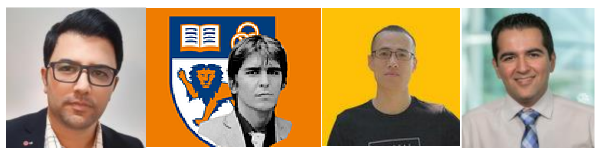
INFO: An Efficient Optimization Algorithm based on Weighted Mean of Vectors

Iman Ahmadianfar[a]*, Ali Asghar Heidari[b,c], Saeed Noshadian[a], Huiling Chen[d], Amir H Gandomi[e]**



[a] *Department of Civil Engineering, Behbahan Khatam Alanbia University of Technology, Behbahan, Iran*
Email: im.ahmadian@gmai.com, i.ahmadianfar@bkatu.ac.ir (Iman Ahmadianfar).
Saeed.noshadian@gmail.com (Saeed Noshadian).
[b] *School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1439957131, Iran*
Email: as_heidari@ut.ac.ir, aliasghar68@gmail.com
[c] *Department of Computer Science, School of Computing, National University of Singapore, Singapore 117417, Singapore*
Email: aliasgha@comp.nus.edu.sg, t0917038@u.nus.edu
[d] *College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, Zhejiang 325035, China*
Email: chenhuiling.jlu@gmail.com
[e] *Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia*
Email: gandomi@uts.edu.au

***Corresponding Authors: Iman Ahmadianfar and Huiling Chen**

E-mail: *im.ahmadian@gmail.com* (Iman Ahmadianfar), chenhuiling.jlu@gmail.com (Huiling Chen)

All codes and source files are available at https://aliasgharheidari.com/INFO.html

Abstract
This study presents the analysis and principle of an innovative optimizer named weIghted meaN oF vectOrs (INFO) to optimize different problems. INFO is a modified weight mean method, whereby the weighted mean idea is employed for a solid structure and updating the vectors' position using three core procedures: *updating rule*, *vector combining*, and a *local search*. The *updating rule* stage is based on a mean-based law and convergence acceleration to generate new vectors. The *vector combining* stage creates a combination of obtained vectors with the updating rule to achieve a promising solution. The updating rule and vector combining steps were improved in INFO to increase the exploration and exploitation capacities. Moreover, the *local search* stage helps this algorithm escape low-accuracy solutions and improve exploitation and convergence. The performance of INFO was evaluated in 48 mathematical test functions, and five constrained engineering test cases. According to the literature, the results demonstrate that INFO outperforms other basic and advanced methods in terms of exploration and exploitation. In the case of engineering problems, the results indicate that the INFO can converged to 0.99% of the global optimum solution. Hence, the INFO algorithm is a promising tool for optimal designs in optimization problems, which stems from the considerable efficiency of this algorithm for optimizing constrained cases. The source codes of this algorithm will be publicly available at https://imanahmadianfar.com. and https://aliasgharheidari.com/INFO.html.

Keywords: Optimization; Swam-intelligence; Exploration; Exploitation; Weighted Mean of Vectors Algorithm

1. Introduction

With the development of society, people will face more and more complex problems. However, solving a class of complex problems is the essential requirement for promoting social development. Although many traditional numerical and analytical methods have carried out relevant analysis research, some deterministic methods cannot provide a fitting solution to solve several challenging problems with non-convex and highly non-linear search domains since the complexity and dimensions of these problems grow exponentially. Optimizing the problems by applying some deterministic methods, such as the Lagrange and Simplex methods, requires some initial information of the problem and complicated computations. Thus, exploring global optimum solution problems using such methods for those levels of problems is not always possible or feasible [1]. Therefore, it is still urgent to develop an efficient method to solve the increasingly complex optimization problems. Actually, optimization methods can have multiple forms and formulations, maybe no limit in form, and what they essential for them in stochastic class is a core for exploration and a core for exploitation, which can be utilized to deal with those forms of problems, such as multi-objective optimization, fuzzy optimization, robust optimization, memetic optimization, large scale optimization, many-objective optimization methods, and single-objective optimization. One common optimization method, named swarm intelligence (SI) algorithms, is swarm-based optimization based on the evolution of an initial set of agents and attraction of agents towards better solutions, which, in an extreme case, is the optimum solution and avoids locally optimal solutions. The swarm intelligence optimization algorithm has intelligent characteristics such as self-adaptation, self-learning, and self-organization and is convenient for large-scale parallel computing. It is a trendy optimization technology.

In recent years, some classes of swarm-based optimization algorithms have been applied as simple and reliable methods for realizing the solutions of problems in both the computer science field and industry. Numerous researchers have demonstrated that swarm-based optimization is very promising for tackling many challenging problems [2-4]. Some algorithms employ methods that mimic natural evolutionary mechanisms and basic genetic rules, such as selection, reproduction, mutation, and migration [5]. One of the most popular evolutionary methods is the genetic algorithm (GA) introduced by Holland [6]. With its unique three core operations of crossover, variation, and selection, GA has achieved outstanding performance in many optimization problems, such as twice continuously differentiable NLP problems [7], predicting production, and neural architectures searching. Other well-regarded evolutionary algorithms include differential evolution (DE) [8] and evolutionary strategies (ES) [9]. This kind of evolutionary algorithm simulates the way of biological evolution in nature and has strong adaptability to problems. Moreover, the rise of deep neural networks (DNN) in recent years has made people pay more attention to how to design neural network architecture automatically. Therefore, network architecture search (NAS) based on evolutionary algorithms has become a hot topic [10]. Some methods are motivated by physical laws, such as simulated annealing (SA) [11]. As one of the most well-known methods in this family presented by Kirkpatrick et al. [11], SA simulates the annealing mechanism utilized in physical material sciences. Also, with its excellent local search capabilities, SA can find more potential solutions in many engineering problems than other traditional SI algorithms [12-14]. One of the latest well-established methods is the gradient-based optimizer (GBO)[1], which considers Newtonian logic to explore suitable regions and achieve the global solution [15]. The method has been applied in many fields, including feature selection[16] and parameter estimation of photovoltaic models[16]. Most swarm methods mimic the equations of particle swarm optimization (PSO) by varying the basis of inspiration around collective social behaviors of animal groups [17]. Particle swarm optimization (PSO) is one of the most successful algorithms in this class, which was inspired by birds' social and individual intelligence when flocking [18]. In detail, PSO has a few parameters that need to be adjusted, also, unlike other methods, PSO has a memory machine, and the knowledge of particles with better performance can be preserved, which can help the algorithm find the optimal solution more quickly. Currently, PSO has taken its place in the fields of large-scale optimisation problems[19], feature selection[20], single-objective optimization problem[21], multi-objective optimisation problems[22], and high-dimensional expensive problem[23]. Ant colony optimization (ACO) is another popular approach based on ants' foraging behavior [24]. In particular, the concept of pheromone is a major feature of ACO. According to pheromone secreted by the ants in the process of searching for food, it can help the population to find a better solution at a faster rate. As soon as ACO was proposed, it was applied to the traveling salesman problem of 3-spots [25] and some complex optimization problems[26], and achieved satisfactory results.

While these optimization methods can solve various challenging and real optimization problems, the No Free Lunch (NFL) theorem authorizes researchers to present a new variant of methods or a new optimizer from scratch [27]. This theory states that no optimization method can work as the best tool for all problems. Accordingly, one algorithm can be the most suitable approach to solve several problems but is incompetent for other optimization problems. Hence, it can be declared that this theory is the basis of many studies in this field. In this research, we were motivated to improve upon novel metaheuristic methods that suffer from weak performance, have verification bias, and underperform compared to other existing methods [28-30]. As such, the proposed INFO algorithm is a forward-thinking, innovative attempt against such methods that provides a promising platform for the future of

[1] https://imanahmadianfar.com

optimization literature in computer science. Furthermore, we aim to apply this method to a variety of optimization problems and make it a scalable optimizer.

In this paper, we designed a new optimizer (INFO) by modifying the weighted mean method and updating the vectors' position, which can help form a more robust structure. In detail, updating rule, vector combining, and local search are the three core processes of INFO. Unlike other methods, the updating rule based on the mean is used to generate new vectors in INFO, thus accelerating the convergence speed. In the vector combination stage, two vectors acquired in the vector update stage are combined to produce a new vector for improving local search ability. This operation ensures the diversity of the population to a certain extent. Taking into account the global optimal position and the mean-based rule, a local operation is executed, which can effectively improve the problem of INPO being vulnerable to local optimal. This work's primary goal was to introduce the above three core processions for optimizing various kinds of optimization cases and engineering problems, such as structural and mechanical engineering problems and water resources systems. The INFO algorithm employs the concept of weighted mean to move agents toward a better position. This main motive behind INFO emphasizes its performance aspects to potentially solve some of the optimization problems that other methods cannot solve. It should be noted that there is no inspiration part in INFO, and it is tried to move the field to go beyond the metaphor.

The rest of this paper is organized as follows. In Sections 2 and 3, the main structures of INFO are described in detail. The set of mathematical benchmark functions employed to assess the efficiency of INFO is presented in Section 4. Section 5 solves four real engineering problems to show the capability of the proposed algorithm. Lastly, Section 6 expresses the conclusions of this study and gives some ideas for future researches.

2. Literature review

This section describes the previous studies on optimization methods and presents this research's primary motivation. Generally, evolutionary algorithms are classified into two types: single-based and population-based algorithms [17, 31]. In the first case, the algorithm's search process begins with a single solution and updates its position during the optimization process. The most well-known single-solution-based algorithms include simulated annealing (SA) [11], tabu search (TS) [32], and hill-climbing [33]. These algorithms allow easy implementation and require only a small number of function evaluations during optimization. However, the disadvantages are the high possibility of trapping in local optima and failure to exchange information because these methods have only a single trend.

Conversely, the optimization process in population-based algorithms begins with a set of solutions and updates their position during optimization. GA, DE, PSO, artificial bee colony (ABC) [34], ant colony optimization (ACO) [35-37], slime mould algorithm (SMA)[2] [38], and Harris hawks optimization (HHO)[3] [39-41] are some of the population-based algorithms. These methods have a high capacity to escape local optimal solutions because they use a set of solutions during optimization. Moreover, the exchange of information can be shared between solutions, which helps them to better search in difficult search spaces. However, these algorithms require a large number of function evaluations during optimization and high computational costs.

According to the above discussion, the population-based algorithms are considered more reliable and robust optimization methods than single-solution-based algorithms.

[2] https://aliasgharheidari.com/SMA.html
[3] https://aliasgharheidari.com/HHO.html

Generally, an algorithm's best formulation is explored by evaluating it on different types of benchmark and engineering problems.

Ordinarily, optimizers employ one or more operators to perform two phases: exploration and exploitation. An optimization algorithm requires a search mechanism to find promising areas in the search space, which is done in the exploration phase. The exploitation phase improves the local search ability and convergence speed to achieve promising areas. The balance between these two phases is a challenging issue for any optimization algorithm. According to previous studies, no precise rule has been established to distinguish the most appropriate time to transit from exploration to exploitation due to the unexplored form of search spaces and the random nature of this type of optimizer [17, 31]. Therefore, realizing this issue is essential to design a robust and reliable optimization algorithm.

Considering the main challenges of creating a high-performance optimization algorithm and all critical points highlighted in the literature above [42-44], we introduce an efficient optimizer based on the concept of the weighted mean of vectors. By avoiding a basis of nature inspiration, INFO offers a promising method to avoid and reduce the challenges of other optimization algorithms, thus providing a strong step in the direction towards a metaphor-free class of optimization algorithms.

3. Definition of weighted mean

The optimization algorithm introduced in this study is based on a weighted mean, which demonstrates a unique location in an object or system [45]. A detailed definition of this concept is subsequently provided.

3.1. Mathematical definition of weighted mean

The mean of a set of vectors is described as the average of their positions (x_i), as weighted by the fitness of a vector (w_i) [45]. In fact, this concept is used due to its simplicity and ease of implementation. Fig. 1 depicts the weighted mean of the set of solutions (vectors), in which the solutions with bigger weights are more effective in calculating the weighted mean of solutions.

The formulation of weighted mean (WM) is defined by Eq. (1) [45]:

$$WM = \frac{\sum_{i=1}^{N} x_i \times w_i}{\sum_{i=1}^{N} w_i} \tag{1}$$

where N is the number of vectors.

To provide a better explanation, WM can be considered as two vectors, as shown in Eq. (1.1) [45]:

$$WM = \frac{x_1 \times w_1 + x_2 \times w_2}{w_1 + w_2} \tag{1.1}$$

In this study, each vector's weight was calculated based on a wavelet function (WF) [46, 47]. Generally, the wavelet is a useful tool for modeling seismic signals by compounding translations and dilations of an oscillatory function (i.e., mother wavelet) with a finite period. This function is employed to create effective fluctuations during the optimization process. Fig. 2 displays the mother wavelet used in this study, which is defined as:

$$w = \cos(x) \times \exp\left(-\frac{x^2}{\omega}\right) \tag{2}$$

where ω is a constant number called the dilation parameter.

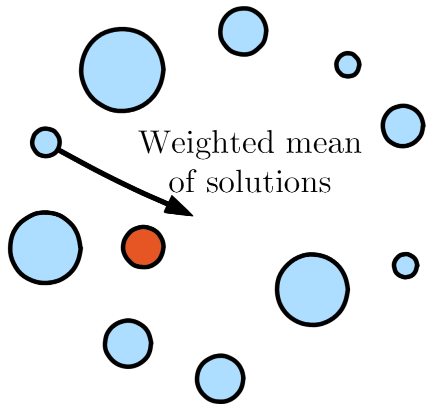


Fig. 1. The weighted mean of a set of solutions

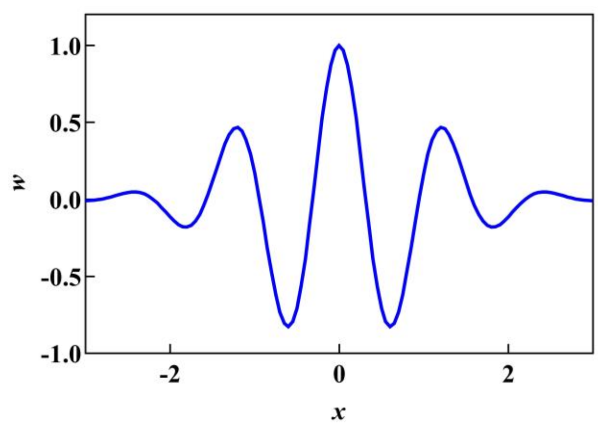


Fig. 2. Mother wavelet

Figs. 3a and 3b display three vectors, and the differential between them are shown in Fig 3c. The weighted mean of vectors is calculated by Eq. (3):

$$WM = \frac{w_1 \times (x_1 - x_2) + w_2 \times (x_1 - x_3) + w_3 \times (x_2 - x_3)}{w_1 + w_2 + w_3} \tag{3}$$

in which

$$w_1 = \cos((f(x_1) - f(x_2)) + \pi) \times \exp(\frac{f(x_1) - f(x_2)}{\omega}) \tag{3.1}$$

$$w_2 = \cos((f(x_1) - f(x_3)) + \pi) \times \exp(\frac{f(x_1) - f(x_3)}{\omega}) \tag{3.2}$$

$$w_3 = \cos((f(x_2) - f(x_3)) + \pi) \times \exp(\frac{f(x_2) - f(x_3)}{\omega}) \tag{3.3}$$

where $f(x)$ denotes the fitness function of the vector x.

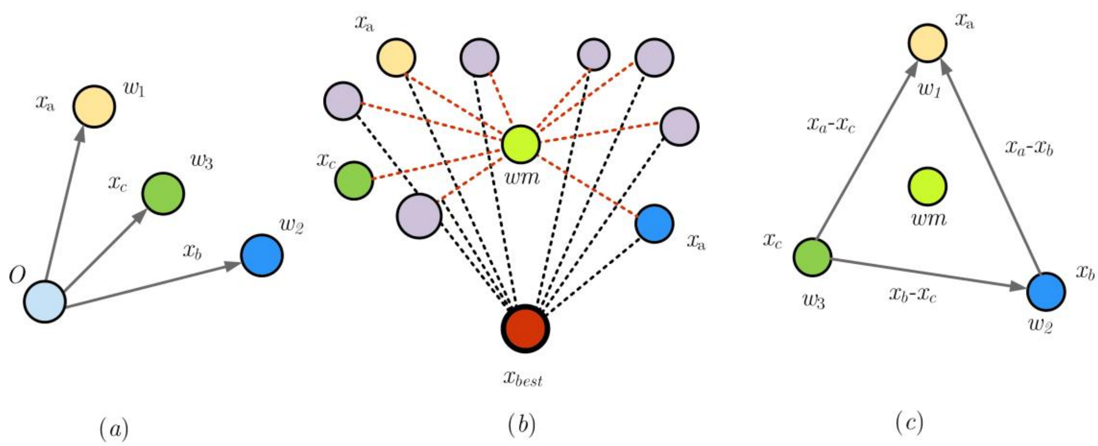


Fig. 3. The weighted mean of vectors for three vectors

4. Weighted mean of vectors (INFO) algorithm

The weighted mean of vectors algorithm (INFO) is a population-based optimization algorithm that calculates the weighted mean for a set of vectors in the search space. In the proposed algorithm, the population is comprised of a set of vectors that demonstrate possible solutions. The INFO algorithm finds the optimal solution over several successive generations.

The three operators update the vectors' positions in each generation:

- Stage 1: Updating rule
- Stage 2: Vector combining
- Stage 3: Local search

Herein, the problem of *minimizing* the objective function is considered as an example.

4.1. Initialization stage

The INFO algorithm is comprised of a population of *Np* vector in *D* dimensional search domain ($X_{l,j}^g = \{x_{l,1}^g, x_{l,2}^g, ..., x_{l,D}^g\}, l = 1, 2, ..., Np$). In this step, some control parameters are introduced and defined for the INFO algorithm. There are two main parameters: weighted mean factor δ and scaling factor σ.

Generally speaking, the scaling rate is used to amplify the obtained vector via the updating rule operator, which is dependent on the size of the search domain. The σ factor is used to scale the weighted mean of vectors. Its value is specified based on the feasible search space of problems and reduced according to an exponential formula. These two parameters do not need to be adjusted by the user and change dynamically based on generation. The INFO algorithm uses a simple method to generate the initial vectors called random generation.

4.2. Updating rule stage

In the INFO algorithm, the updating rule operator increases the population's diversity during the search procedure. This operator uses the weighted mean of vectors in order to create new vectors. Indeed, this operator distinguishes the INFO algorithm from other algorithms and consists of two main parts. In the first part, a mean-based rule is extracted from the weighted mean for a set of random vectors. The mean-based method begins from a random initial solution and moves to the next solution using the weighted mean information of a set of randomly selected vectors. The second part is convergence acceleration, which enhances convergence speed and promotes the algorithm's performance to reach optimal solutions.

