177	32.2751193	31.3551753
ASO	21.8253462	50.3570031	16.1836084	74.5078216	48.4431920	46.2127909	23.6914401
WOA	21.4509030	75.1072727	19.0712694	53.5237155	13.5066656	18.7398833	16.9564618
AOA	6.5100000	29.8284066	6.5100000	6.5100000	22.5792345	76.4900000	38.7160058
CS	22.8319517	67.7883454	19.4456799	42.3782572	33.4306549	30.8757169	11.6342883
MFO	11.5390392	44.2294683	13.9562516	41.9728599	19.0229488	20.0644674	6.5100000
SSA	21.8232650	41.9903317	12.6348002	42.1032357	20.5090481	47.4472418	24.4583760
HHO	19.5284809	38.5407431	24.4256942	47.1376717	11.3927661	27.5469795	12.7156775
LFD	19.3736552	42.9518961	20.8075117	17.2857937	31.8593047	50.8736617	9.1004831
GA	23.6748404	67.6247725	14.2534169	28.0834350	8.6403710	34.1165924	33.6492054
PSO	22.8351330	41.6302931	18.0488027	42.1711503	24.7030950	39.7748424	17.9217115
Optimal variables							
x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
31.7564219	1.2396396	0.7204342	1.0659409	0.9855664	5.5794980	7.7173185	3.0157597
66.6267251	2.5538288	1.0817633	1.0790235	0.9200755	3.2311267	6.6214073	3.1439079
31.7449684	0.7839253	2.3852765	2.1968636	1.4824749	1.6536862	5.8863965	6.0631836
43.9191504	0.8888765	0.6594883	0.6557952	0.5614106	5.3665860	0.7739539	7.9372889
25.2699147	0.9429569	1.0114472	1.5686283	1.0119855	1.0794015	5.2043161	4.8705649
53.1715934	2.2142433	3.8361804	1.2836839	0.5100000	5.5279491	1.8997096	3.3928884
39.8843496	1.2414510	1.8416281	1.6158316	1.2795502	4.9348745	5.5577416	6.1808004
12.1519764	0.5100000	0.5100000	0.7634870	1.9715920	6.5779761	4.7016982	6.2342800
33.7769907	1.3417879	1.3259020	1.4533495	1.2532296	4.5968439	4.0173060	3.9958704
26.9732924	1.0520754	1.3211232	2.2943342	3.1044923	2.4950569	2.2539569	3.2305047
64.7863251	0.7099946	1.8271917	1.4979411	1.0939593	5.0596845	2.3610365	4.2752881
32.2623533	1.2495325	2.5899320	2.2472223	1.3490456	2.3989676	2.9101372	4.3659615
51.7659930	0.9521778	1.4350819	0.6575705	1.1509970	2.9211993	5.8158220	6.7337701
Optimal variables							Minimum weight
x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	
3.3326391	4.3790319	1.8703261	3.8204405	3.8788949	3.9909437	4.6956342	42.8956429
6.5564482	5.7666035	1.0305707	3.5853615	7.2970861	4.3367241	5.9538141	1.4837E+18
6.4332222	5.5791072	5.2120256	5.2510418	4.4078405	4.3690275	4.7735839	6.5699E+14
7.5242310	3.5307400	2.8604115	2.6725497	6.8735789	7.4038960	8.4458495	2.2373E+18
3.5995375	1.5272127	0.5501181	3.5585530	4.4294311	1.5572201	3.7991284	1.2852E+16
4.2592222	7.3606603	9.4900000	9.4900000	5.6994818	3.5811399	2.0737722	2.6601E+18
6.0462198	3.8538329	8.8622262	4.7824496	3.9398236	3.1173123	4.7399214	2.5013E+16
6.6580559	9.4900000	0.5100000	5.3894793	5.1642296	3.7137536	1.9339295	9.1635E+16
1.5303126	3.2394703	1.6875180	6.3822867	5.5085997	3.1688993	3.7925887	1.3757E+16
4.8641156	4.4167684	9.4900000	6.3111962	5.6728513	6.9347563	4.3181403	3.4287E+15
8.7574312	4.3401617	4.8021146	2.5517866	1.9013983	0.7242689	1.3789659	4.2545E+18
5.3149607	2.4781967	8.5266161	4.8062799	3.9474417	5.2144122	5.6056668	7.4368E+15
2.5749526	3.7535375	2.8123018	5.5351170	2.5693876	7.4454280	6.5334025	5.2654E+15

where

$$f_c = 37.91\left[1+\left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)}\right)^{0.4}\right\}^{10/3}\right]^{-0.3}\left(\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{f_o(1+\gamma)^{\frac{1}{3}}}\right)\left(\frac{2f_i}{2f_i-1}\right)^{0.41},$$

$$\gamma = \frac{D_b}{D_m},\ f_i = \frac{r_i}{D_b},\ f_o = \frac{r_o}{D_b},$$

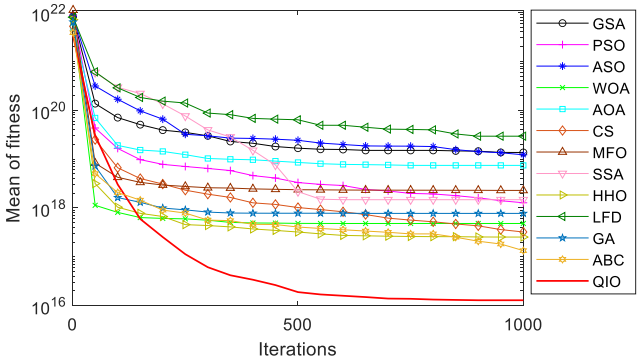

Fig. 36. Convergence comparisons of various optimizers for four-stage gearbox design.

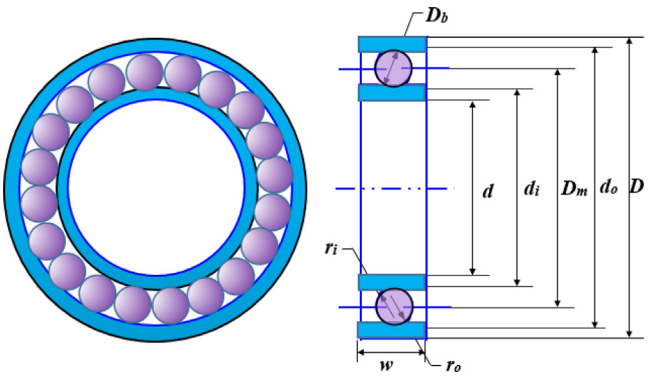

Fig. 37. Rolling element bearing design.

$$\phi_o = 2\pi - 2\cos^{-1}\frac{\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - (T/4) - D_b\}^2 - \{d/2 + (T/4)\}^2}{2\{(D-d)/2 - 3(T/4)\}\{D/2 - (T/4) - D_b\}},$$

$$T = D - d - 2D_b,\ D = 160,\ d = 90,\ B_w = 30,\ r_i = r_o = 11.033.$$

Variable range:

$$0.5(D+d) \le D_m \le 0.6(D+d),\ 0.15(D-d) \le D_b \le 0.45(D-d),\ 4 \le Z \le 50,$$
$$0.515 \le f_i \le 0.6,\ 0.515 \le f_o \le 0.6,\ 0.4 \le K_{D\min} \le 0.5,\ 0.6 \le K_{D\max} \le 0.7,$$
$$0.3 \le \varepsilon \le 0.4,\ 0.02 \le e \le 0.1,\ 0.6 \le \zeta \le 0.85.$$

We use 50 search individuals and 1000 iterations to deal with this design over 30 runs. Table 27 shows the statistical comparisons of various optimizers for this design. According to the results of 'Ave', 'Std', and 'Worst' indexes, QIO is better than other competition algorithms in various aspects when solving this design. It is also common to find that QIO can accurately approximate the global optimum at every run, indicating its stability in the search for the global optimum. Table 28 presents the minimal weight of various optimizers and corresponding design variables. Fig. 38 demonstrates the convergence comparisons of various optimizers for this case.

4.8. Tabular column design

This design is to carry a compressive load with a minimum cost, which consists of material and construction costs [105]. It has two design variables and six constraints. The structure diagram of this design is plotted in Fig. 39. This design is formulated below.

Consider variable: $\vec{x} = [d, t] = [x_1, x_2]$

Table 27
Statistical comparisons of various optimizers for rolling element bearing design.

Algorithms	Ave	Std	Worst	Best
QIO	**85549.0334**	**0.0858**	**85548.8417**	85549.1706
GSA	71 913.6295	6958.7224	55 087.0241	82 552.0749
ABC	85 547.2837	0.8819	85 545.3143	85 548.6512
ASO	74 516.9716	5446.3630	63 353.0169	81 489.5276
WOA	82 551.0323	2947.4815	70 414.6027	85 475.9529
AOA	78 055.8918	4255.2026	67 334.0645	84 426.9163
CS	85 548.7900	0.2356	85 548.3175	85 549.1557
MFO	85 328.7083	441.9399	84 459.7849	**85549.2391**
SSA	67 094.7185	6786.5577	50 517.3279	82 331.5931
HHO	83 639.3523	774.8560	82 498.5495	85 004.0033
LFD	53 143.6189	11 205.8199	33 212.8240	80 119.5117
GA	76 342.6688	2734.4092	72 701.0001	82 967.2047
PSO	75 463.5998	3511.0129	68 075.5893	80 898.7683

Table 28
The best results of various optimizers for rolling element bearing design.

Algorithms	Optimal variables										Optimal load-carrying capacity
	D_m	D_b	Z	f_i	f_o	K_{Dmin}	K_{Dmax}	ε	e	ζ	
QIO	125.7190308	21.4255815	10.6300809	0.5150000	0.5150184	0.4931640	0.6304266	0.3000004	0.0962602	0.6890717	85 549.1706
GSA	126.2159573	21.0046487	10.8576780	0.5150161	0.5967712	0.4754116	0.6932595	0.3027011	0.0818825	0.6606288	82 552.0749
ABC	125.7189494	21.4255466	11.2319012	0.5150001	0.5157571	0.4237518	0.6196589	0.3000025	0.0418011	0.6742941	85 548.6512
ASO	125.2525765	20.8485627	10.6669663	0.5150000	0.5800945	0.4750573	0.6512696	0.3134326	0.0419156	0.6724730	81 489.5276
WOA	125.7357751	21.4151405	11.4994919	0.5150000	0.5150000	0.4000000	0.7000000	0.3000000	0.0816970	0.6000000	85 475.9529
AOA	125.0000000	21.2703922	11.0036762	0.5150000	0.5794477	0.4128673	0.7000000	0.3000000	0.0200000	0.6000000	84 426.9163
CS	125.7190051	21.4255786	11.2533838	0.5150000	0.5150000	0.4000000	0.6454340	0.3000006	0.0731943	0.6116723	85 549.1557
MFO	125.7190556	21.4255902	10.7097995	0.5150000	0.5150000	0.4000000	0.7000000	0.3000000	0.1000000	0.6380107	85 549.2391
SSA	125.0505281	20.9683857	10.5041438	0.5150000	0.5212626	0.4421407	0.6149440	0.3089046	0.0592631	0.6413655	82 331.5931
HHO	125.3609740	21.3494995	10.9340913	0.5150001	0.5207056	0.4068857	0.6142903	0.3104052	0.0271832	0.6180408	85 004.0033
LFD	125.8487808	20.7674255	10.9239259	0.5153862	0.5855381	0.4897282	0.6617550	0.3185130	0.0544005	0.6276224	80 119.5117
GA	126.0736080	21.0662259	10.5320286	0.5150301	0.5150026	0.4570400	0.6086311	0.3052173	0.0590993	0.6284123	82 967.2047
PSO	125.4898111	20.8758090	10.7785021	0.5153696	0.5625308	0.4697216	0.6545016	0.3138232	0.0636192	0.6122001	80 898.7683

Minimize: $f_8(\vec{x}) = 9.82dt + 2d$

Subject to:

$$\begin{cases} g_1(\vec{x}) = \dfrac{x_1}{14} - 1 \le 0 \\ g_2(\vec{x}) = \dfrac{8PL^2}{\pi 3Eh_1h_2(h_1^2 + h_2^2)} - 1 \le 0 \\ g_3(\vec{x}) = \dfrac{2}{h_1} - 1 \le 0, \\ g_4(\vec{x}) = \dfrac{P}{\pi h_1 h_2 \sigma_y} - 1 \le 0 \\ g_5(\vec{x}) = \dfrac{0.2}{x_2} - 1 \le 0 \\ g_6(\vec{x}) = \dfrac{x_2}{0.8} - 1 \le 0 \end{cases}$$

where $P = 2500$, $\sigma_y = 500$, $E = 0.85 \times 10^6$, $\sigma_y = 500$

Variable range: $2 \le d \le 14$, $0.2 \le t \le 0.8$.

This design is solved by various algorithms using 50 search individuals and 500 iterations over 30 runs. Table 29 shows the statistical comparisons of various optimizers for this design. QIO exhibits the best performance in terms of the 'Ave', 'Std', and 'Worst' indexes. Meanwhile, it can provide very competitive results in terms of the 'Best' index. Table 30 shows the minimal weight of various optimizers and corresponding design variables. Fig. 40 presents the convergence comparisons of various optimizers for this case.

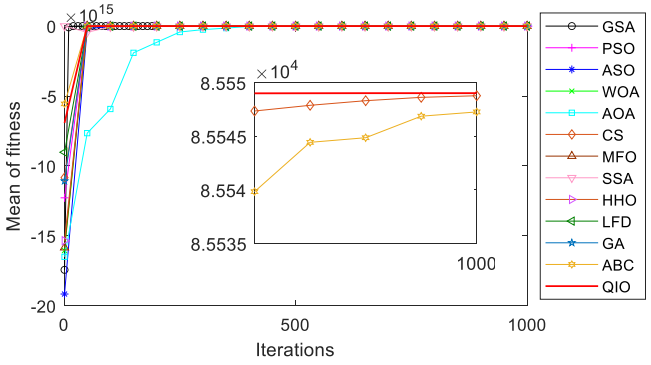

Fig. 38. Convergence comparisons of various optimizers for rolling element bearing design.

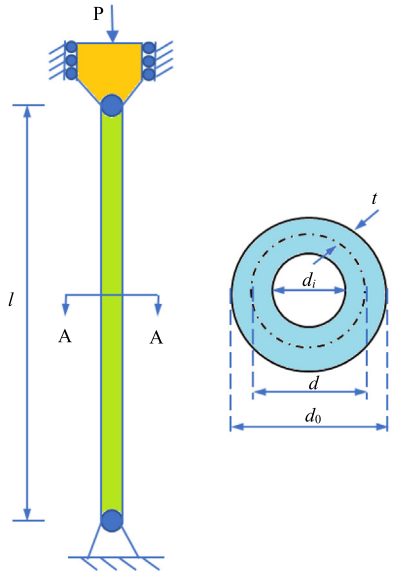

Fig. 39. Tabular column design.

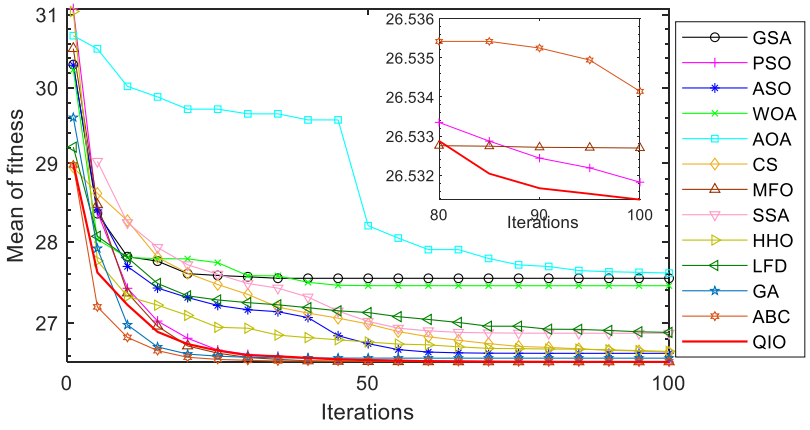

Fig. 40. Convergence comparisons of various optimizers for tabular column design.

Table 29
Statistical comparisons of various optimizers for tabular column design.

Algorithms	Ave	Std	Worst	Best
QIO	**26.5313803**	**4.3499E−05**	**26.5314830**	26.5313320
GSA	27.5445104	0.5815444	28.9738846	26.5646327
ABC	26.5341422	2.0662E−03	26.5422744	26.5316697
ASO	26.6337393	0.1145514	26.9945692	26.5321919
WOA	27.4530757	0.9488199	30.1390502	26.5771428
AOA	27.6093417	0.5181492	29.4140259	26.8075117
CS	26.6441996	5.7435E−02	26.7850671	26.5509042
MFO	26.5326962	5.5968E−03	26.5598784	26.5313279
SSA	26.8746617	0.5016529	28.8402202	26.5319206
HHO	26.6566543	0.2109851	27.3153500	26.5337214
LFD	26.8847314	0.2057044	27.3270006	26.5878430
GA	26.5761298	6.9689E−02	26.7969925	**26.5313279**
PSO	26.5318246	5.2957E−04	26.5332676	26.5313785

Table 30
The best results of various optimizers for tabular column design.