In general, INFO first employs a set of selected randomly differential vectors to obtain the weighted mean of vectors rather than move the current vector toward a better solution. In this work, increasing the population's diversity is considered the *MeanRule* based on the best, better, and worst solutions. It should be noted that the better solution is randomly determined from the top 5 solutions (regarding the objective function value). Therefore, the mean-based rule is conducted to the *MeanRule*, as defined in Eq. (4):

$$MeanRule = r \times WM1_l^g + (1-r) \times WM2_l^g \qquad l = 1,2,\ldots,Np \tag{4}$$

$$WM1_l^g = \delta \times \frac{w_1(x_{a1}-x_{a2}) + w_2(x_{a1}-x_{a3}) + w_3(x_{a2}-x_{a3})}{w_1+w_2+w_3+\varepsilon} + \varepsilon \times rand, \qquad l = 1,2,\ldots,Np \tag{4.1}$$

where

$$w_1 = \cos((f(x_{a1}) - f(x_{a2})) + \pi) \times \exp\left(-\frac{f(x_{a1}) - f(x_{a2})}{\omega}\right) \tag{4.2}$$

$$w_2 = \cos((f(x_{a1}) - f(x_{a3})) + \pi) \times \exp\left(-\frac{f(x_{a1}) - f(x_{a3})}{\omega}\right) \tag{4.3}$$

$$w_3 = \cos((f(x_{a2}) - f(x_{a3})) + \pi) \times \exp\left(-\frac{f(x_{a2}) - f(x_{a3})}{\omega}\right) \tag{4.4}$$

$$\omega = \max(f(x_{a1}), f(x_{a2}), f(x_{a3})) \tag{4.5}$$

$$WM2_l^g = \delta \times \frac{w_1(x_{bs}-x_{bt}) + w_2(x_{bs}-x_{ws}) + w_3(x_{bt}-x_{ws})}{w_1+w_2+w_3+\varepsilon} + \varepsilon \times rand, \qquad l = 1,2,\ldots,Np \tag{4.6}$$

where

$$w_1 = \cos((f(x_{bs}) - f(x_{bt})) + \pi) \times \exp\left(-\frac{f(x_{bs}) - f(x_{bt})}{\omega}\right) \tag{4.7}$$

$$w_2 = \cos((f(x_{bs}) - f(x_{ws})) + \pi) \times \exp\left(-\frac{f(x_{bs}) - f(x_{ws})}{\omega}\right) \tag{4.8}$$

$$w_3 = \cos((f(x_{bt}) - f(x_{ws})) + \pi) \times \exp(-\frac{f(x_{bt}) - f(x_{ws})}{\omega}) \tag{4.9}$$

$$\omega = f(x_{ws}) \tag{4.10}$$

where $f(x)$ is the value of the objective function; $a1 \neq a2 \neq a3 \neq l$ are different integers randomly selected from the range [1, *NP*]; ε is a constant number and has a very small value; *randn* is a normally distributed random value; x_{bs}, x_{bt}, and x_{ws} are the best, better, and worst solutions among all vectors in the population for the g^{th} generation, respectively. In fact, these solutions are determined after sorting the solution at each iteration. *r* is a random number within the range [0, 0.5]; and w_1, w_2, and w_3 are three WFs to calculate the weighted mean of vectors that help the proposed INFO algorithm to search in the solution space globally.

In fact, the WFs are used to vary the *MeanRule* space according to the wavelet theory, which is considered for two reasons: (1) to assist the algorithm to explore the search space more effectively and achieve better solutions by creating efficient oscillation during the optimization procedure; and (2) to generate fine-tuning by controlling the dilation parameter introduced in the WFs, which is used to adjust the amplitude of WF. In this study, the value of the dilation parameter was varied using Eq. (4.10) during the optimization process. In Eq. (5), δ is the scale factor, and β can be changed based on an exponential function defined in (5.1):

$$\delta = 2\beta \times rand - \beta \tag{5}$$

$$\beta = 2\exp(-4 \times \frac{g}{Maxg}) \tag{5.1}$$

where *Maxg* is the maximum number of generations.

The convergence acceleration part (*CA*) is also added to the updating rule operator to promote global search ability, using the best vector to move the current vector in a search space. In the INFO algorithm, it is supposed that the best solution is the nearest solution to global optima. In fact, *CA* helps vectors move in a better direction. The *CA* presented in Eq. (6) is multiplied by a random number in the range [0,1] ($rand$) to ensure that each vector has a different step size in each generation in INFO:

$$CA = randn \times \frac{(x_{bs} - x_{a1})}{(f(x_{bs}) - f(x_{a1}) + \varepsilon)} \tag{6}$$

where $randn$ is a random number with a normal distribution.

Finally, the new vector is calculated using Equation (7):

$$z_l^g = x_l^g + \sigma \times MeanRule + CA \tag{7}$$

An optimization algorithm should generally search globally to discover the search domain's promising spaces (exploration phase). Accordingly, the proposed updating rule based on x_{bs}, x_{bt}, x_l^g and x_{a1}^g is defined using the following scheme:

$$if rand < 0.5$$

$$
\begin{aligned}
&\qquad z1_l^g = x_l^g + \sigma \times MeanRule + randn \times \frac{(x_{bs} - x_{a1}^g)}{(f(x_{bs}) - f(x_{a1}^g) + 1)} \\
&\qquad z2_l^g = x_{bs} + \sigma \times MeanRule + randn \times \frac{(x_{a1}^g - x_{a2}^g)}{(f(x_{a1}^g) - f(x_{a2}^g) + 1)} \\
&\quad else \\
&\qquad z1_l^g = x_a^g + \sigma \times MeanRule + randn \times \frac{(x_{a2}^g - x_{a3}^g)}{(f(x_{a2}^g) - f(x_{a3}^g) + 1)} \\
&\qquad z2_i^g = x_{bt} + \sigma \times MeanRule + randn \times \frac{(x_{a1}^g - x_{a2}^g)}{(f(x_{a1}^g) - f(x_{a2}^g) + 1)} \\
&end
\end{aligned}
\tag{8}
$$

where $z1_l^g$ and $z2_l^g$ are the new vectors in the g^{th} generation; and σ is the scaling rate of a vector, as defined in Eq. (9). It should be noted that in Eq. (9), α can be changed based on an exponential function defined in Eq. (9.1):

$$\sigma = 2\alpha \times rand - \alpha \tag{9}$$

$$\alpha = c\exp(-d \times \frac{g}{Maxg}) \tag{9.1}$$

where c and d are constant numbers equal to 2 and 4, respectively. It is worth noting that for large values of the parameter σ , the current position tends to diverge from the weighted mean of vectors (exploration search), while small values of this parameter force the current position to move toward the weighted mean of vectors (exploitation search).

4.3. Vector combining stage

In this study, for enhancing the population's diversity in INFO, the two vectors calculated in the previous section ($z1_l^g$ and $z2_l^g$) are combined with vector x_l^g regarding the condition $rand < 0.5$ to generate the new vector u_l^g , according to Eq. (10). In fact, this operator is used to promote the local search ability to provide a new and promising vector:

$$
\begin{aligned}
&\boldsymbol{if}\ rand < 0.5 \\
&\quad \boldsymbol{if}\ rand < 0.5 \\
&\qquad u_l^g = z1_l^g + \mu.|z1_l^g - z2_l^g| && (10.1) \\
&\quad \boldsymbol{else} \\
&\qquad u_l^g = z2_l^g + \mu.|z1_l^g - z2_l^g| && (10.2) \\
&\quad \boldsymbol{end} \\
&\boldsymbol{else} \\
&\quad u_l^g = x_l^g && (10.3) \\
&\boldsymbol{end}
\end{aligned}
$$

where u_l^g is the obtained vector using the vector combining in gth generation; and μ is equal to $0.05 \times randn$.

4.4. Local search stage

Effective local search ability can prevent INFO from deception and dropping into locally optimal solutions. The local operator is considered using the global position (x_{best}^{g}) and the mean-based rule defined in Eq. (11) to further promote the exploitation, search, and convergence to global optima. According to this operator, a novel vector can be produced around x_{best}^{g} , if $r < 0.5$, where *rand* is a random value in [0, 1]:

$$\begin{aligned}
&if \;\; rand < 0.5 \\
&\quad if \;\; rand < 0.5 \\
&\qquad u_l^g = x_{bs} + randn \times (MeanRule + randn \times (x_{bs}^g - x_{a1}^g)) && (11.1) \\
&\quad else \\
&\qquad u_l^g = x_{rnd} + randn \times (MeanRule + randn \times (\upsilon_1 \times x_{bs} - \upsilon_2 \times x_{rnd})) && (11.2) \\
&\quad end \\
&end
\end{aligned}$$

in which

$$x_{rnd} = \phi \times x_{avg} + (1-\phi) \times (\phi \times x_{bt} + (1-\phi) \times x_{bs}) \tag{11.3}$$

$$x_{avg} = \frac{(x_a + x_b + x_3)}{3} \tag{11.4}$$

where ϕ denotes a random number in the range of (0, 1); and x_{rnd} is a new solution, which combines the components of the solutions, x_{avg} , x_{bt} , and x_{bs} , randomly. This increases the randomness nature of the proposed algorithm to better search in the solution space. υ_1 and υ_2 are two random numbers defined as:

$$\upsilon_1 = \begin{cases} 2 \times rand & if \;\; p > 0.5 \\ 1 & otherwise \end{cases} \tag{11.5}$$

$$\upsilon_2 = \begin{cases} rand & if \;\; p < 0.5 \\ 1 & otherwise \end{cases} \tag{11.6}$$

where p denotes a random number in the range of (0, 1). The random numbers υ_1 and υ_2 can increase the impact of the best position on the vector. Finally, the proposed INFO algorithm is presented in Algorithm 1, and Fig. 4 depicts the flowchart of the proposed algorithm.

The calculation complexity of an optimization algorithm is used to assess the run-time, which is determined based on the algorithm's structure. INFO's computational complexity depends on the number of vectors, the total number of iterations and the number of objects and is calculated as follows:

$$O(CMV) = O(T \times (N \times d)) = O(TNd) \tag{12}$$

where N is the number of vectors (population size), T is maximum generations, and d is the number of objects.

Algorithm 1. Pseudo-code of the INFO algorithm.

STEP 1. Initialization
Set parameters *Np* and *Maxg*
Produce an initial population $P^0 = \{X_i^0, ..., X_{Np}^0\}$
Calculate the objective function value of each vector $f(X_i^0), \quad i = 1, ..., Np$
Determine the best vector x_{bs}
STEP 2. for $g = 1$ to *Maxg* **do**
for $i = 1$ to *Np* **do**
Select randomly $a \neq b \neq c \neq i$ within the range [1, *Np*]
► Updating rule stage
Calculate the vectors $z1_i^g$ and $z2_i^g$ using Eq. (8)
► Vector combining stage
Calculate the vector u_i^g using Eq. (10)
► Local search stage
Calculate the local search operator using Eq. (11)
Calculate the objective function value $f(u_{i,j}^g)$
if $f(u_{i,j}^g) < f(x_{i,j}^g)$ **then** $x_{i,j}^{g+1} = u_{i,j}^g$
Otherwise $x_{i,j}^{g+1} = x_{i,j}^g$
end for
Update the best vector (x_{bs})
end for
STEP 3. Return Vector $x_{best,j}^g$ as the final solution

To demonstrate the potential of INFO to solve optimization problems, its capabilities are described below:

- INFO generates and promotes a set of random vectors for a problem and inherently has a high ability to explore and escape local optimal solutions to single-solution-based algorithms.
- The proposed updating rule in the INFO mechanism uses the mean rule and convergence acceleration part to find the search space's up-and-coming areas.
- The proposed vector combining operator can explore the search space to improve the search capability and local optima avoidance.
- Adaptive parameters smoothly implement the transition from exploration to exploitation.
- A complement strategy called a local search operator is used to promote the exploitation and convergence speed further.

- A variable is used as the global best position to record an appropriate approximation of the global optimum and promote it during optimization.
- Since the vectors can change their positions according to the best position generated so far, this will tend toward the best regions of the search spaces during the optimization.

The next sections verify the performance of INFO in several test functions and real engineering problems.

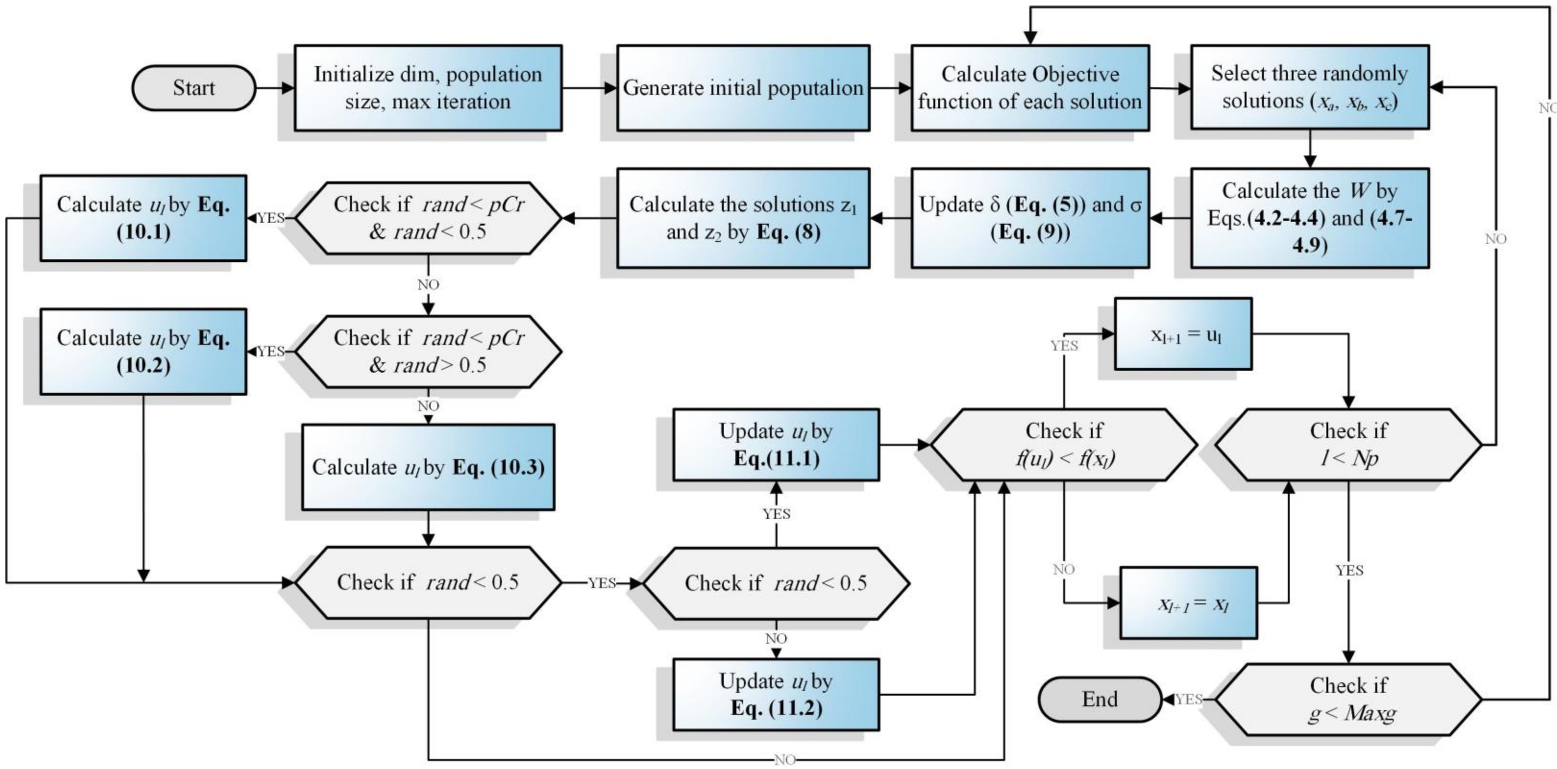


Fig. 4. Flowchart of INFO

5. Results and discussion

To evaluate and confirm the efficiency of an optimization algorithm, several test problems should be considered. Therefore, this work tested the INFO algorithm's performance on 19 mathematical benchmark functions, 13 of which (f_1-f_7 and f_8-f_{13}) have been widely utilized in previous studies [38, 39, 48] and have unimodal (UF) and multimodal (MF) search spaces, respectively. Functions f_{14}- f_{19} are composite functions that have also been considered in several previous studies [17, 31]. In the challenging composite functions (CFs), the global solution position is shifted to a random position, and the functions are rotated, which occasionally places the global solution within infeasible space boundaries occasionally and combines variants of the benchmark functions.

A detailed explanation of these functions is reported in Tables 2-4. INFO was compared with the GWO, GSA, SCA, GA, PSO, and BA optimization algorithms. We followed fair comparison standards. For all the optimizers, the population size and the total number of iterations were set to 30 and 500. The values of the main parameters for all algorithms are given in Table 1. It is pertinent to mention that all of the control parameters were set based on their developer or within the range of the suggestions to achieve the best efficiency of the optimizers. The parameter settings of GWO and SCA were obtained from previous works [49]. The benchmark functions for each optimization algorithm were tested 30 times. Table 5 presents the average and standard deviation of the fitness functions for the 30 runs.

Table 1. Values of control parameters for all comparative algorithms

Algorithms	Values of parameters
GWO	Convergence constant a = [2, 0]
BA	*A* (loudness) = 0.5, *r* (plus rate) = 0.5, f_{min} = 0, f_{max} = 2
GA	Crossover probability =0.8 , mutation probability = 0.05
PSO	$c_1 = c_2 = 1.5$, *w* (inertial weight) linearly decreased from 0.7 to 0.3
GSA	G_0 (initial gravitational constant) = 100, α = 20
SCA	A = 2
INFO	c = 2, d = 4

Table 2. UF test problems

Function	Dimension	Range	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	30	[-100,10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$f_4(x) = max_i\{\|x_i\|, 1 \leq i \leq n\}$	30	[-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1})^2 + (x_i - 1)^2]$	30	[-30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100,100]	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-1.28,1.28]	0

Table 3. MF test problems

Function	Dimension	Range	f_{min}
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500,500]	-418.9829×5
$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	[-5.12,5.12]	0
$f_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-32,32]	0
$f_{11}(x) = \frac{1}{400}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i,a,k,m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$f_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i,5,100,4)$	30	[-50,50]	0

Table 4. CF test problems

Function	Dimension	Range	f_{min}
$f_{14}(CF1)$: $f_1, f_2, f_3, \ldots, f_{10}$ = Sphere Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[5/100, 5/100, 5/100,\ldots, 5/100]$	30	[-5,5]	0
$f_{15}(CF2)$: $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[5/100, 5/100, 5/100,\ldots, 5/100]$	30	[-5,5]	0
$f_{16}(CF3)$: $f_1, f_2, f_3, \ldots, f_{10}$ = Griewank's Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[1,1,1,\ldots,1]$	30	[-5,5]	0
$f_{17}(x)$: f_1, f_2 = Ackley's Function f_3, f_4 = Rasrigin's Function f_5, f_6 = Sphere's Function f_7, f_8 = Weierstras's Function f_9, f_{10} = Griewank's Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,2,1.5,1.5,1,1,1.5,1.5,2,2]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[2*5/32, 5/32, 2*1,1, 2*5/100, 5/100, 2*10,10, 2*5/60, 5/60,]$	30	[-5,5]	0
$f_{18}(x)$: f_1, f_2 = Rasrigin's Function f_3, f_4 = Weierstras's Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley's Function f_9, f_{10} = Sphere's Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[1,1,10,10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]$	30	[-5,5]	0
$f_{19}(x)$: f_1, f_2 = Rasrigin's Function f_3, f_4 = Weierstras's Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley's Function f_9, f_{10} = Sphere's Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[1,1,10,10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]$	30	[-5,5]	0