Algorithms	Optimum variables		Optimum cost
	d	t	
QIO	5.4511577	0.2919654	26.5313320
GSA	5.4437074	0.2932667	26.5646327
ABC	5.4513060	0.2919582	26.5316697
ASO	5.4509323	0.2920020	26.5321919
WOA	5.4387396	0.2939548	26.5771428
AOA	5.4051568	0.3013867	26.8075117
CS	5.4542122	0.2920533	26.5509042
MFO	5.4511562	0.2919655	26.5313279
SSA	5.4509957	0.2919911	26.5319206
HHO	5.4518069	0.2919510	26.5337214
LFD	5.4437484	0.2936971	26.5878430
GA	5.4511562	0.2919655	26.5313279
PSO	5.4511432	0.2919676	26.5313785

4.9. Heat exchanger network design

Heat exchanger network synthesis is a very important subject in the last two decades [26]. This design involves designing an optimal heat exchanger network to minimize the overall heat exchange area. This system consists of three hot streams and one cold stream. The cold stream needs to be heated from 100 °F to 500 °F using three hot streams. The system diagram of this design is plotted in Fig. 41. This design is formulated below.

Consider variable: $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]$

Minimize: $f_9(\vec{x}) = x_1 + x_2 + x_3$

Subject to:

$$\begin{cases} g(1) = -1 + 0.0025\,(x_4 + x_6) \le 0 \\ g(2) = -1 + 0.0025\,(x_5 + x_7 - x_4) \le 0 \\ g(3) = -1 + 0.01\,(x_8 - x_5) \le 0 \\ g(4) = -x_1 x_6 + 833.33252 x_4 + 100 x_1 - 83333.333 \le 0 \\ g(5) = -x_2 x7 + 1250 x_5 + x_2 x_4 - 1250 x_4 \le 0 \\ g(6) = -x_3 x_8 + 1250000 + x_3 x_5 - 2500 x_5 \le 0 \end{cases}$$

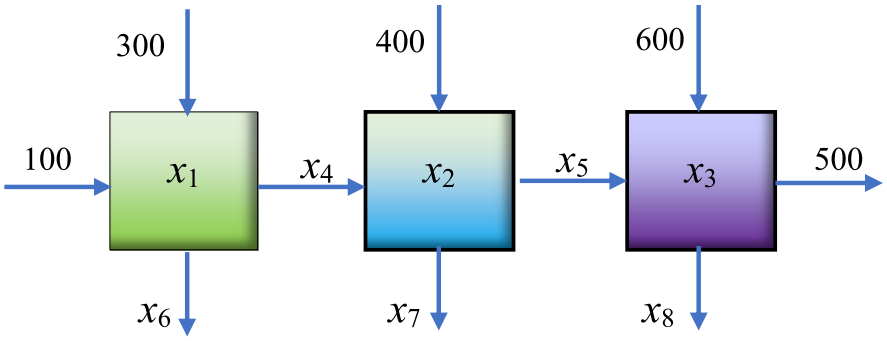

Fig. 41. Heat exchanger network design.

Table 31
Statistical comparisons of various optimizers for heat exchanger network design.

Algorithms	Ave	Std	Worst	Best
QIO	**7247.8133823**	**140.8927584**	**7616.5196760**	**7061.3421110**
GSA	1.2173E+15	7.8329E+14	3.1941E+15	2.3734E+13
ABC	7978.3555105	259.9309007	8412.3562874	7328.4836460
ASO	1.6986E+11	9.2759E+11	5.0811E+12	7972.2784296
WOA	1.2819E+14	2.5690E+14	1.1127E+15	1.2211E+04
AOA	4.7484E+13	9.3613E+13	3.9715E+14	1.8835E+04
CS	9008.4682177	395.6138848	1.0136E+04	8352.4842704
MFO	8.5290E+10	3.2850E+11	1.4721E+12	7062.8399207
SSA	5.4740E+08	2.2155E+09	1.1784E+10	9596.4816914
HHO	1.1789E+04	2429.8087355	1.8805E+04	8708.0144451
LFD	1.0287E+14	1.0714E+14	4.6232E+14	1.3202E+04
GA	1.0563E+04	2137.7232057	1.5516E+04	7271.2350228
PSO	8933.2685263	1205.5054684	1.1977E+04	7216.5238684

Table 32
The best results of various optimizers for heat exchanger network design.

Algorithms	Optimal values for variables								Optimal heat exchange area
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
QIO	557.4608264	1524.4180533	4979.4632313	180.0047021	300.8329573	219.7384992	279.1203573	400.8279120	7061.3421110
GSA	6875.0863513	2484.4902356	8799.9713311	322.4135805	240.1803348	130.8486141	282.8185381	347.9287637	2.3734E+13
ABC	574.0237284	1789.5879887	4964.8719289	175.9454023	305.2935052	218.6421463	268.3612058	404.7328914	7328.4836460
ASO	460.6792572	2969.0472571	4542.5519152	165.0207171	324.0753050	219.8007219	235.2520585	421.6180540	7972.2784296
WOA	1731.0847225	5025.4812519	5454.4804380	171.6141130	283.3021564	164.8147685	238.8983738	382.6231894	12 211.0464124
AOA	100.0000000	8734.9484647	10 000.0000000	10.2953250	239.9386185	272.4158309	44.1757490	320.8374131	18 834.9484647
CS	100.0000000	2072.5055386	6179.9787318	57.4844053	267.0655426	271.1728871	184.5670927	364.6814053	8352.4842704
MFO	448.5617611	1355.9447149	5258.3334447	169.9828591	289.6668118	230.0142298	280.3159421	389.6667787	7062.8399207
SSA	1198.1904480	2267.3355354	6130.9557079	155.5161434	254.7617968	230.2036672	272.8885406	354.7617909	9596.4816914
HHO	930.2718299	2787.2991728	4990.4434424	174.0349210	300.8900275	167.1059685	243.1165259	400.6390519	8708.0144451
LFD	998.9191173	4131.7951837	8071.0196676	117.9207972	197.2835485	268.5167045	307.9711729	295.1598962	1.3202E+04
GA	355.8919809	1645.9629548	5269.3800870	152.8825327	291.5760263	224.1354730	258.3789607	390.4831244	7271.2350228
PSO	654.6157007	1941.9134014	4619.9947662	184.3822960	315.3093909	209.4530363	268.8681769	415.2974337	7216.5238684

Variable range: $100 \leq x_1 \leq 10000$, $1000 \leq x_2, x_3 \leq 10000$, $10 \leq x_4, x_5, x_6, x_7, x_8 \leq 1000$

The competing algorithms are run to find the global optimum of this design with 50 search individuals and 500 iterations over 30 runs. Table 31 shows the statistical comparisons of various optimizers for this design. The performance of QIO is remarkably superior to its competitors concerning the 'Ave', 'Std', 'Best', and 'Worst' indexes, which verify the good stability and effectiveness of QIO in solving this design. Table 32 shows the minimal weight of various optimizers and corresponding design variables. Fig. 42 shows the convergence comparisons of various optimizers for this case.

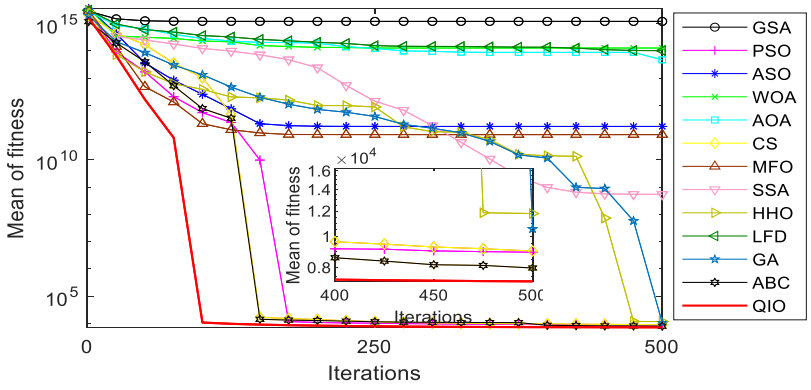

Fig. 42. Convergence comparisons of various optimizers for heat exchanger network design.

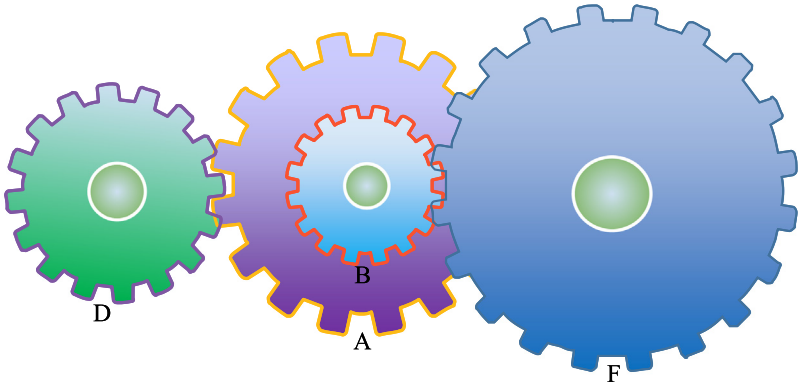

Fig. 43. Gear train design.