Table 5. Statistical results and comparison for test functions

Function		INFO	GWO	GSA	SCA	PSO	BA	GA
f_1	Mean	**2.59E-43**	2.02E-27	2.49E-01	1.49E+00	2.43E-16	3.92E+00	1.74E-01
	SD	1.04E-43	4.27E-27	2.19E-01	8.94E-01	1.09E-16	6.08E+00	2.66E-02
f_2	Mean	**3.23E-21**	8.03E-17	8.36E-01	3.30E-01	1.26E-07	1.37E-02	1.68E-01
	SD	2.29E-21	5.53E-17	2.56E-01	8.34E-02	1.90E-07	2.62E-02	6.61E-02
f_3	Mean	**6.46E-39**	1.72E-05	1.22E+02	1.55E+03	8.55E+02	9.21E+03	3.50E-01
	SD	2.98E-38	6.95E-05	4.21E+01	1.01E+03	2.15E+02	5.30E+03	1.76E-01
f_4	Mean	**8.28E-22**	8.51E-07	1.51E+00	1.02E+01	7.10E+00	3.71E+01	3.57E-01
	SD	4.49E-22	1.33E-06	2.22E-01	2.71E+00	2.54E+00	1.13E+01	2.81E-01
f_5	Mean	**2.47E+01**	2.72E+01	2.62E+02	2.78E+02	4.01E+01	1.74E+04	2.78E+01
	SD	7.45E-01	8.02E-01	1.51E+02	1.45E+02	2.06E+01	2.83E+04	2.33E+00
f_6	Mean	**1.54E-06**	6.75E-01	2.53E-01	1.89E+00	3.73E+00	2.19E+01	1.60E-02
	SD	3.93E-06	3.27E-01	1.79E-01	7.01E-01	4.08E+00	3.03E+01	8.51E-02
f_7	Mean	**1.62E-03**	1.79E-03	1.56E+00	1.15E-01	8.73E-02	1.43E-01	8.55E-02
	SD	1.34E-03	6.61E-04	1.12E+00	3.86E-02	3.45E-02	2.29E-01	1.95E-02
f_8	Mean	**-9.47E+03**	-6.16E+03	-6.23E+03	-6.63E+03	-2.76E+03	-3.70E+03	-5.61E+03
	SD	6.40E+02	8.55E+02	1.07E+03	6.69E+02	5.19E+02	2.73E+02	6.76E+02
f_9	Mean	**0.00E+00**	3.02E+00	1.02E+02	9.27E+00	2.49E+01	3.60E+01	3.26E+00
	SD	0.00E+00	4.52E+00	2.24E+01	3.17E+00	4.67E+00	4.00E+01	4.51E+00
f_{10}	Mean	**8.88E-16**	1.03E-13	1.01E+00	9.59E-01	1.11E-08	1.65E+01	5.70E+00
	SD	0.00E+00	1.82E-14	5.90E-01	5.92E-01	2.64E-09	7.10E+00	3.10E+00
f_{11}	Mean	**0.00E+00**	2.97E-03	2.06E-02	9.71E-01	2.78E+01	1.03E+00	3.91E-01
	SD	0.00E+00	6.45E-03	8.48E-03	7.43E-02	7.07E+00	3.91E-01	6.41E-01
f_{12}	Mean	**1.04E-02**	5.93E-02	1.96E-02	1.64E+00	2.28E+00	8.65E+00	6.98E-01
	SD	3.16E-02	9.16E-02	3.96E-02	1.63E+00	1.05E+00	7.77E+00	9.24E-01
f_{13}	Mean	**4.30E-02**	6.59E-01	7.31E-02	6.81E+00	8.94E+00	5.57E+05	8.77E+00
	SD	7.36E-02	3.18E-01	4.94E-02	7.71E+00	6.51E+00	1.92E+06	8.05E+00
f_{14}	Mean	**1.11E-04**	6.42E+01	4.33E+01	5.56E-01	3.51E-02	2.29E+02	3.18E+01
	SD	9.52E-05	7.66E+01	8.17E+01	1.23E-01	1.91E-01	7.53E+01	7.47E+01
f_{15}	Mean	**5.94E+01**	2.25E+02	1.54E+02	2.58E+02	3.04E+02	2.98E+02	3.15E+02
	SD	6.31E+01	1.27E+02	1.31E+02	1.45E+02	1.42E+02	8.53E+01	1.59E+02
f_{16}	Mean	**2.82E+01**	5.27E+02	8.77E+01	2.18E+02	1.77E+02	1.68E+03	1.27E+02
	SD	4.63E+01	2.92E+02	7.71E+01	1.19E+02	1.49E+02	5.78E+01	9.41E+01
f_{17}	Mean	9.08E+02	9.52E+02	8.32E+02	9.48E+02	1.02E+03	**5.37E+02**	1.08E+03
	SD	5.10E+00	2.70E+01	2.20E+01	3.88E+01	2.74E+01	7.75E+01	1.61E+02
f_{18}	Mean	**2.48E+02**	3.04E+02	3.48E+02	4.28E+02	2.52E+02	5.35E+02	4.31E+02
	SD	8.96E+01	1.41E+02	1.47E+02	7.85E+01	1.94E+02	8.02E+01	1.07E+02
f_{19}	Mean	**2.78E+02**	4.69E+02	3.83E+02	8.27E+02	3.56E+02	7.04E+02	3.73E+02
	SD	8.05E+01	1.04E+02	6.27E+01	1.03E+02	1.08E+02	1.49E+02	1.18E+02

5.1. Assessment of the exploitative behavior

In this section, the exploitation ability of INFO is investigated using the UFs. Functions f_1- f_7 are unimodal and have one global solution. Table 5 reveals that INFO is very promising and competitive with the comparative algorithms. Specifically, it was the best method to optimize all functions in terms of the average objective function values for 30 runs. The proposed algorithm can outperform the others on all functions according to standard deviation values, except for GWO on function f_5. Therefore, INFO can afford appropriate exploitation search ability due to the embedded exploitation phase.

5.2. Assessment of the exploratory behavior

To inspect the exploration search capability of INFO in the study, MFs (f_8-f_{13}) were used. These functions have many local optima solutions whose number rises exponentially with the dimension of the problems and, thus, are suitable to verify optimizers' exploration search ability. Table 5 presents the results of INFO and other optimization methods, revealing that the proposed algorithm for MFs presents a very suitable exploration search ability. Specifically, INFO outperformed all other algorithms in terms of the average of the objective function and standard deviation values found for all the functions, except on the standard deviations of functions f_8 and f_{13}. The presented results demonstrate that the INFO algorithm has an excellent competency in exploration search.

5.3. Assessment of ability escaping from local optimum

In this section, the ability of INFO on CFs (f_{14}-f_{19}) is examined. These functions are very challenging for optimizers because only an appropriate balance between exploitation and exploration can escape local optimum solutions. Table 5 displays the results of all examined algorithms on the CFs. It is evident that INFO achieved favorable results regarding the average of objective function values for all functions and lower standard deviations for functions f_{14}-f_{17} compared to the other algorithms but failed on functions f_{18} and f_{19}. This means that the proposed algorithm has a suitable balance between exploitation and exploration, preventing it from getting stuck in high local optimum solutions. This efficient performance is due to the obtained vectors by using the updating rule and vector combining operators. The updating rule provides two vectors to improve the local (exploitation) and global (exploration) search in the search space (Eq. 8). Then, the vector combining operator combines them with the current vector with a certain probability. This process helps the INFO algorithm explore the search space on both the global and local scales with a suitable balance. Also, the local operator makes the optimization process safe from trapping in local optima positions.

By employing the Friedman test [50], it was found that the INFO algorithm achieved a top rank, followed by GWO and GSA, as seen in Table 6. This further verifies that INFO's performance is better than the other well-known optimizers.

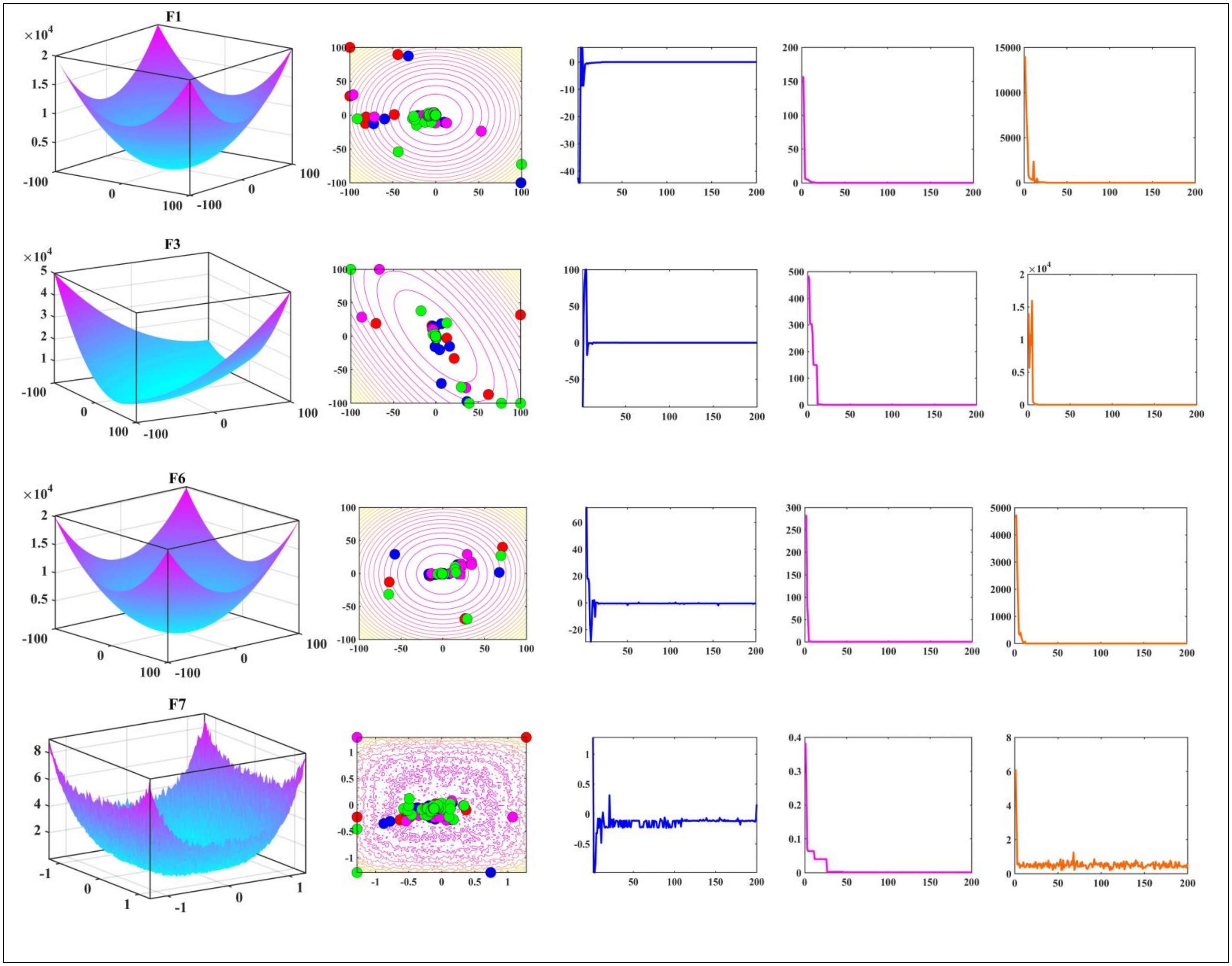
Table 6. Mean rankings computed by Friedman test for test functions

Function	INFO	GWO	GSA	SCA	PSO	BA	GA
f_1	1	2	3	7	5	4	6
f_2	1	2	3	4	7	5	6
f_3	1	2	5	7	4	3	6
f_4	1	2	5	7	4	3	6
f_5	1	2	4	7	5	3	6
f_6	1	4	6	7	3	2	5
f_7	1	2	4	6	7	3	5
f_8	1	4	7	6	3	5	2
f_9	1	2	5	6	7	3	4
f_{10}	1	2	3	7	5	6	4
f_{11}	1	2	7	6	3	4	5
f_{12}	1	3	6	7	2	4	5
f_{13}	1	3	6	7	2	5	4
f_{14}	1	6	2	7	5	4	3
f_{15}	1	3	6	5	2	7	4
f_{16}	1	6	4	7	2	3	5
f_{17}	3	5	6	1	2	7	4
f_{18}	1	3	2	7	4	6	5
f_{19}	1	5	4	7	2	6	3
Mean Rank	**1.11**	**3.16**	**4.63**	**6.21**	**3.89**	**4.37**	**4.63**
Final Rank	**1**	**2**	**3**	**5**	**4**	**6**	**4**

5.4. Investigation of convergence speed

In this paper, three metrics, including search history, trajectory curve, and convergence rate, were considered to assess the INFO algorithm's convergence behavior. Accordingly, 8 different benchmark functions, i.e. f_1, f_3, f_6, f_7, f_9, f_{10}, f_{11}, and f_{13}, each with a dimension of 2, were chosen. To solve these test functions, INFO used five solutions over 200 iterations.

The search history and trajectory curves of the five solutions in their first dimension are depicted in Fig. 5. Generally, a low-density distribution illustrates the exploration, and a high-density distribution represents the exploitation. According to this figure, the solutions' distribution density demonstrates how INFO can search globally and locally in the solution space, where the solutions have a high density in the region close to the global optima and have a low density in the regions far from the global optima. Therefore, it can be concluded that INFO can successfully explore promising regions in the solution space to explore the best position.


F1
F3
F6
F7

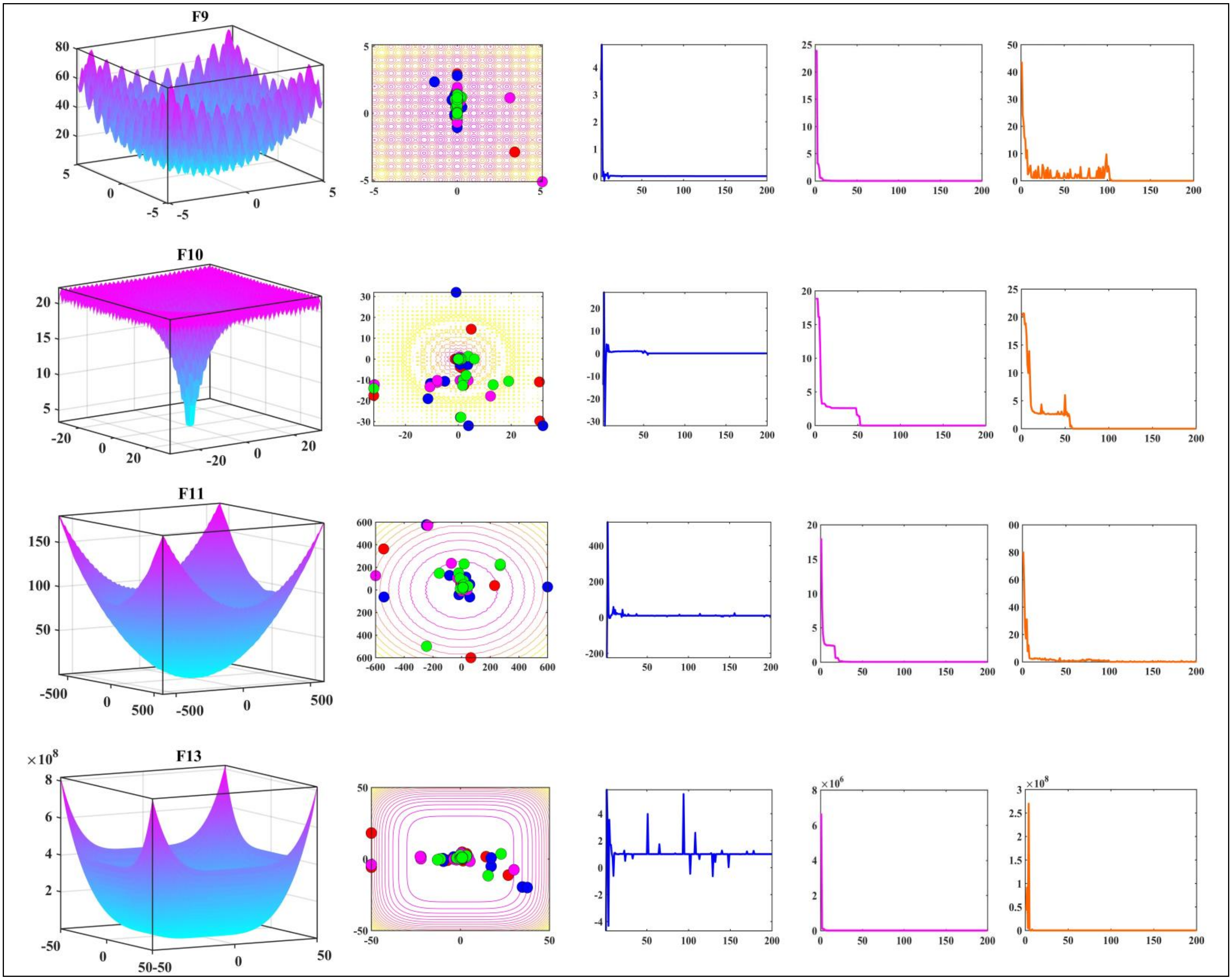


Fig. 5. Search history, trajectory, average fitness, and convergence metrics

In the average fitness graph in Fig. 5, the varied history of the fitness of solutions in INFO during the optimization process can be seen, where the average fitness curve suddenly decreased in the early iterations. This behavior confirms the superior convergence speed and accurate search capability of INFO.

The trajectory curves represent the global and local search behaviors of each optimizer. Fig. 5 displays the trajectory graphs of five solutions for the first dimension, revealing the high variation of curves in the initial generation. As the number of iterations increased, the variation of the curves decreased. Since high and low variations indicate the exploration and exploitation, respectively, it can be deduced that the INFO algorithm first performs the exploration and then the exploitation.

The most crucial goal of each optimization algorithm is to obtain global optima, which can be achieved by obtaining the convergence curves to visualize the algorithms' behaviors. According to Fig. 5, the convergence variations of functions f_1, f_6, f_9, f_{11}, and f_{13} dropped rapidly in the early iterations, demonstrating that INFO implemented the exploration search more effectively than the exploration. In opposition, the convergence variations of functions f_3, f_7, and f_{10} decreased relatively slow, indicating the better efficiency of INFO in the exploration search than the exploitation.

Different variations of convergence graphs for the optimizers are displayed in Fig. 6, which compares the convergence speeds of the INFO algorithms and other optimizers on some of the benchmark functions. Accordingly, INFO can compete on the same level as the other contestant algorithms with a very promising convergence speed.

As first reported by Berg et al. [51], sudden changes in the convergence curve during the early optimization steps effectively reach the best solution. This behavior helps an optimization method to better search the search space, whereby the optimization process should be decreased to support exploitation search in the end stages of the optimization process. According to this viewpoint, during the optimization procedure, the exploitation phase assists in probing the search space, and then the solutions converge to the best solution in the exploration phase.

As can be observed in Fig. 6, INFO presents two convergence modes for optimizing the problems. The first mode is rapid convergence in the initial iterations, whereby the INFO algorithm can reach a more accurate solution than the contestant algorithms. This high convergence rate can be seen in f_1, f_2, f_4, f_{10}, and f_{13}. The second convergence mode tends to increase the convergence rate by increasing the number of iterations. This is owed to the proposed adaptation method in INFO that aids it to explore appropriate areas of the search domain in the primary iterations and improves the convergence speed after almost 100 iterations. Regarding functions f_7, f_{15} to f_{19} as the most complex and challenging benchmark functions, the optimization results of these problems indicate that INFO profits from a suitable balance of exploitation and exploration that helps it achieve the global solution. Consequently, these results demonstrate the efficient performance of INFO to optimize complex problems.

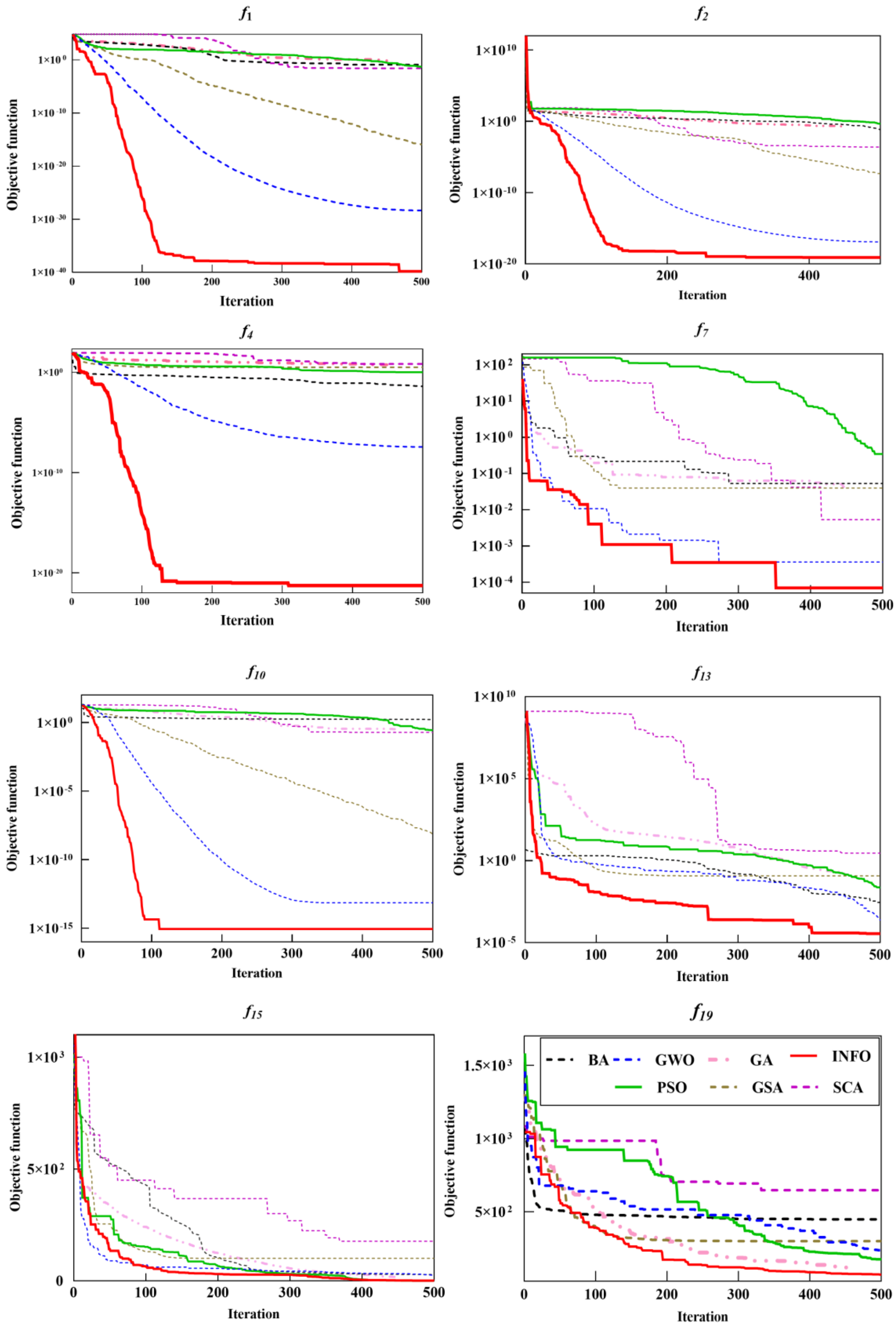


Fig. 6. Comparison of convergence curve variations of INFO and the other optimizers in some of the test functions

5.5. Wall-clock time analysis of INFO

This section investigates the run-time of INFO compared with other methods on 13 benchmark functions. All optimization methods were run ten times on each test function separately, and the results are reported in Table 7. From the table, the INFO optimization process took a relatively long time due to the calculation of its two operators (i.e., vector combining and local search stage). Nevertheless, INFO outperformed some methods, such as GA and GSA. Generally, albeit with a relatively time-consuming run-time, INFO has considerable advantages over the other methods.