4.10. Gear train design

This design, as shown in Fig. 43 [63], is to minimize the cost of the gear ratio of the gear train. It has four integer variables. The schematic diagram of this design is plotted in Fig. 43. This design is formulated below.

Consider variable: $\vec{x} = [x_1, x_2, x_3, x_4] = [T_a, T_b, T_d, T_f]$.

Minimize: $f_{10}(\vec{x}) = (\frac{1}{6.931} + \frac{T_b \cdot T_d}{T_a \cdot T_f})$.

Variable range: $0.01 \leq x_1, x_2, x_3, x_4 \leq 60$.

This design is solved by various algorithms using 50 search individuals and 100 iterations over 30 independent runs. Table 33 shows the statistical comparisons of various optimizers for this design. QIO provides the same results as ABC in terms of the 'Worst' and 'Best' indexes, which are better than the rest competitors, and QIO provides the second-best results in terms of the 'Ave' and 'Std' indexes which are slightly inferior to those of ABC. Table 34 tabulates the minimal cost of various optimizers and corresponding design variables. Fig. 44 shows the convergence comparisons of various optimizers for this case.

5. The application of QIO to operation management of microgrid with energy storage system

A microgrid is an important part of the smart grid. On the one hand, the energy storage battery can adjust the power of a microgrid; on the other hand, it can achieve the effect of peak cutting and valley filling. Therefore, microgrids can play a crucial role in improving energy efficiency. In this section, the QIO algorithm is applied to a hybrid renewable energy system to reduce the total cost and improve operating efficiency.

Table 33
Statistical comparisons of various optimizers for gear train design.

Algorithms	Ave	Std	Worst	Best
QIO	3.9822E−10	6.7500E−10	**2.3576E−09**	**2.7009E−12**
GSA	1.4974E−04	2.7406E−04	1.0613E−03	9.9216E−10
ABC	**3.7859E−10**	**5.9703E−10**	**2.3576E−09**	**2.7009E−12**
ASO	1.9949E−08	2.8721E−08	1.0321E−07	**2.7009E−12**
WOA	5.4982E−09	7.6589E−09	2.7265E−08	**2.7009E−12**
AOA	6.3600E−08	8.8061E−08	2.5670E−07	1.3616E−09
CS	3.4598E−09	6.7774E−09	2.7265E−08	1.1661E−10
MFO	5.8318E−09	7.7254E−09	2.7265E−08	9.9216E−10
SSA	7.1856E−07	3.0176E−06	1.6584E−05	1.0936E−09
HHO	4.6709E−09	6.4203E−09	2.7265E−08	**2.7009E−12**
LFD	6.5116E−08	1.5124E−07	7.1469E−07	2.3078E−11
GA	4.7181E−09	9.7616E−09	3.9808E−08	**2.7009E−12**
PSO	4.0685E−08	7.1066E−08	3.0059E−07	**2.7009E−12**

Table 34
The best results of various optimizers for gear train design.

Algorithms	Optimal variables				Optimal cost
	T_a	T_b	T_d	T_f	
QIO	49	19	16	43	2.7009E−12
GSA	46	12	26	47	9.9216E−10
ABC	49	19	16	43	2.7009E−12
ASO	43	19	16	49	2.7009E−12
WOA	43	19	16	49	2.7009E−12
AOA	60	34	14	55	1.3616E−09
CS	48	17	22	54	1.1661E−10
MFO	46	26	12	47	9.9216E−10
SSA	31	15	17	57	1.0936E−09
HHO	43	16	19	49	2.7009E−12
LFD	51	13	30	53	2.3078E−11
GA	49	19	16	43	2.7009E−12
PSO	49	19	16	43	2.7009E−12

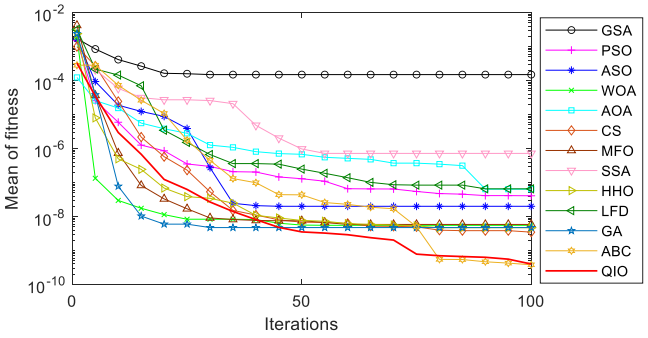

Fig. 44. Convergence comparisons of various optimizers for gear train design.

5.1. The microgrid model

Microgrid is a small power generation and distribution system, which is mainly composed of wind turbine (WT) generators, photovoltaic (PT) panels, diesel generators, energy storage batteries, load, and other ancillary equipment, including energy conversion devices, monitoring and protection devices and so on. The model structure of the microgrid is depicted in Fig. 45. The PV, WT, and diesel units can generate electricity; the energy storage batteries

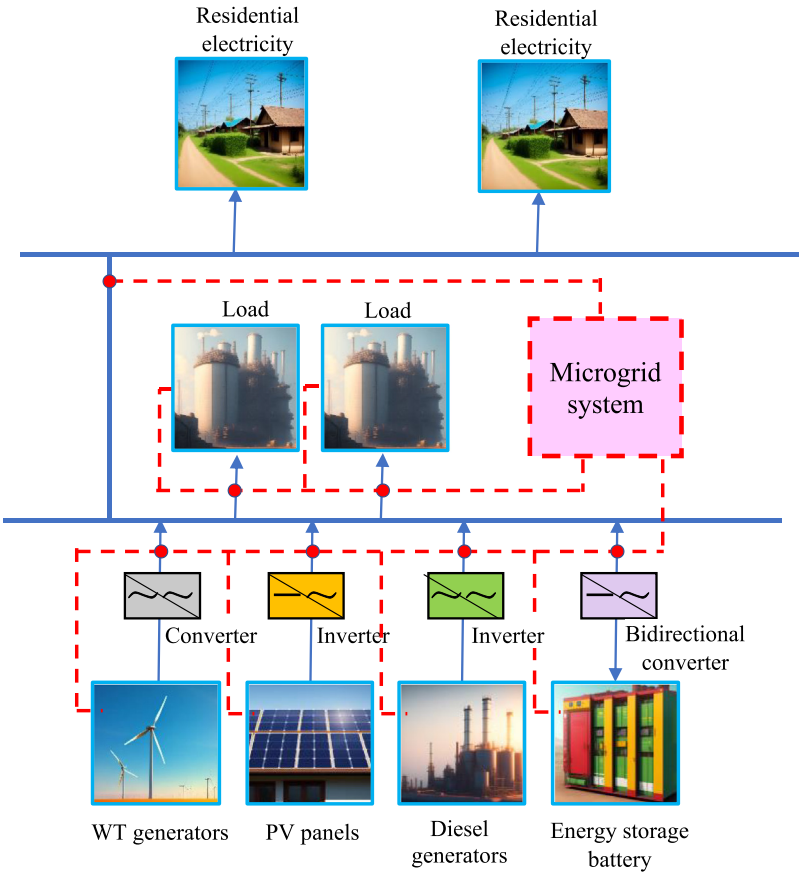

Fig. 45. The model structure of microgrid.

can be charged or discharged to adjust the microgrid; and the load can achieve a balance between supply and demand of energy.

The output power of the WT generators is expressed as [106]:

$$P_{WT}(t) = \begin{cases} 0 & if\ w(t) \le w_i \\ a \times w^3(t) - \beta \times P_r & if\ w_i \le w(t) < w_R \\ P_r & if\ w_R \le w(t) < w_o \\ 0 & if\ w(t) \ge w_o \end{cases} \tag{40}$$

where w is the wind velocity, P_r is the rated power of the wind turbine, w_R is the rated wind velocity, w_i andw_o are the cut-in and cut-out, respectively. The values of α and β are given as [106]:

$$\begin{cases} a = P_r/(w_R^3(t) - w_i^3) \\ \beta = w_i^3/(w_R^3(t) - w_i^3) \end{cases} \tag{41}$$

The output power of PV panels is expressed as [107]:

$$P_p(t) = R_t \cdot \eta_p \cdot A_p \tag{42}$$

$$\eta_p = \eta_r \cdot \eta_p \cdot (1 - N_T \cdot ((T_a - \frac{R_t(T_n - 20)}{800}) - T_r)) \tag{43}$$

where A_p denotes the PV area, R_t is the solar radiation, η_p is the efficiency of the PV panels, A_p is the area of PV panels, η_r is the reference efficiency, η_p is the power conditioning efficiency, T_r is the cell temperature at the reference conditions, T_a is the ambient air temperature, and T_n is the nominal cell operating temperature.

The consumption of the diesel generators can be expressed as [108]:

$$q(t) = a_0 \cdot P(t) + b_0 \cdot P_r \tag{44}$$

where $P(t)$ is the generated power, a_0 and b_0 are constant values.

The overall efficiency of the diesel generators is given as [108]:

$$\eta_o = \eta_{br} \cdot \eta_g \tag{45}$$

where η_o is the overall efficiency and η_{br} is the brake thermal efficiency.

The storage battery capacity is expressed as [108]:

$$C_b = \frac{E_l \cdot AD}{DOD \cdot \eta_i \cdot \eta_b} \tag{46}$$

where E_l is the load demand, AD is the autonomy of the battery, DOD represents the depth of discharge, η_i and η_b are the inverter and battery efficiency, respectively.

The storage battery can be charged or discharged by tracking changes in wind and solar energy outputs. The charging or discharging states of the storage battery are expressed as [109]:

$$C_{sb} = \begin{cases} C_{sb}(t-1) + (P_{total}(t) - \dfrac{P_{load}(t)}{\eta_{inv}})\eta_{sb}\Delta t & \text{charging} \\ C_{sb}(t-1) - (\dfrac{P_{load}(t)}{\eta_{inv}} - P_{total}(t))\eta_{sb}\Delta t & \text{discharging} \end{cases} \tag{47}$$

where C_{sb} is the battery capacity at time t, P_{total} is the total output, P_{load} is the total load, and η_{inv} and η_{sb} are the inverter efficiency and the charging–discharging efficiency of the battery, respectively. When the total output is more than the total load, the battery charges, or the battery discharges.

5.2. Objective function and constraints

The cost of the diesel generators is calculated as [109]:

$$C_{dg} = a + b \cdot P_{dg} + c \cdot P_{dg}^2 \tag{48}$$

where a, b and c are the coefficients of fuel cost, P_{dg} is the output power.

The capital cost of the PV generators is expressed as [109]:

$$C_{PV} = a_{PV} \cdot A_{PV} \tag{49}$$

where a_{PV} is the given cost of PV generators initially.

The constraints of the microgrid system are summarized as [109,110]:

$$\begin{cases} P_{i\,\min} \le P_i \le P_{i\,\max} \\ P_{bs\,\min} \le P_{bs} \le P_{bs\,\max} \\ C_{bs\,\min} \le P_{bs} \le C_{bs\,\max} \\ P_{dg\,\min} \le P_{dg} \le P_{dg\,\max} \\ P_{load} = P_{WT} + P_{PV} + P_{dg} + P_{battery} + P_{grid} \end{cases} \tag{50}$$

where P_{imin} and P_{imax} are the lower and upper limits of the output power of the battery, respectively, P_{bsmin} and P_{bsmax} are the lower and upper limits of the charging–discharging power, respectively, C_{bsmin} and C_{bsmax} are the lower and upper limits of the charging–discharging power, respectively, P_{dgmin} and P_{dgmax} are the lower and upper limits of the output power of diesel generators, respectively, $P_{battery}$ is the generating power of the battery, and P_{grid} is the power from that microgrid buys from the power grid.

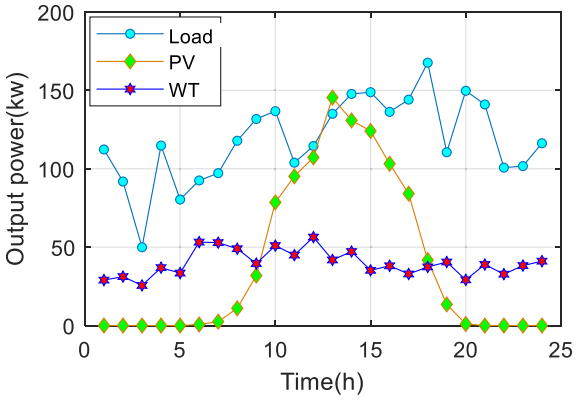

Fig. 46. Predicted output power of load, PV, and WT.

The objective function of a microgrid is to minimize the overall cost and the maximum economic income by adjusting the decision variables (P_i, P_{dg}). The objective function is expressed as:

$$\text{Minimize } f(x) = \sum_{t=1}^{T} (y_{\cos t}(t) - y_{income}(t)) \tag{51}$$

where T is the total scheduling time, y_{cost} is the total generation cost of the microgrid, and y_{income} is the economic income.

5.3. Results and discussion

This study takes a certain region as an example and simulates the output of each component in the microgrid. The predicted output powers of the load system, the PV operators, and the WT operators are plotted in Fig. 46.

The operation management of a microgrid is solved by QIO method as well as GSA, ABC, ASO, WOA, AOA, CS, MFO, SSA, HHO, LFD, GA, and PSO. The population size is set as 50, and the maximum number of iterations is set as 100. The total time of operation management is set to 24 h, so the number of the variables to be optimized is 48, including 24 output powers of diesel operators and 24 output powers of batteries. Fig. 47 depicts the convergence curves of all the algorithms for this issue. QIO obtains a better convergence rate, exhibiting a superior convergence performance over its competitors. The optimal output powers of batteries provided by various algorithms are illustrated in Fig. 48, in which the charging or the discharging states of the batteries at different time periods are given quantitatively, it is obvious that QIO identifies the best battery management scheme which contributes to the minimum cost.

Figs. 48 and 49 depicts the output powers of batteries and diesel generators provided by various algorithms at different time periods. Although all the algorithms can adjust the operating states of diesel generators, QIO provides the best operating results among all the algorithms. Obviously, the output powers of the batteries and diesel generators are adjusted simultaneously according to the optimal objective. The results highlight that QIO can address this challenging task in the best way. This again proves the superior optimization capacity of QIO when tackling complex real-world problems.

6. Conclusions and future research

This research introduces a novel math-based optimizer named QIO for global optimization. In this study, a GQI method is developed initially to obtain the minimizer of any quadratic interpolation function. Building upon this, exploration and exploitation strategies are established. In the exploring strategy, QIO utilizes randomly chosen individuals from the population along with the current individual to generate the minimizer and update the individual. In the exploiting strategy, QIO utilizes the individuals randomly chosen from the population and the best individual found thus far to generate the minimizer and update the individual. The classical benchmark functions consisting

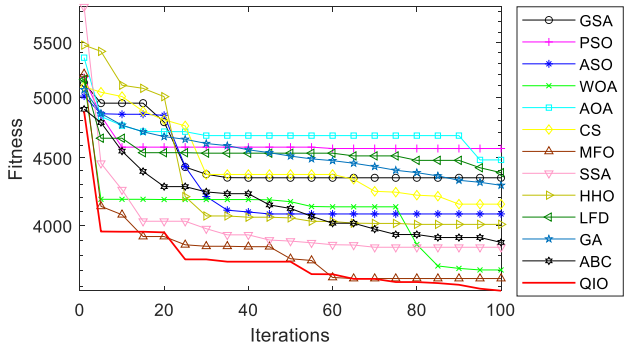

Fig. 47. Convergence curves of various algorithms for operation management of the microgrid.