Table 7 Comparison of the run-time of INFO and other methods

	INFO	GA	GSA	GWO	PSO	SCA	BA
f_1	4.70	5.84	8.46	1.30	0.64	1.31	1.56
f_2	4.70	5.96	7.18	1.42	0.77	1.14	1.67
f_3	7.91	9.34	10.47	4.72	4.51	2.50	5.42
f_4	4.08	5.04	7.06	1.18	0.52	1.13	1.55
f_5	4.24	5.51	7.23	1.38	0.66	1.41	1.73
f_6	4.07	5.05	7.03	1.20	0.50	1.52	1.39
f_7	4.77	5.90	7.66	1.87	1.18	2.40	2.09
f_8	4.38	5.75	7.28	1.44	0.81	1.80	1.83
f_9	4.33	5.15	7.08	1.38	0.62	1.61	1.82
f_{10}	4.14	5.36	7.17	1.65	0.76	1.78	1.76
f_{11}	4.56	5.75	7.55	1.44	0.88	2.20	2.22
f_{12}	6.50	8.04	9.38	3.37	2.80	4.59	4.28
f_{13}	6.44	8.24	9.07	3.34	2.79	4.86	4.05

5.6. Assessment of INFO on CEC-BC-2017 test functions

To further evaluate the INFO algorithm, widely-used and complicated CEC-BC-2017, benchmark problems were utilized, including rotated and shifted unimodal, multimodal, hybrid, and composite test functions [52]. The characteristics of these problems are available in Appendix A (Table 8). The mathematical formulation of these test functions is also available in the initial IEEE report. The proposed INFO algorithm was assessed against these benchmark functions, and its results were compared to those of other well-known optimizers. For all test functions, the dimension was equal to 10. The optimizers were running 30 times with 1000 iterations for each test function. The control parameters for each optimizer were the same as those considered in Section 5. Table 9 reports the results acquired by INFO and the other algorithms, including each function's average and standard deviation over 30 runs.

According to Table 10, INFO takes the top Friedman mean rank, and the efficiency of the INFO, PSO, and GWO are much better than the other optimizers. This performance illustrates INFO's capability to outperform other well-known optimizers and further indicates that INFO can solve complex optimization problems.

Table 8. Properties of the CEC-BC-2017 test functions [52]

Type	No.	Functions	Global	Domain
Unimodal Function	f_1	Shifted and Rotated Bent Cigar Function	100	[-100,100]
	f_3	Shifted and Rotated Zakharov Function	300	[-100,100]
Multimodal Functions	f_4	Shifted and Rotated Rosenbrock's Function	400	[-100,100]
	f_5	Shifted and Rotated Rastrigin's Function	500	[-100,100]
	f_6	Shifted and Rotated Expanded Scaffer's Function	600	[-100,100]
	f_7	Shifted and Rotated Lunacek Bi_Rastrigin Function	700	[-100,100]
	f_8	Shifted and Rotated Non-Continuous Rastrigin's Function	800	[-100,100]
	f_9	Shifted and Rotated Levy Function	900	[-100,100]
	f_{10}	Shifted and Rotated Schwefel's Function	1000	[-100,100]
Hybrid Functions	f_{11}	Hybrid Function of Zakharov, Rosenbrock and Rastrigin's	1100	[-100,100]
	f_{12}	Hybrid Function of High Conditioned Elliptic, Modified Schwefel and Bent Cigar	1200	[-100,100]
	f_{13}	Hybrid Function of Bent Cigar, Rosenbrock and Lunache Bi-Rastrigin	1300	[-100,100]
	f_{14}	Hybrid Function of Eliptic, Ackley, Schaffer and Rastrigin	1400	[-100,100]
	f_{15}	Hybrid Function of Bent Cigar, HGBat, Rastrigin and Rosenbrock	1500	[-100,100]
	f_{16}	Hybrid Function of Expanded Schaffer, HGBat, Rosenbrock and Modified Schwefel	1600	[-100,100]
	f_{17}	Hybrid Function of Katsuura, Ackley, Expanded Griewank plus Rosenbrock, Modifed Schwefel and Rastrigin	1700	[-100,100]
	f_{18}	Hybrid Function of high conditioned Elliptic, Ackley, Rastrigin, HGBat and Discus	1800	[-100,100]
	f_{19}	Hybrid Function of Bent Cigar, Rastrigin, Expanded Grienwank plus Rosenbrock, Weierstrass and expanded Schaffer	1900	[-100,100]
	f_{20}	Hybrid Function of Happycat, Katsuura, Ackley, Rastrigin, Modified Schwefel and Schaffer	2000	[-100,100]
Composition Functions	f_{21}	Composition Function of Rosenbrock, High Conditioned Elliptic and Rastrigin	2100	[-100,100]
	f_{22}	Composition Function of Rastrigin's, Griewank's and Modified Schwefel's	2200	[-100,100]
	f_{23}	Composition Function of Rosenbrock, Ackley, Modified Schwefel and Rastrigin	2300	[-100,100]
	f_{24}	Composition Function of Ackley, High Conditioned Elliptic, Girewank and Rastrigin	2400	[-100,100]
	f_{25}	Composition Function of Rastrigin, Happycat, Ackley, Discus and Rosenbrock	2500	[-100,100]
	f_{26}	Composition Function of Expanded Scaffer, Modified Schwefel, Griewank, Rosenbrock and Rastrigin	2600	[-100,100]
	f_{27}	Composition Function of HGBat, Rastrigin, Modified Schwefel, Bent-Cigar, High Conditioned Elliptic and Expanded Scaffer	2700	[-100,100]
	f_{28}	Composition Function of Ackley, Griewank, Discus, Rosenbrock, HappyCat, Expanded Scaffer	2800	[-100,100]
	f_{29}	Composition Function of shifted and rotated Rastrigin, Expanded Scaffer and Lunacek Bi-Rastrigin	2900	[-100,100]
	f_{30}	Composition Function of shifted and rotated Rastrigin, Non-Continuous Rastrigin and Levy Function	3000	[-100,100]

Table 9. Statistical results and comparison for CEC-BC- 2017 functions

Function		INFO	GWO	GSA	SCA	PSO	BA	GA
f_1	Mean	**1.00E+02**	3.24E+07	3.89E+02	7.32E+08	1.87E+03	1.24E+09	1.64E+03
	SD	2.39E-05	1.10E+08	4.21E+02	2.91E+08	2.46E+03	1.10E+09	1.30E+03
f_3	Mean	**3.00E+02**	1.69E+03	1.06E+04	1.97E+03	**3.00E+02**	1.29E+04	2.30E+03
	SD	2.04E-09	1.91E+03	1.96E+03	1.26E+03	4.60E-14	1.27E+04	1.31E+03
f_4	Mean	**4.00E+02**	4.14E+02	4.06E+02	4.47E+02	4.05E+02	5.25E+02	4.09E+02
	SD	5.33E-01	1.59E+01	5.58E-01	1.81E+01	1.21E+01	9.19E+01	1.75E+01
f_5	Mean	**5.12E+02**	5.16E+02	5.58E+02	5.52E+02	5.36E+02	5.53E+02	5.40E+02
	SD	6.08E+00	7.58E+00	1.10E+01	5.36E+00	1.34E+01	2.07E+01	1.27E+01
f_6	Mean	**6.00E+02**	6.01E+02	6.23E+02	6.17E+02	6.07E+02	6.41E+02	6.20E+02
	SD	6.65E-03	1.23E+00	8.57E+00	2.91E+00	5.05E+00	1.34E+01	1.06E+01
f_7	Mean	7.24E+02	7.29E+02	**7.15E+02**	7.73E+02	7.24E+02	7.84E+02	7.49E+02
	SD	6.79E+00	8.89E+00	2.58E+00	8.89E+00	6.36E+00	3.14E+01	1.86E+01
f_8	Mean	**8.12E+02**	8.15E+02	8.20E+02	8.38E+02	8.21E+02	8.36E+02	8.21E+02
	SD	5.13E+00	6.98E+00	4.14E+00	7.22E+00	9.81E+00	1.32E+01	1.03E+01
f_9	Mean	9.00E+02	9.08E+02	9.00E+02	9.97E+02	**9.00E+02**	1.65E+03	1.05E+03
	SD	7.77E-01	1.69E+01	0.00E+00	3.04E+01	4.72E-14	4.53E+02	1.02E+02
f_{10}	Mean	**1.64E+03**	1.65E+03	2.80E+03	2.24E+03	1.92E+03	2.28E+03	2.03E+03
	SD	2.40E+02	2.43E+02	3.38E+02	2.27E+02	2.21E+02	3.13E+02	3.19E+02
f_{11}	Mean	**1.11E+03**	1.12E+03	1.14E+03	1.20E+03	1.14E+03	1.83E+03	1.15E+03
	SD	8.72E+00	1.43E+01	1.16E+01	4.33E+01	1.73E+01	6.76E+02	3.87E+01
f_{12}	Mean	**2.78E+03**	7.09E+05	7.61E+05	1.70E+07	1.57E+04	1.36E+06	9.90E+05
	SD	1.80E+03	8.27E+05	4.90E+05	1.24E+07	1.08E+04	1.99E+06	1.06E+06
f_{13}	Mean	**1.44E+03**	1.14E+04	1.14E+04	2.51E+04	9.80E+03	1.79E+04	9.37E+03
	SD	1.23E+02	8.43E+03	2.36E+03	2.15E+04	7.18E+03	1.63E+04	5.33E+03
f_{14}	Mean	**1.43E+03**	3.03E+03	6.78E+03	1.70E+03	1.73E+03	2.46E+03	3.81E+03
	SD	1.02E+01	1.79E+03	2.01E+03	6.61E+02	4.41E+02	1.22E+03	2.19E+03
f_{15}	Mean	**1.52E+03**	3.68E+03	2.00E+04	2.25E+03	2.35E+03	4.03E+04	3.68E+03
	SD	1.72E+01	2.54E+03	4.62E+03	5.90E+02	1.42E+03	5.84E+04	2.09E+03
f_{16}	Mean	**1.65E+03**	1.76E+03	2.16E+03	1.73E+03	1.85E+03	2.02E+03	1.93E+03
	SD	5.73E+01	1.31E+02	1.19E+02	4.71E+01	7.58E+01	1.70E+02	1.10E+02
f_{17}	Mean	**1.72E+03**	1.76E+03	1.84E+03	1.78E+03	1.77E+03	1.92E+03	1.76E+03
	SD	1.50E+01	3.59E+01	1.22E+02	2.00E+01	3.03E+01	1.36E+02	2.95E+01
f_{18}	Mean	**1.86E+03**	2.51E+04	8.37E+03	1.07E+05	9.60E+03	1.83E+04	1.21E+04
	SD	4.13E+01	1.31E+04	4.13E+03	7.17E+04	8.67E+03	1.66E+04	1.11E+04
f_{19}	Mean	**1.91E+03**	1.50E+04	4.28E+04	6.75E+03	3.00E+03	2.35E+04	7.47E+03
	SD	7.06E+00	4.70E+04	1.92E+04	5.98E+03	1.77E+03	2.97E+04	4.58E+03
f_{20}	Mean	**2.01E+03**	2.06E+03	2.29E+03	2.10E+03	2.08E+03	2.23E+03	2.14E+03
	SD	1.22E+01	4.99E+01	9.60E+01	3.12E+01	4.47E+01	1.16E+02	7.08E+01

Table 9 (continued). Statistical results and comparison for CEC-BC- 2017 functions

Function		INFO	GWO	GSA	SCA	PSO	BA	GA
f_{21}	Mean	2.28E+03	2.30E+03	2.36E+03	**2.25E+03**	2.30E+03	2.32E+03	2.32E+03
	SD	5.39E+01	3.57E+01	2.22E+01	6.29E+01	5.82E+01	5.82E+01	4.36E+01
f_{22}	Mean	2.30E+03	2.33E+03	**2.30E+03**	2.36E+03	2.30E+03	2.63E+03	2.31E+03
	SD	1.65E+01	1.12E+02	6.85E-11	3.83E+01	9.27E-01	1.45E+02	1.03E+01
f_{23}	Mean	**2.62E+03**	2.62E+03	2.75E+03	2.66E+03	2.69E+03	2.66E+03	2.72E+03
	SD	7.28E+00	9.60E+00	5.60E+01	6.79E+00	3.96E+01	2.37E+01	4.77E+01
f_{24}	Mean	2.75E+03	2.75E+03	**2.55E+03**	2.78E+03	2.73E+03	2.77E+03	2.81E+03
	SD	7.87E+00	1.13E+01	1.17E+02	3.92E+01	1.32E+02	8.47E+01	1.38E+02
f_{25}	Mean	2.92E+03	2.94E+03	2.94E+03	2.96E+03	**2.92E+03**	3.01E+03	2.93E+03
	SD	3.07E+01	2.40E+01	1.36E+01	2.14E+01	2.29E+01	5.10E+01	2.31E+01
f_{26}	Mean	3.11E+03	3.06E+03	3.70E+03	3.07E+03	**3.07E+03**	3.42E+03	3.55E+03
	SD	3.71E+02	3.16E+02	6.96E+02	3.48E+01	2.85E+02	3.99E+02	4.82E+02
f_{27}	Mean	**3.09E+03**	3.09E+03	3.26E+03	3.10E+03	3.16E+03	3.13E+03	3.23E+03
	SD	1.61E+00	2.39E+00	3.95E+01	1.43E+00	4.39E+01	3.48E+01	4.36E+01
f_{28}	Mean	3.30E+03	3.36E+03	3.46E+03	3.29E+03	**3.17E+03**	3.43E+03	3.29E+03
	SD	1.66E+02	9.23E+01	3.08E+01	6.58E+01	4.16E+01	1.14E+02	1.58E+02
f_{29}	Mean	**3.17E+03**	3.19E+03	3.45E+03	3.23E+03	3.23E+03	3.38E+03	3.28E+03
	SD	3.11E+01	3.14E+01	1.42E+02	3.23E+01	4.08E+01	1.12E+02	6.89E+01
f_{30}	Mean	8.55E+04	9.19E+05	9.83E+05	1.04E+06	**9.80E+03**	2.18E+06	4.45E+05
	SD	2.49E+05	1.08E+06	2.61E+05	8.05E+05	5.69E+03	3.14E+06	1.01E+06

5.7. Ranking analysis of INFO

In this section, multiple statistical tests, including Friedman's [50], Bonferroni-Dunn's [53], and Holm's tests [54], are considered to evaluate the difference between the efficiency of INFO and the other optimizers in the test functions. To implement a trustworthy comparison, this research divided the test functions into three groups: G1 (group 1) includes unimodal, multimodal, and composite functions (Tables 2-4); G2 (group 2) consists of CEC-BC-2017 test functions (Table 8), and G3 (group 3) is the combination of G1 and G2.

Table 10. Mean rankings computed by Friedman test for CEC-BC-2017 functions

Function	INFO	GWO	GSA	SCA	PSO	BA	GA
f_1	1	5	2	6	4	7	3
f_3	1.5	3	6	4	1.5	7	5
f_4	1	5	3	6	2	7	4
f_5	1	2	7	5	3	6	4
f_6	1	2	6	4	3	7	5
f_7	2.5	4	1	6	2.5	7	5
f_8	1	2	3	7	4.5	6	4.5
f_9	2	4	2	5	2	7	6
f_{10}	1	2	7	5	3	6	4
f_{11}	1	2	3.5	6	3.5	7	5
f_{12}	1	3	4	7	2	6	5
f_{13}	1	4.5	4.5	7	3	6	2
f_{14}	1	5	7	2	3	4	6
f_{15}	1	4.5	6	2	3	7	4.5
f_{16}	1	3	7	2	4	6	5
f_{17}	1	2.5	6	5	4	7	2.5
f_{18}	1	6	2	7	3	5	4
f_{19}	1	5	7	3	2	6	4
f_{20}	1	2	7	4	3	6	5
f_{21}	2	3.5	7	1	3.5	5.5	5.5
f_{22}	2	5	2	6	2	7	4
f_{23}	1.5	1.5	7	3.5	5	3.5	6
f_{24}	3.5	3.5	1	6	2	5	7
f_{25}	1.5	4.5	4.5	6	1.5	7	3
f_{26}	4	1	7	2.5	2.5	5	6
f_{27}	1.5	1.5	7	3	5	4	6
f_{28}	4	5	7	2.5	1	6	2.5
f_{29}	1	2	7	3.5	3.5	6	5
f_{30}	2	4	5	6	1	7	3
Mean Rank	**1.55**	**3.38**	**5.02**	**4.59**	**2.86**	**6.07**	**4.53**
Final Rank	**1**	**3**	**6**	**5**	**2**	**7**	**4**

In the multiple statistical tests, the optimizers' results were first investigated to determine their equality. When inequality was found, post-hoc analysis was performed to find out which optimizer's performance is significantly different from INFO. Therefore, the Friedman test was conducted once again to obtain the optimizers' average ranks on the three groups, as shown in Fig. 7.

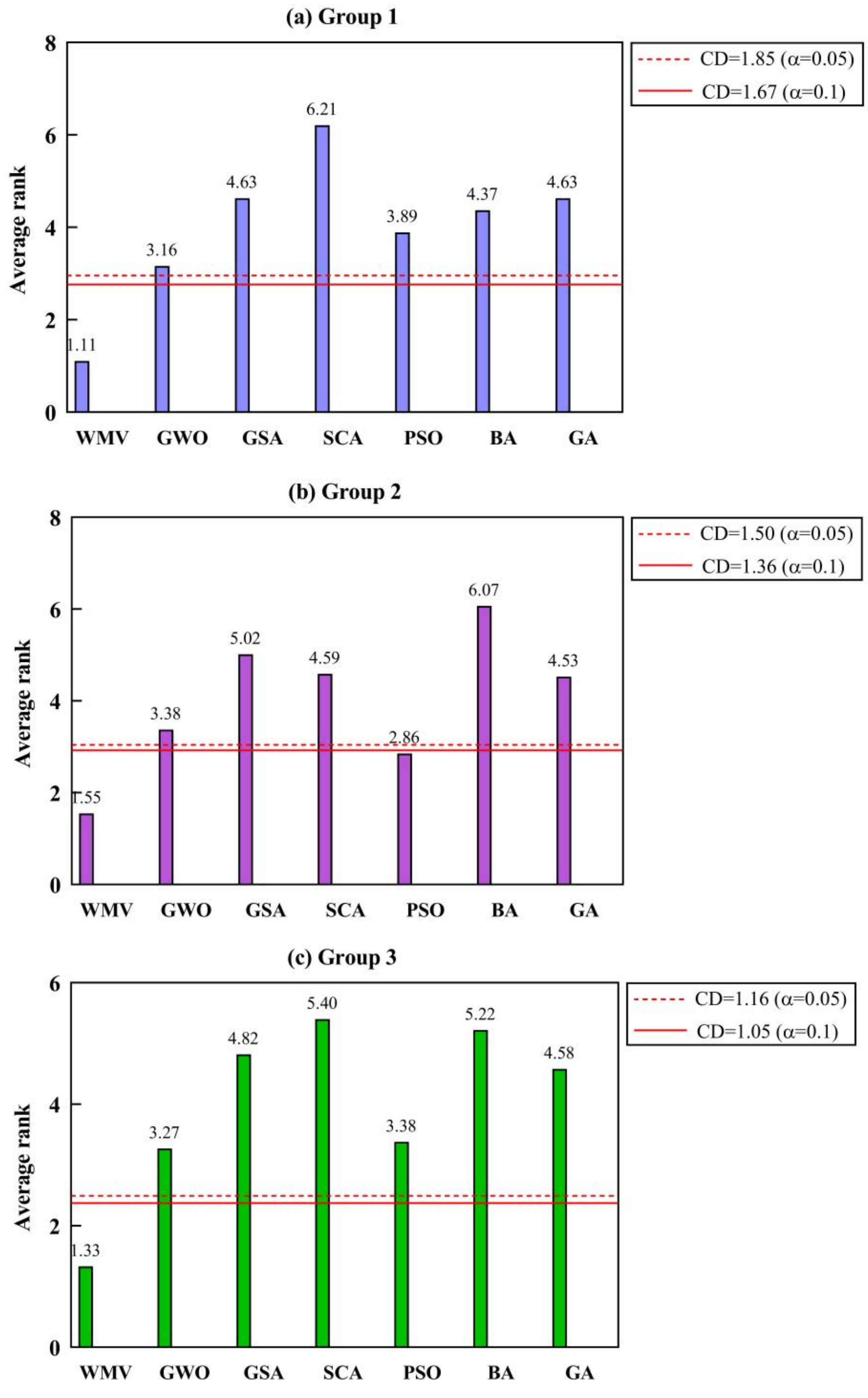


Fig. 7. Bonferroni-Dunn test for all optimizers and different groups

The Bonferroni-Dunn test is a post-hoc analysis used to determine if the efficiency of two optimizers is significantly different and if the difference between mean ranks of optimization methods is larger than the critical difference (CD):

$$CD = Q_{\alpha}\sqrt{\frac{m\left(m+1\right)}{6n}} \tag{13}$$

where Q_{α} is the critical value, calculated based on work in [55], and m and n are the numbers of optimizers and test problems, respectively. In this work, INFO was introduced as the control optimizer. In Fig. 8, the horizontal lines show CD as the threshold for the INFO algorithm. For two common significant levels of 0.05 and 0.1, the threshold lines were determined and are displayed in Fig. 8 as dashed and dotted lines, respectively. In the three groups, INFO had the lowest mean ranks (G1 = 1.11, G2 = 1.55, G3 = 1.33) and, thus, can outperform the other optimizers, which have mean ranks above the CD lines. It is pertinent to note that the PSO rank is below the threshold line in G2.

However, the Bonferroni-Dunn test does not determine the main difference between the optimizers if their mean ranks are less than the threshold line. Therefore, the present research used Holm's test to specify whether there is a substantial difference between the optimizers, with ranks less than the threshold line. To implement Holm's test, all optimizers were sorted based on their p-value and were compared with α/i, where i is the algorithm number. If the p-value is less than the corresponding significant level (α/i), the optimizer is significantly different. Tables 11 and 13 (G1 and G3) show the Bonferroni-Dunn test results for levels $\alpha = 0.05$ and 0.1, revealing a significant difference between the efficiency of INFO and the other optimizers. For G2 (Table 12), the Bonferroni-Dunn test indicates no significant difference between INFO and PSO, while Holm's test demonstrates a significant difference between these two. Consequently, in Fig. 8, the mean ranks of INFO in the three groups are very close to each other, while the other optimizers have unstable performance in various groups. Finally, it may be concluded that INFO has a reliable and accurate efficiency in all groups compared to the other optimizers.

Table 11. Holm's test for G1 test functions (INFO is as the control optimizer)

INFO VS.	Rank	Z-value	P-value	α/i ($\alpha = 0.05$)	α/i ($\alpha = 0.1$)
SCA	6.21	7.276	1.71E-13	0.00833	0.01667
GA	4.63	5.022	2.55E-07	0.01000	0.02000
GSA	4.63	5.022	2.55E-07	0.01250	0.02500
BA	4.37	4.651	1.65E-06	0.01667	0.03333
PSO	3.89	3.966	3.64E-05	0.02500	0.05000
GWO	3.16	2.924	1.72E-03	0.05000	0.10000

Table 12. Holm's test for G2 test functions (INFO is as the control optimizer)

INFO VS.	Rank	Z-value	P-value	α/i ($\alpha = 0.05$)	α/i ($\alpha = 0.1$)
BA	6.069	7.967	7.77E-16	0.00833	0.01667
GSA	5.017	6.116	4.78E-10	0.01000	0.02000
SCA	4.586	5.358	4.19E-08	0.01250	0.02500
GA	4.534	5.252	7.48E-08	0.01667	0.03333
GWO	3.379	3.225	6.28E-04	0.02500	0.05000
PSO	2.862	2.309	1.04E-02	0.05000	0.10000

Table 13. Holm's test for G3 test functions (INFO is as the control optimizer)

INFO VS.	Rank	Z-value	P-value	α/i ($\alpha = 0.05$)	α/i ($\alpha = 0.1$)
BA	5.40	5.807	3.18E-09	0.00833	0.01667
SCA	5.22	5.550	1.42E-08	0.01000	0.02000
GSA	4.82	4.979	3.18E-07	0.01250	0.02500
GA	4.58	4.637	1.76E-06	0.01667	0.03333
PSO	3.38	2.924	1.72E-03	0.02500	0.05000
GWO	3.27	2.767	2.82E-03	0.05000	0.10000

5.8. Performance comparison of INFO with advanced algorithms

In this section, the performance of INFO is compared with advanced algorithms, including SCADE [56], CGSCA [57], OBLGWO [58], RDWOA [59], CCMWOA [60], BMWOA [61], CLPSO [62], RCBA [63], and CBA [64] on the CEC-BC-2017 test functions. For all the optimization algorithms, the population size and the total number of generations were set to 30 and 500, respectively. To decrease the effect of random behavior in each optimizer on the results, all optimizers were run 30 different times for each test function.

Table 14 indicates the average (AVG) and standard deviation results obtained by all optimization methods, which confirm that INFO is a very competitive optimizer to solve CEC-BC-2017 functions. In f_1, f_3-f_{19}, f_{21}, f_{23}, f_{24}, f_{26}, f_{29}, and f_{30}, the AVG of INFO was smaller than that of the other optimizers. These results show that INFO performed better on 24 out of 29 test functions than other advanced algorithms.

Furthermore, the Wilcoxon Signed Ranks (WSR) test [65] was utilized to compare all optimizers' overall efficiency on 30 independent runs. In the WSR, R^+ indicates the sum of ranks during all runs in which INFO outperformed the competitor algorithm. Comparatively, R^- represents the sum of ranks during all runs in which the competitor algorithm outperformed INFO. P-value specifies the significance of the results in a statistical hypothesis test ($\alpha = 0.05$). The comparisons of the optimizers by the WSR test are reported in Table 15, where the symbol '+' shows that INFO has better efficiency than its competitor algorithm; '-' indicates that the competitor algorithm's efficiency is better than INFO; and '=' denotes similar performance between INFO and the competitor algorithm. Each test's statistical results for the 30 runs are presented in Table 16, which shows that the INFO algorithm can perform impressively better than its competitors.

Fig. 8 also depicts the convergence curve of some CEC-BC-2017 test functions. According to this figure, INFO can explore a superior solution at a fast convergence rate compared with the other optimizers. Moreover, the Friedman test was utilized to calculate all algorithms' average ranks, revealing that INFO has the best rank value (1.22) and performed much better than the other optimizers (Table 17).

Table 14. Statistical results and comparison for CEC-BC- 2017 functions

Function		INFO	SCADE	CGSCA	OBLGWO	RDWOA	CCMWOA	BMWOA	CLPSO	RCBA	CBA
f_{1}	Mean	**1.32E+05**	2.85E+10	2.53E+10	1.69E+08	1.01E+09	1.17E+10	1.23E+09	1.23E+10	7.87E+05	1.99E+06
	SD	2.15E+05	3.80E+09	3.85E+09	8.23E+07	1.35E+09	3.83E+09	5.66E+08	2.54E+09	2.65E+05	2.17E+06
f_{3}	Mean	**2.09E+04**	7.72E+04	7.06E+04	5.09E+04	6.27E+04	7.36E+04	8.10E+04	1.57E+05	9.56E+04	1.01E+05
	SD	8.46E+03	5.86E+03	8.92E+03	8.86E+03	1.21E+04	5.19E+03	7.01E+03	2.14E+04	4.39E+04	6.20E+04
f_{4}	Mean	**5.00E+02**	6.12E+03	3.46E+03	5.49E+02	6.39E+02	1.78E+03	7.29E+02	3.02E+03	5.08E+02	5.14E+02
	SD	2.78E+01	1.24E+03	1.15E+03	2.56E+01	1.18E+02	7.37E+02	7.13E+01	9.93E+02	2.58E+01	2.78E+01
f_{5}	Mean	**6.15E+02**	8.79E+02	8.53E+02	7.00E+02	7.32E+02	8.05E+02	8.26E+02	8.26E+02	8.20E+02	8.12E+02
	SD	2.92E+01	2.11E+01	2.63E+01	5.54E+01	6.57E+01	3.78E+01	3.01E+01	2.12E+01	5.21E+01	6.58E+01
f_{6}	Mean	**6.13E+02**	6.80E+02	6.71E+02	6.33E+02	6.36E+02	6.71E+02	6.68E+02	6.58E+02	6.73E+02	6.77E+02
	SD	6.73E+00	6.29E+00	6.67E+00	1.45E+01	8.44E+00	7.70E+00	1.05E+01	6.32E+00	9.04E+00	1.50E+01
f_{7}	Mean	**9.61E+02**	1.27E+03	1.24E+03	9.69E+02	1.06E+03	1.26E+03	1.24E+03	1.22E+03	1.87E+03	1.78E+03
	SD	7.97E+01	4.96E+01	4.69E+01	6.49E+01	8.65E+01	7.00E+01	9.11E+01	4.34E+01	2.88E+02	2.23E+02
f_{8}	Mean	**9.11E+02**	1.12E+03	1.11E+03	9.81E+02	9.92E+02	1.03E+03	1.02E+03	1.10E+03	1.05E+03	1.05E+03
	SD	3.41E+01	1.52E+01	2.90E+01	4.33E+01	4.26E+01	2.40E+01	3.45E+01	3.21E+01	5.66E+01	6.31E+01
f_{9}	Mean	**2.22E+03**	1.04E+04	9.62E+03	4.88E+03	7.18E+03	6.78E+03	8.59E+03	1.16E+04	9.02E+03	9.17E+03
	SD	5.49E+02	1.24E+03	1.48E+03	2.78E+03	2.65E+03	7.55E+02	1.41E+03	3.14E+03	2.78E+03	2.15E+03
f_{10}	Mean	**5.24E+03**	8.78E+03	8.84E+03	7.03E+03	6.07E+03	6.61E+03	7.95E+03	8.14E+03	5.70E+03	5.85E+03
	SD	5.98E+02	3.92E+02	3.34E+02	1.22E+03	8.02E+02	6.61E+02	5.30E+02	4.13E+02	5.26E+02	6.91E+02
f_{11}	Mean	**1.29E+03**	5.38E+03	4.17E+03	1.38E+03	2.00E+03	2.41E+03	2.16E+03	6.57E+03	1.33E+03	1.40E+03
	SD	5.82E+01	1.22E+03	1.20E+03	6.80E+01	9.12E+02	2.89E+02	5.14E+02	1.81E+03	9.98E+01	9.75E+01
f_{12}	Mean	**1.23E+06**	3.61E+09	3.19E+09	3.96E+07	2.79E+07	8.36E+08	1.48E+08	1.51E+09	1.40E+07	2.08E+07
	SD	1.45E+06	6.62E+08	8.69E+08	2.91E+07	2.17E+07	7.81E+08	1.01E+08	5.93E+08	9.70E+06	1.55E+07
f_{13}	Mean	**2.69E+04**	1.49E+09	1.46E+09	1.12E+06	2.06E+06	5.66E+07	1.99E+06	9.90E+08	1.71E+05	2.85E+05
	SD	1.83E+04	5.41E+08	8.40E+08	1.02E+06	4.22E+06	1.20E+08	2.28E+06	5.20E+08	1.31E+05	3.29E+05
f_{14}	Mean	**1.52E+04**	1.16E+06	9.53E+05	1.79E+05	1.42E+06	1.03E+06	1.18E+06	1.16E+06	6.23E+04	1.35E+05
	SD	2.53E+04	5.50E+05	5.37E+05	1.45E+05	1.49E+06	9.79E+05	9.67E+05	1.04E+06	4.53E+04	1.56E+05
f_{15}	Mean	**1.27E+04**	2.15E+07	4.38E+07	1.97E+05	1.52E+05	1.10E+06	4.51E+05	7.16E+07	6.63E+04	9.32E+04
	SD	1.18E+04	1.32E+07	4.45E+07	1.50E+05	2.89E+05	1.30E+06	5.16E+05	6.03E+07	4.42E+04	5.84E+04
f_{16}	Mean	**2.55E+03**	4.29E+03	4.24E+03	3.05E+03	3.13E+03	3.82E+03	3.67E+03	4.00E+03	3.96E+03	4.14E+03
	SD	3.49E+02	2.68E+02	2.65E+02	4.16E+02	3.26E+02	6.76E+02	5.07E+02	3.34E+02	4.85E+02	5.22E+02
f_{17}	Mean	**2.18E+03**	2.84E+03	2.94E+03	2.37E+03	2.44E+03	2.59E+03	2.53E+03	2.80E+03	2.76E+03	3.00E+03
	SD	2.00E+02	1.95E+02	2.04E+02	2.53E+02	2.50E+02	3.04E+02	2.34E+02	2.34E+02	2.12E+02	3.60E+02
f_{18}	Mean	**2.04E+05**	1.00E+07	1.53E+07	2.41E+06	3.70E+06	6.94E+06	5.50E+06	5.51E+06	7.43E+05	1.78E+06
	SD	1.37E+05	6.28E+06	9.82E+06	2.25E+06	3.23E+06	9.97E+06	6.22E+06	3.77E+06	6.03E+05	1.67E+06
f_{19}	Mean	**8.85E+03**	7.63E+07	9.17E+07	1.66E+06	1.77E+05	1.97E+06	1.30E+06	9.33E+07	1.13E+06	4.07E+06
	SD	8.50E+03	5.08E+07	3.77E+07	1.81E+06	3.45E+05	4.39E+06	1.27E+06	6.90E+07	6.91E+05	2.39E+06
f_{20}	Mean	2.63E+03	2.93E+03	2.94E+03	2.64E+03	**2.59E+03**	2.63E+03	2.77E+03	2.81E+03	3.07E+03	3.00E+03
	SD	2.10E+02	1.24E+02	1.41E+02	2.26E+02	1.99E+02	1.67E+02	1.77E+02	1.50E+02	2.17E+02	2.39E+02
f_{21}	Mean	**2.40E+03**	2.63E+03	2.62E+03	2.47E+03	2.53E+03	2.60E+03	2.55E+03	2.60E+03	2.65E+03	2.67E+03
	SD	2.23E+01	2.77E+01	2.41E+01	6.26E+01	5.16E+01	5.10E+01	4.01E+01	2.48E+01	6.48E+01	7.44E+01

Table 14. Statistical results and comparison for CEC-BC- 2017 functions (Continued)

Function		INFO	SCADE	CGSCA	OBLGWO	RDWOA	CCMWOA	BMWOA	CLPSO	RCBA	CBA
f_{22}	Mean	5.69E+03	6.69E+03	5.98E+03	**3.58E+03**	7.16E+03	7.02E+03	3.59E+03	7.37E+03	7.68E+03	7.63E+03
	SD	2.16E+03	1.15E+03	1.99E+03	2.29E+03	1.53E+03	1.57E+03	1.95E+03	1.82E+03	9.62E+02	7.15E+02
f_{23}	Mean	**2.78E+03**	3.08E+03	3.08E+03	2.84E+03	2.92E+03	3.12E+03	2.98E+03	3.10E+03	3.48E+03	3.38E+03
	SD	4.10E+01	3.82E+01	4.30E+01	4.92E+01	7.89E+01	1.02E+02	7.75E+01	4.98E+01	2.26E+02	2.11E+02
f_{24}	Mean	**2.94E+03**	3.23E+03	3.24E+03	3.00E+03	3.13E+03	3.27E+03	3.12E+03	3.24E+03	3.56E+03	3.50E+03
	SD	3.97E+01	4.51E+01	4.70E+01	5.04E+01	9.86E+01	1.01E+02	7.54E+01	5.79E+01	1.50E+02	1.77E+02
f_{25}	Mean	2.92E+03	3.91E+03	3.67E+03	2.96E+03	2.99E+03	3.22E+03	3.10E+03	3.74E+03	**2.92E+03**	2.93E+03
	SD	2.26E+01	2.85E+02	2.45E+02	2.46E+01	4.32E+01	1.19E+02	6.40E+01	1.71E+02	2.51E+01	2.14E+01
f_{26}	Mean	**5.08E+03**	8.25E+03	7.99E+03	5.45E+03	6.28E+03	8.55E+03	6.47E+03	7.88E+03	9.88E+03	1.00E+04
	SD	8.80E+02	5.70E+02	7.47E+02	9.26E+02	9.89E+02	7.57E+02	1.24E+03	5.57E+02	2.52E+03	2.52E+03
f_{27}	Mean	3.26E+03	3.57E+03	3.55E+03	**3.25E+03**	3.26E+03	3.51E+03	3.36E+03	3.58E+03	3.48E+03	3.50E+03
	SD	2.56E+01	5.40E+01	7.01E+01	2.60E+01	3.44E+01	9.90E+01	9.46E+01	8.89E+01	1.55E+02	1.90E+02
f_{28}	Mean	3.27E+03	5.12E+03	4.55E+03	3.35E+03	3.39E+03	4.08E+03	3.52E+03	4.85E+03	**3.24E+03**	3.36E+03
	SD	4.16E+01	4.33E+02	3.65E+02	4.04E+01	5.43E+01	3.03E+02	1.10E+02	4.12E+02	3.77E+01	5.93E+02
f_{29}	Mean	**4.21E+03**	5.64E+03	5.36E+03	4.27E+03	4.24E+03	5.13E+03	4.75E+03	5.06E+03	5.10E+03	5.54E+03
	SD	2.97E+02	3.57E+02	2.31E+02	2.71E+02	3.30E+02	5.55E+02	3.47E+02	2.46E+02	4.94E+02	8.40E+02
f_{30}	Mean	**1.77E+04**	2.40E+08	1.95E+08	6.58E+06	7.46E+05	3.74E+07	9.22E+06	7.41E+07	4.84E+06	8.96E+06
	SD	7.17E+03	8.45E+07	7.89E+07	5.37E+06	1.06E+06	3.23E+07	5.95E+06	4.08E+07	4.32E+06	5.37E+06

Table 15. Statistical results of single-problem-based WSR for INFO vs. SCADE, CGSCA and OBLGWO.