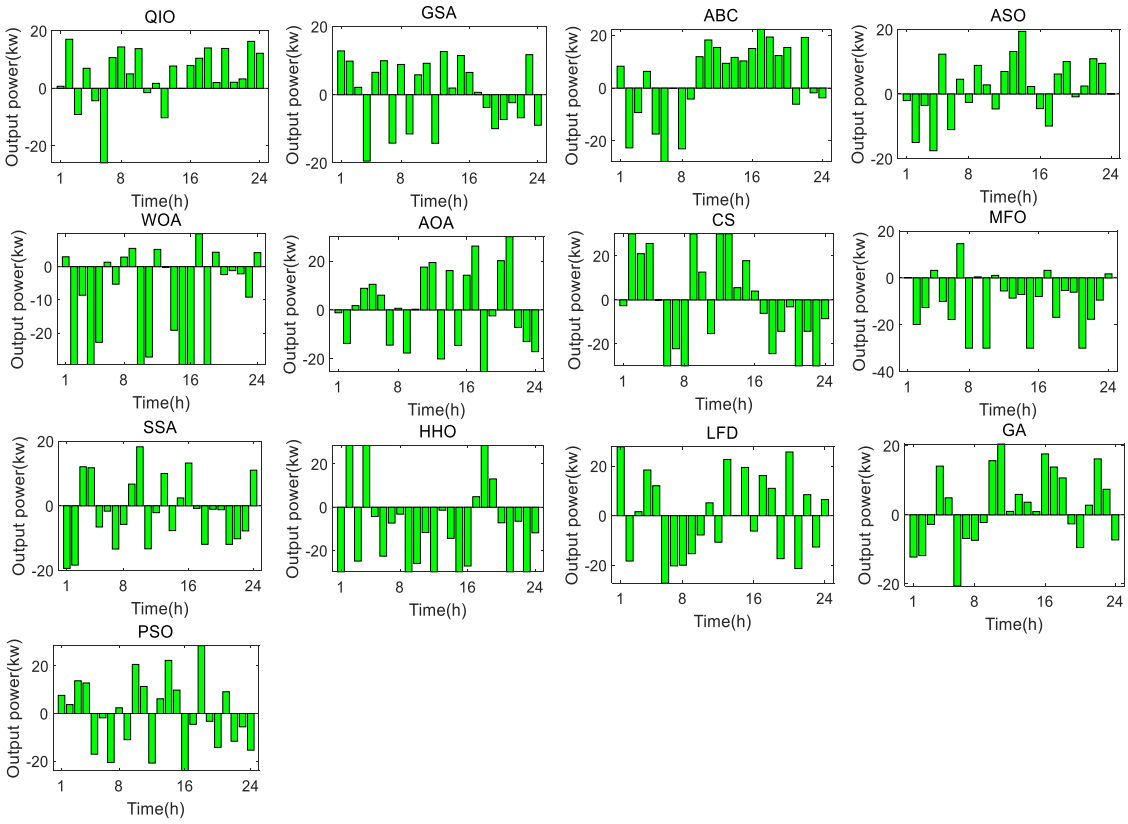

Fig. 48. The optimal output powers of batteries provided by different algorithms.

of 23 functions and the CEC-2014 test suite are employed to evaluate the efficiency of QIO by comparing with 12 other meta-heuristic methods, including GSA, ASO, WOA, AOA, CS, MFO, SSA, HHO, LFD, PSO, GA, and ABC. Several important metrics, including WRST, FAT and QR, are used for comparison from various aspects. These results demonstrate that QIO yields very promising performance and is superior to its competitors in the majority of the benchmark functions, indicating QIO is an excellent alternative optimizer when tackling a wide range of different optimization problems. The practicality of QIO is examined by solving ten engineering cases and its performance is compared with that of other optimizers. The results of the EQPs also show that QIO has better optimization capacity in tackling some problems with unknown search spaces. Eventually, QIO is applied in the operation management of a microgrid with energy storage systems. The comparative results show that QIO is

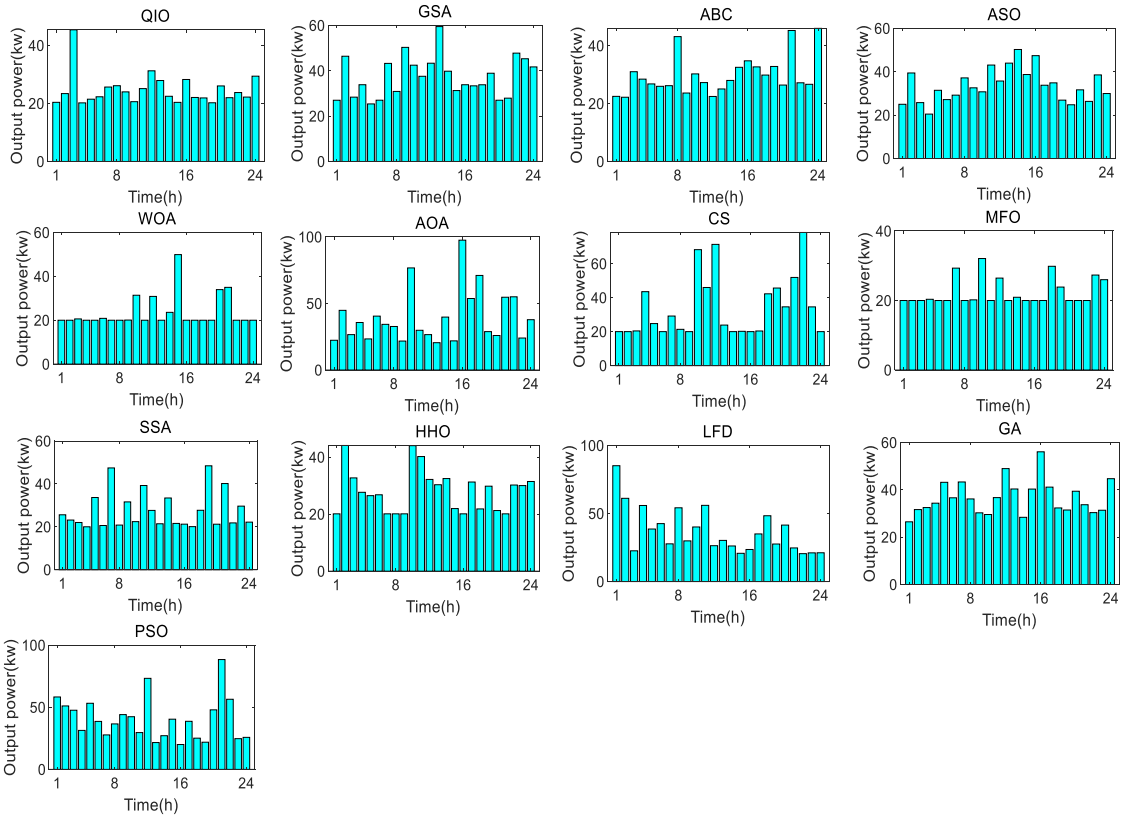

Fig. 49. The optimal output powers of diesel generators provided by different algorithms.

superior to its competitors when searching for the power output of diesel operators and energy batteries. These results from engineering and real-world problems prove that QIO is remarkably capable to solve nonlinear, high-dimension, and complex engineering problems.

The above experimental results provide clear evidence of QIO's significant success in comparison to other optimization algorithms. This achievement can be attributed to the following key points.

- The proposed GQI method can find the minimizer of the quadratic interpolation function formed by any three points. This method allows the algorithm to perform efficient searches along each dimension in the search space.
- The GQI method via the randomly chosen individuals contributes significantly to the algorithm's exploration. The GQI method via the best individual found so far is beneficial to the algorithm's exploitation.
- The exploration weight can effectively avoid local extrema and premature convergence.
- The QIO algorithm requires fewer parameters when solving optimization problems, significantly enhancing its ease of use and simplicity.

Despite its superior performance, QIO also presents several limitations. One limitation is QIO is less powerful in handling certain discrete problems, this is because the solution space of binary problems exhibits exponential growth, which results in a reduced efficiency of optimization algorithms when handling complex binary problems. Another limitation is the algorithm is relatively inefficient in solving certain engineering problems, which attribute to the inherent nature of optimization algorithms. Nevertheless, the limitations of QIO open up avenues for promising research directions and pave the way for future breakthroughs. For future studies, QIO can be employed to solve different hydraulic engineering problems, like the optimal operation of hydraulic units, the optimal allocation of water resources, and construction schedule optimization. Moreover, developing the binary and multi-objective versions of QIO for real-world problems is an enticing research task. Some aspects of QIO can be further improved

in the future. For instance, other interpolation methods, such as the cubic interpolation method and the Newton interpolation method, can be considered in QIO to improve its performance.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (11972144 and 12072098).

Appendix

See Table A.1.

Table A.1
23 benchmark functions.

Name	Functions	d	Range	f_{opt}
Sphere	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,\ 100]^d$	0
Schwefel 2.22	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10,\ 10]^d$	0
Schwefel 1.2	$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-100,\ 100]^d$	0
Schwefel 2.21	$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	$[-100,\ 100]^d$	0
Rosenbrock	$F_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i)^2) + (x_i - 1)^2$	30	$[-30,\ 30]^d$	0
Step	$F_6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	30	$[-100,\ 100]^d$	0
Quartic	$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random\,[0, 1)$	30	$[-1.28,\ 1.28]^d$	0
Schwefel	$F_8(x) = -\sum_{i=1}^{n} (x_i \sin(\sqrt{\lvert x_i \rvert}))$	30	$[-500,\ 500]^d$	-12569.5
Rastrigin	$F_9(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)^2$	30	$[-5.12,\ 5.12]^d$	0
Ackley	$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i) + 20 + e$	30	$[-32,\ 32]^d$	0
Griewank	$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} (x_i - 100)^2 - \prod_{i=1}^{n} \cos(\frac{x_i - 100}{\sqrt{i}}) + 1$	30	$[-600,\ 600]^d$	0

(*continued on next page*)

Table A.1 (*continued*).