Functions	INFO vs. SCADE				INFO vs. CGSCA				INFO vs. OBLGWO			
	R+	R-	P-Value	Winner	R+	R-	P-Value	Winner	R+	R-	P-Value	Winner
F1	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F2	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F3	465	0	1.73E-06	+	465	0	1.73E-06	+	464	1	1.92E-06	+
F4	465	0	1.73E-06	+	465	0	1.73E-06	+	460	5	2.88E-06	+
F5	465	0	1.73E-06	+	465	0	1.73E-06	+	459	6	3.18E-06	+
F6	465	0	1.73E-06	+	465	0	1.73E-06	+	268	197	4.65E-01	=
F7	465	0	1.73E-06	+	465	0	1.73E-06	+	453	12	5.75E-06	+
F8	465	0	1.73E-06	+	465	0	1.73E-06	+	441	24	1.80E-05	+
F9	465	0	1.73E-06	+	465	0	1.73E-06	+	459	6	3.18E-06	+
F10	465	0	1.73E-06	+	465	0	1.73E-06	+	423	42	8.92E-05	+
F11	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F12	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F13	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F14	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F15	465	0	1.73E-06	+	465	0	1.73E-06	+	430	35	4.86E-05	+
F16	464	1	1.92E-06	+	465	0	1.73E-06	+	363	102	7.27E-03	+
F17	465	0	1.73E-06	+	465	0	1.73E-06	+	454	11	5.22E-06	+
F18	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F19	435	30	3.11E-05	+	437	28	2.60E-05	+	235	230	9.59E-01	=
F20	465	0	1.73E-06	+	465	0	1.73E-06	+	445	20	1.24E-05	+
F21	293	172	2.13E-01	=	283	182	2.99E-01	=	98	367	5.67E-03	-
F22	465	0	1.73E-06	+	465	0	1.73E-06	+	414	51	1.89E-04	+
F23	465	0	1.73E-06	+	465	0	1.73E-06	+	426	39	6.89E-05	+
F24	465	0	1.73E-06	+	465	0	1.73E-06	+	440	25	1.97E-05	+
F25	465	0	1.73E-06	+	465	0	1.73E-06	+	313	152	9.78E-02	=
F26	465	0	1.73E-06	+	465	0	1.73E-06	+	210	255	6.44E-01	=
F27	465	0	1.73E-06	+	465	0	1.73E-06	+	444	21	1.36E-05	+
F28	465	0	1.73E-06	+	465	0	1.73E-06	+	264	201	5.17E-01	=
F29	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+

Table 15. Statistical results of single-problem-based WSR for INFO vs. RDWOA, CCMWOA, and BMWOA (Continued).

Functions	INFO vs. RDWOA				INFO vs. CCMWOA				INFO vs. BMWOA			
	R^+	R^-	P-Value	Winner	R^+	R^-	P-Value	Winner	R^+	R^-	P-Value	Winner
F1	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F2	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F3	462	3	2.35E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F4	462	3	2.35E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F5	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F6	421	44	1.06E-04	+	465	0	1.73E-06	+	464	1	1.92E-06	+
F7	450	15	7.69E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F8	464	1	1.92E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F9	410	55	2.61E-04	+	459	6	3.18E-06	+	465	0	1.73E-06	+
F10	448	17	9.32E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F11	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F12	279	186	3.39E-01	=	465	0	1.73E-06	+	465	0	1.73E-06	+
F13	465	0	1.73E-06	+	464	1	1.92E-06	+	464	1	1.92E-06	+
F14	380	85	2.41E-03	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F15	461	4	2.60E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F16	422	43	9.71E-05	+	436	29	2.84E-05	+	444	21	1.36E-05	+
F17	462	3	2.35E-06	+	464	1	1.92E-06	+	465	0	1.73E-06	+
F18	351	114	1.48E-02	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F19	206	259	5.86E-01	=	224	241	8.61E-01	=	345	120	2.07E-02	+
F20	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F21	389	76	1.29E-03	+	349	116	1.66E-02	+	85	380	2.41E-03	-
F22	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F23	464	1	1.92E-06	+	465	0	1.73E-06	+	462	3	2.35E-06	+
F24	464	1	1.92E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F25	413	52	2.05E-04	+	465	0	1.73E-06	+	414	51	1.89E-04	+
F26	246	219	7.81E-01	=	465	0	1.73E-06	+	446	19	1.13E-05	+
F27	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F28	250	215	7.19E-01	=	457	8	3.88E-06	+	437	28	2.60E-05	+
F29	464	1	1.92E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+

Table 15. Statistical results of single-problem-based WSR for INFO vs CLPSO, RCBA and CBA (Continued).

Functions	INFO vs. CLPSO				INFO vs. RCBA				INFO vs. CBA			
	R^+	R^-	P-Value	Winner	R^+	R^-	P-Value	Winner	R^+	R^-	P-Value	Winner
F1	465	0	1.73E-06	+	464	1	1.92E-06	+	420	45	1.15E-04	+
F2	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F3	465	0	1.73E-06	+	296	169	1.92E-01	=	314	151	9.37E-02	=
F4	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F5	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F6	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F7	464	1	1.92E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F8	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F9	465	0	1.73E-06	+	369	96	4.99E-03	+	380	85	2.41E-03	+
F10	465	0	1.73E-06	+	317	148	8.22E-02	=	431	34	4.45E-05	+
F11	465	0	1.73E-06	+	464	1	1.92E-06	+	465	0	1.73E-06	+
F12	465	0	1.73E-06	+	454	11	5.22E-06	+	457	8	3.88E-06	+
F13	465	0	1.73E-06	+	442	23	1.64E-05	+	446	19	1.13E-05	+
F14	465	0	1.73E-06	+	458	7	3.52E-06	+	465	0	1.73E-06	+
F15	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F16	464	1	1.92E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F17	464	1	1.92E-06	+	443	22	1.49E-05	+	455	10	4.73E-06	+
F18	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F19	390	75	1.20E-03	+	450	15	7.69E-06	+	431	34	4.45E-05	+
F20	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F21	350	115	1.57E-02	+	397	68	7.16E-04	+	428	37	5.79E-05	+
F22	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F23	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+
F24	465	0	1.73E-06	+	197	268	4.65E-01	=	335	130	3.50E-02	+
F25	465	0	1.73E-06	+	456	9	4.29E-06	+	456	9	4.29E-06	+
F26	465	0	1.73E-06	+	460	5	2.88E-06	+	465	0	1.73E-06	+
F27	465	0	1.73E-06	+	107	358	9.84E-03	-	163	302	1.53E-01	=
F28	464	1	1.92E-06	+	462	3	2.35E-06	+	464	1	1.92E-06	+
F29	465	0	1.73E-06	+	465	0	1.73E-06	+	465	0	1.73E-06	+

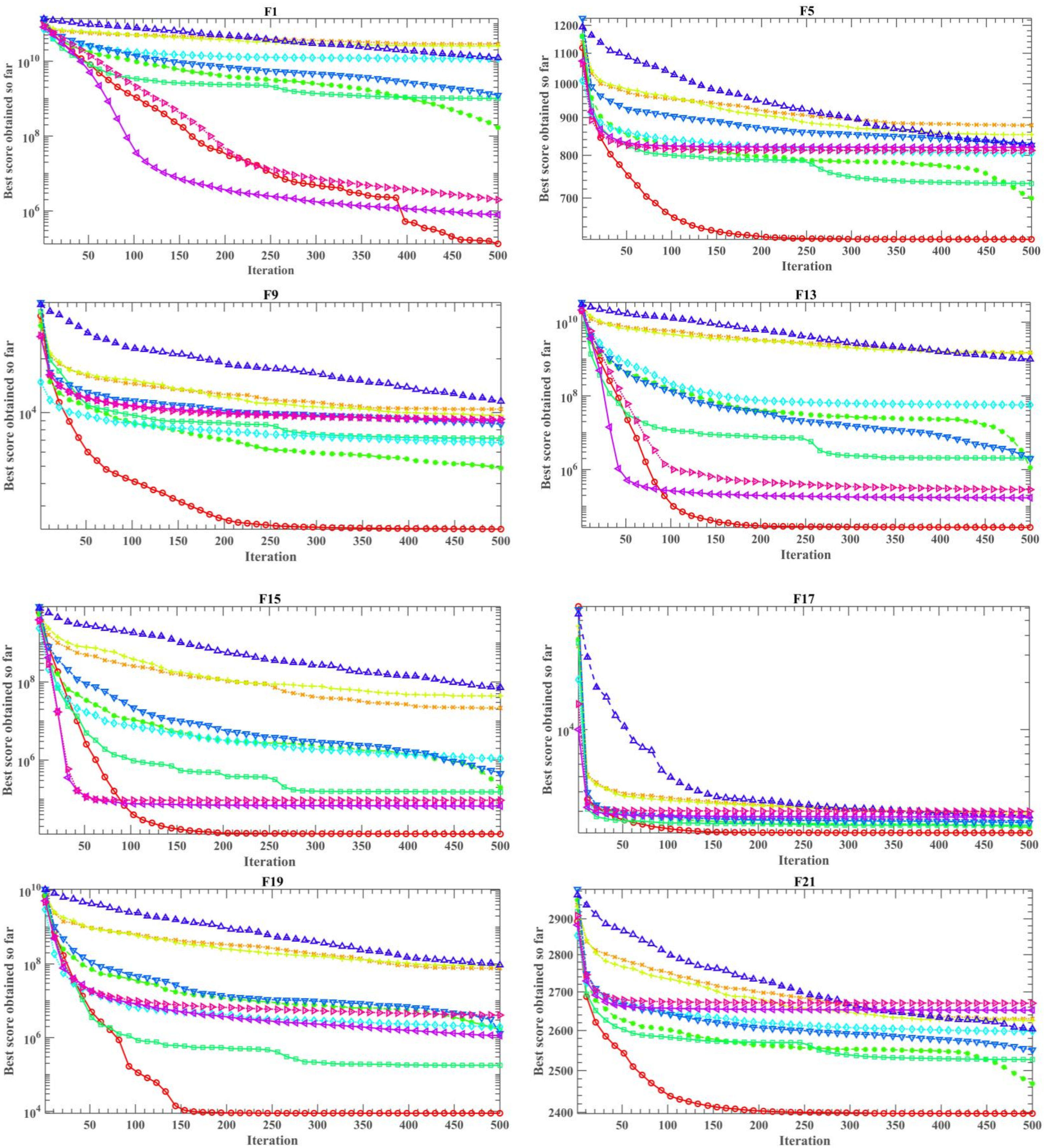
Table 16. Statistical results of the WSR of INFO

Functions	INFO vs. SCADE	INFO vs. CGSCA	INFO vs. OBLGWO	INFO vs. RDWOA	INFO vs. CCMWOA	INFO vs. BMWOA	INFO vs. CLPSO	INFO vs. RCBA	INFO vs. CBA
	(+/=/-)	(+/=/-)	(+/=/-)	(+/=/-)	(+/=/-)	(+/=/-)	(+/=/-)	(+/=/-)	(+/=/-)
Unimodal	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
Multimodal	7/0/0	7/0/0	6/1/0	7/0/0	7/0/0	7/0/0	7/0/0	6/1/0	6/1/0
Hybrid	10/0/0	10/0/0	9/1/0	8/2/0	9/1/0	10/0/0	10/0/0	9/1/0	10/0/0
Composition	9/1/0	9/1/0	6/3/1	8/2/0	10/0/0	9/0/1	10/0/0	8/1/1	9/1/0
Total	**28**/1/0	**28**/1/0	**23**/5/1	**25**/4/0	**28**/1/0	**28**/0/1	**29**/0/0	**25**/3/1	**27**/2/0

Table 17. Average ranking values of the INFO and other optimizers utilizing the Friedman test

Algorithm	Friedman ranking	Rank
INFO	**1.22**	**1**
SCADE	8.55	10
CGSCA	7.76	9
OBLGWO	3.03	2
RDWOA	3.88	3
CCMWOA	6.22	6
BMWOA	5.21	4
CLPSO	7.62	8
RCBA	5.24	5
CBA	6.26	7

All codes and source files are available at https://aliasgharheidari.com/INFO.html



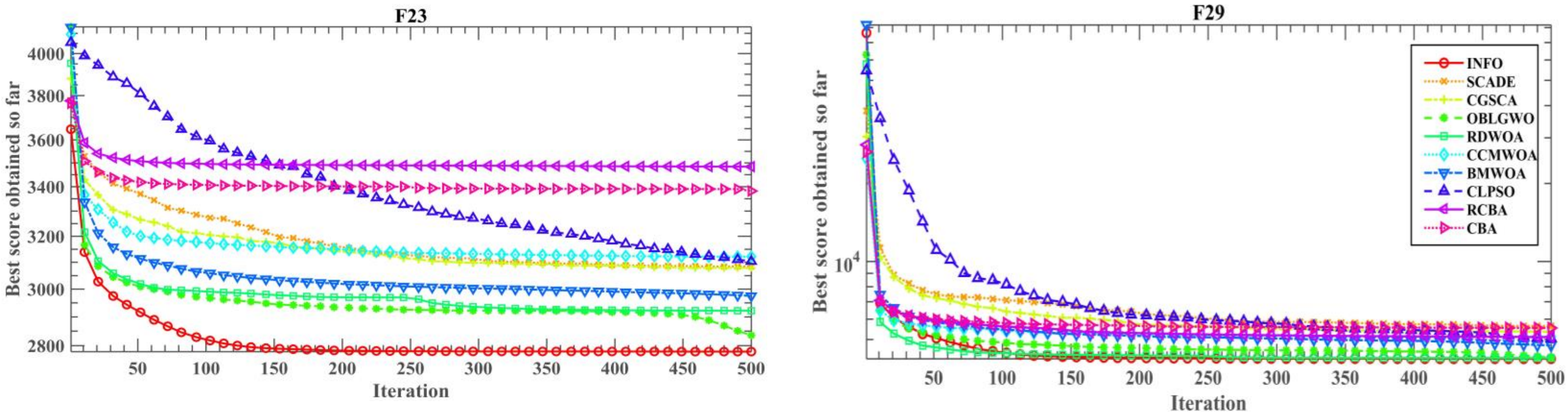
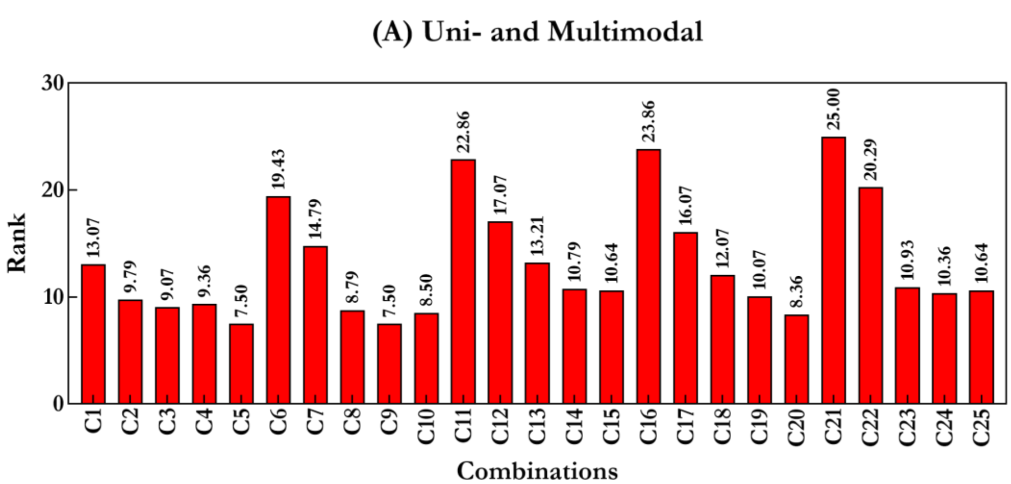


Fig. 8. Comparison of convergence curve of INFO and the other optimizers in some of the CEC-BC-2017 test functions.

5.9. Sensitivity analysis of INFO

A sensitivity analysis (SA) was performed to determine the best values for the two-parameter settings of INFO (i.e., *c* and *d*). Various combinations of the parameter settings were evaluated on 14 test functions to design INFO, including two sets: 7 uni- and multimodal test functions (set 1: f_1, f_3, f_5, f_7, f_{10}, f_{11}, f_{13}) and 7 test functions of CEC-2017 (set 2: f_1, f_5, f_{10}, f_{15}, f_{20}, f_{25}, f_{30}). Therefore, the amounts of each parameter were expressed by c = [64 8, 10] and d = {2, 4, 6, 8, 10}. Considering that each parameter has five values, there are 25 combinations for designing the parameter settings. In addition, each combination was assessed by the average objective function calculated over 30 various runs. The Friedman rank test was considered for ranking of each combination. The average ranks of each test function set are displayed in Fig. 9, and the average ranks of the two sets are depicted in Fig. 10. According to Fig. 10, the best rank belongs to C3 ($c = 2$ and $d = 4$). Therefore, the results showed that the values selected for the parameter settings in this study are the same as those obtained by the SA.



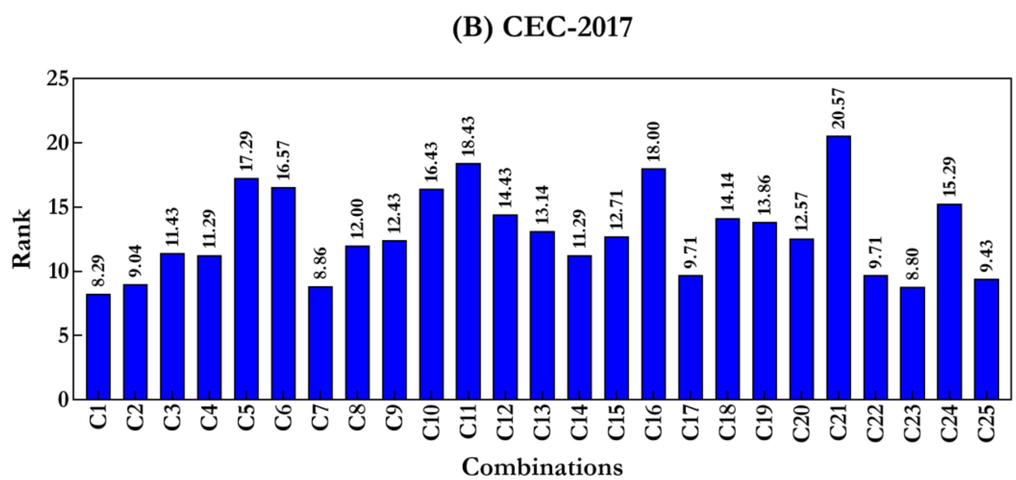


Fig. 9. Sensitivity analysis of INFO: ranks of (A) uni- and multi-modal test functions and (B) CEC-2017

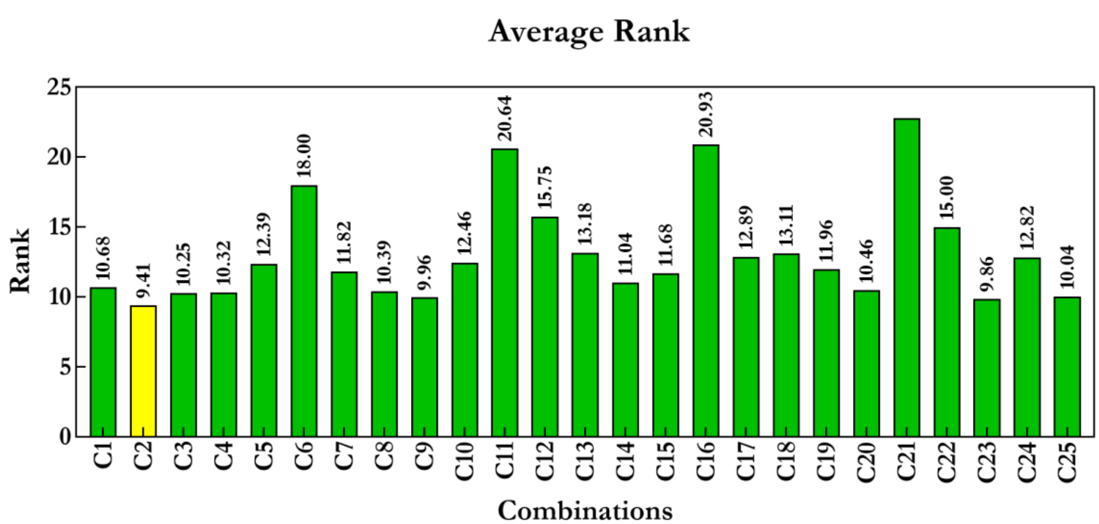


Fig. 10. Comparison of average ranks of all combinations for INFO algorithm

6. Performance of INFO on engineering design problems

Four constrained real-world engineering optimization problems were considered to evaluate the efficiency of the INFO algorithm further. Since these problems have various constraints, it is essential to implement a method that can handle them. Therefore, this study applied a simple method, called the death penalty, for handling the constraints, which devotes a significant value to the objective function (for minimization problems) if solutions violate each constraint [66].