Penalized	$F_{12}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1)] + (y_n-1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$	30	$[-50, 50]^d$	0
Penalized 2	$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{29}(x_i-1)^2 p[1+\sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+\sin^2(2\pi x_{30})]\} + \sum_{i=1}^{30} u(x_i, 5, 10, 4)$	30	$[-50, 50]^d$	0
Foxholes	$F_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{j=1}^{2}(x_i - a_{ij})^6}\right]^{-1}$	2	$[-65.536, 65.536]^d$	0.998
Kowalik	$F_{15}(x) = \sum_{i=1}^{11}\left\lvert a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right\rvert^2$	4	$[-5, 5]^d$	3.075×10^{-4}
Six Hump Camel	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^d$	-1.0316
Branin	$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1-\frac{1}{8\pi})\cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
GoldStein–Price	$F_{18}(x) = \left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right] \times \left[30+(2x_1+1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$	2	$[-2, 2]^d$	3
Hartman 3	$F_{19}(x) = -\sum_{i=1}^{4}\exp\left[-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	3	$[0, 1]^d$	-3.86
Hartman 6	$F_{20}(x) = -\sum_{i=1}^{4}\exp\left[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right]$	6	$[0, 1]^d$	-3.322
Shekel 5	$F_{21}(x) = -\sum_{i=1}^{5}\left\lvert(x_i-a_i)(x_i-a_i)^T + c_i\right\rvert^{-1}$	4	$[0, 10]^d$	-10.1532
Shekel 7	$F_{22}(x) = -\sum_{i=1}^{7}\left\lvert(x_i-a_i)(x_i-a_i)^T + c_i\right\rvert^{-1}$	4	$[0, 10]^d$	-10.4028
Shekel 10	$F_{23}(x) = -\sum_{i=1}^{10}\left\lvert(x_i-a_i)(x_i-a_i)^T + c_i\right\rvert^{-1}$	4	$[0, 10]^d$	-10.5364

References

[1] F. Glover, M. Laguna, Tabu Search, Springer US, 1998, pp. 2093–2229.

[2] K. Deb, Multi-Objective Optimisation using Evolutionary Algorithms: An Introduction, Springer London, 2011, pp. 3–34.

[3] H. Ishibuchi, T. Murata, Multi-objective genetic local search algorithm, in: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE, 1996, pp. 119–124.

[4] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341.

[5] J. Kennedy, Eberhart R., Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, vol. 4, IEEE, 1995, pp. 1942–1948.

[6] S. Lee, J. Park, N. Kim, T. Lee, L. Quagliato, Extreme gradient boosting-inspired process optimization algorithm for manufacturing engineering applications, Mater. Des. (2023) 111625.

[7] C. Klanke, S. Engell, Scheduling and batching with evolutionary algorithms in simulation–optimization of an industrial formulation plant, Comput. Ind. Eng. 174 (2022) 108760.

[8] L. Liu, X. Qiao, W.Z. Liang, J. Oboamah, J. Wang, D.R. Rudnick, Y. …Shi, An edge-computing flow meter reading recognition algorithm optimized for agricultural IoT network, Smart Agric. Technol. 5 (2023) 100236.

[9] Y. Jiang, K. Hao, X. Cai, Y. Ding, An improved reinforcement-immune algorithm for agricultural resource allocation optimization, J. Comput. Sci. 27 (2018) 320–328.

[10] F. Lazzari, G. Mor, J. Cipriano, F. Solsona, D. Chemisana, D. Guericke, Optimizing planning and operation of renewable energy communities with genetic algorithms, Appl. Energy 338 (2023) 120906.
[11] H. Youssef, S. Kamel, M.H. Hassan, L. Nasrat, Optimizing energy consumption patterns of smart home using a developed elite evolutionary strategy artificial ecosystem optimization algorithm, Energy (2023) 127793.
[12] S.P. Singh, W. Viriyasitavat, S. Juneja, H. Alshahrani, A. Shaikh, G. Dhiman, A. .Kaur, Dual adaption based evolutionary algorithm for optimized the smart healthcare communication service of the internet of things in smart city, Phys. Commun. 55 (2022) 101893.
[13] N. Priyadharshini, N. Selvanathan, B. Hemalatha, C. Sureshkumar, A novel hybrid extreme learning machine and teaching–learning-based optimization algorithm for skin cancer detection, Healthc. Anal. 3 (2023) 100161.
[14] Z. Pan, L.W. Zhang, K.M. Liew, Adaptive surrogate-based harmony search algorithm for design optimization of variable stiffness composite materials, Comput. Methods Appl. Mech. Engrg. 379 (2021) 113754.
[15] P.C. Chervith, P. Abhilash, A. Chandrasekhar, Optimizing the tool life of a hybrid material using genetic algorithm, Mater. Today: Proc. 5 (13) (2018) 27199–27204.
[16] M. Mallick, A. Chakrabarty, N. Khutia, Genetic algorithm based design optimization of crashworthy honeycomb sandwiched panels of AA7075-t651 aluminium alloy for aerospace applications, Mater. Today: Proc. 54 (2022) 690–696.
[17] J.F. Wang, J. Periaux, M. Sefrioui, Parallel evolutionary algorithms for optimization problems in aerospace engineering, J. Comput. Appl. Math. 149 (1) (2002) 155–169.
[18] H.O. Bae, S.Y. Ha, M. Kang, H. Lim, C. Min, J. Yoo, A constrained consensus based optimization algorithm and its application to finance, Appl. Math. Comput. 416 (2022) 126726.
[19] S.R. Das, D. Mishra, M. Rout, Stock market prediction using firefly algorithm with evolutionary framework optimized feature reduction for OSELM method, Expert Syst. Appl.: X 4 (2019) 100016.
[20] G. Hu, J. Zhong, G. Wei, C.T. Chang, DTCSMO: An efficient hybrid starling murmuration optimizer for engineering applications, Comput. Methods Appl. Mech. Engrg. 405 (2023) 115878.
[21] S. Yang, X. Gu, Y. Liu, R. Hao, S. Li, A general multi-objective optimized wavelet filter and its applications in fault diagnosis of wheelset bearings, Mech. Syst. Signal Process. 145 (2020) 106914.
[22] R. Zulfiqar, T. Javid, Z.A. Ali, V. Uddin, Novel metaheuristic routing algorithm with optimized energy and enhanced coverage for WSNs, Ad Hoc Netw. 144 (2023) 103133.
[23] N. Wang, J. Liu, J. Lu, X. Zeng, X. Zhao, Low-delay layout planning based on improved particle swarm optimization algorithm in 5G optical fronthaul network, Opt. Fiber Technol., Mater. Devices Syst. 67 (2021) 102736.
[24] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 114 (2022) 105082.
[25] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[26] M. Pant, R. Thangaraj, V.P. Singh, Optimization of mechanical design problems using improved differential evolution algorithm, Int. J. Recent Trends Eng. 1 (5) (2009) 21.
[27] J.R. Koza, J.R. Koza, Genetic Programming: On the Programming of Computers By Means of Natural Selection, MIT Press, 1992.
[28] H.-G. Beyer, H.-P. Schwefel, Evolution strategies–a comprehensive introduction, Nat. Comput. 1 (1) (2002) 3–52.
[29] I. Zelinka, SOMA—self-organizing migrating algorithm, in: Self-Organizing Migrating Algorithm: Methodology and Implementation, 2016, pp. 3–49.
[30] L. Ren, Z. Wang, J. Zhao, J. Wu, R. Lin, J. Wu, Y. Fu, D. Tang, Shale gas load recovery modeling and analysis after hydraulic fracturing based on genetic expression programming: A case study of southern sichuan basin shale, J. Nat. Gas Sci. Eng. 107 (2022) 104778.
[31] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[32] J. Kennedy, R. Eberhart, Particle Swarm Optimization, vol. 4, 1942–1948.
[33] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (1) (2009) 108–132.
[34] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[35] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Futur. Gener. Comput. Syst. 111 (2020) 300–323.
[36] H. Zamani, M.H. Nadimi-Shahraki, A.H. Gandomi, Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization, Comput. Methods Appl. Mech. Engrg. 392 (2022) 114616.
[37] M. Braik, A. Hammouri, J. Atwan, M.A. Al-Betar, M.A. Awadallah, White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems, Knowl.-Based Syst. 243 (2022) 108457.
[38] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Gazelle optimization algorithm: A novel nature-inspired metaheuristic optimizer, Neural Comput. Appl. (2022) 1–33.
[39] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey badger algorithm: New metaheuristic algorithm for solving optimization problems, Math. Comput. Simulation 192 (2022) 84–110.
[40] M. Dehghani, Z. Montazeri, E. Trojovská, P. Trojovský, Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, Knowl.-Based Syst. 259 (2023) 110011.
[41] W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, Comput. Methods Appl. Mech. Engrg. 388 (2022) 114194.
[42] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2019) 715–734.
[43] D. Połap, M. Woźniak, Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism, Symmetry 9 (10) (2017) 203.
[44] D. Połap, M. Woźniak, Red fox optimization algorithm, Expert Syst. Appl. 166 (2021) 114107.