6.1. Tension/compression spring design using INFO

The goal of the tension/compression spring problem is to minimize the weight of the design [67]. The constraints of this problem include surge frequency, shear stress, and minimum deflection [68, 69] (See Fig. 11). The decision variables (d_w) are the number of active coils (m), mean coil diameter (D_c), and wire diameter, which are formulated as follows:

Consider $\vec{x} = [m, D_c, d_w]$

Minimize $Z(\vec{x}) = (d_w + 2)D_c\text{m}^2$

Subject to: $g_1(\vec{x}) = 1 - \frac{{D_c}^3 d_w}{71785m^4} \leq 0,$

$$g_2(\vec{x}) = \frac{4{D_c}^2 - md_w}{12566(D_c m^3 - m^4)} + \frac{1}{5108m^2} \leq 0,$$

$$g_3(\vec{x}) = 1 - \frac{140.45m}{{D_c}^2 d_w} \leq 0, \tag{14}$$

$$g_4(\vec{x}) = \frac{m + D_c}{1.5} - 1 \leq 0,$$

$0.05 \leq m \leq 2.00,$

$0.25 \leq D_c \leq 1.30,$

$2.00 \leq d_w \leq 15.00,$

This test problem was optimized using various meta-heuristic methods, such as GA [70], DE [71], PSO [72], harmony search (HS) [73], Evolution Strategy (ES) [74], and whale optimization algorithm (WOA) [17]. The calculated results using INFO and the other optimizers are listed in Table 18, which clearly indicates that the INFO algorithm has a more suitable performance.

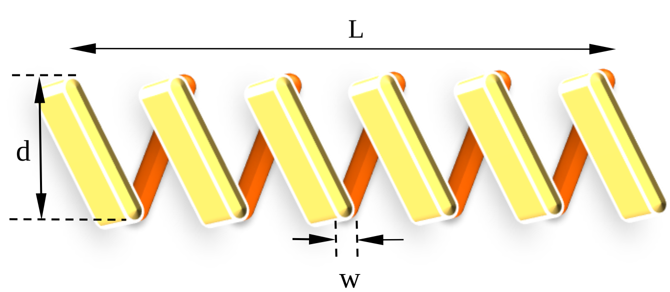


Fig. 11. The shape of the tension/compression spring problem

Table 18. Results of INFO, WOA, DE, PSO, GA, HS, and ES algorithms for the tension/compression problem

Optimizers	Optimal decision variables			Optimal weight
	m	D_c	d_w	
INFO	0.051555	0.353499	11.48034	**0.012666**
WOA [17]	0.051207	0.345215	12.00403	0.012676
DE [71]	0.051609	0.354714	11.41083	0.012670
PSO [72]	0.051728	0.357644	11.24454	0.012674
GA [70]	0.051480	0.351661	11.63220	0.012704
HS [73]	0.051154	0.349871	12.07643	0.012678
ES [74]	0.51989	0.363965	10.89052	0.012681

6.2. Three-bar truss design using INFO

The goal of the three-bar truss problem is to minimize the weight of the design [75, 76]. Fig. 12 displays the components of this case, which have buckling, stress, and deflection constraints. The decision variables are the cross-sectional areas of the truss bars (A_1, A_2). This problem has been widely used due to its complex constrained search domain [77, 78] and is expressed as follows:

$$\text{Minimize} \quad Z(\vec{x}) = (2\sqrt{2}A_1 + A_2) \times l,$$

$$\text{Subject to:} \quad g_1(\vec{x}) = \frac{\sqrt{2}A_1+A_2}{\sqrt{2}A_1^2+2A_1A_2}P - \sigma \leq 0,$$

$$g_2(\vec{x}) = \frac{A_2}{\sqrt{2}A_1^2+2A_1A_2}P - \sigma \leq 0, \quad (15)$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}A_2+A_1}P - \sigma \leq 0,$$

$$0 \leq A_1, A_2 \leq 1, l = 100\ cm,\ P = 2\frac{kN}{cm^2}, \sigma = 2\frac{kN}{cm^2}$$

The problem was optimized using several meta-heuristic algorithms, such as moth-flame optimization (MFO) [79], ant lion optimizer (ALO) [77], cuckoo search (CS) [76], differential evolution with dynamic stochastic (DEDS) [80], mine blast algorithm (MBA) [78], and hybridizing particle swarm optimization with differential evolution (PSO-DE) [81]. According to Table 19, INFO yielded highly promising results with a better calculated best objective function than the other contestant algorithms, except for the PSO-DE algorithm. It is noteworthy that the proposed algorithm and PSO-DE produced very similar solutions. The calculated results again confirm the ability of INFO to optimize complicated and constrained problems efficiently.

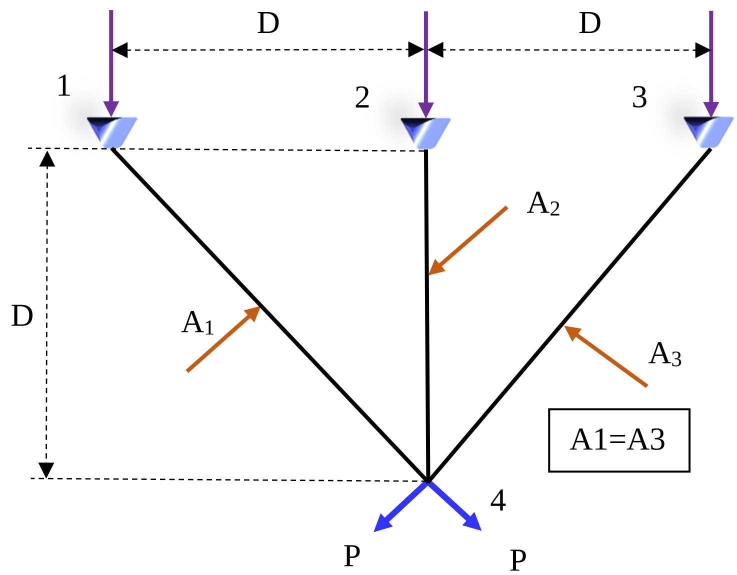


Fig. 12. Components of the 3-bar truss design problem.

Table 19. Results of INFO, MFO, ALO, CS, DEDS, MBA, and PSO-DE for three-bar truss problem

Optimization algorithm	Optimal decision variables		Optimal weight
	x_1	x_2	
INFO	0.788672734	0.408255081	263.8958434
MFO [79]	0.788244770	0.409466905	263.8959796
ALO [77]	0.7886628	0.408283133	263.8958434
CS [76]	0.78867	0.40902	263.9716
DEDS [80]	0.78867513	0.40824828	263.8958434
MBA [78]	0.7885650	0.4085597	263.8958522
PSO-DE [81]	0.7886751	0.4082482	263.8958433

6.3. I-beam design using INFO

The ability of INFO was further verified using another real engineering design problem with four variables [76, 82]. The problem aims to minimize the vertical deflection of the I-beam depicted in Fig. 13. The decision variables are height (h), length (l), and thicknesses of the beam web (t_w) and flange (t_f). The problem formulation is defined in Eq. (16):

$$Minimize\ \ f(x) = \frac{5000}{\frac{1}{12}t_w(h-2t_f)^3 + \frac{1}{6}lt_f^3 + 2lt_f\left(\frac{h-t_f}{2}\right)^2}$$

Subject to: $g_1(x) = 2lt_f + t_w(h-2t_f)^3 \leq 300$ (16)

$$g_2(x) = \frac{180000x_1}{t_w(h-2t_f)^3 + 2lt_f[4t_f^2 + 3h(h-2t_f)]} + \frac{15000x_2}{(h-2t_f)t_w^3 + 2t_f l^3} \leq 6$$

The initial design space has the following dimensions:

$$10 \leq h \leq 80, \quad 10 \leq l \leq 50, \quad 0.9 \leq t_w \leq 5, \quad 0.9 \geq t_f \leq 5,$$

The problem was optimized by the adaptive response surface method (ARSM) [83], CS [76], improved ARSM (IARSM) [83], and symbiotic organisms search (SOS) [75]. The results obtained by all optimizers are reported in Table 20, which reveals that INFO has superior performance to minimize vertical deflection compared to the other optimizers.

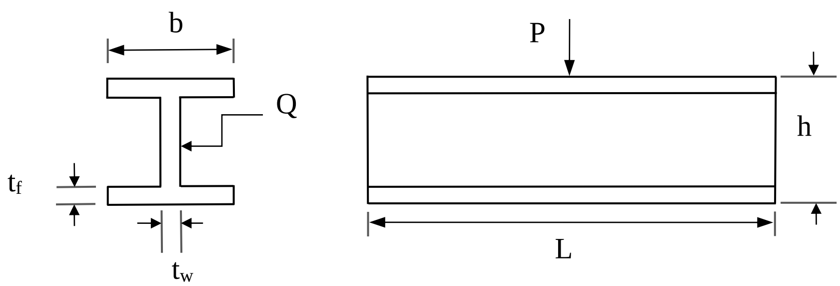


Fig. 13. The shape of the I-beam design problem

Table 20. Results of INFO, ARSM, CS, IARSM, and SOS for I-beam problem

Optimization algorithm	Optimal decision variables				Minimal vertical deflection
	x_1	x_2	x_3	x_4	
INFO	80	50	0.90	2.32	0.0130741

ARSM [83]	80	37.05	1.71	2.31	0.0157
CS [76]	80	50	0.90	2.32	0.0130747
IARSM [83]	79.99	48.42	0.90	2.40	0.131
SOS [75]	80	50	0.90	2.32	0.0130741

6.4. Optimal operation of a four-reservoir system using INFO

In this problem introduced by Chow and Cortes-Rivera [84], a system with four reservoirs is considered to evaluate the capability of INFO in the field of water resources systems. The goal of the problem is to maximize benefits throughout the operation period. Fig. 14 displays the schematic of this system. The decision variables are the water release volumes from reservoirs during the operation period (12 months), in which the number of these variables is 48. The objective function of the problem is presented as follows:

$$Maximize\ Z = \sum_{m=1}^{M}\sum_{t=1}^{T}(c^m{}_t . O^m{}_t) \tag{17}$$

where Z is the maximum benefit from reservoirs; M is the total number of reservoirs; T is the total number of periods; $c^m{}_t$ is the benefit obtained from the *m*-th reservoir in period *t*; and $O^m{}_t$ is the volume of release from *m*-th reservoir in period *t*.

The main constraints of this problem are expressed as follows:

$$V^m{}_{t+1} = V^m{}_t + I^m{}_t - O^m{}_t \tag{17.1}$$

$$O^m_{\min} \le O^m{}_t \le O^m_{\max} \tag{17.2}$$

$$V^m_{\min} \le V^m{}_t \le V^m_{\max} \tag{17.3}$$

$$V^m_{T+1} = V^m_1 \tag{17.4}$$

where V^m_t is the storage of *m*-th reservoir in the period of t; $I^m{}_t$ is the volume of inflow into *m*-th reservoir in operation period *t*; $V^m_{\min}$ and $V^m_{\max}$ are the minimum, and maximum *m*-th reservoir storage; and $O^m_{\min}$ and $O^m_{\max}$ are the minimum and maximum *m*-th reservoir water release, respectively. As stated before, the decision variables are water release volumes, thus the constraints related to the release are handled by INFO, while the constraints on the volumes of reservoir storage are handled by the penalty functions (PFs). Therefore, the objective function presented in Eq. (17) can be rewritten by adding the PFs, which are expressed as:

$$P_1 = \begin{cases} g_1 . (V^m_{T+1} - V^m_1)^2 & if\ V^m_{T+1} \ne V^m_1 \\ 0 & other\ wise \end{cases} \tag{18}$$

$$P_2 = \begin{cases} g_2.(V_{min}^m - V_t^m)^2 & if\ V_t^m < V_{min}^m \\ g_3.(V_{min}^m - V_t^m)^2 & if\ V_t^m > V_{max}^m \\ 0 & otherwise \end{cases} \tag{19}$$

where g_1, g_2, and g_3 express the PF coefficients and are equal to 40, 60, and 60, correspondingly [85-89]. Therefore, the objective function can be defined as:

$$Maximize\ \ Z = \sum_{m=1}^{M}\sum_{t=1}^{T}(C_t^m . O_t^m) - (P_1 + P_2) \tag{20}$$

In this problem, the number of function evaluations (NFE) and population size were 0.22×10^6 and 50, respectively.

Table 21 compares the results obtained by the INFO algorithm on this problem to those of the honey-bee mating optimization (HBMO) [86], gradient evolution (GE) [85], water cycle algorithm (WCA) [90], weed optimization algorithm (WOA) [91], and improved bat algorithm (IBA) [92]. According to Table 20, the best value of the objective function provided by the INFO is identical to those found by LP and IBA. It is also evident that INFO could outperform HBMO, GE, WCA, and WOA, while the number of function evaluations in Table 20 indicates that INFO can reach the global optimum with a lower NFE than the other algorithms.

The results of all problems confirm the high capability of INFO to solve constrained and challenging mathematical and real engineering problems. Accordingly, this robust optimization method can be suggested to explore the optimal solutions in various types of problems.

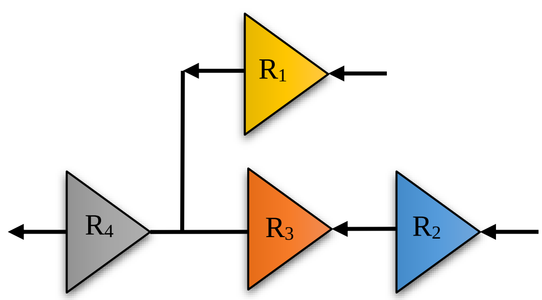


Fig. 14. Schematic of 4-reservoir problem

Table 21. Results of LP, INFO, HBMO, GE, WCA, WOA, and IBA for four-reservoir problem

Study	Optimization method	The best objective function value	NFE*
[84]	LP	308.29	-
Present study	INFO	**308.8324**	0.30×10^6
[86]	HBMO	308.07	1.10×10^6

[85]	GE	308.26	0.80×10^{6}
[90]	WCA	306.92	0.50×10^{6}
[91]	WOA	308.15	1.60×10^{6}
[92]	IBA	308.29	0.23×10^{6}
[87]	GSA	308.70	0.50×10^{6}
[93]	ABC	257.5169	0.20×10^{6}

6.5. Optimal operation of a ten-reservoir system using INFO

A well-known benchmark problem in the field of optimizing reservoir operation, namely the ten-reservoir problem, was considered in this study to further evaluate the efficiency of INFO. The problem presented by Murray and Yakowitz (1979) [94] has remarkable complexity due to many decision variables and constraints. The main aim of this system is the maximization of hydropower generation. Similar to the four-reservoir problem in terms of objective function and constraints, the decision variables are water release volumes over the operation period (12 time steps), in which the number of decision variables is equal to 120. Fig. 15 displays the schematic of the problem, and further details of the problem are presented in [94]. In addition, all coefficients of PFs are equal to 60 (i.e., $g_1 = g_2 = g_3 = 60$) [86, 95] for this problem.

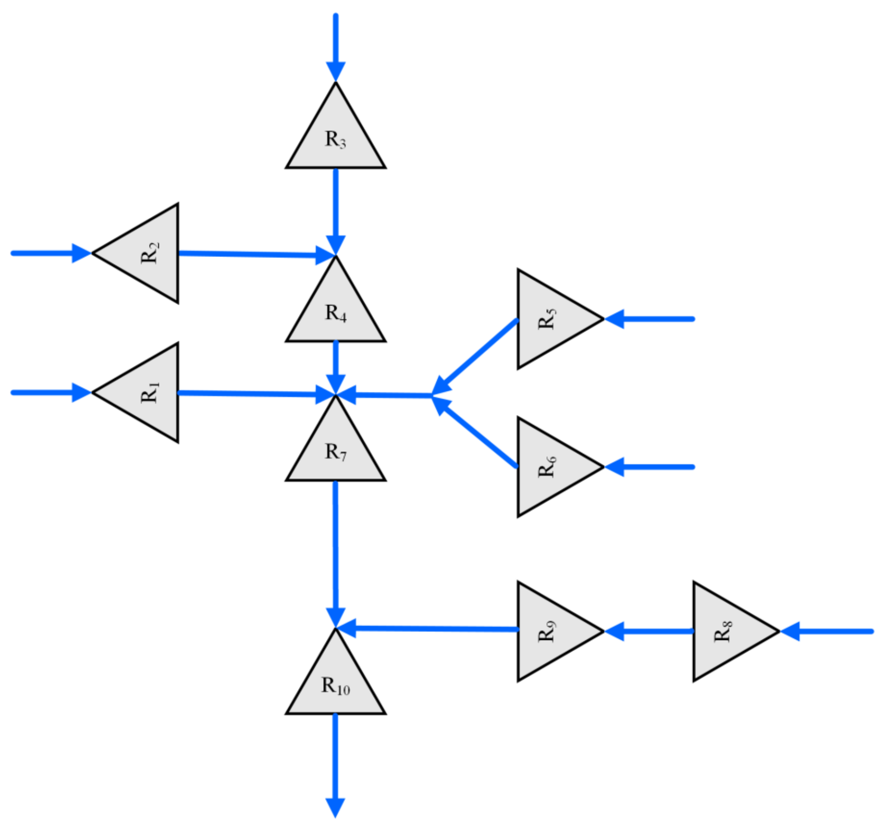


Fig. 15. Schematic of ten-reservoir problem

Table 22 compares the results achieved by the INFO algorithm on this problem to those of the differential dynamic programming (DDP) ([94]), honey-bee mating optimization (HBMO) [86], firefly algorithm (FA) [85], fully constrained improved artificial bee colony (FCIABC) [90], interior search algorithm (ISA) [91], hybrid Whale-GA algorithm (HWGA) [96], and improved bat algorithm (IBA) [92]. In addition, the results showed that the INFO can converged to (11.94.164) 0.999% of the global optimum solution (1194.44), while the DDP, HBMO, ISA, FA, FCIABC, WOA, GA, HWGA, and IBA can converged to (1148.05) 0.996, (1186.481) 0.961, (1088.23) 0.911, (1181.32) 0.989, (1189.97) 0.996, (913.80) 0.765, (1035.27) 0.866, (1136.03) 0.951, (1192.89) 0.998% of global optimum. According to Table 21, the best value of the objective function achieved by INFO is more accurate than those found by other methods, again demonstrating the superior ability of INFO.

Table 22. Comparative results of INFO and other reported studies on the ten-reservoir problem

Study	Optimization method	Best objective function value	Worst objective function value	Mean objective function value
[94]	DDP	11,90.65	---	---
Present study	**INFO**	**11,94.37**	**11,90.795**	**11,94.164**
([86, 89]	HBMO	11,56.79	11,39.43	11,48.05
	LP	11,94.44	---	---
[97]	ISA	11,93.8607	11,86.4081	11,86.4081
[88]	FA	1,107.85	1,097.41	1,088.23
	MFA	1,185.00	1,183.59	1,181.32
[93]	FCIABC	1,192.02	1,188.23	1,189.97
[96]	WOA	1,008.36	826.77	913.80
	GA	1,069.52	1,010.24	1,035.27
	HWGA	1,161.51	1,115.67	1,136.03
[98]	IBA	1,193.92	1,191.22	1,192.89

7. Conclusions and future directions

This research introduces a new population-based optimization algorithm developed based on the weighted mean of vectors, namely the INFO algorithm. INFO employs three leading operators, updating rule, vector combining, and local search operators, to change the population's position (vectors) in the search domain. To calculate the weight of vectors, a wavelet function was employed in this study. Forty-eight mathematical benchmark functions were applied to verify the INFO algorithm's efficiency concerning exploitation, exploration, escaping local optimum, and convergence speed. The calculated results prove the superior capability of the proposed INFO algorithm to solve benchmark functions compared to other optimizers, such as GA, PSO, BA, GWO, SCA, GSA, and the advanced algorithms like the SCADE, CGSCA, OBLGWO, RDWOA, CCMWOA, BMWOA, CLPSO, RCBA, and CBA. Initially, the exploitative and exploratory behaviors of INFO were evaluated according to the obtained results on unimodal and multimodal functions, respectively. Then, the composite functions were applied to indicate the capability of INFO to escape local optimum solutions, and the reasonable convergence rate of this algorithm was validated. In order to further evaluate the efficiency of the INFO algorithm, the CEC-BC-2017 test functions were considered. The post-doc statistical analysis results disclose that the INFO algorithm is a very competitive optimizer and outperforms other optimizers.