[45] M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, Knowl.-Based Syst. (2023) 110248.
[46] D. Bertsimas, J. Tsitsiklis, Simulated annealing, Stat. Sci. 8 (1) (1993) 10–15.
[47] A. Lam, V.O. Li, Chemical reaction optimization: a tutorial, Memet. Comput. 4 (1) (2012) 3–17.
[48] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[49] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, Knowl.-Based Syst. 163 (2019) 283–304.
[50] J.L.J. Pereira, M.B. Francisco, C.A. Diniz, G.A. Oliver, S.S. Cunha Jr., G.F. Gomes, Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization, Expert Syst. Appl. 170 (2021) 114522.
[51] L. Abualigah, Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications, Neural Comput. Appl. 32 (16) (2020) 12381–12401.
[52] M. Abdel-Basset, D. El-Shahat, M. Jameel, M. Abouhawwash, Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115652.
[53] L. Xie, J. Zeng, Z. Cui, General framework of artificial physics optimization algorithm, in: 2009 World Congress on Nature and Biologically Inspired Computing (NaBIC), IEEE, 2009.
[54] S. Moein, R. Logeswaran, Kgmo: A swarm optimization algorithm based on the kinetic energy of gas molecules, Inform. Sci. 275 (2014) 127–144.
[55] F.A. Hashim, et al., Henry gas solubility optimization: A novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[56] A.H. Kashan, A new metaheuristic for optimization: optics inspired optimization (oio), Comput. Oper. Res. 55 (2015) 99–125.
[57] C.L. Chuang, J.A. Jiang, Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space–time, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 3157–3164.
[58] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[59] A. Kaveh, Siamak Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3) (2010) 267–289.
[60] B. Alatas, ACROA: artificial chemical reaction optimization algorithm for global optimization, Expert Syst. Appl. 38 (10) (2011) 13170–13180.
[61] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Appl. Intell. 51 (2021) 1531–1551.
[62] M. Azizi, U. Aickelin, H.A. Khorshidi, M. Baghalzadeh Shishehgarkhaneh, Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization, Sci. Rep. 13 (1) (2023) 1–23.
[63] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[64] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 4661–4667.
[65] Q. Askari, I. Younas, M. Saeed, Political optimizer: A novel socio-inspired meta-heuristic for global optimization, Knowl.-Based Syst. 195 (2020) 105709.
[66] D. Panwar, G. Saini, P. Agarwal, Human eye vision algorithm (HEVA) a novel approach for the optimization of combinatorial problems, in: Artificial Intelligence in Healthcare, Springer, 2022, pp. 61–71.
[67] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[68] S.H.S. Moosavi, V.K. Bardsiri, Poor and rich optimization algorithm: A new human-based and multi populations algorithm, Eng. Appl. Artif. Intell. 86 (2019) 165–181.
[69] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, Swarm Evol. Comput. 17 (2014) 14–24.
[70] Y. Shi, Brain storm optimization algorithm, in: International Conference in Swarm Intelligence, Springer, Berlin, Heidelberg, 2011, pp. 303–309.
[71] M. Dehghani, E. Trojovská, T. Zuščák, A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training, Sci. Rep. 12 (1) (2022) 1–24.
[72] N. Ghorbani, E. Babaei, Exchange market algorithm, Appl. Soft Comput. 19 (2014) 177–187.
[73] A.H. Kashan, League championship algorithm: a new algorithm for numerical function optimization, in: 2009 International Conference of Soft Computing and Pattern Recognition, IEEE, 2009, pp. 43–48.
[74] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[75] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609.
[76] I. Ahmadianfar, A.A. Heidari, A.H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: An efficient optimization algorithm based on runge kutta method, Expert Syst. Appl. 181 (2021) 115079.
[77] M. Abdel-Basset, D. El-Shahat, M. Jameel, M. Abouhawwash, Exponential distribution optimizer (EDO) a novel math-inspired algorithm for global optimization and engineering problems, Artif. Intell. Rev. (2023) 1–72.
[78] I. Ahmadianfar, A.A. Heidari, S. Noshadian, H. Chen, A.H. Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, Expert Syst. Appl. 195 (2022) 116516.
[79] Tanyildizi E., Demir, G, Golden sine algorithm: A novel math-inspired algorithm, Adv. Electr. Comput. Eng. 17 (2017) 71–78.
[80] M.H. Qais, H.M. Hasanien, R.A. Turky, S. Alghuwainem, M. Tostado-Véliz, F. Jurado, Circle search algorithm: A geometry-based metaheuristic optimization algorithm, Mathematics 10 (10) (2022) 1626.

[81] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[82] K. Vandebogert, Method of Quadratic Interpolation (Ph.D. thesis), University of South Carolina, Columbia, SC, USA, 2017.
[83] Optimization Theory and Methods: Nonlinear Programming. Wenyu Sun, YaXiang Yuan. 89–98.
[84] Y. Sun, T. Yang, Z. Liu, A whale optimization algorithm based on quadratic interpolation for high-dimensional global optimization problems, Appl. Soft Comput. 85 (2019) 105744.
[85] X. Chen, C. Mei, B. Xu, K. Yu, X. Huang, Quadratic interpolation based teaching-learning-based optimization for chemical dynamic system optimization, Knowl.-Based Syst. 145 (2018) 250–263.
[86] M. Qaraad, S. Amjad, N.K. Hussein, M.A. Elhosseini, An innovative quadratic interpolation salp swarm-based local escape operator for large-scale global optimization problems and feature selection, Neural Comput. Appl. 34 (20) (2022) 17663–17721.
[87] K. Deep, K.N. Das, Quadratic approximation based hybrid genetic algorithm for function optimization, Appl. Math. Comput. 203 (2008) 86–98.
[88] W. Zhao, Z. Zhang, L. Wang, Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications, Eng. Appl. Artif. Intell. 87 (2020) 103300.
[89] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Technical Report, vol. 635, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, Nanyang Technological University, Singapore, 2013, (2).
[90] X.S. Yang, S. Deb, Cuckoo search: recent advances and applications, Neural Comput. Appl. 24 (1) (2014) 169–174.
[91] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[92] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[93] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[94] E.H. Houssein, M.R. Saad, F.A. Hashim, H. Shaban, M. Hassaballah, Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 94 (2020) 103731.
[95] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (1937) 674–701.
[96] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[97] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[98] C.V. Camp, M. Farshchin, Design of space trusses using modified teaching-learning based optimization, Eng. Struct. 62-63 (2014) 87–97.
[99] H. Adeli, O. Kamal, Efficient optimization of space trusses, Comput. Struct. 24 (3) (1986) 501–511.
[100] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization method for optimum discrete design of truss structures, Comput. Struct. 139 (2014) 43–53.
[101] A. Osyczka, S. Krenich, Some methods for multicriteria design optimization using evolutionary algorithms, J. Theoret. Appl. Mech. 42 (3) (2004) 565–584.
[102] Y. Wang, H. Liu, H. Long, Z. Zhang, S. Yang, Differential evolution with a new encoding mechanism for optimizing wind farm layout, IEEE Trans. Ind. Inf. 14 (3) (2018) 1040–1054.
[103] G. Hu, J. Zhong, B. Du, G. Wei, An enhanced hybrid arithmetic optimization algorithm for engineering applications, Comput. Methods Appl. Mech. Engrg. 394 (2022) 114901.
[104] B.R. Rao, R. Tiwari, Optimum design of rolling element bearings using genetic algorithms, Mech. Mach. Theor. 42 (2) (2007) 233–250.
[105] Y.L. Hsu, T.C. Liu, Developing a fuzzy proportional–derivative controller optimization engine for engineering design optimization problems, Eng. Optim. 39 (6) (2007) 679–700.
[106] D. Guangqian, K. Bekhrad, P. Azarikhah, A. Maleki, A hybrid algorithm based optimization on modeling of grid independent biodiesel-based hybrid solar/wind systems, Renew. Energy 122 (2018) 551–560.
[107] A. Maleki, H. Hafeznia, M.A. Rosen, F. Pourfayaz, Optimization of a grid-connected hybrid solar-wind-hydrogen CHP system for residential applications by efficient metaheuristic approaches, Appl. Therm. Eng. 123 (2017) 1263–1277.
[108] M.A. Ramli, H.R.E.H. Bouchekara, A.S. Alghamdi, Optimal sizing of PV/wind/diesel hybrid microgrid system using multi-objective self-adaptive differential evolution algorithm, Renew. Energy 121 (2018) 400–411.
[109] J. Wang, L. Huang, Y. Yang, Optimal dispatch of microgrid based on mult-objective particle swarm optimization, Power Syst. Clean Energy 30 (1) (2014) 49–54.
[110] Y. Zhang, Economic scheduling of microgrid based on improved genetic algorithm, Light. Electr. 5 (2022) 210–212.