The proposed algorithm's ability to solve real engineering problems was tested by four constrained, complex, and challenging problems. To demonstrate the superior capability of INFO, its results were compared to the previously-mentioned optimizers. The obtained results confirm the high competency of INFO to optimize real problems with complicated and unknown search domains. Regarding this research, the following concluding remarks can be expressed:

- The proposed mean rule combines the weighted mean of two sets of vectors (a set of random vectors and another with local best, better, and worst vectors) as a strategy to promote exploration ability.
- The proposed updating rule operator updates vectors' position using the mean rule and convergence acceleration (CA) part, which guarantees the search ability and convergence speed of INFO.
- The scaling rate parameter (σ) can balance the exploration and exploitation search ability.
- To calculate the weighted mean of vectors, a wavelet function is considered to obtain vectors' weight, which allows the algorithm to search the solution space globally.
- The proposed vector combining operator combines global exploration and local exploitation phases to promote the search ability and escape from local optima.
- To ensure avoidance of locally optimal solutions, the proposed local search operator is included.
- The convergence speed of INFO is very promising because the positions of vectors always tend to move toward the regions with better solutions.
- The INFO algorithm can solve real complex and challenging optimization problems with constrained and unknown search domains.

For future studies, the following amendments and considerations are suggested to enhance INFO. Firstly, it may be wise to consider using different types of local search operators for INFO. Second, it may be beneficial to develop a binary and multi-objective version of INFO. The proposed INFO can also be enriched in terms of exploratory and exploitative trends using different concepts, such as chaotic maps, opposition-based learning, memory, multi-population structure, co-evolutionary methods, evolutionary population dynamics, greedy search, random learning mechanisms, and orthogonal learning methods. The conventional INFO or its enhanced variants can be applied to new horizons in tune and adjustment of neural networks, enhancement of prediction methods, distributed optimization, deep learning simulations, evolution, and practical studies, parameters optimization, optimal resource allocation, deployment optimization, and more CEC global optimization problems.

Acknowledgments

We sincerely acknowledge the efforts of the editor and the comments of anonymous reviewers that enhanced the work.

Uncategorized References

[1] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[2] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Optimizing multiple linear rules for Multi-Reservoir hydropower systems using an optimization method with an adaptation strategy," *Water Resources Management,* vol. 33, no. 12, pp. 4265-4286, 2019.

[3] I. Ahmadianfar, Z. Khajeh, S.-A. Asghari-Pari, and X. Chu, "Developing optimal policies for reservoir systems using a multi-strategy optimization algorithm," *Applied Soft Computing,* vol. 80, pp. 888-903, 2019.

[4] I. Ahmadianfar, A. Kheyrandish, M. Jamei, and B. Gharabaghi, "Optimizing operating rules for multi-reservoir hydropower generation systems: An adaptive hybrid differential evolution algorithm," *Renewable Energy,* 2020.

[5] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE control systems,* vol. 22, no. 3, pp. 52-67, 2002.

[6] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[7] Q. Yuan and F. Qian, "A hybrid genetic algorithm for twice continuously differentiable NLP problems," *COMPUTERS & CHEMICAL ENGINEERING,* Article vol. 34, no. 1, pp. 36-41, 2010 JAN 11 2010, doi: 10.1016/j.compchemeng.2009.09.006.

[8] R. Storn and K. Price, *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*. ICSI Berkeley, 1995.

[9] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–A comprehensive introduction," *Natural computing,* vol. 1, no. 1, pp. 3-52, 2002.

[10] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A Survey on Evolutionary Neural Architecture Search," *IEEE transactions on neural networks and learning systems,* Journal Article vol. PP, 2021 Aug 06 (Epub 2021 Aug 06) 2021, doi: 10.1109/TNNLS.2021.3100554.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science,* vol. 220, no. 4598, pp. 671-680, 1983.

[12] M. Antonio Cruz-Chavez, M. G. Martinez-Rangel, and M. H. Cruz-Rosales, "Accelerated simulated annealing algorithm applied to the flexible job shop scheduling problem," *INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH,* Article vol. 24, no. 5, pp. 1119-1137, 2017 SEP 2017, doi: 10.1111/itor.12195.

[13] X. Geng, J. Xu, J. Xiao, and L. Pan, "A simple simulated annealing algorithm for the maximum clique problem," *INFORMATION SCIENCES,* Article vol. 177, no. 22, pp. 5064-5071, 2007 NOV 15 2007, doi: 10.1016/j.ins.2007.06.009.

[14] N. Siddique and H. Adeli, "Simulated Annealing, Its Variants and Engineering Applications," *INTERNATIONAL JOURNAL ON ARTIFICIAL INTELLIGENCE TOOLS,* Article vol. 25, no. 6, 2016 DEC 2016, doi: 10.1142/S0218213016300015.

[15] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new Metaheuristic optimization algorithm," *Information Sciences,* vol. 540, pp. 131-159, 2020.

[16] Y. Jiang, Q. Luo, Y. Wei, L. Abualigah, and Y. Zhou, "An efficient binary Gradient-based optimizer for feature selection," *MATHEMATICAL BIOSCIENCES AND ENGINEERING,* Article vol. 18, no. 4, pp. 3813-3854, 2021 2021, doi: 10.3934/mbe.2021192.

[17] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances In Engineering Software,* vol. 95, pp. 51-67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[18] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, 1995, vol. 1: New York, NY, pp. 39-43.

[19] R. Cheng and Y. Jin, "A Competitive Swarm Optimizer for Large Scale Optimization," *IEEE TRANSACTIONS ON CYBERNETICS,* Article vol. 45, no. 2, pp. 191-204, 2015 FEB 2015, doi: 10.1109/TCYB.2014.2322602.

[20] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. Abd Elaziz, and S. Lu, "Improved salp swarm algorithm based on particle swarm optimization for feature selection," *JOURNAL OF AMBIENT INTELLIGENCE AND HUMANIZED COMPUTING,* Article vol. 10, no. 8, pp. 3155-3169, 2019 AUG 2019, doi: 10.1007/s12652-018-1031-9.

[21] M. R. Bonyadi and Z. Michalewicz, "Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review," *EVOLUTIONARY COMPUTATION,* Review vol. 25, no. 1, pp. 1-54, 2017 SPR 2017, doi: 10.1162/EVCO_r_00180.

[22] A. R. Yildiz and K. N. Solanki, "Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach," *INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY,* Article vol. 59, no. 1-4, pp. 367-376, 2012 MAR 2012, doi: 10.1007/s00170-011-3496-y.

[23] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-Assisted Cooperative Swarm Optimization of High-Dimensional Expensive Problems," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* Article vol. 21, no. 4, pp. 644-660, 2017 AUG 2017, doi: 10.1109/TEVC.2017.2675628.

[24] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, 1999, vol. 2: IEEE, pp. 1470-1477.

[25] M. Mahi, O. K. Baykan, and H. Kodaz, "A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem," *APPLIED SOFT COMPUTING,* Article vol. 30, pp. 484-490, 2015 MAY 2015, doi: 10.1016/j.asoc.2015.01.068.

[26] W. Deng, H. Zhao, L. Zou, G. Li, X. Yang, and D. Wu, "A novel collaborative optimization algorithm in solving complex optimization problems," *SOFT COMPUTING,* Article vol. 21, no. 15, pp. 4387-4398, 2017 AUG 2017, doi: 10.1007/s00500-016-2071-8.

[27] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation,* vol. 1, no. 1, pp. 67-82, 1997.

[28] M. P. Saka, O. Hasançebi, and Z. W. Geem, "Metaheuristics in structural optimization and discussions on harmony search algorithm," *Swarm and Evolutionary Computation,* vol. 28, pp. 88-97, 2016.

[29] A. Tzanetos and G. Dounias, "Nature inspired optimization algorithms or simply variations of metaheuristics?," *Artificial Intelligence Review,* pp. 1-22, 2020.
[30] S. Fong, X. Wang, Q. Xu, R. Wong, J. Fiaidhi, and S. Mohammed, "Recent advances in metaheuristic algorithms: Does the Makara dragon exist?," *The Journal of Supercomputing,* vol. 72, no. 10, pp. 3764-3786, 2016.
[31] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications,* vol. 27, no. 2, pp. 495-513, 2016/02/01 2016, doi: 10.1007/s00521-015-1870-7.
[32] F. Glover, "Tabu search—part I," *ORSA Journal on computing,* vol. 1, no. 3, pp. 190-206, 1989.
[33] M. Mitchell, J. H. Holland, and S. Forrest, "When will a genetic algorithm outperform hill climbing," in *Advances in neural information processing systems*, 1994, pp. 51-58.
[34] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization,* vol. 39, no. 3, pp. 459-471, 2007.
[35] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 26, no. 1, pp. 29-41, 1996.
[36] D. Zhao *et al.*, "Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy," *Knowledge-Based Systems,* p. 106510 (https://doi.org/10.1016/j.knosys.2020.106510), 2020/10/22/ 2020, doi: https://doi.org/10.1016/j.knosys.2020.106510.
[37] D. Zhao *et al.*, "Ant Colony Optimization with Horizontal and Vertical Crossover Search: Fundamental Visions for Multi-threshold Image Segmentation," *Expert Systems with Applications,* p. 114122, 2020.
[38] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems,* vol. 111, pp. 300-323, 2020/04/03/ 2020, doi: https://doi.org/10.1016/j.future.2020.03.055.
[39] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems,* vol. 97, pp. 849-872, 2019/08/01/ 2019, doi: https://doi.org/10.1016/j.future.2019.02.028.
[40] S. Song *et al.*, "Dimension decided Harris hawks optimization with Gaussian mutation: Balance analysis and diversity patterns," *Knowledge-Based Systems,* p. 106425 (https://doi.org/10.1016/j.knosys.2020.106425), 2020/10/13/ 2020, doi: https://doi.org/10.1016/j.knosys.2020.106425.
[41] E. Rodriguez-Esparza *et al.*, "An efficient Harris hawks-inspired image segmentation method," *Expert Systems with Applications,* vol. 155, p. 113428, 2020.
[42] C. L. C. Villalón, T. Stützle, and M. Dorigo, "Grey Wolf, Firefly and Bat Algorithms: Three Widespread Algorithms that Do Not Contain Any Novelty," in *International Conference on Swarm Intelligence*, 2020: Springer, pp. 121-133.
[43] K. Sörensen, "Metaheuristics—the metaphor exposed," *International Transactions in Operational Research,* vol. 22, no. 1, pp. 3-18, 2015.
[44] M. A. Lones, "Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms," *SN Computer Science,* vol. 1, no. 1, pp. 1-12, 2020.
[45] M. Levi, *Classical mechanics with calculus of variations and optimal control: an intuitive introduction*. American Mathematical Soc., 2014.
[46] S. Paterlini and T. Krink, "High performance clustering with differential evolution," 2004, vol. 2: IEEE.

[47] J. C. Lai, F. H. Leung, and S.-H. Ling, "A new differential evolution with wavelet theory based mutation operation," in *2009 IEEE Congress on Evolutionary Computation*, 2009: IEEE, pp. 1116-1122.

[48] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software,* vol. 69, pp. 46-61, 2014.

[49] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based systems,* vol. 96, pp. 120-133, 2016.

[50] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association,* vol. 32, no. 200, pp. 675-701, 1937.

[51] F. Van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information sciences,* vol. 176, no. 8, pp. 937-971, 2006.

[52] N. H. Awad, Ali, M. Z., Suganthan, P. N., Liang, J. J., & Qu, B. Y., "Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single

Objective Real-Parameter Numerical Optimization," *2017 IEEE Congress on Evolutionary Computation (CEC),* 2017.

[53] O. J. Dunn, "Multiple comparisons among means," *Journal of the American statistical association,* vol. 56, no. 293, pp. 52-64, 1961.

[54] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics,* pp. 65-70, 1979.

[55] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research,* vol. 7, no. Jan, pp. 1-30, 2006.

[56] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing,* vol. 62, pp. 1019-1043, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.asoc.2017.09.039.

[57] N. Kumar, I. Hussain, B. Singh, and B. K. Panigrahi, "Single sensor-based MPPT of partially shaded PV system for battery charging by using cauchy and gaussian sine cosine optimization," *IEEE Transactions on Energy Conversion,* vol. 32, no. 3, pp. 983-992, 2017.

[58] A. A. Heidari, R. A. Abbaspour, and H. Chen, "Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training," *Applied Soft Computing,* vol. 81, p. 105521, 2019.

[59] H. Chen, C. Yang, A. A. Heidari, and X. Zhao, "An efficient double adaptive random spare reinforced whale optimization algorithm," *Expert Systems with Applications,* p. 113018, 2019/10/13/ 2019, doi: https://doi.org/10.1016/j.eswa.2019.113018.

[60] J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, and C. Li, "Multi-strategy boosted mutative whale-inspired optimization approaches," *Applied Mathematical Modelling,* vol. 73, pp. 109-123, 2019/09/01/ 2019, doi: https://doi.org/10.1016/j.apm.2019.03.046.

[61] A. A. Heidari, I. Aljarah, H. Faris, H. Chen, J. Luo, and S. Mirjalili, "An enhanced associative learning-based exploratory whale optimizer for global optimization," *Neural Computing and Applications,* pp. 1-27, 2019.

[62] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE transactions on evolutionary computation,* vol. 10, no. 3, pp. 281-295, 2006.

[63] H. Liang, Y. Liu, Y. Shen, F. Li, and Y. Man, "A hybrid bat algorithm for economic dispatch with random wind power," *IEEE Transactions on Power Systems,* vol. 33, no. 5, pp. 5052-5061, 2018.

[64] B. Adarsh, T. Raghunathan, T. Jayabarathi, and X.-S. Yang, "Economic dispatch using chaotic bat algorithm," *Energy,* vol. 96, pp. 666-675, 2016.

[65] J. Alcala-Fdez *et al.*, "A software tool to assess evolutionary algorithms for data mining problems. J," *Multiple-Valued Logic Soft Comput,* vol. 17, pp. 2-3, 2011.

[66] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering,* vol. 191, no. 11-12, pp. 1245-1287, 2002.

[67] J. Liu, C. Wu, G. Wu, and X. Wang, "A novel differential search algorithm and applications for structure design," *Applied Mathematics and Computation,* vol. 268, pp. 246-269, 2015.

[68] J. S. Arora, *Introduction to optimum design*. Elsevier, 2004.

[69] A. D. Belegundu and J. S. Arora, "A study of mathematical programming methods for structural optimization. Part I: Theory," *International Journal for Numerical Methods in Engineering,* vol. 21, no. 9, pp. 1583-1599, 1985.

[70] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry,* vol. 41, no. 2, pp. 113-127, 2000.

[71] F.-z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and computation,* vol. 186, no. 1, pp. 340-356, 2007.

[72] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering applications of artificial intelligence,* vol. 20, no. 1, pp. 89-99, 2007.

[73] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied mathematics and computation,* vol. 188, no. 2, pp. 1567-1579, 2007.

[74] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *International Journal of General Systems,* vol. 37, no. 4, pp. 443-473, 2008.

[75] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: a new metaheuristic optimization algorithm," *Computers & Structures,* vol. 139, pp. 98-112, 2014.

[76] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with computers,* vol. 29, no. 1, pp. 17-35, 2013.

[77] S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software,* vol. 83, pp. 80-98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[78] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Applied Soft Computing,* vol. 13, no. 5, pp. 2592-2612, 2013.

[79] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems,* vol. 89, pp. 228-249, 2015/11/01/ 2015, doi: https://doi.org/10.1016/j.knosys.2015.07.006.

[80] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences,* vol. 178, no. 15, pp. 3043-3074, 2008.

[81] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing,* vol. 10, no. 2, pp. 629-640, 2010.

[82] S. Gold and S. Krishnamurty, "Trade-offs in robust engineering design," in *Proceedings of DETC*, 1997, vol. 97, p. 1997.

[83] G. G. Wang, "Adaptive response surface method using inherited latin hypercube design points," *Journal of Mechanical Design,* vol. 125, no. 2, pp. 210-220, 2003.

[84] V. T. Chow and G. Cortes-Rivera, "Application of DDDP in water resources planning," 1974.

[85] A. Samadi-koucheksaraee, I. Ahmadianfar, O. Bozorg-Haddad, and S. A. Asghari-pari, "Gradient Evolution Optimization Algorithm to Optimize Reservoir Operation Systems," *Water Resources Management,* 2018/10/26 2018, doi: 10.1007/s11269-018-2122-2.

[86] O. B. Bozorg-Haddad, A. Afshar, and M. A. Mariño, "Multireservoir optimisation in discrete and continuous domains," *Proceedings of the Institution of Civil Engineers-Water Management,* vol. 164, no. 2, pp. 57-72, 2011.

[87] O. Bozorg-Haddad, M. Janbaz, and H. A. Loáiciga, "Application of the gravity search algorithm to multi-reservoir operation optimization," *Advances in water resources,* vol. 98, pp. 173-185, 2016.

[88] I. Garousi-Nejad, O. Bozorg-Haddad, and H. A. Loáiciga, "Modified firefly algorithm for solving multireservoir operation in continuous and discrete domains," *Journal of Water Resources Planning and Management,* vol. 142, no. 9, p. 04016029, 2016.

[89] O. B. Haddad, A. Afshar, and M. A. Mariño, "Multireservoir optimisation in discrete and continuous domains," in *Proceedings of the Institution of Civil Engineers-Water Management*, 2011, vol. 164, no. 2: Thomas Telford Ltd, pp. 57-72.

[90] O. B. Haddad, M. Moravej, and H. A. Loáiciga, "Application of the water cycle algorithm to the optimal operation of reservoir systems," *Journal of Irrigation and Drainage Engineering,* vol. 141, no. 5, p. 04014064, 2014.

[91] H.-R. Asgari, O. Bozorg Haddad, M. Pazoki, and H. A. Loáiciga, "Weed optimization algorithm for optimal reservoir operation," *Journal of Irrigation and Drainage Engineering,* vol. 142, no. 2, p. 04015055, 2015.

[92] I. Ahmadianfar, A. Adib, and M. Salarijazi, "Optimizing multireservoir operation: Hybrid of bat algorithm and differential evolution," *Journal of Water Resources Planning and Management,* vol. 142, no. 2, p. 05015010, 2015.

[93] R. Moeini and F. Soghrati, "Optimum outflow determination of the multi-reservoir system using constrained improved artificial bee colony algorithm," *Soft Computing,* vol. 24, no. 14, pp. 10739-10754, 2020.

[94] D. M. Murray and S. J. Yakowitz, "Constrained differential dynamic programming and its application to multireservoir control," *Water Resources Research,* vol. 15, no. 5, pp. 1017-1027, 1979.

[95] I. Ahmadianfar, A. Samadi-Koucheksaraee, and O. Bozorg-Haddad, "Extracting Optimal Policies of Hydropower Multi-Reservoir Systems Utilizing Enhanced Differential Evolution Algorithm," *Water Resources Management,* vol. 31, no. 14, pp. 4375-4397, 2017.

[96] M. Mohammadi, S. Farzin, S.-F. Mousavi, and H. Karami, "Investigation of a new hybrid optimization algorithm performance in the optimal operation of multi-reservoir benchmark systems," *Water Resources Management,* vol. 33, no. 14, pp. 4767-4782, 2019.

[97] M. Moravej and S.-M. Hosseini-Moghari, "Large scale reservoirs system operation optimization: the interior search algorithm (ISA) approach," *Water Resources Management,* vol. 30, no. 10, pp. 3389-3407, 2016.

[98] I. Ahmadianfar, A. Adib, and M. Salarijazi, "Optimizing multireservoir operation: hybrid of bat algorithm and differential evolution," *Journal of Water Resources Planning and Management,* vol. 142, no. 2, p. 05015010, 2016.